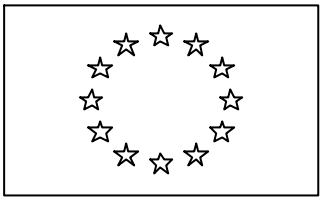

**REPORT**

Guidelines for the Use of

# Names and Keys

in a Global TTP Infrastructure

by

**Prof. Dr. Hartmut Pohl**

**ISIS – InStitute for Information Security**

Essen, Germany

This work has been funded by the European Commission.
The opinions and views expressed may not represent official opinions and policies of the E.C.

Many independent contributions, published elsewhere on the Internet or in print were extremely important and helpful to this study. The individuals, agencies, institutions, task forces etc. listed in the chapter "references" were the primary sources of our work. The inspiration we received from these publications can not be emphasised enough and is gratefully acknowledged. We would like to thank all the authors for their significant work. Because of the wealth of information from all these different sources we refrained to mark all the quotations, especially when borrowing text from standards.

# Table of Contents

# List of Figures

# Foreword

There is an almost unanimous agreement that today's society has started to become an information society, i.e. a society where information and knowledge play a major role in everyday´s life. This development would not have been possible without the pervasive nature of modern communication and information systems and their rapid deployment. The near future will see that these systems will be accessible from almost every location on earth. The history of the Internet and its growth rate in size as well as in applications provides an excellent example of the dynamics of such a development.

Electronic commerce will be one of the major application areas for the potential inherent in these systems. National as well as international trade will increasingly be conducted electronically over global accessible networks. The logical and very often personal business relations will manifest themselves in a physical information infrastructure which will become the backbone for trading activities. Therefore conventional methods for establishing and maintaining trust between trading partners need to be adjusted to the modern technologies.

The European Commission has started preparatory work to address these problems the results of this study are intended to be one of the inputs for the concept of a future European Trust Infrastructure.

# Executive Summary

## Introduction

The use and exchange of information in electronic form is the key feature of a European and Global Information Infrastructure. The satisfactory operation will be highly dependent on the ease with which users are able to identify and address correspondents. For this reason it is important to be able to establish the electronic equivalent for the identity of individual users, organisational units and organisations from which the information originated and to which the information is sent.

Modern cryotographic mechanisms allow entities who were previously unknown to each other to establish trust relationships and to communicate trusted with each other without establishing a trusted communication path beforehand by distributing public keys.

When applying public key cryptography the standard method for guaranteeing the authenticity of a public key is to use a Trusted Third Party (TTP) called a certification authority (CA). The certification authority guarantees the authenticity of an entity's public key by digitally signing and publishing a certificate which allows communicating parties to trust a public key and to securely conduct electronic transactions using that public key.

In the case of digital signatures certificates certify the public key used for verifying the digital signature for a document, they represent the strong binding between the name of a person and its public key. Such certificates allow parties to trust in the authenticity of the digital signature - if the parties trust the certificate generating authority.

For the generation of the certificate it is necessary to find ways of assigning and managing names which are sufficiently detailed for the purposes intended. The trustworthy binding between a name and its legitimate owner and name and its public key is an essential prerequisite for the functioning of and for the trust in a global TTP infrastructure.

## Aim

Currently several approaches exist for the establishment of trust infrastructures which are based on the use of public key cryptography as a tool for providing trust. Some of these approaches are only proposals, some are trial projects and some are currently in transition to a limited operational phase. Most of the proposals and projects are based on available standards. The experience documented by the projects and the concepts of the proposals can be a valuable input for the European Commission in its deliberations towards a European Trust System.

This study covers certificates and names, two aspects which are of significant concern if concepts for a trusted infrastructure are being developed. The approaches to certificates and names in the different projects and proposals will be described and the experience assessed with the aim to derive recommendations for the work on a European Trust System. As the whole area is rather dynamic, it has to be mentioned that the study reflects the status of knowledge in the first quarter of 1997.

## Project Stages

The work for this study has been performed in three stages. In stage 1 projects and proposals have been selected which have been determined to be relevant for the subject of this study. The analysis has been based on available literature from a multitude of sources.

Special consideration has been given to the following aspects:

- Certification of signatures (method, process),
- Certificate semantics and format,
- Certificate management (issuance, revocation, renewal etc.),
- Naming requirements and principles to be addressed by the directory services,
- Directory management and possible extensions to the standard X.509.

The results of stage 1 and their assessment have been described in chapter 2 „Certificates and Names in Trust Infrastructures." Based on the results of the analysis and their assessment, recommendations have been produced in stage 2 which are documented in chapter 3 „Certificates and Names in a European Trust Infrastructure - ETI". In stage 3, guidelines have been drafted that are intended to support handling of certificates and names in a standardised way, they are documented at Appendix C and D.

## Results

The study has confirmed that there is a multitude of projects and proposals which cover concepts for trust infrastructures based on the use of certificates. Almost all of them employ existing standards.

X.509 is the standard used for certificates and certificate revocation lists. After the improvements added to the original X.509 standard, the current version 3 (X.509 v3) has received widespread recognition as being feasible and mature enough for application in trust infrastructures. However the broad spectrum of features which are contained in the standard call for deliberations whether profiles should be developed which are targeted specifically at a European Trust Infrastructure. If this should be determined to be sensible, care has to be taken not to prohibit interoperability with other infrastructures. The possibility to define private extensions for the certificates which contributes to their flexible use has the inherent danger, that when this features are used without care, the certificates may become unintelligible. Current discussions in the Internet PKIX working group show that details of the standard are still in discussion but this does not prevent the general acceptance.

The management of certificates needs a repository where certificates and associated information can be stored, distributed and accessed easily. The concept of a Directory as specified in X.500 has been developed to support such a requirement. Again, almost all projects and proposals use such a Directory. Having been criticised heavily at its inception because of its claim to be a global repository for all kinds of information the Directory concept has found recognition in the meantime. One of the major arguments still used against such a Directory are reasons of privacy because there is a fear that too much information about a person or an organisation could be revealed to the public. The possibility to structure the database of a Directory in a way to allow for distributed (decentralised) management of the information base is seen as a major advantage of the concept. Products that support the X.500 concept are available on the market.

The problem of identifying partners, previously unknown, in a distributed system and the need to bind public keys to names of individuals call for clearly defined conventions for naming. In almost all the projects and proposals naming conventions as specified in X.520 are used. The Distinguished Names proposed in that concept allow to identify an individual uniquely and additionally they provide information about his position in an organisational structure. Distinguished Names are claimed to be cumbersome in handling, however this seems to be more a problem of the technical interface than of the name itself. A disadvantage of Distinguished Name is that they represent the structure of the Directory database and therefore in most cases the structure of the organisation which is mapped into the database. This restricts flexibility and calls for a very careful design of the database structure in order to minimise later modifications as far as possible.

An ongoing discussion relates to local names. Initiated by the perception that a global directory might be too complex to ever become reality, global unique names may be too difficult to realise, and organisations have a natural need to assign names for their own area of responsibility, the use of local names has been proposed. These names basically have a local area of validity which however can be extended to the outside if needed. In contrary to the X.500 approach this would be a bottom up approach for a directory system.

In summary, the study has shown that the current available standards are a good basis to build a European Trust Infrastructure on. Minor modifications to the standards however are recommended in the following paragraph and more detailed in paragraph 3 of this study. Especially the work for the US Federal Public Key Infrastructure (FPKI) which, from its dimension, can be compared somehow with a European Trust Infrastructure and which is described on purpose in more detail in this study than the other projects is a good example of the work necessary to develop the concept for a trusted infrastructure.

## Recommendations

To establish a concept for a European Trust Infrastructure the following general recommendations, which are described in more detail in chapter 3 of the study and which are seen from the perspective of certificates and names, are made:

- to base a European Trust Infrastructure on several policy approving authorities (PAA), which co-operate on commonly agreed principles. The PAAs might be established by Member States, by business sectors, by administrations or on another basis deemed to be applicable. The number of PAAs is not limited to one in the European Union or to one by Member State,
- to develop a certification structure that combines the advantages hierarchical and networked topologies,
- to base the application of certificates and certificate revocation lists on the concepts defined in X.509 v3. This standard allows for a broad spectrum of implementations, which can restricted if needed by defining application specific profiles,
- to base the management of certificates, certificate revocation lists, keys and names on a Directory based on concepts specified in X.500,
- to base the naming of the participating entities on the concepts of X.520 and as far as family names are concerned on the proposal of the World Electronic Messaging Association (WEMA),
- to follow the developments in the Simple Distributed Security Infrastructure (SDSI) project carefully in order to allow for a simple use of local names,
- to provide mechanisms for interoperability of certificates and names which are implemented technically and relieve the user from being responsible for interoperation.

# 1 Project Overview

## 1.1 Background

As a result of the work which has been carried out in the „Security Investigations“ under the auspices of the Senior Officials Group Information Security (SOG-IS) and funded by the European Commission under the „Council Decision in the field of security of Information Systems“ of 31 March 1992 and other projects performed on behalf of the European Commission there was broad agreement that the security of the information technology systems (IT-systems) will have a major impact on the evolution of a European Information Society and on its acceptance by industry, governments, and the general public.

Whereas a lot of work has been done in the Security Investigations to improve the security in the end-user area and to make people aware of threats to IT-systems, the work under a holistic view (secure co-operation of information- and communications systems) has only just started. However it has become evident in the work done, that the application of the concept of trust, that is the confidence that all co-operating entities in an application adhere to a given security policy, is essential for the operation of an information infrastructure.

Looking at such an infrastructure which will interconnect industry, governments and most probably the general public of different Member States, it seems feasible to develop an infrastructure which provides a defined set of trust services which are accepted by and can be used in all Member States and which are flexible enough to allow co-operation on an international level.

This study looks at one part of the problem by providing the necessary inputs for the establishment of a concept for the use of public key certificates and for the name assignment and management. It is seen as a contribution to the development of a European Trust Infrastructure, whose technical components will be developed elsewhere. It will also provide an assessment whether the existing standards in this area are sufficient and if not, which modifications are seen to be necessary.

## 1.2 Approach

This study borrows heavily from the work that has been done or is currently being done in the area of public key infrastructures and which has a strong relation to the problems associated with certificates and names and it uses the results whenever possible. There has been no special phase of establishment of user requirements, user requirements - as far as they have been considered as being necessary - have been taken from the available literature.

The following issues have particular significance in this context and are guidance for the work:

- Certification of signatures (method, process),
- Certificate semantics and format,
- Certificate management (issuance, revocation, renewal etc.),
- Naming requirements and principles to be addressed by the Directory services,
- Directory management and possible extensions to the standard X.509,
- Standards.

At the time of writing of this report (first quarter of 1997) it was not possible to find a commonly agreed definition of what European Trust Services will be and by what kind of architecture they will be supported. In order to have a framework for discussion, a working model of a European Trust Infrastructure has been developed. This model consists of three logical layers which together could represent a European Trust Infrastructure (ETI). The advantage of that view is that it allows each application (e.g. electronic payment, healthcare, electronic tendering etc.) - placed on top of the three layers - to select the service, trust center and mechanism which serves its purpose best.

The discussion of certificates and names is based on the analysis of certification and naming schemes which are currently implemented or might be implemented in the near future. An assessment is made whether they are sufficient enough for an application within a European and Global trust infrastructure as envisioned by the working model or whether additional efforts are required for an eventual modification of these schemes.

The results of the assessment will be used to structure and formulate guidelines on the use and administration of certificates and names.

# 2 Certificates and Names in Trust Infrastructures

## 2.0 Introduction

### 2.0.1 The Need for Trust

Recent years have witnessed a growth of computer use to the point that, in many countries, every individual and each company and agency is an actual or potential user of computer and communication networks. This leads to a situation that business, administration, information and entertainment services can and will increasingly be provided by electronic means. The consequences of this could change the way business and government operate and how we communicate with our colleagues, families and friends.

In former times and until today in the physical world we had the following ways to distribute messages trustworthy.

| Traditional Levels of Trust | |
|---|---|
| **Level** | **Description** |
| 0 | Message exchange by postcard (present mail). |
| 1 | Message by regular letter (present mail). |
| 2 | Letter with hand-written signature (present mail). |
| 3 | Usage of tamper resistant paper in a letter (present mail). |
| 4 | Letter inside a sealed envelope (present mail). |
| 5 | Message by registered letter (present mail). |
| 6 | Notary-certified hand-written signature on a letter (present mail). |
| 7 | Message exchange by a human courier. |
| ... | ... |

Figure 1: Traditional Levels of Trust

At the same time the threats to the security of information are growing because of the diversification and multiplication of communication services and their lacking security features. Because of the increasing dependency from information systems the trustworthiness and protection of information is essential for the functioning of a modern business and even society. A lack of user confidence will undermine the rapid development of the information society and will prevent to take advantages of the related incentives. Therefore it is necessary that the security of information systems evolves to reduce the threats to security and privacy while avoiding to obstruct innovation or economic and social developments. Traditional techniques of securing information, such as signatures, envelopes, registration, sealing, depositing and special delivery need therefore to be matched by electronic equivalents.

As the information exchange becomes global, and the interrelationship between the different actors tighter, it is necessary that theses techniques will be developed for European-wide application with the option for world wide interoperability.

### 2.0.2 Security Services and Functions

The general scenario is where one user wishes to exchange information (for example personal e-mail, electronic commerce transactions, a client/server session) with another user and where the information exchange may require either one or a combination of the following security services:

- Identification and authentication of the user entities.
- Proof of origination of the message.
- Support of integrity of the message.
- Confidentiality for the message.
- Non-repudiation of a received message.

These services may be supported by the following security functions:

- Key management with generation, distribution, revocation, archival, storage and retrieval, reconstruction (recovery), certification, recertification (renewal) and protection and deletion of services for symmetric cryptosystems and services for asymmetric cryptosystems.
- Access control support services.

Some additional functions may be necessary:

- Claimant/verifier exchanges.
- Evidence generation, recording and verification.
- Dispute resolution.
- Time stamping.
- Audit.
- Delivery.

There may be other security services of a trust infrastructure like a key escrowing service and a backup (recovery) service for example for software, documents, data, keys, etc. All these services and functions contribute to the trust that the user can have into the information exchange process itself and into the proper handling of the information exchanged.

### 2.0.3 Establishing and Managing Trust

Currently communication partners are able to exchange keys that allow for trusted communications. But they are not in the position to authenticate each other – especially in an open infrastructure with communication partners previously unknown. In an open and large-scale network, it is impractical and unrealistic to expect each user to have previously established personal or physical relationships with all the expected communication partners.

To establish a trusted communication path in a modern communication infrastructure, a sender of an object (a message or a data file) must be able to identify and authenticate the receiver (the business or communication partner) reliably without having to meet him personally in order to trust him.

The generally accepted way today to authenticate a new communication partner or a new receiving entity is, to authenticate him by a third authority or party. This concept allows that two individuals implicitly trust each other although they have not previously established a personal relationship. In this scenario, the guarantee for the correct identity is provided by a third party that assures to each of the communication partners that the other partner is authentic. Such a party which has to be trusted by the entities participating in the information exchange is called a Trusted Third Party (TTP). Trusted Third parties can be established independently for different applications, business sectors or geographical regions, however there is a need for co-operation if an information exchange between these different areas is required like for example for electronic commerce. The technical basis for this co-operation is called a trust infrastructure.

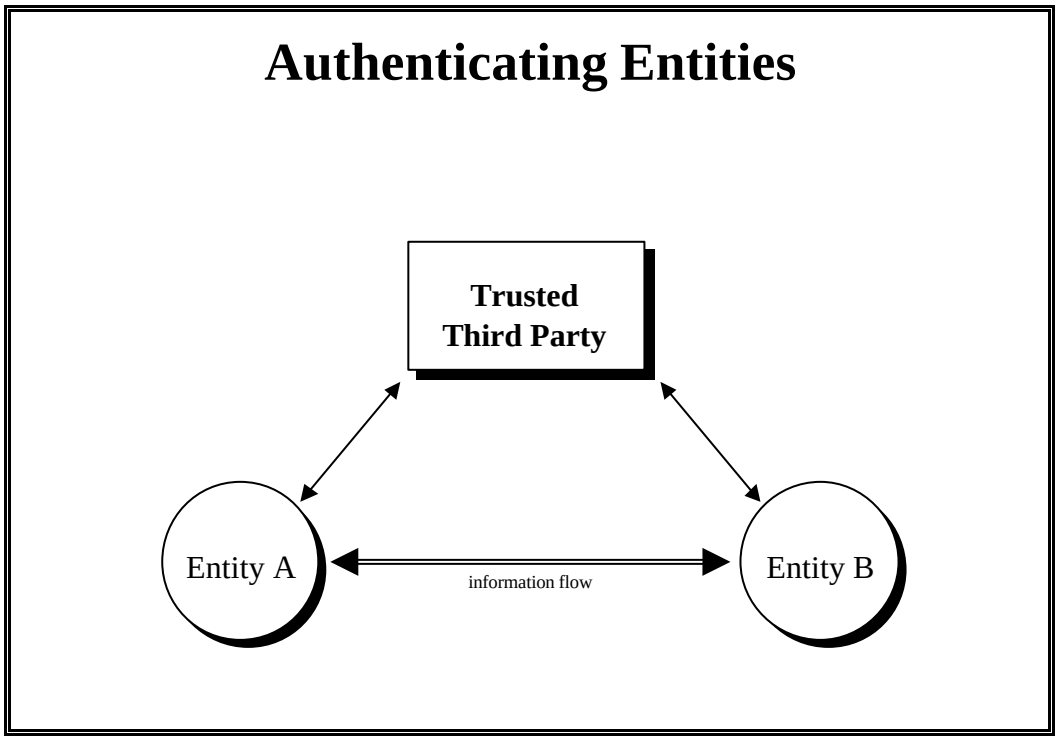


Figure 2: Authenticating Entities

## 2.1 Definition of a European Trust Infrastructure

At the time of writing of this report there exists no commonly agreed definition of what a European

Trust System (ETS) is. However there is currently work going on in several projects sponsored by the European Commission, which will help to arrive at such a definition.

The minimum functions of a trust infrastructure are authorities issuing, storing and distributing keys and certifying these keys; and features necessary to generate trust in these authorities.

These functions need to be provided to the different applications in an uniform manner and with a high level of trust independent from a geographic area, that is, areas of different trust are not acceptable. Therefore the ETS is not directed towards the support of one specific application. On the contrary it will provide a range of services that can be used in different application scenarios and these services may be provided with different levels of trust over Europe.

The entities which are necessary to deliver such trusted services and the connectivity between them can be viewed as a special infrastructure on top of the communications infrastructure. It is this special infrastructure - in the European area - which for the conduct of this work will be considered as the European Trust System.

The European Trust Infrastructure may change the world from the historical physical world, where identities of persons were considered necessary, to a digital world, where we increasingly interact with entities like people or companies we never met before or will meet – with whom we have no relationship in the physical world. Therefore a transfer of the concepts from the physical world to the digital world may be meaningful only historically. In such a digital world it is not the most important task to identify a person but to bind documents, authorisations or whatever to a digital key.

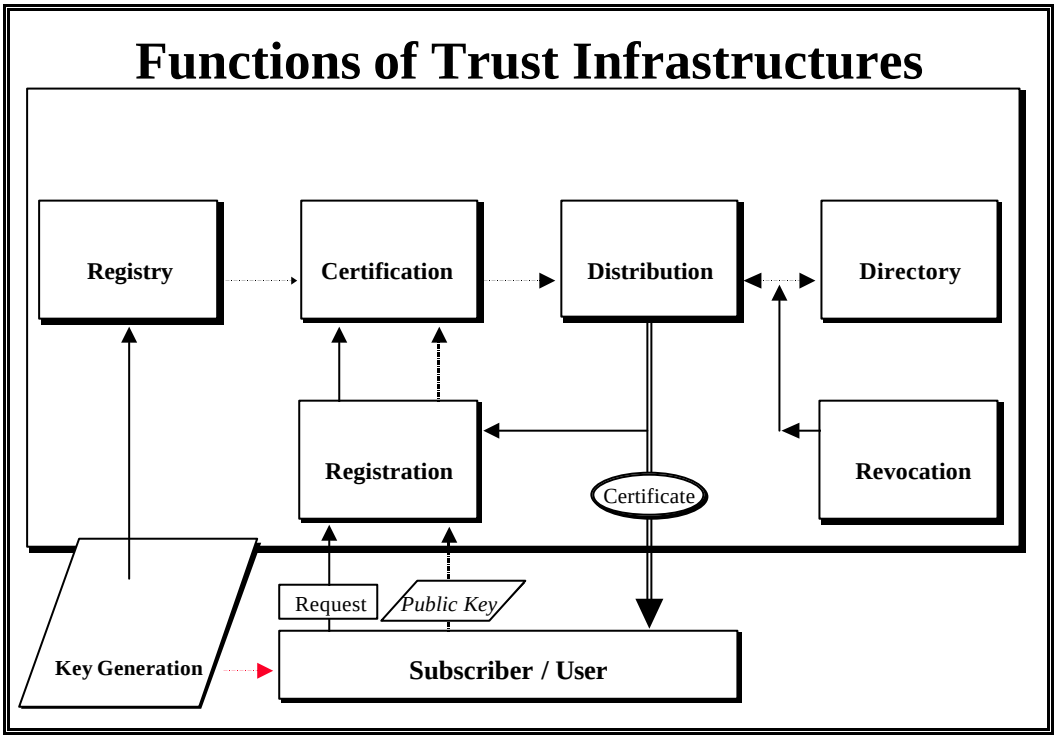


Figure 3: Functions of Trust Infrastructures

Such an infrastructure, at the logical level, can be seen as a distributed database of public key certificates and further information (e.g. revocation lists).

### 2.1.1 Users of a Trust Infrastructure

Users of a trust infrastructure could be:

- Private individuals conducting private business – e.g. as customers of a bank.
- Staff within a commercial organisation exchanging internal business-related traffic.
- Customers being supported by a TTP in doing business with the TTP provider or associated company (e.g. bank customers doing electronic banking, with the bank's TTP providing the support).
- Customers that have asked their Internet Service Provider (ISP) to provide them with a Virtual Private Network (VPN).
- Commercial organisations exchanging electronic commerce traffic of a wide variety (general e-mail, enquiries, proposals, orders and advices, reports, bills, payments, credit notes, etc.).
- Individual professionals practising a licensed profession (e.g. solicitors, notary publics, doctors, etc.).

### 2.1.2 The Communication Infrastructure

For the purpose of this study it is assumed that there is an existing communications infrastructure which however does not provide the required integrity and confidentiality. In order to be able to fully exploit

the advantages of modern technology the users of this communication infrastructure have to be provided with features which allow for a trusted operation of a multitude of applications, like for example:

- Teleworking,
- Traffic management,
- Healthcare,
- Electronic tendering,
- Electronic trade,
- Electronic payment,

and the supporting functions like,

- Secure mail,
- Secure file transfer, secure hypertext transfer, and
- Secure database (directory) access.

## 2.2 Cryptography and its Applications

The basic tool for providing trust is cryptography. Cryptography is a discipline that embodies the principles, means, and methods for the transformation of data in order to hide their semantic content, prevent their unauthorised use, or prevent their undetected modification rendering plaintext unintelligible and for converting encrypted messages into intelligible form.

### 2.2.1 Traditional Secret Key Cryptography

Traditional cryptography is based on the sender and receiver of a message knowing and using the same secret key: The sender uses the secret key to encrypt the message, and the receiver uses the same secret key to decrypt the message. This method is known as secret key cryptography or symmetric cryptography because both communication partners, sender and receiver, use the same key, which has to be kept secret. The main problem is getting the sender and receiver to agree on the secret key without anyone else finding it out. If they are in a separate physical location, they must trust a courier, or a phone system, or some other transmission medium to prevent the disclosure of the secret key being communicated. Anyone who overhears or intercepts the key in transit can later read, modify, and forge all messages encrypted or authenticated using that key.

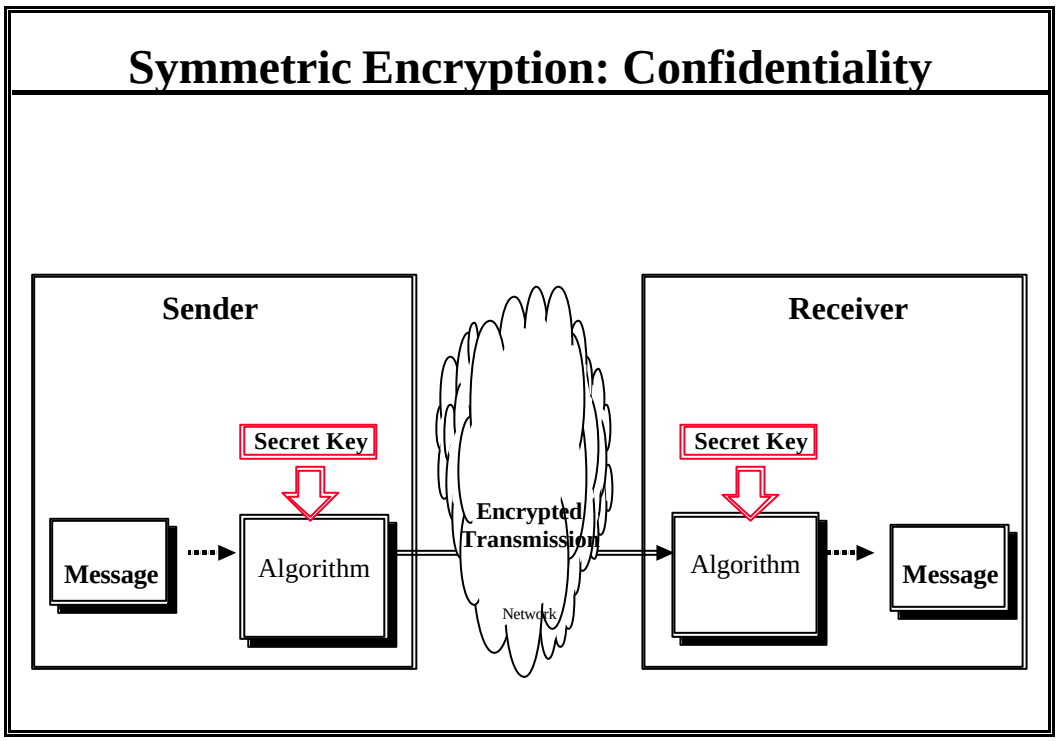


Figure 4: Symmetric Encryption: Confidentiality

The generation, transmission and storage of keys is called key management. All crypto systems must deal with key management issues. Because all keys in a secret key crypto system must remain secret, secret key cryptography often has difficulties providing secure key management, especially in open systems with a large number of users.

Examples for symmetric algorithms are: DES, DEA, Triple-DES, IDEA, FEAL, RC2, RC4.

### 2.2.2 Public Key Cryptography

The concept of public key cryptography was introduced in 1976 by Diffie and Hellman in order to solve

the key management problem. In their concept, each person gets a pair of keys, one called the public key and the other called the private key. Each person's public key is published while the private key has to be kept secret. The need for the sender and receiver to share secret information is eliminated: All communications involve only public keys, and no private key is ever transmitted or shared. No longer is it necessary to trust some communications channel to be secure against eavesdropping or betrayal.

The only requirement is that public keys are associated with their users in a trusted (authenticated) manner (for instance, in a trusted directory). Anyone can send a confidential message by just using public information, but the message can only be decrypted with a private key, which is in the sole possession of the intended recipient. Furthermore, public key cryptography can be used not only for authentication by using digital signatures but also for privacy by encrypting messages.

Examples for asymmetric algorithms are: RSA, Diffie-Hellman, ElGamal.

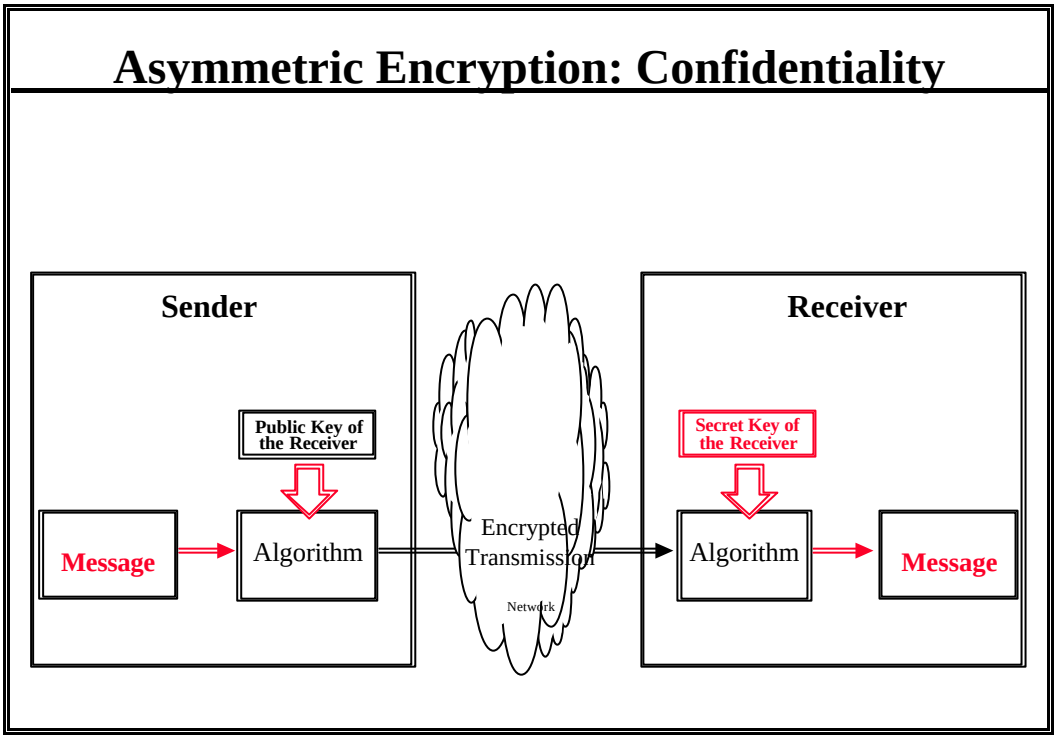


Figure 5: Asymmetric Encryption: Confidentiality

When a sender Alice wishes to send a secret message to a receiver Bob, Alice looks up Bob´s public key in a directory, uses it to encrypt the message and sends it off. Bob then uses his private key – which he keeps and stores secret, and therefore only he knows - to decrypt the message and to read it. No one listening in can decrypt the message. Anyone can send an encrypted message to Bob but only Bob can read it. Clearly, one requirement is that no one can figure out the private key from the corresponding public key.

**Advantages**

In a secret key system the secret keys must be transmitted (either manually or through a communication channel), and there may be a chance that an enemy can discover the secret keys during their transmission.

The primary advantage of public key cryptography is increased security and convenience: Private keys, by contrast, never need to be transmitted or revealed to anyone. The major advantage of public key cryptography techniques, compared to conventional cryptographic techniques, is their asymmetry: while only an entity knowing an appropriate secret key can perform a certain operation (e.g. decrypt or sign a message), everyone knowing the corresponding public key can perform a corresponding operation (e.g. encrypt a message).

The other major advantage of public key systems is providing a method for digital signatures.

Technically a Trusted Third Party certifies the validity of the public key by digitally signing the combination of the public key and some identity information like the name of the entity. The process is called binding the public key to the entity. The signed combination of the key and the name of ist holder is called a certificate.

### 2.2.3 The Digital Signature Process

#### 2.2.3.1 Signature Key

Digital signatures are used for sender-authentication, non-repudiation and message integrity purposes.

In order for a user to trust these security services the user needs to be assured that the public key used to verify a signature is actually the key of the person who signed the transaction.

This verification process is done by an authority signing the combination of public key and name of the key holder and the result is documented in a certificate. The importance of a certificate requires that certificates should be generated by and obtained from trusted sources.

Authentication via secret key systems requires the sharing of some secrets. In secret key systems a sender can repudiate a previously authenticated message by claiming that the shared secret was somehow compromised by one of the parties sharing the secret.

Public key authentication, on the other hand, prevents this type of repudiation. Each user has sole responsibility for protecting his private key. This property of public key authentication is often called non-repudiation

The traditional business process is based on paper and hand-written signatures. As our information society begins to rely more and more on electronic data processing and interchange, an electronic equivalent to hand-written signatures is needed in order to allow for fully electronic business processes. In general, hand-written signatures are easy to forge and hard to verify. Digital signatures based on modern public key technology are hard to forge and easy to verify. However, digital signatures introduce new qualities that cause some uneasiness:

- You can use several digital signatures at a time,
- you may change your signature,
- your signature may be "broken",
- your signature may be performed by a device or a program.

In order to obtain general recognition of digital signatures in open systems a management infrastructure is needed which addresses trusted processes like the user registration, certification, etc. of digital signatures or more precisely the public keys.

To sign a message, Alice does a computation involving both her private key and the message itself. The output is called the digital signature and is attached to the message, which is then sent. Bob, to verify the signature, does some computation involving this message, the purported signature, and Alice's public key. If the result property holds in a simple mathematical relation, the signature is verified as being genuine. Otherwise, the signature may be fraudulent or the message might have been altered.

A digital signature attests to the contents of a message as well as to the identity of the signer. There is no way to take someone's signature from one document and attach it to another, or to alter a signed message in any way. The slightest change in a signed document will cause the digital signature verification process to fail. Thus, public key authentication allows people to check the integrity of signed documents. If a signature verification fails, however, it will generally difficult to determine whether there was an attempted forgery or simply a transmission error.

RSA and DSA are the most popular signature algorithms used. The RSA algorithm is named after it's inventors Rivest, Shamir, and Adleman. The Digital Signature Algorithm (DSA) was developed by the U.S. Government.

#### 2.2.3.2 Hash Functions

Actually the digital signature does not depend on the whole document, but on the hashed document only. A hash function is a transformation that takes a variable size (long) input and returns a fixed size (short) string, which is called the hash value of the document or the thumbprint. Thus signing and verifying long documents becomes quicker than encrypting the whole document.

The basic requirements for a cryptographic hash function are:

- The input can be of any length,
- the output has a fixed length,
- it is relatively easy to compute for any given document,
- it is a one way function (figure 6), that means it is a mathematical function significantly easier to perform in one direction (the forward direction) than in the opposite direction (the inverse direction may take months or years).
- it is collision free, that means for a given hash value it is hard to find another document with the same hash value.

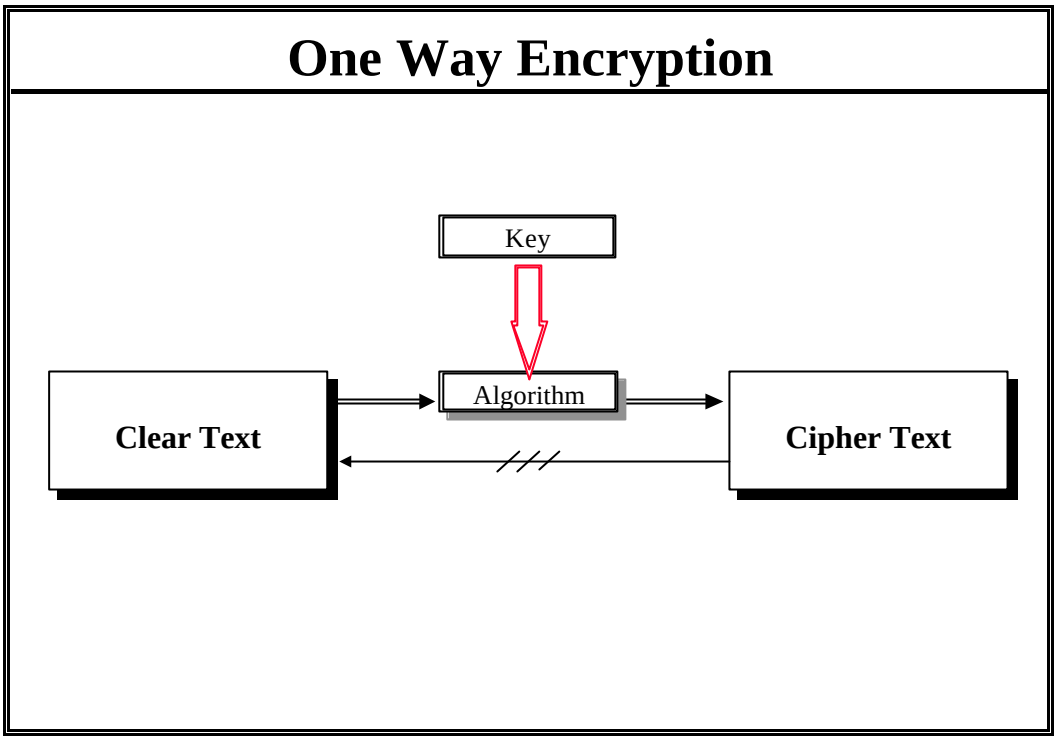


Figure 6: One Way Encryption

Hash functions are used because they are generally faster than digital signature algorithms.

Examples for Hash algorithms are: RIPEMD, RIPEMD-160, MDC-2, SHA, SHA-1, MD2, MD5, BHF, BSAH, Square-mod.

The digital signature process with hashing the document and encrypting the hash by the sender and hashing the document and encrypting the digital signature is shown in Figure 7: Digital Signature

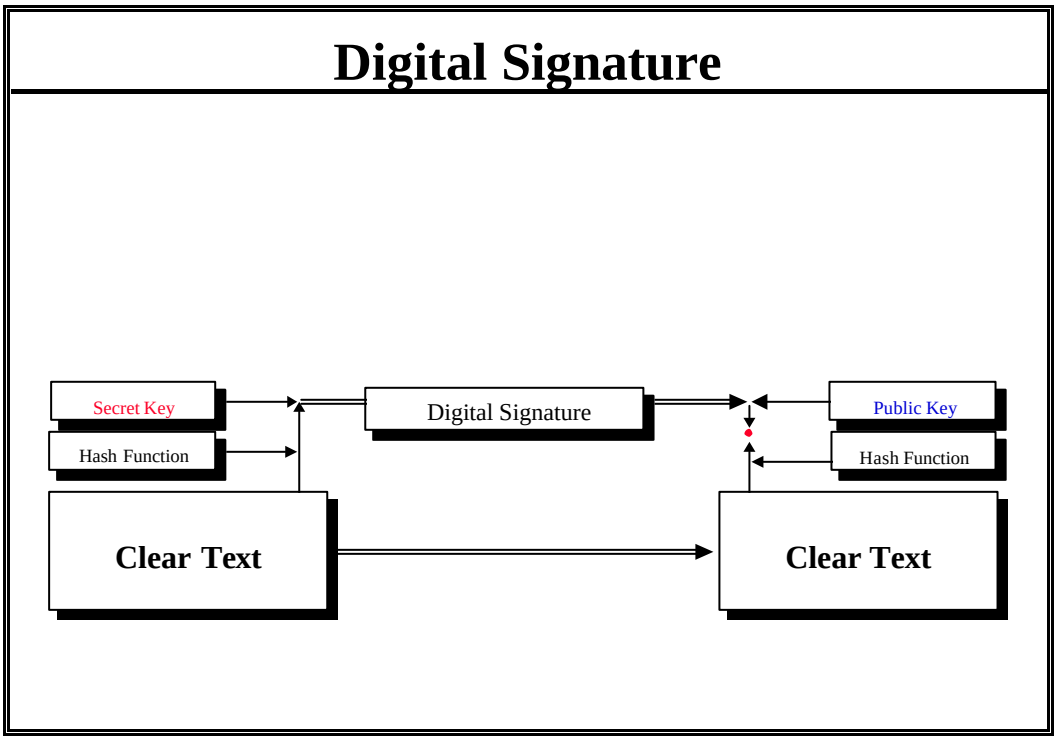


Figure 7: Digital Signature

Legally, the digital signature certifies that the entity's manual signature may be replaced by the corresponding digital signature.

#### 2.2.3.3 Types of Signatures

There are several types of signatures used. Examples are the dual signature, where a document is signed

with two signatures or the blind signature where a document is signed without knowing the content.

## 2.3 Trusted Entities

### 2.3.1 Key Distribution Centers (KDC)

Key distribution centers are trusted entities that are used for key management. In the past they have primarily been used to store and distribute secret keys. The general mode of operation is that the KDC shares a master secret key with each of its customers and distributes secret session keys encrypted with the master key.

### 2.3.2 Trusted Third Parties

Digital signatures are used for authentication of a sender, non-repudiation and message integrity purposes. In order for a user to trust these security services the user needs to be assured that the public key used to verify a signature is actually the key of the person who signed the transaction.

There are several possible ways to find the public key of a sender. The receiver could call the sender and ask him. Or the public key can be requested and sent via e-mail. Or all the public keys are stored in a directory. But such a directory must be secure against tampering, so that users can be confident - can trust - that a public key listed in the directory actually belongs to the person listed. Otherwise, they might send private encrypted information to the wrong person. The party providing such a trustworthy directory is called a trusted third party (TTP).

Third-party trust refers to a situation in which two individuals implicitly trust each other even though they have not previously established a personal relationship. The TTP certifies by digitally signing the combination of the public key together with some identity information like the name of the entity. This process is called binding the public key to the entity.

Trusted Third Parties are security authorities trusted by other entities with respect to security related activities. They perform three major groups of functions:

#### 2.3.2.1 Primary Functions

Registration: This is the process whereby a user first makes itself known to a certification authority (directly, or through a local registration authority), prior to that certification authority issuing a certificate or certificates for that user. The registration authority registers and enrols the participating users and user groups. Registration of anonymous users. Registration of the public keys of the users or the secure generation of key pairs with secure distribution of the private key to the user, and deletion of the private key in the trusted third party.

Naming: Naming, determining the validity of Distinguished Names.

Initialisation: Before a client system can operate securely it is necessary to provide it with the necessary key materials which have the appropriate relationship with keys stored elsewhere in the infrastructure. For example, the user needs to be securely initialised with the public key of a certification authority, to be used in validating certificate paths. Furthermore, a user typically needs to be initialised with its own key pair(s).

#### 2.3.2.2 Secondary Functions

Identification and authentication of registered users. It is not simple to identify, authenticate and verify a person or entity in a very trustworthy way. The only way to bypass this risky and complex process is when the entity generates her own certificates.

Key generation: Generation of private and public keys.

Key storage: The generated public keys have to be stored in a directory. Generated private keys have to be deleted.

Key distribution: In the case the private keys are generated by the trusted third party, these keys have to be distributed to the key holder. Public keys have to be distributed on request.

#### 2.3.2.3 Tertiary Service Functions

Certification: The process by which a certification authority issues a certificate for a registered user's public key, and returns that certificate to the user's client system and/or posts that certificate in a repository. It might be useful to include some information about the user: Attributes like affiliations, authorities etc.

Validity check: Check of the validity of the certificates.

Directory management: Storage of common information, such as mail addresses and security information and public keys. List of valid and invalid certificates. Storage of expired certificates.

| | |
|---|---|
| Time stamping: | Time stamping of documents. |
| Change of Keys: | Revocation change of registered public keys. |
| CRL management: | Management of the certificate revocation list (CRL), the list of revoked (and therefore invalid) certificates. Storage and distribution of these lists. |
| Co-operation: | With other trusted third parties. |

### 2.3.3 Types of Trusted Third Parties

Just as there will be several different types of users of the network, there could be many different types of TTPs. Many of those TTPs would serve only their own special community of users whereas others might be more general, for example providing services to the general public.

The following functional types of TTPs can be distinguished:

Off-line TTPs*:*

An off-line TTP does not interact with the user entities during the process of the given security service. Instead the interaction to provide, or register, security-related information is carried out off-line as a separate interaction. The results of such an interaction may be cached and reused to avoid having to communicate with the server each time communication is initiated.

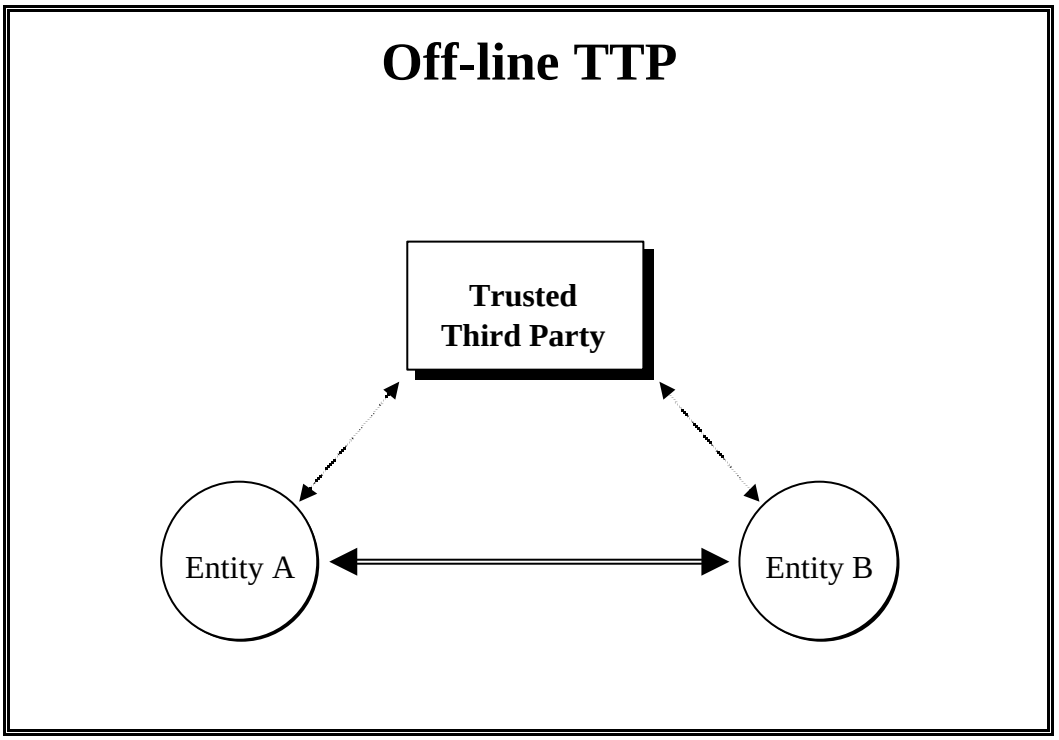


Figure 8: Off-line TTP

On-line TTPs*:*

An on-line TTP is requested by one or both entities in real-time to provide, or register, security-related information. Such a TTP is not in the communications path between the two entities.

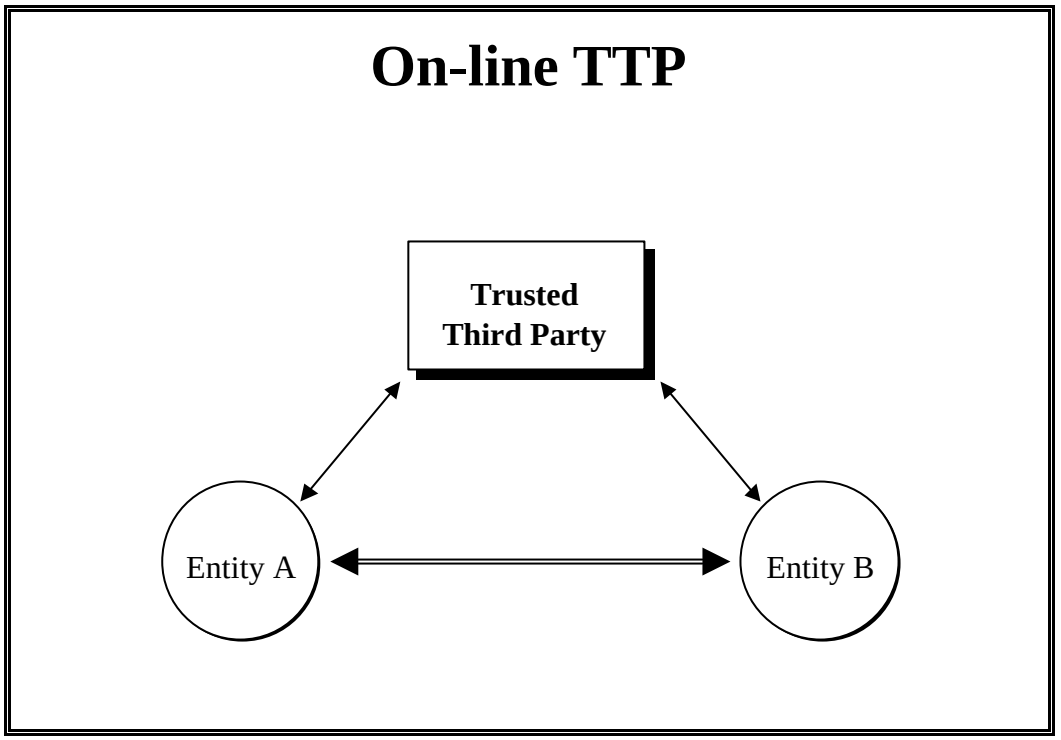


Figure 9: On-line TTP

In-line TTPs:

An in-line TTP is positioned in the communication path between the entities. Such an arrangement allows the TTP to offer a wide range of security services directly to users. Since the TTP interrupts the communication path, different security domains can exist on either side of it.

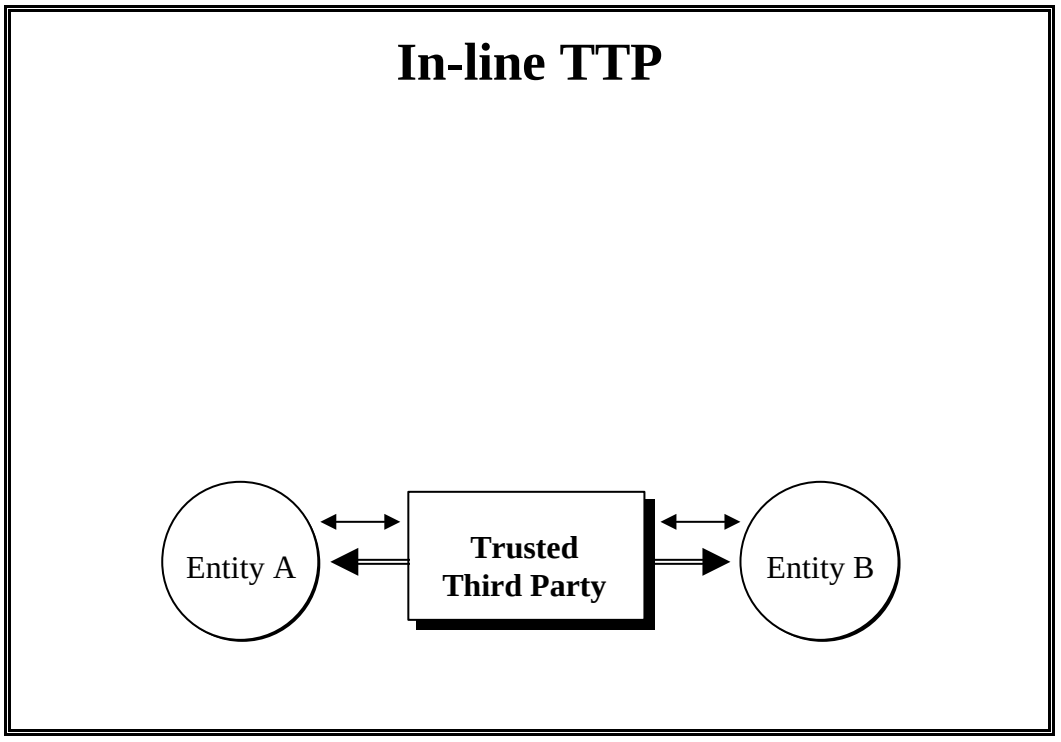


Figure 10: In-line TTP

Functionally Trusted Third Parties (FTTP)

This type arises from the obvious need for reliable registration of users of the system. If public key methods are used, this will usually include certification of public keys as belonging to certain users. A TTP trusted to perform this function is called functionally trusted. It is clear that if the registration is not done in a reliable manner, users cannot even be sure with whom they are communicating. So functional trust represents a minimal amount of trust that must be placed in a TTP. Note that this type of TTP does

not need to know the secret key of any user, nor does it need to know any conventional keys used for data communication between users.

The functionality required in this instance is comparable to the functionality of a phone book: It provides a reliable connection between people, or their residence, rather, and their phone numbers.

Unconditionally Trusted Third Parties (UTTP)

This type of TTP is typically needed in systems that use conventional cryptography only, or systems, where a current registration of the primary messages take place (like securities). In addition to the registration function mentioned above, such an unconditionally trusted TTP will generate keys for data communication and then communicate them securely to the users who need them, or be responsible for the primary messages in the system. This means that the TTP knows and in principle could make use of all the secret information in the system, or be responsible for the information related to a particular user. Thus elaborate measures must be taken to prevent misuse. The latest approach is to use escrow-agents, which is an example of an unconditionally trusted third party, as they have access to all the user's secrets.

This type of trusted third party is similar to the former used Key Distribution Center (KDC).

### 2.3.4 Co-operation of Trusted Third Parties

There are three basic ways, how TTPs may co-operate:

| | |
|---|---|
| Stand-alone TTPs: | Where, for example, a corporate might have its own TTP to support only its own staff and used only for internal traffic. This stand-alone TTP would not need to be connected to the public network of TTPs as there would be no need for the certificates it issued to be verified outside its domain. |
| A public network of TTPs: | The open world-wide network of TTPs, each able to recognise certificates issued by any other TTP. |
| Private networks of TTPs: | Each comprising a subset of TTPs where a number of TTPs would form a closed community for the exchange of certain types of traffic. |

In the future TTPs will exist in both public and private domains, at the local, national and international level, they are therefore required to interwork. TTPs should have agreements arranged with other TTPs to form a network of trust, thus allowing a user to communicate securely with every user of every TTP with whom his TTP has an agreement.

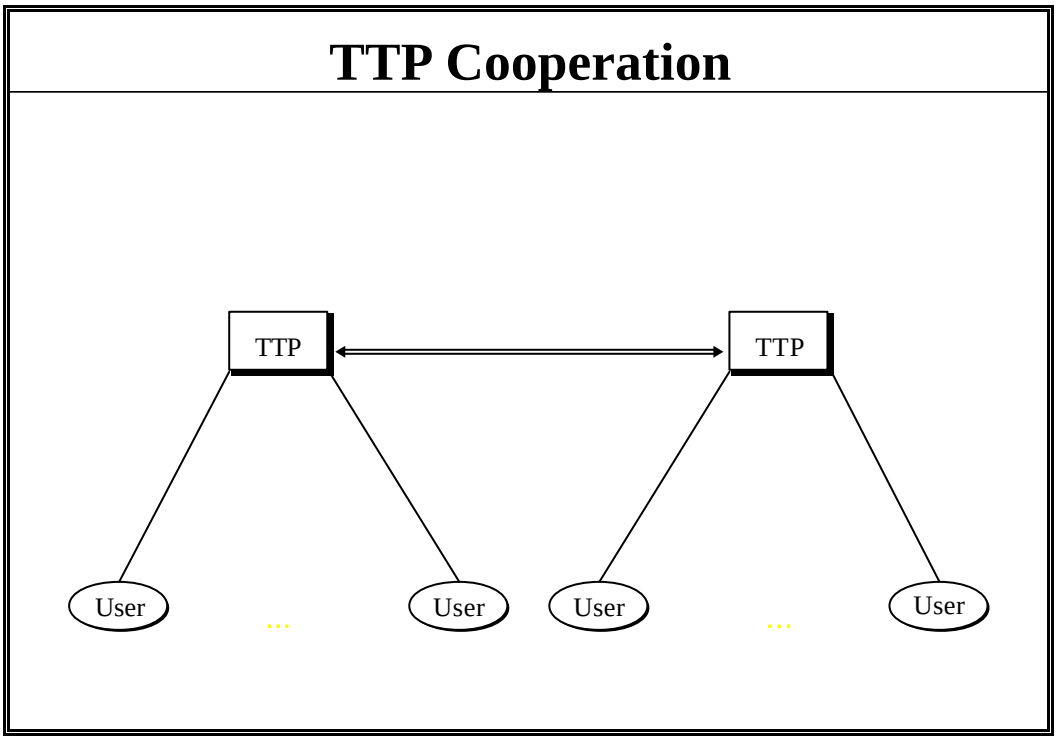


Figure 11: Co-operation between Trusted Third Parties

### 2.3.5 TTP Interoperability

Communicating with entities of other trust infrastructures needs to process and verify certificates of the other infrastructures. There may be two ways to go.

- Implement processes in each computer of an entity which are able to process all certificates that may originate from other infrastructures. The problem is, that each communicating entity needs to implement all the features to process all the possible certificates of all trust infrastructures of the world.
- Establish processes which are able to process the certificates of the other trust infrastructures and translate them and implement this processes on a single gateway. The gateway needs to be certified, it translates the certificates, certifies the translated certificate and sends it to the receiver in the other infrastructure. The advantage its that there is only one implementation of all the special processes necessary.

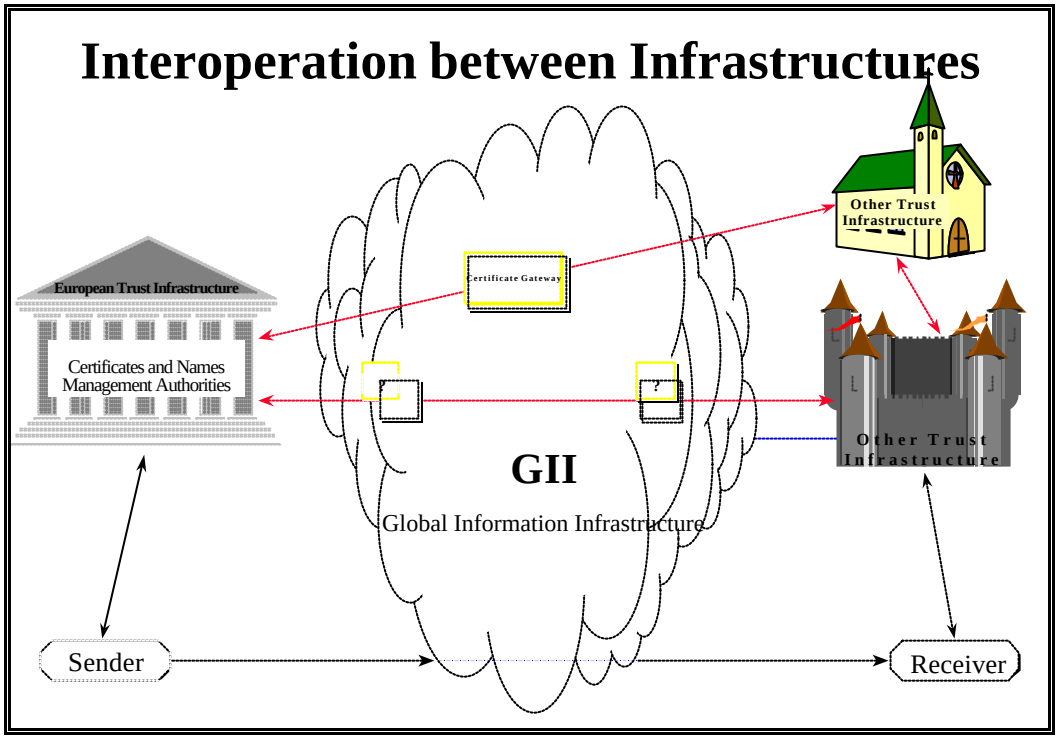


Figure 12: Interoperation between Trust Infrastructures

### 2.3.6 Services delivered by Trusted Third Parties

The following list provides some examples where Trusted Third Parties can deliver services to specific user communities:

- Governmental Authority:
  Such an authority may be local or federal certifying a local resident's public key allowing him or her to sign documents with a unique digital signature.
- Internet Service Providers and Online Service Providers:
  They provide communications services for private individuals and companies and governmental agencies with e-mail, file transfer, etc. and provide value added services.

And there are TTPs serving only their own community of users including registration authorities such as:

- General Medical Council (GMC) in the UK or the Bundesärztekammer or the DATEV in Germany.
  The GMC could run a TTP for issuing certificates to registered family doctors for their public digital signature keys. Pharmacists would be able to verify the doctor's signature on a prescription before fulfilling it.
- Society of Notary Publics:
  This society could run a TTP for issuing digital signature certificates to registered Notary Publics. This is discussed among others in the UK and Germany also.
- Law Society:
  This society could run a TTP for issuing certificates to registered solicitors.
- Personal Investment Association:
  This association could issue certificates to independent Financial Advisers.

### 2.3.7 Policy and Licensing Authorities

To grant trust to the users of a trust infrastructure, it is necessary to establish trustworthy TTPs. Otherwise the users may not be sure that they know either the user they communicate with nor the TTP as the issuer of the certificate for the correspondent user. Therefore the TTPs or certification authorities are to

be certified by a policy certification authority, which issues certificates for the TTPs.

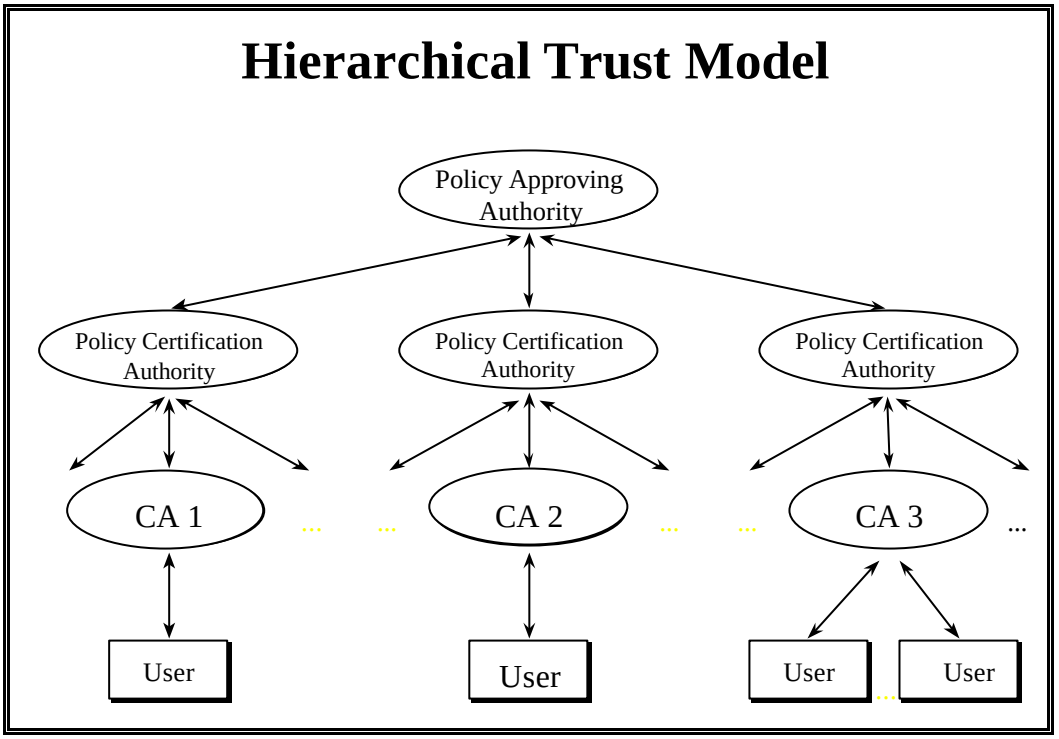


Figure 13: Hierarchical Trust Model of PAA, PCA and Users

The following types of authorities can be distinguished.

Policy Approval Authority (PAA)

An authority which establishes the overall infrastructure security policy and creates guidelines that all subordinate entities must follow. The PAA also acts as a root certification authority, issuing certificates for the next tier of certification authorities (PCAs).

Policy Certification Authority (PCA)

An authority which establishes policy for a single organisation or single community of interest. A PCA also acts as a certification authority for the next tier of certification authorities (CAs).

Certification Authorities (CA)

An authority trusted by one or more users to create, assign and issuing public verification key certificates to end entities and other certification authorities certificates - by binding the public key and an entity - may be an individual - by name. Optionally the certification authority may create the users´ keys. Certification authorities issue certificate revocation lists periodically, and post certificates and certificate revocation lists to a repository.

Organisational Registration Authority (ORA)

An entity that acts as an intermediary between the CA and a prospective certificate subject; the CA trusts the ORA to verify the subject's identity and that the subject possesses the private key corresponding to the public key to be bound to that identity in a certificate.

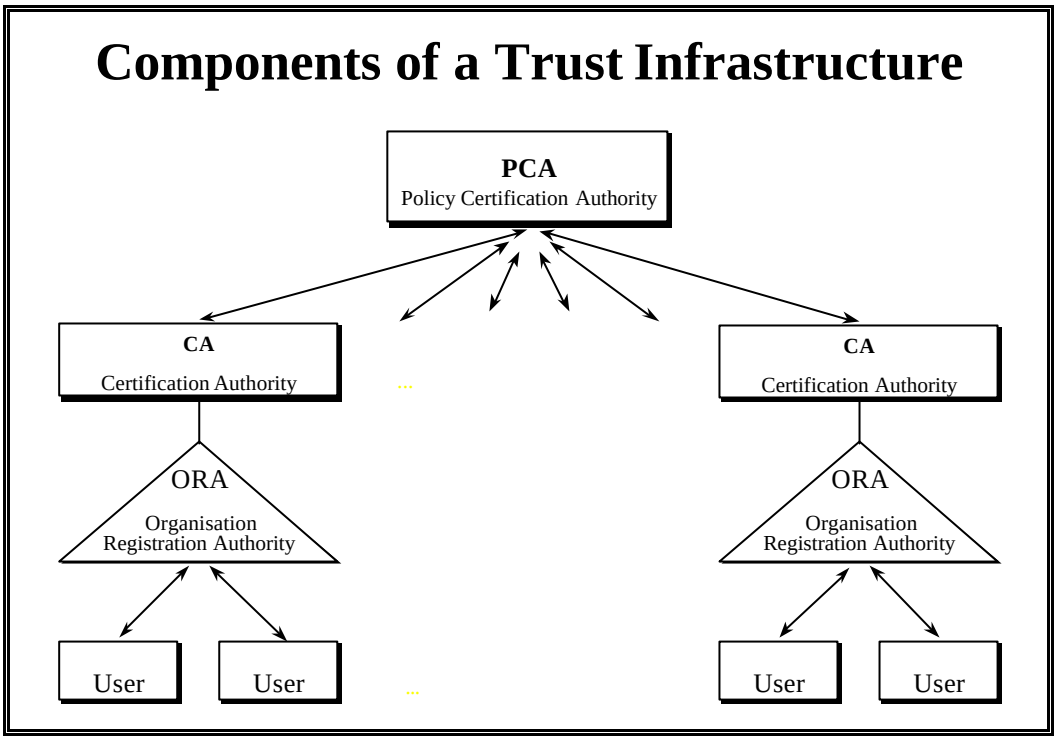


Figure 14: Components of a Trust Infrastructure

**Attestation, Registration and Enrolment**

A prerequisite for the issuing of a certificate is the initial attestation, registration and enrolment of users and the validation of their credentials by a registration authority. This will involve a paper-based process with written signatures to establish the credentials of the individual or organisation concerned. The user will participate in some way in the generation of their digital signature key pairs. The secret key may be put onto a smartcard and is protected by a PIN which the user can change from time to time. The public key is signed by the user (in paper form) and sent to the certification authority. Whoever takes the registration authority role will need to offer a local presence to facilitate registration; they will also need to be perceived as trusted in handling user credentials. A number of organisations could offer this service. The certification authority is connected by means of secured communications to the registration authorities, through which any user may register. A registration is acknowledged by a certification authority.

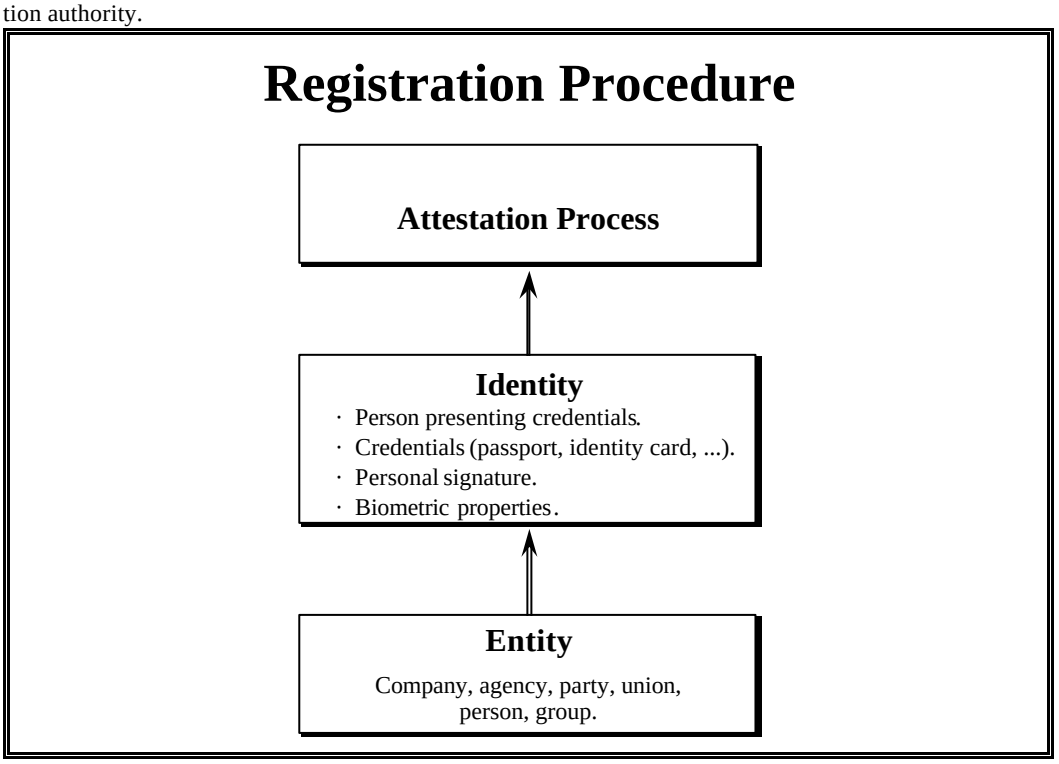


Figure 15: Registration Procedure

**Credentials**

The following tokens may be used as credentials in the registration process. Three assurance levels are used:

Low Assurance
- Birth certificate.
- Government identity card.
- Driver's licence.
- Passport.

Medium Assurance
- Authorisation letter of a supervisor or sponsor.

High Assurance
- Certified authorisation letter of a supervisor or sponsor.
- Fingerprints.

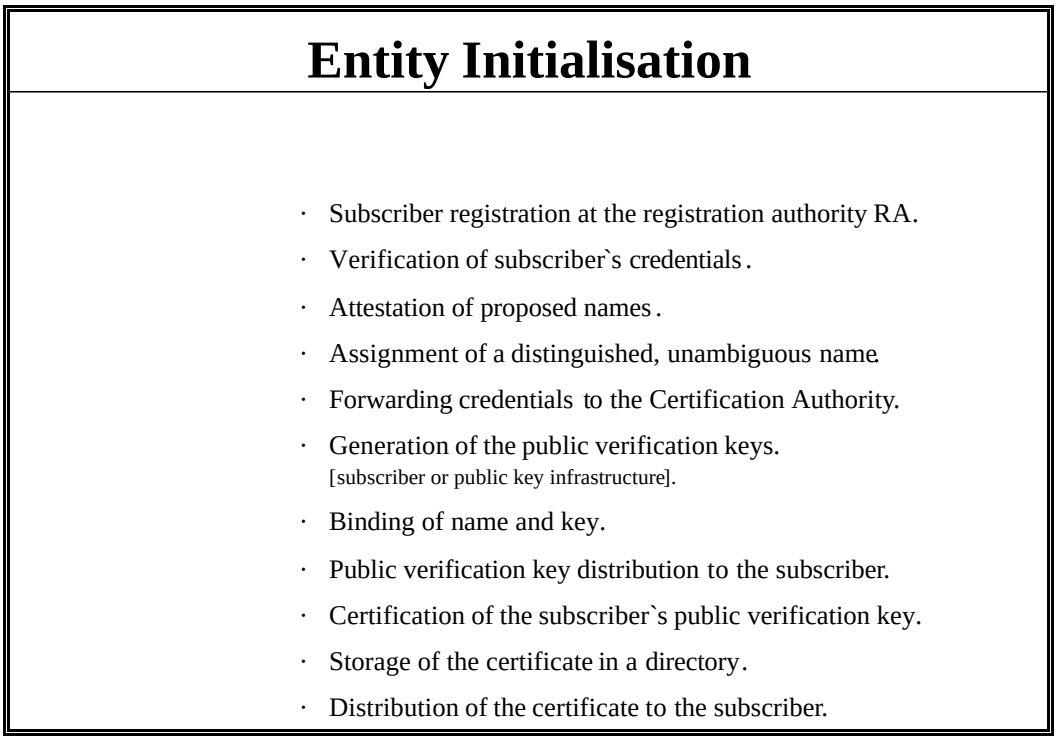


Figure 16: Entity Initialisation

**Registries**

There are two types of registries for the cryptographic algorithms.
- Registry of hash algorithms and a
- registry of digital signature algorithms.

**Fraud Detection Center**

An authority which can alert various parties to possible fraudulent behaviour based on real-time event sequence monitoring and audit. The advantage of using an external authority for this purpose, is that the authority can independently monitor all the entities in a particular system to identify fraud scenarios that occur between different entity domains. The user trusts the fraud detection center not to fabricate evidence regarding the user's involvement in fraudulent activity.

### 2.3.8 Security Policies and Measures

Security Policies

One can distinguish between (at least) two types of security policies that can be involved in distributed systems security:
- Policies that specify how entities and organisations should behave when participating in the development of the infrastructure and
- The individual users` policies used for deriving conclusions from the available information.

Document and Management Policies

For each of the components of a trust infrastructure shown in Figure 14: Components of a Trust Infrastructure there exist policies as described in the following figure.

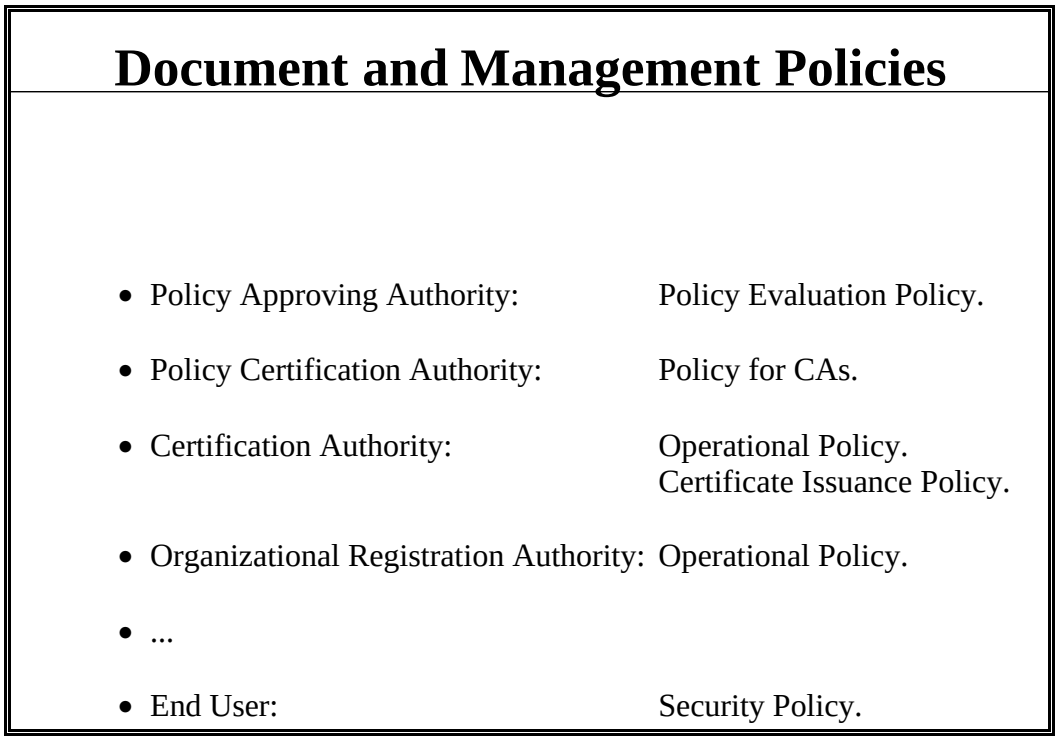


Figure 17: Document and Management Policies

Security Measures

The keys and certificates are to be distributed in the trust infrastructure over unsecure channels and therefore should be distributed encrypted (confidentiality) and/or digitally signed (integrity). Physical and personnel security measures are also to be implemented.

### 2.3.9 Management Protocols

Management protocols are required to support on-line interactions between trust infrastructure components. For example, a management protocol might be used between a certification authority and a client system with which a key pair is associated, or between two certification authorities which cross-certify each other. The set of functions which potentially need to be supported by management protocols include:

Registration.

Initialisation.

Certification.

Key pair recovery: As an option, user client key materials (e.g., a user's private key used for encryption purposes) may be backed up by a certification authority or a key backup system. If a user needs to recover these backed up key materials (e.g., as a result of a forgotten password or a lost key chain file), an on-line protocol exchange may be needed to support such recovery.

Key pair update: All key pairs need to be updated regularly, i.e., replaced with a new key pair, and new certificates issued.

Revocation request: An authorised person advises a certification authority of an abnormal situation requiring certificate revocation.

Cross-certification: Two certification authorities exchange the information necessary to establish cross-certificates between those certification authorities.

Searching Directories for an (e-mail) address.

On-line protocols are not the only way of implementing these functions. For all functions there are off-line methods of achieving the same result. For example, when hardware tokens are used, many of the functions may be achieved through a part of the physical token delivery. Furthermore, some of the discussed functions may be combined into one protocol. In particular, two or more of the registration, initialisation, and certification functions can be combined into one protocol exchange.

For those functions the following protocols are used FTP, DAP, LDAP, and OCSP.

## 2.4 Certificates

A structured message which delegates an attribute of some form to a public key in a trustworthy way is called a certificate. It certifies with its digital signature a public verification key of an entity together with some other information, rendered unforgeable (signed) by encipherment with the private key of the

certification authority which issued the certificate for authentication of a digital signature.

### 2.4.0 Introduction

To avoid tampering the public key, the trusted third party digitally signs the public key of a user together with its name. The digital signature of the trusted third party together with the public key and the name of its owner is called a certificate. The trusted third party sends the certificate to the requesting party. The requesting party verifies the certificate and can trust in the received public key. This requires that certificates should be generated by and obtained from trusted sources.

Therefore a certificate is a digitally signed set of documents (like a public key), an entity (like an identity, a principal, a human being, a device) binding especially a key to an entity. Technically a Trusted Third Party certifies the validity of the public key by digitally signing the combination of the public key and some identity information like the name of the entity. This combination is called a certificate. This process is called binding the public key to the entity.

The two components public key and name of the subject who is the owner of the key digitally signed are the minimum content of a certificate.

| Name of the Field | Contents and Functions |
|---|---|
| Name | Distinguished Name of the authenticated subject or entity. The registration authority processes the entity's application, for validating the entity's credentials and for assigning a Distinguished Name to the entity. |
| Key | Subject or entity's public key information: Algorithm, parameter, key: The entity generates an asymmetric key pair for signatures and for submitting the public key for verification to the registration authority. |
| Signature | Overall signature by the certification authority: The signature of the certification authority binds the public key to the entity's name. |

Figure 18: Possible Minimum Information Content of a Certificate

It may be necessary to have some more information to make a certificate useful.

| Part | Contained Information | |
|---|---|---|
| Version | Version number; an integer, value is "2" for version 3 | |
| Serial number | Unique identifier for each certificate generated by issuer; integer | |
| Signature | Algorithm identifier | algorithm used to sign certificate |
| | Parameters | any parameter needed |
| Issuer | Name of issuer (X.500 "Distinguished Name", a sequence of RelativeDistinguishedNames that uniquely identify a directory object). | |
| Validity | NotBefore | UTCTime |
| | NotAfter | UTCTime |
| Subject | Name of subject (X.500 "Distinguished Name") | |
| Subject's public key info | Algorithm identifier | subject's signature algorithm |
| | Public key | subject's public key |
| Issuer unique identifier | contains additional information about the subject; must be version 2 or higher (optional). | |
| Subject unique identifier | contains additional information about the issuer; must be version 2 or higher (optional). | |
| Extensions | (optional) | |
| Issuer's signature | | |

Figure 19: X.509 Certificate Version 3

There must be a serial number associated with the certificate generated by the certification authority to retrieve the certificate. The certification authority has to specify the type of certificate, the version, the algorithm used for the public/private key system together with its parameters, the name and address of the issuer to authenticate the certificate, and the validity of the certificate etc.


Figure 20: Sample Certificate

#### 2.4.0.1 Types of Certificates

In the simplest form, certificates contain only a public key and a name. As commonly used, a certificate also additionally contains an expiration date, the name of the certifying authority that issued the certificate, a serial number, and perhaps other information. Most importantly, it contains the digital signature of the certificate issuer.

There are several types of certificates.

Identity Certificates

An identity certificate binds a public key to the Distinguished Name of an entity and thereby delegates all the attributes of the named entity to the public key.

The today most widely accepted format for certificates is defined by the ITU-T X.509 international standard. Thus, certificates can be read or written by any application complying with this standard.

Authorisation Certificates

An authorisation certificate grants a specific authority to a subject rather than binding an identity (such as a person's name) with all its attributes to that key. For example, an authorisation certificate might grant permission for a given subject to authenticate log-ins over some protocol to a host for some period of time.

Privilege Attribute Certificate

A TTP can be used to certify privileges for access control. Certified privileges may be obtained from a TTP either in the form of a privilege attribute certificate (PAC). Such a PAC will contain a list of resources that the user may wish to access and the associated privilege level that the user has been assigned. A user will send a request for a given privilege attribute to a TTP, which will then identify and authenticate the user. If the current security policy states that the user is authorised to make the requested access, then the TTP will generate and certify the privilege attribute. The privilege attribute certificate is distributed by the TTP by publishing it in a directory.

The veracity of privilege information can be verified on demand by anyone who obtains the TTPs public certification key. The privilege attribute certificate may need to be revoked if changes in access control privileges are required, or if a compromise of sensitive information is suspected.

Privilege attribute certificates are functionally comparable to the authorisation certificate.

Public Key Certificates

A public key certificate is a digitally signed data structure that binds the distinguishing identifier of an entity (certificate holder, subject) to the public key of this entity and which indicates the validity of the corresponding private key.

In order to verify the signatures of an entity it is necessary to know the public key of that entity to verify. This public key is therefore called the public verification key of the entity. It is crucial that the entity's public key is distributed in an authenticated fashion. The distributor of the public verification key

should be a trusted party vouching for the legitimacy of the entity's public verification key.

The trusted party will digitally sign the name and the public key of an entity: The signature of the trusted party binds the public key to the entity. The result is the public key certificate. To verify an entity public key certificate it is necessary to know the certification authority's public verification key.

Attribute certificates

It may be useful to certify other attributes of an entity besides the public key. There are attributes which directly relate to the public key certificate, such as

- method of delivery of the credentials,
- method of identification of the entity, entity type (e.g. individual, corporation, governmental agency, device, program, etc.),
- information on trusted third parties involved in the registration process,
- information on domains and the type of the structure of the certification authorities e.g. hierarchical or net (cross-certificates).

There may be other attributes which relate to the signing capabilities of an entity, such as:

- Authorised signatory: Formal authorisation of individuals to sign for the organisation.
- Transaction limit: Maximum monetary value of a message which the entity may sign).
- Transaction type: Transaction types which the entity may sign.
- Time of day: During which signatures of the entity are considered valid.
- Days of the week, months etc. During which signatures of the entity are considered valid.
- Location: From which signatures of the entity are considered valid.
- Pre-approved counter party: To indicate with which parties the entity can conduct signed transactions.
- Delegation control: Indicate the amount of authority that the entity may delegate to some other entity.

These attribute certificates refer directly to a public key certificate.

The attributes could be included in the basic public key certificate. However, a number of reasons suggests that they are not:

- The attributes are not needed in every transaction,
- the attributes would increase the size of the basic certificate,
- attributes may vary much more often than the basic certificate information, which would require a reissuance of the certificate at each change.

Existing standards such as X.500 assume that these attributes are not included in the basic public key certificate. Rather these attributes shall be carried in a separate structure called the attribute certificate. To emphasise that the attribute certificate need not be generated by the same certification authority as the basic public key certificate, the notion of attribute certification authority is useful.

For example see the following Figure 21: Structure of an ANSI X9.30-3 Attribute Certificate Information.

| Attribute Certificate Information | Explanation |
|---|---|
| baseCertificateID | Issuer and serial number of the associated public key certificate. |
| Issuer | Distinguished Name of the certification authority. |
| SerialNumber | Serial number of the attribute certificate. |
| Validity | Validity period of the attribute certificate. |
| Attributes | A set of attributes to be certified. |
| IssuerUniqueID | Unique identifier of the attribute certification authority public key (optionally). |

Figure 21: Structure of an ANSI X9.30-3 Attribute Certificate Information

The attribute certificate is signed by the attribute certification authority.

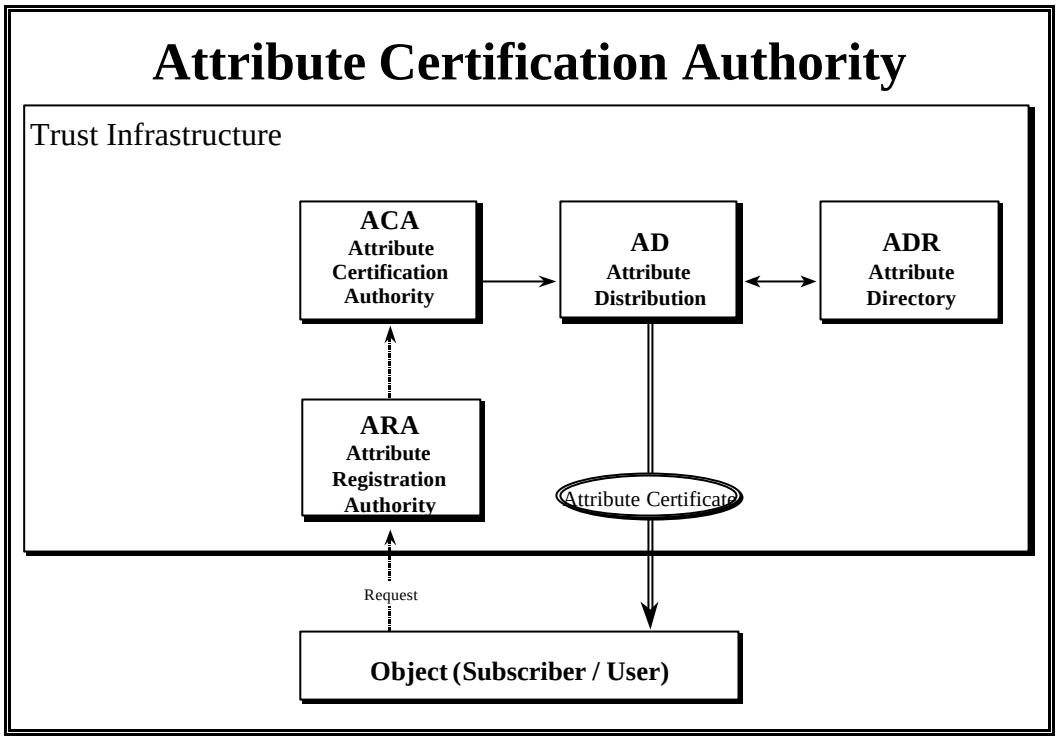


Figure 22: Attribute Certification Authority

Cross Certificates

In contrary to a hierarchical model of TTPs and policy certification authorities, in a network model of co-operating TTPs or certification authorities these are certifying each other by generating a cross-certificate for each other so that each can effectively certify the trustworthiness of the other's key. The user who receives a certificate verifies it by requesting these two certification authorities only.

| Part | Contained Information |
|---|---|
| version | |
| serial number | |
| signature | |
| issuer | name of issuer of this certificate (e.g. "CA 1"). |
| validity | |
| subject | name of the co-operating certification authority (e.g. "CA 2"). |
| subject's public key info | |
| issuer unique identifier | |
| subject unique identifier | |
| extensions | |
| issuer's signature | |

Figure 23: Forward Cross Certificate

The cross certificate pair construct contains two certificates: One forward certificate and one reverse certificate. The subject of the forward certificate is the issuer of the reverse certificate and vice versa. Cross certificates build bi-directional chains of trust between certification authorities.

| Part | Contained Information |
|---|---|
| version | |
| serial number | |
| signature | |
| issuer | name of issuer of this certificate (e.g. "CA 2"). |
| validity | |
| subject | name of the co-operating certification authority (e.g. "CA 1"). |
| subject's public key info | |
| issuer unique identifier | |
| subject unique identifier | |
| extensions | |
| issuer's signature | |

Figure 24: Reverse Cross Certificate

In contrary to a hierarchical model of TTPs and policy certification authorities, in a network model of co-operating TTPs or certification authorities these are certifying each other by generating a cross-certificate so that each can effectively certify the trustworthiness of the other's key. The user who receives the certificate verifies this pair of certificates by requesting these two certification authorities only.

Revocation Certificate

The revocation certificate contains the following items:

- Original certificate information.
- Date (and time) of revocation.
- Reason of revocation.
- Date (and time) of known or suspected compromise.
- Revocation requesting party (initiator).
- Revocation performing certification authority.
- Digital signature of the revoking certification authority.

| Part | Contained Information |
|---|---|
| Certificate | The original certificate (or certificate serial number). |
| Signature | Signature algorithm identifier. |
| Issuer | Certification authority who performed the revocation. |
| Revocation date | Date when certificate has been revoked. |
| Reason | Reason why the certificate has been revoked (e.g. key compromise)[1]. |
| Reason (compromise) date | Date of occurred reason (e.g. suspected compromise) (optionally). |
| Initiator | Party who requested the revocation (optionally). |
| Issuer's signature | |

Figure 25: Generic Revocation Certificate Structure

Confidential Certificates

There may be the risk of leaking of sensitive information into the contents of a certificate – e.g. in authorisation certificates. In this case the certificates should be kept confidential and the related fields may be encrypted with a key only known by the intended verifier before being returned to the subject. Presumably the subject knows the effect of the certificate, but anyone observing the certificate on the network would be prevented from gaining knowledge of it.

Similarly, an entire certificate could be encrypted with a key known only to the issuer, with the same effect.

[1] In some countries, data privacy laws may prevent the use of detailed revocation reason codes.

#### 2.4.0.2 Validity Period of Certificates

When a certificate is issued, it is expected to be in use for its entire validity period. The validity period is an interval of time during which a certificate is valid. Because of the time to communicate, a validity period can never go to zero. The shorter the period, the less risk an issuer runs of having a withdrawn authorisation active in the world. The longer the period, the less communication and revalidation overhead is incurred. Choice of validity period is left up to the issuer of the certificate.

**Validity Period**

The validity period is an interval of time during which a certificate is valid. Because of the time to communicate, a validity period can never go to zero. The shorter the period, the less risk an issuer runs of having a withdrawn authorisation active in the world. The longer the period, the less communication and revalidation overhead is incurred. Choice of validity period is left up to the issuer of the certificate.

A certificate must not necessarily have a validity period. If a certificate has no validity period, then the receiver does not know if the received certificate has expired.

#### 2.4.0.3 Revocation Process

There are several business requirements relating to the revocation of certificates:

- It might be required to have access to a revoked certificate in order to perform verifications with the revoked public key. Such a situation may occur when a document was signed long before the revocation but has to be verified after the revocation.
- It seems inadequate to request users to download the complete revocation list in order to find out about individual revocations. In addition, the revocation list typically gives only the serial numbers of revoked certificates. Therefore, these revocation lists are only useful to check the status of certificates which have obtained through other means.
- It seems desirable to have a legally binding status for revoked certificates. The same way as the certification authority certifies the association between an entity and its public key, the certification authority should certify the revocation of a certificate in a self-contained manner.
- Periodically issued certificate revocation lists do not allow for non-repudiation. Given the currently valid revocation list (say, from last month) and a business partner signature that was computed after the issuance of the revocation list, the recipient does not have assurance of the validity of the signature (and hence, no non-repudiation).
- One might wish to check the history of a user for the reasons and frequencies of revocations. This is a question of a black list see chapter 2.4.0.3 Revocation Process with black lists.

In order to be able to trust in a certificate it is necessary to know whether it is still valid or whether it has been revoked i.e. his validity has ended. This is accomplished by certificate revocation lists. A certificate revocation list has to be provided and digitally signed by the issuing authority – and accompanied by a certificate binding the relating public verification key to the name of the authority publishing the certificate revocation list.

Certificates typically have a fixed validity period. However, there is a number of reasons why a certificate may have to be declared invalid before the end of that validity period.

A certificate may need to be revoked for a number of reasons (e.g., compromise of the private key).

In the certification process, the certification authority issues public keys as self-contained electronic documents. In the revocation process, the certificate serial number is included in a list also called certificate revocation list - CRL. The certificate revocation list is made available to users on a periodic basis.

There are other methods possible. All the methods of validity checking are functionally equivalent but they have different performance impacts.

**Reasons for Revocation**

There may be several incidents, which make a certificate unvalid, e.g.:

User level:

- Compromise or lost of the entity's private key (suspected or verified).
- Changes in the purpose for which the key was being used.
- Change in the identification information contained in the certificate.
- Change of the certificate information.
- Termination of the entity's operation.
- Change of the company (job change, retirement).

Certification authority level[2]:

- Compromise of the certification authority's private key (suspected or verified).
- Change of the certification authority's certificate information.
- Termination of the certification authority's operation.
- Change of policy etc.

Revocation is a very serious process since it could potentially result in disastrous business conse-

[2] It must be distinguished between the user level and certification authority level, since the compromise of a certification authority's private key may require the revocation and subsequent re-certification of all user keys in a domain.

quences. There must be utmost care in verifying the authenticity of a revocation request.

The following figures describe the revocation notice and the revocation list of the ANSI X9.30-3 Standard.

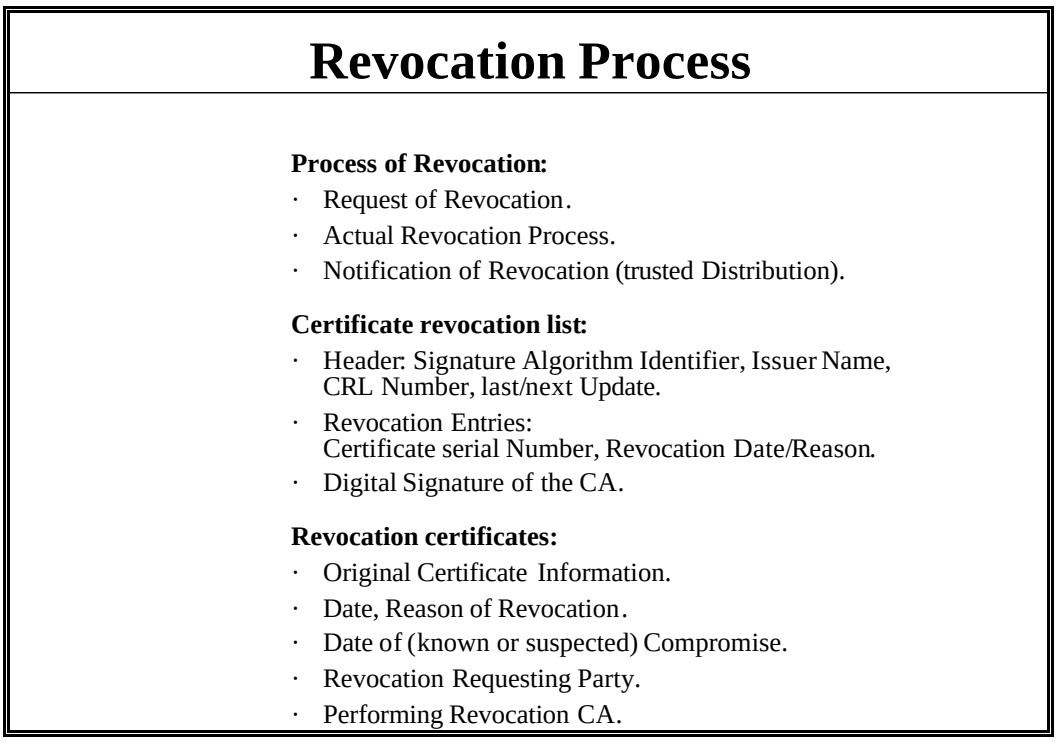


Figure 26: Revocation Process

**Certificate Revocation Process**

There are the following phases in the revocation process:

1. Revocation request:
- Who has the authority to request the revocation of a certificate?
- Under what conditions can or must a certificate be revoked?
- How to prove and verify a revocation request?

2. Revocation:
- Who performs the revocation?
- How is the certificate revoked?
- Can certificates be on alert or on hold?

3. Revocation notification:
- How to notify users (and other relevant entities) about the revocation of a certificate?
- How to prevent misuse of revoked certificates?

In case that a certificate has to be revoked there must be an information flow to the users concerning the revoked certificate. Several revoked certificates are published together building a list called certificate revocation list (CRL). To be trusted, such a list has to be signed digitally – and accompanied by a certificate binding the relating public verification key to the name of the authority publishing the CRLs.

The revoked certificates are named by their serial numbers. Certificates are uniquely identified by the combination of the issuer name or issuer alternative name along with the user certificate serial number. The date on which the revocation occurred is specified. The time for the revocation date is expressed.

The certificate revocation list certificate may not be trusted by a user. Therefore he is able to request a verification of the relating certificate.

The point of certificate revocation lists is to avoid the need for online services. It's not so much the replication of the database; rather, it's the requirement that all possible acceptors of certificates be online to check certificates.

The certificate revocation list is signed by the certification authority and published and distributed on a periodic basis.

There may be a small delay by the point of time e.g. the compromise, the message to the certification authority, the registering in the directory and, the distribution of a supposed list.

Implementations

There are two ways possible for handling certificate revocation.

Revocation List

The relevant public key together with the certificate is marked as deleted in the public key directory. Additionally the revoked certificate is registered in the revocation list with its serial number, and the date and time of revocation as described above.

The certificate revocation list contains the following items:

- signature algorithm identifier,
- issuer name,
- certificate revocation list number,
- last update,
- next update,
- certificate serial number,
- revocation date,
- revocation reason.

| **Certificate Revocation List** | | |
|---|---|---|
| Version | | |
| Signature algorithm identifier | | |
| Issuer name | | |
| Certificate revocation list serial number | | |
| Last update | | |
| Next update | | |
| Revoked certificate serial number | Revocation date | Revocation reason |
| Revoked certificate serial number | Revocation date | Revocation reason |
| ... | ... | ... |
| CRL issuer's signature | | |

Figure 27: Example for a Certificate Revocation List

Revocation Certificate

Periodically issued certificate revocation lists do not allow for non-repudiation. Given the currently valid certificate revocation list (say from last month) and a business partner signature that was computed after the issuance of the certificate revocation list, the recipient does not have assurance of the validity of the signature (and hence, no non-repudiation).

To satisfy this requirement, the concept of revocation certificates is proposed. The revocation certificate is issued by a certification authority. It consists of the original certificate information and additional information about the revocation such as date and time and reason of revocation, the date of known or suspected compromise, the party who requested the revocation, and the name of the certification authority who performed the revocation. The certification authority then signs the revocation information and thereby creates a self-contained revocation certificate. Such revocation certificates can be stored and handled in the same way as public key certificates are. Legally, they also have a self-contained status.

| **Part** | **Contained Information** |
|---|---|
| Revocation Notice Header | Name of issuer. |
| | Unique identifier of issuer (optional). |
| Components of CRL entry | Certificate serial number. |
| | Revocation date. |
| | Reason. |

Figure 28: Structure of an ANSI X9.30-3 Revocation Notice

| CRL-Part | Contained Information |
|---|---|
| CRL Header | Signature algorithm identifier. |
| | Issuer name. |
| | Issuer unique identifier (optional). |
| | CRL number. |
| | Last update. |
| | Next update. |
| CRL Entry | Certificate serial number. |
| | Revocation date. |
| | Reason. |

Figure 29: Structure of an ANSI X9.30-3 Certificate Revocation List

Possible States of Certificates

There are three states of certificates:

**Valid**: The actual date and time is between the start and the end date and time of the certificate if there is a validity field. If there is no validity field, the certificate is valid if not in another state.

**Revoked**: The certificate has been revoked because of its compromising. The certificate may be compromised by the user, by an attacker or by the certification authority.

**Terminated**: The actual date and time is after the end date and time of the certificate.

Lifecycle

The history of a certificate including all states, dates etc. should be stored in the directory. Users should be able to access the directory and ask for a non-revoked terminated certificate.

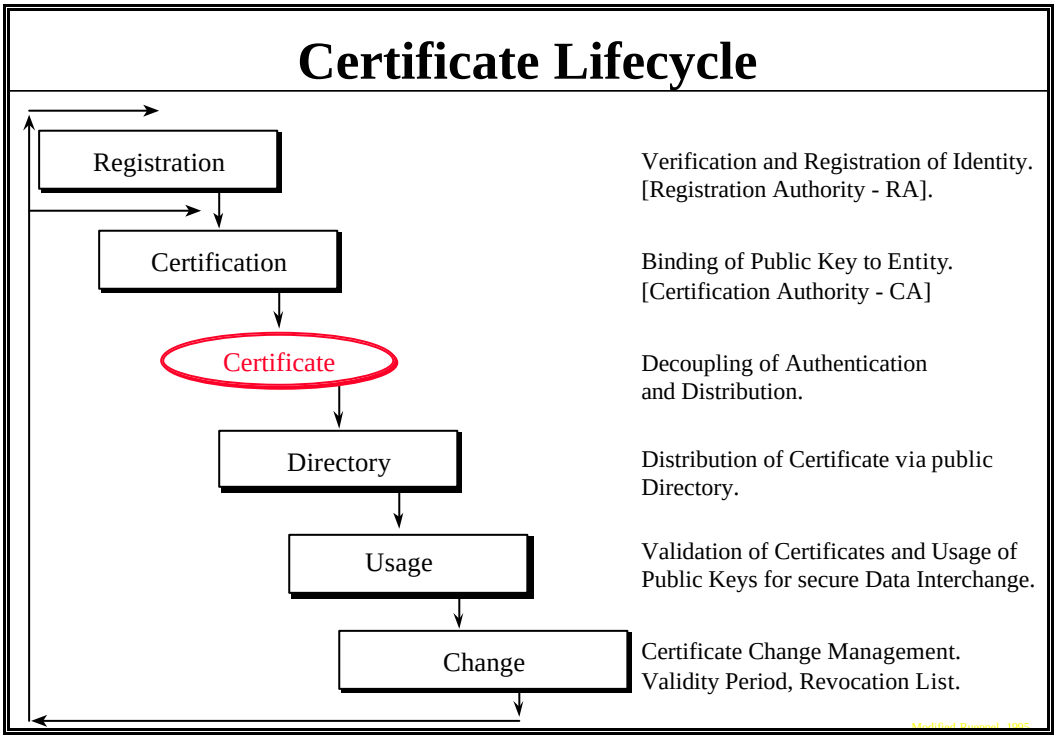


Figure 30: Certificate Lifecycle

**Black List**

The names of users who have been blacklisted, or whose keys have been revoked are listed in a socalled black list. One might wish to check the history of a user for the reasons and frequencies of revocations.

Unexpired certificates for users in the black list shall be revoked and included in the certificate revocation list. No new certificates are created for these users until they are no longer on the revoked user list.

| Black List | | |
|---|---|---|
| Version | | |
| Signature algorithm identifier | | |
| Issuer name | | |
| Black list serial number | | |
| Last update | | |
| Next update | | |
| Blacklisted user name | Date/time | Reason |
| Blacklisted user name | Date/time | Reason |
| ... | ... | ... |
| Black List issuer's signature | | |

Figure 31: Example for a Black List

### 2.4.1 Possible Content Fields of Certificates and Revocation Lists

The possible content fields of certificates and revocation lists are listed in the appendix.

### 2.4.2 Verification of Certificates

The verification of a certificate requires that each certificate in the certificate chain is verified and that each certificate correctly maps to the certification authority that issued the certificate.

In case the entity does not know the certification authority, it must be able to ask another party it trusts to get a certification of the authenticity of the first certification authority. For this the first certification authority has to send the address of the other certification authority. Such a certification on request must be possible irrespective of the nature of the trust infrastructure, and must be possible in cooperation with other trust infrastructures (interoperation).

In a hierarchical architecture of subordinated certification authorities the verifier has to verify many certificates - one after another in the hierarchical chain of authorities.

In a network model of certification authorities there may be cross-certificates of co-operating certification authorities (co-operation).

**On-line Verification**

By on-line verification of a certificate the current status of the certificate is determined by querying the appropriate certification authority on line and not by using a CRL. If on-line verification is allowed as a mode of operation the field for the validity period might be dropped.

The communications load for on-line verification depends on the number of processed certificates.

**Periodic Reverification**

If a certificate is used more than once during its validity period, periodic reverification provides a caching effect, saving the communications load for on-line checking. It also permits the supplicant to perform the on-line check and distributes that load away from the verifier.

The cached certificate verification is digitally signed by the verifier. Such a certificate is called certificate result certificate. It might be useful to add a validity period depending of the validity period of the certificate or a possible revocation.

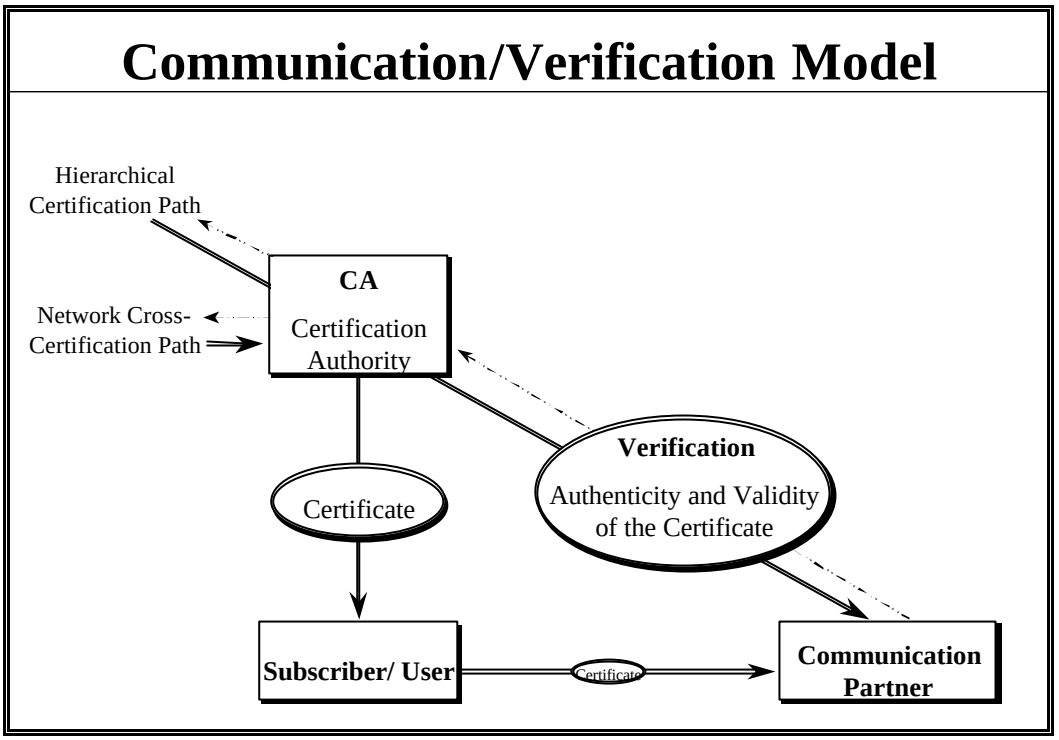


Figure 32: Communication/Verification Model

### 2.4.3 Time Stamps

There are situations where it might be important to know at what time and date a data item or document was published, signed or presented for the first time to somebody. This requirement can be fulfilled by a time stamp. A time stamp is a data item associated with time and date information assured by a (trusted) time stamping authority.

Time stamping is an important function in certification, and non-repudiation evidence recording services.

Time stamps can be used at a later date to prove that an electronic document existed at the time stated on its time stamp. For example if the certificate of a public key is cancelled on the grounds that the corresponding secret key has been compromised, a previously calculated digital signature by the said secret key would retain its legal value, if an independent time stamping had taken place before the cancellation of the certificate.

Because only the hashed document is time stamped, one can get a document time stamped without revealing its contents to the time stamping service. Actually the service generates a certificate containing the relevant time and date together with the hashed document. There is a procedure by which any receiver of such a certificate can verify it with the time stamp.

The use of time stamps is extremely important, if not essential, for maintaining the validity of documents over many years.

## 2.5 Directory Service

Directory services constitute a primary means of distributing certificates and other information regarding people and functional components that use or form part of the trust infrastructure. These services are used to support both digital signature management and key management. Directory services employ a distributed (as opposed to centralised) directory system.

The main directory-related services provided by a trust infrastructure are:

Encryption entities: Maintaining directory entries for subscribers or users; information included in such an entry might include, for example, public key certificates, network address, and contact information.

Signers: Maintaining directory entries for signers that participate in the digital signature system; information included in such an entry might include, for example, public key certificate and contact information.

Certification authorities: Maintaining directory entries for certification authorities; information included in such an entry includes public key certificates with that certification authority as the subject, public key certificates for other certification authorities issued by that certification authority, and certificate revocation lists issued by that certification authority. Also cross certificates issued.

| | |
|---|---|
| Names: | Ensuring that unique names exist for all objects (including encryption entities, digital signature entities, and certification authorities) in the trust infrastructure. |
| External directories: | Delivering non-confidential directory information to external directory servers used by entities of the trust infrastructure and/or entities outside the trust infrastructure. |
| Access control: | Maintaining access control information, and enforcing access control, to ensure that only properly authorised persons or systems can read information in the directory entries, and that only properly authorised persons can create or modify such entries. |
| Interoperation: | Providing access as needed to external directory services used in supporting interoperation between the trust infrastructure and external trust infrastructures. |
| Users: | Users store in their own directory certificates issued by licensed certification authorities. |

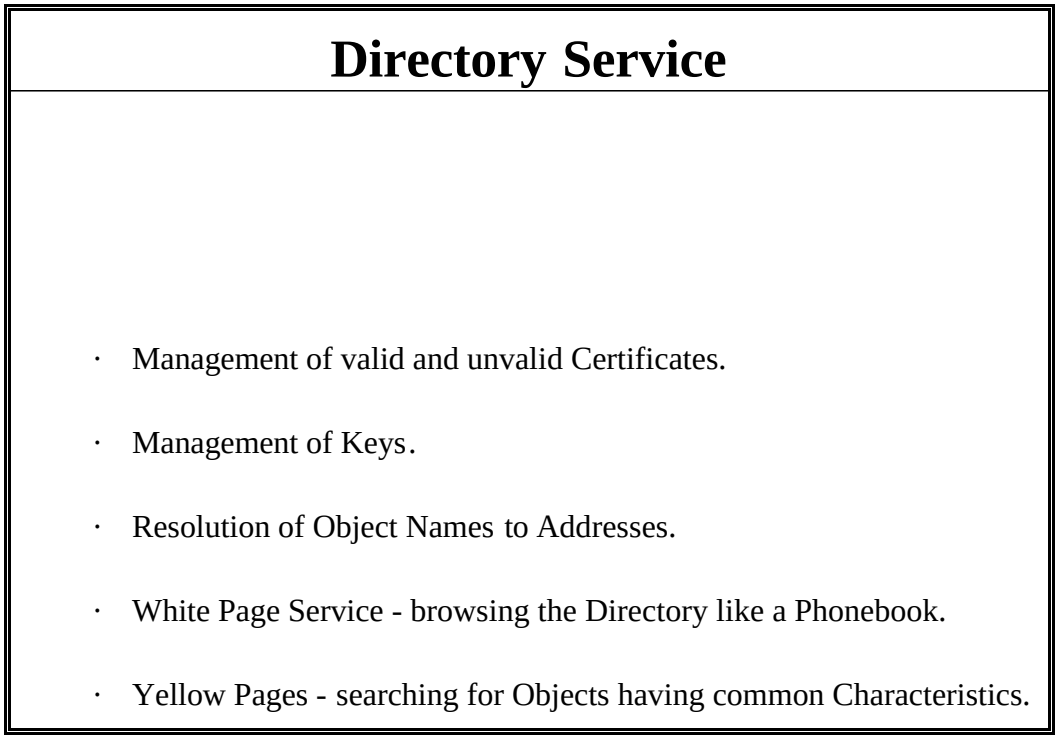


Figure 33: Directory Service

Distribution of Certificates

Among the procedures articulated by each policy certification authority in its policy statement are procedures for the distribution of certificates and certificate revocation lists by the policy certification authority itself and by its subordinate certification authorities.

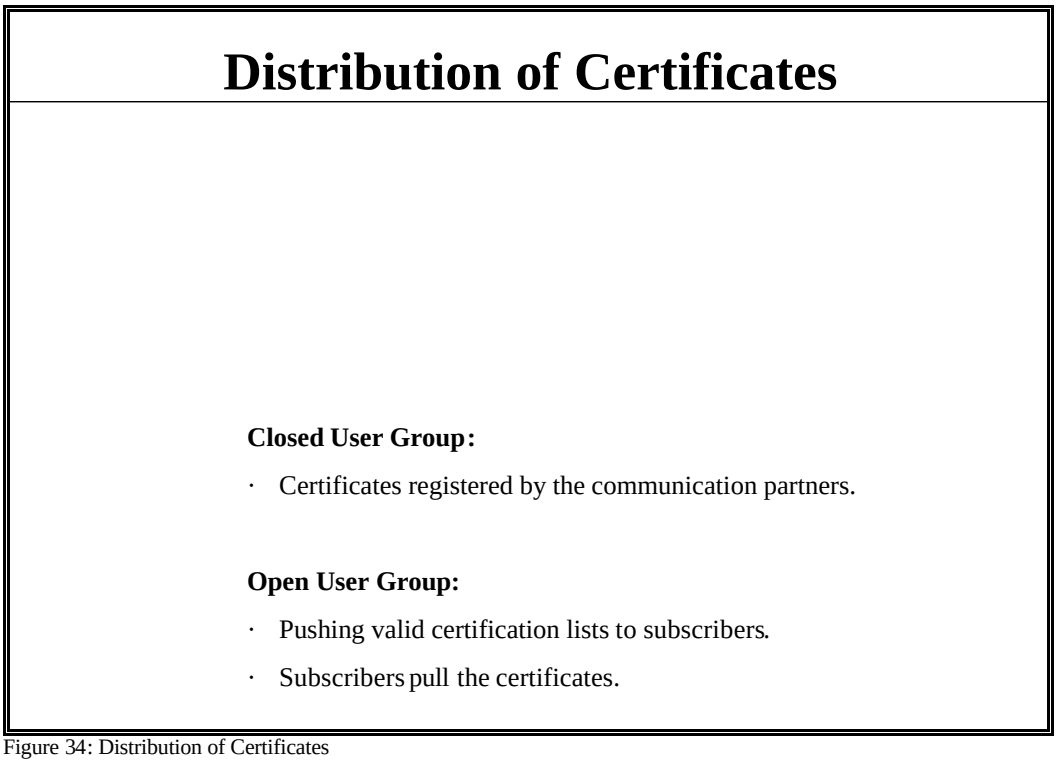


Figure 34: Distribution of Certificates

Distribution of Certificate Revocation Lists

The frequency of issue of certificate revocation lists may vary according to policy certification authority specific policy, but every policy certification authority and certification authority must issue a certificate revocation list upon inception to provide a basis for uniform certificate validation procedures throughout the hierarchy. The certificate revocation list will be updated periodically.

## 2.6 Architectural Alternatives for Public Key Infrastructures

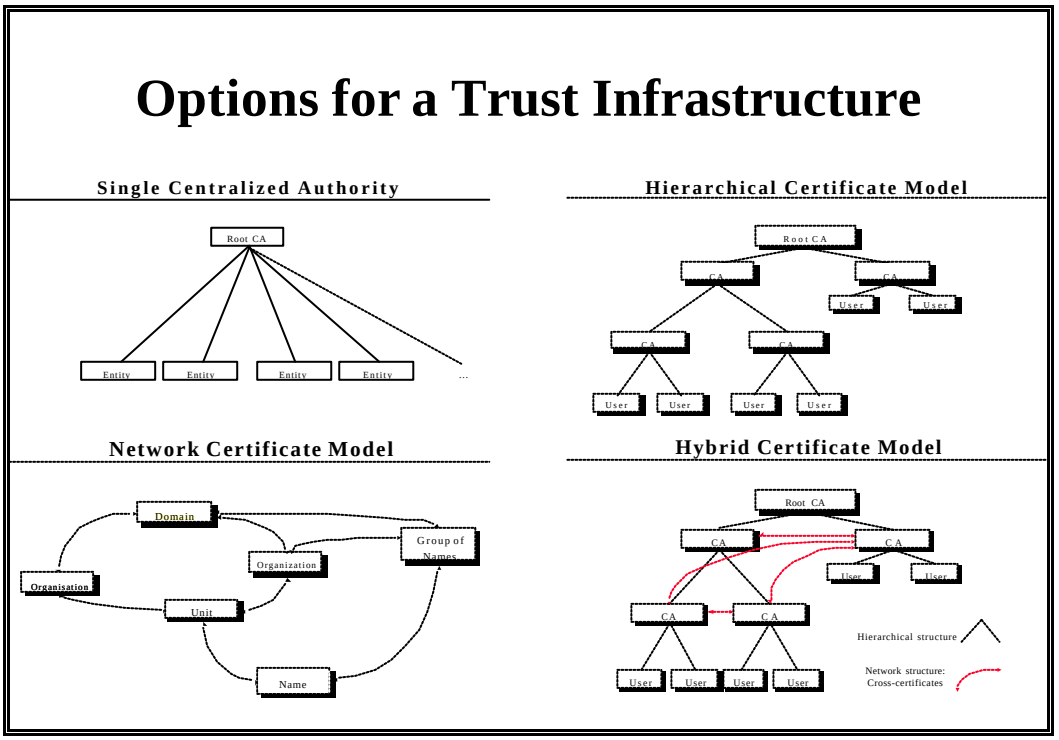


Figure 35: Options for a Trust Infrastructure

Certificates may be chained to form a certification path. Certification authorities can certify each other in some systematic manner to form a trust infrastructure. A certification authority may be issued a certificate by another certification authority. Two certification authorities may issue each other certificates; this is known as cross-certification, and the pair together is a cross-certificate. Three alternative trust infrastructure topologies, illustrated below are:

### 2.6.1 Centralised

A centralised architecture consists of only one centralised authority. Such a centralised authority is relatively inflexible and does not scale well in an environment where different sectors, different application areas and different countries might be involved. It seems also difficult for political reasons to promote such a centralised authority if different member states are involved.

### 2.6.2 Hierarchical

**Description**

Authorities are arranged hierarchically under a "root" certification authority that issues certificates to subordinate certification authorities. These certification authorities may in turn issue certificates to subordinate certification authorities, or to users. Every user knows the public key of the root certification authority, and any user's certificate may be verified by verifying the certification path that leads back to the root certification authority.

**Advantages**

The hierarchical trust architecture has some advantages. The structure of many organisations such as the government is largely hierarchical and trust relationships are frequently aligned with organisational structure. A hierarchical trust infrastructure may be aligned with hierarchical directory names and the certification path search strategy is straightforward. Each user has a certification path back to the root; the user can provide this path to any other user and any user can verify the path, since all users know the root's public key.

It is likely, however, that the strongest reason why early trust infrastructures have been hierarchical is that the hierarchy can be aligned with security policies and this alignment can be used to manage and determine the trust accorded to a particular certification path. While earlier versions of X.509 allowed networks of cross-certified certification authorities, they provided no mechanism to manage trust in such networks. Version 3 certificates provide alternative means for managing policies and trust.

**Disadvantages**

A strictly hierarchical certification path architecture has some disadvantages. It is improbable that there will be a single root certification authority for the world, therefore cross-certificates must exist at some level, and certification path verifiers must be able to cope with topologies that are not entirely hierarchical. Commercial and government trust relationships are not necessarily hierarchical, so using the hierarchy itself to manage trust relationships is surely not optimal. Moreover, compromise of the root private key is catastrophic because every certification path is compromised and recovery requires the secure "out-of-band" distribution of the new public key to every user.

### 2.6.3 Network

**Description**

Independent certification authorities cross-certify each other, resulting in a general network of trust relationships between certification authorities. A user knows the public key of a certification authority near himself, generally the local certification authority that issued his certificate, and verifies certificates by verifying a certification path that leads back to that trusted certification authority.

**Advantages**

The network certification path architecture has the advantage that it is flexible, facilitates ad hoc associations and trust relationships, and readily reflects bilateral trust relationships. It is likely that a national or world-wide trust infrastructure will evolve in an ad hoc fashion, from isolated certification authorities, and this is more easily accommodated in a network than a hierarchy. Certification authorities that are organisationally remote, but whose users work together with a high degree of trust, can be directly cross-certified under a high trust policy that is higher than would be practical through a long, hierarchical chain of certificates. The certification authorities whose users communicate frequently, can cross-certify directly, reducing certification path processing.

Perhaps the most compelling argument for a network trust infrastructure is that it is more convenient and natural for a certificate holder to place his trust in the local certification authority that issued his certificate, rather than a remote root certification authority, and make this the foundation of all trust relationships. Moreover, this simplifies the out of band secure distribution of the certification authority public key and recovery from the compromise of any certification authority's private key now requires only that the new public key be securely distributed to the holders of certificates from that certification authority, and new certificates be generated for them.

**Disadvantages**

The network trust infrastructure has at least two disadvantages: (1) Efficient certification path search strategies are more complex, and (2) a user cannot provide a single certification path that is guaranteed to enable verification of his signatures by all other users of the trust infrastructure.

### 2.6.4 Combined Architecture

**Description**

The hierarchical and network trust infrastructure architectures are not mutually exclusive. The following is a hybrid certification path architecture:

There will be a hierarchical path of certificates leading from the root certification authority to its subordinate certification authorities, and from each of these certification authorities to their subordinates, and so on, until every end user is issued a certificate with a certification path from the root certification authority. Each certification authority will have a single parent.

In parallel to the certificates hierarchically linking certification authorities to the root will be cross-certificate pairs attributes also linking those certification authorities. These parallel cross-certificate pairs are required. This will allow client applications that perform certification path verification from the verifier's parent certification authority, using the cross-certificate pair directory attribute, to operate from any certification authority.

Certification authorities may cross-certify each other along paths that do not parallel the hierarchy.

## 2.7 Naming

### 2.7.1 Naming: General

The most necessary conditions to be able to use modern telematic services securely are the availability of an infrastructure for a powerful directory service and for the management of public keys. In order to use the full benefits of these services a common naming concept is necessary that allows the different users of the telematic services to be able to identify each other. A name structure, common to the different application sectors is therefore required.

#### 2.7.1.1 Introduction

Whenever a person in the „conventional“ world needs to identify somebody or plans to engage in an information exchange with somebody he needs to have a criterion which allows him to reliably address that „somebody“. Names fulfil that requirement in that „a name is a word or a combination of words by which a person or thing is regularly known“ [Webster 96]. Names in general are identifiers which distinguish objects from one another and which can be used by different people to refer to the same object. Although they can be chosen arbitrarily, normally there exist conventions which allow for their systematic composition and assignment. The degree of binding between the object which has to be named and its name can range from very informal (by using temporary nicknames) to very strict (by formally certifying the name of a newly born baby in a birth certificate).

In software engineering naming is an important concept. Names are used to denote objects like users, programs, services, ports, or devices which may be the sender or the recipient of data. The use of names simplifies the structuring of software and improves its readability and understandability because detailed implementation structures of an object need not to be known if reference to the object is made. However in order to be applied systematically it is required that these names need to be unique, at least for a defined domain.

The interconnection of computers has expanded the domain over which names must be valid and unique from the local computer to global networks. Processes, like an information exchange by e-mail or a financial transaction between objects across networks, are only possible, if the participating objects can be mutually identified and authenticated beforehand. By adapting the concept of names from the "conventional" world to the „cyberworld“ - the world of networks - this requirement can be fulfilled.

The rapid deployment of computer networks and the growing number of objects participating in network communications has led to several conventions for the naming of objects which were developed to define and manage the namespace in networks in a simple and efficient way. The introduction of certificates to support trust policies in networks extends the application of naming conventions beyond the hitherto existing requirements for addressing in that a name of an object and its position in a sequence of objects in many cases plays an important role in the assessment of the trust that can be associated with a certificate.

It has to be noted that a name is not the only means by which an object can be located, an address fulfils the same purpose. However conceptually there is a clear distinction between the two concepts. Whereas a name is used to denote an object, an address specifies where this object can be found. If location attributes are added to a name (like Miller, 9. Ave. Beaulieu, 5$^{th}$ floor ) such a combination can be considered as an address. In most of the references that are the basis for this study, the expression „name“ is used for such a combination of a name and location information.

#### 2.7.1.2 Internet (Domain Name System, DNS)

Originally the ARPANET used two types of names, usernames and hostnames and the simple naming convention

"<username>@<hostname>"

was used by the user community for addressing e-mail. Both types of names did not contain structural

information.

The mappings from host name to network address were maintained centrally by the ARPA Network Information Center (NIC) in a single file. Changes in addresses were mailed to the NIC and to get up-to-date address information, this file was accessed through FTP by the hosts in the network and was duplicated in each individual host.

With the increasing number of hosts participating in the ARPANET and the addition of workstations connected by local networks this approach became unmanageable not only because of the size and the update frequency of the central name database but also because of the resulting problems with the dissemination of naming/addressing information. The old concept that required to wait for the Network Information Center to change its central address file before these changes could be made visible to the ARPANET members was not feasible anymore.

The decentralisation of computing power from central hosts to locally distributed workstations and the move from strictly local applications to distributed ones made it desirable for organisations to have the capability to structure and manage their names. The necessity for consistency of the name/address information in the network and the need to avoid name collisions led to the requirement for the creation of a general purpose name service.

The solution to these problems was the „Domain Name System (DNS)“. This system is based on the concept of „domains“. A domain is an administrative identity which can be considered as a region of jurisdiction for name assignment. The domain concept allows to partition the name management which would be required of a central administration and delegate it to subordinate administrations.

The DNS is a distributed database, which implements a hierarchical naming convention and allows for the maintenance and distribution of addresses and other information. The structure of the database can be represented as an inverted tree with the root at the top. The root domain is subdivided into top level domains (TLDs) which again are subdivided into lower level subdomains, the set of domains forms a hierarchy. The leaves of the tree are the lowest-level domains.

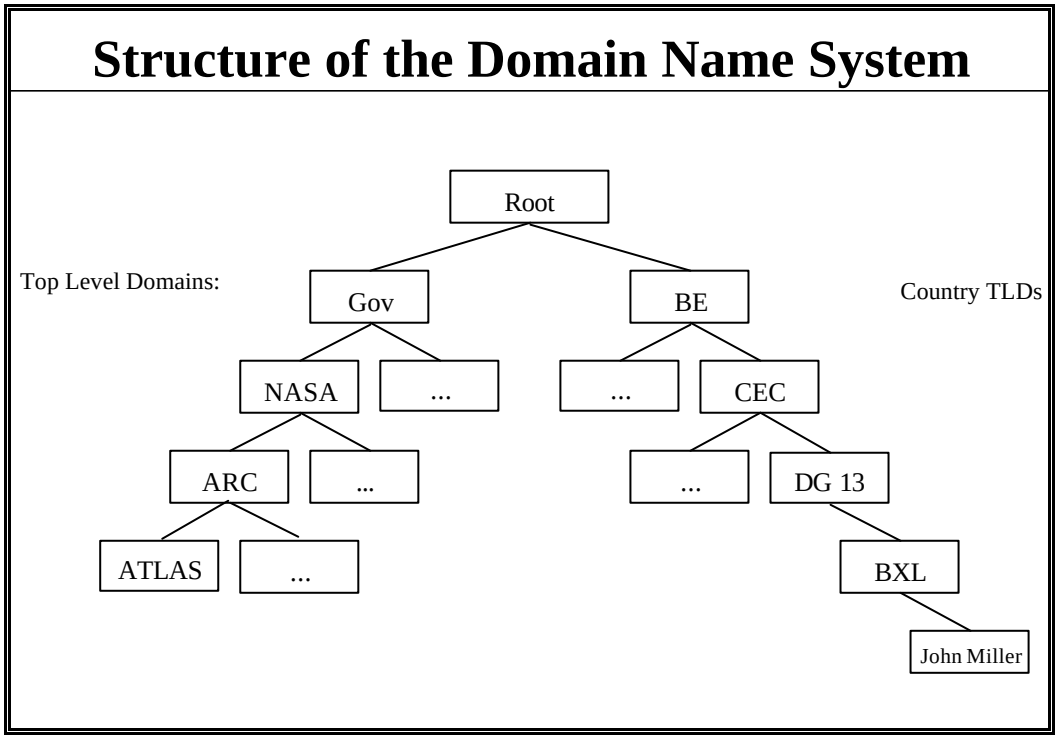


Figure 36: Structure of the Domain Name System

Each node in the tree represents a domain and is denoted by a label (simple name). Nodes originating from the same superior node have to be named uniquely in order to guarantee that a domain name uniquely identifies a single node in the tree. In addition to the label each domain has a „domain name“ which is unique in the Internet and identifies the position of the domain in the database structure. It consists of a concatenation of all the labels on the way from a specific domain to the root. The period sign („ . “) is used to mark the boundary between hierarchy levels and the names are written from the most specific on the left to the most general on the right.

Example: bxl.dg13.cec.be

In the Domain Name System, the first level of domains below the root consists of five generic top-level domains and of country code domains, they divide the Internet domain name space organisationally and geographically.

The names of the top level domains are:

| | |
|---|---|
| COM | Companies, commercial organisations. |
| EDU | Educational institutions. |
| INT | International organisations. |
| NET | Network organisations. |
| ORG | Non commercial organisations. |

and two domains which are generic to the United States of America

| | |
|---|---|
| GOV | Government organisations. |
| MIL | Military organisations. |

The addition of seven new top level domains, „FIRM“, „STORE“, „WEB“, „ARTS“, „REC“, „INFO“, and „NOM“ is currently under consideration by the International Ad Hoc Committee (IAHC), a group formed at the initiative of the Internet Society.

The country code domains are denoted by the English two letter code of the ISO Standard 3166 for "Codes for the Representation of names of Countries" e.g.

| | |
|---|---|
| BE | Belgium |
| DE | Germany |
| FR | France |
| UK | United Kingdom (GB can also be used a valid top level domain) |
| US | United States |

Domain names reflect organisational structures. For example in the domain name „cs.ucl.ac.uk“, „cs“ is the computer department, „ucl“ is the name of the university (University College London), „ac“ is the second-level domain in the UK which administers the academic domain, and „uk“ is the top level domain (United Kingdom).

In addition to the domain name, which is expressed in a user friendly form, each node on the Internet has a numeric address called its IP address. The Domain Name System is responsible for mapping domain names into IP addresses.

Examples:

| Domain Name | IP Address |
|---|---|
| atlas.arc.nasa.gov | 128.183.10.4 |
| askhp.ask.uni-karlsruhe.de | 192.67.194.33 |

Domain names, used for addressing e-mail are expressed as follows:

<user-name>@<domain-name>

Example:

John Miller@bxl.dg13.cec.be

#### 2.7.1.3 X.400 Message Handling System (MHS)

The X.400 Message Handling System (MHS) is a standard defined by CCITT and ISO. Its implementation enables users to exchange electronic mail (interpersonal messages) and provides the functionality for the exchange of formatted messages between computers (electronic data interchange) both on a store and forward basis. A message submitted by an originator will be delivered to the one or more message recipients by the message transfer component of the message handling system.

In a system for interpersonal messages, the principal object that requires naming is the user who is the originator and recipient of messages. A user in MHS is identified by one or more originator/recipient (O/R) names.

An O/R name consists of a directory name, an O/R address or of both. Either of the two components identifies a user unambiguously and can be used when submitting a message. These names and addresses are assigned by naming authorities.

A directory name is a user friendly name which is provided in a directory and can be used to find out the corresponding O/R address. The structure and components of directory names follow the rules described in the CCITT X.500-Series of Recommendations. Directory names are more stable and user-friendly than O/R addresses and their use will grow if a directory will be widely available. In the meantime as long as there is no directory available, a mnemonic O/R address can be used which provides a user friendly means of identifying users.

Mnemonic O/R addresses i.e., addresses which are intended for human usage, consist of a set of stan-

dard attribute types defined in X.402. These attribute types can be represented in a shorthand form by labels to make their use more convenient.

Examples:

| Attribute type | Label | Attribute value |
|---|---|---|
| Country | C | BE |
| Administrative Management Domain | A or ADMD | RTT |
| Private Management Domain | P or PRMD | CEC |
| Organisational unit 1 | OU1 | BXL |
| Organisation | O | DG13 |
| Surname | S | Miller |
| Given Name | G | John |

The attribute values area case insensitive, however if label/value pairs appear in sequence on a line, the user of upper case for labels and lower case for values is recommended to make the address better readable for the user.

An example of a typical X.400 O/R address would be:

G=john;S=miller;O=dg13;OU1=bxl;P=cec;A=rtt;C=be

The sequence of attribute/value pairs of an address is recommended in X.401, however this sequence can be adapted to the respective cultural conventions. Basically the sequence is not critical.

If it should be necessary to indicate that a X.400 address is meant, the term "X.400:" should be added at the beginning of the address.

X.400:G=john;S=miller;O=dg13;OU1=bxl;P=cec;A=rtt;C=be

In addition to these standard attributes there is a possibility for domain defined attributes - if needed - whose syntax and semantics can be defined by the respective management domains.

#### 2.7.1.4 X.500

X.500 is an international standard for providing electronic online directory services. In X.500 each object and entity has a directory name which unambiguously and uniquely identifies an individual entry in the Directory Information Base (DIB). There are two types of Directory names:

- Distinguished Names (DN) and
- Alias names.

To support the administration of the Directory Information Base each entry also has a Relative Distinguished Name (RDN)

**Relative Distinguished Name (RDN)**

A Relative Distinguished Name is a name that identifies a particular entry in the Directory Information Base. Each entry in the Directory Information Base has a Relative Distinguished Name except the root entry. The allocation of a Relative Distinguished Name is an administrative task, that is performed when the entry that represents an object is registered for the first time in the Directory Information Base.

When specifying a Directory Information Base, each entry has to be classified according to the object it represents taken from an object class that is typical for that object.

Examples for object classes are:

| Object Class | Entry representing |
|---|---|
| Country | countries |
| Organisation | organisations |
| Locality | localities or regions |
| Organisational Person | people employed by an organisation |

Examples for object classes and related attributes are:

| Object Class | Attributes |
|---|---|
| Country | country name (mandatory)<br>description (optional) |
| Organisation | organisation name (mandatory)<br>business category (optional)<br>description (optional) |
| Organisational person | common name (mandatory)<br>surname (mandatory)<br>organisational unit name optional)<br>telephone number (optional)<br>title (optional) |

Figure 37: Examples for Object Classes and Related Attributes

The definition for the object class determines which mandatory and optional attributes belong to the entry. At least one attribute type and value (distinguished value) of the entry is used to specify a Relative Distinguished Name for an entry.

For example the Relative Distinguished Name of an entry with the object class „country“ could be:

C=BE

where „C“ is a label for the country name attribute and „BE“ is the value for the country name attribute (Belgium) expressed as a two letter code, taken from International Standard ISO 3166.

**Distinguished Name (DN)**

A Distinguished Name identifies an object and its entry within the Directory Information Base unambiguously and uniquely. These properties are derived from the tree structure in which the entities are arranged. A Distinguished Name of a given object is defined as that name which consists of the concatenation of all Relative Distinguished Names from the root down to and including the entry itself. The resulting Distinguished Name therefore has a hierarchical structure.

Every object entry and alias entry has precisely one Distinguished Name, these names are intended to be user friendly, with values that are taken directly from the real world environment of the user community when specifying the Relative Distinguished Names.

The syntax to express Distinguished Names is based upon the syntax used to define mail addresses in X.400 (O/R address). For example, if a hierarchy includes the hierarchical classes: country, organisation, organisation unit, the Distinguished Name of a person named John Miller might be:

Country=Belgium;Organisation=European Commission;Organisational Unit=DG13;Person=John Miller or

C=BE;O=CEC;OU=DG13;CN=John Miller.

An example which clarifies the concepts of Relative Distinguished Names and Distinguished Names is shown in the following Figure 38.

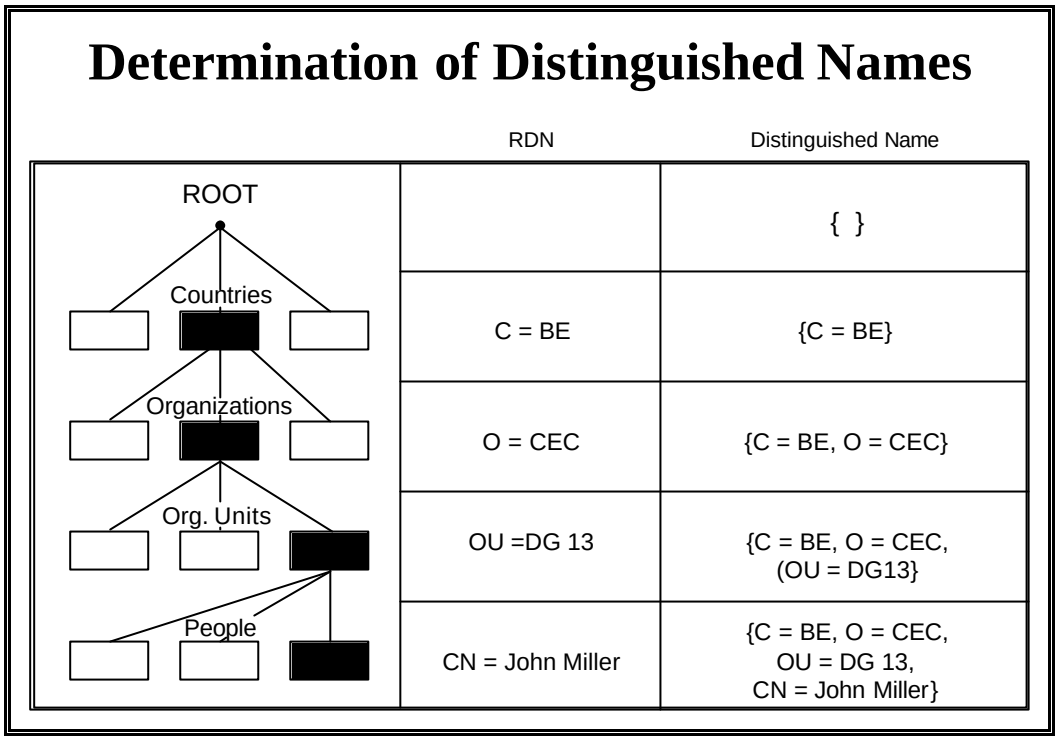


Figure 38: Determination of Distinguished Names

This approach to use Relative Distinguished Names and Distinguished Names for naming is similar to the numbering scheme used for telephony where a tree structure is used to arrive at a telephone number, which is unique in the world. In such a system the concatenation of the local number, the area code and the country code guarantee the uniqueness in the world. Local number, area code and country code can be viewed as Relative Distinguished Names in a numbering tree and the concatenation can be looked at as the Distinguished Name (number) of a particular telephone extension.

**Alias Names**

Although each entry has one Distinguished Name, it may have several alias names. An entry therefore can be referred to by either using its Distinguished Name or one of its alias names. Syntactically an alias name looks like a Distinguished Name, however it does not represent the direct path from the root of the Directory Information Tree to an entry. The path leads to an alias entry, and the alias entry has an attribute that contains the name of the entry to which the alias refers.

#### 2.7.1.5 X.435 (Electronic Data Interchange, EDI)

The capability to exchange a structured set of information between computers, such as invoices or purchase orders, is a prerequisite for the Electronic Data Interchange (EDI). One of the major applications of EDI is the exchange of business data between the computer systems of trading partners.

A trading relationship exist between at least two parties. The identity of these parties must be associated with the specific EDI message being exchanged. EDI trading partners identify each other by an „EDI name“ which is a contractual name or an identifier needed to identify uniquely an organisation for trading purposes. Although EDI messaging users may be persons or computer processes the naming conventions are expressed in a notation which is more applicable for communication between computers than between individuals.

This EDI name (“EDI sender/recipient identifier“) is represented as a string of alphanumeric characters and structured as follows:

| | | | | |
|---|---|---|---|---|
| 1234 | 123 | 123456 | 123456 | 12 |
| **A** | **B** | **C** | **D** | **E** |

Figure 39: Example for an EDI Name

A: ICD (International Code Designator) value allocated by ISO to the national Registration Authority (e.g. Chambers of Commerce, SWIFT, EAN, Dunn & Bradstreet) used as a qualifier for the scheme (4 digits).

B: numerical value allocated by the Registration Authority to the regional sub-authority (3 digits).

C: numerical value allocated by the sub-authority to the registered organisation (mandatory part of

the identifier; 6 digits)

D: numerical value used by the registered organisation (free part; 6 digits).

E: numerical check digits calculated by the registered organisation (2 digits).

An EDI name does not contain a geographic element, such as country of operation.

EDI names are issued by internationally recognised registration authorities (e.g., Dun & Bradstreet, EAN International, SWIFT) or by standards organisations (e.g. EDIFACT and ANSI X12) or they are formalised names issued by a multi-national company where the name is unique within the company's trading community and the multi-national company acts as a naming authority within this community. Finally there is the possibility for a free form name assigned by the trading partners themselves, subject only to a uniqueness check by the organiser or operator of the community, acting in the role of a naming authority.

To allow EDI users to take advantage of a comprehensive and widespread X.500 Directory which might be available in the future, the EDI Registration Authorities (EDIRA) have elaborated and signed a Memorandum of Understanding which defines the procedure how to convert an EDI name into a X.500 Distinguished Name.

#### 2.7.1.6 X.400/Internet Personal Naming Recommendation

The World Electronic Message Associations (WEMA) has taken the position that global messaging will only become a reality after a simple naming and addressing standard has been agreed. Therefore the following recommendation for adoption as a „standard“ has been published:

- Every electronic messaging user should be addressable by at least his/her First Name and Family Name irrespective of the messaging system he/she is using, X.400 or Internet. (Initials should only be used to make a name unique in case of duplicate names in one domain).
- The success of the use of e-mail in any organisation depends on a high quality directory from which users can select names to address their messages.

To allow „auto-registration“ of names and addresses from ones customers and suppliers into ones directory it is essential to have a standard format for names and addresses.

Whereas the current recommendation only looks at personal names, recommendations for non-personal names are planed to follow in separate documents.

The recommendation is based on the existing international X.400/X.500 standards for e-mail exchange and directories.

The syntax for personal names is defined as follows:

| | |
|---|---|
| X.400 | G=FirstName;S=Surname;I=Initials |
| Internet | FirstName,SurName |

FirstName and Surname are mandatory fields, Initials is optional.

Examples of valid name formats:

John Miller

Surname = Miller

Given Name = John

X.400: G=John;S=Miller;O= .....

Internet: John Miller@somewhere.somewhere.com

### 2.7.2 Name management

#### 2.7.2.1 Name service and directory

One of the basic problems of communication in networks is to find the specific object with whom one wants to communicate with. In small networks simple mechanisms have been sufficient to provide such a location service. For example address lists or a single central database have been used in early ARPANET configurations to provide for a cross-reference between names and addresses for all objects in the network. As the network became larger, a growing number of users recognised that theses simple services were not sufficient anymore and the requirement for a comfortable name service evolved.

A name service in a network is a service whose main task is to provide access to cross references between names of objects and the location of these objects or more technically, to provide name-to-address mapping. Performance, flexibility, and manageability are essential requirements for a name service and its easy use combined with reliable operation contribute significantly to the acceptance of the services of a network.

Basis for a name service in networks is a database which allows to store and retrieve name and address information of objects reliably, timely and in an economic manner. Such a database is commonly called a directory, it holds information about objects and provides users with services to access that information.

The use of a directory has the following advantages:

- it allows to refer to objects by name rather than by address,
- it provides the mapping from names to addresses,
- it is theoretically unlimited in size,
- it allows to facilitate the management of the name space supported by the network by allowing to partition the name space into domains that can be physically distributed and be managed separately;
- it provides the possibility, in addition to address information, to store and retrieve a broad spectrum of data about an object, like for example certificates belonging to that object, organisational or postal address information etc.,
- it provides functions to search, find and use information about the network itself and therefore supports its administration;
- it isolates as much as possible the user of the network from the frequent changes to that network,
- it allows for operations that cross national boundaries, by providing functions to translate names from/to different naming systems,
- it can provide support for authentication of the objects taking part in the information exchange.

The most widely known directory is the X.500 Directory as defined by ISO/ITU which can be used for all types of distributed electronic directories. Originally it has been designed in connection with the X.400 electronic messaging standard to serve as a world-wide directory of electronic mail addresses.

In its current form the X.500 Directory standard provides the framework for implementing directory capabilities required by OSI applications, OSI management processes, other OSI layer entities, and telecommunications services. An X.500 Directory allows to locate objects wanting to communicate electronically with each other and it provides features for “user-friendly naming“, whereby objects can be referred to by names which are familiar to human users.

By establishing the X.500 Directory it is intended to create through a single, unified, name space, one logical directory (in theory world wide) capable of being distributed and serving many different applications.

The Directory uses a database (Directory Information Database, DIB) to store information about objects of interest and to provide access to them. An object can be anything that is identifiable. Typically, it is a person, an application-entity, a file, a distribution list etc. The Directory Information Base is composed of entries, each of which consists of a collection of information on one specific object and there is precisely one object entry which represents an object. Alias entries are used to provide alternative names for object entries.

In order to satisfy the requirements for a simple distribution and management of a very large Directory Information Base, and to ensure that entries can be unambiguously named and rapidly found, a hierarchical structure of the database has been selected which is derived from the observation that relationships commonly found among objects in the working environment are hierarchical.

The experience with the international telephone numbering system or with file systems in operating systems (UNIX, DOS) has shown, that hierarchical structures are well suited to fulfil the requirement for uniqueness of names. The use of hierarchical structures in a name service supports the decentralisation of naming authority, it allows for an easy growth of the name universe and it provides flexibility because a new domain or an additional hierarchical layer can easily be added.

If the entries are arranged according to this model the result is a tree-like structure which is called Directory Information Tree (DIT), where, by convention, the root of the tree is at the top and the nodes in the tree represent the entries. Object entries can either be located at the nodes or the leaves of the Directory Information Tree, alias entries are always located at its leaves. Entries higher in the tree will generally represent objects such as countries or organisations, while entries lower in the tree will represent people or application processes.

The following figure represents a hypothetical example of a Directory Information Tree:

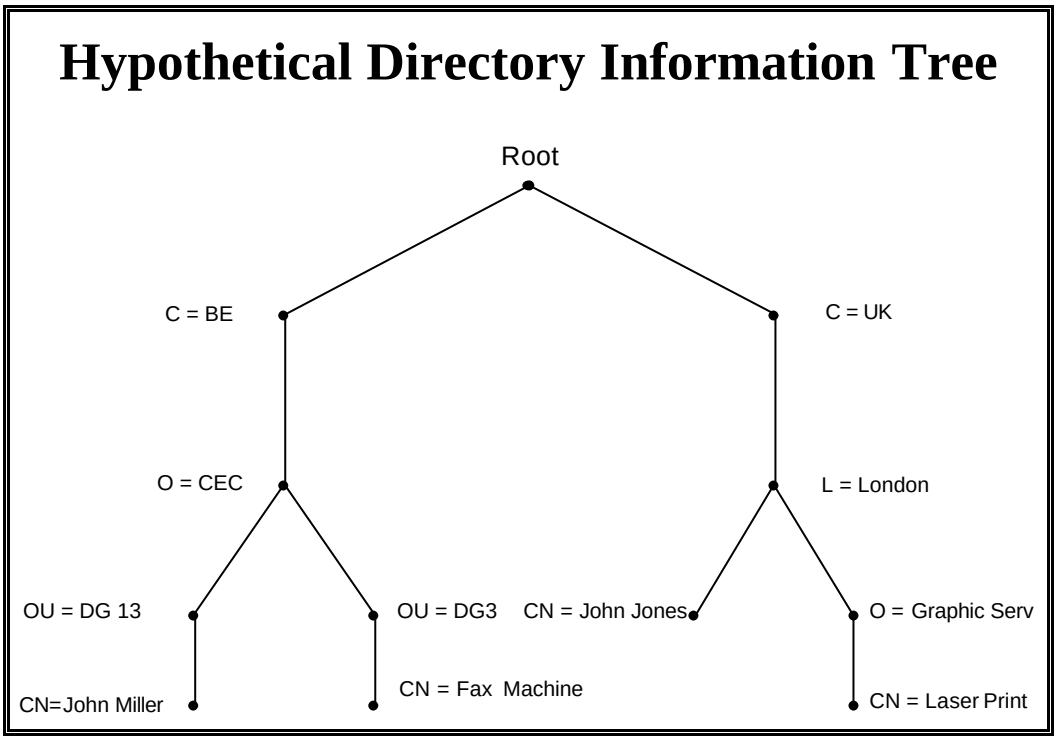


Figure 40: Hypothetical Directory Information Tree

Each entry in the Directory Information Base stores its information about an object in a set of attributes and each attribute consists of an attribute type which can have one or more values.

The following list shows some examples for attribute types and their values, it also gives examples for the notation using labels which are used as a shorthand expression for the attribute type:

| Attribute Type | Attribute Value | Attribute Label | Example |
|---|---|---|---|
| country name | BE; DE; ES; FR; GB, US | C | C=BE |
| locality name | Brussels | L | L=Brussels |
| organisation name | European Commission | O | O=CEC |
| organisational unit name | DG XIII | OU | OU=DG XIII |
| common name | John Miller | CN | CN=John Miller |

Entries for each object in the Directory Information Base consist of a set of attribute types, this set is open ended. However there is a set of attribute types which is internationally standardised, others might be defined by national administrative authorities and/or private organisations.

The growth and form of the Directory Information Tree, the definition of the Directory schema (the logical description of the database), and the selection of names for entries as they are added, is the responsibility of various authorities, whose hierarchical relationship is reflected in the shape of the tree. The authorities shall ensure, for example, that all of the entries in their jurisdiction have unambiguous names, by carefully managing the attribute types and values which appear in those names. Responsibility is passed down the tree from superior to subordinate authorities, with control being exercised by means of the schema.

In order to allow different applications to access the Directory without significant modifications the naming conventions should specify that only the leaf entries or entries close to leaf entries should have application specific forms of names.

#### 2.7.2.2 Naming conventions

Most people prefer names in computer systems and networks which can easily be related to the object in the real world which they represent. The same preference is true for names in certificates, these names should be descriptive, i.e. they should clearly indicate the object in the real world to whom the public key in the certificate belongs (exception is if the object wants to remain anonymous) and who has signed the certificate.

For example if using digital signatures, names for signers are most critical, because they are used in certificates and because they must be recognisable to verifiers (potentially world wide).

Naming conventions establish the rules which determine the syntax and semantics of names and therefore provide the basis for the fulfilment of these requirements. Because the distinguishability of objects depends largely on the uniqueness of their names. A general, systematic, and practical method of naming is necessary.

Initially it seems to be relatively easy to design naming conventions that allow for the creation of unique names. Such a convention could be based on pure numbers which could be used as names and assigned to objects. This approach might be applicable for certain application domains but it does not seem to be feasible globally.

A simple naming system is one of the essential prerequisites for the acceptance of a trust infrastructure. It has to be based on well defined naming conventions, that specify common rules to which all names must adhere to and which are primarily oriented towards the user. Additionally it contributes to the trust which is granted to the infrastructure if the naming system can easily be understood and used.

The proper choice of the naming convention is a trade-off between various conflicting requirements from the user side, like user-friendliness and from the technical side like validation and translation overheads for names into addresses. There are some basic requirements that a naming convention should satisfy:

- it should be general enough to allow to name a broad spectrum of objects in different applications as well as in different environments. This spectrum may reach from people on one side to hosts and terminals on the other side,
- it should be possible to apply the same naming conventions for all name assignments and to use standard attributes for name design,
- it should provide for stability of names,
- it should be user-friendly i.e. names should be easy to construct, to be understood world-wide and to remember, and it should exclude the possibility for misinterpretation of names as far as possible,
- it should have the capability to allow to identify an object unambiguously,
- it should be governed primarily by the end-user requirement for user-friendliness. Aspects like mapping to network addresses, conversions to internal character sets, translation to other formats etc. should be left to the technical implementation and should be hidden from the user,
- it should be possible that an object can be known by different names in different environments, therefore the naming convention should allow the feature of defining multiple names and leave the problem of name resolution to some address translation mechanisms,
- it should be flexible enough not to force any change in existing name schemes, even when they are not in line with existing international proposals.
- Although in theory there is a clear distinction between a name and an address, both expressions are often used synonymously. This has resulted in two views on names which have to be considered and combined in a naming convention:
- the personal name of an object that represents its identity (e.g. firstname/surname for persons, type for devices) which is used in day-to-day life and
- the name of an object that allows to identify it uniquely in a distributed system by combining his common (personal) name with structural information (e.g. a Distinguished Name in an X.500 environment).

As far as common (personal) names are concerned which represent the identity of a user, the different cultures have conventions which define how to compose these names. The naming conventions should build on these cultural conventions and should state how they implement them.

The conventions for names necessary to identify objects uniquely in a distributed network might not only have to take into account the topology of the network but there might also be administrational and geographical aspects which need to be supported by the naming convention. The name can be used to convey organisational information related to the object and his position in the system and it can also be used to support certain operational properties like trust (e.g. “PEM subordination rule“).

| Naming convention | Sequence of Attributes in the Tree Structure | Significance of Attributes in Writing | Example |
|---|---|---|---|
| DNS | leaf to root | left to right | dg13.cec.be |
| X. 400 | --- | not critical | OU=dg13;O=cec;C=be |
| X. 500 | root to leaf | right to left | C=BE;O=CEC;OU=DG13 |

Figure 41: Naming Conventions

Figure 41: Naming Conventions shows examples of the sequence of attributes in a name and the resulting different notations. In the Domain Name System the most significant attribute is at the far right end of the name, whereas in X.500 it is at the far left end. In X.400 mnemonic O/R addresses there is no

mandatory sequence of attribute/value pairs required.

A naming convention also has to define how a name should be technically represented, for example as

- Character string: Character strings offer user-friendliness because a name can be represented as an expression in a natural language which is meaningful to the user.
- Numerical string: Numeric strings allow for a direct use and do not need complicate translation mechanisms however they are difficult for the user to learn and to remember.
- Variable length string: Variable length strings offer almost unlimited opportunities for the allocation of names. Such names facilitate the communication between objects of different networks. However variable length names, sooner or later, will complicate the process of name translation.
- Fixed length string: Fixed length strings offer advantages in the processing of names but limit the flexibility,

or combinations of these basic alternatives. In addition things like case sensitivity and separators between attributes should be defined.

A final aspect that has to be considered in a naming convention is data protection, that is how much information about an individual user can be part of his name without infringing his privacy.

#### 2.7.2.3 Naming authorities and user registration

The uniqueness of names is a fundamental prerequisite for the correct operation of a certification scheme for public keys. To fulfil this requirement in a rapidly changing environment, as represented by today's information society, is a complex task which needs a considerable administrative effort and an organisation or organisational elements dedicated to that task. Naming authorities are trusted to take this responsibility, they can be established at different levels for example at international, national, corporate or department level.

In general a naming authority is responsible for the allocation of names in a specific domain based on agreed naming conventions, which determine the syntax and the semantics of names. The definition of such domains can be based on geographical, technical, sectorial or other appropriate criteria.

Technically speaking, a naming authority has control over some part of the structure of a data repository. In a hierarchically structured X.500 Directory database for example, this could mean that the naming authority has control over some region of the Directory Information Tree. Naming authorities might reside at different levels of the Directory Information Tree, however on every level they have to assure that the names assigned at the immediately succeeding levels are unambiguous, thereby contributing to the uniqueness of names.

A naming authority has the following basic functions:

- to develop and implement policy for name assignment,
- to specify the procedures and means to validate a users claimed identity,
- to assign names and ensure proper distinction among names,
- to assure the conformity of intended names with the naming convention,
- to specify deviations from the naming conventions, if necessary, and publish these,
- to maintain a repository of assigned names,
- to detect potential, unintended duplicates,
- to develop rules for resolving name conflicts.

Depending on the selected architecture for a trust infrastructure these basic functions can be assigned to different levels of an infrastructure and to different authorities. The analysis of existing or proposed architectures indicates that generically three levels of authorities can be anticipated: upper-level, middle-level, and lower-level naming authorities, which in most cases are co-located with Certification Authorities.

Naming authorities

An upper-level naming authority has
- to ensure that the names of middle level naming authorities follow the naming conventions,
- to detect potential, unintended duplicate certification of the names of lower-level naming authorities and it has to provide this information to the middle-level naming authorities. This information is the basis for ensuring global uniqueness of the names of lower-level naming authorities.

A middle level naming authority has
- to specify the policy and procedures which govern the naming of lower-level naming authorities it certifies, and how this policy applies transitively to entities (end-users or subordinate lower level authorities) certified by these lower level naming authorities.
- to state what procedure has to be used to verify the claimed identity of a lower level naming authority,
- to specify the requirements and mechanisms that have to be used by lower-level naming authorities to validate the identity of end-users,
- to specify the procedures used to resolve name conflicts.

A lower level naming authority has
- to validate an end-user's claimed identity,
- to assign names to end-users based on a name space given by the middle-level naming authority
- to maintain a database of the names which it has certified and to take measures to ensure that it does not certify duplicate names for users or subordinate lower-level naming authorities,
- to implement procedures to ensure that the same subject name isn't issued to multiple users in case of issuance of „PERSONA certificates" (certificates for users who wish to hide their identity).

There are several mechanisms that can support a naming authority:

| | |
|---|---|
| Technically | by installing software that screens the Directory Information Base periodically for naming conflicts and that supports the naming authority in the selection of new names by providing an up-to-date overview of already assigned names, by implementing a directory. |
| Organisationally | by establishing structures which allow for decentralisation of the naming process (e.g. tree structure), |
| Procedurally | by setting up rules that avoid duplication of names (like the PEM subordination rule), by establishing an information exchange process on assigned names between different naming authorities in an infrastructure. |

Registration Authorities

The function of user registration is performed by registration authorities. The details of user registration including the decision which organisational element should fulfil that function are a local matter, subject to policies established by the user's lower level naming authority and the middle level naming authority under which that authority has been certified. In general a user must provide, at a minimum, his public key and a name to a lower level naming authority, or a representative thereof, for inclusion in the user's certificate. The lower level naming authority will specify procedures and credentials (e.g. birth certificate, personal ID card, notarial attestation, drivers licence) in accordance with the policy of its middle level naming authority, to validate the user's claimed identity and to ensure that the public key provided is correctly associated with the user whose name is to be bound into the certificate.

## 2.8 Proposals, Projects, Products and Implementations of Certificates

### 2.8.1 ISO/IEC/ITU X.509 Certificates

**Generic Certificate Structure**

A certificate has the following generic structure:

| Name and Part of the Certificate | Functions and Responsibilities | Example |
|---|---|---|
| Version | The certification authority adds certification specific data required for the proper functioning of the overall system. | X.509 v1 |
| Serial number | Identifies the certificate. The number is assigned by the Certification Authority. | 471234560078 |
| Algorithm identifier and parameters of the signature | Specifies the signature algorithm and associated hash function used to sign the certificate. | FEE with RIPEMD-160 |
| Issuer | Distinguished Name of the Certification Authority that issued the certificate. | The_Trusted_Third_Party_Inc. Brussels. |
| Validity | Time period that the certificate is valid. Start and end of a period. | October 1, 1996<br>December 31, 1996 |
| Subject | Distinguished Name of the certificate user (entity). The registration authority processes the entity's application, for validating the entity's credentials and for assigning a Distinguished Name to the entity. | European Commission |
| Subject public key information | The entity generates an asymmetric key pair for signatures and for submitting the public key for verification to the registration authority. | 9G82KrD2g ... |
| Signature | The overall signature of the certification authority binds the public key to the entity's name. | M2qGrfqZOS5Aeo5 ... |

Figure 42: Generic Certificate Structure (X.509)

**Version 1 and 2**

The standard known as ITU-T X.509 (formerly CCITT X.509) or ISO/IEC 9594-8, which was first published in 1988 as part of the X.500 directory recommendations, defines a standard certificate format. The certificate format in the 1988 standard is called the version 1 (v1) format.

| Part | Contained Information | |
|---|---|---|
| Version | Version number; an integer, value is "2" for version 3. | |
| Serial number | Unique identifier for each certificate generated by issuer.<br>Identifies the certificate. A unique integer is assigned by the Certificate Authority. | |
| Signature | Algorithm identifier. | Hash and signature algorithm used to sign. |
| | Parameters. | Any parameter needed. |
| Issuer | Distinguished Name of certification authority that issued the certificate. (X.500 "Distinguished Name", a sequence of RelativeDistinguishedNames that uniquely identify a directory object). | |
| Validity | NotBefore. | UTCTime. |
| | NotAfter. | UTCTime. |
| | Time period that the certificate is valid. | |
| Subject | Distinguished Name of the certificate user (X.500 "Distinguished Name"). | |
| Subject's public key info | Algorithm identifier. | Subject's signature algorithm. |
| | Public key. | Subject's public key. |
| | Contains the user's public key. For DAS, it may also contain parameters. | |
| Issuer's signature | | |

Figure 43: X.509 Certificate Version 1

When X.500 was revised in 1993, two more fields were added, resulting in the version 2 (v2) format. These two fields are used to support directory access control. A widely accepted format for certificates is specified in the version 2.

| Part | Contained Information |
|---|---|
| Issuer unique identifier | Contains additional information about the subject; must be version 2 or higher(optional). |
| Subject unique identifier | Contains additional information about the issuer; must be version 2 or higher(optional). |
| Extensions | Sequence of fields (optional). |

Figure 44: X.509 Version 2 & 3 Certificate Description

The experience gained in attempts to use v1 and v2 certificates, especially in PEM (RFC 1422) made it clear that these two certificate formats were deficient and too restrictive.

**Version 3**

In response to the practical experience with v1 and v2 certificates, ISO/IEC and ANSI X9 developed the X.509 version 3 (v3) certificate format. The v3 format extends the v2 format by adding additional extension fields. Some extension field types are specified in standards, other may be defined and registered by any organisation or community. In 1966, standardisation of the basic v3 format was completed.

X.509 specifies the contents and the structure for a user certificate, certification authority certificate and a cross certificate pair.

The X.500 standard, especially the version 3 and the ISO standard extensions, is widely used for public key certificate formats. This will provide maximum compatibility and will facilitate the fulfilment of local requirements through extensions.

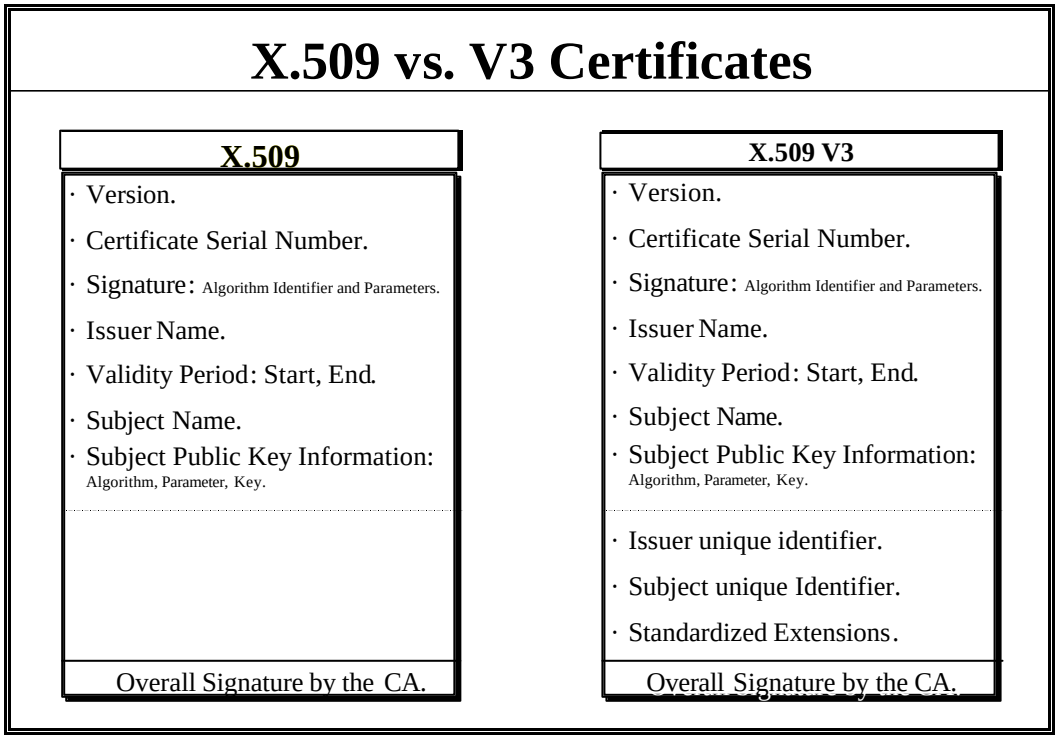


Figure 45: X.509 vs. Version 3 Certificate

However, the ISO/IEC and ANSI standard extensions are very broad in their applicability. In order to develop interoperable implementations of X.509 v3 systems for Internet use, it is necessary to specify a profile for use of the X.509 v3 extensions tailored for the Internet. For example the Internet Public Key Infrastructure (IETF-PKIX) working group has specified a profile for Internet WWW, e-mail, and IP security applications. Environments with additional requirements may build on this profile or may replace it. The extensions supported by SECUDE for example are a subset of the PKIX profile.

| Part | Contained Information | |
|---|---|---|
| Version | Version number; an integer, value is "2" for version 3 | |
| Serial number | Unique identifier for each certificate generated by issuer; integer | |
| Signature | Algorithm identifier | algorithm used to sign certificate |
| | Parameters | any parameter needed |
| Issuer | Name of issuer (X.500 "Distinguished Name", a sequence of RelativeDistinguished-Names that uniquely identify a directory object). | |
| Validity | NotBefore | UTCTime |
| | NotAfter | UTCTime |
| Subject | Name of subject (X.500 "Distinguished Name") | |
| Subject's public key info | Algorithm identifier | subject's signature algorithm |
| | Public key | subject's public key |
| Issuer unique identifier | Contains additional information about the subject; must be version 2 or higher(optional). | |
| Subject unique identifier | Contains additional information about the issuer; must be version 2 or higher(optional). | |
| Extensions | (optional) | |
| Issuer's signature | | |

Figure 46: X.509 Version 3 Certificate

**Extensions in X.509 v3 Certificates**

The basic certificate fields of a version 3 certificate are the same like those of a v2 certificate, with the exception of the extension field. Version 3 introduces a mechanism whereby certificates can be extended, in a standardised and generic fashion, to include additional information ("standard extensions"). However, certificates are not constrained to only the standard extensions and anyone can register an extension with the appropriate authorities (e.g., ISO). Over time, it is expected that new broadly-applicable extensions will be added to the set of standard extensions. It is important to recognise, however, that the extension mechanism itself is completely generic.

The extension field allows addition of new fields to the structure. An extension field consists of an extension identifier (type), a criticality flag and a data value of the extension.

| Type | Criticality | Value |
|---|---|---|

Figure 47: Structure of an Extension X.509 v3

The extensions defined for X.509 v3 certificates provide methods for associating additional attributes with users or public keys, for managing the certification hierarchy, and for managing certificate revocation list distribution. The X.509 v3 certificate format also allows communities to define private extensions to carry information unique to those communities. Each extension in a certificate may be designated as critical or non critical. Use of each extension is at the option of the certification authority issuing a certificate. A certificate using system (an application validating a certificate) must reject the certificate if it encounters a critical extension it does not recognise. A non-critical extension may be ignored if it is not recognised. Extension definitions indicate if they are always critical, always non-critical, or if criticality can be decided by the certificate issuer. For all extensions, there shall be no more than one instance of each extension type in any certificate.

Conforming certification authorities are required to support the Basic Constraints extension, the keyUsage extension and certificatePolicies extension. If the certification authority issues certificates with an empty sequence for the subject field, the certification authority must support the altSubject-Name extension. If the certification authority issues certificates with an empty sequence for the issuer field, the certification authority must support the altIssuerName extension. Support for the remaining extensions is optional. Conforming certification authorities may support extensions that are not identified within this specification; certificate issuers are cautioned that marking such extensions as critical may inhibit interoperability.

At a minimum, applications conforming to this profile shall recognise extensions which shall or may be critical. These extensions are:

- Key usage.
- Certificate policies.

- The alternative subject name.
- Basic constraints.
- Name constraints.
- Policy constraints.

Useful are also the following extensions.

- Key identifiers.
- Certificate revocation list distribution points.
- Authority information access.

Of high interest is the basicConstraints extension, which allows to distinguish between an end user and a certification authority certificate. Since this extension is always flagged critical end users cannot act as certification authority without notice. Moreover the basicConstraints extension may optionally constrain the certification path length through that certification authority.

In environments with a variety of policies the certificatePolicies extension identifies the policy under which the certificate has been issued. If the validation software does not recognise the identified policy, i.e. the policy is not configured to be trusted, the verification will fail, regardless of the criticality flag.

In environments with multiple key pairs additional information about the subject's key may be indicated by the keyUsageRestriction extension. Depending on the type of the Personal Security Environment (PSE) , i.e. whether the PSE belongs to a certification authority or an end user, the key usage is set to the respective default value, when the PSE is created. For end user certificates the key usage bits digitalSignature, NonRepudiation, KeyEncipherment, and dataEncipherment are set. For certification authority certificates KeyCertSign, and cRLSign are additionally set. In case of PSE which is not restricted to hold one key pair only the encryption key certificate limits the use to KeyEncipherment, and dataEncipherment, the signature key certificate to digitalSignature and NonRepudiation. This default value may be changed by the issuing certification authority. If all bits are set to zero, it indicates that the key is intended for some other purposes not listed above. If it is flagged critical the key usage is restricted to that purpose, otherwise it just indicates the intended purpose. It is up to the issuing certification authority to decide whether this extension should be critical or non-critical, but it is recommended that the key usage should always be critical.

In case of multiple key pairs the Authority Key Identifier and the Subject Key Identifier are useful to identify the key that has been used to sign a certificate or used in an application. If no reference to the key has been specified at the time of issuing a certificate the SHA-1 hash of the public key is computed and used as key identifier.

| Name of Extension |
|---|
| **Key and policy information** |
| authorityKeyIdentifier |
| keyIdentifier |
| certIssuer |
| certSerialNumber |
| keyAttributes |
| keyIdentifier |
| keyAtttributes |
| privateKeyValidity |
| certificatePolicies |
| keyUsageRestriction |
| policyMappings |
| **Subject and issuer attributes** |
| subjectAltName |
| issuerAltNAme |
| subjectDirectoryAttributes |
| **Certification path constraints** |
| basicConstraints |
| subjectType |
| pathLengthConstraint |
| subtreesConstraint |
| nameConstraints |
| nameSpaceConstraint |
| nameSubordConstraint |
| policyConstraints |
| policySet |
| requireExplicitPolicy |
| inhibitPolicyMapping |

Figure 48: X.509 v3 Standard Extensions

Standard Extensions

This section identifies standard certificate extensions defined.

Private Extensions

The X.509 v3 certificate format also allows communities to define private extensions to carry information unique to those communities.

**Certificate Revocation**

X.509 defines one method of certificate revocation. This method requires each certification authority periodically issuing a signed data structure called a certificate revocation list (CRL). A certificate revocation list is a time stamped list identifying revoked certificates which is signed by a certification authority and made freely available in a public repository. Each revoked certificate is identified in a certificate revocation list by its certificate serial number. When a certificate-using system uses a certificate (e.g., for verifying a remote user's digital signature), that system not only checks the certificate signature and validity but also acquires a suitably-recent certificate revocation list and checks that the certificate serial number is not on that certificate revocation list. The meaning of suitably-recent may vary with local policy, but it usually means the most recently-issued certificate revocation list. A certification authority issues a new certificate revocation list on a regular periodic basis (e.g., hourly, daily, or weekly). Entries are added to certificate revocation lists as revocations occur, and an entry may be removed when the certificate expiration date is reached.

An advantage of this revocation method is that certificate revocation lists may be distributed by exactly the same means as certificates themselves, namely, via untrusted communications and server systems.

| Part | Contained Information | |
|---|---|---|
| Signature | Specifies the signature algorithm (identifier) and associated hash function used to sign the CRL. | |
| | Any parameters needed. | |
| Issuer | Distinguished Name (X.500 "Distinguished Name" a sequence of RelativeDistinguishedNames that uniquely identify a directory object) of CRL issuer (certification authority) responsible for this CRL. | |
| This update | The date and time when this CRL was issued. UTCTime. Update timestamp. | |
| Next update | The optional date and time by which the issuer will issue the next edition of the CRL. | |
| Revoked certificates | List of revoked certificates. Is a sequence of entries consisting of the serial number of the revoked certificate and the revocation date, when the certificate was revoked. | |
| CRL extensions (optional) | Optional extensions. | If "true" extension must be processed. |
| Zero or more extensions | Extension parameters. | |
| Issuer's signature | | |

Figure 49: X.509 Version 1 Revocation List

One limitation of the certificate revocation list revocation method, using untrusted communications and servers, is that the time granularity of revocation is limited to the certificate revocation list issued period. For example, if a revocation is reported now, that revocation will not be reliably notified to certificate-using systems until the next period certificate revocation list is issued – this may be up to one hour, one day, or one week depending on the frequency that the certification authority issues certificate revocation lists.

Another potential problem with certificate revocation lists is the risk of a certificate revocation list growing to an entirely unacceptable size. In the 1988 and 1993 versions of X.509, the certificate revocation list for the end-user certificate needed to cover the entire population of end-users for one certification authority. It is desirable to allow such populations to be in the range of thousands, or possibly even hundreds of thousands of users. The end-user certificate revocation list is therefore at risk of growing to such sizes, which present major communication and storage overhead problems. With the version 2 certificate revocation list format, introduced along with the v3 certificate format, it becomes possible to arbitrarily divide the population of certificates for one certification authority into a number of partitions, each partition being associated with one certificate revocation list distribution point (e.g., directory entry or URL) from which certificate revocation lists are distributed. Therefore, the maximum certificate revocation list size can be controlled by a certification authority. Separate certificate revocation list distribution points can also exist for different revocation reasons. For example, routine revocations (e.g., name change) may be placed on a different certificate revocation list to revocations resulting from suspected key compromises, and policy may specify that the latter certificate revocation list be updated and issued more frequently than the former.

As with the X.509 v3 certificate format, in order to facilitate interoperable implementations from multiple vendors, the X.509 v2 certificate revocation list format needs to be profiled for use in a European Trust Infrastructure.

| Part | Contained Information | |
|---|---|---|
| Signature | Specifies the signature algorithm (identifier) and associated hash function used to sign the CRL. | |
| | Parameters | Any parameters needed. |
| Issuer | Distinguished Name (X.500 "Distinguished Name" a sequence of RelativeDistinguishedNames that uniquely identify a directory object) of CRL issuer (certification authority) responsible for this CRL. | |
| This update | The date and time when this CRL was issued. UTCTime. Update timestamp. | |
| Next update | The optional date and time by which the issuer will issue the next edition of the CRL. | |
| Revoked certificates | List of revoked certificates. Is a sequence of entries consisting of the serial number of the revoked certificate and the revocation date, when the certificate was revoked. | |
| CRL extensions (optional) | Optional sequence of fields pertaining to the whole CRL. If "true" extension must be processed. | |
| CRL Entry Extensions | Optional sequence of fields pertaining to a specific CRL entry. | |
| Zero or more extensions | Extension parameters. | |
| Issuer's signature | | |

Figure 50: X.509 Version 2 Revocation List

One goal of this X.509 v2 certificate revocation list is to foster the creation of an interoperable and reusable trust infrastructure.

Furthermore, it is recognised that on-line methods of revocation notification may be applicable in some environments as an alternative to the X.509 certificate revocation list. On-line revocation checking eliminates the latency between a revocation report and certificate revocation list the next issue. Once the revocation is reported, any query to the on-line service will correctly reflect the certificate validation impacts of the revocation. However, these methods impose new security requirements; the certificate validator must trust the on-line validation service while the repository did not need to be trusted.

| Part | Contained Information | |
|---|---|---|
| Serial number | Serial number of revoked certificate (unique for the issuer) | |
| Revocation date | UCT Time | |
| CRL entry extensions (optional) | Critical flag | If "true" extension must be processed |
| Zero or more extensions | Extension parameters | |

Figure 51: Certificate Revocation List Extensions

For an automatic retrieval of certificate revocation lists during the verification procedure the certification path constraints extension may be used. These extensions identify the certificate revocation list distribution point or points to which a certificate user should refer to ascertain if the certificate has been revoked.

Certificate Revocation List Extensions

The extensions defined by ANSI X9 and ISO for X.509 v2 certificate revocation lists provide methods for associating additional attributes with certificate revocation lists. The X.509 v2 certificate revocation list format also allows communities to define private extensions to carry information unique to those communities. Each extension in a certificate revocation list may be designated as critical or non-critical. A certificate revocation list validation must fail if it encounters a critical extension which it does not know how to process. However, an unrecognised non-critical extension may be ignored.

Delta Certificate Revocation List Indicator

The delta certificate revocation list indicator is a critical certificate revocation list extension that identifies a delta certificate revocation list. The use of delta certificate revocation lists can significantly improve processing time for applications which store revocation information in a format other than the certificate revocation list structure. This allows changes to be added to the local database while ignoring unchanged information that is already in the local database.

Reason Code

The reason code is a non-critical certificate revocation list entry extension that identifies the reason for the certificate revocation. Certification authorities are strongly encouraged to include reason codes in certificate revocation list entries.

**Operational Protocols**

Operational Protocols are required to deliver certificates and certificate revocation lists to certificate using client systems. Provision is needed for a variety of different means of certificate and certificate revocation list delivery, including request/deliver procedures. These may be based on e-mail, http and WHOIS++. In X.500 the following is standardised.

Directory Protocols

There are four directory protocols:

The Directory Access Protocol (DAP), which defines the exchange of requests and outcomes between a Directory User Agent and a Directory System Agent.

- The Directory System Protocol (DSP), which defines the exchange of requests and outcomes between two Directory System Agents.
- The Directory Information Shadowing Protocol (DISP), which defines the exchange of replication information between two Directory System Agents that have established shadowing agreements.
- The Directory Operational Binding Management Protocol (DOP), which defines the exchange of administrative information between two Directory System Agents to administer operational bindings between them.

Lightweight Directory Access Protocol

The Lightweight Directory Access Protocol was designed to overcome the problems resulting from the requirements of the X.500 Directory Access Protocol (DAP). The goal was to encourage the use of a standardised directory format and a uniform access protocol. While X.500 never really found its way into mainstream computing, LDAP clients and servers are now announced by nearly all of the manufacturers of operating systems and a lot of software companies, including those offering networking products. While queries and answers are still encoded using ASN.1, LDAP makes restrictions on the type and format. Another simplification is the use of string notation in most of the attributes. On the transport side TCP connections are usually used to communicate with an LDAP server, eliminating the need for an OSI protocol stack. This all leads to smaller code and more acceptance on the implementers` side. A slight variation connectionless LDAP (CLDAP) uses UDP packets to communicate with a server without the overhead of a TCP connection set-up. CLDAP only supports querying servers and does not support the alteration of the directory's content.

**Possible Problems of Certificates**

The meaning of the certificate is available only in some policy statement which is on file someplace.

Implicitly the meaning is in the key used to sign the certificate. If a user wants to generate a certificate with a second meaning, he needs a second key of his certification authority.

The identity association itself may be a problem. There may be users who, in a desire to maintain privacy, want to perform anonymous transactions.

The X.509 and X.500 standards currently do not have provisions for the concept of trusted distribution point. A trusted distribution point will require some features to associate and secure the response to particular request.

## 2.8.2 Implementations

### 2.8.2.1 Privacy-Enhanced Mail (PEM)

The Internet Privacy Enhanced Mail proposals, published in 1993 as RFC 1421 - 1424, include specifications for a public key infrastructure based on X.509 v1 certificates.

Privacy Enhanced Mail has been developed to enhance the privacy of e-mail in the Internet by implementing privacy enhancement services to provide confidentiality, authentication, message integrity assurance, and non-repudiation of origin based on symmetric and asymmetric cryptography. The architecture for managing keys that has been selected is based on the use of public key certificates and is compatible with the authentication framework described in X.509.

Public key certificates are the central elements of the key management architecture of PEM. A public key certificate in PEM is a data structure which contains a representation of its subject's identity ("subject name"), the public key of that subject which has to be bound to the subject's name, and the representation of the identity of the issuer („issuer name") who guarantees for the binding between the name of the subject and the public key contained in the certificate by cryptographically signing the certificate with his private key. Both, the name of the subject and the name of the issuer in the certificate are expressed as Distinguished Names as defined in the X.500 directory system concept.

In order to ensure the uniqueness of DNs (no two entities may have the same DN) and to minimise the complexity of validating user certificates, naming conventions have been introduced which indicate explicitly the relationship between a certificate issuer and the subject (user) via the naming hierarchy. The so-called „DN subordination rule“ requires that CAs in general are expected to sign certificates only if the subject DN in the certificate is subordinate to the DN of the issuer CA. This ensures that certificates issued by a CA are syntactically constrained to refer only to a set of subordinate entities in the X.500 Directory Information Tree which it can certify.

For example, if an organisation CEC acts as a CA with a DN

C=BE,L=BRUSSELS,O=CEC

then the subject DN in the certificate of a user John Miller who has been given a certificate from this CA should have the form

C=BE,L=BRUSSELS,O=CEC,OU=DG13,CN=John Miller.

This approach, although not required by X.509, not only supports that systems using certificates can automatically verify that a particular CA (below the PCA level) is authorised to sign a certificate for a given subject, it improves also the confidence that the certificate has been granted under the policy which has been specified by the Policy Certification Authority.

Certification Path

There are different ways in which certification authorities might be configured in order for public key users to be able to find certification paths. RFC 1422 defines a rigid hierarchical structure of certification authorities. There are three types of certification authorities.

Internet Policy Registration Authority (IPRA)

This authority operated under the auspices of the Internet society, acts as the root of the PEM certification hierarchy at level 1. It issues certificates only for the next level of authorities, PCAs. All certification paths start with the IPRA.

Policy Certification Authorities (PCAs)

PCAs are at level 2 of the hierarchy, each PCA being certified by the IPRA. A PCA must establish and publish a statement of its policy with respect to certifying users or subordinate certification authorities. Distinct PCAs aim to satisfy different user needs. For example, one PCA (an organisational PCA) might support the general e-mail needs of commercial organisations, and another PCA (a high assurance PCA) might have a more stringent policy designed for satisfying legally binding signature requirements.

Certification Authorities (CAs)

CAs are at level 3 of the hierarchy and can also be at lower levels. Those at level 3 are certified by PCAs. CAs represent, for example, particular organisational units (e.g., departments, groups, sections), or particular geographical areas.

The PEM certificate is compared with the X.509 certificate in the following figure.

| CCITT X.509 | PEM |
|---|---|
| Version | |
| Serial number | |
| Signature | Algorithm identifier |
| | Parameters |
| Issuer | |
| Validity | NotBefore |
| | NotAfter |
| Subject | |
| Subject's public key information | Algorithm identifier |
| | Subject's public key parameters |
| Issuer's signature | |

Figure 52: Comparing CCITT X.509 and PEM Certificate Format

#### 2.8.2.2 Pretty Good Privacy (PGP)

PGP is a high security cryptographic software application for a multitude of computer platforms.

It allows for a secure exchange of messages or files by using the public key of the receiver for encrypting the message or the file. An encrypted information exchange requires that the public key of the receiver is made available to the sender by some means. The assessment of the trust which can be put into that public key, once available, is the responsibility of the individual user, the trust model used by PGP assumes that only the individual user has the competence to decide whom to trust.

Public keys are kept in individual „key certificates“ which are generated by the user and which include his user ID, a timestamp which denotes the time of generation of the public/secret key pair, and the actual key material. The user ID consists of the common name of the user and normally of an e-mail address where the e-mail address is used to make the user ID unique. If no e-mail address is available, any other information can be used, which makes the user ID unique.

Example:

John Miller <100000.1111@compuserve.com>

The binding between the public key and the user ID is certified by the user himself and he is responsible for distributing his certificate to the others.

PGP has no official certifying authorities that sign public keys of individuals.

PGP key signatures have a disadvantage compared to X.509 certificates in that the person signing the key is not necessarily either known or trusted. PGP attempts to compensate for that by allowing multiple, presumably independent signatures to vote a binding into validity.

## 2.8.3 Proposals

### 2.8.3.1 Internet Public Key Infrastructure (IPKI)

The PKIX working group of the IETF is charged with the development of Internet standards needed to support a Public Key Infrastructure based on the X.509 authentication framework. The goal of this infrastructure is to facilitate the adoption/use of X.509 certificates in multiple applications which make use of the Internet and to promote interoperability between different implementations choosing to make use of X.509 certificates. The resulting infrastructure is intended to provide a framework which will support a range of trust/hierarchy environments and a range of usage environments. Candidate applications to be served include, but are not limited to, e-mail, Internet payment protocols, and WWW protocols.

It is envisioned that the resulting standard will consist of four parts.

Part I: X.509 Certificate and CRL Profile.
Part II: Operational Protocols.
Part III: Certificate Management Protocols.
Part IV: Certificate Policy and Certificate Practices Framework.

Drafts of all four documents are currently being discussed by the Internet community. An additional document „Architecture for Public Key Infrastructure“, which serves as a requirement paper for the work on a public key infrastructure has also been produced by the working group.

Basis for the deliberations of the working group is the X.509 v3 certificate, as standardised in June 1996 by ISO/IEC and its standard extensions. Because these extensions are very broad in their applicability it is considered necessary to specify a profile for use of the X.509 v3 extensions tailored to the Internet if the development of interoperable implementations of X.509 v3 systems for Internet use should become possible.

The attributes which will be used in the profile certificate are determined in the first instance by the goal of the IPKI, to provide automated identification, authentication, access control, and authorisation functions. In addition, attributes for management will also be necessary.

Because of the shortcomings of PEM discussed above, it is proposed that more flexible certification authority structures than the PEM hierarchy be supported by the IPKI specifications. In fact, the main reason for the structural restrictions imposed by PEM was the restricted certificate format provided with X.509 v1. With X.509 v3, most of the requirements addressed by PEM can be fulfilled by using certificate extensions, without a need to restrict the certification authority structures used in PEM. In particular, the certificate extensions relating to certificate policies obviate the need for policy certification authorities and the constraint extensions obviate the need for the name subordination rule.

| Name of Extension | Comments |
|---|---|
| Subject Information Access | Recommended, non-critical |
| Authority Information Access | Recommended, non-critical and critical |
| CA Information Access | Recommended, non-critical and critical |

Figure 53: PKIX Internet Certificate Extensions

There are two entities in the certificate which require to be named, the end-entity and the certification authority. The end-entity is the entity to be named in the subject field of a certificate, e.g. a user or an end user system (e-mail software, WWW browser etc.) and the certification authority is the entity to be named in the issuer field of a certificate.

The convention for IPKI is as follows:

issuer name: Identifies the entity who has signed and issued the certificate. The issuer name is expressed by a Distinguished Name

subject name: Identifies the entity associated with the public key listed in the subject public key field of the certificate. The subject name is expressed by a Distinguished Name.

The IPKI proposal does not assume the deployment of an X.500 directory system but other means of distributing certificates and certificate revocation lists are also supported. If end user systems are used which require alternative name forms like an Internet e-mail address (RFC 822 name), a DNS name, an

IP address, a X.400 address, an EDI party name or an uniform resource identifier and the need arises to bind additional identities to the subject of a certificate, a standard extension will be used which allows to enter such alternative names into the certificate.

A name constraints extension in the extension part of the certificate provides the definition of permitted and excluded subtrees that place restrictions on names that may be included within a certificate issued by a given CA. Restrictions may apply to the subject Distinguished Name or to subject alternative names. This feature allows to overcome the undesirable restrictions that have been the result of the name subordination rule used in PEM.

#### 2.8.3.2 The Federal Public Key Infrastructure (FPKI)

The goal of the FPKI project under the leadership from NIST is to work with industry and U.S. government to design a public key infrastructure based on standards and interoperable commercial-off-the-shelf products which support digital signatures and other public key enabled security services. It will be used for sensitive-but-unclassified communications inside the US Federal Government and with outside organisations and citizens. It also will be interoperable with other public key infrastructures.

The work until now has resulted in the publication of initial versions of a requirements document, a concept of operations, a technical security policy, a X.509 v3 certificate profile with ISO standard extensions, and an interoperability report. In addition a Minimum Interoperability Specification for PKI Components (MISPC) has been developed and published.

Although it is recognised that the resulting infrastructure will eventually have to support certificates for confidentiality key management, that is outside the scope of the current work specification. It is anticipated that a sound digital signature public key infrastructure should provide the basic foundation needed for issuing any kind of public key certificate.

Basis for the FPKI certificate will be the X.509 v3 certificate format. The management of certificates and keys will depend significantly on the availability of unique names for the digital signature entities; the certification authorities, and for the operations personnel involved in the operation of certificate management nodes. X.500 Distinguished Names have been chosen for that purpose. The federal PKI public key certificate will support three basic classes of certificates:

- Public key certificates to support signature applications.
- Key management certificates to support key exchange for confidentiality purposes.
- Attribute certificates which specify user privileges or role or indicate federal PKI accreditation of third party services.

FPKI certificates will contain the following name related fields according to the X.509 v3 standard which relate to names:

- The issuer name and the subject name in the certificate have to be globally unique Distinguished Names.
- The subjectUniqueID and issuerUnique ID fields will not be used by FPKI.
- The nameConstraint field, which is used in certificates from a certification authority to limit the name space for which certificates are issued, will be supported.

The issuer name in the certificate revocation list will be expressed in the same format as in the certificate., i.e. as Distinguished Name

Enforced Policies

The PKI must enforce appropriate policies. These policies must be sufficient for legally binding signatures, so that parallel paper trails with physical signatures are not required. Four criteria have been proposed for digital signature applications:

- The private key (s) must be under the sole control of the user.
- The key pair must be unique to the user.
- It must be capable of being verified.
- It must be linked to the data in such a manner that if the data is changed the signature is invalidated upon verification.

The Federal PKI must support these criteria.

The PKI must provide users with the capability to create and post certificates, to revoke certificates, and to obtain, interpret, and verify certificates. These are the basic functions of a certificate management system. Note that the latter capability implies that the users can generate and verify digital signatures, and can obtain the current status of a certificate.

Such a PKI can enable users to:

- Verify the identity of a message sender based upon the senders` public key certificate.
- Verify message integrity and detect message alteration.
- Exchange keys for confidentiality.
- Authenticate identity based on a public key certificate.
- Specify subject attributes, such as security clearance, privileges, etc.

These services can be enabled by the implementation of a certificate management system, but are not required to implement such a system. The federal PKI must not prevent such services, but does not support them directly.

Certificates

All signature and key exchange certificates will contain, at a minimum:

- The Distinguished Name of the user or entity.
- The validity period.
- The public key and its parameters.
- Key usage field.
- Algorithm field.
- One or more policy identifiers.
- The name of the issuing authority.
- The issuing authority's DSA signature.

A government wide naming scheme, probably based on X.520, is required to ensure assignment of unique names. The maximum validity periods will be specified in the technical security policy. The public key parameters must be included with the public key to ensure interoperability with other PKIs. The key usage field will differentiate between signature and key management certificates. The algorithm filed will support multiple signature and public key encryption algorithms as new algorithms are approved for federal use as well as ranges of key sizes for immediate use with the digital signature. A user certificate will also include one or more policy identifier fields indicating the security policies governing its issuance. Certification authority certificates may include policy constraint fields indicating the set of policies for which it may issue certificates.

The certificate may also contain a non-negative integer field specifying the number of electronic renewals permitted for this certificate.[3]

Certificates issued to certification authorities may also contain the following fields:

- SubtreesConstraint field.
- NameConstraint field.
- Policy restrictions.
- Policy mappings.

Attribute certificates will contain, at a minimum:

- The Distinguished Name of the user or entity.
- The validity period.
- The attribute field.
- The name of the issuing authority.
- The issuing authority's DSA signature.

Attribute formats and the procedures for creation, distribution, and revocation of attribute certificates will reflect local requirements and will not be determined by the federal PKI.

Cross-Certificate Pairs

The federal PKI will support the X.509 cross certificate pair construct to build bi-directional chains of trust between certification authorities. The cross certificate pair construct contains two certificates:

A "forward" certificate and a "reverse" certificate. The subject of the forward certificate is the issuer of the reverse certificate and vice versa.

For the certificates defining the hierarchical architecture of the federal PKI, the forward certificate will define the path leading from the root. Only the forward certificate is required in this case; the reverse certificate may not be included in some of these cross certificate pairs.

Cross certificate pairs may also define new paths that do not parallel the hierarchy leading to the root. In this case, there is no semantic difference between the forward and reverse certificates. Constraints for cross certificate pairs that do not parallel the hierarchy have to be specified.

Certification authorities generate cross certificate pairs, containing at a minimum the forward certificate, to represent the hierarchy leading from the root. Optionally, certification authorities may be able to generate reverse certificates to provide "full" cross certificate pairs. The root certification authority must be able to generate "full" cross certificate pairs to support mutual recognition with non-federal PKIs.

Create and Post Certificates

The creating and posting of certificates is an infrastructure service that is required by the user. This service permits a user to:

- Obtain a public key pair.[4]
- Obtain the corresponding public key certificate.
- Have the certificate bound to their identity with a specified level of assurance.
- Make the certificate available to any user who wishes to obtain it.

Binding the certificate to its owners identity requires personal presentation of credentials to a federal PKI representative. The type and number of credentials required may vary; this should be reflected in

[3] The number of electronic renewals permitted is decremented by one upon each renewal, and when it reaches zero, the certificate must be re-issued through the initial certification procedure.

[4] In most cases, users should generate their own public key pairs. This provides the strongest binding for non-repudiation purposes. In special cases, the PKI may generate key pairs for users.

the certificate[5]. For convenience, initial registration should be a local service. Subsequent generation of certificates should be performed remotely, in an automated fashion.

Obtain, Interpret, Verify and Revoke Certificates

Users must be able to determine if they can trust a signature for a particular application. This requires verification of certificates and certificate chains, and evaluating the assurance associated with the binding of identity to a certificate.

Users must be able to obtain certificates for a particular user in an automated fashion, parse it correctly, and determine the current status of the certificate. This process must be entirely automated. The certificate format must be known, so that it can be interpreted in an automated fashion.

The certificates must be signed by an appropriate algorithm so that users can verify the certificate chain and verify the binding of a user and certificate. To verify the certificate chains, the user must be able to verify a series of digitally signed messages (certificates). This requires that:

- The keys are associated with an appropriate algorithm.
- The messages are hashed with an appropriate algorithm.
- The message formats are known to the receiver.

Finally the user must be able to determine if the policy under which a certificate was issued is acceptable for a particular application.

Management Requirements for a Federal PKI

The Federal PKI must meet both technical and policy-oriented management requirements. The infrastructure must be:

- Scalable and manageable.
- Issue certificates.
- Manage certificates.

The infrastructure must also:

- Ensure interoperability within the government.
- Set policy on certificates and their use.
- Negotiate and set policy for interoperation with users of non-federal PKIs.

An acceptable federal PKI will meet the technical requirements and allow management to set and enforce appropriate policies.

Issue certificates

The federal PKI will issue signature and key exchange certificates to properly identified federal users at their organisations` request. Renewal of certificates will be performed electronically where assurance requirements permit.[6]

Keys and algorithms

The federal PKI will only issue certificates signed with strong keys and algorithms. Where certificates contain encryption and signature keys, the algorithm and key length must be sufficient that it is computationally infeasible to:

- Forge signatures.
- Make undetected alterations to signed files.
- Obtain confidentiality keys exchanges through supported key distribution mechanisms during the expected lifetime of the information.

The federal PKI will not generate signature keys for PKI users, although it may provide the equipment. Each agency will determine if the PKI will generate key management keys for their PKI users.

Attributes

The federal PKI will manage attribute certificates for owners of PKI signature certificates at their organisation's request. Standard attributes will be defined to enhance interoperability, but locally defined attributes will be permitted to maximise utility of the infrastructure.

Management of Certificates

Certificate Distribution

At a minimum, the federal PKI will support distribution of certificates through directory services. The directory server(s) will chain to the directory servers of other trust infrastructures, as approved by the PAA. Additional distribution services may be offered to meet local requirements.

Revocation

Users must be able to revoke the binding between their identity and a public key when the private key

[5] Sometimes there are specified three levels of assurance reflecting the type and number of credentials. See 2.3.7 Policy and Licensing Authorities (credentials).

[6] The assurance associated with the binding for electronically renewed certificates is lower than for certificates issued by an Organisational Registration Authority. The assurance falls with each electronic renewal.

has been lost or compromised, or the status of the user changes.[7] Certificate revocation information and compromised key information also must be available to ensure compromised certificates and keys are not exploited. This information must be available in a timely fashion. The federal PKI compromised key lists will support rapid dissemination of key compromises throughout the infrastructure.

The federal PKI will provide one or more mechanisms to verify the validity of certificates. CRLs will be available from on-line directories. Unauthorised generation or modification of CRLs must be prevented by employing digital signatures. CRLs must be posted at regular intervals so that users can ensure that they have the most current information. Optionally, current CRLs may be available from the certification authorities themselves. The federal PKI must also support push-down mechanisms for key compromise and other critical events for communities with special security requirements. The federal PKI will issue certificate revocation lists conforming to the X.509 version 2 certificate revocation list format. The federal PKI will use optional features of the format, such as delta certificate revocation lists or distribution points, to control communication costs for certificate revocation lists. The format will support a modular presentation of CRLs, and the knowledge of why certificates were revoked (reason).

| Extension | Used for |
|---|---|
| CRL extensions | |
| cRLNumber | Used to provide a CRL sequence number |
| CRL entry extensions | |
| cRLReason | Identifies reason for CRL entry, e.g. key compromise, affiliation change, ... |
| expirationDate | Used if certificates are held, applies to expiration of hold |
| instructionCode | Identifier to indicate action taken on encountering a held certificate |
| invalidity Date | Date of known or suspected compromise |

Figure 54: Certificate Revocation List Extension Fields

The federal PKI may also include a trusted directory mechanism for real-time certificate validation. Such a mechanism would provide the most current information to critical applications. Trusted directories could be maintained by the PKI, or may be offered by vendors as value-added services.

Archiving

The federal PKI will maintain a permanent archive of certificates so that business and legal requirements for electronic commerce can be met. The certification authority can maintain its own archive, or use of an approved third party server. At a minimum, all certificates issued to certification authorities and all CRLs must be archived. It is unclear if a record of all certificates issued or required to meet business and legal requirements.

Archives must be maintained, even when a certification authority or archive server is disbanded. These archives will be maintained by the federal PKI.

Certification authorities will only issue certificates to users at the request of their organisations. A certification authority may optionally confirm that the organisation has requested a certificate for this user. The certification authority must delegate this function to its ORAs if it does not perform it directly.

Directory Servers

Federal PKI directory servers will support the X.500 directory service, including both the DAP and LDAP access protocols. The directory servers will chain to directories maintained by cross-certified PKIs. The directory will make all published certificates available upon request. The directory will include all current certificates, cross-certificate pairs and certificate revocation lists issued by the federal PKI certification authorities. Certificates and cross certificate pairs must be maintained in the directory for some time period after expiration or revocation to facilitate verification of documents signed just before the certificate was revoked or expired.

Optionally, certification authorities may maintain a database of information regarding owners of certificates issued. This information would include contact information, such as phone numbers and e-mail addresses, and may include private user information such as social security numbers and their mothers maiden name. This would facilitate recovery from certification authority compromise and remote user revocation requests, but would also impose additional security requirements from the privacy act.

#### 2.8.3.3 PASSWORD

PASSWORD was a project funded by the EC VALUE programme 1992 to create a security infrastructure based on X.509 technology for the research community in Europe and to pilot secured applications with users.

An important aspect of the project was the use of the existing X.500 infrastructure to register certification authorities and public keys. Part of the project plan was to create a suite of secured applications from each consortia to be used in conjunction with the toolkits. The project successfully demonstrated

[7] For example, the identity stated in the certificate may imply organisational relationships. The relationships change (e.g., job change or retirement) and the certificate becomes invalid.

interoperability between these heterogeneous secured applications which included X.400, X.500, PEM and ODA

The project PASSWORD is a predecessor of the project ICE-TEL.

#### 2.8.3.4 Interworking Public Key Certification Infrastructure (ICE-TEL)

ICE-TEL is a project funded by the European Commission under the Telematics programme. The aim of the project is to provide a large scale public key certification infrastructure in a number of European countries for the use of public key based security services. It will offer solutions to the problem of security on the Internet as used by industrial and academic research. The project will provide all techno-logy components which allow the deployment of user tools and applications with a common integrated public key security technology. Interoperability with approaches that are pursued in the standardisation work for the Internet and by the World Wide Web Consortium will be achieved through participating in that work and by adopting the evolving standards.

The resulting certification infrastructure will provide certification services to educational, research, business, and government communities all over Europe. In the first phase ICE-TEL starts out by using X.509 v1 certificate and certificate revocation list formats and it enforces the trust model developed for Privacy Enhanced Mail (PEM). In the second phase of the project X.509 v3 certificates and X.509 v2 certificate revocation lists will be used.

Naming in both phases will be based on X.500 Distinguished Names. Uniqueness of names is an essential requirement for the envisioned certification infrastructure for certification authority as well as for end-users. Name spaces are specified by policy certification authorities and they have to assure that the Distinguished Names of the certification authorities, which are in their domain, are unique.

Each certification authority in return guarantees the uniqueness of names of its end-users. According to X.509 v1 all certificates below a certification authority have to contain a unique Distinguished Name as subject, which has to be subordinate to the issuers Distinguished Name (PEM subordination rule). After switching over to X.509 v3 this constraint can be omitted and the name related fields in the X.509 v3 certificate and the X.509 v2 certificate revocation list, as described in paragraph 2.4.0.3 Revocation Process, will be used with the exception of the subjectUniqueID and issuerUniqueID fields.

#### 2.8.3.5 Secure Electronic Transaction System (SET)

The Secure Electronic Transaction (SET) protocol has been developed as a method to secure bankcard transactions over open networks.

SET uses certificates according to X.509 v3 for different user communities, like:

- cardholder certificates,
- merchant certificates,
- payment gateway certificates,
- acquirer certificates,
- issuer certificates.

The naming convention follows the definitions in X.520.

The SET certificate binds an account number to a signature key. Any card-holder name which might also be in the certificate is of no importance in the process of deciding whether to honour a given purchase request.

| | |
|---|---|
| **Cardholder** | CountryName = <Country of Issuing Financial Institution><br>OrganisationName=<BrandID><br>OrganisationalUnitName=<Name of issuing Financial Institution><br>OrganisationalUnitName=<Optional - Promotional CardName><br>CommonName=<Unique Cardholder ID>[1] |
| **Merchant** | CountryName=<Country of Acquiring Financial Institution><br>OrganisationName=<BrandID><br>OrganisationalUnitName=<Name of Acquiring Financial Institution><br>OrganisationalUnitName=<Name of Merchant as printed on Cardholder statement><br>CommonName=<Unique Merchant ID> |
| **Payment gateway** | country Name=<Country of Acquiring Financial Institution><br>organisationName=<BrandID><br>organisationalUnitName=<Name of Acquiring Financial Institution><br>commonName=<Unique Payment Gateway ID> |
| **RootCertification Authority** | countryName=<Country where CA is located><br>organisationName=<Name of organisation responsible for the Root Certificate><br>commonName=>Unique CA ID> |

[1] Hashed Account Number

Figure 55: Naming Conventions defined for SET

SET certificates are verified through a hierarchy of trust. Each certificate is linked to the signature certificate of the entity that digitally signed it. By following the trust tree to a known trusted party, one can be assured that the certificate is valid. The root key will be distributed in a self-signed certificate.

#### 2.8.3.6 Simple Distributed Security Infrastructure (SDSI)

SDSI is a recent proposal supported by the US Defense Advance Research Agency (DARPA) for "a new distributed security infrastructure which combines a simple public key infrastructure design with a means of defining groups and issuing group-membership certificates". It has to be considered as „work in progress“. Especially a merger between SDSI and SPKI (described in paragraph 2.8.3.7 Simple Public Key Infrastructure) which is currently being discussed might lead to modifications.

The SDSI activities were motivated by the perception that the existing protocols for public-key infrastructures (such as X.509 based schemes that require global certificate hierarchies) are excessively complex as well as incomplete. One of the reasons for the complexity is the dependency on global name spaces. Therefore the proposed design emphasises linked local name spaces rather than a hierarchical global name space as for example required by PEM and it allows to define access-control lists and security policies in a simple and clear terminology. Issues regarding liability and/or legal interpretation of certificates are not addressed in the proposal.

Active entities in SDSI are called principals. A principal is a key, specifically the private key that can sign statements and that is identified by its corresponding public key. Because of the strong binding between a private key and its associated public key, the individual that possesses the private key will actually be in control of that key, however the principal (the public key) "speaks" for him.

A principal can represent an individual or a group and both can be given names. Each principal can create its own local name space by assigning arbitrary local names to other principals which he then can use to refer to them. The reception of another individuals public key and associating it with a local name is consideredd to be such an important activity that it has to be a manual process.

The entity defining a local name is responsible for issuing certificates acknowledging the binding of a principal (public key) key to the name or declaring that a principal is a member of the named group.

There is no requirement for principals to have an unique name from a fixed global name space, because all names are local, therefore different principals might use the same name differently.

End-users deal with names and not with keys therefore they need to have a user friendly interface that allows them to choose their own local names for principals. There is no specified syntax, aside from some standard conventions about the usage of characters. Local names can be chosen arbitrarily for example:

passportno-123456789

sweety

charly@<hostname>.<domainname>.de

Linking of name spaces allows principals to use definitions another principal has made. To accomplish this, each principal can export his bindings of a local name to a principal (name/value bindings) to other principals by issuing signed name/value certificates.

Example:

A name/value certificate that has the following parts that are relevant to naming

(issuer (ref <my-key> „John Miller“))

(subject <John´s-key>)

defines <John´s-key> as the value of the name „John Miller“ in my key´s name space.

The following examples for explaining this process in a different notation and in more detail are taken from the original paper of Rivest and Lampson:

If a local name BOB refers to some principal (e.g. <my-key>), then one can refer to the principal that BOB calls ALICE as

( ref: BOB ALICE )

This can be stated in colloquial terms as

BOB's ALICE ("the principal whom BOB has given the name ALICE")

BOB can export his binding by issuing a signed name/value certificate, which binds the local name ALICE to that particular principal. The certificate can be stored on BOB´s server and distributed on demand.

One can also have longer references, such as

( ref: BOB ALICE MOTHER)

which means

BOB´s ALICE´s MOTHER

This reference is well-defined because the name BOB is bound to some principal in a local name space, who in turn has bound ALICE to some second principal, who in turn has bound MOTHER to some third principal.

Although SDSI favours local name spaces, the need for interoperation with standard roots and global name spaces is recognised and covered by introducing reserved names (indicated by double exclamation marks at the end) which will be universally recognized, like for example:

USPS!!

DNS!!

These reserved names are bound to the same principal in every name space. This binding has to be arranged with suitable procedures (appropriate publicity, manual installation, cross-checking, etc.). The following expression gives SDSI access to the USPS standard name space:

( ref: USPS!! USA DOD DCI)

Such an expression should be viewed as a global name that evaluates to the same principal in every name space, since the first name (e.g. USPS!!) always evaluates to the same principal, and all of the subsequent names are relative.

In order to be able to handle DNS names (Internet e-mail) there is a special treatment for these names, so that

Bob.Miller@penguin.microsoft.com

is equivalent to:

( ref: DNS!! com microsoft penguin Bob.Miller ) .

Implementations of SDSI Version 1.0 are currently running at MICROSOFT and MIT.

#### 2.8.3.7 Simple Public Key Infrastructure (SPKI)

Goal

Many Internet protocols and applications employ public key technology for security purposes and therefore require an infrastructure to manage public keys and the related certificates. The SPKI working group of the Internet Engineering Task Force (IETF) is tasked with the development of standards „for an IETF sponsored public key certificate format, associated signature and other formats, and key acquisition protocols“. The key certificate format and the associated protocols to be developed should be simple to understand, implement, and use. They should contain the minimum information necessary to accomplish authentication and authorisation of access control and confidentiality for the Internet.

The SPKI is intended to provide mechanisms to support security in a wide range of internet applications, including IP security protocols, encrypted e-mail and WWW documents, payment protocols, and any other application which will require the use of public key certificates and the ability to access them. It is also intended that the Simple Public Key Infrastructure will support a range of trust models.

The currently available draft on the Simple Public Key Infrastructure (dated 25 March 1997) is "concerned with certificate and signature formats, using the certificates defined here both for verification of

identity and for proof of authorisation. Other elements of a full Public Key Infrastructure are not covered" and legal and liability considerations are intentionally not treated.

The draft has to be considered as "work in progress" and a merger with SDSI is currently in discussion. The publication of a SPKI Request for Comment is planned for May 1997.

Approach

The Internet changed the world from the one in which identity certificates were considered necessary. As its use increases, there is increasingly interaction with people or companies with whom there exists no prior relationship in the physical world. A transfer of relationship from the physical world to the digital world is meaningless in such cases. Instead, it is necessary to establish relationships directly in the digital world through an instrument, which assigns attributes (authority, permission, to the digital principal. Such an instrument is called an authorisation certificate.

In the literature, the word "certificate" has been generally taken to mean "identity certificate" - a signed statement which binds a key to the name of an individual and has the intended meaning of delegating authority from that named individual to the public key (see, for example, PEM.). This process is designed to transfer a relationship between two entities from the physical world into the digital world.

There are two major areas where the SPKI approach differs from the X.509 concept:

- Use of authorisation certificates,
- Use of public keys as names

Certificates

In SPKI the concept of „identity certificates“ is considered not to be sufficient from an operational point of view, because the transfer of authority in such a certificate is too unspecific. Instead „authorisation certificates“ are introduced which grant a specific authority (authorise some action, give permission, grant some capability) to a public key without binding an "identity" (such as a person's name) to that key. (There is an indirect but strong binding between the public key and the respective identity (key holder) because the private key that is associated with the public key is in the possession of that identity).

Authorisation certificate

SPKI does not want to establish identity with certificates. A process using public key authentication (telnet, ftp, an electronic commerce server, etc.) and verifying a public key certificate in the process, needs to check that the indicated key (i.e. the key holder) has authority to access the controlled data or to execute the requested function: The SPKI certificate is an authorisation certificate which grants a specific authority to a public key rather than binding an "identity" (such as a person's name) to that key. For example, one SPKI certificate might grant permission for a given public key to authenticate logins over telnet as user on a host for a period of time.

A SPKI certificate can be used instead of an access control list: The receiver checks the certificate that contains all the access rights. For completeness however, SPKI certificates can be defined granting all authority of an indicated person to a key and are - in this case - identity certificates.

Contents and Structure of the SPKI Certificate

In the following tables the contents and structure of the SPKI certificate is shown.

| Name | Contents and Functions |
|---|---|
| Issuer | The public key of the issuing party or something reducible to that public key, both as a means for verifying the certificate signature and as a name for the issuing principal. |
| Subject | The object receiving authority via this certificate or something reducible to that object (usually a key or the hash of a key or a document). |
| Modifiers | Modifiers on the certificate. At present the only one defined is "May-delegate:" - the permission to delegate the authority presented in the certificate (or part of it) to some other subject. |
| Authorisation | The specific authorisation(s) being delegated in this certificate.<br>It is possible to include a whole document like a policy in this field. |
| Validity | At least an expiration date but possibly a more complex procedure for determining certificate validity. |

Figure 56: SPKI Certificate Conceptual 5 Fields

| Name | Contents and Functions |
|---|---|
| Issuer | The public key of the issuing party or something reducible to that public key or the hash of the issuer's public key, both as a means for verifying the certificate signature and as a name for the issuing principal. |
| Issuer location | Location of the issuer public key and any certificate needed to establish the authority to do the delegation represented by this certificate. Optional. |
| Subject | The subject receiving authority via this certificate or something reducible to that object (usually a key or the hash of a key or a document). |
| Subject location | The field gives the location of the subject public key and a self-signed validity certificate for that key. Such a location may be an URL or domain name (for look-up in the DNS database. Optional. |
| Authority | The specific authorisation(s) being delegated in this certificate. |
| Modifiers | Modifiers on the certificate. At present the only one defined is "May-delegate:" - the permission to delegate the authority presented in the certificate (or part of it) to some other subject. |
| Validity | At least an expiration date but possibly a more complex procedure for determining certificate validity. Optional. If there is no validity field the certificate is valid forever. |
| Signature | Overall signature by the issuer.<br>In the case of dual signature, the certificate must also be signed by the subject in order to be valid. |

Figure 57: The Fields of an SPKI Certificate

| Name | Contents and Functions |
|---|---|
| Certificate version | Optional. |
| Issuer | The public key of the issuing party or something reducible to that public key, both as a means for verifying the certificate signature and as a name for the issuing principal. |
| Issuer location | Optional. |
| Subject | The object receiving authority via this certificate or something reducible to that object (usually a key or the hash of a key or a document). |
| Subject location | Optional. |
| Delegation | Optional. |
| Authorisation | The specific authorisation(s) being delegated in this certificate. |
| Comments | ... |
| Validity | At least an expiration date but possibly a more complex procedure for determining certificate validity. |
| Modifiers | Modifiers on the certificate. At present the only one defined is "May-delegate:" - the permission to delegate the authority presented in the certificate (or part of it) to some other subject. |
| Signature | Overall signature by the issuer. |

Figure 58: SPKI entire Certificate Structure

### Authorisation Field

When one grants some authority, it might be desired also to grant the permission to delegate that authority to others. Each authorisation field is assumed to have a modifier "delegation", specifying the subject's permission to delegate the authorisation to other keys; the delegated access rights may be "all". There are other fields possible in the authorisation field like ftp, telnet, http, employee, known-to-me-as, comment.

If someone delegates authority, it may be useful for the delegator to get a notification of receipt of the

certificate with the delegated authority. Otherwise it might not be possible to decide who used the authority.

From the perspective of naming there are two basic differences from the X.509-approach. Firstly the view that a centralised proof of identity is useful, is not maintained and secondly the concept of „identity certificates“, which bind a public key to a named individual whose identity ideally is globally unique is considered not to be feasible. Establishing identity is not the central problem in SPKI.

Instead „authorisation certificates“ are introduced which grant a specific authority to a public key without binding an "identity" (such as a person's name) to that key. (There is an indirect binding between the public key and the respective identity because the private key which is associated with the public key is in the possession of that identity). If required, a mechanism is available which allows to certify identity with a SPKI certificate and which supports the SDSI naming mechanism.

The issuer of an authorisation certificate as well as the subject who receives a specific authority via the certificate are identified in the certificate by their public keys respective by a secure hash of those keys.

#### 2.8.3.8 Public Key Cryptography Standards (PKCS)

The family of PKCS standards comprises standards for RSA encryption. The PKCS #10 standard describes a syntax for certification requests. A certification request consists of a Distinguished Name, a public key, and optionally a set of attributes, collectively signed by the entity requesting certification. Certification requests are sent to a certification authority, who transforms the request to an X.509 public key certificate.

PKCS #6 defines a standard syntax for public key certificates based on X.509.

#### 2.8.3.9 Royal Holloway Scheme

An architecture is proposed that will enable TTPs to perform the dual role of providing users with key management services and providing law enforcement agencies with warranted access to a particular user's communications. The mechanism allows users to update their keys according to their own internal security policies. The TTP is an on-line TTP. The proposed mechanism is based on the Diffie-Hellman algorithm for key exchange. X.509 certificates are used.

### 2.8.4 Products

In this chapter are several products mentioned using different approaches.

**TIS/PEM**

Trusted Information Systems (TIS) implemented PEM (TIS/PEM) on various platforms (UNIX VMS, DOS, Windows). The TIS/PEM specifications allow multiple certifications hierarchies. Different certificates can be used.

**UniCERT™**

UniCERT™ is a product of Baltimore Technologies facilitating authenticating of parties for electronic transactions by establishing a secure certified relationship between issued public keys and the owners of these keys. UniCERT generates, registrates, delivers, certifies and revokes unique RSA keys to the PKCS #6, X.509 v1 and v3 standard and RFC 1422 and also maintains a database of valid and revoked certificates. It supports SSL, S/MIME and SET.

**Entrust NORTEL Manager**

This system establishes a public key infrastructure with key management and generating, recovering, hierarchical and cross-certification, revocation, using X.509 v3 for certificates and X.500 for directories.

**S/MIME**

The S/MIME Message specification combines the security enhancement of PEM and the multi-purpose content-types of MIME.

**SECUDE**

SECUDE is a security toolkit for open networks. It provides a variety of well known and intensively studied crypto-algorithms as security basis for higher level applications.

SECUDE supports X.509 v1 certificates, while the next release will contain the X.509 v3 certificate. Both local PSE-located certificates and directory-located certificates can be addressed. Obtaining public security information, like public keys, certificates, cross certificates and certificate revocation lists used to be done using the X.500 Directory Access Protocol (DAP). The latest version uses the Lightweight Directory Access Protocol (LDAP) instead of DAP to retrieve security information from directory servers.

**authentic8 PC**

authentic8 PC is a smart card based system of access control designed to protect stand-alone, single or multi-user IBM compatible PC's and notebooks from unauthorised use. This achieved through a security solution featuring challenge-response authentication (using RSA, with encryption performed on the actual smart card) and disk encryption. The certificate is stored on the smart card and complies with

ITU X.509 certificate format. The product is in the evaluation process with the target E1 of the ITSEC of the Defence Signals Directorate of Australia.

## 2.9 Interoperability of Trust Infrastructures

**Directory Based EDI Certificate Access and management (DEDICA)**

DEDICA is a research and development project funded by the European Commission under the Telematics Programme. The aim of the project is to specify a mapping strategy between data elements of X.509 and UN/EDIFACT certificates. This mapping will be implemented as a gateway that will translate certificates of one type into the other, allowing users having a certificate of one type to exchange messages with applications that require certificates of the other type. At the end of the project a certification infrastructure is envisioned to be available based on the DEDICA gateway that will allow the interworking between X.509 and UN/EDIFACT certification infrastructures through recognition of this gateway by the certification authorities of both infrastructures.

Currently there are two basic approaches available that can be used to establish an infrastructure for applying digital signatures and authentication mechanisms to large scale systems, the X.509 authentication model and the EDIFACT certification model. However the certificates used in those two models, although they contain similar information, are technically not compatible to each other. Thus there is no way to link systems, respectively their certification infrastructures if they adhere to different certification technology although their infrastructures may be logically compatible and the information carried in the certificates that allow their users to interact is basically the same.

The only solution up to now would be to bring all users, that are to interact with each other, under one common certification infrastructure. If infrastructures already exist these lead to a major effort for reorganisation and recertification. Moreover the application toolkits and programs used by the end-user are normally developed for interoperation with one particular certification technology. So either EDIFACT applications would have to get modified to use X.509 certificates or X.509 compatible applications would have to get changed to use the UN/EDIFACT certificate models. This normally is not feasible with reasonable effort. So this would lead either to have all the users use two kinds of programs and being registered with both security infrastructures or to provide connectivity by non-electronic means or electronically but without security.

DEDICA enables the interoperability between the two approaches by providing a gateway that allows for interconnecting security infrastructures of both types without users having to register in both systems or having to change application programs if working in the other infrastructure. This is accomplished by implementing functionality for translation and retrieval of certificates in the gateway.

An essential part of the translation of UN/EDIFACT and X.509 certificates is the mapping of the names of the respective participating entities in the other system.

X.500 Distinguished Names can not be used within UN/EDIFACT certificates, mainly due to length restrictions and UN/EDIFACT names are not structured like X.500 Distinguished Names and therefore cannot be used in the corresponding field of a X.509 certificate. In addition the name of entities in X.509 certificates normally show some kind of hierarchical relationship which is not available in names used in UN/EDIFACT certificates.

The DEDICA project has developed mechanisms for the mapping of names which are based on the general guidelines for registration and identification defined in the EDIRA Memorandum of Understanding. For example, the basic approach for mapping a name from UN/EDIFACT to X.509 certificates in very general terms is to generate a synthetic X.500 Distinguished Name which contains the relevant DEDICA gateway and append the UN/EDIFACT name as the common name to the synthetic X.500 Distinguished Name.

The DEDICA gateway is to be used similar to a mail gateway that moves messages from one mail world to another. DEDICA does not move messages but instead allows for translation and retrieval of certificates. It provides on request certificates in the format requested no matter under which certification technology the requested certificate is really held. The transport of secured messages and documents is not part of the gateway.

The DEDICA gateway differs from typical mail gateways in three aspects.

- Mail gateways are normally used in store-and-forward operation: They receive a message, store it, transform it and then forward it to other mail systems.
  DEDICA will be used more interactively. From both sides (X.509 and UN/EDIFACT) it will be used as a virtual repository of all the certificates of the other side. Users, respectively their user agents will query the DEDICA gateway as they would do with any other certificate directory in their environment and they will expect similar behaviour, functionality and performance.
- Mails can be transformed from one mail format to the other just by reformatting and possibly recoding of the mail message, eventually dismissing some of the header information.
  Certificates consist of the certificate structure containing the information to be certified and a certificate signature. The certificate structure may get transformed by some transformation rules changing structure and coding similar as this is done for the transformation of mail messages.
  The certificate signature instead depends on the structure and coding of the certificate structure. Since it is designed to be cryptographically secure to prohibit signature forgery and thus certificate forgery, modification of the certificate structure invalidates the certificate signature which can not

be reconstructed by the DEDICA gateway. Otherwise the signature function used would be insecure. Thus, since the transformed certificates also need a digital signature, the DEDICA gateway has to generate the respective certificate signature itself and to this respect act as an automatic operated certification authority.

- Once a mail message is transformed, the mail gateway is done with it. Maybe it additionally has to keep a backlog for backtracking purposes.
  When a certificates is translated and delivered, DEDICA is not done with it. It still has to provide information on the status, especially on the validity of the certificates it translated, signed and distributed. The DEDICA gateway will do so on the one hand by maintaining a table that allows the gateway to find the original certificate given the translated version and on the other hand by acting as a proxy: The translated certificates will get signed by the gateway itself due to the reasons given above, but status information will not be generated by the gateway but will be retrieved from the repository of the original certificate and then transformed and forwarded to the user. Thus although the translated certificate is signed by the DEDICA gateway in replacement of the signature of the original certification authority, the status and validity of the derived certificates are always the same as those of the original certificate. No additional means have to be established inside the certification infrastructures to forward status changes through the DEDICA gateway but the DEDICA gateway will forward this information automatically.
  The mapping table from derived certificates is required, since the derived certificates are now signed by the gateway itself and thus the information concerning the original certification authority normally also giving a point for lookup and status requests is no longer contained in the derived certificates. The DEDICA gateway needs this table to find out the original certification authority given a derived certificate to be able to do the status lookup on behalf of a gateway user as described above.

## 2.10 Assessment of Naming and Certificate Conventions

### 2.10.1 Assessment of Naming Conventions

#### 2.10.1.1 Functionality

An analysis of existing naming functionality shows that there are currently four naming conventions available which have a large distribution and which can be used for networks, the Internet Domain Name System, X.400, and X.500 and EDIFACT. The application of the X.500 naming conventions dominates in concepts for trust infrastructures. Whereas the Domain Name System, X.400 and X.500 are intentionally designed to support a broad spectrum of applications, the EDIFACT naming conventions are primarily oriented towards the commercial and administrative sector in the broadest sense.

The functionality of names has to be judged by looking at the syntax and the semantics used to construct names. An additional criterion that has to be looked at in the context of a distributed system, like the future European Trust Infrastructure, is the range of validity of names.

##### 2.10.1.1.1 Name Semantics

Common names

The use of common (personal) names (first name, middle initial, and family name) in certificates contributes to the acceptance of a trust infrastructure because common names allow the end-user to associate defined persons with a certificate. Basically all implementations and proposals which were analysed used these names in their usual form, i.e. without modifications or additions. Even in SDSI and in SPKI mechanisms are foreseen which would allow to use personal names, if needed.

Directory names

A modern trust infrastructure depends largely on the use of a directory. Such a directory allows for a relative uncomplicated management of the name space and as an additional benefit for storage of supplementary information about the objects using such an infrastructure. The objects are identified by directory names that are structured and consist of a set of concatenated attributes added to a (possibly ambiguous) common name for the object, which are used to provide for its uniqueness. These attributes provide information about the position of an entity in a directory structure and in addition they show the position of an object in an organisational structure if the directory is structured accordingly.

Directory names in X.500 systems as well as the comparable names in the Internet Domain Name System reflect the structure of the supporting infrastructure. A very detailed structure of the directory leads to very long directory names which are difficult to remember and to handle. On the other hand directory names can be very informative if adequate attributes are used for the composition of a directory name and are therefore easy to comprehend by the end user. Dependent on the depth of the infrastructure (number of concatenated attributes) their handling however can be cumbersome.

O/R names

O/R names are primarily used for interpersonal messages (Message Handling Systems, MHS) and in electronic data interchange (EDI). They can be very long and therefore difficult to remember and to handle. Their advantage is that they convey a lot of organisational information about the members of a communication process which is relatively easy to understand and that there is no prescribed sequence

of the attribute/value combinations which have to be concatenated to build an address.

Absolute names

Absolute naming implies that a complete name is assigned with respect to a universal reference point (for example the root of a directory tree). The advantage of absolute naming is that a name thus assigned can be universally interpreted with respect to the universal reference point. Both the Internet naming convention and X.520 allow for absolute naming.

Relative names

For relative naming, an object is named depending upon the position of the naming object relative to that of the named object. For example relative naming is used by the uucp protocol between UNIX-hosts. In that case a mail receiver is typically named by a route specifying a chain of locally known hosts linking the senders host to the recipient's host.

Experience from using the uucp protocol in the ARPANET/Internet community has demonstrated that there are problems inherent to relative naming, like for example in routing optimisation.

Experience with PEM on the other hand has shown that the pure top-down hierarchy with all certification paths starting from the root, is too restrictive for many purposes. For some applications, verification of certification paths should start with a public key of a CA in a user's own domain, rather than mandating that verification commence at the top of a hierarchy. In many environments, the local domain is often the most trusted.

#### 2.10.1.1.2 Name Syntax

Numerical names

In general, pure numerical names, as well as pure numerical addresses have the advantage that they can be used in computer systems without complicated and time-consuming translation mechanisms. Although it is conceivable to provide them with a structure they are not user-friendly and the maintenance of such a naming system based on numbers in a very dynamic European and global environment would require considerable administrative effort, especially because of the requirement for unique names.

The same is true for the approach of SPKI where a public key or its hash value are used to represent objects, however in these cases the numerical values can not even be structured.

#### 2.10.1.1.3 Name Validity

Global names

The majority of projects and proposals analysed use a hierarchically organised structure of the directory as proposed by X.500. This requires the use of names that have to be globally unique. Practical experience with the available naming conventions has opened the discussions whether a global name-space (globally unique names) or a local name-space are best suited for the problems connected with handling trust. The need for globally unique names is being challenged at least by the SDSI and SPKI projects for the following reasons:

- a global Directory with globally unique names (as envisioned by the creators of X.500) will be too complex and therefore still not yet in existence,
- global name spaces are politically and technically difficult to implement,
- a name space that is too large defeats security,
- the effort required to set up an X.500 Directory is considered to be too high,
- it is possible that an X.500 Directory will never come for reasons of protection of company or private data (companies don't want to reveal their internal structure, people do not want to reveal too much about their personal data),
- the dependence on global names in the end-user domain is considered to be a limiting factor to flexibility,

Local names

SDSI and SPKI propose to use local names. From their point of view a name in order to be meaningful, must be known by the person or program using it (the verifier) and must be unambiguously associated by that verifier with the person or object (the subject) to which it refers. This is claimed only to be feasible in a trustful manner in single communities of interest but not on a global level with global names.

### 2.10.1.2 Flexibility

Basically the question of the flexibility of a naming convention is the question of the flexibility of the directory system which provides the name service. In the Internet Domain Name System as well as in X.500 the name services are based on a hierarchical structure. Almost all projects and proposals that have been analysed have chosen a hierarchical structure for naming.

Naming systems based on a hierarchical structure have their merits because it is easy to provide unique names by appropriately designing the structure and it is also easy to decentralise (distribute) the management of names while maintaining global name uniqueness. The consequence for the Internet as well as for X.500 systems however is that the hierarchical structure of the naming system is mapped onto the names. The names therefore are not independent from the structure of the system or the infrastructure anymore.

The rigid rules required for naming limit the flexibility especially in the local area. Modifications caused e.g. by reorganisations internal to a company are difficult to implement in a global naming structure. Problems exist also at higher levels in the naming structure where the introduction of new elements results in an implementation of new Relative Distinguished Names with the consequence that all the effected Distinguished Names might have to be modified also in order to restore the verification path. This however might be tolerable, because the need to change RDNs at higher levels might be minimal if the initial structuring of the name service has been done carefully.

The approach to use names for supporting trust as implemented in PEM (name subordination) leads to undesirable constraints upon the X.500 naming system because names have to be built into the verification logic, which means that an organisation is forced either to distort its overall X.500 naming structure to accommodate artificial nodes in the directory structure for certification authorities or to have organisation or organisational units share X.500 names with their certification authorities

As to the flexibility of name composition, in contrary to X.400 where the sequence of attribute/value pairs is not prescribed but only recommended by the standard, X.520 Distinguished Names and also names in the Internet Domain Name System have to follow a defined sequence.

#### 2.10.1.3 Ease of use

The requirement for ease of use of names has inherently been addressed in all of the projects and proposals that have been analysed. In theory the naming conventions have to be based on the requirements of the user and not on the requirements of the technical system. Although this requirement is not fulfilled totally because of the dependency of the names from the structure, the naming conventions as specified for the Internet Domain Name System, for X.400, and X.520 can be considered as an acceptable compromise and they can be recognised and understood quite well by people throughout the world.

In certificates, especially in identity certificates, Distinguished Names allow to identify the subject and the issuer very clearly.

Distinguished names may convey the impression of being complex and difficult to handle, however this could be hidden to the user by an adequate user interface.

#### 2.10.1.4 Support of User Trust

User trust is a major prerequisite for the acceptance of a trust infrastructure. Naming conventions can contribute to that trust by providing the end user with clear and comprehensible information to recognise the objects involved in an exchange of all kinds of information. The projects and proposals analysed acknowledge this need and by applying the X.520 naming rules they allow for the construction of names which can be read and understood by end users. SPKI, although in favour of using public keys to identify principals, has also mechanisms to use human readable names, if needed.

An additional approach to use names for supporting trust has been used in PEM, where a trust relationship between a certificate issuer and the subject (user) is expressed via the naming hierarchy (Distinguished Name subordination rule). This solution has proved to be overly restrictive.

#### 2.10.1.5 Interoperability

The projects and proposals analysed recognise the likelihood that different certification structures eventually have to co-operate. If the other structures use the X.520 naming conventions to denote subjects and issuers in the certificates, interoperability from the viewpoint of names is no problem. However if the naming conventions of the other structures are different (e.g. X.500 and EDIFACT), the solution that has been proposed by the DEDICA project is a process which translates the names. Such a translation process can either be implemented locally (at each user's server) with the consequence that such a process is necessary for each different naming convention the user has to deal with or it can be implemented centrally (e.g. by a gateway) which has the advantage that the translation process only needs to be implemented once.

In the case of message exchange the problem of different naming conventions is solved by either incorporating a foreign name domain anywhere in the hierarchy (Internet) or by creating a new object class in the X.500 Directory Information Tree (e.g. object class EDI user).

#### 2.10.1.6 Adherence to Standards

The projects and proposals analysed that use a hierarchical certification structure are following the relevant standards X.520 for naming subjects and issuers in the certificates (Distinguished Names). They also allow for using the specified extensions of X.509 v3 to use alternative names, expressed in the form of the relevant standards. To describe the values for the country attribute the ISO Standard 3166 is used.

The „subjectUniqueID“ field which would allow to combine the subject name with a unique identifier of the subject (personal number, payroll number) and therefore could be used to allow for the reuse of the name specified in the subject field of the certificate is not being used, mainly because there are other possibilities in a Distinguished Name to make it unique (like adding an additional attribute/value to the RDN, which world provide for uniqueness).

The SDSI proposal that is in favour of local names is capable of supporting all forms of naming standards, there is no requirement to adhere to one specific standard.

#### 2.10.1.7 Legal Aspects

There is a user interest to know as much about the subject and the issuer of a certificate, because such knowledge would contribute to the trust that is invested into the certificate. There is also a danger in X.500 systems that in order to arrive at unique names the Directory Information Tree will be specified in too much detail. In both cases the result would be a very detailed Distinguished Name which would eventually reveal too much specific information about a person to the public. It has also been published that companies are reluctant to connect their organisational directories to a global directory infrastructure because of privacy concerns. The same danger arises when using X.400 addresses. At the moment it is not known whether there exists any legislation that regulates the composition of a Distinguished Name from the viewpoint of privacy.

### 2.10.2 Assessment of Certificate Conventions

X.509 is the standard used for certificates and certificate revocation lists. After the improvements added to the original X.509, the current version 3 (X.509 v3) has received widespread recognition as being feasible and mature enough for application in trust infrastructures.

There are only two different approaches: the SDSI proposal and the SPKI proposal – but implementations are not fully operational.

However the broad spectrum of features which are contained in the standard call for deliberations whether profiles should be developed which are targeted specifically at a European Trust Infrastructure. If this should be determined to be sensible, care has to be taken not to prohibit interoperability with other infrastructures. The possibility to define private extensions for the certificates which contributes to their flexible use has the inherent danger, that when this features are used without care, the certificates may become unintelligible. Current discussions in the Internet PKIX working group show that details of the standard are still in discussion but this does not prevent the general acceptance.

In this chapter therefore only the X.509 v3 certificate is discussed.

#### 2.10.2.1 Functionality

##### 2.10.2.1.1 Certificate Semantics

The certificate semantics are fully described including extensions and certificate revocation lists.

##### 2.10.2.1.2 Certificate Syntax

There is a formal definition of the syntax. Parsers are available.

##### 2.10.2.1.3 Certificate Validity

A certificate may become invalid because of the algorithm used may become weak or vulnerable to attacks, or because the key length is not longer sufficient.

The revocation of a certificate may not be published in a timely fashion. Currently certificate revocation lists cannot be revoked.

#### 2.10.2.2 Flexibility

The problem of invalid certificates and keys is solved by certificate revocation lists which are flexible enough.

All the discussed activities can be processed automatically, which would not require understanding the actual meaning of a request or verification part by the user but the user only specifies the required trust level.

The user entity may decide what level and quality of trust it needs and this may differ depending on the application.

The certificate extension fields allows the addition of new fields to the structure without modification of the formal definition. There are standard extensions and private extensions may be added by communities to carry information unique to those communities. The combination of critical and non-critical extensions satisfies all requirements.

X.509 allows only special public encryption systems with a special property of symmetry to get integrity and confidentiality with the same system (RSA). This prevents flexibility.

#### 2.10.2.3 Ease of use

The requirement for ease of use of certificate handling has inherently been addressed in all of the projects and proposals that have been analysed. Until today there is no trust infrastructure implemented handling certificates fully automated and transparently for the user.

#### 2.10.2.4 Support of User Trust

User trust is a major prerequisite for the acceptance of a trust infrastructure. But until today there are no procedures implemented to specify specific trust levels. Furthermore there are no procedures developed to evaluate the useful trust level for a specific service.

#### 2.10.2.5 Interoperability

The projects and proposals analysed recognise the likelihood that different certification structures

eventually have to co-operate.

#### 2.10.2.6 Adherence to Standards

X.509 v3 is an international standard, widely accepted and implemented in many projects.

#### 2.10.2.7 Legal Aspects

The certificates, but especially the certificate revocation lists and also the black lists contain much information – some may be privacy information.

#### 2.10.2.8 Security Aspects

If the issuer public key parameters are used from the X.509 certificate, an attacker who wants to replace, modify or create bogus certificates and certificate revocation lists, can substitute these values in the certificate and in the certificate revocation lists and resign the certificates and certificate revocation lists (substitution attack). This allows the attacker to translate a hard public key cryptography problem into one of finding a new set of parameters and private key that are consistent with the trusted public key. The implications for various cryptosystems are different.

The X.509 certificate has only a two digit field for the year. That might be a problem facing the "millennium bug" and should be fixed.

Until today there is no risk analysis or specifically no vulnerability analysis available which covers the security of the certificate structure and its handling or the tools used as well as for the formal specifications like the Abstract Syntax Notation 1 (ASN.1) for structuring complex data objects.

There is an inherent security problem in the design of trust infrastructures, the case where a certificate becomes invalid shortly after verification and before computing it. There is a second (similar) unprotected time in the time between getting a certificate revocation list and computing it. Such untrusted situations are not avoidable.

For certificates with a long life-time, the security of binding between a public key and the subject becomes even more problematic.

# 3 Certificates and Names in a European Trust Infrastructure - ETI

## 3.1 Recommendations for a ETI

There are several design issues for a trust infrastructure like avoiding heterogeneity, scalability, manageability, performance, cost, simplicity, modularity, etc.

Functional requirements and standard requirements have to be merged to produce a set of detailed requirements for the ETI. This section describes the requirements for the primary objects and components of the ETI.

The primary objects in the ETI are the
- certificate format,
- certificate revocation list, and
- compromised key list.

The ETI components are the
- client workstations,
- certification authorities, and the
- directory servers.

The requirements for ETI objects determine their format and content and the general functionality of each component.

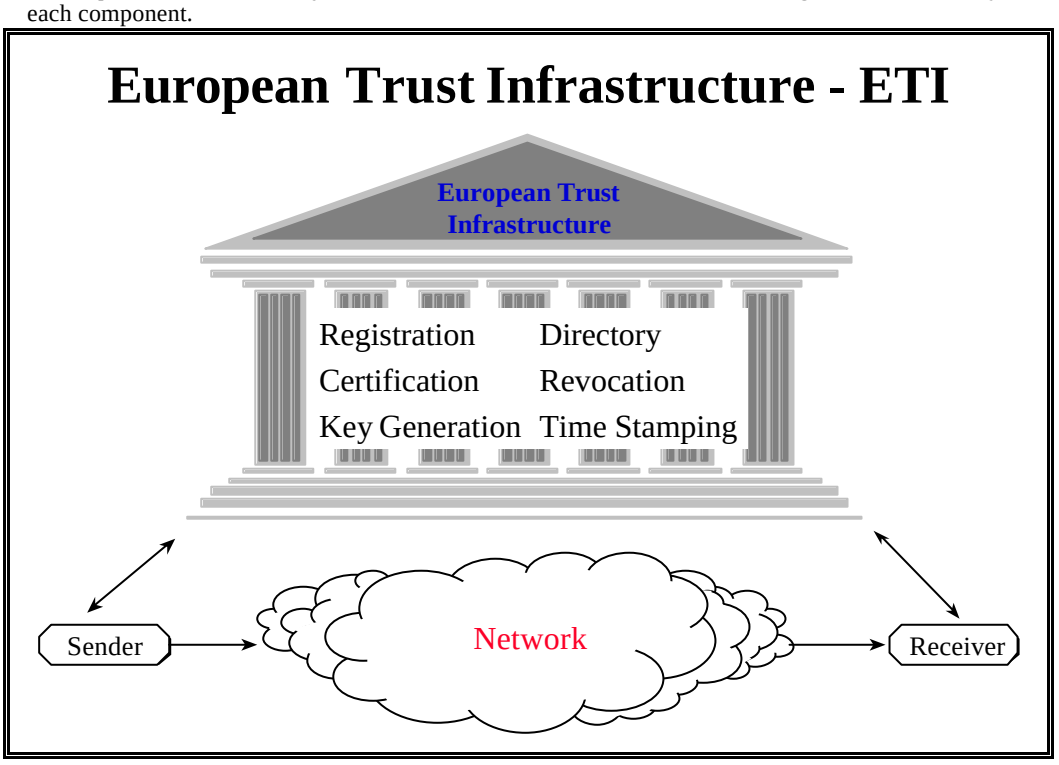


Figure 59: European Trust Infrastructure

**Transparency**

A well designed trust infrastructure is transparent to its users in the daily operations with the exemption of the necessary access control to the system, which allows the user to apply the security services. A necessary feature are user-friendly interfaces and tools for managing certificates. The handling of certificates shall be fully transparent e.g. the verification of a certificate on different levels of trust and also in the different steps in the chain of hierarchical certification authorities or cross-certified authorities depending on the verification architecture.

**Scalability and Manageability**

The ETI must be able to manage certificates for a large distributed work force cost-effectively while enforcing clear and consistent policy. This implies that the ETI must be flexible, scalable, manageable, and uniform. These requirements directly affect the structure of the ETI.

There are three basic architectures for a ETI: First an architecture with a single centralised authority. Second a hierarchical structure with several levels of certificate management authorities. Third a collection of cross-certified certification authorities. A centralised authority is relatively inflexible and does not scale well. A hierarchical ETI is scalable, can enforce uniform policy, ensure interoperability within the ETI, and set policy for interoperation with users of non-ETIs. A collection of cross-certified

certification authorities is flexible and scalable, but difficult to manage; it requires each CA to enforce policy and set policy for interoperation with users of other certification authorities and non-ETIs.

Managerial requirements argue for a hierarchy of certificate authorities. However, the flexibility of cross-certification is required to meet local requirements, enhance performance, and interoperate with non-ETIs. A hybrid architecture can be devised with a hierarchical management structure that permit cross-certification.

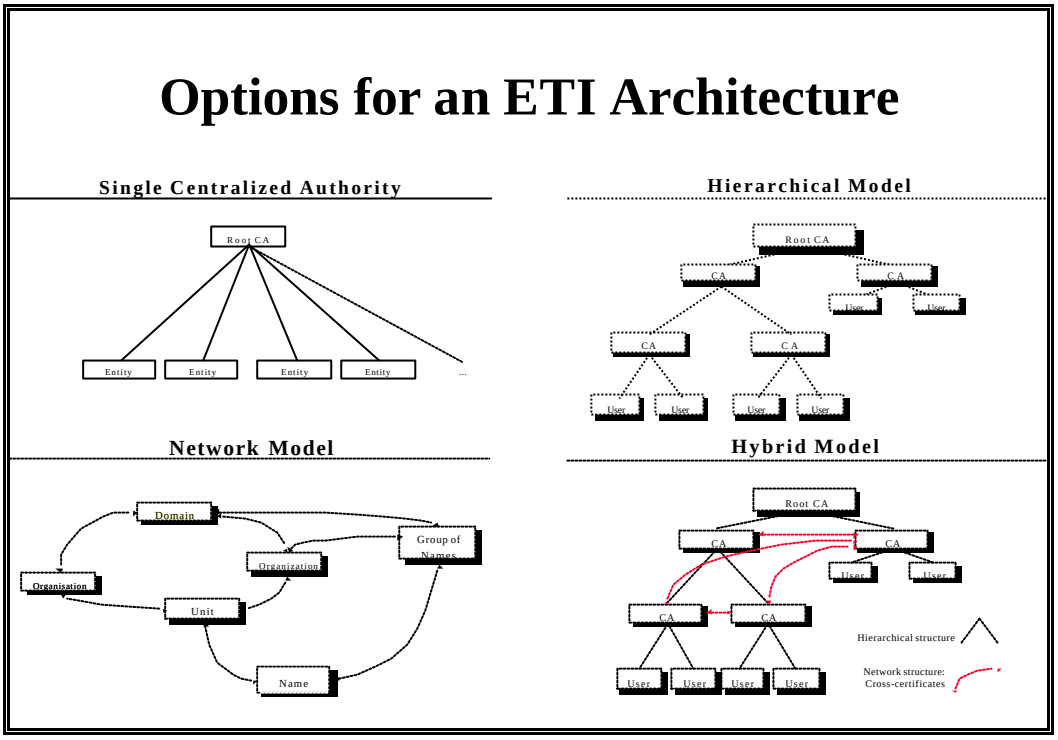


Figure 60: Options for a ETI Architecture

A policy approval authority (PAA) manages the certification authorities through this hierarchical structure. Cross-certificates which violate the hierarchical structure are permitted to meet local requirements and enhance performance, as permitted by the PAAs guidelines.

To be completely general, one may logically separate the functionality of certification from the functionality of revocation and introduce a separate revocation authority. Naturally this does not mean that the two functions have to be performed by physically different authorities.

**Certification and Accreditation of Authorities**

To ensure the integrity of the infrastructure, all certification authorities and organisational registration authorities have to be periodically reviewed to ensure that they are implementing the appropriate ETI security policy. A certification authority accreditation process must be established to perform these reviews. The policy approving authority will initiate third party audits against the stated policy and review the results.

If the security and the procedures are found to jeopardise the binding of the user and certificate, all subordinate certificates will be revoked. New certificates may be issued after the violation are corrected.

The policy approving authority will determine the period between reviews, credentials for auditors, and the response associated with particular failures of oversight or oversights will be specified in the ETI policy.

**Security**

Because of the importance of available and reliable communications in Europe it is necessary for the ETI to fulfil some security requirements. Those requirements concern especially security aspects like the trustworthiness of the organisation, the hardware and software and the employees; those requirements are to be checked at the establishment of each authority and also periodically.

One requirement for example is, that the scheme should ensure that any attempted abuse of a authority or the trust infrastructure as a whole by a user or an employee or other entities can be detected, and in addition those with lawful authorisation for access to information cannot fabricate false evidence.

It is strongly recommended to analyse the level of security of the certificate type recommended together with all the tools used as well as the formal specifications like the Abstract Syntax Notation 1 (ASN.1) for structuring complex data objects, the organisational procedures, etc. by conducting a risk analysis and especially a vulnerability analysis.

All the tools used defining objects, and processing shall be evaluated using the ITSEC or - if generally

accepted - the Common Criteria. ETI policy will specify the assurance level required of implementations for ETI components and users.

Certification authorities and organisational registration authorities will be established and maintained in a secure fashion, to ensure that the certificates and CRLs of the ETI can be trusted. The ETI will establish security requirements for certification authorities and organisational registration authorities. These requirements will specify minimum levels of assurance (of Hardware and software), physical security, personnel security and emanation security. ETI certification authorities will be required to develop and implement disaster recovery plans.

The ETI will develop requirements for cryptographic modules for ETI entities and users. ETI policy will identify appropriate signature, encryption, and key exchange algorithms and authentication protocol for use by ETI users. ETI policy will encourage and support the use of these standard algorithms.

To maintain the integrity of the system, minimum standards of identification for creation of new user certificates will be specified in the ETI policy. For specific application areas certificates may be issued under more stringent requirements to meet their requirements.

## 3.2 ETI Trust Policy

The ETI trust policy shall define two types of policies for certification authorities:

- certification authority operational policies and
- certificate issuance policy.

Certification authorities operate under one explicitly defined certification authority operational policy, but may issue certificates under several certificate issuance policies depending on the assurance level for the certificate requested. The combination of the certification authority operational policy and the specific issuance policy used to authenticate the identity of a certificate holder defines the certificate policy as identified in the extensions to the X.509 v3 certificate.

Real time evaluation of the actual policies used to issue the signer's certificate and to operate the certification authorities involved in the signer's certificate chain is too complicated to be efficient and reliable. To avoid that problem, it is recommended to define seven levels of trust (see Figure 62: Hierarchical Levels of Trust).

The assessment of the trustworthiness of the information in a certificate is made by the PAA upon evaluating the policies and supported procedures followed by the certifying certification authority. Certification authorities may assign only one trust level to any certificate.

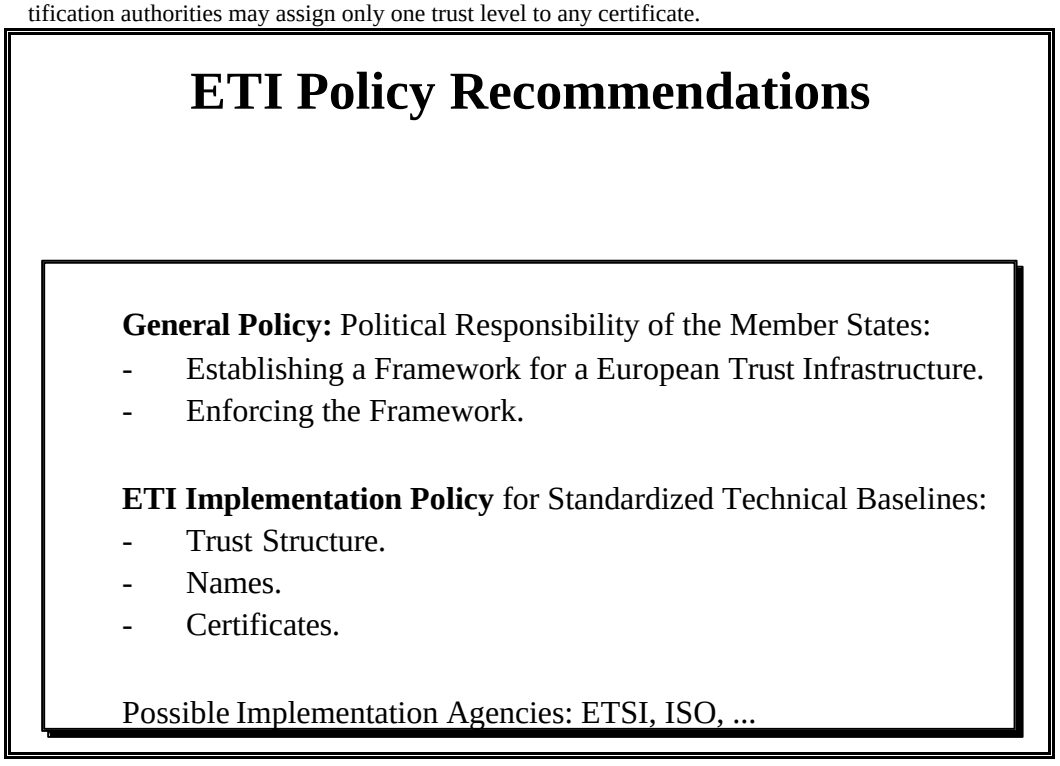


Figure 61: ETI Policy Recommendations

### 3.2.1 Levels of Trust

The quality of the verification procedure shall be an indication of the general level of trust that can be placed on a certificate. The levels are described in the Figure 62: Hierarchical Levels of Trust.

| Hierarchical Levels of Trust in Messages, Signatures, Certificates and Revocation Lists | | |
|---|---|---|
| **Level** | **Description** | |
| 0 | An entity sends a pure (digital) **message**. The expected receiver may get and accept a message.<br>Both, sender and receiver trust in the integrity and authenticity, non-repudiation etc. of each other. | |
| 1 | An entity sends a digitally **signed message**.<br>The expected receiver trusts in the digital signature to be the signature of the expected sender. | |
| 2 | An entity sends a digitally signed message together with a **certificate** certifying the public verification key sent, to be the key of the entity. The expected receiver trusts in the certificate, the issuing authority and the digital signature to be the signature of the expected sender. There may be or may be not a field with a validity period. | |
| 3 | An entity sends a digitally signed message together with a certificate certifying the public verification key sent, to be the key of the entity.<br>The expected receiver checks the validity period of the certificate. If the **validity period** is not expired, the certificate is valid. The receiver trusts in the certificate, the issuing certification authority and the digital signature to be the signature of the expected sender. | |
| | **On-line Verification** | **Certificate Revocation List** |
| 4 | An entity sends a digitally signed message together with a not expired certificate, certifying the public verification key sent, to be the key of the entity.<br>The expected receiver verifies the digital signature by verifying the certificate and sending a request to the authority issuing the certificate. The issuing certification authority sends back a digitally signed message certifying the combination (**binding**) of the public key and sending entity. | An entity sends a digitally signed message together with a certificate certifying the public verification key sent, to be the key of the entity. The validity period of the certificate is not expired.<br>But there may be other incidents: For example the private key may be compromised in the meantime. The expected receiver checks the validity of the certificate by requesting (pull-model) a certification **revocation list**. The requested certification authority sends a digitally signed CRL together with a certificate. The receiver trusts in all the certificates sent, the issuing authority and the digital signature to be the signature of the expected sender. The receiver stores the CRL certificate until the validity period of the CRL certificate expires for further checks.<br>There is no relevant difference to the push-model of CRL distribution: The receiver checks the last received CRL. |
| 5 | An entity sends a digitally signed message together with a valid (validity period not expired) certificate, certifying the public verification key sent, to be the key of the entity. The receiver verifies the certificate by requesting the issuing authority.<br>The expected receiver verifies the issuing authority by requesting the **hierarchical higher or cross-certified** authority in the certificate chain for the issuing authority. This (higher or cross-certified) certification authority sends a certificate back to the receiver, certifying the issuing authority. | To avoid problems with incidents like compromised keys, again a request for a CRL is necessary for **verifying the CRL**-issuing authority. |
| 6 | To get more trust, the expected receiver will request the more higher or cross-certified certificate authority for a certificate. And so on, and so on ... until the hierarchical **highest or last cross-certified** national or European-wide certification authority is reached.<br>The number of requested certification authorities in the hierarchy may be limited by a policy. | And so on, and so on ... on each level there is a **check for the CRL** possible. |

Figure 62: Hierarchical Levels of Trust

But there may arise some problems in short time affairs:

- In the case the receiver fears the certificate becomes invalid by an incident like a compromised key during the time he requests a verification and gets an answer back from an authority, certifying another authority or the binding of some data. During this time the requestor is unprotected by the trust infrastructure. The time of unprotection depends on the net traffic load, the bandwidth capacity of the net and the execution time of his computer and the computers of the participating authorities.
- There is a second (similar) unprotected time in the time between getting a requested or pushed CRL and computing it.

It is not possible to avoid such untrusted situations. Certificates are generated only for a time interval in the past. Trust in this discussion is the hope the keys are not compromised in the meantime and there are no other incidents which attack the validity of a certificate.

It is no solution, to shorten such unprotected time intervals by realising the mentioned affairs only after a distinct time interval, which is definitely longer then the mentioned execution time of the computers and the time needed for the net transactions: The validity of the certificate may be endangered just in the moment after the interval.

All the mentioned activities shall be processed automatically, which would not require understanding the actual meaning of a request or verification part by the user but the user only specifies the required trust level.

The categories of trust seem to be out of bands not hierarchical – nevertheless the user entity decides what level and quality of trust it needs and this may differ depending on the application.

### 3.2.2 ETI Objects

To support the required security services, the ETI must manage five basic objects:

- certificates,
- cross-certificate pairs,
- certificate revocation lists (CRL),
- compromised key lists (CKL), and
- security policies.

#### 3.2.2.1 Certificates

The ETI public key certificate format should be an X.509 Version 3 certificate, and should support two basic classes of certificates:

- public key certificates to support signature applications,
- attribute certificates which specify user privileges or role or indicate ETI accreditation of third party services.

#### 3.2.2.2 Cross-Certificate Pairs

The ETI will support the X.509 cross certificate pair construct to build bi-directional chains of trust between certification authorities. The cross certificate pair construct contains two certificates:

A "forward" certificate and a "reverse" certificate. The subject of the forward certificate is the issuer of the reverse certificate and vice versa.

Cross certificate pairs may define new paths that do not parallel the hierarchy.

#### 3.2.2.3 Certificate Revocation Lists

The ETI certificate revocation list shall be based on the X.509 Version 2 CRL. The format shall support:

- Modular presentation of CRLs, and
- knowledge of why certificates were revoked.

Reason codes and date and time of revocation will be required for all CRL entries.

The ETI certificate revocation lists will support rapid dissemination of revoked certificates throughout the infrastructure. This requires that all certification authorities support a common list format and generate certificate revocation lists for exchange within the infrastructure. The frequency of certificate revocation list generation is a technical policy issue.

It might be useful to support revoked (compromised) key lists as well. All the unexpired certificates based on revoked keys have to be listed in a certificate revocation list also.

#### 3.2.2.4 Black Lists

The names of users who have been blacklisted, or whose certificates and/or keys have been revoked are listed in a so-called black list. It is necessary to check the history of a user for the reasons and frequencies of revocations. This is a question of a black list see chapter 2.4.0.3 Revocation Process.

Unexpired certificates for users in the black list shall be revoked and included in the certificate revocation list. No new certificates are created for blacklisted users.

| Black List | | |
|---|---|---|
| Version | | |
| Signature algorithm identifier | | |
| Issuer name | | |
| Black list serial number | | |
| Last update | | |
| Next update | | |
| Blacklisted user name | Date/time | Reason |
| Blacklisted user name | Date/time | Reason |
| ... | ... | ... |
| Black List issuer's signature | | |

Figure 63: Fields of the Black List

### 3.2.3 ETI Organisational Components

This section describes the functionality of each class of ETI components. The ETI shall include the following components:

- A policy approving authority (PAA),
- Policy certification authorities (PCAs),
- Certification authorities (CAs),
- Organisational registration authorities (ORAs),
- Directory servers, and
- ETI client workstations.

#### 3.2.3.0 Service primitives

The following services to be provided in the ETI are:

- Enrolment primitives:
  - Registration,
  - certification,
  - key generation.
- Distribution primitives:
  - Key distribution,
  - key storage and protection,
  - key authentication and verification.
- Maintenance primitives:
  - Key activation,
  - key replacement and deletion,
  - revocation of certificates.
- Basic functions:
  - User entity registration and certification,
  - certificate revocation,
  - time stamping,
  - renewal of a certificate,
  - enquiry of the directory of certificates,
  - user scenarios

The services will be based on a general model which separates the main functions into logical entities:

- Key generation (KG) - the entity responsible for the generation of an asymmetric key pair.
- Registration authority (RA) - the entity responsible for the registration of a user entity.
- Certification authority (CA) - the entity responsible for certifying the public key of a user entity.
- Time stamping authority (TA) - the entity responsible for time stamping digital signatures.
- Directory (DIR) - the entity responsible for holding the information contained in the directory (registration, certificate) online for ready use by the user entities.
- User entity (U) - an object which is unambiguously identified by its distinguished name.

The user entity requests enrolment by the registration authority. The registration authority verifies the authenticity of the user's credentials and possibly adds system-specific data.

The user entity further submits its public key to the registration authority in an authenticated way and proves its knowledge of the corresponding private key. The registration authority verifies the integrity

of the public key using an authentication value e.g. a hash code calculated on the public key and transmitted from the user entity to the registration authority by some other means.

The tasks of the time stamping authority and the certification authority may be carried out by one and the same trusted third party, may be the central certification authority.

Since the key functions within the system are in the hands of the registration authorities and certification authorities, identified as trusted third parties, one also needs to define accreditation criteria for such authorities in the certification authority policy.

#### 3.2.3.1 Policy Approving Authority

The ETI policy approving authority must:

- Establish policy for policy certification authorities, certification authorities, and organisational registration authorities, including:
  - personnel security,
  - archiving, logging and auditing, and
  - physical and systems security.
- Review compliance of policy certification authorities and ORAs,
- establish policy identifiers for use throughout the ETI,
- set policy for cross certification between the certification authorities.
- approve interoperation between the ETI and non-ETIs.

#### 3.2.3.2 Policy Certification Authority

Each Policy certification authority (PCA) being certified by the PAA must establish and publish a statement of its policy with respect to certifying users or subordinate certification authorities. Distinct PCAs aim to satisfy different user needs. For example, one PCA (an organisational PCA) might support the general e-mail needs of commercial organisations, and another PCA (a high assurance PCA) might have a more stringent policy designed for satisfying legally binding signature requirements.

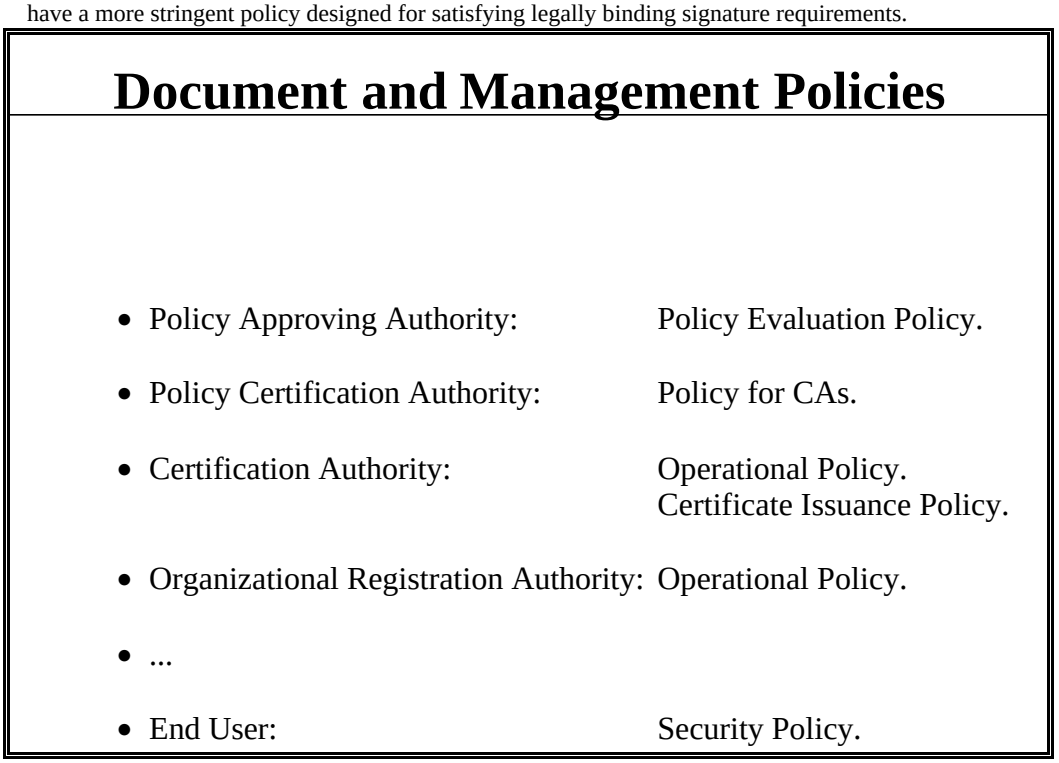


Figure 64: Document and Management Policies

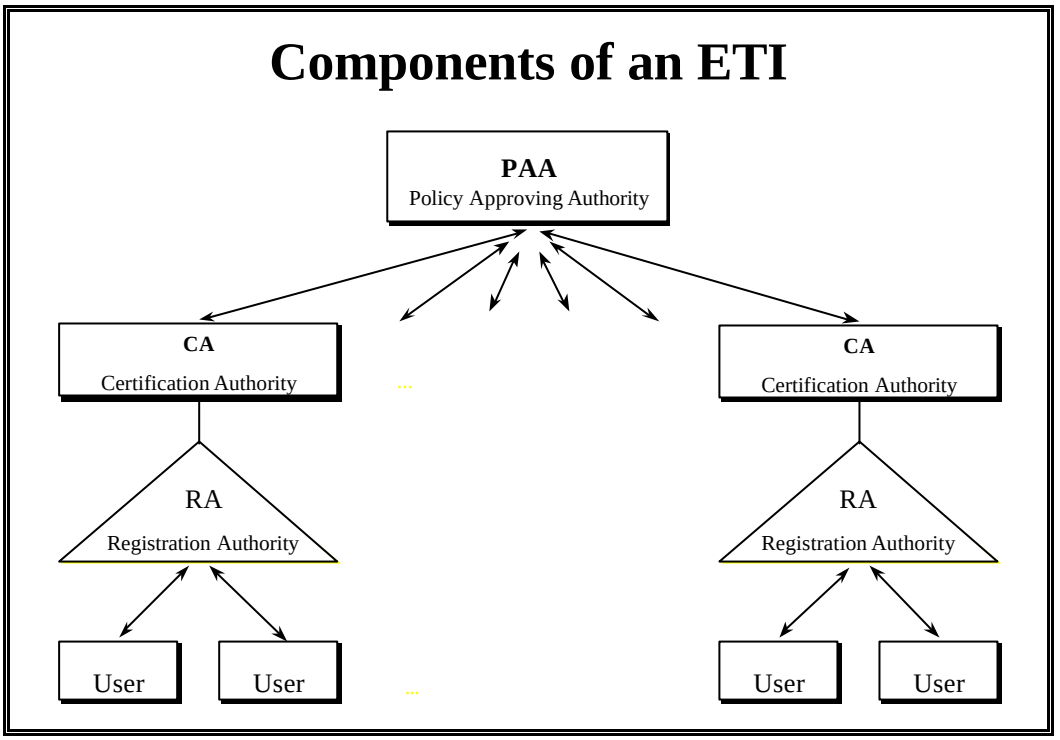


Figure 65: Organisational Components of a ETI

#### 3.2.3.3 Certification Authorities

The ETI certification authorities will:

- Generate and verify ETI certificates and especially digital signatures,
- access X.500 directory service using a directory access protocol,
- generate certificate revocation lists and compromised key lists,
- on request by the user entities generate key (public and private) pairs and perform quality tests,
- verify uniqueness of public keys,
- confirm that certificate requestors possess the private key corresponding to the public key in the requested certificate,
- maintain a permanent archive directly or through use of a third party digital archiving service,
- accept electronic certificate issuance and renewal requests,
- verify uniqueness of subordinate names,
- generate cross certificate pairs,
- maintain secure off-line storage for private key back-up on request,
- perform periodical back-ups ,
- log transactions to non-volatile storage media, and
- support required communication protocols.

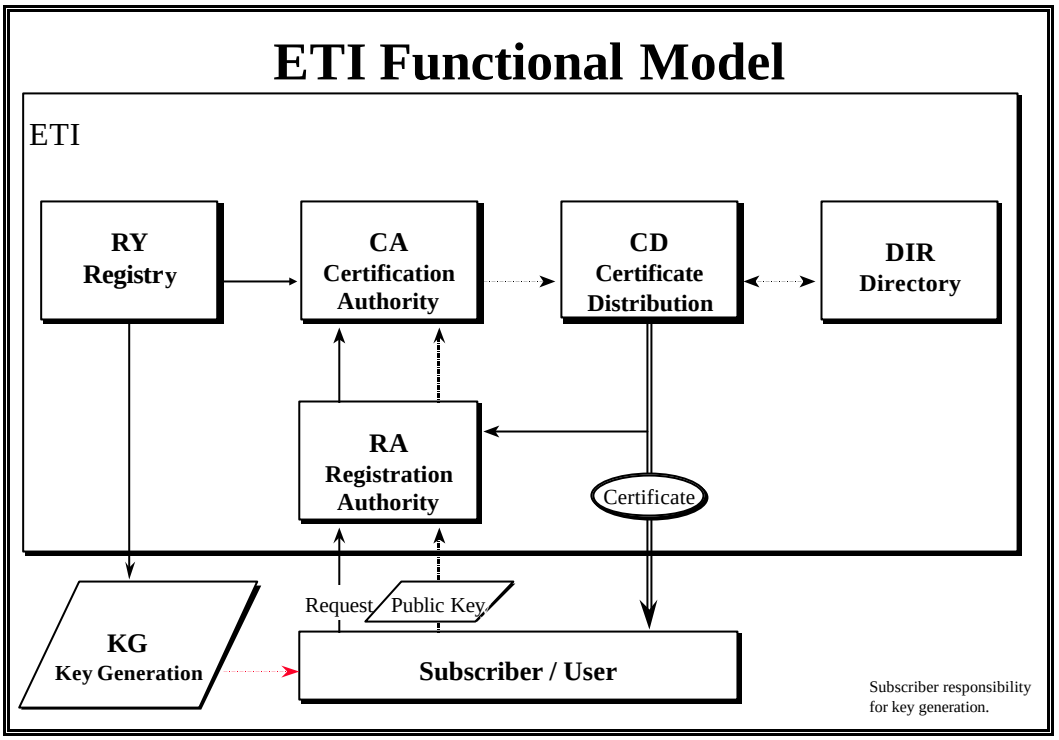


Figure 66: ETI Functional Model

The ETI certification authorities must be able to generate cross certificate pairs, containing at a minimum the forward certificate, to represent the hierarchy leading from the root. Optionally, certification authorities may be able to generate reverse certificates to provide "full" cross certificate pairs. The root certification authority must be able to generate "full" cross certificate pairs to support mutual recognition with non-ETIs.

Certification authorities shall support the generation of cross certificate pairs with other certification authorities within the ETI to define trust paths that are not parallel to the hierarchical structure.

Certification authorities shall use standard protocols for all transactions. Transactions may be implemented using any appropriate networking technology or via dial-up access as required by local systems.

Certification authorities shall maintain a database of information regarding owners of certificates issued. This information might include contact information, such as postal address, phone numbers and electronic mail addresses, and may include private user information such as social security numbers and their mother's maiden name. This would impose additional security requirements from the privacy act.

Certification authorities will only issue certificates to members of an organisation (e.g. employees) at the request of their organisation. A certification authority may optionally confirm that the organisation has requested a certificate for this user.

#### 3.2.3.4 Organisational Registration Authorities

The organisational registration authority (ORA) will sign certificate requests for users who appear in natural or legal person with appropriate identification. Organisational registration authorities may also confirm that the organisation has requested a certificate for natural users.

ORAs will interact with certification authorities using communications protocols as determined by the certification authority.

For convenience, initial registration shall be a local service.

#### 3.2.3.5 ETI Clients

ETI clients must provide the following capabilities to participate fully in the ETI:

- Generate digital signatures,
- verify digital signatures,
- interpret X.500 V3 certificates,
- interpret ETI certificate revocation lists and compromised key lists,
- determine the status of a ETI certificate,
- obtain certificates from the X.500 directory servers.

This is the minimum functionality for generation and verification of certificates.

Clients with a limited role (i.e. sign only) may implement a subset of this functionality.

#### 3.2.3.6 Directory Servers

The ETI shall maintain a permanent archive of certificates so that business and legal requirements for

electronic commerce can be met. The certification authority can maintain its own archive, or use of an approved third party server.

X.500 provides the basic specification for implementing ETI directory service. The directory servers shall support public access via dial-up and on-line access. The directory servers will chain to directories maintained by cross-certified ETIs.

The directory will include current certificates issued, indicating when the individual certificate was issued, when it expires or when it was revoked, including cross-certificate pairs and certificate revocation lists issued by the ETI certification authorities. Certificates and cross certificate pairs must be maintained in the directory for some time period after expiration or revocation to facilitate verification of documents signed just before the certificate was revoked or expired. The length of time the certificates should be maintained is a technical policy issue. The directory will make all published certificates available upon request. It is unclear if a record of all certificates issued is required to meet business and legal requirements.

Directories must be maintained, even when a CA or archive server is disbanded. These archives will be maintained by the ETI.

The ETI must support X.500 directory services for key distribution to private sector entities. Proprietary solutions for trusted directories may be required for certain government and private sector applications.

The ETI shall be liable to any natural or legal person who has acted in good faith in reliance on a certificate issued by the certification authority for any circumstances. The liability shall be limited.

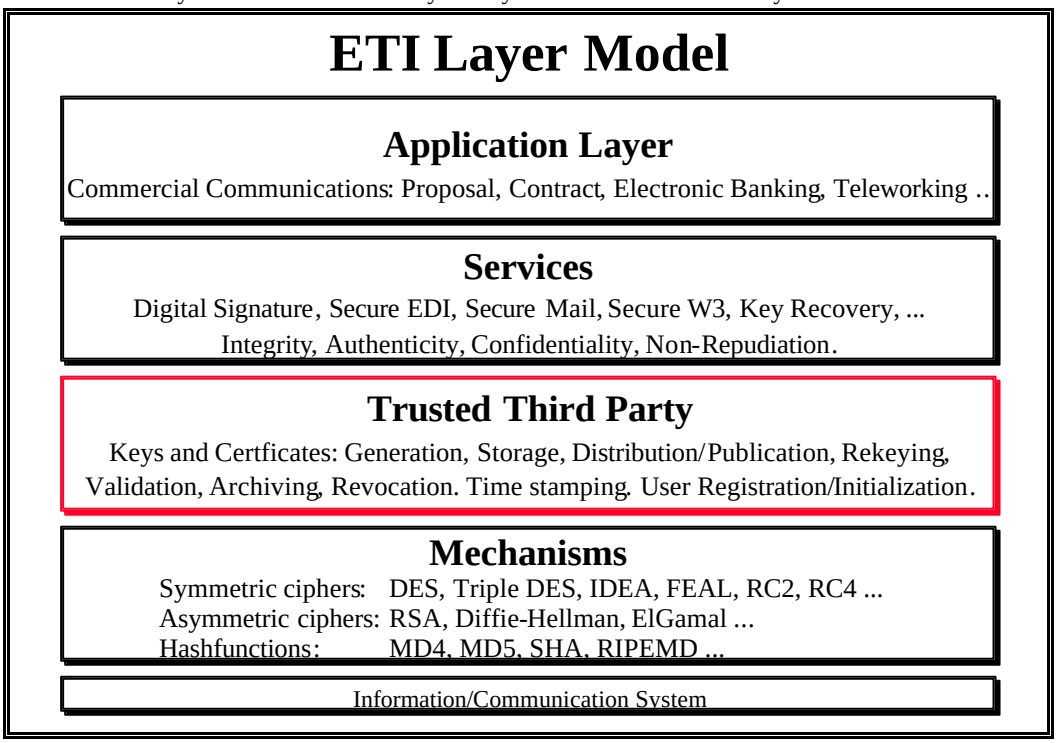


Figure 67: ETI Layer Model

## 3.3 ETI Certificates

### 3.3.1 Certificate Format

All signature certificates will contain, at a minimum:

- The distinguished name of the subject (user or entity),
- the validity period,
- the public key and its parameters,
- key usage field,
- algorithm field,
- one or more policy identifiers,
- the name of the issuing authority, and
- the issuing authority`s digital signature.

| Field | Description and Example |
|---|---|
| Type, Version | X.509. 0 for version 1.<br>1 for version 2.<br>2 for version 3. |
| Serial Number | Unique identifier for each certificate generated by issuer; integer: 08154711 |
| Issuer Name | The name may be a public key or a hashed public key: S07123AK6539BM4R<br>Name of issuer (X.500 "Distinguished Name", a Sequence of RelativeDistinguishedNames that uniquely identify a directory object). |
| Subject's Public Key | Identifier of the subject. Distinguished Name. |
| Type of Hash-Algorithm used | MD4, MD5, SHA, RIPEMD, ... |
| Type of Signature Algorithm used | Identifier for an algorithm as stored and registered.<br>Type of algorithm used to sign the certificate: RSA, ElGamal, ... |
| Parameters for the Algorithm | Any parameter needed. |
| Subject Name or complete Subject itself | Name of subject (X.500 "Distinguished Name"). Text string like: "Andrew Smith"<br>or a pseudonym like: R2D2K2R or a bit string like: "CEC DG XIII/B (Ed.): Green Paper on the Security of Information Systems. DG XIII/B. Brussels April 1994". |
| Validity | Indicates the first and last date of which the certificate is valid respectively. |
| Attribute | 1 Bit, pointing to (optional) attribute certificates. |
| Issuer's Signature | Digital signature of the issuer signing the certificate content. |

Figure 68: Minimum Fields of a ETI Certificate and Description and Examples

A European wide naming scheme, based on X.520, is required to ensure assignment of unique names. The maximum validity periods will be specified in the technical security policy. The public key parameters must be included with the public key to ensure interoperability. The algorithm shall support multiple signatures. A user certificate shall also include one or more policy identifier fields indicating the security policies governing its issuance. Certification authority certificates may include policy constraint fields indicating the set of policies for which it may issue certificates.

The certificate may also contain a non-negative integer field specifying the number of electronic renewals permitted for this certificate.[8]

Attribute formats and the procedures for creation, distribution, and revocation of attribute certificates should primarily reflect local requirements and should only in exceptional cases be determined by the ETI.

[8] The number of electronic renewals permitted is decremented by one upon each renewal, and when it reaches zero, the certificate must be re-issued through the initial certification procedure.

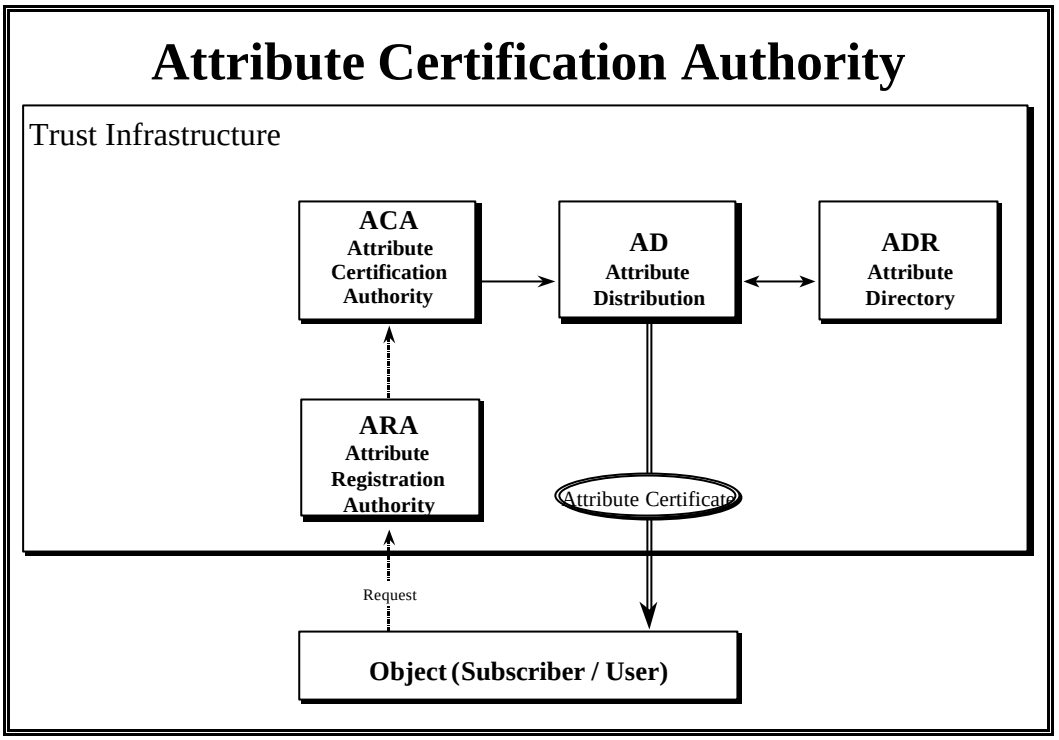


Figure 69: Attribute Certification Authority

The proposed fields for the ETI certificate are shown in Figure 70: Fields of the ETI Certificate: Description and Examples.

| Field | Contents, Functions, Description and Example |
|---|---|
| Type, Version | X.509. 0 for version 1. 1 for version 2. 2 for version 3. |
| Serial Number | Unique identifier for each certificate generated by issuer; integer: 08154711 |
| Issuer Name | The name may be a public key or a hashed public key: S07123AK6539BM4R<br>Name of issuer (X.500 "distinguished name", a Sequence of RelativeDistinguishedNames that uniquely identify a directory object). |
| Issuer Address | |
| Public Key of Subject | Subject or entity's public key information: Algorithm, parameter, key: The entity generates an asymmetric key pair for signatures and for submitting the public key for verification to the registration authority. |
| Type of Hash-Algorithm used | MD4, MD5, SHA, RIPEMD, ... |
| Type of Signature Algorithm used | Identifier for an algorithm as stored and registered.<br>Type of algorithm used to sign the certificate: RSA, ElGamal, ... |
| Parameters for the Algorithm | Any parameter needed. |
| Subject Name or complete Subject itself | Name of subject (X.500 "distinguished name"). Text string like: "Andrew Smith" or a pseudonym like: R2D2K2R or a String like: "CEC DG XIII/B (Ed.): Green Paper on the Security of Information Systems. DG XIII/B. Brussels April 1994".<br>Distinguished name of the authenticated subject or entity. The registration authority processes the entity's application, for validating the entity's credentials and for assigning a distinguished name to the entity. |
| Attribute | 1 Bit, pointing to attribute certificates. |
| Issuer's Signature | Overall digital signature by the issuing certification authority signing the certificate content: The signature of the certification authority binds the public key to the entity's name. |

Figure 70: Fields of the ETI Certificate: Description and Examples

### 3.3.2 Certificate Management

#### 3.3.2.1 Certificate Issuance

**Identification**

The ETI shall issue certificates to properly identified natural or legal persons. Renewal of certificates will be performed electronically where assurance requirements permit.[9]

At the request of their organisation the ETI shall issue certificates to users with appropriate identification and organisational credentials. Certificates will not be issued without management approval, regardless of credentials. The type and number of credentials will be defined by the ETIs policy.

**Keys and algorithms**

The ETI shall only issue certificates signed with strong keys and algorithms. Where certificates contain encryption and signature keys, the algorithm and key length must be sufficient that it is computationally infeasible to:

- forge signatures,
- make undetected alterations to signed files, or
- obtain confidentiality keys exchanges through supported key distribution mechanisms during the expected lifetime of the information.

The ETI shall not generate signature keys for ETI users, although it may be requested by the user.

The ETI will maintain logs of public keys, and will reject requests to issue new certificates which re-use old keys. This ensures uniqueness of private keys.

**Accreditation Criteria**

It is obvious that any organisation wanting to carry out a registration and/or certification function can only be really trustworthy if it is a strictly independent third party. Therefore, for a third party to achieve trust amongst partners and users, a number of criteria should be agreed for any registration authority and certification authority organisation. It should be:

- Independent from any particular commercial, industrial or financial interest: Public authorities, governmental agencies/ministries, political parties, pressure groups, telecom providers, hardware and software manufacturers and suppliers, security experts, standardisation organisations.
- Independent from sectoral business interest, defend the general economic interest of the business community as a whole without any discrimination as opposed to the organisations defending particular interests.
- Be a recognized promoter and facilitator of international trade without any discrimination. Have recognized qualities of integrity and impartiality by the business community, public and governmental administrations and international institutions.
- Be service oriented and involved in education, awareness, guidance to business. Be non-profit oriented organisations with a proven capacity to operate on a self-funding basis on long term. Have a proven historical stability to ensure future survival independently of changing economical and political situations.
- Have a proven capacity to operate at international level through international structures as well as national and regional (close to business) structures. Enjoy a recognized international standing and world-wide reputation. Enjoy an international arbitration conciliation capability, e.g. through arbitration conciliation court. Be accepted as a moral authority by the international business community. Have the capability to assume liability connected with registration and certification activities through an international guaranteeing scheme to further reinforce trust.
- Have actual registration and certification expertise – even though in paper and ink world - in related areas.

[9] The assurance associated with the binding for electronically renewed certificates is lower than for certificates issued by an organisational registration authority. The assurance falls with each electronic renewal.

#### 3.3.2.2 Distribution and Archiving

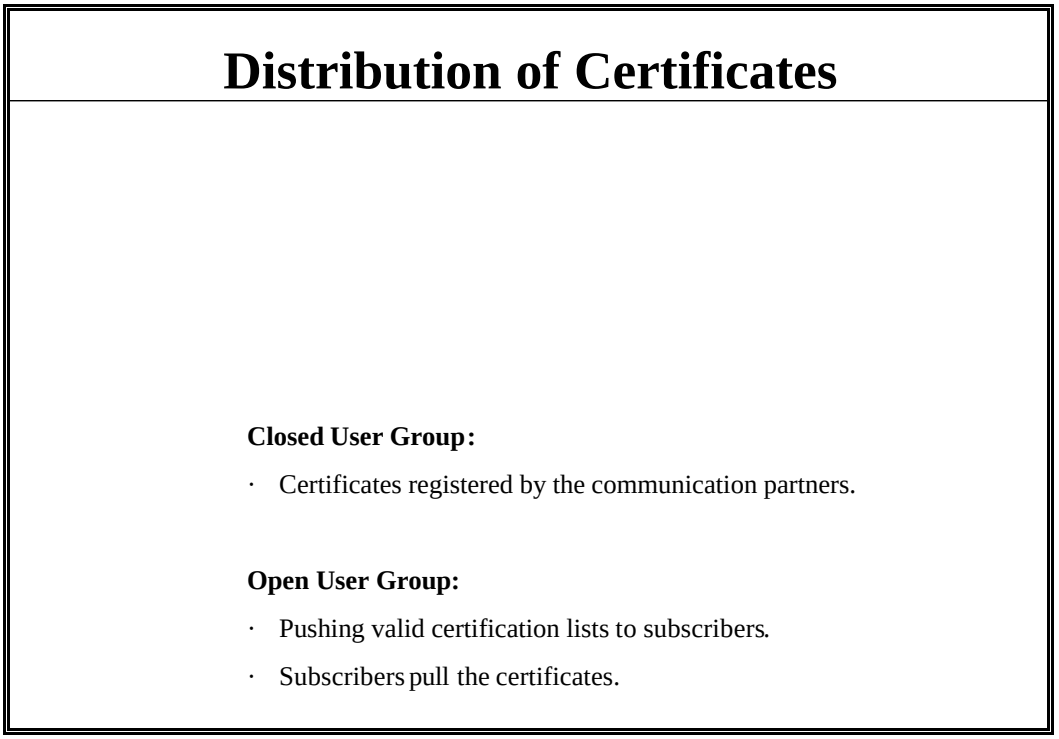


Figure 71: Distribution of Certificates

**Creation and Posting**

The creating and posting of certificates is a service of the ETI that is required by the user. This service permits a user to:

- obtain a public key pair,[10]
- obtain the corresponding public key certificate,
- have the certificate bound to their identity with a specified level of assurance; and
- make the certificate available to any user who wishes to obtain it.

Binding the certificate to its owners identity requires personal presentation of credentials to a ETI representative. The type and number of credentials required may vary; this should be reflected in the certificate[11]. Subsequent generation of certificates should be performed remotely, in an automated fashion.

**Distribution**

At a minimum, the ETI will support distribution of certificates through directory services. The directory server(s) will chain to the directory servers of other trust infrastructures. Additional distribution services may be offered to meet local requirements.

#### 3.3.2.3 Reception, Interpretation, and Verification of Certificates

Users must be able to determine if they can trust a signature for a particular application. This requires verification of certificates and certificate chains, and evaluating the assurance associated with the binding of identity to a certificate.

**Reception and Interpretation**

Users must be able to obtain certificates for a particular user, parse it correctly, and determine the current status of the certificate. This process must be entirely automated.

| | |
|---|---|
| Obtaining: | Users must be able to obtain certificates for a named entity in an automated fashion. |
| Format: | The certificate format must be known, so that it can be interpreted in an automated fashion. |
| Algorithm: | The certificates must be signed by an appropriate algorithm so that users can verify the certificate chain and verify the binding of a user and certificate. |
| Revocation: | Certificate revocation information must be available to ensure compromised certificates are not exploited. |

**Verification**

[10] In most cases, users should generate their own public-key pairs. This provides the strongest binding for non-repudiation purposes. In special cases, the ETI may generate key pairs for users.

[11] Sometimes there are specified three levels of assurance reflecting the type and number of credentials.

A single certificate is often not adequate to provide assurance that the issuing authority is trustworthy and has the authority to issue the certificate. The issuing authority needs to be checked via a certificate of another authority.

To verify the certificate chains, the user must be able to verify a series of digitally signed messages (certificates). This requires that:

- The keys are associated with an appropriate algorithm,
- the messages are hashed with an appropriate algorithm, and
- the message formats are known to the receiver.

Finally the user must be able to determine if the policy under which a certificate was issued is acceptable for a particular application.

#### 3.3.2.4 Revocation and Re-Certification

**Revocation**

Users must be able to revoke the binding between their identity and a public key when the private key has been lost or compromised, or the status of the user changes.[12] This information must be available in a timely fashion.

The user entity shall be able to verify each certificate by requesting the certification authority on-line. The validity field is only useful for lower levels of trust, where you do not want to request an on-line verification, but trust the certificate as well.

The ETI shall issue certificate revocation lists conforming to the X.509 version 2 CRL format for the lower levels of trust.

The ETI shall provide mechanisms to verify the validity of certificates. CRLs will be available from on-line directories. CRLs must be posted at regular intervals so that users can ensure that they have the most current information.

The ETI must also support push-down mechanisms for key compromise and other critical events. Such mechanisms must be available to communities with special security requirements.

The ETI may also include a trusted directory mechanism. Such a mechanism would provide the most current information to critical applications. Trusted directories could be maintained by the ETI, or may be offered by vendors as value-added services.

In the certification authority policy it must be specified who can revoke a certificate. In order to make false revocations unlikely it is perhaps useful to specify for each certificate a list of entities authorised to revoke a certificate.

Furthermore it is useful to include the date and time of known or suspected compromise which need not coincide with the time of revocation. The party who requested the revocation (initiator) need not be the user itself, it may, for instance, be the employer or a trusted third party.

**Re-Certification**

For certificates with a long lifetime, the security of binding between a public key and the subject becomes even more problematic. Therefore the old signature shall be stripped off by the certification authority and the certificate shall be newly signed with actual certified algorithms.

**Registry of Mechanisms**

Nevertheless: In the case of re-certification these old, no longer used algorithms have to be archived permanently for access of users for verification purposes and must be identified for the verification process of older certificates in a registry of certified algorithms.

There must be an agency evaluating and certifying proposed algorithms. These algorithms are registered in an official way by the registry (authority). Only registered algorithms for hashing and signing are possible to use. Those registered algorithms for the type of hash-algorithm and the type of signature algorithm get an identification number by the registry. The identification of the used algorithms is shown in the certificate.

#### 3.3.2.5 Time stamps

The use of time stamps is an extremely important function in the ETI, if not essential, for maintaining the validity of documents over many years, in certification, and non-repudiation evidence recording services.

## 3.4 ETI Names

### 3.4.1 Functionality

#### 3.4.1.1 Name Semantics

**Common names**

[12] For example, the identity stated in the certificate may imply organisational relationships. The relationships change (e.g., job change or retirement) and the certificate becomes invalid.

It is recommended that personal names are to be shown in certificates to denote the subject. The recommendation of the World Electronic Messaging Association (WEMA) for an agreement on a simple and unified naming standard for personal names in order to simplify global messaging should be supported to equalise the minor differences in the naming of persons in the different Member States. Measures to assure the uniqueness of common names, if necessary at all, should be realised in the directory.

**Directory names**

For a European Trust Infrastructure the naming conventions as specified in X.520 should be used as the basis for naming. Also the use of alias names should be possible.

**O/R names**

It is not recommended to use O/R names in the main part of a certificate, because the undefined sequence of attribute/value combinations would make the evaluation of these names too complex for the respective application programs. It is however recommended to allow for using them in the certificate extensions where they can be used to reference to objects that have X.400 addresses.

**Absolute names**

When using X.509 certificates and the X.520 naming conventions for a European Trust Infrastructure, absolute names should be used. However when structuring the Directory Information Tree measures should be taken to avoid that Distinguished Names become too long.

**Relative names**

The use of relative names in a European Trust Infrastructure should not be excluded and they should be used whenever an application has an advantage from their use.

#### 3.4.1.2 Name Syntax

**Numerical names**

For a European Trust Infrastructure numerical names should not be used to denote issuers and subjects in the basic certificate fields of identity certificates. Their use in the certificate extensions should not be precluded, if there are applications that can make use of such a feature. If there are application sectors where it might be sensible to use numerical names to identify persons, clear naming conventions should be established and the transformation rules into common or directory names should be clearly defined. The technical system should then facilitate the use of numerical names by offering an adequate user interface. In case authorisation certificates will be used it has to be determined by the respective application whether a public key (interpreted as a numerical name) is sufficient to denote issuers and subjects in the certificate or whether other - human understandable - name forms would be preferable from a users point of view.

#### 3.4.1.3 Name Validity

**Global names**

Although there might be problems with the establishment of a directory with „global“ names, the use of such a directory for a European Trust Infrastructure is strongly recommended. Particularly it would allow to decentralise the name management in a simple way to the Member States. In addition of being used as a repository for names and certificates it is capable to provide a lot of additional information which would allow to support the co-operation of the Member States of the European Union.

**Local names**

It is recommended not to preclude the use of local names in a European Trust Infrastructure, especially in cases where it is clear that there is no requirement for names to be global. This would contribute to user-friendliness. It would also facilitate the organisation of the Directory.

### 3.4.2 Flexibility

In spite of the problems with hierarchical naming systems it is recommended that X.500 is being followed for a European Trust Infrastructure. However it is strongly recommended to follow the SDSI/SPKI activities (which are currently still in a formative stage) very carefully in order to assess whether this approach that builds on local names, might be worth to be considered for a European Trust Structure. In applications where names have only a local validity, local names should be permitted.

The possibility of specifying alternative names in certificate extensions should be supported.

### 3.4.3 Ease of use

It is recommended for a European Trust Infrastructure to use names according to the X.520 standards. Special attention should be paid in implementing a user interface which allows the user to handle the sometimes cumbersome X.520 names.

The names themselves should only contain the minimum of attribute/value pairs that are necessary to clearly identify an object.

### 3.4.4 Support of user trust

For a European Trust Infrastructure it is recommended to use human readable names in certificates to provide the user with information that can support his trust into the certificates and thereby into the infrastructure in general. However names should not be used to derive security properties, therefore it is not recommended to implement trust through the naming hierarchy (like in PEM), one of the reasons being the reduction of flexibility in structuring the directory. To restrict the length and the variations of certification paths the feature of the X.509 v3 extension, which allows to constrain certification paths should be used.

### 3.4.5 Adherence to Standards

For a European Trust Infrastructure it is recommended to follow the naming conventions as defined in the X.500 series of standards. The main reasons being that X.500 has matured since its publication, experience with its application is available and products are on the market that can be implemented.

### 3.4.6 Legal Aspects

A European Trust Infrastructure has to take into consideration the interest of persons and organisations (companies etc.) in protecting their privacy. It is recommended that names in certificates should have only the minimum information about the subject and the issuer that is necessary for the purpose. The use of alias names should be possible.

## 3.5 Interoperability with other Trust Infrastructures

The ETI shall ensure trustworthy interoperability of entities (certification authorities, organisational registration authorities, and ETI clients) by selecting appropriate standards to support ETI transactions. Certain algorithms and information formats must be supported within the ETI due to their status as formal or de facto standards.

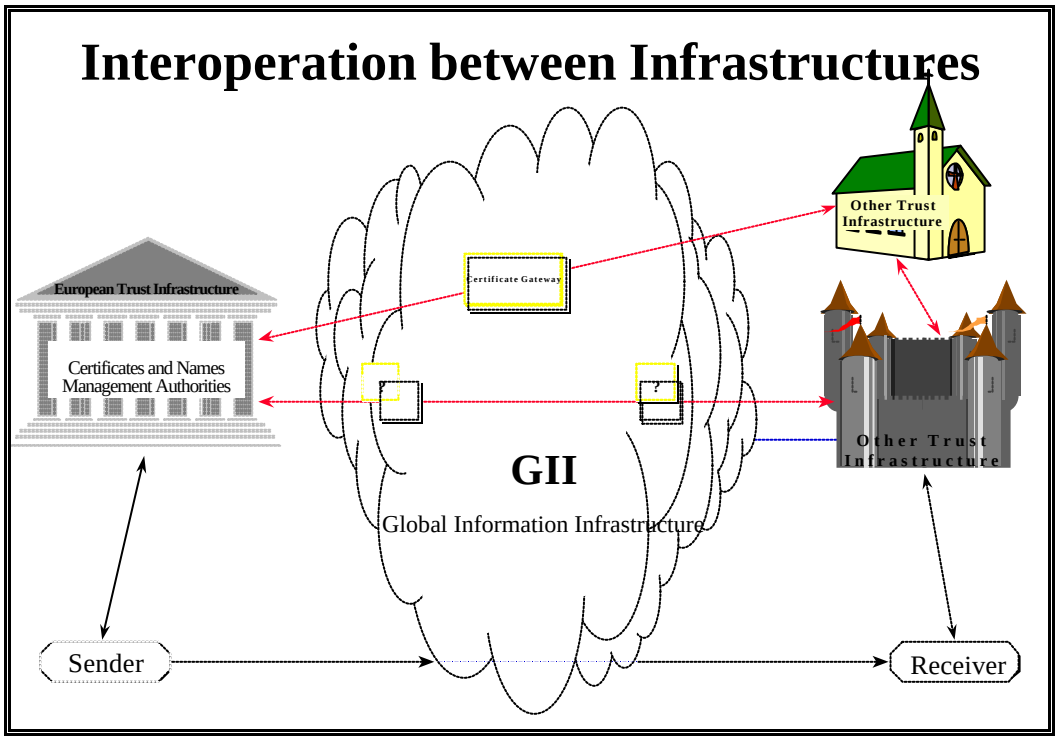


Figure 72: Interoperation between Infrastructures

The ETI shall have clearly defined policies regarding non-ETIs, and will only cross-certify with trust infrastructures that meet or exceed those policies. Policies must be reviewed and a third party audit of the ETI confirming adherence to the written policy must be performed before cross certification.

The ETI shall implement policies, procedures and central servers as gateways to other trust infrastructures to process and verify trustworthy certificates of users of other trust infrastructures. The gateway needs to be certified and send the interpreted other certificates certified. This might be a commercial available service.

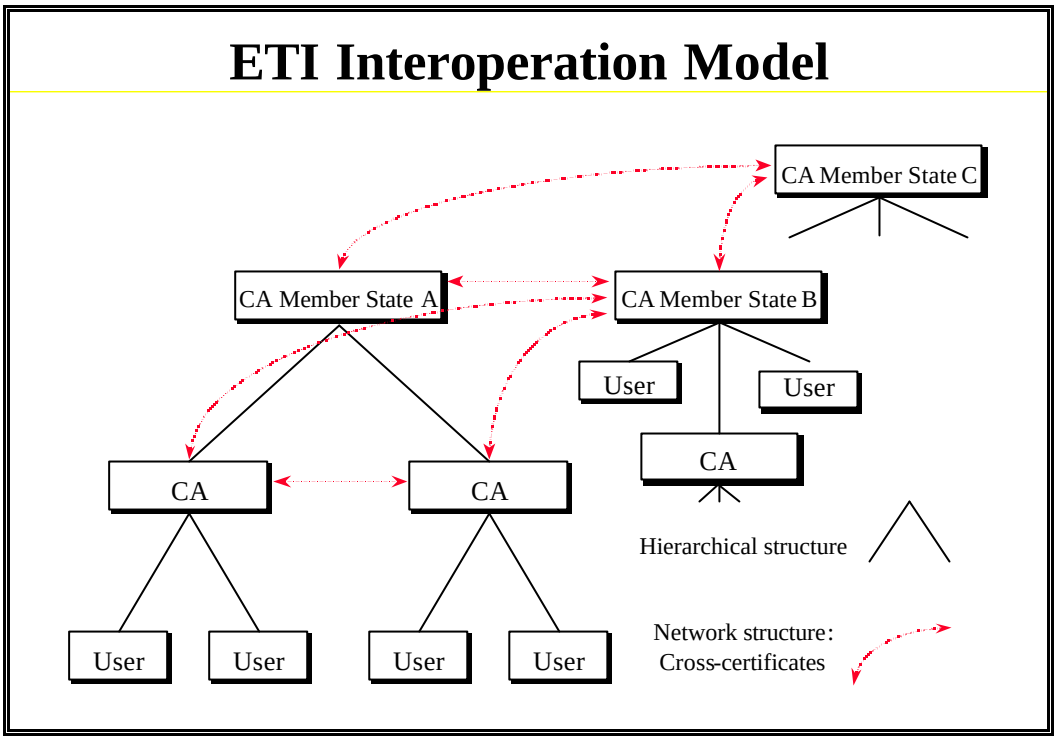


Figure 73: ETI Interoperation Model

**Interoperability of names**

Although there are mechanisms available to deal with different naming conventions in a certificate infrastructure, they should be avoided in an European Trust Infrastructure. X.520 naming concepts should be used, they are applicable to all cultures and sectors therefore there is no need for separate regional or sectorial approaches. Naming conventions that have already been established and that differ generically from the X.520 concept should be accommodated by implementing appropriate translation mechanisms, on the long run however it should be analysed whether they can be converted to X.520 conventions. The translation concept should be applied to the co-operation with systems outside of the European Union that might use other standards. The extensions in X.509v3 which allow to use names defined in other standards than X.520 should be supported.

# 4 Glossary, Abbreviations and Acronyms

## 4.1 Glossary of Terms

| | |
|---|---|
| Abstract Syntax Notation 1 (ASN.1) | An abstract notation for structuring complex data objects. |
| accountability | The property that ensures that the actions of an entity may be traced uniquely to the entity. |
| accreditation | Recognising an entity or person to perform a specific action; certification authorities accredit organisational registration authorities to act as their intermediary.<br>⇒ organisational registration authority. |
| alias | An alternative name for an entity, provided by the use of alias entries. |
| asymmetric cryptography | ⇒ public-key cryptography. |
| attestation | Declaration of an authority that a person is the one who it claims to be.<br>⇒ authentication, identification. |
| attribute | Additional quality or property of a subject or entity. The most commonly considered attributes are trust and authority.<br>⇒ authority, trust |
| attribute certificate | A certificate signed by the attribute certification authority. It includes a reference to the entity's distinguished name, arbitrary attributes, optionally a validity time frame, and technical information. It can only exist in combination with a public key certificate because the public key associated with the public key certificate which must be used to prove that an entity is the authentic subject of the attribute certificate. It is sealed in the associated public key certificate of the certification authority.<br>⇒ certificate. |
| attribute certification | Attribute signing process. |
| attribute certification authority | An entity trusted by one or more entities (users) to create, assign and issue attribute certificates relating to public verification key certificates - by binding the public key and an entity - may be an individual - by a name. |
| attribute directory | Storage of attributes for distribution and archiving. |
| attribute registration | Registration process of attributes. |
| attribute registration authority | Authority registrating attributes. |
| authentication | Verification of an entity's identity. The property of knowing that the claimed sender is in fact the actual sender. Provision of the assurance of the claimed identity of an entity. |
| authority | Authorised entity that has been given official power to act. |
| binding | The process of confirming that data items belong together. E.g. linking a public key with an entity represented by a name, the owner of this key, or an attested person. |
| black list | Revoked entities list, contains the names of entities who have been blacklisted, or whose keys have been revoked. Unexpired certificates for entities in the black list shall be revoked (included in the certificate revocation list). No new certificates are created for these entities until they are no longer on the revoked entities list.<br>⇒ certificate revocation list. |
| certificate | Structured message which delegates in a trustworthy way an attribute of some form to a public key.<br>Digitally signed object certifying a public verification key of an entity together with some other information, rendered unforgeable (signed) by encipherment with the private key of the certification authority which issued the certificate for authentication of a digital signature. |

| | |
|---|---|
| | ⇒ certification authority. |
| certificate cancellation notice | ⇒ certificate revocation notice. |
| certificate holder | An entity that is named as the subject of a certificate issued by a certification authority. |
| certificate management service | A system that signs and manages certificates and certificate revocation lists. |
| | ⇒ certificate revocation list. |
| certificate practice statement | A statement of the practices which a certification authority employs in issuing certificates. |
| certificate policy | A named set of rules that indicates the applicability of a certificate to a particular community and/or class of application with common security requirements. For example, a particular certificate policy might indicate applicability of a type of certificate to the authentication of electronic data interchange transactions for the trading of goods within a given price range. |
| certificate result certificate | The verifier of a certificate of another trust infrastructure can cache the result of the verification process with its possibly own expiration time. He generates a certificate result certificate by signing that certificate result cache entry. |
| certificate revocation | Cancellation of a certificate on account of a special reason (e.g. lost or compromised key). |
| certificate revocation list (CRL) | A list of revoked certificates issued by a certification authority. |
| certificate revocation notice | Message to an authority revoking a valid certificate. |
| certificate serial number | Serial number of a certificate assigned by the issuer to access it on request. |
| certificate type | An indication of the purpose of a certificate. |
| | ⇒ identity certificate, attribute certificate. |
| certificate-using system | An implementation of those functions used by a certificate user. |
| certification | Process by which a third party gives assurance that all or part of a data processing system conforms to security requirements.The act of issuing a certificate. |
| certification authority | A trusted entity that issues public verification key certificates to end entities and other certification authorities - by binding the public key and an entity - may be an individual user - by name. |
| | An authority trusted by one or more entities (users) to create and assign certificates. Optionally the certification authority may create the users keys. Certification authorities issue certificate revocation lists periodically, and post certificates and certificate revocation lists to a repository. |
| certification management authority | A general term embracing all certification authority types, including policy approval authority and policy certification authority. |
| certification path | An ordered sequence of certificates, leading from a certificate whose public key is known by a user, to a certificate whose public key is to be validated by the user. |
| certification practice statement | A statement of the practices which a certification authority employs in issuing certificates. |
| ciphertext | Data, that cannot be interpreted without the use of cryptographic techniques. |
| cleartext | Data that can be interpreted without the use of cryptographic techniques. |
| | ⇒ plaintext. |
| client function | A function that uses the PKI to obtain certificates and validate certificates and signatures. Client functions are present in certification authorities and end entities. Client functions may also be present in entities that are not certificate holders. That is, an entity (system or user) that verifies signatures and validation paths is a client, even if |

| | |
|---|---|
| | it does not hold a certificate itself. |
| compromised key list | All the keys stored in a directory, which are no longer valid because they are suspected to be lost or used unauthorised or of other reasons. |
| confidentiality | The property of communicating such that the intended recipients knows what was being sent, but unintended parties cannot determine what was sent.<br>The prevention of the unauthorised disclosure of information |
| | co-signature The signature of a document by two entities.<br>⇒ dual signature. |
| credentials | Data that is presented or transferred to establish the claimed identity of an entity.<br>Such credentials may be the passport of the entity, the personal signature, biometric characteristics etc. |
| CRL distribution point | A directory entry or other distribution source for CRLs. A CRL distributed through a CRL distribution point may contain revocation entries for only a subset of the full set of certificates issued by one certification authority or may contain revocation entries for multiple certification authorities.. |
| cross certificate | Certificate in a network model where two certification authorities certify each other. The cross certificate pair construct contains two certificates: One forward certificate and one reverse certificate. The subject of the forward certificate is the issuer of the reverse certificate and vice versa. |
| cross certificate pair | A pair of certificates for two certification authorities certifying each other in a network certification model. The certificate receiving entity verifies this pair of certificates by requesting these two certification authorities only. |
| cross certification | In a network model a process in which two certification authorities securely exchange keying information so that each can effectively certify the trustworthiness of the others key by cross-certifying each other: Generating a cross-certificate certifying the other certificate authority. |
| cryptanalysis | The steps and operations performed in converting encrypted messages into plaintext without initial knowledge of the key employed in the encryption algorithm.<br>⇒ encryption, key. |
| crypto system | ⇒ cryptographic system. |
| cryptographic system | A system providing confidentiality for the transmission of data.<br>In cryptography, it comprises five components: A message space, a ciphertext space, a key space, a family of enciphering transformations, and a family of deciphering transformations.<br>In cryptology, the documents, devices, equipment and associated techniques that are used as a unit to provide a single means of encryption (enciphering or encoding).<br>⇒ cryptography, encryption. |
| cryptography | The science or discipline that embodies the principles, means, and methods for the transformation of data in order to hide their semantic content, prevent their unauthorised use, or prevent their undetected modification rendering plaintext unintelligible and for converting encrypted messages into intelligible form. |
| cryptology | The field that encompasses both cryptography and cryptanalysis.<br>⇒ cryptography, cryptanalysis. |
| data integrity | The property that data have not been altered or destroyed in an unauthorised manner. |
| data origin authentication | The corroboration that the source of data received is as |

| | |
|---|---|
| | claimed. |
| data storage | A means for storing information from which data are submitted for delivery, or into which data are put by the delivery authority. |
| decipherment | The reversal of a corresponding (reversible) encipherment. |
| decryption | The process of obtaining, from a ciphertext, the original corresponding data. |
| delivery authority | An authority trusted by the sender to deliver the data from the sender to the receiver, and to provide the sender with evidence on the submission and transport of data upon request. |
| delta certificate revocation list | A partial certificate revocation list indicating only changes since a prior certificate revocation list issue. |
| delta certificate revocation list indicator | A critical certificate revocation list extension that identifies a delta certificate revocation list. The use of delta certificate revocation lists can significantly improve processing time for applications which store revocation information in a format other than the certificate revocation list structure. This allows changes to be added to the local database while ignoring unchanged information that is already in the local database. |
| digital signature | Encrypted data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit or message to verify the source (identity of the signatory) and integrity of the data unit and protect against forgery e.g. by the recipient. For the signature public key systems are used.<br>$\Rightarrow$ hash function, public key system. |
| digital signature algorithm | The algorithm used to digitally sign a document. |
| directory | A collection of open systems co-operating to provide some services like storage of data items and access to these items. |
| directory information base | The complete set of information to which the directory provides access, and which includes all of the pieces of information which can be read or manipulated using the operations of this directory. |
| directory information tree | The directory information base considered as a tree, whose vertices (other than the root) are the directory entries. |
| directory name | A construct that singles out a particular object from other objects. A name shall be unambiguous (that is, denote just one object), however it need not to be unique (that is, be the only name which unambiguously denotes the object). |
| directory service | A distributed database service capable of storing information, such as certificates and certificate revocation lists, in various nodes or servers distributed across a network (directory) user. The end user of the directory, i.e. the entity or person which accesses the directory.user. |
| distinguished encoding rules | Rules for encoding ASN.1 objects which give a consistent encoding for each ASN.1 value. Implementations conforming to this specification shall encode ASN.1 objects using the distinguished encoding rules. |
| distinguished name (of an entry) | The name of an entry which is formed from the sequence of the RDNs of the entry and each of its superior entries. Every entity entry, alias entry and subentry has precisely one distinguished name. It is a unique string supposedly with an individual or other named entity in the world. |
| distinguishing identifier | Information which unambiguously distinguishes an entity in the authentication process. |
| dual legality | A legal request from a foreign agency must satisfy legal access conditions in both the requesting country and the country being asked. |
| elliptic curve digital signature algorithm | A digital signature algorithm that is an analogue of DSA using elliptic curve mathematics and specified in ANSI draft standard X9.62 [X9.62]. |
| encipherment | The cryptographic transformation of data to produce ci- |

| | |
|---|---|
| | phertext. |
| encryption | A mechanism commonly used to provide confidentiality during storage and transmission of data to an unintelligible form in such a way that the original data either cannot be obtained (one-way encryption) or cannot be obtained without using the inverse decryption process (two-way encryption). The result of encryption is ciphertext.<br>⇒ public-key cryptography, symmetric cryptography, irreversible cryptography, decryption. |
| end entity | A subject and/or end user system that is the subject of a certificate. |
| Entity | A things existence - contrasted with its attributes, qualities, relations, etc. I.e. a user, a subject, person, process, communication partner etc. |
| entity authentication | The corroboration that an entity is the one claimed. |
| entry name | A construct that singles out a particular entry from all other entries.<br>⇒ name. |
| evidence | Information that either by itself or when used in conjunction with other information is used to establish proof about an event action.<br>Note: Evidence does not necessarily prove truth or existence of something but contributes to establish proof.<br>⇒ proof . |
| evidence generator | An entity that produces evidence - e.g. for a security function like non-repudiation. |
| evidence requester | An entity requesting evidence to be generated either by another entity or by a trusted third party. |
| evidence subject | An entity whose involvement in an event or action is established by evidence. |
| evidence user | An end entity that uses evidence. |
| evidence verifier | An entity that verifies evidence. |
| expiration date | Date of the expiration of a certificate, end of the validity period. |
| extension | Information about a subject, added in a separate document. |
| government access to keys | A capability that allows authorised persons or agencies, under certain prescribed conditions, to read the keys used with the help of information supplied by one or more trusted parties storing escrowed parts of the used keys. Also known as mandatory key escrow, where the keys must be escrowed with the government, e.g. Clipper. |
| hash code | The string of bits which is the output of a hash function. |
| hash function | A one-way function which maps strings of bits to fixed-length strings of hash function. A one-way function which maps strings of bits to fixed-length strings of bits, satisfying the following two properties:<br>• It is computationally infeasible to find to a given output an input which maps to this output,<br>• it is computationally infeasible to find for a given input a second input which maps to the same output.<br>Hash functions are used for example to compress the text of a document before digitally signing it.<br>⇒ digital signature. |
| identification | Process of proving that a claimed entity (sender) is in fact the actual entity (sender). |
| identity | Distinguishing character or personality of an individual or entity. Not necessary a name but a role: Some characteristics like responsibility in a company (a manager representing a company), union (a secretary representing a union), party (chairman) or governmental agency (president).<br>In private life it may be only a name representing an individual person. |

| | |
|---|---|
| | ⇒ role. |
| identity certificate | Certificate which binds the name of an entity to a public key. It`s putative meaning is to delegate all the attributes of the named entity to the public key.<br>⇒ attribute, certificate. |
| imprint | A string of bits, either the hash-code of a message or the message itself. |
| information | The meanings assigned to data by the agreed conventions used in its representation. Any communication or reception of knowledge such as facts, opinions in numerical, graphical or narrative form maintained in any medium. |
| integrity | The property of ensuring that data is transmitted from source to destination without undetected alteration. |
| irreversible encryption | Encryption that produces ciphertext from which the original data cannot be reproduced.<br>⇒ encryption, one-way encryption. |
| irreversible encipherment | ⇒ irreversible encryption. |
| issuer | Entity generating and distributing a certificate. |
| issuer name | Identifying name of an issuer. |
| issuer unique identifier | Issuer distinguishing data item. |
| item | Single article or unit in a list, sometimes a detail or paragraph. |
| key | A symbol or sequence of symbols (variable-length bit string) that controls the operations of encipherment and decipherment.<br>A sequence of symbols that controls the operations of encryption and decryption.<br>⇒ encipherment, decipherment. |
| key distribution | A set of procedures to provide key management information objects to authorised entities in a trustworthy way. |
| key distribution center | Authority storing and distributing keys in a trustworthy way. |
| key escrow | A capability that allows authorised persons or agencies, under certain prescribed conditions, to read the keys used with the help of information supplied by one or more trusted parties storing escrowed parts of the used keys. |
| key generation | The process concerned with the generating keys. |
| key management | The process concerned with the administration, generation, registration, certification, deregistration, distribution, installation, storage, archiving, revocation, derivation and destruction of cryptographic keying material and related information like initialisation vectors in accordance with a security policy. |
| key management infrastructure | A group of co-operating trusted third parties.<br>⇒ certification authority. |
| key pair | A set of a public and a private key which belong together. |
| key recovery | A capability that allows authorised persons or agencies, under certain prescribed conditions, to recover keys used with the help of information supplied by one or more trusted parties storing escrowed parts of the used keys.<br>⇒ government access to keys. |
| key revocation | Notification that a (public) cryptographic key is no longer valid. |
| keyholder | An entity who holds a given private key. This key may be indicated by either its corresponding public key or a secure hash of that public key. Every key has a keyholder, by definition. |
| law enforcement agency | An organisation established by national law, responsible for law enforcement. |
| lawful authorisation | Permission granted to a law enforcement agency under certain conditions to intercept specified telecommunica- |

| | |
|---|---|
| | tions. Typically this refers to a warrant or an order issued by body authorised by law. |
| lawful interception | The action (based on the law), performed by a network operator or service provider, of making available certain information and providing that information to a law enforcement monitoring facility. |
| leaf certification authority | The last certification authority in a trust infrastructure; those certification authorities have no subordinate certification authorities. All trust in the infrastructure propagates from this certification authority. |
| local registration agent | ⇒ organisational registration agent. |
| local registration authority | ⇒ organisational registration authority. |
| message | String of bits of any length. |
| message authentication code | A data item derived from a message using symmetric cryptographic techniques and a secret key. It is used to check the integrity and origin of a message by an entity holding the secret key. |
| message digest | The fixed size result of hashing a message. |
| name | Word or a combination of words by which a person or thing is regularly known. Names are identifiers which distinguish entities from one another and which can be used by different people to refer to the same entity. (Logical) pointer (like an address) to a (physical) entity.<br>A construct that singles out a particular object from other objects. |
| naming authority | An authority responsible for the allocation of names in some region of the directory information tree. |
| non-repudiation | The property of a receiver being able to prove that the sender of some data did in fact send the data even though the sender might later desire to deny ever having sent that data. |
| notarisation | The registration of data with a trusted third party that allows the assurance of the accuracy of the data's characteristics such as content, origin, time and delivery. |
| notary | A trusted third party which provides assurance about the properties of data communicated between two or more entities, such as the data's integrity, origin, time or destination. |
| non-repudiation of delivery token | Data item which allows the originator to establish non-repudiation of delivery for a message. |
| non-repudiation of origin token | Data item which allows recipients to establish non-repudiation of origin for a message. |
| non-repudiation of submission token | Data item which allows either the originator (sender) or the delivery authority to establish non-repudiation of submission for a message having been submitted for transmission. |
| non-repudiation of transport token | Data item which allows either the originator or the delivery authority to establish non-repudiation of transport for a message. |
| object | A passive entity. An entity to which access is controlled.<br>⇒ entity, subject. |
| observer | An entity observing the activities of the evidence subject. It is trusted to record correctly the events or actions that occur, and to generate evidence on that records. |
| one-way encryption | ⇒ irreversible encryption. |
| one-way function | A (mathematical) function f which is easy to compute, but which for a general value y, it is computationally difficult to find a value x in the domain such that f(x) = y. There may be a few values y for which finding x is not computationally difficult.<br>⇒ hash function. |
| organisational registration agent | An agent that acts for an organisational registration authority. |

| | |
|---|---|
| | ⇒ organisational registration authority. |
| organisational registration authority | An entity that acts as an intermediary between the certification authority and a prospective user. |
| originator | The entity that sends a message to the recipient or makes available a message for which non-repudiation services are to be provided. |
| Originator/Recipient name: | Name of a user of a message handling system (originator or receiver) which may comprise a directory name, an O/R address, or both. |
| period of validity | Interval of time during which an item (e.g. a certificate) is valid.<br>⇒ validity period. |
| plaintext | Data, the semantic content of which is available without using cryptographic techniques. |
| policy approving authority | Authority approving the overall policy for other authorities. |
| policy certification authority | Authority approving the policy for certification authorities. |
| policy management authority | Policy issuing authority. The certification authorities issue certificates under these specified certificate issuance policies. |
| policy mapping | Recognising that, when a certification authority in one domain certifies a certification authority in another domain, a particular certificate policy in the second domain may be considered by the authority of the first domain to be equivalent (but not necessarily identical in all respects) to a particular certificate policy in the first domain. |
| principal | Person directly responsible - for whom another acts as an agent.<br>Entity (e.g. person, processor, process, device like a printer, ...) which supplies a service or requests action in a distributed computer system. |
| private key | That key of the entity's asymmetric key pair which is known and usable only by that entity (owner) for decryption. In the case of an asymmetric signature system, the private key and the associated algorithms define the signature transformation. |
| privilege attribute | Attribute associated with a security subject that, when matched against control attributes of a security object, is used to grant or deny access to that security object. |
| privilege attribute certificate | A set of privilege attributes issued by a security authority or TTP, that is protected by integrity and data origin authentication, and includes an indication of a time period of validity. |
| proof | The corroboration that evidence is valid in accordance with the non-repudiation policy in force.<br>Note: Proof is evidence that serves to prove truth or existence of something. |
| public key | That key of an entity's asymmetric key pair which is publicly known and intended for use by any entity for encrypted communication with the owner of the corresponding private key.<br>In the case of an asymmetric signature system, the public key and the associated algorithms define the verification transformation. Signed object for verification of the digital signature of an entity. |
| public key certificate | A digitally signed data structure that binds the distinguishing identifier of an entity (certificate holder, subject) to the public key of this entity and which indicates the validity of the corresponding private key.<br>⇒ certificate. |
| public key cryptography | Cryptography in which a public key and a corresponding private key are used for encryption and decryption. One of |

| | |
|---|---|
| | the applications of public key cryptography is the digital signature.<br>⇒ digital signature. |
| public key cryptosystem | Both parties of a system receive the public key of the other party.<br>⇒ public key distribution system. |
| public key distribution system | A system where the sending party only receives the public key of the receiving party.<br>⇒ public key cryptosystem. |
| public key infrastructure | Supporting infrastructure, including non-technical aspects, for the management of public keys. |
| public key infrastructure client | ⇒ client. |
| public verification key | That key of an entity's asymmetric key pair which is publicly known and intended for use by an asymmetric signature system: The public key and the associated algorithms define the verification transformation.<br>⇒ public key, verification. |
| public verification key certificate | Certificate binding the public verification key to an identity. |
| receiver | Data accepting entity. |
| recipient | The entity that gets (receives or fetches) a message for which non-repudiation services are to be provided. |
| reconfirmation | Confirming a fact again. |
| record | Information about a particular certification authority that may be of some help to a certificate user in evaluation the suitability of a certificate issued by that certification authority. |
| registration | Recording an entity after having verified some offered credentials of the entity. |
| registration authority | An (optional) system to which a certification authority delegates certain management functions. E.g. registrating an entity as a user of the system. |
| rekey | Extension of the validity period. The maximum validity period is limited. |
| relative distinguished name | A set of one or more attribute type and value pairs, each of which matches a distinct distinguished attribute value of the entry. |
| repository | A system or collection of distributed systems (database service) that store certificates and certificate revocation lists and serves as a means of distributing these certificates and certificate revocation lists to end entities, allowing unauthenticated information retrieval. Repositories include, but are not limited to, directory services. |
| repudiation | Denial by one of the parties involved in a communication of having participated in all or part of the communication. |
| revocation | Process of invalidation of a certificate in the case there is an incident like a lost or compromised key or changes in the purpose for which the key was being used (change of the company). |
| revocation certificate | In the case of revocation of a certificate, the revocation is published. Several revoked certificates are published together building a list called certificate revocation list. To be trusted, such a list has to be signed digitally – and accompanied by a certificate binding the relating public verification key to the name of the authority publishing the certificate revocation lists. |
| revocation list | List of revoked certificates. |
| revocation time | Date and time of the revocation. |
| role | Some characteristics like responsibility in a company (a manager representing a company), union (a secretary representing a union), party (chairman) or governmental agency (president). |

| | |
|---|---|
| | In private life it may be only a name - representing an individual person. |
| | root certification authority The first certification authority in a trust infrastructure. All trust in the infrastructure propagates from this certification authority. |
| RSA | Acronym formed from the names of the researchers R. L. Rivest, A. Shamir and L. M. Adleman. RSA is a public-key signature algorithm developed by these researchers specified by PKCS 1. As a reversible public-key algorithm, it may also be used for encryption. |
| secret key | A key used with symmetric cryptographic techniques and usable only by a set of specified entities. A key that is intended for use by a limited number of correspondents for encryption and decryption.<br>⇒ private key. |
| secret key cryptography | ⇒ symmetric cryptography. |
| secure envelope | A set of data items which is constructed by an entity in such a way that any entity holding the secret key can verify their integrity and origin. For the purpose of generating evidence, the secure envelope is constructed and verified by a trusted third party with a secret key known only by a trusted third party. |
| secure key issuing authority | Trusted local authority which generates the public and secret part of a user key pair. |
| security association | The set of security information relating to a given network connection or set of connections. |
| security certificate | A set of digitally signed security relevant data issued by a security authority or trusted third party, together with security information which is used to provide the integrity and data origin authentication. |
| security parameters index | An unstructured opaque index which is used in conjunction with the destination address to identify a particular security association. |
| security policy | The set of criteria for the provision of security services. |
| security token | A set of security relevant data that is protected by integrity and data origin authentication from a source which is not considered a security authority. |
| serial number | Number of a certificate, counting the digitally signed documents issued by an authority for the purpose to retrieve the distributed certificate by the issuer. |
| signature | The string of bits resulting from the signature process.<br>Note: This string of bits may have an internal structure specific to the signature mechanism. |
| signature algorithm | Algorithm used to digitally sign a document. |
| signature algorithm identifier | Identifier for a digital signature algorithm for communicating the used algorithm. |
| signature key | A secret data item specific to an entity and usable only by this entity in the signature process. |
| signature process | A process which takes as input the message or the document, the signature key and the domain parameters, and which gives as output the digital signature. |
| signed message | A set of data items consisting of the signature, the part of the message, which cannot be separated from the signature, and an optional text field. |
| signer | The entity generating a digital signature. |
| subject | An active entity, generally in the form of a person, process or device that causes information to flow among objects or changes the system state. An entity can access objects.<br>⇒ information, object, process. |
| subject key | Public key of the entity requesting the certificate. |
| subject name | Name of the entity requesting and using the certificate. Field of a certificate. |

| | |
|---|---|
| subject public key information | Signature algorithm and public key of the subject requesting the certificate. Field of a certificate. |
| subject unique identifier | Additional Information about the subject requesting the certificate. Field of a certificate. |
| subscriber | Entity requesting to be attested by a registration authority. |
| suspension of certificates | ⇒ revocation of certificates. |
| symmetric cryptography | Cryptography in which the same key is used for encryption and decryption. |
| symmetric cryptographic technique | A cryptographic technique that uses the same secret key for both the originator's and the recipient's transformation. Without knowledge of the secret key, it is computationally infeasible to compute either the originator's or the recipient's transformation.<br>⇒ asymmetric cryptography, public-key cryptography. |
| S-expression | Data structure (Byte-string or list of strings) representing objects in the Simple Distributed Security Infrastructure (SDSI) represented in ASCII. |
| thumbprint | Hash of a certificate or of an certificate revocation list. It is built as a performance optimisation. |
| time stamp | A data item with time and date information assured by a (trusted) time stamping authority. |
| time stamping authority | A (trusted) third party providing evidence which includes the time when the time stamp is generated.<br>⇒ trusted time stamping authority. |
| time stamping token | Token generated by the time stamping authority. |
| trust | A relationship between two elements, a set of activities and a security policy in which element x trusts element y if and only if x has confidence that y will behave in a well defined way (with respect to the activities) that does not violate the given security policy. |
| trust center | A security authority, or its agent, trusted by other entities with respect to security-related services and activities.<br>⇒ trusted third party. |
| trusted cryptographic device | An unforgeable device trusted to process the data as specified and to keep secret data secret. It is not a trusted third party but may be part of or be controlled by a trusted third party. |
| trusted third party | A security authority, or its agent, trusted by other entities with respect to security-related services and activities. |
| trusted time stamp | A data item with time and date information assured by a trusted time stamping authority.<br>⇒ time stamp. |
| trusted time stamping authority | A trusted third party trusted to provide evidence which includes the time when the trusted time stamp is generated.<br>⇒ time stamping authority. |
| user | Any principal (human or machine) who owns, holds and uses a user key pair and can be uniquely identified.The end user of a directory, i.e. the entity or person which accesses the directory. |
| validity period | Interval of time during which an item (e.g. a certificate) is valid. Field of a certificate. |
| verification | Comparing an activity, a process or a product with the corresponding requirements or specifications. |
| verification function | A function in the verification process which is determined by the verification key and which gives a recomputed value of the witness as output. |
| verification key | A data item which is mathematically related to an entity's signature key and which is used by the verifier in the verification process. A value required to verify a cryptographic check value. |
| verification process | A process which takes as input the signed message, the verification key and the domain parameters, and which |

| | |
|---|---|
| | gives as output the result of the signature verification: Valid or invalid. |
| verifier | An entity that verifies an evidence. |
| witness | A data item which provides evidence to the verifier. |

*** *** ***

## 4.2 Abbreviations and Acronyms

| | |
|---|---|
| AA | Accrediting Agency |
| AC | Application Context |
| ACA | Attribute Certification Authority |
| ACDF | Access Control Decision Function |
| ACI | Access Control Information |
| ACIA | Access Control Inner Area |
| ACSA | Access Control Specific Area |
| ACSE | Application Control Service Element |
| ADDMD | Administration Directory Management Domain |
| ADR | Attribute Directory |
| AE | Application Element |
| AI | Application Interface |
| AKI | Authentication Key Infrastructure |
| APCI | Application Protocol Control Information |
| APDU | Application Protocol Data Unit |
| ARA | Attribute Registration Authority |
| ASE | Application Service Element |
| ASN.1 | Abstract Syntax Notation 1 |
| AVA | Attribute Value Assertion |
| BER | Basic Encoding Rules (ASN.1) |
| CA | Certification Authority |
| CCA | Central Certification Authority |
| CCITT | Consultative Committee for International Telegraph & Telephone |
| CCN | Certificate Cancellation Notice |
| CKE | Commercial Key Escrow |
| CKI | Confidentiality Key Infrastructure |
| CKL | Compromised Key List |
| CMA | Certificate Management Authority |
| CMIP | Common Management Information Protocol |
| CMS | Certificate Management Service |
| CONOPS | Concept Of Operations |
| COTS | Commercial Of The Shelf |
| CRA | Central Registration Authority |
| CRC | Certificate Result Certificate |
| CRCert | Certificate Result Certificate |
| CRL | Certificate Revocation List |
| CSO | Central Security Officer |
| CSOR | Computer Security Objects Register |
| DA | Delivery Authority |
| DACD | Directory Access Control Domain |
| DAP | Directory Access Protocol |
| DEDICA | Directory based EDI Certificate Access and management |
| DER | Distinguished Encoding Rules |
| DIB | Directory Information Base |
| DIR | Directory |
| DISP | Directory Information Shadow Protocol |
| DIT | Directory Information Tree |
| DMA | Directory Management Authority |
| DMD | Directory Management Domain |
| DMO | Domain Management Organisation |
| DN | Distinguished Name |
| DOP | Directory Operational binding management Protocol |
| DS | Directory Service |
| DS | Directory System |
| DSA | Digital Signature Algorithm |
| DSA | Directory Service Agent |
| DSA | Directory System Agent |
| DSE | DSA-Specific Entry |
| DSP | Directory System Protocol |
| DSS | Digital Signature Standard |
| DUA | Directory User Agent |
| EAN | European Article Numbering |
| ECDSA | Elliptic Curve Digital Signature Algorithm |
| ECS | Electronic Commerce Services |
| EDI | Electronic Data Interchange |
| EDIRA | EDI Registration Authority |
| EDIFACT | Electronic Data Interchange For Administration, Commerce and Transport |
| ETI | European Trust Infrastructure |
| ETS | European-wide network of Trusted third party Services |
| ETS | European Telecommunication Standard |
| EWOS | European Workshop for Open Systems |
| FEE | Fast Elliptic Encryption algorithm |
| FIPS | Federal Information Processing Standard |
| FQN | Fully Qualified Name |
| FTP | File Transfer Protocol |
| FTTP | Functionally Trusted Third Party |
| GAK | Government Access to Keys |
| HOB | Hierarchical Operational Binding |
| IANA | Internet Assigned Numbers Authority |
| ID | Identity |
| ID | Identifier |
| IETF | Internet Engineering Task Force |
| IPR | Internet Policy Registration |
| IPRA | Internet Policy Registration Authority |
| IPSEC | IETF Internet Protocol Security |
| IS | Information System |
| ISO | International Organisation for Standardisation |
| ISP | Internet Service Provider |
| IT | Information Technology |
| ITU | International Telecommunication Union |
| ITU-T | Telecommunication Standardisation Sector of ITU |

| | |
|---|---|
| I&A | Identification and Authentication |
| KD | Key Distribution |
| KDC | Key Distribution Center |
| KE | Key Escrowing |
| KEA | Key Exchange Algorithm |
| KG | Key Generation |
| KMI | Key Management Infrastructure |
| KMID | Key Management Identifier |
| KMID | Key Material Identifier |
| LDAP | Lightweight Directory Access Protocol |
| LEA | Law Enforcement Agency |
| LRA | Local Registration Authority |
| LSO | Local Security Officer |
| MAC | Message Authentication Code |
| MSP | Message Security Protocol |
| NHOB | Non-specific Hierarchical Operational Binding |
| NRD | Non-Repudiation of Delivery |
| NRDT | Non-Repudiation of Delivery Token |
| NRI | Non-Repudiation Information |
| NRO | Non-Repudiation of Origin |
| NROT | Non-Repudiation of Origin Token |
| NRS | Non-Repudiation of Submission |
| NRST | Non-Repudiation of Submission Token |
| NRT | Non-Repudiation of Transport |
| NRTT | Non-Repudiation of Transport Token |
| NSAP | Network Service Access Point |
| NSSR | Non-Specific Subordinate Reference |
| OCSP | Online Certificate Status Protocol |
| OID | Object Identifier |
| ORA | Organisational Registration Agent |
| ORA | Organisational Registration Authority |
| O/R | Originator/Recipient |
| OSI | Open System Interconnection |
| PAA | Policy Approving Authority |
| PAC | Privilege Attribute Certificate |
| PCA | Policy Certification Authority |
| PCA | Policy Creation Authority |
| PCA | Principal Certification Authority |
| PEM | Privacy Enhanced Mail |
| PGP | Pretty Good Privacy |
| PIN | Personal Identification Number |
| PKCS | Public Key CryptoSystem |
| PKI | Public Key Infrastructure |
| PKIX | Public Key Infrastructure |
| PKPFS | Public Key Protected File System |
| PMC | Personal Mobile Communicator |
| PRDMD | Private Directory Management Domain |
| PSAP | Presentation Service Access Point |
| PSE | Personal Security Environment |
| RA | Registration Authority |
| RDN | Relative Distinguished Name |
| RFC | Request For Comments |
| RHOB | Relevant Hierarchical Operational Binding (i.e. either a HOB or NHOB, as appropriate) |
| ROSE | Remote Operations Service Element |
| RR | Resource Record |
| SDSE | Shadowed DSE |
| SDSI | Simple Distributed Security Infrastructure |
| SECUDE | |
| SENV | Secure Envelope |
| SET | Secure Electronic Transactions |
| SHA | Secure Hash Algorithm |
| SHS | Secure Hash Standard |
| SKIA | Secure Key Issuing Authority |
| SO | Security Officer |
| SPI | Security Parameters Index |
| SPKI | Simple Public Key Infrastructure Proposal |
| TC | Trust Center |
| TCD | Trusted Cryptographic Device |
| TLCMA | Top Level Certificate Management Authority |
| TRP | Trusted Recovery Party |
| TSA | Time Stamping Authority |
| TSP | Technical Security Policy |
| TST | Time Stamping Token |
| TTP | Trusted Third Party |
| T-I | Inter-TTP-Interface |
| UA | User Agent |
| UMTS | Universal Mobile Telecommunications System |
| UN/EDIFACT | United Nations / EDIFACT |
| URI | Uniform Resource Identifier |
| USPS | United States Postal Service |
| UTTP | Unconditionally Trusted Third Party |
| U-I | User Interface |
| VAS | Value Added Service |
| VPN | Virtual Private Network |
| WEMA | World Electronic Message Association |

*** *** ***

## 4.3 References


Albitz, P.; Liu, C.: DNS and BIND in a Nutshell. Sebastopol 1994

American Bar Association: Digital Signature Guidelines. Legal Infrastructure for Certification Authorities and Electronic Commerce. Science and Technology Section. Information Security Committee. Electronic Commerce and Information Technology Division Chicago 1995

Anderson, R.; Roe, M.: The GCHQ Protocol and its Problems. Cambridge 1996

Atkinson, R.: Security Architecture for the Internet Protocol. Internet Network Working Group. RFC 1825[13]. 1995

Atkinson, R.: Toward a More Secure Internet. IEEE Computer 1, 57 - 61, 1997

Aziz, A.; Markson, T.; Prafullchandra, H.: X.509 Encoding of Diffie-Hellman Public Values. IPSEC Working Group INTERNET-DRAFT. August 5, 1996

Aziz, A.; Markson, T.; Prafullchandra, H.: Simple Key-Management For Internet Protocols (SKIP). IPSEC Working Group INTERNET-DRAFT. August 14, 1996

Bahreman, A.: PEMToolKit: Building a Top-Down Certification Hierarchy for PEM from the Bottom Up. Proceedings of the Symposium on Network and Distributed System Security. Internet Society - IEEE, 161 - 171, Los Alamitos 1995

Balenson, D.: Privacy Enhancement for Internet Electronic Mail: Part III: Algorithms, Modes, and Identifiers. Internet Network Working Group. RFC 1423, Feb. 1993

Barker, P.; Kille, S.; Lenggenhager, T.: Naming and Structuring Guidelines for X.500 Directory Pilots. Internet Network Working Group. RFC 1617. RARE Technical Report 111. London 1994

Bauspieß, F.; Cruellas, J. C.; Rubia, M.: DEDICA - Directory based EDI Certificate Access and Management. In: Horster, P. (Ed.): Digitale Signaturen. 123 - 134. Braunschweig 1996

Bizer, J.; Hammer, V.; Pordesch, U.; Roßnagel, A.: Entwurf gesetzlicher Regelungen zum Datenschutz und zur Rechtssicherheit in Online-Multimedia-Anwendungen. Gutachten für das Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie. 15. Februar 1996

Black, D. K.: The Digital Signature Standard: Overview and Current Status. Computers & Security 12, 5, 437 - 446, 1993

Blakley, B.: Architecture for Public-Key Infrastructure. Internet-Draft IETF PKIX WG. November 1996

Blaze, M.; Feigenbaum, J.; Lacy, J.: Distributed Trust Management. Proc. IEEE Symposium on Security and Privacy. 164 - 173. Oakland 1996.

Blaze, M.; Feigenbaum, J.; Resnick, P.; Strauss, M.: Managing Trust in an Information-Labelling System. Preliminary Version. Murray Hill 1996

Boeyen, S.; Housley, R.; Howes, T.; Myers, M.; Richard, P.: Internet Public Key Infrastructure. Part 2: Operational Protocols. Internet Draft PKIX Working Group. March 1997

Bundesrat (Ed.): Entwurf eines Gesetzes zur Regelung der Rahmenbedingungen für Informations- und Kommunikationsdienste (Informations- und Kommunikationsdienste-Gesetz - IuKDG). Gesetzentwurf der Bundesregierung. Drucksache 966/96 vom 20. 12. 1996

Bundesrat (Ed.): Gesetz zur digitalen Signatur (Signaturgesetz - SigG). In: Bundesrat (Ed.): Entwurf eines Gesetzes zur Regelung der Rahmenbedingungen für Informations- und Kommunikationsdienste (Informations- und Kommunikationsdienste-Gesetz - IuKDG). Art. 3. Gesetzentwurf der Bundesregierung. Drucksache 966/96 vom 20. 12. 1996

Burr, W. E.: A Proposed Federal PKI Using X.509 V3 Certificates. NIST Gaithersburg 1996

Burr, W. E.: Public Key Infrastructure (PKI) Technical Specifications (V 1). Part C: Concept of Operations. NIST/MITRE Gaithersburg 1996

Burr, W.; Dodson, D.; Nazario, N.; Polk, W. T.: MISPC - Minimum Interoperability Specification for PKI Components. Draft Version 1. 1996

Carpenter, B.: Architectural Principles of the Internet. RFC 1958. June 1996

Carter, G.: TESTFIT – TTP & Electronic Signature Trial for Inter-modal Transport. Proceedings of the TEDIS - EDI Trusted Third Party Workshop 17 - 26. Barcelona 1995


[13] RFC: Request for Comments.

Cerny, D.: Verteilung und Verwaltung von Schlüsseldaten durch Vertrauensinstanzen - Eine mögliche Lösung zur Kryptokontroverse? KES - Zeitschrift für Kommunikations- und EDV-Sicherheit 4/1996. S. 45 - 47.

Chen, L.; Hitz, H.-J.; Horn, G.; Howker, K.; Kessler, V.; Knudsen, L.; Mitchell, C.J.; Radu, C.: The Use of Trusted Third Parties and Secure Billing in UMTS.

Cheriton D.; Mann T. .P.: Decentralizing a Global Naming Service for Improved Performance and Fault Tolerance; ACM Transactions on Computer Systems. 7, .8, 147-183, 1989

Chokhani, S.: A Security Flaw in the X.509 Standard. In: National Institute of Standards and Technology (NIST) - National Computer Security Center: 19th National Information Systems Security Conference. 463 - 470, Baltimore 1996

Chokhani, S.; Ford, W.: Internet Public Key Infrastructure. Part IV: Certificate Policy and Certification Practices Framework. March 25, 1997

Chokhani, S.; Ford, W.: Certificate Policy and Certification Practice Statement Framework. Draft. NIST Gaithersburg 1996

Coe, D. E.; Smith, F. J.: Developing and Deploying a Corporate-Wide Digital Signature Capability. Bedford o.J.

Common Criteria Editorial Board (Ed.): Common Criteria for Information Technology Security Evaluation Version 1.0, January 1996

Communications Electronics Security Group. Government Communications Headquarters: Confidentiality Key Infrastructure. Part 1: Architecture & Concept of Operation. Cheltenham 1997

Communications Electronics Security Group. Government Communications Headquarters: Confidentiality Key Infrastructure. Part 3: Certificate Profile. Cheltenham 1997

Conolly, D.: Character Set Considered Harmful. May 2, 1995. Internet-Draft. IETF HTML Working Group W3C

Crepin-Leblond, O. M. J.: International Domain Names (Standard ISO 3166). o.J.

CygnaCom Solutions, Inc. (Ed.): Federal Public Key Infrastructure (PKI). Technical Specification (Version 1). Part D: Interoperability Profiles. NIST/MITRE Gaithersburg 1995

Danish, H.: The Exponential Security System TESS: An Identity-Based Cryptographic Protocol for Authenticated Key Exchange. Internet Network Working Group. RFC 1824. 1995

Deloitte and Touche (Ed.): BOLERO. Final Report. Infosec 94. Contract S 2302. Brussels 1995

Denning, D. E.; Sacco, G. M.: Timestamps in Key Distribution Protocols. Communications of the ACM 24, 8, 533 - 536, 1981

Denning, D. E.: Protecting Public Keys and Signature Keys. IEEE Computer, 27 - 35, February 1983

Denning, D. E.; Branstad, D. K.: A taxonomy for key escrow encryption systems. Communications of the ACM 30, 3, 34 - 40, 1996

Department of Commerce: Charter of the Technical Advisory Committee to develop a Federal Information Processing Standard for the Federal Key Management Infrastructure. Washington 1996

Department of Trade and Industry (DTI): Paper on Regulatory Intent Concerning Use of Encryption on Public Networks. London 1996

Department of Trade and Industry (DTI): Licensing of Trusted Third Parties for the Provision of Encryption Services. Public Consultation Paper on Detailed Proposals for Legislation. London 1997

DesAutels, P. A.: Digital Signature Initiative Proposal. W3C Digital Signature Initiative Proposal. 1996

DesAutels, P. A.; Khare, R.: Proposed Digital Signature Architecture. W3C Working Draft. 1996

Diffie, W.; Hellman, M. E.: New Directions in Cryptography. IEEE Transaction on Information Theory IT-22, 644 - 654, 1976

Digital Equipment Corporation (Ed.): DEC X.500 Directory Service. Management. AA-QEVEB-TE o.J.

Dix, A.: Das weltweite Directory und persönliche Numerierungssysteme. Datenschutz und Datensicherheit 9, 484 - 488, 1994

Dobbertin, H.: Welche Hash-Funktionen sind für digitale Signaturen geeignet? In: Horster, P. (Ed.): Digitale Signaturen. Braunschweig 1996

Eastlake, D. E.: Mail Ubiquitous Extensions (MUSE). Internet-Draft 1996

Eastlake, D. E.; Kaufman, C. W.: DNS Protocol Security Extensions. Boston 1994

Eastlake, D. E.; Kaufman, C. W.: Domain Name System Security Extensions. DNS Security Working Group. RFC 2065. Updates RFC 1034, 1035. Boston 1997

EDIRA Memorandum of Understanding for the Operation of EDI Registration Authorities. 1 February 1995.

Electronic Commerce Promotion Council of Japan: Announcement of Certification Authority Guidelines in Japan. 7th April 1997

ElGamal, T.; Treuhaft, J.; Chen, F.: Securing Communications on the Intranet and over the Internet. July 1996

Ellison, C. M.: Emergency Key Recovery Without Third Parties.

Ellison, C. M.: Establishing Identity Without Certification Authorities. USENIX Security Symposium. San Jose 1996

Ellison, C. M.: Generalized Certificates. 12. September 1996.

Ellison, C. M.: SPKI Certificates. DIMACS Workshop 1996

Ellison, C. M.: SPKI Requirements. INTERNET-DRAFT. 17 March 1997

Ellison, C. M.; Frantz, B.; Thomas, B. M.: Simple Public Key Certificate. Internet-Draft. Baltimore 25 March 1997

European Commission: The TEDIS Programme. A Proposal Concerning the Use of Digital Signatures in EDIFACT. Brussels 1990

European Commission Information Technology Security Evaluation Criteria (ITSEC). Provisional Harmonised Criteria. COM(90) 312. Brussels1991

European Commission: Report to The Commission of the European Communities for the Requirements for Trusted Third Party Services. Infosec 93. Task S.01 Contract S2101. Brussels 1993

European Commission DG XIII/B: Green Paper on the Security of Information Systems. DG XIII/B. Brussels April 1994

European Commission: Proposal for a Council Decision adopting a multi-annual Action concerning the Establishment of Europe-wide Trust Services for non-classified Information Services (ETS). Brussels 1994. Datenschutz und Datensicherung 8, 453 - 456, 1994

European Commission DG XIII/B/6: TESTFIT – TTP & Electronic Signature Trial For Inter-modal Transport. Infosec 94. Contract S2303. Brussels September 1995

European Commission: Richtlinie 95/47/EG des Europäischen Parlaments und des Rates vom 24. Oktober 1995 zum Schutz natürlicher Personen bei der Verarbeitung personenbezogener Daten und zum freien Datenverkehr. Amtsblatt der Europäischen Gemeinschaften L 281, 31 - 50, 38. Jahrg. 23. Nov. 1995

EWOS EWOS/ETG 67. Guidance on EDI Use of Directory. 1996

Farmer, W. M.; Guttman, J.D.; Swarup, V.: Security for Mobile Agents: Issues and Requirements. In: National Institute of Standards and Technology (NIST) - National Computer Security Center: 19th National Information Systems Security Conference. 591 - 597, Baltimore 1996

Farmer, W. M.; Guttman, J.D.; Swarup, V.: Security for Mobile Agents: Authentication and State Appraisal. Proc. European Symposium on Research in Computer Security. 1996

Farrel, S.; Adams, C.: Internet Public Key Infrastructure. Part III: X.509 Certificate Management Protocols. Internet Draft PKIX Working Group. December 1996

Farrel, S.; Kaijser, P.: A non-repudiation Service Architecture and Certification Infrastructure. Proceedings of the TEDIS - EDI Trusted Third Party Workshop 85 - 96. Barcelona 1995

Fjelbye, P.: TeleSeC - a Solution to Implementing Digital Signature in EDI/EDIFACT. Proceedings of the TEDIS - EDI Trusted Third Party Workshop 121 - 129. Barcelona 1995

Ford, W.: A Public Key Infrastructure for U.S. Government Unclassified but Sensitive Applications. NIST 1995

Ford, W.: Advances in Public-Key Certificate Standards. ACM SIGSAC Security Audit & Control Review 13, 3, 1995

Ford, W.: Advances in Public-Key Certificate Standards. Ottawa o.J.

Fox, D.; Müller, M.: Neue Zertifikate für asymmetrische Sicherheitsprotokolle. In: Horster, P. (Ed.): Trust Center. Braunschweig 1995

Freier, A.O.; Karlton, P.; Kocher, P.C.: The SSL Protocol Version 3.0. Internet Draft. March 1996

Fries, O. et al. (Ed.): Sicherheitsmechanismen. Bausteine zur Entwicklung sicherer Systeme. 1993

| | |
|---|---|
| Froomkin, A. M.: | The Essential Role of Trusted Third Parties in Electronic Commerce. 1996 |
| Fumy, W.; Landrock, P.: | Principles of Key Management. IEEE Journal on Selected Areas in Communications II, 5, 6, 785 - 793, 1993. |
| Fumy, W.: | Standardisierung kryptographischer Mechanismen. Datenschutz und Datensicherung 8, 479 - 483, 1996 |
| Geihs, K.: | Infrastrukturen für heterogene verteilte Systeme. Informatik Spektrum 1, 1993 |
| GMD (Ed.): | SECUDE 5.0 – Hyperlink Documentation. Darmstadt 1996 |
| Glöckner, P.: | ITU-T Recommendation X.509 v3 Certificate. Report 1996 |
| Gollmann, D.; Beth, Th.; Damm, F.: | Authentication services in distributed systems. Computers & Security 12, 8, 753 - 764, 1993 |
| Grimm, R.: | Sicherheit für offene Kommunikation - Verteilte Telekooperation. Mannheim 1994 |
| Grimm, R.: | Kryptoverfahren und Zertifizierungsinstanzen. Datenschutz und Datensicherung 1, 27 - 36, 1996 |
| Gutmann, P.: | X.509 Style Guide. Auckland 1996 |
| Haber, S.; Stornetta, W. S.: | Secure Names for Bit-Strings. Proceedings of the ACM Conference on Computer and Communications Security. 1997 |
| Hallam-Baker, P. M.: | Micro Payment Transfer Protocol (MPTP) Version 0.1. W3C Working Draft 22-Nov-1995 |
| Hammer, V. (Ed.): | Sicherungsinfrastrukturen - Gestaltungsvorschläge für Technik, Organisation und Recht. Heidelberg 1995 |
| Hammer, V.: | Beweiswert elektronischer Signaturen. In: Weck, G.; Horster, P. (Ed.): Verläßliche Informationssysteme. VIS 93. Braunschweig 1993 |
| Hammer, V.: | Digitale Signaturen mit integrierter Zertifikatkette - Gewinne für den Urheberschafts- und Autorisierungsnachweis. In: Brüggemann, H.H.; Gerhardt, W.; Häckl, W. (Ed.): Verläßliche IT-Systeme VIS 95. 265 - 274. Braunschweig 1995 |
| Hammer, V.: | Gestaltungsanforderungen zum Nachweis der Urheberschaft digital signierter Dokumente. In: Bauknecht, K.; Teufel, S. (Ed.): Sicherheit in Informationssystemen. Zürich 1994 |
| Hammer, V.: | Vor- und Nachteile von Mehrfachzertifikaten für öffentliche Schlüssel. In: Bundesamt für die Sicherheit in der Informationstechnik (Ed.): Fachvorträge 4. Deutscher IT-Sicherheitskongreß. Bonn 1995 |
| Hammer, V.; Bizer, J.: | Beweiswert elektronisch signierter Dokumente. Datenschutz und Datensicherheit 12, 689 ff. 1993 |
| Hardy, G. (Ed.): | Trusted Third Parties. Oxford 1996 |
| Herda, S.; Seidel, U.; Struif, B.: | Bestandsaufnahme über die elektronischen Signaturverfahren. Studie der GMD für das Bundesamt für Sicherheit in der Informationstechnik (Bundesamt für Sicherheit in der Informationstechnik (BSI)). St. Augustin 1992 |
| Herson, D.: | Ethical Trusted Third Party Services - A New Security Paradigm. Brussels 1996 |
| Housley, R.; Ford, W.; Farrel, S.; Solo, D.: | Internet Public Key Infrastructure. Herndon 1995 |
| Housley, R.; Ford, W.; Farrel, S.; Solo, D.: | Internet Public Key Infrastructure. Part I: X.509 Certificate and CRL Profile. PKIX Working Group Internet Draft. Dec. 1996 Internet Draft |
| Hueske, Th.: | Sicherheitsinfratsruktur im TeleTrusT-Projekt "MailTrusT". In: Glade, Reimer, Struiff: Digitale Signatur und Sicherheitssensitive Anwendungen. 250 – 258, Braunschweig 1995 |
| IETF HTML Working Group W3C : | Micro Payment Transfer Protocol. 22-November-95. |
| IETF HTML Working Group W3C : | Proposed Digital Signature Architecture. 26-October-96. |
| IFIP TC11: | The IFIP Position on Cryptopolicies. Klagenfurt 1997 |
| Intel (Ed.): | Intel`s Common Data Security Architecture. December 1996 |
| ISO 3166 | Codes for the representation of names of countries. |
| ISO IS 8824: | Specification of Abstract Syntax Notation One (ASN.1), May 1987 |
| ISO/IEC JTC 1 DIS 2382-8 | Information Technology - Vocabulary. Part 8: Security. 1996 |
| ISO/ IEC 7498-2 | CCITT X.800ISO/IEC 9594-2 DAM 4 Certificate Extensions. (X.501) 1996 |
| ISO/IEC 9594-7 DAM 1 | Certificate Extensions. (X.521) 1996 |

| | |
|---|---|
| ISO/IEC 9594-8 DAM 1 | Certificate Extensions. (X.509) 1996 |
| ISO 9594-8 | Directory Authentication. S. ITU-T Recommendation X.509. 1988 |
| ISO/IEC CD 9796-2 (review): | Information Technology - Security Techniques - Digital Signature Scheme giving Message Recovery - Part 2: Mechanisms using Hash-Function. 1996 |
| ISO/IEC 9797: | Information Technology - Security Techniques - Data Integrity Mechanisms using a cryptographic Check Function employing a Block Cipher Algorithm. SC 27 N 790 DIN NI - 27. 169-93.1993 |
| ISO/IEC CD 9798-1: | Information Technology - Security Techniques - Entity Authentication mechanisms - Part 1: General model. SC27 N 1257 DIN NI-27 31-96. 1996 |
| ISO/IEC 9798-4: | Information Technology - Security Techniques - Entity Authentication. Part 4: Mechanisms using a cryptographic Check Function. SC27 N 952 DIN NI-27 183-94. 1994 |
| ISO/IEC CD 9798-5: | Information Technology - Security Techniques - Entity Authentication. Part 5: Mechanisms using Zero Knowledge Techniques. SC27 N 1375 DIN NI-27 119-96. 1996 |
| ISO/IEC 10118-1: | Information Technology - Security Techniques - Hash Functions - Part 1: General. SC27 N 828 DIN NI-27 4-94. 1994 |
| ISO/IEC 10118-2: | Information Technology - Security Techniques - Hash Functions - Part 2: Hash Functions using an n-bit Block Cipher Algorithm. SC27 N829 DIN NI-27 5-95. 1994 |
| ISO/IEC CD 10118-3: | Information Technology - Security Techniques - Hash Functions - Part 3: Dedicated Hash Functions. SC27 N1209 DIN NI-27 80-96. 1996 |
| ISO/IEC CD 10118-4: | Information Technology - Security Techniques - Hash Functions - Part 4: Hash Functions using modular Arithmetic. SC27 N1211 DIN NI-27 81-96. 1996 |
| ISO/IEC DIS 11770-1: | Information Technology - Security Techniques - Key Management - Part 1: Framework. SC27 DIN NI-27 06-96. 1996 |
| ISO/IEC 11770-2: | Information Technology - Security Techniques - Key Management - Part 2: Mechanism using Symmetric Techniques. SC27 DIN NI-27 32-96. 1996 |
| ISO/IEC 11770-3: | Information Technology Security Techniques - Key Management - Part 3: Mechanisms using asymmetric Techniques. With Annexes A, B, C, D and E. 1996 |
| ISO/IEC DIS 13888-1: | Information Technology - Security Techniques - Non-repudiation. Part 1: General. 1996 |
| ISO 14516-2: | Guidelines for the Use and Management of TTP Services. Part 2: Technical Aspects. 1997 |
| ISO/IEC CD 14888-1: | Information Technology - Security Techniques - Digital signatures with appendix - Part 1: General. 1997 |
| ISO/IEC JTC 1/SC 27: | Draft of the ISO/TC 68 Catalogue of Security related Standards. SC 27 Information Technology Security Techniques. N 1312 DIN NI - 27. 82-96. 1996 |
| ISO/IEC JTC 1/SC 27: | Proposal for a New Work Item: Guidelines for the Use and Management of Trusted Third Party (TTP) Services. SC 27 Information Technology Security Techniques. N 858 DIN NI - 27. 33-94.1994 |
| ITU-T: | Recommendation X.400 Data Communication Networks: Message Handling Systems. Message Handling Services: Message Handling System and Service Overview (Previously: CCITT X.400 Recommendation) |
| ITU-T | Recommendation X.435 Message Handling: Electronic Data Interchange Messaging Service. (Previously: CCITT X.435 Recommendation) |
| ITU-T | Recommendation X.500 Series. ISO/IEC 9594, 1-9. Information Technology - Open Systems Interconnection: The Directory. (Previously: CCITT X.500 Recommendation) |
| ITU-T | Recommendation X.509 Open Systems Interconnection: The Directory: Authentication Framework. (Previously: CCITT X.509 Recommendation) |
| ITU-T | Recommendation X.511 Data Network and Open System Communications – Directory. Information Technology - Open Systems Interconnection - The Directory: Abstract Service Definition. (Previously: CCITT X.520 Recommendation) |

| | |
|---|---|
| ITU-T | Recommendation X.520 Data Networks and Open System Communications – Directory. Information Technology - Open Systems Interconnection - The Directory: Selected Attribute Types. (Previously: CCITT X.520 Recommendation) |
| ITU-T | Recommendation X.521 Data Networks and Open System Communications – Directory. Information Technology - Open Systems Interconnection - The Directory: Selected Object Classes. (Previously: CCITT X.521 Recommendation) |
| Jakobs, K.: | Directory-Systeme. Informatik Spektrum 10, 4, 220 - 221, 1987 |
| Jansen, W. A.: | A Second Look at the SDNS Key Management Protocol. Proceedings of the Symposium on Network and Distributed System Security. Internet Society - IEEE, 74 - 81, Los Alamitos 1995 |
| Jefferies, N.; Mitchel, C.; Walker, M.: | A proposed Architecture for Trusted Third Party Services. Information Security Group Royal Holloway, University of London, 1995 |
| Jordan, F.; Medina, M.: | Certificate Infrastructure and Unique Identification. Proceedings of the TEDIS - EDI Trusted Third Party Workshop 109 - 118. Barcelona 1995 |
| Jurg, P.: | Introduction to White Pages Services based on X.500. RFC 1684, August 1994 |
| Kaliski, B.S.: | Privacy Enhancement for Internet Electronic Mail: Part IV: Key Certificates and Related Services. RFC 1424, Feb. 1993 |
| Kapidzik, N.; Davidson, A.: | A Certificate Management System: Structure, Functions and Protocols. Proceedings of the Symposium on Network and Distributed System Security. Internet Society - IEEE, 153 - 160, Los Alamitos 1995 |
| Kapidzik, N.; Young, A.; Glöckner, P.; Farell, S.: | Architecture and General Specifications of the Public Key Infrastructure. ICE-TEL - Interworking Public Key Certification Infrastructure for Europe. 1996 |
| Kent, S. T.: | Internet Privacy Enhanced Mail. Communications of the ACM 36, 8, 48 - 60, 1993 |
| Kent, S. T.: | Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management. RFC 1422, Feb. 1993 |
| Kille, S.: | Using the OSI Directory to Achieve User Friendly Naming. RFC 1781, March 1995 |
| Killie, S.; Wahl, M.: | An Approach for Using Domains in LDAP DNs. February 1997 |
| Kowalski, B.: | Trust Center Dienstleistungen für Anwender von Signatur-Funktionen. In: Reimer, H.; Struif, B. (Ed.): Kommunikation und Sicherheit. 27 ff. Darmstadt 1992 |
| Kruse, D.; Stöcker, E.: | Functionality Classes: Digital Signature for Electronic Data. TeleTrusT AG2. Erfurt 1995 |
| Lampson B.; Abadi M.; Burrows M.; Wobber, E.: | Authentication in Distributed Systems: Theory and Practice. ACM Transactions on Computer Systems, 10, 11, 265 – 310, 1992 |
| Landrock, P.: | Roles and Responsibilities in BOLERO. Proceedings of the TEDIS - EDI Trusted Third Party Workshop 97 - 107. Barcelona 1995 |
| Laurie, B.: | A Supplementary Analysis of the Royal Holloway TTP-based Key Escrow Scheme. 16. Nov. 1996 |
| Levien, R.; McCarthy, L.; Blaze, M.: | Transparent Internet E-mail Security. Draft. Murray Hill 1996 |
| Linn, J.: | Privacy Enhancement for Internet Electronic Mail: Part I - Message Encipherment and Authentication Procedures. RFC 1421, Feb. 1993 |
| Longley, D.; Shain, M.: | Data & Computer Security. Dictionary of Standards, Concepts and Terms. Basingstoke 1989 |
| Lowry, J.: | Location-Independent Information Object Security. Proceedings of the Symposium on Network and Distributed System Security. Internet Society - IEEE, 54 - 62, Los Alamitos 1995 |
| Madan, M. S.: | On Naming Considerations for Networks. ACM Comp. Com. Review 15, 33, 6 - 9, 1985 |
| Magalhaes, J.-P.: | The Registration Authority for the Bank Identifier Code. Proceedings of the TEDIS - EDI Trusted Third Party Workshop 147 - 155. Barcelona 1995 |
| Mansfield, N.P.: | Organisation of Electronic Business in Shell companies. Proceedings of the TEDIS - EDI Trusted Third Party Workshop 133 - 146. Barcelona 1995 |

Maughan, D.; Schertler, M.; Schneider, M.; Turner, J.: Internet Security Association and Key Management Protocol (ISAKMP). IPSEC Working Group INTERNET-DRAFT. February 21, 1997

Maurer, U.: Modelling a Public-Key Infrastructure. In: Bertino, E. (Ed.): ESORICS `96. Berlin 1996

Maurer, U.; Yacobi, Y.: A Non-interactive Public-Key Distribution System. EUROCRYPT `91. 498 – 507. Berlin 1991

McConnel, B. W.; Appel, E. J.: Enabling Privacy, Commerce, Security and Public Safety in the Global Information Infrastructure. Draft Paper. Executive Office of the President, Interagency Working Group on Cryptography Policy (EPIC) - Office of Management and Budget (OMB). Washington May 20, 1996

Medina, M. (Ed.): TEDIS EDITT. EDI Trusted Third Parties Workshop. Barcelona 1995

Mendes, S.; Huitema, C.: A New Approach to the X.509 Framework: Allowing a Global Authentication Infrastructure without a Global Trust Model. Proceedings of the Symposium on Network and Distributed System Security. Internet Society - IEEE, 172 - 189, Los Alamitos 1995

Mitchell, C. J.: The Royal Holloway TTP-based Key Escrow Scheme. London 1996

Mockapetris P. Domain Names - Implementation and Specification. RFC 1035. USC-ISI, Marina del Rey, Nov 1987

Nazario, N. A.: Federal Public Key Infrastructure (PKI) Technical Specifications (V 1). Part B: Technical Security Policy. NIST/MITRE 1996

Nazario, N. A.: Security Policies for the Federal Public Key Infrastructure. In: National Institute of Standards and Technology (NIST) - National Computer Security Center (NCSC): 19th National Information Systems Security Conference. 438 - 444, Baltimore 1996

Nelson, R.: SDNS Services and Architecture. In: National Institute of Standards and Technology (NIST) - National Computer Security Center: 10th National Information Systems Security Conference. 348 - 352, Baltimore 1987

NII Federal Information: Security Infrastructure Program Management Office: Action Plan. Washington 1995

NIST (Ed.): Secure Hash Standard (SHS), Draft 1991

NIST (Ed.): A Proposed Federal Information Processing Standard for Digital Signature Standard (DSS), National Institute of Standards and Technology Aug., 30, 1991

N.N.: Verordnung zur digitalen Signatur (Signaturverordnung - SigV). Bonn 16. Dezember 1996

Peereman, M.: FAST - Fist Attempt to Secure Trade. 91 – 99, 1994

Petersen, L. L.: The Profile Naming Service; ACM Transactions on Computer Systems, 6, 4, 341 - 364, 1988

Pohl, H.: Informationssicherheit der Global Information Infrastructure (GII) - Einige Bemerkungen zu Problemen und Entwicklungen. In: Tauss, J. et al. (Ed.): Deutschlands Weg in die Informationsgesellschaft. Herausforderungen und Perspektiven für Wirtschaft, Wissenschaft, Recht und Politik. S. 358 - 390. Baden Baden 1996

Polk, W.: Federal Public Key Infrastructure (PKI). Technical Specifications (Version 1). Part A: Requirements. NIST/MITRE 1996

Pordesch, U.: Risiken elektronischer Signaturverfahren. Datenschutz und Datensicherheit 10, 561 - 569, 1993

Postel, J.: Domain Name System Structure and Delegation. RFC 1591. 1994

Raubold, E.: Sicherheitskonzepte für "offene" IT-Anwendungen. In: Cyranek, G.; Bauknecht, K. (Ed.): Sicherheitsrisiko Informationstechnik - Analysen, Empfehlungen, Maßnahmen in Staat und Wirtschaft. 35 - 43. Braunschweig 1994

Reimer, H.; Struif, B. (Ed.): Kommunikation und Sicherheit. Darmstadt 1992

Reitenspiess, M.: Open System Security Standards. Computers & Security 12, 4, 341 - 361, 1993

Rihaczek, K.: TeleTrusT-OSIS and Communication Security. Computers & Security 6, 206 - 218, 1987

Rihaczek, K.: Die Handshake-Protokolle zur gegenseitigen Teilnehmerauthentikation. Datenschutz und Datensicherheit 2, 70 - 78, 1989

Rihaczek, K.: Die KEG und "Trusted Services" - Kommunikationsdienste mit öffentlichem Vertrauen. Datenschutz und Datensicherheit 8, 452 - 456, 1994

Rihaczek, K.: Data Interchange and Legal Security - Signature Surrogates. Computers & Security 13, 4, 287 - 293, 1994

Rihaczek, K.: Neue französische Kryptogesetzgebung. Datenschutz und Datensicherung 8, 484 - 489, 1996

Rihaczek, K.: Die US-Kryptoinitiative. Datenschutz und Datensicherung 10, 603 - 611, 1996

Rivest, R.L. SPKI/SDSI 2.0 A Simple Distributed Security Infrastructure. Maryland Theoretical Computer Science Day 4/11/97.

Rivest, R. L.; Lampson, B.: SDSI - A Simple Distributed Security Infrastructure. Cambridge 1996

Roe, M. (et al.): PASSWORD. A report of the value project. Brussels 1992/93

Roßnagel, A.: Das Signaturgesetz - eine kritische Bewertung des Gesetzentwurfs der Bundesregierung. Darmstadt 1997

Rotenberg, M.: Communications Privacy: Implications for Network Design. Communications of the ACM 36, 8, 61 - 60, 1993

Rubinowitz, H. H.: Issues 94 - Public Key - Trials and Tribulations. Bedford 1994

Rueppel, R. A.: Revocation and Revocation Certificates. In: Medina, M. (Ed.): TEDIS EDITT. EDI Trusted Third Parties Workshop. Barcelona 1995

Rueppel, R. A.: Standardization of Digital Signatures in Europe? Proceedings of the TEDIS - EDI Trusted Third Party Workshop. Barcelona 1995

Rueppel, R. A.; Wildhaber, B.: Public Key Infrastructure - Survey and Issues. In: Horster, P. (Ed.): Trust Center. Braunschweig 1995

Sandhu, R. S.; Coyne, E. J.; Feinstein, H. L.; Youman, C. E.: Role-based Access Control Models. IEEE Computer 29, 2, 38 - 47, 1996

Schneider, W.: PASSWORD. Ein EG-Projekt zur pilotmäßigen Erprobung von Authentisierungsdiensten. In: Reimer, H.; Struif, B. (Ed.): Kommunikation und Sicherheit. 63 - 67. Darmstadt 1992

Schneier, B.: Applied Cryptography. New York 1996

Schneier, B.: Angewandte Kryptographie. Protokolle, Algorithmen und Sourcecode in C. Bonn 1996

Shoch, J. F.: Inter-Network Naming, Addressing, and Routing; Proceedings COMPCON 1978

Soete, De M.; Landrock, P.; Rueppel, R.: Some Fundamental New Services using TTPs and their Realisation. Brussels 1993

Sollins, K.; Masinter, L.: Functional Requirements for Uniform Resource Names. RFC 1737. 1994

Stein, L. H.; Stefferud, E. A.; Borenstein, N. S.; Rose, M. T.: The Green Commerce Model. First Virtual Holdings Inc. June 1995

Struif, B.: Das elektronische Rezept mit digitaler Unterschrift. In: Reimer, H.; Struif, B. (Ed.): Kommunikation und Sicherheit. 71 - 75. Darmstadt 1992

Tardo, J.J.; Alagappan, K.: SPX: Global Authentication using Public Key Certificates. IEEE Symposium on Security and Privacy 232 - 244, 1991

Tauss, J. et al. (Ed.): Deutschlands Weg in die Informationsgesellschaft. Herausforderungen und Perspektiven für Wirtschaft, Wissenschaft, Recht und Politik. Baden Baden 1996

TeleTrusT: Security Information Objects. The TeleTrusT / TeleSec Certificate. Erfurt 1992

United Nations (Ed.): Planning of Future Work on Electronic Commerce: Digital Signatures, Certification Authorities and Related Legal Issues. Commission on International Trade Law. Working Group on Electronic Commerce. Thirty-first session. New York, 18 - 28 February 1997

U. S. Department of Commerce (Ed.): Security Requirements for Cryptographic Modules. Federal Information Processing Standards Publication 140-1. National Institute of Standards and Technology. Gaithersburg 1994

U. S. Department of Commerce (Ed.): Secure Hash Standard. Federal Information Processing Standards Publication 180-1. National Institute of Standards and Technology. Gaithersburg 1995

U. S. Department of Commerce (Ed.): Digital Signature Standard. Federal Information Processing Standards Publication 186. National Institute of Standards and Technology. Gaithersburg 1994

Verheul, E.; Koops, B.-J.; Tilborg, H. v.: Binding cryptography. A fraud-detectible alternative to key-escrow proposals. The Computer Law & Security Report 3 – 14, 1-2, 1997

VeriSign (Ed.): Certification Practice Statement. Version 1. Mountain View 1996

Wahl, M.; Howes, T.; Killie, S.: Lightweight Directory Access Protocol (v3). 10/1996.

Webster`s New Encyclopedic Dictionary. New revised edition. New York 1996

Weider, C.; Wright, R.: A Survey of Advanced Usages of X.500. RFC 1491. July 1993

World Electronic Messaging Association (Ed.): WEMA X.400/INTERNET Personal Naming Recommendation. Anaheim 1996

Wichmann, P.: Die DSSA Sicherheitsarchitektur für verteilte Systeme. Datenschutz und Datensicherung 1, 23 - 27, 1993

Wildhaber, B.: European Trends in Banking Security - Experiences in Switzerland and Standardisation in Europe. Aathal 1996

Wildhaber, B.: Legal Aspects and Security in electronic Markets for Tourism. Proceedings of the Conference Information and Communication Technologies in Tourism. Innsbruck 1995

Wittgenstein, L.: Tractatus logico-philosophicus. Frankfurt 1960

Woo, T. Y. C.; Lam, S. S.: Authentication for Distributed Systems. IEEE Computer, January, 39 - 52, 1992

Yeong, W.; Howes, T.; Kille, S.: Lightweight Directory Access Protocol. RFC 1777 March 1995

Young, A.: Connectionless Lightweight Directory Access Protocol. June 1995

*** *** ***

# Appendix

# A Fields of the X.509 v3 Certificate

## 1 Basic Fields of the X.509 v3 Certificate

Version

The version field can be 0, 1, or 2 depending on the version number. The version number must be v3 if any extension filed is present.

Serial Number

The serial number is an integer assigned by the certification authority to each certificate. It must be unique for each certificate issued by a given certification authority. The issuer name and serial number identify a unique certificate.

Signature

This field contains the algorithm identifier for the algorithm used by the certification authority to sign the certificate.

Issuer Name

The issuer name identifies the entity who has signed (and issued the certificate). The syntax of the issuer name is an X.500 Distinguished Name.

Validity

Indicates the first and last date (notBefore, notAfter) of which the certificate is valid respectively. The validity period is given in universal time encoding (UTC) or Greenwich Mean Time (GMT).

Subject Name

The subject field is of the same type (name) as the issuer identifier. The purpose of the subject name is to provide a unique identifier of the subject of the certificate. The syntax of the subject name is an X.500 Distinguished Name. According to X.509 v1 and v2 certificates the subject name must be present, only in a v3 certificate it may be left empty.

Subject Public Key Information

This field is used to carry the public key and identify the algorithm with which the key is used.

Unique Identifiers

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.

# 2 Standard Extensions of X.509

This section identifies standard certificate extensions defined.

Authority Key Identifier

The authority key identifier extension provides a means of identifying the particular public key used to sign a certificate. This extension would be used where an issuer has multiple signing keys (either due to multiple concurrent key pairs or due to changeover). In general, this extension should be included in certificates.

The identification can be based on either the key identifier (the subject key identifier in the issuer's certificate) or on the issuer name and serial number. The key identifier method is recommended in this profile. Conforming certification authorities that generate this extension shall include or omit both authorityCertIssuer and authorityCertSerialNumber. If authorityCertIssuer and authorityCertSerialNumber are omitted, the keyIdentifier field shall be present. This extension shall not be marked critical.

Key Identifier

The subject key identifier extension provides a means of identifying the particular public key used in an application. Where a reference to a public key identifier is needed (as with an authority key identifier) and one is not included in the associated certificate, a hash (SHA-1) of the subject public key shall be used. The hash shall be calculated over the value (excluding tag and length) of the subject public key field in the certificate. This extension should be marked non-critical.

Key Usage Restrictions

The key usage extension defines the purpose (e.g., encipherment, signature, certificate signing) of the key contained in the certificate. The usage restriction might be employed when a multipurpose key is to be restricted (e.g., when an RSA key should be used only for signing or only for key encipherment). The profile recommends that when used, this should be marked as a critical extension.

Private Key Validity

The private key usage period extension allows the certificate issuer to specify a different validity period for the private key than the certificate. This extension is intended for use with digital signature keys. This extension consists of two optional components notBefore and notAfter. The private key associated with the certificate should not be used to sign objects before or after the times specified by the two components, respectively. Certification authorities conforming to this profile shall not generate certificates with private key usage period extensions unless at least one of the two components is present.

Certificate Policies

The certificate policies extension contains a sequence of policy information terms, each of which consists of an object identifier (OID) and optional qualifiers. These policy information terms indicate the policy under which the certificate has been issued and the purposes for which the certificate may be used.

Policy Mappings

This extension is used in certification authority certificates. It lists pairs of object identifiers; each pair includes an issuerDomainPolicy and a subjectDomainPolicy. The pairing indicates the issuing certification authority considers its issuerDomainPolicy equivalent to the subject's certification authority's subjectDomainPolicy.

The issuing certification authority's users may accept an issuerDomainPolicy for certain applications. The policy may accept an issuerDomainPolicy for certain applications. The policy mapping tells the issuing certification authority's users which policies associated with the subject certification authority are comparable to the policy they accept.

This extension may be supported by certification authorities and/or applications, and it is always non-critical.

Subject Alternative Name

The subject alternative names extension allows additional identities to be bound to the subject of the certificate. Defined options include an RFC 822 name (e-mail address), a DNS name, an IP-address, and a uniform resource identifier (URI). Other options exist, including completely local definitions. Multiple instances of a name and multiple name forms may be included. Whenever such identities are to be bound into a certificate, the subject alternative name (or issuer alternative name) extension shall be used. A form of such an identifier may also be present in the subject distinguished name; however, the alternative name extension is the preferred location for finding such information.

Further if the only subject identity included in the certificate is an alternative name form (e.g., an e-mail address), then the subject distinguished name form (e.g., an e-mail address), then the subject distinguished name shall be empty (an empty sequence), and the subjectAltName extension shall be present. If the subject field contains an empty sequence, the subjectAltName extension shall be marked critical.

Where the subjectAltNAme extension contains a uniformResouceIdentifier, this name the following semantics shall be assumed: The URI is a pointer to a sequence of certificates issued by this certifica-

tion authority (and optionally other certification authorities) to this subject.

The URI must be an absolute, not relative, pathname and must specify the host. This specification recognises the following values for the URI scheme: ftp, http, LDAP, and mailto. The mailto scheme indicates that mail sent to the specified address will generate an e-mail response (to the sender) containing the subject's certificates. No message is required. If the URI scheme is ftp, then the information is available through anonymous ftp. If the URI scheme is http or LDAP, then the information may be retrieved using that protocol.

Issuer Alternative Name

As with the subject alternative name, this extension is used to associate Internet style identities with the certificate issuer. If the only issuer identity included in the certificate is an alternative name form (e.g., an e-mail address), then the issuer distinguished name shall be empty (an empty sequence), and the issuerAltName extension shall be present. If the subject field contains an empty sequence, the issuerAltNAme extension shall be marked critical.

Subject Directory Attributes

This extension is always non-critical.

Basic Constraints

The basic constraints extension identifies whether the subject of the certificate is a certification authority and how deep a certification path may exist through that certification authority. This profile requires the use of this extension, and it shall be critical for all certificates issued to certification authorities.

Name Constraints

The name constraints extension provides permitted and excluded subtrees that place restrictions on names that may be included within a certificate issued by a given certification authority. Restrictions may apply to the subject distinguished name or subject alternative names. Any name matching a restriction in the excluded subtrees field is invalid regardless of information appearing in the permitted subtrees. This extension may be critical or non-critical.

Restrictions for the RFC 822, dNSName, and URI name forms are all expressed in terms of strings with wild card matching.

Policy Constraints

The policy constraints extension can be used in certificates issued to certification authorities. The policy constraints extension constrains path validation in two ways. It can be used to prohibit policy mapping or limit the set of policies that can be used in subsequent certificates. This extension may be critical or non-critical.

Certificate Revocation List Distribution Points

The certificate revocation list distribution points extension identifies how certificate revocation list information is obtained. The extension shall be non-critical, but this profile recommends support for this extension by certification authorities and applications.

# B Possible Content Fields of Certificates and Revocation Lists.

## 1 Generic Content Fields of Certificates

Authority

Grants the subject all the authority (access rights) of the issuer. It is anticipated that this field will be used to delegate permissions to a temporary signing key or to indicate members of a group, where the issuer is a group. There may be permissions granted like ftp – to use ftp into a host; other grants as protocols like http, telnet etc. are possible.

Bundle

Provided for someone who wants to bundle a set of public keys and certificate bodies and signatures into one expression for communication.

Comment

May contain any text or other strings the issuer desires. These strings may have display instructions.

Content

Content of a certificate – may be an entire document or a authorisation of the subject.

Date

ASCII byte string.

Delegation

A modifier to the authority field, to whom the authority may be given – may be in a restricted way.

Dual Signatures

This field makes explicit, the subject accepts the issuance of the given certificate by signing the certificate not only by the issuer but by the subject too. Otherwise there is the potential problem that a certificate might be issued which the keyholder does not want. This may be an option.

Fully Qualified Name

A fully qualified (SDSI) name; specifies a key (and therefore its name space) for the first name in the list. That first name can then be resolved to a key and the process recurses until the fully qualified name is reduced to a key.

Hash

Identifier for the hash algorithm (name) used for the digital signature process and the hash value of some object hashed like a document etc. in this case the same hash algorithm is used.

There may be applications without hashing objects.

Issuer

The issuer of the certificate – may be indicated by the issuer's public key or its hash.

Issuer Location

Location of the issuer's public key – it gives a network address of a web page or server which can return the issuer's certificate authorising the current certificate.

Online

Validity option gives the location of a remote service which will pass judgement on the current validity of a certificate.

Ref

A relative name. It is converted to a fully qualified (SDSI) name by using the issuer of the certificate of which it is a part as the key defining the name-space.

Secret Signature Key

Gives the secret key values for computing a message authentication code (MAC), keyed hash or encrypted objects, to permit a symmetric algorithm to be used for signatures. This is expected to be faster than a public key algorithm, but is limited to cases where the signer and the verifier have mutual trust.

Signature

Digital signature of the entire certificate body by the issuer.

Subject

The subject is the entity holding the certificate. It may be a public key or its hash, a fully qualified name, or a relative name. It may also be secret key or the hash of an object.

Subject Location

Location of the subject's public key. Provides the location of a network resource where one can learn about the subject: Find the subject's public key, the home page etc.

Validity

Time until the certificate expires, needs to be revalidated or needs a certificate revocation list update. There may be a renew location field, where to get a revalidation certificate revocation list. The validity period can give the lifetime in seconds.

There may be not-after or not-before fields also. These dates are normal validity dates for a certificate.

Version

Version number of the certificate type.

## 2 Generic Content Fields of Certificate Revocation Lists

The certificate revocation list (CRL) contains a header and a sequence of revocation entries. The certificate revocation list is signed by the certification authority and published and distributed on a periodic basis.

The reason of revocation may be one of the following: Key compromise, certification authority compromise, changed affiliation, supersede by new certificate, termination and other.

CRL entry extensions

Sequence of fields pertaining to a specific CRL entry.

CRL extensions

Sequence of fields pertaining to the whole CRL.

Hold Instruction CodeIssuer

Distinguished name of the certification authority responsible for this CRL.

Issuer Name

The issuer name identifies the entity who has signed (and issued the CRL).

Issuing Distribution Point

The CRL Distribution Point is used in defining entries for objects which act as CRL Distribution Points.

Next Update

This field indicates the date by which the next CRL will be issued. The next CRL could be issued before the indicated date, but it will not be issued any later than the indicated date.

Revocation Date

Date of revocation of this certificate. May be a field or the field revoked certificates.

Revoked Certificates

Revoked certificates are listed. The revoked certificates are named by their serial numbers. Certificates are uniquely identified by the combination of the issuer name or issuer alternative name along with the user certificate serial number. The date on which the revocation occurred is specified. When a CA wishes to revoke a certificate that it issued to another CA, the revocation shall appear on the CRL.

Serial number

Number of the revoked certificates. May be a field or the field revoked certificates.

Signature

Identifier for specifying the signature algorithm and associated hash function used to sign the CRL. Digital signature of the revocation list by the issuer.

This Update

This field indicates the issue date of this CRL.

Version

This field describes the version of the encoded CRL.

CRL Extensions

| | |
|---|---|
| Authority Key Identifier: | The authority key identifier extension provides a means of identifying the particular public key used to sign a CRL. The identification can be based on either the key identifier (the subject key identifier in the CRL signer's certificate) or on the issuer name and serial number. This extension would be used where an issuer has multiple signing keys, either due to multiple concurrent key pairs or due to changeover. In general, this non-critical extension should be included in certificates. |
| Issuer Alternative Name: | The issuer alternative names extension allows additional identities to be associated with the issuer of the CRL. |
| CRL Number: | The CRL number is a non-critical CRL extension which conveys a monotonically increasing sequence number for each CRL issued by a given CA through a specific CA Directory entry or CRL distribution point. This extension allows users to easily determine when a particular CRL supersedes another CRL. |
| Issuing Distribution Point: | The issuing distribution point is a critical CRL extension that identifies the CRL distribution point for a particular CRL, and it indicates whether the CRL covers revocation for end entity certificates only, CA certificates only, or a limited set of reason codes. Since this extension is critical, all |

certificate users must be prepared to receive CRLs with this extension. The CRL is signed using the CA's private key. CRL Distribution Points do not have their own key pairs. If the CRL is stored in the Directory, it is stored in the Directory entry corresponding to the CRL distribution point, which may be different to that the Directory entry of the CA.

Delta CRL Indicator: The delta CRL indicator is a critical CRL extension that identifies a delta-CRL. The use of delta-CRLs can significantly improve processing time for applications which store revocation information in a format other than the CRL structure. This allows changes to be added to the local database while ignoring unchanged information that is already in the local database. When a delta-CRL is issued, the CAs shall also issue a complete CRL. The value of BaseCRLNumber identifies the CRL number of the base CRL that was used as the starting point in the generation of this delta-CRL. The delta-CRL contains the changes between the base CRL and the current CRL issued along with the delta-CRL. It is the decision of a CA as to whether to provide delta-CRLs.

CRL Entry Extensions: The CRL entry extensions already defined by ANSI X9 and ISO for X.509 v2 CRLs [X.509-AM] [X9.55] provide methods for associating additional attributes with CRL entries. The X.509 v2 CRL format also allows communities to define private CRL entry extensions to carry information unique to those communities. Each extension in a CRL entry may be designated as critical or non-critical. A CRL validation must fail if it encounters an critical CRL entry extension which it does not know how to process. However, an unrecognised non-critical CRL entry extension may be ignored.

Reason Code: The reason Code is a non-critical CRL entry extension that identifies the reason for the certificate revocation.

Hold Instruction Code: The hold instruction code is a non-critical CRL entry extension that provides a registered instruction identifier which indicates the action to be taken after encountering a certificate that has been placed on hold.

Invalidity Date: The invalidity date is a non-critical CRL entry extension that provides the date on which it is known or suspected that the private key was compromised or that the certificate otherwise became invalid. This date may be earlier than the revocation date in the CRL entry, but it must be later than the issue date of the previously issued CRL.

*** *** ***

# C Guidelines on Naming

## Foreword

Guidelines on naming require that the system whose objects should be named is clearly defined and it`s technical implementation details are well known. This is not yet the case for a European Trust Infrastructure. Therefore these guidelines are structured based on generally accepted functions identified in the main part of this study. This approach will allow to adapt these guidelines to a future European Trust Infrastructure if the details of its architecture become available.

It is also assumed that the procedures for assigning names in message exchange systems (Internet electronic mail and X.400 MHS systems) and EDI/EDIFACT systems are well established and neither have to nor can be changed on short notice and therefore do not have to be covered here.

These guidelines therefore have their emphasis in the area of an infrastructure that provides trust by using certificates for validating public keys and that will be based on the use of X.509 v3 certificates [1], the X.500 directory concept [2], and X.520 for the generation of names [3].

## 0 Table of Contents



## 1 Introduction

The most necessary prerequisite to be able to use modern telematic services securely is the availability of an infrastructure for an efficient directory service and for the management of public keys. In order to use the full benefits of these services a system for naming is necessary that allows the different users of the telematic services to be able to identify each other reliably. Conventions for the naming of objects are needed which help to define and manage the namespace in such an infrastructure in a simple and efficient way.

A name is not the only means by which an object can be located, an address is equally suitable for that purpose. However conceptually there is a clear distinction between the two concepts. Whereas a name is used to denote an object, an address specifies where this object can be found. In theses guidelines the expression „name“ is used for both, name and a combination of a name and location information.

The guidelines on naming are based on the following prerequisites:

- a future European Trust Infrastructure will consist of co-operating trusted authorities which will be positioned on different levels of a hierarchical topology,
- the naming service will be based on a directory service which will use X.500 concepts,
- naming will be based on X.520 concepts.

If these conditions change, the guidelines have to be adapted accordingly.

In paragraph 7 general steps in the generation process of names are described and recommendations are

given which should be considered in that process. The details of the day to day execution of these steps and the feasibility of the recommendations depends very much on the tools being used for the support of the naming process. Both have to be adapted and detailed accordingly if the tools become known.

# 2 Purpose and Audience

These guidelines on naming are intended to support a standardised way of assigning and managing names. They build on concepts defined in the X.500 series of standards and try to adapt them to the requirements of an envisioned European Trust Infrastructure.

The guidelines are targeted primarily towards persons in functions that assign and manage names in a European Trust Infrastructure and who are responsible for assuring that assigned names are unique. They can also be used by those who are responsible for establishing policy for naming.

# 3 Definitions

Definitions are listed at Annex 8.1

# 4 General Properties of a Naming Service

A naming service has the following general properties:

- it allows to name a broad spectrum of objects in different applications as well as in different environments,
- it allows to identify an object unambiguously,
- it is user-friendly, i.e. names can be constructed easily, can be understood world-wide and are easy to remember,
- it allows for a simple management of the name space
- it allows to apply the same naming conventions for all name assignments and to use standard attributes for name design,
- it supports the end user in that technical details like conversions to internal character sets or translation to other name formats are hidden from him,
- it is flexible enough to be able to interoperate with other name schemes.

# 5 Functions in the Naming Process

The major functions that are involved in the naming process are specified for two levels of authority, the policy level and the execution level. The decision to which distinct authorities these functions will be assigned has to be made when establishing the infrastructure.

## 5.1 Policy Level Functions

### 5.1.1. Development of Name Assignment Policy

The development of a name assignment policy requires the following activities:

- Specification of the name space for the area of responsibility,
- Specification for the treatment of alias names and users who want to be anonymous,
- Co-ordination of the specified name space with other areas of responsibility,
- Specification of the credentials and procedures required for identity validation,
- Specification of protocols for the identity validation and for the name assignment process,
- Specification of procedures for name distribution,
- Agreement to or specification of exceptions from naming conventions.

### 5.1.2 Monitoring the Execution of Name Assignment Policy

The monitoring of the execution of the name assignment policy requires the following activities:

- Specification of procedures for name co-ordination,
- Specification of procedures for the detection of name conflicts and deviations from naming conventions,
- Specification of procedures for the resolution of name conflicts.

## 5.2 Execution Level Functions

### 5.2.1 Registration

For the registration of users the following activities are necessary:

- Reception of an application for a public/private key pair,
- Validation of the claimed identity of the applicant,
- Registration and enrolment of applicant.

### 5.2.2 Assignment of Names

The assignment of names comprises the following activities:
- Generation of name (as described in para. 7),
- Assignment of name to public key,
- Registration of the name/public key pair.

### 5.2.3 Management of Names,

The management of names requires the following activities:
- Assurance of uniqueness of names,
- Detection of deviations from naming conventions,
- Maintenance of repository of assigned names (Directory),
- Revocation and deletion of names,
- Distribution of names,
- Resolution of name conflicts.

### 5.2.4 Support for Name Management

To support the function of name management the following mechanisms might be used:

| | |
|---|---|
| Technically | - by software that screens the Directory Information Base periodically for naming conflicts and that supports the selection of new names by providing an up-to-date overview of already assigned names, |
| | by implementing a directory. |
| Organisationally | - by establishing structures which allow for decentralisation of the naming process while maintaining uniqueness of names (e.g. tree structures), |
| Procedurally | - by setting up rules that avoid duplication of names (like the use of the nameConstraint in the X.509 v3 extensions), |
| | - by establishing an information exchange process on assigned names between different naming authorities in an infrastructure. |

# 6 Organisations in the Naming Process

The uniqueness of names is a fundamental prerequisite for the correct operation of a certification scheme for public keys. To fulfil this requirement a considerable administrative effort and organisational elements dedicated to that task are needed. In a European Trust Infrastructure naming authorities are trusted to take this responsibility, they have to be established at different levels of an infrastructure, for example at international, national, corporate or department level.

In general a naming authority is responsible for the allocation of names in a specific domain based on agreed naming conventions, which determine the syntax and the semantics of names. The definition of such domains can be based on geographical, technical, sectorial or other appropriate criteria.

Technically speaking, a naming authority has control over some part of the structure of a data repository. In a hierarchically structured X.500 Directory database for example, this means that the naming authority has been assigned control over some region of the Directory Information Tree.

Naming authorities might be co-located with other authorities and might reside at different levels of the certification scheme (technically: of the Directory Information Tree). The level at which a naming authority is established determines which of the functions in the naming process it has to perform.

For a European Trust Infrastructure it is assumed that there are three level of authorities for managing trust, an upper-level authority and several middle and lower level authorities. All three types will have functions in the naming process.

An upper-level naming authority has
- to ensure that the names of middle level naming authorities follow the naming conventions,
- to detect potential, unintended duplicate certification of the names of lower level naming authorities and it has to provide this information to the middle level naming authorities. This information is the basis for ensuring global uniqueness of the names of lower level naming authorities.

A middle level naming authority has
- to specify the policy and procedures which govern the naming of lower level naming authorities it certifies, and how this policy applies transitively to entities (end-users or subordinate lower level authorities) certified by these lower level naming authorities.
- to state what procedure has to be used to verify the claimed identity of a lower level naming authority,
- to specify the requirements and mechanisms that have to be used by lower level naming authorities to validate the identity of end-users,
- to specify the procedures used to resolve name conflicts.

A lower level naming authority has
- to validate an end-user's claimed identity,
- to assign names to end-users based on a name space given by the middle-level naming authority,
- to maintain a database of the names which it has certified and to take measures to ensure that it does not certify duplicate names for users or subordinate lower level naming authorities,

- to implement procedures to ensure that the same subject name isn't issued to multiple users in case of issuance of „PERSONA certificates" (certificates for users who wish to hide their identity).

Registration Authorities

The function of user registration is performed by registration authorities. The details of user registration including the decision which organisational element should fulfil that function are a local matter, subject to policies established by the user's lower level naming authority and the middle level naming authority under which that authority has been certified. In general a user must provide, at a minimum, his public key and a name to a lower level naming authority, or a representative thereof, for inclusion in the user's certificate. The lower level naming authority will specify procedures and credentials (e.g. birth certificate, personal ID card, notarial attestation, drivers licence) in accordance with the policy of its middle level naming authority, to validate the user's claimed identity and to ensure that the public key provided is correctly associated with the user whose name is to entered into the certificate.

# 7 Generation of Names

Names in certificates, from a user point of view, should be descriptive, i.e. they should clearly indicate the object in the real world to whom the public key in the certificate belongs and who has signed the certificate. Naming conventions establish the rules which determine the syntax and semantics of names and therefore provide the basis for the fulfilment of these requirements. Because the distinguishability of objects depends largely on the uniqueness of their names, a general, systematic, and practical method of naming is necessary.

The following paragraphs describe the main activities associated with name generation. The described steps and recommendations are independent from a specific implementation. If the implementation is known they have to be adapted accordingly.

## 7.1 Development of the Name Structure

The X.500 Directory uses a database (Directory Information Database, DIB) to store information about objects of interest and to provide access to them. An object can be anything that is identifiable. Typically, it is a person, an application-entity, a file, a distribution list etc. The Directory Information Base is composed of entries, each of which consists of a collection of information on one specific object and there is precisely one object entry which represents a object.

In order to allow for an efficient management and a simple distribution of a very large Directory Information Base, and to ensure that entries can be unambiguously named and rapidly found, a hierarchical structure of the database has been selected. If the entries are arranged according to this model the result is a tree-like structure which is called Directory Information Tree (DIT), where, by convention, the root of the tree is at the top and the nodes in the tree represent the entries (Figure 75).

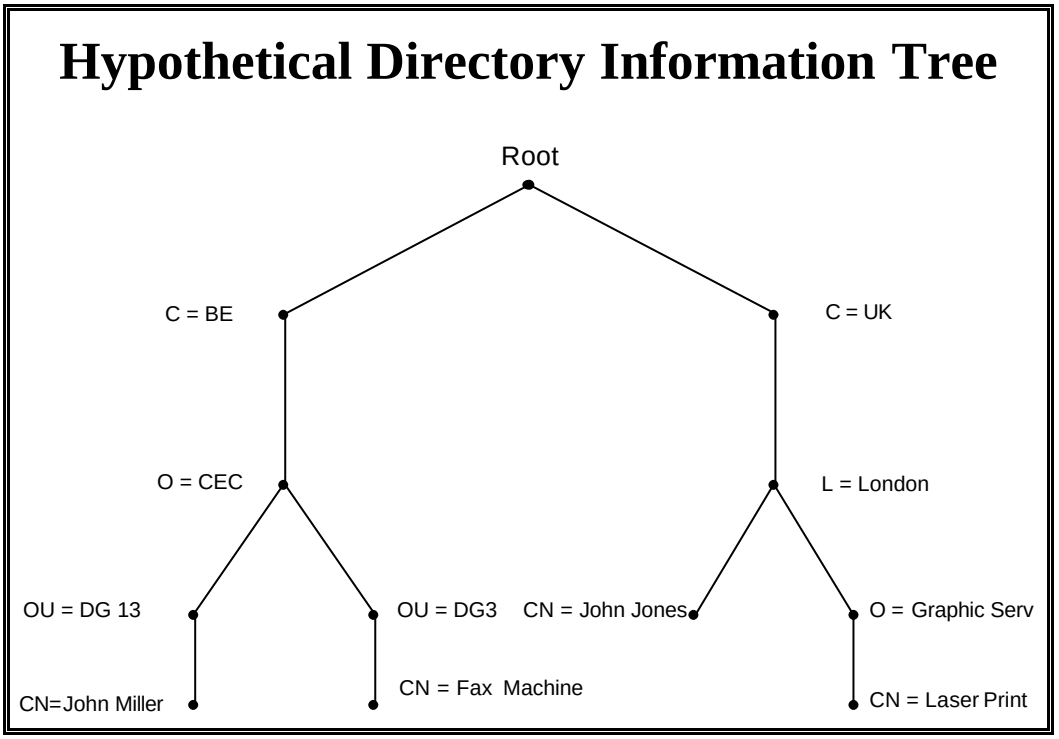


Figure 74: Hypothetical Directory Information Tree

To establish this structure, which is also the basis for the generation of each entry's name, the position of the entries in the hierarchy should be derived from the relationships between the objects they represent in the real world. For example in Figure 75 the left branch means that

John Miller works for Directorate XIII of the European Commission

Object entries can either be located at the nodes or the leaves of the Directory Information Tree. Entries higher in the tree will generally represent objects such as countries or organisations, while entries lower in the tree will represent people or application processes.

Each entry in the Directory Information Base stores and maintains its information about an object in a set of attributes and each attribute consists of an attribute type which can have one or more values.

Example:

An entry containing information about John Miller might look as follows:

| Common Name | John Miller | J. Miller | J.B. Miller | Johnny Miller |
|---|---|---|---|---|
| Surname | Miller | | | |
| Telephone Number | +32 2 296 999 | | | |
| Postal address | 9, Avenue Beaulieu | | | |
| Description | Head of Unit | | | |
| Object Class | Organisational Person | | | |

Figure 75. Example of a Directory entry

The entry in Figure 76 consists of six attributes, the attribute type „common name“ has four values which all describe different name forms for the same person.

The following list shows some additional examples for attribute types and their values, it also gives examples for the notation using attribute labels that are used as a shorthand expression for the attribute type:

| **Attribute Type** | **Attribute Value** | **Attribute Label** | **Example** |
|---|---|---|---|
| country name | BE; DE;ES; FR; GB | C | C=BE |
| locality name | Brussels | L | L=Brussels |
| organisation name | European Commission | O | O=CEC |
| organisational unit name | DG XIII | OU | OU=DG XIII |

The set of attribute types which are allowed for an entry is open ended, however there is a set of attribute types which is internationally standardised [3], others might be defined by national administrative authorities and/or private organisations.

### General Steps:

1. Identify participating objects,
2. Determine their working relationship,
3. Determine necessary attributes for objects,
4. Design DIT,
5. Co-ordinate DIT with responsible authorities,
6. Document and implement DIT.

### Recommendations for designing the DIT:

1. Base work on an organisational structure known to the future users,
2. Make an early decision whether to base the structure on organisational or geographical aspects,
3. Do not use preliminary organisational structures,
3. Avoid too many levels of hierarchy,
4. Consider stability and mobility of the organisation,
5. Use a structure that avoids name clashes.

## 7.2 Common Names

A common name is a name which is commonly known in some limited environment, such as an organisation, and which therefore might be ambiguous. It is expressed according to the naming conventions of the country or culture with which the name holder is associated. It specifies an identifier of an object, however it is no directory name.

The format for the inclusion of common names for persons into the Directory Information Base should

follow the cultural conventions as closely as possible. Deviations should be specified and justified in the name assignment policy. The recommendations of the World Electronic Message Association (WEMA) on naming of persons [4] should be followed.

The syntax for names is described as follows:

G=FirstName;S=Surname;I=Initials

FirstName and Surname are mandatory fields, Initials is optional.

**General Steps:**

1. Specify name based on organisational naming policies or extract it from required credentials,
2. Implement and document common name.

**Recommendations:**

1. Use syntax and semantics of respective culture to express common names as far as possible,
2. If necessary, adapt name to database structure.

## 7.3 Relative Distinguished Names (RDNs)

A Relative Distinguished Name is a name that identifies a particular entry in the Directory Information Base and that is primarily used to support its organisation and administration. Each entry in the Directory Information Base has a Relative Distinguished Name except the root entry. The allocation of a Relative Distinguished Name is an administrative task, that is performed when the entry that represents an object is registered for the first time in the Directory Information Base.

When specifying a Directory Information Base, each entry has to be classified according to the specific characteristics of the object it represents by selecting an object class from a set of object classes [5]. This object class determines the mandatory and optional attributes associated with that entry.

Examples for object classes are:

| Object Class | Entry representing |
|---|---|
| Country | countries |
| Organisation | organisations |
| Locality | localities or regions |
| Organisational Person | people employed by an organisation |

Examples for object classes and related attribute types are:

| Object Class | Attributes |
|---|---|
| Country | country name (mandatory)<br>description (optional) |
| Organisation | organisation name (mandatory)<br>business category (optional)<br>description (optional) |
| Organisational person | common name (mandatory)<br>surname (mandatory)<br>organisational unit name optional)<br>telephone number (optional)<br>title (optional) |

Figure 76: Examples for Object Classes and related Attributes

At least one attribute type and value (distinguished value) of the entry is used to specify a Relative Distinguished Name for an entry.

For example the Relative Distinguished Name of an entry with the object class „country“ could be:

C=BE

where „C“ is a label for the country name attribute and „BE“ is the value for the country name attribute expressed as a two letter code, taken from International Standard ISO 3166 [6].

Basically it is possible to use more than one attribute/value pair to specify the Relative Distinguished Name of an entry, but this feature is rarely being used because it would lead to directory names that are not user-friendly.

### General Steps:

1. Determine object class for entry,
2. Select optional attributes for entry,
3. Select attribute/value pair which should determine RDN,
4. Generate RDN,
5. Check for uniqueness of RDN under immediately superior entry,
6. Co-ordinate RDN with other naming authorities,
7. Assign RDN to entry,
8. Implement and document RDN.

### Recommendations:

1. Use standard attributes as far as possible,
2. If using non standard attributes make them user friendly,
3. Select attribute values that are short and familiar to users,
4. Consider language aspects, if relevant (international operations),
5. Avoid multi valued RDNs, (RDN consisting of two value/pairs),
6. Consider data protection aspects.

## 7.4 Directory Names

The example of an entry in Figure 2 shows a situation where an object possesses a common name attribute which has several attribute values and a surname attribute which has only one attribute value. All of these represent names by which the object is known to its community and by which the entry could be accessed. However in order to be able to unambiguously identify a specific entry, each entry also is given a directory name. A directory name in X.500 unambiguously identifies a specific entry in the Directory Information Base (DIB).

There are two types of directory names:
- Distinguished Names (DN) and
- Alias names.

### 7.4.1 Distinguished Names (DN)

A Distinguished Name identifies an object and its entry within the Directory Information Base unambiguously and uniquely. These properties are derived from the tree structure in which the entries are arranged. A Distinguished Name of a given object is defined as that name which consists of the concatenation of all Relative Distinguished Names from the root down to and including the entry itself. Every object entry has exactly one Distinguished Name

An example for a Distinguished Name of an entry where the hierarchy includes the RDNs (written in extended form) Country=Belgium, Organisation=European Commission, Organisational Unit=DG13, and Common Name=John Miller, the Distinguished Name would be (written in abbreviated form):

C=BE;O=CEC;OU=DG13;CN=John Miller.

Figure 78: Determination of Distinguished Names visualises the concepts of Relative Distinguished Names and Distinguished Names.

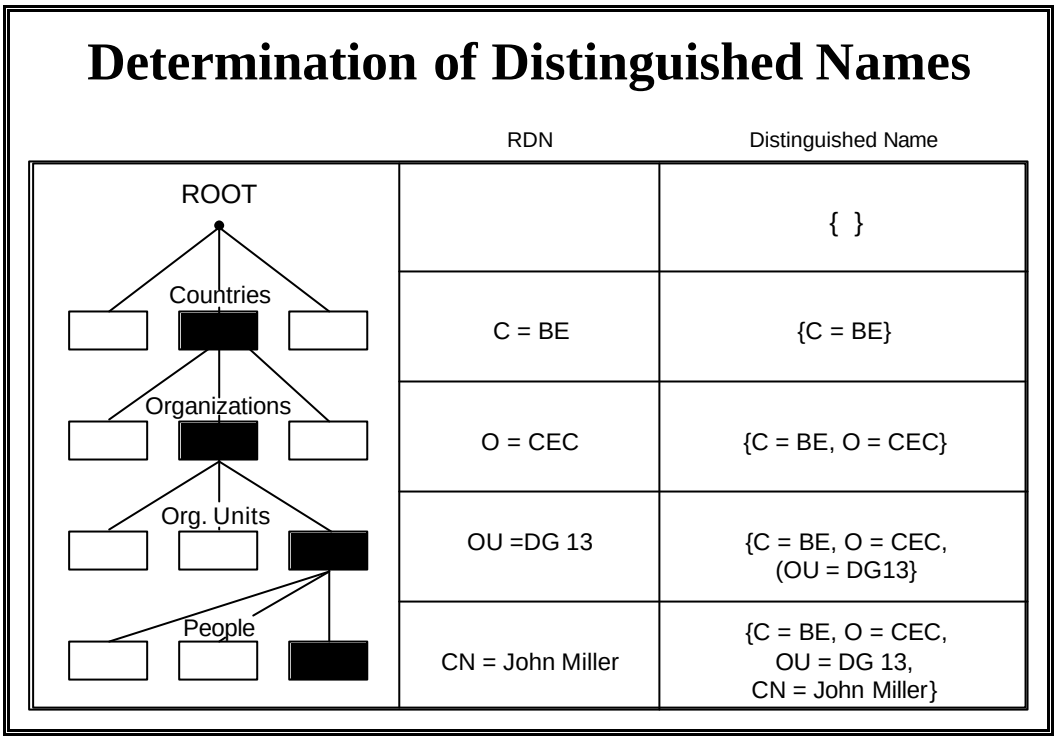


Figure 77: Determination of Distinguished Names

**General Steps:**

1. Define RDN for entry to be named,
2. Determine RDNs of entries on the way from root to entry to be named,
3. Concatenate RDNs to form DN of entry,
4. Assign DN to entry,
5. Implement and document DN,
6. Inform object associated with entry about DN.

**Recommendations:**

1. Consider data protection aspects.

### 7.4.2 Alias Names

Although each entry has one Distinguished Name, it may have several alias names. An entry therefore can be referred to by either using its Distinguished Name or by one of its alias names. Syntactically an alias name looks like a Distinguished Name, however it does not represent the direct path from the root of the Directory Information Tree to an entry. The path leads to an alias entry, and the alias entry has an attribute that contains the name of the entry to which the alias refers.

**General Steps:**

1. Define RDN for alias entry to be named,
2. Determine RDNs of entries on the way from root to alias entry,
3. Concatenate RDNs to form DN of alias entry,
4. Assign DN to alias entry,
5. Define attribute in alias entry that contains directory name leading to entry to which alias name refers,
6. Implement and document DN,
7. Inform object associated with alias entry about DN.

**Recommendations:**

1. Select common name attribute value that is easy to remember,
2. Consider language aspects (if relevant),
3. Consider data protection aspects.

## 7.5 Conventions

### 7.5.1 Syntax of Names

The syntax of Relative Distinguished Names, Distinguished Names and alias names is defined by the syntax of the attribute types and their possible values.

#### 7.5.1.1 Attributes Types

The set of attribute types is open ended. However there is a standard set defined in [3] which should be used as far as possible.

The definition of new attributes for special sectors or application areas is possible, however they are not automatically available outside the area of responsibility of the authority who defined them.

#### 7.5.1.2 Attribute Values

The basic syntax for attribute values is defined in [3], these definitions provide the capability for user friendly naming, e.g. taking values from the working environment of the user. The values for the Country Name attribute are defined by [6].

#### 7.5.1.3 Value Assignment

The assignment of values to attribute types is expressed by the „=“ character.

### 7.5.2 Sequence of Names

Distinguished Names are sequence sensitive in that their structure maps the sequence of the Relative Distinguished Names from the root to the entry which is named.

The character that is used to separate the Relative Distinguished Names which form a Distinguished Name is dependent of the implemented product (e.g. „space“, „slash“).

# 8 Annexes

## 8.1 Definitions

### Alias name

An alternative name for an object, provided by the use of alias entries (ITU-T Rec. X.501|ISO/IEC 9594-2)

### Attribute

Information of a particular type. Entries are composed of attributes (ITU-T Rec. X.501|ISO/IEC 9594-2).

### Attribute type

That component of an attribute which indicates the class of information given by the attribute (ITU-T Rec. X.501|ISO/IEC 9594-2).

### Attribute value

A particular instance of the class of information indicated by an attribute type (ITU-T Rec. X.501| ISO/IEC 9594-2).

### Entry name

A construct that singles out a particular entry from all other entries. (ITU-T Rec. X.501|ISO/IEC 9594-2).

### Common name

A common name attribute specifies an identifier of an object. (ITU-TRec. X.520|ISO/IEC 9594-6).

### Directory name

A construct that singles out a particular object from all other objects. A name shall be unambiguous (that is, denote just one object), however it need not to be unique (that is, be the only name which unambiguously denotes the object) (ITU-T Rec. X.501|ISO/IEC 9594-2)

### Directory Information Base (DIB)

The set of information managed by the Directory (ITU-TRec. X.500|ISO/IEC 9594-1).

### Distinguished Name

The name of an entry which is formed from the sequence of the RDNs of the entry and each of its superior entries. Every object entry, alias entry and subentry has precisely one distinguished name (ITU-T Rec. X.501|ISO/IEC 9594-2)

### Naming authority

An authority responsible for the allocation of names in some region of the DIT (ITU-T Rec. X.501| ISO/IEC 9594-2)

### Object

Anything in some „world“, generally the world of communications and information processing or some part thereof, which is identifiable (can be named), and which is of interest to hold information on in the DIB (ITU-T Rec. X.501|ISO/IEC 9594-2).

### Object class

An identified family of objects (or conceivable objects) which share certain characteristics. (ITU-T Rec. X.501|ISO/IEC 9594-2).

### Relative Distinguished Name

A set of one or more attribute type and value pairs, each of which matches a distinct distinguished attribute value of the entry. (ITU-T Rec. X.501|ISO/IEC 9594-2).

## 8.2 List of relevant Standards


| | | |
|---|---|---|
| 1. | ITU-T Rec. X.509\|ISO/IEC 9594-8 | Information technology - Open Systems Interconnection - The Directory Authentication Framework. |
| 2. | ITU-T Rec. X.500\|ISO/IEC 9594-1 | Information technology - Open Systems Interconnection - The Directory: Overview of concepts, models and services. |
| 3. | ITU-T Rec. X.520\|ISO/IEC 9594-6 | Information technology - Open Systems Interconnection - The Directory: Selected Attribute Types. |
| 4. | WEMA | X.400/Internet Personal Naming Recommendation, Anaheim 1 May 1996. |
| 5. | ITU-T Rec. X.521\|ISO/IEC 9594-7 | Information technology - Open Systems Interconnection - The Directory: Selected Object Classes. |
| 6. | ISO 3166 | Codes for the representation of names of countries. |

# D Guidelines for the Management of Certificates in a ETI

The European Trust Infrastructure shall be governed by the following aspects.

## 1 Authorities

There shall be a policy approving authority granting authorisation to all authorities to act like policy certification authorities, certification authorities, registration authorities etc. Such authorisation may be revoked.

The policy approving authority shall establish rules governing the terms under which such authorities shall be granted, and promulgate regulations for the operation (operation policy) of the authorities.

The policy approving authority shall evaluate the policies of the subordinate authorities.

Authorised certification authorities may issue certificates to natural and legal persons.

Authorised certification authorities may offer or facilitate registration and time stamping of the transmission and reception of documents as well as other functions regarding communications secured by means of certificates.

## 2 Registration

A registration authority is only allowed to request such information as is necessary to identify and authenticate the user. This information structure shall be delivered by the certification authority by way of a certification practice statement.

## 3 Keys

Users shall have different key parameters from their certification authority.

The holder of a certified key pair may revoke the keys and the corresponding certificates. The revocation becomes effective from the time it is registered by the certification authority.

The holder of a certified key pair is under obligation to revoke the corresponding certificate where the holder learns that the private key has been lost, compromised or is in danger of being misused in other respects. If the holder fails to revoke the certificate in such a situation, the holder is liable for any loss sustained by third parties having relied on the corresponding certificates.

There shall be an unregulated registry of crypto algorithms usable for digital signatures.

## 4 Certificates

The ETI must provide users with the following capabilities:

- Create, post, and distribute certificates.
- Revoke certificates.
- Obtain, interpret, and verify certificates.

The certificates issued by the authorised certification authorities shall indicate at least:

- The user's name – may be attributed by sufficient identification data as address, date and place of birth if the user is a natural person. If the user is a legal person, the name of the legal person and any relevant information for identifying that legal person.
- The name, address or place of business of the certificate issuing certification authority.
- The user's public key.
- Any necessary information indicating how verification of the user's public key is available to the recipient of the digital signature given according to the certificate.
- The serial number of the certificate.
- Some validity information: The date of issuance, the date of expiry, the validity period of the certificate.

Legal persons shall be identified by the public key certified for that legal person.

Revoked certificates shall be published by a certificate revocation list. Such a list may be pushed to or pulled by the users.

There may be black lists useful, listing users whose keys have been revoked.

Certificates shall be verifiable by requesting a hierarchically higher authority or a cross-certified authority in several steps depending on the requested level of trust.

Upon request from legal or natural persons, the certification authority shall deliver information about the following:

- The conditions under which the certificate may be used.
- The conditions associated with the use of digital signatures.

- The policy and practices of the PAA; PCA and especially the certification authority with respect to the use, storage and communication of personal information.
- The technical requirements of the certification authority with respect to the user's communication equipment.
- The conditions under which warnings are given to users by the certification authority in case of irregularities or faults in the functioning of the trust infrastructure.
- All the limitations of the liability of the certification authority.
- Any restrictions imposed by the certification authority on the use of the certificate.
- The conditions under which the user is entitled to place restrictions on the use of the certificate.

Subject to notice, the user and the certification authority may terminate the agreement for connection to the certification authority. Such notice of termination takes effect when received.

# 5 Directory

Authorised certification authorities shall keep a publicly accessible directory of certificates issued, indicating when the individual certificate was issued, when it expires or when it was revoked. The directory shall be maintained by the certification authority until a (limited) number of years after the date of revocation or expiry of the operational period of any certificate issued by that certification authority.

The certification authority shall be liable to any natural or legal person who has acted in good faith in reliance on a certificate issued by the certification authority for any circumstances. The liability shall be limited.

# 6 Interoperability

The infrastructure must:
- ensure interoperability within the ETI,
- set policy on certificates of other trust infrastructures and their use, and
- negotiate and set policy for interoperation with users of non-ETIs.

Certificates issued by certification authorities of other trust infrastructures shall be used if they are approved by an authorised certification authority of the ETI. The policy approving authority is authorised to lay down specific rules for such an approval.

# 7 Architecture

There shall be a hierarchical architecture of the policy approving authority and the policy certification authorities and the certification authorities.

For certificate verification purposes there shall be a hybrid architecture of a hierarchical architecture with a network architecture of certification authorities that permit cross-certification also.

The architecture shall allow on-line verification of certificates.