



**Institut für Telematik** unter Betreuung der  
**Fraunhofer Management GmbH**



**Preprint 00-15**

**Security in Open Networks:  
The Functionality of a  
Public Key Infrastructure**

Bernd Dusemund  
Torsten Becker  
Lutz Gollan  
Thomas Engel  
Christoph Meinel

ISSN 1433-8106

Authors	<p>Bernd Dusemund Dr. rer. nat.</p> <p>Torsten Becker Dipl.-Math. oec.</p> <p>Lutz Gollan Dr. iur.</p> <p>Thomas Engel Dr. rer. nat.</p> <p>Christoph Meinel Univ.-Prof. Dr. sc.</p>
Copyright	Institut für Telematik, Trier
Trademarks	All terms that are mentioned in this paper that are known to be trademarks or service marks have been appropriately capitalised. Use of a term in this paper should not be regarded as affecting the validity of any trademark and service mark. The product or brand names are trademarks of their respective owners.
Printing	12/2000
	<p>Printed in Germany All rights reserved</p> <p>The documentation was accomplished through the Institut für Telematik.</p> <p>The information contained in this document represents the current view of the authors on the issues discussed as of the date of publication. Because the present methodology must respond to changing research conditions, the results of this paper should not be interpreted to be a commitment on the part of the authors. Any information presented after the date of publication are subject to change.</p> <p>The right to copy this documentation is limited by copyright law. Making unauthorised copies, adaptations or compilation works without permission of the authors or institutions mentioned above is prohibited and constitutes a punishable violation of the law.</p>

## Table of Contents

1	Introduction .....	1
2	PKI Cryptography Basics.....	4
2.1	The Use of Cryptography .....	4
2.2	Secret Key Cryptography.....	4
2.3	Public Key Cryptography .....	5
2.4	Digital Signatures And Hash Functions .....	6
3	Certification Authorities in a PKI.....	8
3.1	PKI Functions.....	8
3.1.1	Certification .....	9
3.1.2	Validation.....	9
3.2	Trusted Third Parties - Certification Authorities.....	10
3.2.1	Issuing Certificates .....	10
3.2.2	Certificate Revocation .....	11
3.2.3	Storing and Retrieving Certificates and CRL's .....	11
3.3	Trust Management.....	11
3.3.1	Certification Path Processing .....	11
3.3.2	Cross-Certification.....	14
3.4	Time Stamping .....	14
3.5	Key Management.....	15
3.5.1	Updating Keys.....	15
3.5.2	Backing Up Keys.....	15
3.5.3	Archiving Keys.....	16
3.5.4	Smart Cards .....	16
3.6	Summary.....	19
	References .....	22

## 1 Introduction

As the use of WWW shifts from simple information sharing to business-critical applications, businesses of all sizes are attempting to take advantage of web security technologies and integrate them within their business models, thereby enabling secure delivery of products and services over the Internet (world-wide customers) or Intranets (employees). The foundation for secure distributed applications, including secure messaging, e-commerce, and secure (resp. specifically authorised) intranet applications that emerged during the last few years is **Public Key Infrastructure (PKI)**. A company's PKI constitutes the core of its Internet/Intranet security infrastructure. It ensures authenticated, private and non-repudiable communications and transactions with a single set of standards and services that facilitate the use of public key cryptography (PKC) and X.509 certificates in a network environment.

For the perspective of a customer, business partner or internal client (employee) the result of operating a PKI is measured by the ease of obtaining authorisation or transaction security for business-critical applications or access control. In selecting a business PKI solution, the choice depends on critical factors such as [VeriSign98]:

- *Functionality* – support for certificate issuance, processing and protocols for diverse certificate types, comprehensive administration functions, directory integration and key management.
- *Open architecture and customised administration* – support of new and legacy applications of own choice and establishing an administration that suits the company's policy or terminology.
- *High availability and scalability* – guarantee availability of services and disaster recovery without massive investment.
- *Easy to handle workflow management* – guarantee an easy workflow for obtaining authorisation and delivery of authentication.

Different approaches to meet these critical issues are currently deployed for the development of a business PKI:

- Purchase standalone PKI software and create a standalone PKI service.
- Deploy an integrated PKI platform which combines enterprise-controlled and operated PKI software/hardware, compatibility and certificate processing services with popular applications.

Technically, PKI refers to the technology, infrastructure and practices needed to enable use of public key encryption and/or digital signatures in distributed applications. The main function is to distribute public keys accurately and reliably to those needing to encrypt messages or verify digital signatures (authentication). This process can be complicated and not easy to be implemented.

The flexibility and security of an authentication can be improved by introducing third parties. These third parties are trusted by everyone (who agreed on) to certify the true identity of parties who may have never met prior to the intended communication or to authorise user to access critical resources. The utilisation of a third party within a PKI for the authentication service system is commonly referred to as **Certification Authority (CA)**. A CA issues **certificates** that identify its owner. These are digital documents of identification and as a real life ID-document it contains a set of attributes that defines the owner of a certificate. These attributes are encoded in a certificate according to the standard X.509 that is commonly used. Moreover, it contains the owners' **public key** that is generated during a certificate request. The corresponding **private key**<sup>1</sup> is stored in a local database. Certificates play a central role in a public key infrastructure because they reduce the problem of distributing all public keys to the much smaller problem of distributing keys of a small number of CA's. Integrating the **Secure Socket Layer (SSL)** protocol it is possible to secure content and communication between authenticated users. SSL is a commonly accepted security protocol requiring server and client to have identification certificates issued by trusted CA's, allowing the parties to authenticate to each other. In this mode certificates are exchanged along with data that proves possession of the corresponding private key. This protocol is layered on top of other transport protocols (using communication via the WWW it is layered between the HTTP and TCP/IP protocol suite).

The benefits of establishing such a public key infrastructure (PKI) are obvious. It has the potential to solve different security problems with a single set of technical ideas. However, the set of standards and services is still under development and many questions remain about its implementation or available solutions have been incomplete and difficult to deploy. Central role of such a PKI is the trusted third party. A trust centre or certification authority is responsible for the issuance and management of certificates. As it represents the trusted third party involved in certification of different electronic ID's, it becomes crucial to complex evaluation tasks due to official regulations and law. Different approaches have been met during the last years of establishing trustworthy units.

---

<sup>1</sup> For a detailed description of private and public key mechanism, its use and implementation see [Schneier96]

For example, the German Digital Signature Act 1997 establishes a general condition under which digital signatures are deemed secure and forgeries of signatures or manipulation of signed data can be reliably ascertained. A major element of the legislation is a requirement for CA or trust centre licensing. Therefore, to meet the conditions set by the legislation a trust centre has to be evaluated from a competent authority which in turn is ruled and appointed by the government. The security concept for a CA includes all security measures and, especially, an overview of the technical components used and a description of the procedures used in certification. An European initiative currently works out rules for the regulation of a trust infrastructure for Europe. Its aim is to provide an infrastructure to support E-commerce and tamperproof financial transactions in Europe by deploying interoperable certification authorities that supply digital signatures, time stamping and key-recovery services within a clearly defined legal framework.

The Institut für Telematik in Trier recently established a trust centre guided by the German Digital Signature Act 1997. The implemented workflow of processing and issuing certificates follows strict organisational rules and the terminology of the BSI<sup>2</sup>.

The following chapters investigate the issues of establishing a PKI and a trust centre for certification and certificate management.

Having introduced the features and basic terminology of a PKI in Chapter 2, Chapter 3 underlines the importance of a trust centre and describes its central role within a PKI.

---

<sup>2</sup> BSI - Bundesamt für Sicherheit in der Informationstechnologie; Federal Agency for Security in Information Technology, the competent authority to evaluate technical components of trust centres

## 2 PKI Cryptography Basics

This chapter provides a brief overview of the commonly applied basic cryptographic techniques used for encryption/decryption of messages and verification, authentication and validation of users' identities. This overview focuses on general properties and scenarios which are necessary for understanding the following chapters. A detailed description is beyond the scope of this document. The reader is therefore invited to refer to major references such as [Schneier96].

### 2.1 The Use of Cryptography

Cryptography is the science of keeping messages secure and confidential by ciphering them in such a way that they can only be read by those who have the keys to decipher them. In addition to providing confidentiality, cryptography usually is used to accomplish further tasks [Schneier96]. *Authentication* - a user cannot masquerade as someone else, *integrity* of the information sent and *non-repudiation* - the sender should not deny the sending of a message, are desired goals of a secure data delivery. Often the objectives of information security cannot solely be achieved through mathematical algorithms and protocols. An interaction with law issues and procedural techniques are required as well as the physical protection of documents.

The process of encrypting a message converts *plaintext* into an unintelligible sequence of characters, called *ciphertext*, making use of so called ***cryptographic algorithms*** which are mathematical functions. In modern cryptography the problem of encrypting data is solved by using ***keys***. A key is a large number of possible values. The entirety of possible values is called the ***keyspace***. In modern cryptography we distinguish between two different algorithms: secret key cryptography and public key cryptography.

### 2.2 Secret Key Cryptography

Secret Key Cryptography is the classical form of cryptographic techniques. Within this scheme, sender and receiver agree on a key that is used for encryption and decryption. The agreement needs communication between sender and receiver prior to the intended secure information exchange. Drawback of a secret key solution is the key transferral prior to the intended data encryption. Key distribution and secure delivery of the key can be a non-trivial issue.

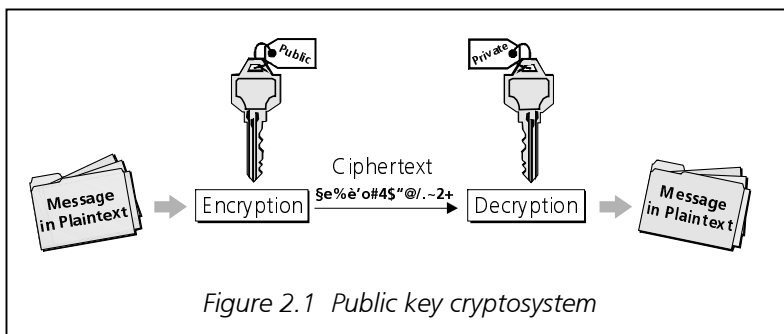
There are many secret key systems, the best-known probably being the Data Encryption Standard (DES, and its newer counterpart Triple-DES) [DES]. There exist systems for communicating securely over public networks using only secret key cryptography, most notably

MIT's Kerberos system [RFC1510]. However, these schemes do not scale well to large, inter-organisational populations, and they also carry extra security procedures that public key systems do not need, such as storing the secret keys on a secure, central server. Still, as we shall see below, secret key systems have their place in a PKI.

### 2.3 Public Key Cryptography

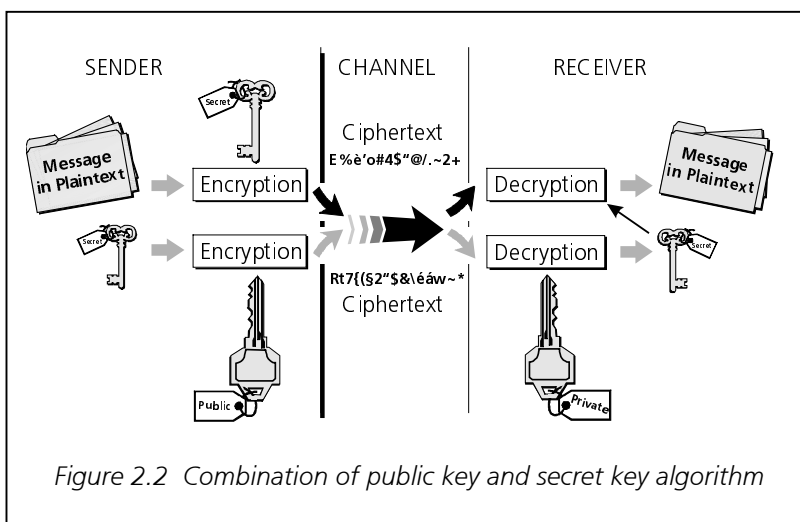
Public key cryptography is, in contrast to secret key cryptosystems<sup>3</sup>, a new technology enabling secure communication. It was first published by Diffie and Hellman in 1976 [DiffHell76] and fully realised by Rivest, Shamir and Adleman in 1977 [RSA78]. Since its first realisation, several proposals for public schemes with different technical nuances have been published (e.g. see [Gamal85]).

Public key algorithms make use of two different keys for encryption and decryption as



shown in Figure 2.1. A *public* key that is widely available (e.g. public directory services) and a different *private* key that is known only to the person, application or service that owns the key. The use of a

public key for encryption nullifies the secure delivery to a communication partner as it can be transmitted unencrypted over insecure lines whereas the private key has to be kept in a secure location. Hence, key distribution is simplified using public key cryptography. A drawback of a



public key system is that the encryption of long messages can be more time consuming than it is the case when using a secret key cryptosystem. Combining both algorithms, a public key algorithm might reduce the complexity of a secret key exchange described in the former section, as a secret key can

<sup>3</sup> A cryptosystem is an algorithm including all possible plaintexts, ciphertexts and keys [Schneier96].

now be encrypted using a public key and thus securely delivered over insecure channels together with the prior encrypted message to the communication partner. Thus, secret and public key algorithms can be combined to effectively implement a secure message exchange (see Figure 2.2).

However, public key cryptography requires an infrastructure assuring the ownership of a public key and managing keys for people, software and services. The use of public key cryptography is crucial for data integrity and authentication. Since digital signatures become more and more important for electronically signing financial transactions or electronic contracts, this will be our next topic.

## 2.4 Digital Signatures And Hash Functions

As mentioned above, public key cryptography accomplishes the task of signing messages. The sender 'encrypts' the message making use of his/her private key. Any participant who has access to the public key of the sender can thus decrypt the message. In doing so, the receiver can be sure that the message was definitely sent by the person (application, software) who claimed to be the sender of the message, since only the sender is in possession of the private key needed for signing. This form of authentication is called *digital signature*.

However, the technical realisation is more complicated than it was described above. To be in compliance with the characteristics of a 'personal' signature, an intruder should not be able to forge or copy the signature to another document. Moreover, documents should not be modified during a communication between two participants without leaving any evidence of the modification [Schneier96]. Hence, to digitally sign a document for practical purposes, another step is introduced before sending any data along insecure channels. The basic idea is to encrypt a hash of the document with the private key rather than encrypt the complete message. This hash is realised using a cryptographic *hash function*. Such a function (mathematical or otherwise) maps a variable length input stream to a fixed length output string, that is usually much smaller than the original input. In choosing a hash function<sup>4</sup>, it has to be ensured that it is impossible for an intruder to find two different messages which have the same hash value (*collision-free* hash function). Hence, currently used hash functions generate a document specific value. The hash value is often referred to as *fingerprint* or *message digest*. The algorithms commonly used are MD2, MD4, and MD5<sup>5</sup> [RFC1321] which were developed by RSA Data

---

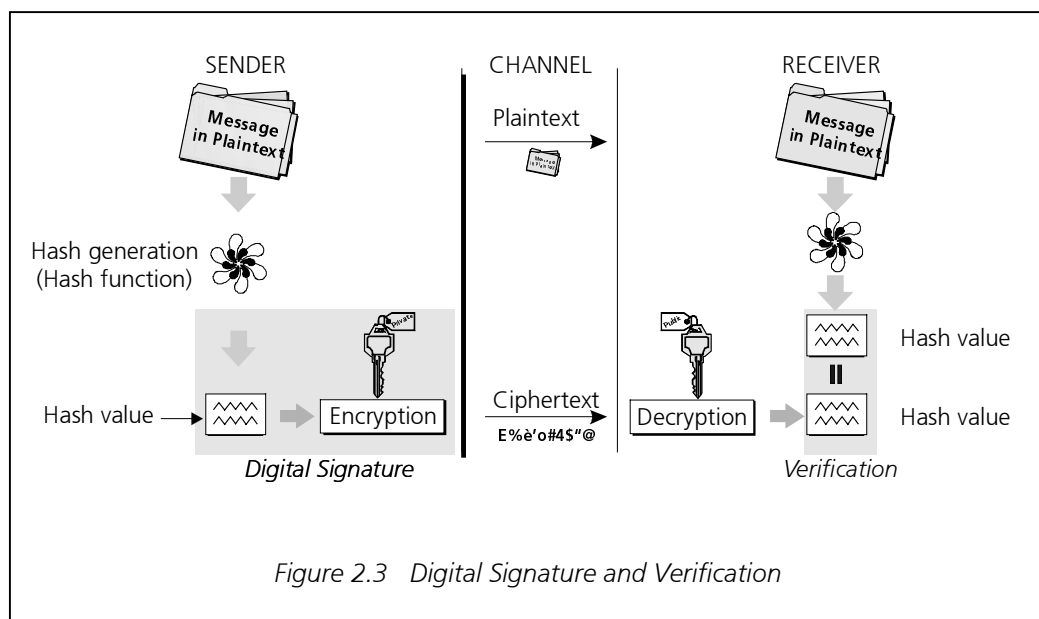
<sup>4</sup> Usually hash functions will create a string that might be the same for several input streams. A **one-way** hash function will ensure that for a produced hash, it is infeasible to find a message with the same hash value.

<sup>5</sup> In sequential order, with the later algorithms more secure than the earlier ones. Commonly they all generate 128-bit hash values where MD5 is the current default recommendation.

Security, Inc. and the Secure Hash Algorithm (SHA)<sup>6</sup>, developed by the National Institute of Standards and Technology (NIST) and by the National Security Agency (NSA) [SHA1].

Digitally signing a document or message is thus accomplished in two steps. First, a hash of the message is produced and second encrypted with the sender's private key - the document has been signed. Through these steps, digital signatures provide a way to verify that data has not been tampered with (either accidentally or intentionally) while in transit from the source to the destination.

Document and signature are sent to the receiver who verifies the encrypted hash with the public key of the sender. The receiver needs to hash the incoming message for himself and to compare the generated value with the one included by the sender. If both values match, the signature is valid. If not, the receiver is assured that the document has been altered (either accidentally or intentionally) during transmission. Figure 2.3 shows the procedure of the signing and verifying process.



<sup>6</sup> This algorithm generates another hash value (160-bit hash) and was developed for use with DSA (Digital Signature Algorithm) or DSS (Digital Signature Standard). See [Schneier96].

## 3 Certification Authorities in a PKI

The main functions within a PKI are a comprehensible processing and the issuance of certificates by a CA. This should be the main aim when establishing trust centre services. A user must be able to easily and comprehensibly retrieve a certificate from a CA. The difficult tasks of certificate management, verification and issuance should be, as far as possible, completely hidden from the end-user. Such an implementation can be difficult to administer and use, or even worse, be insecure. This chapter describes the general definition of a public key infrastructure and provides an overview of characteristics common to certificate-based<sup>7</sup> PKI's. Moreover, it introduces the basic concepts of services that can be obtained from a CA.

### 3.1 PKI Functions

Most major security standards are designed to work with a PKI. For instance, Secure Socket Layer protocol (SSL), Transport Layer Security (TLS), S/MIME<sup>8</sup>, SET<sup>9</sup> and IP Security (IPSEC) all assume the use of a PKI. In its most simple form, a PKI is a system for publishing the public key values used in public key cryptography. There are two basic operations common to all PKI's:

- **Certification** is the process of binding a public key value to an individual, organisation or other entity, or even to some other piece of information, such as a permission or credential.
- **Validation** is the process of verifying that a certification is still valid.

How these two operations are implemented is the basic defining characteristic of all PKI's. A certificate-based PKI deploys certificates<sup>10</sup> for authentication and verification of ID's and public keys. Thus, the most common functions are issuing, revoking certificates, creating and publishing certificate revocation lists (CRL's), storing and retrieving certificates and CRL's, and finally key lifecycle management. Enhanced or emerging services include time stamping and policy-based certificate validation.

---

<sup>7</sup> The reader should keep in mind, that it is possible to establish a PKI without certificates (see PGP - Pretty Good Privacy)

<sup>8</sup> S/MIME - Secure Multipurpose Internet Mail Extensions

<sup>9</sup> SET - Secure Electronic Transaction

<sup>10</sup> Recently deployed certificates follow the X.509 specification ([ITU97])

### 3.1.1 Certification

Certification is the fundamental function of all PKI's. It is the means by which public key values, and information pertaining to those values, are published. Hence, a **certificate** is the form in which a PKI communicates public key values or information about public keys, or both. In more traditional terms, a certificate is a collection of information that has been digitally signed by its issuer. An identity certificate simply identifies an entity, called the *certificate subject*, and lists the public key value for that entity.<sup>11</sup> It is a digital document of identification and as a real life ID-document it contains a set of attributes that defines the owner of a certificate. These attributes are encoded in a certificate according to the standard X.509 that is commonly used. Figure 4 shows a certificate and its display through Microsoft's Internet Explorer.

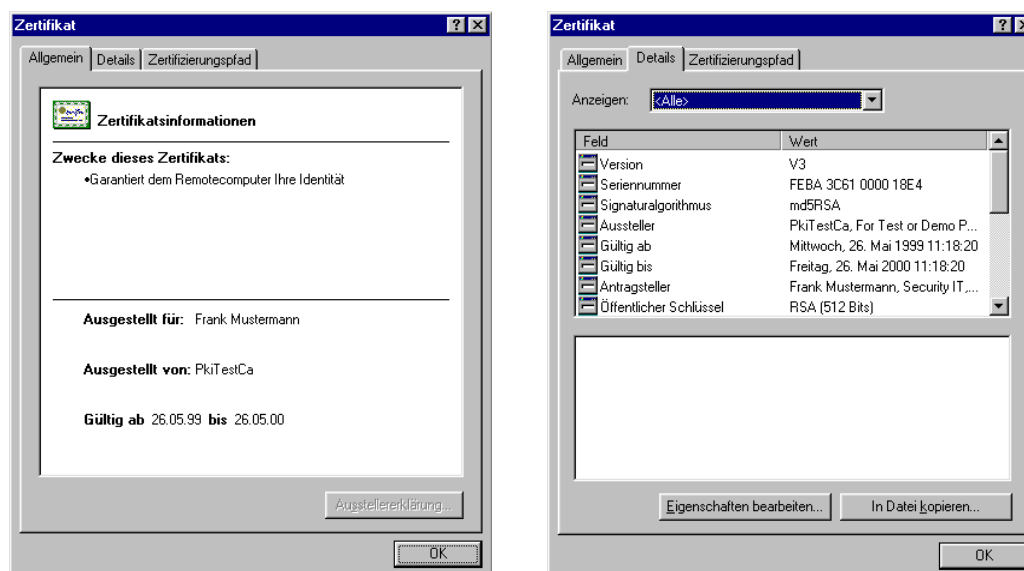


Figure 3.1 Certificate following the X.509 recommendation  
(displayed by Microsoft's Internet Explorer 5.0)

### 3.1.2 Validation

The second basic PKI operation is certificate **validation**. The information in a certificate can change over time. A certificate user needs to be sure that the certificate's data is true – the user needs to validate the certificate. There are two basic methods of certificate validation:

<sup>11</sup> An entity in this context can be an individual, corporation, government or other organisation. It is easiest to think of an entity as some person or party who can control a private key.

- The user can ask the CA directly about a certificate's validity every time it is used. This is known as online validation.
- The CA can include a validity period in the certificate – a pair of dates that define a range during which the information in the certificate can be considered as valid. This is known as offline validation.

However, the user relies upon the information contained in a certificate. The certificate user has to trust the issuing authority to issue valid and 'reliable' certificates. The certificate issuer is commonly referred to as Certification Authority (CA). Another term used for a CA is Trust Centre (TC) or Trusted Third Party (TTP). The synonymity is sometimes not obvious due to the fact that some refer to the term TC as the technical and infrastructural realisation of the certification issues whereas other refer to it as an organisation covering the tasks. The next section describes the functions and tasks that have to be accomplished by a CA.

## 3.2 Trusted Third Parties - Certification Authorities

The flexibility and security of authentication can be improved by introducing third parties. These third parties are trusted by everyone (who agreed on) to certify the true identity of parties who may have never met prior to the intended communication or to authorise user to access critical resources. The utilisation of a third party within a PKI for the authentication service system is commonly referred to as **Certification Authority (CA)**. A CA issues certificates that identify its owner. Moreover, it contains the owners' **public key** that is generated during a certificate request. The corresponding **private key**<sup>12</sup> is stored in a local database. The use of certificates is one example of a PKI, based on the X.509 standard. They play a central role in such a public key infrastructure because they reduce the problem of distributing all public keys to the much smaller problem of distributing keys of a small number of CA's. The following functions have to be accomplished through a CA.

### 3.2.1 Issuing Certificates

The CA signs the certificate, thereby authenticating the identity of the requestor, in the same way that a notary public vouches for a signature and identity of an individual. Additionally, the CA endows the certificate with an expiration date. For the purpose of issuance, the CA returns the certificate to the requesting system and/or posts it in a repository.

---

<sup>12</sup> For a detailed description of private and public key mechanism, its use and implementation see [Schneier96]

### 3.2.2 Certificate Revocation

In case a certificate becomes invalid before the actual date of expiration due to influences like corruption or loss of the private key, there is a need to revoke the certificate. A CA does this by including the certificate's serial number on the next scheduled certificate revocation list (CRL).

### 3.2.3 Storing and Retrieving Certificates and CRL's

The most common way of storing and retrieving certificates and CRL's is via a directory service. Such a directory service is usually accessed via LDAP<sup>13</sup>. Additionally, a directory service can be accessed through either http, ftp or X.500 compatible directories.

## 3.3 Trust Management

To establish trust in a PKI, each public key user must have at least one public key from a CA that the user trusts implicitly. Organisations can establish and maintain trust within a single security domain through a thorough audit of the CA's policies and procedures which ought to be repeated at regular intervals. Such a management will get more complicated when the domain is getting larger and the CA's rely on cross-certification (see below). A cross-certification makes only sense if the validation path through different security domains is unavoidable and a profit of performance can be achieved.

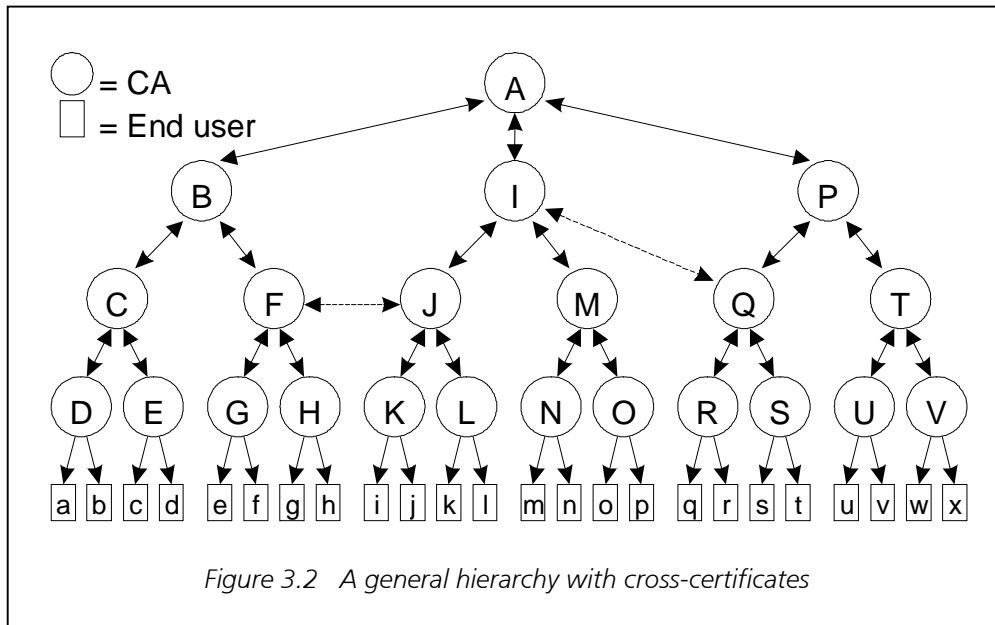
However, a crucial issue of cross-certification is that organisations need to evaluate (finally accept or reject) certificates from CA's not under the direct control of the associated root domain such as CA's of other business units or partners. The problems could be reduced through a thorough implementation of hierarchical structures. Certificate validation is achieved through a processing path involving different authorities.

### 3.3.1 Certification Path Processing

Trust in a PKI is established through a hierarchy of CA's. It is impractical to have a single CA act as the authority for the entire world. Therefore, most PKI's permit CA's to certify other CA's. In effect, one CA validates the certificates of a second CA by cross-certification. The arrangement of the CA's and their relations to each other is a fundamental characteristic of a PKI. In a general hierarchy, each CA certifies its parent and its children. A general hierarchy is illustrated in figure 3.2.

---

<sup>13</sup> LDAP - Lightweight Directory Access Protocol; delivers reduced command set than the full X.500 specification



Typically a hierarchy consists of:

1. A single *root* at the top.
2. The root certifies *Primary Certification Authorities* (PCA's)<sup>14</sup> which issue, suspend, and revoke certificates for all CA's within the hierarchy (In Figure 3.2 this could be B,I,P).
3. PCA's certify CA's (C,F,J,M,Q,T in Fig. 3.2) . They also could cross-certify with PCA-like entities in other vendors' PKI's.
4. CA's authorise *subordinate CA's*, which belong to the PKI service company or the customer.
5. At the bottom of the hierarchy can be *local registration authorities* (LRA's) that evaluate certificate applications on behalf of the root, PCA or CA that issues the certificates.

Certificate path validation is the process of validating the signature of a CA. If a user does not already trust the CA that signed the certificate, the user searches upward through the hierarchy for a trusted CA that has certified the public key of the appropriate CA. Figure 3.3 shows an example of a certification path.

Some PKI's use a variant of the general hierarchy known as a top-down hierarchy, shown in Figure 3.4, in which CA's only certify their children and the top-level CA is the source of all certification paths.<sup>15</sup>

<sup>14</sup> The term PCA was chosen according to the CPS of Verisign [VeriSign98]. This should not be interpreted as commitment.

<sup>15</sup> The source CA of a certification path is also called the *root CA*. This can cause confusion when discussing treelike CA organisations. We use the term *root CA* to indicate the source of a certification path, and *top-level CA* to indicate the CA that is the root of a treelike structure.

**Certificate Chain**

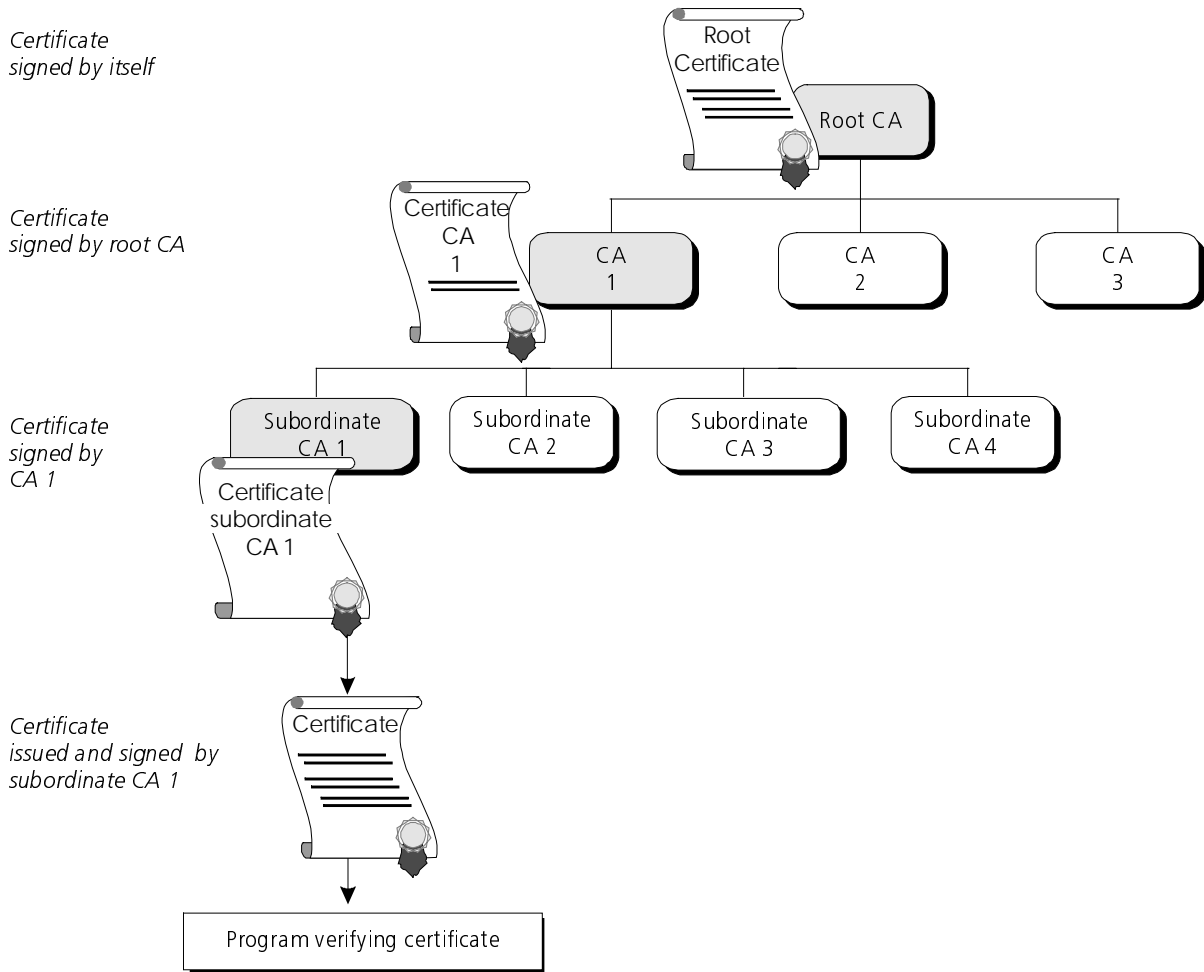


Figure 3.3 Example of a Certification path

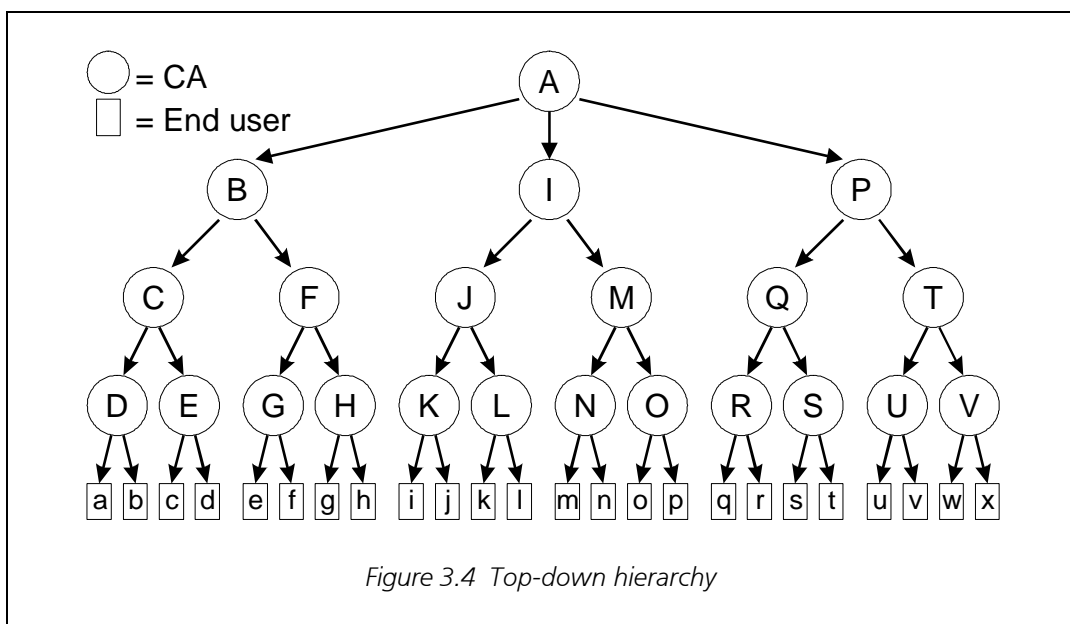


Figure 3.4 Top-down hierarchy

### 3.3.2 Cross-Certification

Also shown in figure 3.2 are some *cross-certificates* (indicated by the dashed arrows) which are certificates that do not follow the basic hierarchy. One CA can issue another CA a certificate that allows the other CA to issue certificates which will be recognised by the first CA. Cross-certification works directly without a third party. Cross-certification helps to reduce path lengths, at the risk of complicating path discovery. For example, in figure 3.2 when entity  $h$  is communicating with entity  $j$ , should it take advantage of the cross-certificate between F and J, making her certification path G-F-J-K? Or should the hierarchical path of G-F-B-A-I-J-K be followed? It depends on how much trust is placed on the different CA's in both paths, and on whether the entity knows about the cross-certification in the first place.

The decision of introducing cross-certification is issue to different PKI establishment policies and considerations.

- What is the organisation's PKI strategy?
- How will interoperability be achieved? (see following chapters for a discussion of legal issues and difficulties)
- Are applications PKI-ready?
- How many clients will be involved in the initial deployment?
- What is the technical staff requirement for planning and establishing a PKI?
- How is scalability guaranteed through cross-certification without complicating path validation?
- Is it legally acceptable to cross-certify different CA's in different countries?

The legal aspects of a cross-certification which does not rely on a third party has to undergo a critical analysis of the different bodies. This becomes complicated as a cross-certification will be accomplished through different countries. A split of trust can arise through different political structures and regulating framework.

## 3.4 Time Stamping

Authenticity and content of a transaction especially within e-commerce scenarios will not provide a log of a transaction. The exact time of the transaction becomes crucial within a financial transfer. The solution to provide trusted transactions is to combine digital signatures with time-stamps. This can be accomplished by incorporating a time stamping service in a PKI. Such a service is usually implemented by the specific CA that handles certificates for crucial transactions or authentication.

### 3.5 Key Management

A CA is responsible for key generation. After the key pair is generated, the CA certifies the public key and issues a certificate for the user. As key generation is the core and crucial part of the CA's tasks, it becomes important to regulate key generation, key length, storage and issuance. The private key must not be stored within a CA's database. As the private key belongs to the owner, and only the owner can use the key for a digital signature, it has to be provided through the CA by a hardware token. The common way of handling the key pair is to make use of smart cards. As the user will receive the certificate, he/she will get a smart card with the generated private key corresponding to the public key contained in the certificate. In practice, the private key and the certificate are both stored on the smart card. The key generation has to be hidden from the technical staff, hence making it obligatory to make use of appropriate software and hardware. A common way of handling this sensitive issue, is to make use of smart cards which provide the possibility of 'on-card' key generation. For an introduction to the card issue refer to [Multi99].

#### 3.5.1 Updating Keys

Each user is likely to have a number of keys that require life cycle management. For instance, users have at least one key pair for each secure application (e.g. e-mail, desktop encryption, VPN, etc..). Some applications use several key pairs for different purposes, such as digital signatures, encryption or authentication. For security reasons, new keys should be issued at regular intervals.

#### 3.5.2 Backing Up Keys

A more complicated task that has to be accomplished by a CA is to restore the keys to the user. The problem with backing up keys is that the technical staff has to make sure that the generated information is stored in a trustworthy place. This imposes strict security management and evaluation by a legal body, hence, an additional security mechanism. If smart cards are provided as storage token, there is no way for key retrieval. The user has to request a new key pair. This is the highest security level that can be achieved. Through the deployment of smart cards with a facility of 'on-card' key generation and storage no one else can interfere with this process. Drawback, as mentioned before, is that the keys cannot be recovered if the card is corrupted or lost.

### 3.5.3 Archiving Keys

This is a task that is basically not a service provided by a CA. When employees leave a company for instance, another issue becomes crucial: archiving the keys. The network manager of the company has to invalidate the encryption keys for future use while retaining them in order to access previously encrypted files and messages. Keys used for digital signatures may be retained for as long as the corresponding signed documents need to be stored, so that signatures can be verified. To avoid the abuse of identity impersonation through the use of the key for a digital signature, organisations have to make sure that only the encryption keys are archived. If the high security smart card approach is used, this becomes redundant.

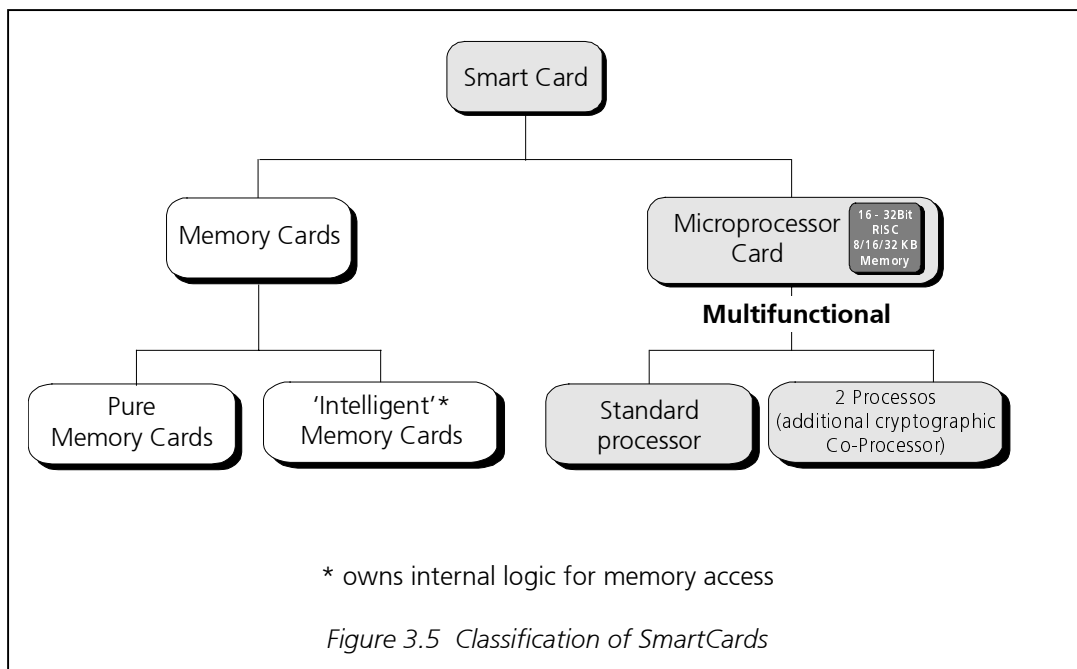
### 3.5.4 Smart Cards

The latter chapters dealt with the difficulties of key management. A CA has to provide secure key generation and certificate handling. Since key generation is the task that requires the highest security level, smart cards are commonly used to provide a secure environment. Modern smart cards provide secure 'on-card' key generation and storage of certificates, such that the technical staff does not interfere with this task. A detailed description of smart cards is beyond the scope of this study, thus the reader should refer to [Multi99], [Rankl97], [Graf99a], and [Graf99b]. This section should only provide a brief overview of the different kinds of smart cards.

#### General Terminology

The smart card, an intelligent token, is a credit card sized plastic card embedded in an integrated circuit chip. It provides not only memory capacity, but computational capability as well. The self-containment of smart card makes it resistant to attack as it does not need to depend upon potentially vulnerable external resources. Because of this characteristic, smart cards are often used in different applications which require strong security protection and authentication. The smart card is becoming more and more significant and will play an important role in our daily life. It will be used to carry a lot of sensitive and critical data about the consumers, ever more than before, when compared with the magnetic stripe card.

Smart cards can be divided into different categories. The terminology in use among experts world wide is not homogeneous. Some categorise smart cards according to their application scenarios, some with regard to technical performance and some according to on-card IT resources. The following figure shows a basic partition of smart cards.



Memory cards contain non-volatile memory and allow 'free' reading and, in many instances, writing or updating of stored data. Such cards are used instead of magnetic stripe cards as they are more reliable and offer more memory. Not all types of memory allow the erasure of data, a feature which is a basic requirement in many applications.

Intelligent memory cards contain a security logic in addition to the non-volatile memory. This allows the introduction of security attributes for reading and writing data. A memory zone may be secret (the data is used for card internal purposes only), public or sensitive. The latter means that it is accessible only after the presentation of a correct 'personal feature' of the user. This is in most cases a Personal Identification Number (PIN) consisting of 4 to 8 digits. The PIN is protected against trial and error attacks by a 'false-presentation-counter'. After a specified number of consecutive false entries the security logic blocks the non-public data against any further access. The new generation of telecommunication chips (e.g. pre-paid telephone applications) also include hardware algorithms for challenge-response mechanisms for the authentication of the card by the system to increase the security against cloning.

Smart cards are chip cards where the chip is a microcomputer with programmable memory. Usually this kind of card contains an own CPU, I/O unit and memory. This category also contains contactless cards. "Contactless" indicates the way the data is transmitted between the chip and the Interface Device (IFD) or the Card Accepting Device (CAD).

The use of cryptocard is important for the deployment of smart cards within a PKI. A cryptocard contains additionally a crypto co-processor which is used for public key calculations

like digital signature, hashing, encryption and decryption algorithms. The 8-bit microcontroller does not have the ability to perform big integer calculations which is necessary for the public key algorithms.

Future smart cards will integrate key-board and display functions. Moreover, it may contain a self-triggered power device for an 'on-card' power supply. For a detailed discussion of smart cards the reader is referred to [Rankl97]. Important to the use of smart cards and its deployment within a security environment is the card life cycle. It is described in the following section.

### Card Life Cycle

The operating system contributes considerably to the guaranteed card life cycle of a smart card. The card life cycle - from production to deactivation of the card (see figure 9) - is divided into the following five phases according to ISO 10202-1 [12]:

Phase 1 - chip and card manufacturing development of the operating system and transport to chip manufacturer; implementation of the operating system, usually as a ROM mask; chip production and transport to card manufacturer.

Phase 2 - card preparation initialisation and pre-personalisation of the card, i.e. loading constants and system-related data; dispatching the cards to the issuers.

Phase 3 - application preparation assignment, personalisation and activation of one or several applications.

Phase 4 - application phase using global card functions and application access, using management functions (administration) of applications (lock, release).

Phase 5 - termination of use delete keys and the complete application.

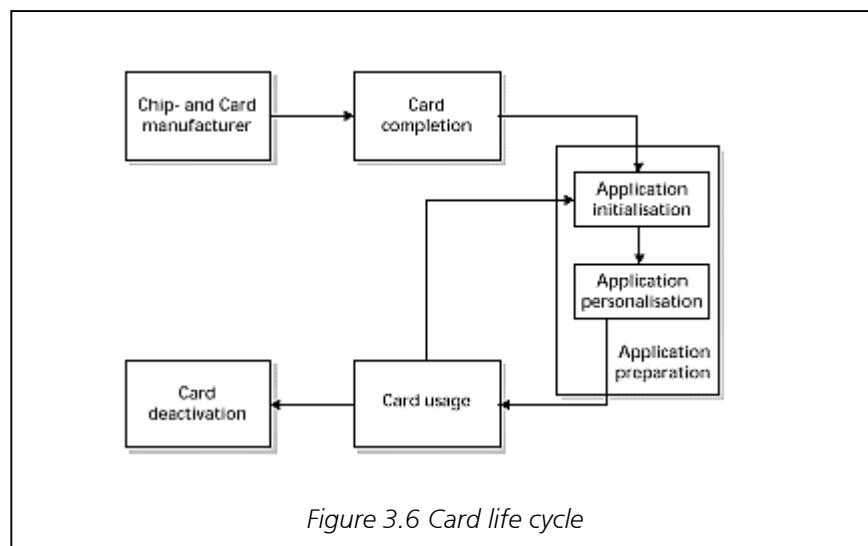


Figure 3.6 Card life cycle

### Application: Banking Cards

The first nation-wide smart card application was a banking card in France. The field-test using memory chips took place in Lyon in 1983. Shortly afterwards microcontrollers from Motorola with EPROM as non-volatile memory were introduced at national level. The transition from magnetic stripe to microcontroller with on-board memory had the following advantages from a security point of view:

- administration, storage and validity check of the PIN by and in the chip;
- authentication and enciphering of data using a cryptographic algorithm.

Smart cards supporting electronic purse applications were introduced on a national level in Austria in 1995 and in Germany in 1996. All these cards, which were issued as a replacement for the eurocheque cards with a magnetic stripe, were hybrid cards with both a magnetic stripe and a microcomputer. Apart from the usual Point of Sale (POS) functions of the magnetic stripe, the chip supports an electronic purse. Since the credit limit is also administered by the chip, POS transactions can now be handled off-line. As in most electronic purse systems (e.g. VISA Cash, STARCOIN) the loading of the purse is done on-line while purchasing can be done off-line. With the availability of on-board hardware units for special arithmetic, the symmetric cryptoalgorithms such as DES are nowadays replaced with asymmetric (public key) algorithms. This significantly improves both the key management and the cryptographic protection.

### Applications within a PKI

Not only banking cards will benefit from the introduction of digital signatures and authentication mechanisms using public key based cryptosystems. The recent development of such systems and the improvement of the added hardware units now allow the computation of a digital signature employing a key of 512 to 1024 bits in much less than 1 second. This is of particular importance for the security of information and transactions for applications in a world-wide network.

## **3.6 Summary**

Public key cryptography and certificates are emerging as the preferred solution allowing strong security for a number of applications including e-mail, Web access, VPN's and digitally signed code. An infrastructure based on public key cryptography manages keys and certificates for people, programmes and systems.

Apart from legal aspects (see below), the main considerations for establishing a PKI should be the technical and secure certificate management aspects. PKI functions should enable

multiple applications to communicate securely through heterogeneous networks. Interoperability, including legal aspects, has to be provided if a strong hierarchy with cross-certification is planned. These legal issues may complicate the deployment of different CA's in different countries. A top-down hierarchy focuses on the implicit trust of the root CA. If the root CA is approved by a legally accepted body, the customer can be assured of the provided implicit trust. Cross-certification has then to be avoided, as this procedure does not rely on a trusted third party and would split the hierarchy of trust. Another advantage of establishing a hierarchical model is - beyond the legal conformity aspect - the scalability of such a trust model.

The question of how to achieve interoperability focuses on two main issues.

- On a particular vendor's product
- On standards

In the past there was a lack of PKI standards or the existing standards have been immature. Hence, enterprises had to cope with vendors' products. A single-vendor strategy is usually not viable on an enterprise level. Different companies or business units will want to interoperate even though they have implemented PKI's based on different vendors' products.

PKI standards founded on the PKIX specifications [PKIX99] have evolved to the point where companies can plan to use them as framework for company-level PKI interoperability. Legal issues are more mature to provide a framework for trans-national interoperability when establishing a PKI and the associated certificates. However, a unique measurement and evaluation of top-level CA's or especially the root CA has to be guaranteed by legal bodies. This can become complicated when trust is provided in different countries through a wide deployment of CA's.

Summary of CA functions:

Function	Description	Implementation
Registering users	Collect user information and verify user identity	Function of a CA or a separate RA
Issuing certificates	Create certificates in response to a user or administrator request	Function of the CA
Revoking certificates	Create and publish certificate Revocation Lists (CRL's)	Administrative software associated with the CA
Storing and retrieving certificates and Certificate Revocation Lists (CRLs)	Make certificates and CRLs conveniently available to authorised users	The repository for certificates and CRLs is usually a secure, replicated directory service accessible via LDAP
Policy-based certificate path validation	Impose policy-based constraints on the certificate chain and validate if all constraints are met	Function of the CA
Time stamping	Put a time-stamp on each certificate	Function of the CA or a dedicated Time Server (TS)
Key lifecycle management	Update, archive and restore keys	Automated in software or performed manually

## References

- [DES] 'Data Encryption Standard'  
Federal Information Processing  
Standards publication 46-2. December 1993.
- [DiffHell76] Diffie, W.  
Hellman, M. 'New directions in cryptography'  
IEEE - Transactions on Information Theory,  
Vol. 22, 1976
- [Gamal85] ElGamal, T. 'A public key cryptosystem and a signature scheme based  
on discrete logarithms'  
IEEE Transactions on Information Theory, 31 (1985), pp.  
469-472.
- [Graf99a] Graf, H.  
Lüpken, B.  
Losemann, F.  
Engel, T.  
Meinel, C. *Multifunktionale Karten: Teil I - Technischer Überblick*  
Institut für Telematik, Trier, 1999
- [Graf99b] Graf, H.  
Vorwerk, L.  
Lüpken, B.  
Losemann, F.  
Engel, T.  
Meinel, C. *Multifunktionale Karten: Teil II - Anwendungspotentiale*  
Institut für Telematik, Trier, 1999
- [PKIX99] Arsenault, A.  
Turner, S. 'Internet X.509 Public Key Infrastructure - PKIX Roadmap'  
<draft-ietf-pkix-roadmap-02.txt>  
PKIX working group - Internet draft, June 1999
- [Rankl97] Rankl, W.  
Effing, W. 'Smart card hand book'  
Wiley & Sons Ltd., 1997
- [RFC1321] Rivest, R. 'The MD5 Message Digest Algorithm'  
RFC, April 1992
- [RSA78] Rivest, R.  
Shamir, A.  
Adleman, L. 'A method for obtaining digital signatures and public key  
cryptosystems'  
Communications of the ACM, 21 (1978), pp. 120-126.
- [Schneier96] Schneier, B. 'Applied Cryptography'  
John Wiley & Sons, 1996
- [SHA1] 'Secure Hash Standard'  
Federal Information Processing Standards  
180-1, 1995
- [VeriSign98] 'Public-Key Infrastructure (PKI) - the VeriSign Difference'  
White Paper, 1998 - VeriSign corporation