

European Master in Multimedia Projects

Technologies du multimédia, des réseaux et de l'Internet

Prof.: M. Van Droogenbroeck

***Introduction to
PKI - Public Key Infrastructure***

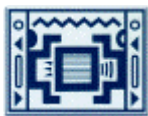


Table of Contents

1. ABSTRACT	4
2. INTRODUCTION TO BASIC SECURITY CONCEPTS	5
2.1. ACCESS CONTROL POLICY	5
2.2. DISTRIBUTED SYSTEMS AND PASSWORD AUTHENTICATION	7
2.3. SYMMETRIC AND ASYMMETRIC ENCRYPTION	10
2.4. HASHING	12
2.5. DIGITAL SIGNATURE	13
2.6. DIGITAL SIGNATURE ASSOCIATED WITH MESSAGE ENCRYPTION	14
2.7. SUMMARY	16
3. PKI	17
3.1. PKI ENTITIES	17
3.2. CERTIFICATION	18
3.2.1. SUBJECT CERTIFICATION	18
3.2.2. CERTIFICATES	19
3.2.3. CROSS CERTIFICATION	19
3.2.4. CERTIFICATION PATH	20
3.2.5. CA RELATIONSHIPS OF A PKI	20
3.3. VALIDATION	23
3.4. REVOCATION	23
3.5. AUTHENTICATION	23
3.6. KEYS	24
3.6.1. KEY PAIR MODELS	24
3.6.2. KEY MANAGEMENT	24
3.7. SUMMARY : CERTIFICATE LIFECYCLE	27
4. RELATED TECHNOLOGIES	28
4.1. CMS - CRYPTOGRAPHIC MESSAGE SYNTAX	28
4.2. SSL	28
4.3. SECURE E-MAIL / S/MIME	29
4.4. VPN	30
4.5. PGP	30



European Master in Multimedia Projects

Introduction to PKI - Public Key Infrastructure

5. GLOSSARY	32
6. INDEX	34
7. FIGURES AND TABLES	36
8. BIBLIOGRAPHY	37
8.1. BOOKS	37
8.2. WEB- GENERAL REFERENCES	37



1. Abstract

Public Key Infrastructure is a system for supporting digital signatures and document encryption for an organization. It is fast becoming essential for an effective secure commerce and to fulfill general security and authentication requirements over non-secure networks (like the Net). The banking services are the most popular usage of this technology, which is quickly spreading over all the applications that need security to be fully operational.

The objective of this document is to briefly describe the general and basic concepts of the PKI to people interested in security and secure commerce (sometimes called S-commerce) but with a low knowledge level about the Internet security. That's why this study starts by introducing some basic security concepts, which are needed to understand the PKI topics.

This document aims to be a good starting point for those interested in the PKI concepts, without analyzing specific implementations. References are widely provided to allow more in-depth investigation about each of the topics, and a glossary contains the most esoteric technical terms that appear in the document. Some references to specific software implementations are given only to serve as a reference for a technical analysis, and should not be used as a "recommendation" or a "case study".

The structure of this document, together with all the pictures is original. The text expresses my own understanding on the concepts and is based on the documents referenced at the end of each chapter or in the bibliography section.



2. Introduction to Basic Security Concepts

2.1. Access Control Policy

To be able to access data and applications from within a company, a user first needs to be authenticated, and then needs to be authorised to perform the operation. **Authentication Procedures** perform the former task, and **Access Control Decision** functions perform the later task.

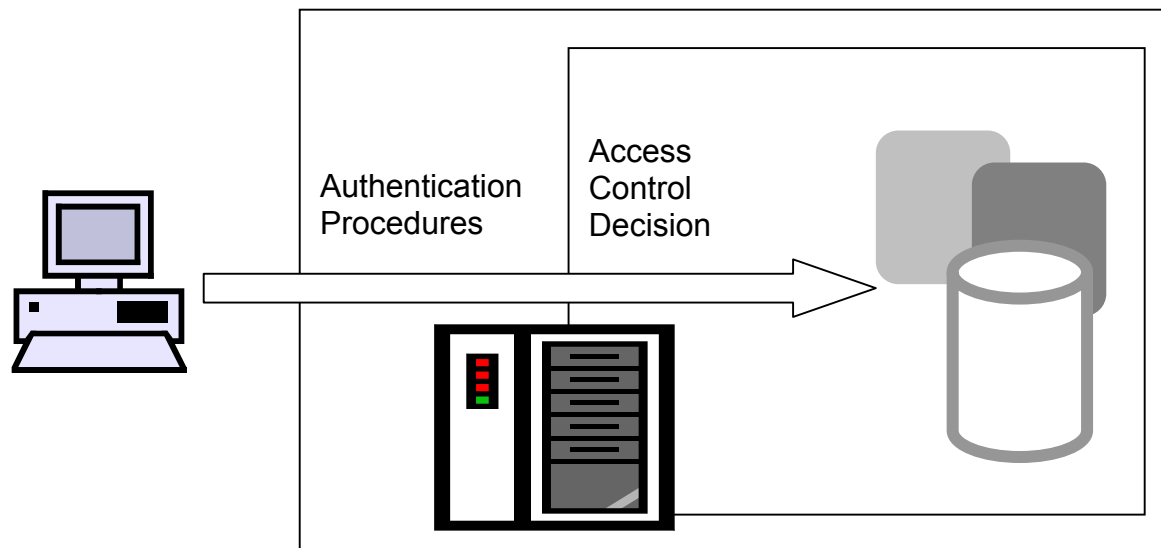


Figure 1 - Authentication and Access Control

The authentication procedures are responsible for the verification of the identity of the user - if (s)he really is who (s)he says (s)he is - because the Access Control functions depend on this. Three different types of information may be used:

Something the user knows - This is the traditional way to validate a remote user based on a password or a "shared secret". It's usually the weakest authentication solution for the following reasons:

1. It can be stolen from the computer by cracker programs - As in a dictionary attack, where an attacker tries to gain access by using a program that cycles through all the words in a dictionary and their combination with numbers and special characters as possible passwords.



Introduction to PKI - Public Key Infrastructure

2. The user identifications and passwords can be intercepted on the network via sniffer programs (see glossary).

Something the user possesses - The user has a physical token, like a proximity card, a smartcard, a visa card, a private key or a passport. This method is normally combined to the previous one so that the user becomes aware of an attack.

Something the user is - This is the strongest form of authentication, since it is very difficult to steal the authentication token - a fingerprint, ten fingerprints, DNA, retina - from the user. However there are two main disadvantages:

1. They are not secret, therefore if a user's biometrics signature is stolen (or its digital representation) it can never be replaced. Thus biometrics can only be used for local authentication and not network authentication.
2. They are not an exact match (like passwords, card identifications) but a fuzzy match, meaning that a certain number of false positives and false negatives may arise.

Even stronger procedures use a combination of two or three different methods described above.



2.2. Distributed Systems and Password Authentication

When a company has several applications hosted by different systems and servers, there are several ways of identity authentication.

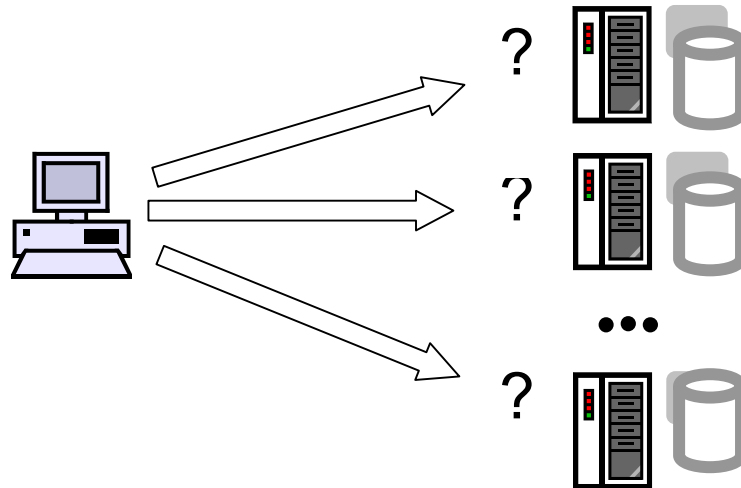


Figure 2 - Multiple Systems, multiple access ?

Multiple passwords, one for each system/application - This is the standard but it's more cumbersome for the users and increases the problems of forgotten passwords.

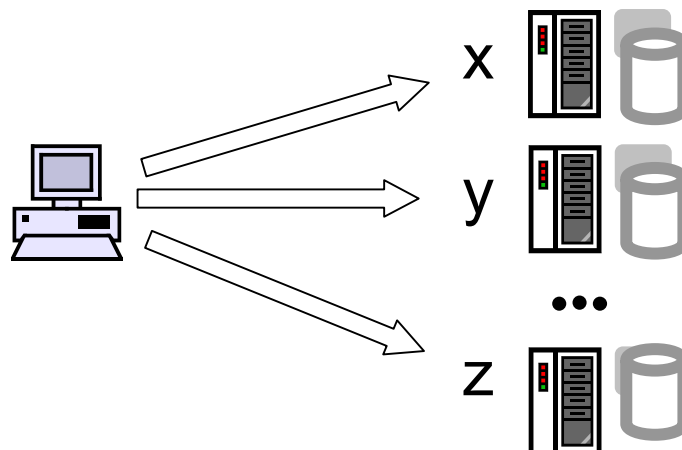


Figure 3 - Multiple Systems, multiple passwords



Introduction to PKI - Public Key Infrastructure

Same password, replicated in each system - This is not usual, although possibleⁱ. It is considered to be extremely vulnerable, since knowledge of a single password (obtained by an attack to the weakest system) gains access to all the systems. The benefit is that users with multiple Ids over the systems only have to remember one password.

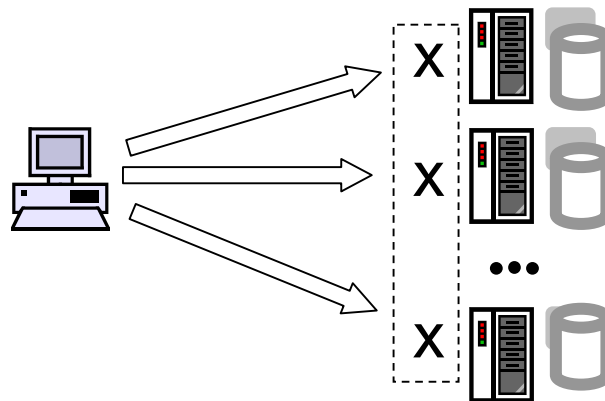


Figure 4 - Password replication

Single sign-on software - These systems are able to store different user names and passwords for each system the user is allowed to use. The single logon software shows a list of authorised applications (menu style) and is able to retrieve the username / password pair needed to log onto the application. The weak points are the password database (protected by cryptography) and the network communication between the single log on server and the other applications (must be a secure network). The strongest implementations of this method are called Secured single Sign-on (SSO).

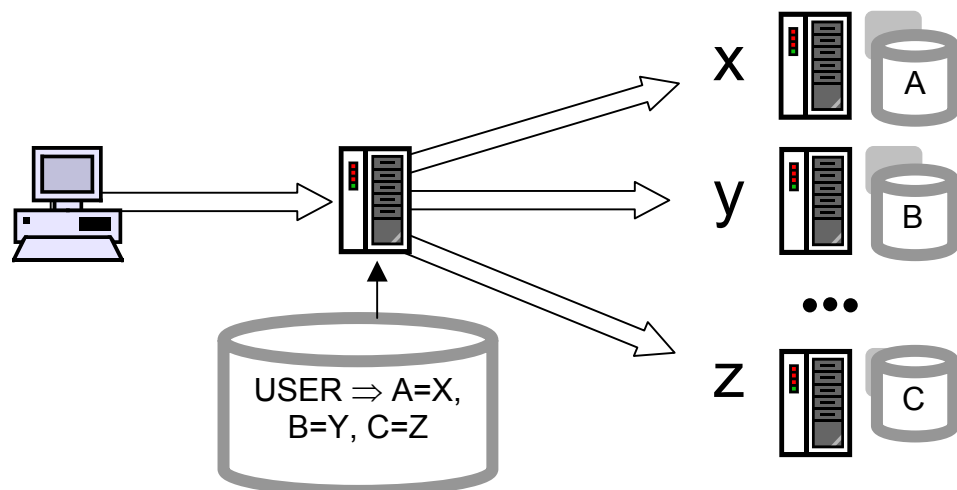


Figure 5 - Single log on software

ⁱ As documented at <http://publibz.boulder.ibm.com/epubs/pdf/ich1a721.pdf>, this feature is allowed by the RACF Remote Sharing Facility, provided by the IBM Secureway Security Server.



Directory Server - Each user has only one password, which is stored in a central system. The user logs onto the trusted central system, which validates its identity. When the user logs onto a second system, this one authenticates itself to the central system informing the user and password and asking for a response. If this is positive, the user can then access the second system and its applications.

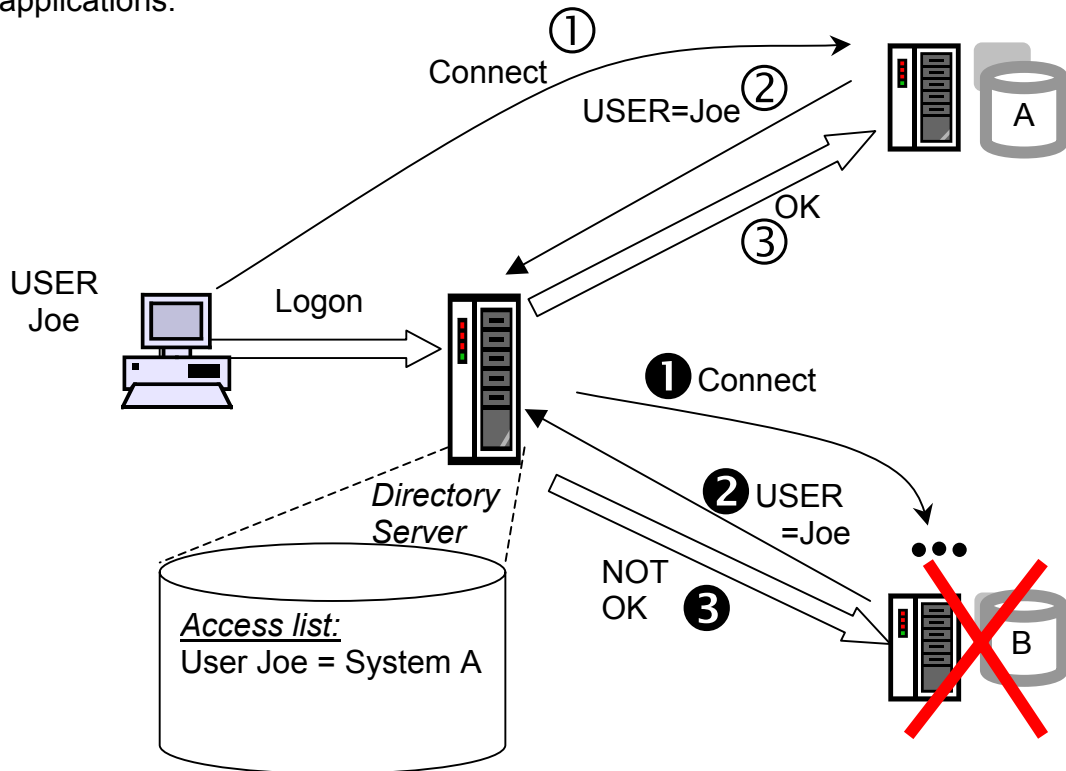


Figure 6 - Directory Server



2.3. Symmetric and Asymmetric Encryption

The objective of encryption is to transform a message (which may contain plain text, images, sound, or other binary objects) to a ciphertext, ensuring confidentiality. It is mainly used to protect passwords and extremely sensitive information stored in databases or transmitted in unsafe networks. A decryption key provides the algorithm to reverse the encryption and is needed to read the message. Two different types of encryption exist: Symmetric and Asymmetric.

In the symmetric encryption schemes (the classical form of cryptography) the same key (called the secret key) is used to both encrypt and decrypt the text. The problem with these systems is to transport the secret key from the sender to the receiver, without security exposures. Some systems (notably MIT's Kerberos System) use only symmetric secret-keys to communicate securely over public networks, but they are difficult to implement in large organisations and need some extra security procedures like a central "trusted and secure" server. The DES (Data Encryption Standard) algorithm is one good example of the symmetric encryption implementation.

Asymmetric cryptosystems (also called *public key* cryptosystems) use one key (the public key) to encrypt a message and a different key (the private key) to decrypt it. Given an encryption key it is virtually impossible to determine the decryption key (and vice versa). The main disadvantage is its slower computing speed when compared to the symmetric encryption (due to its computing complexity). Two different asymmetric algorithms are **RSA** (Rivest Shamir Adleman) which is permutable (one key may either encrypt or decrypt) and **ECDSA** (Elliptic Curve Digital Signature Algorithm, a variant of the well-known **DSA**), that may implement existing algorithms using elliptic curves. The keys are smaller (without compromising security) and consequently faster processing times.

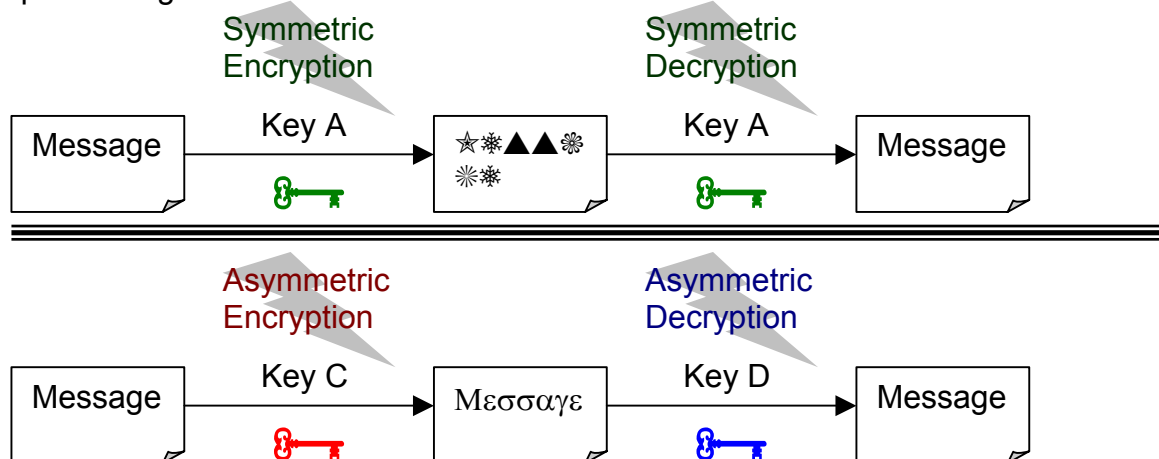


Figure 7 - Comparison between symmetric and asymmetric encryption/decryption



This difference of speed and computing power requirements lead the implementation of secret-key systems to encrypt the message, and the public-key systems to encrypt the secret key (usually shorter than the message, and often limited to 1024 bits = 128 bytes), as shown below:

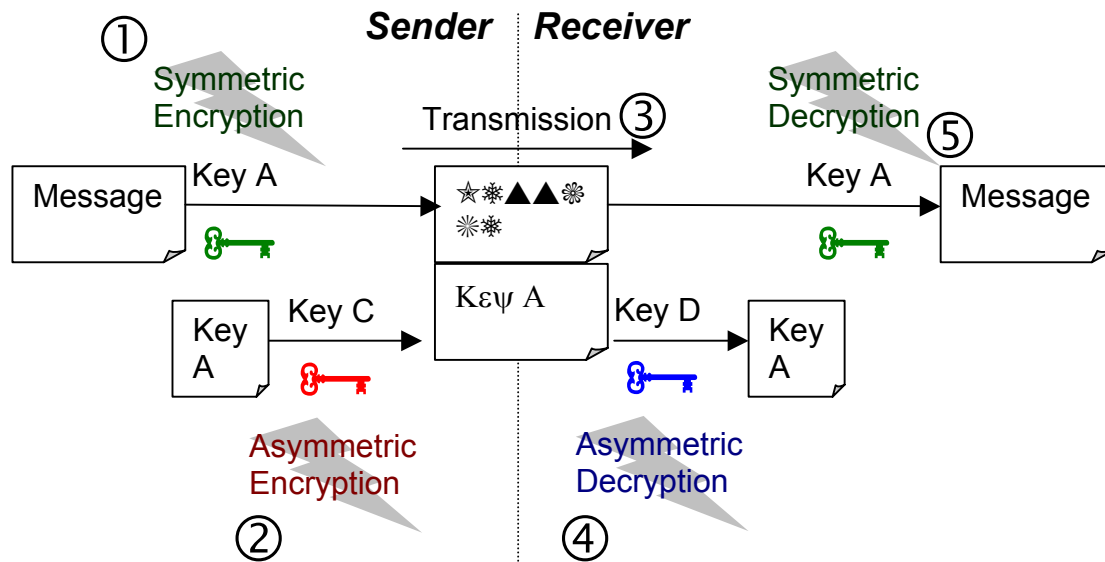


Figure 8 - Secret-key systems structure

The sender encrypts the message (1). The randomly chosen secret key used for this step is also encrypted using the receiver's public key (2). The encrypted message and secret key are sent to the receiver (3), which uses his private key to decrypt the secret key (4), and then uses this one to decrypt the message (5).

References:

As an in-depth discussion of encryption algorithms is beyond the scope of this study, here are some links that can be exploited to obtain more information:

1. <http://world.std.com/~frani/crypto.html>
Cryptography: The Study of Encryption
This page points to several different cryptographic sources
2. <http://theory.lcs.mit.edu/~rivest/chaffing.txt>
Chaffing and Winnowing: Confidentiality without Encryption
A new security technique that can be used instead of encryption.
3. <http://world.std.com/~frani/crypto/rsa-guts.html>
The Mathematical Guts of RSA Encryption
This page explains the concepts of the RSA encryption in a simple (although mathematical) way.



2.4. Hashing

Hashing is the method used to obtain a "digital fingerprint" (hash) for a given message, which may be used to validate the message integrity but not to reproduce it. The hash code has a fixed-length (normally 128 or 160 bits) and it's designed to be unique (different messages produce different hashes).

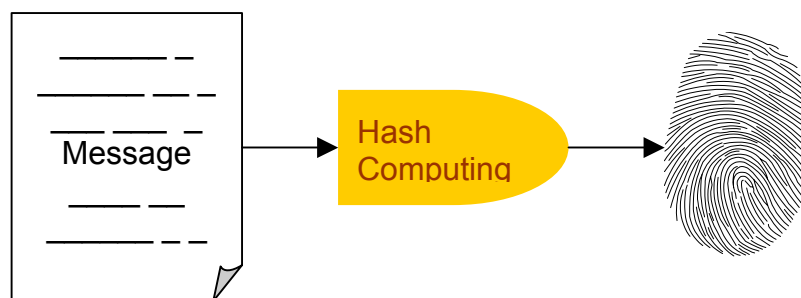


Figure 9 - Hashing

Hashing algorithms are also called one-way hash functions, message digest algorithms, cryptographic checksum, digital fingerprint, message integrity check (MIC) and manipulation detection code (MDC). Some examples are MD2, MD4, MD5 (which use 128 bits and have been created by Ron Rivest) and SHA1 (Secure Hash Algorithm, which uses 160 bits and has been invented by the US National Institute of Science and Technology)ⁱⁱ

References:

More information about hashing algorithms may be found on:

1. <http://www.rsasecurity.com/rsalabs/faq/2-1-6.html>
RSA Security : What is a hash function?
General explanation with links to the algorithms
2. <http://www-theory.dcs.st-and.ac.uk/~mda/cs2001/hashing/general.html>
Hashing
General explanation about double, open and close hashing

ⁱⁱ See bibliography 1



2.5. Digital Signature

To obtain a secure digital signature, several steps (which mix the concepts explained above) must be executed:

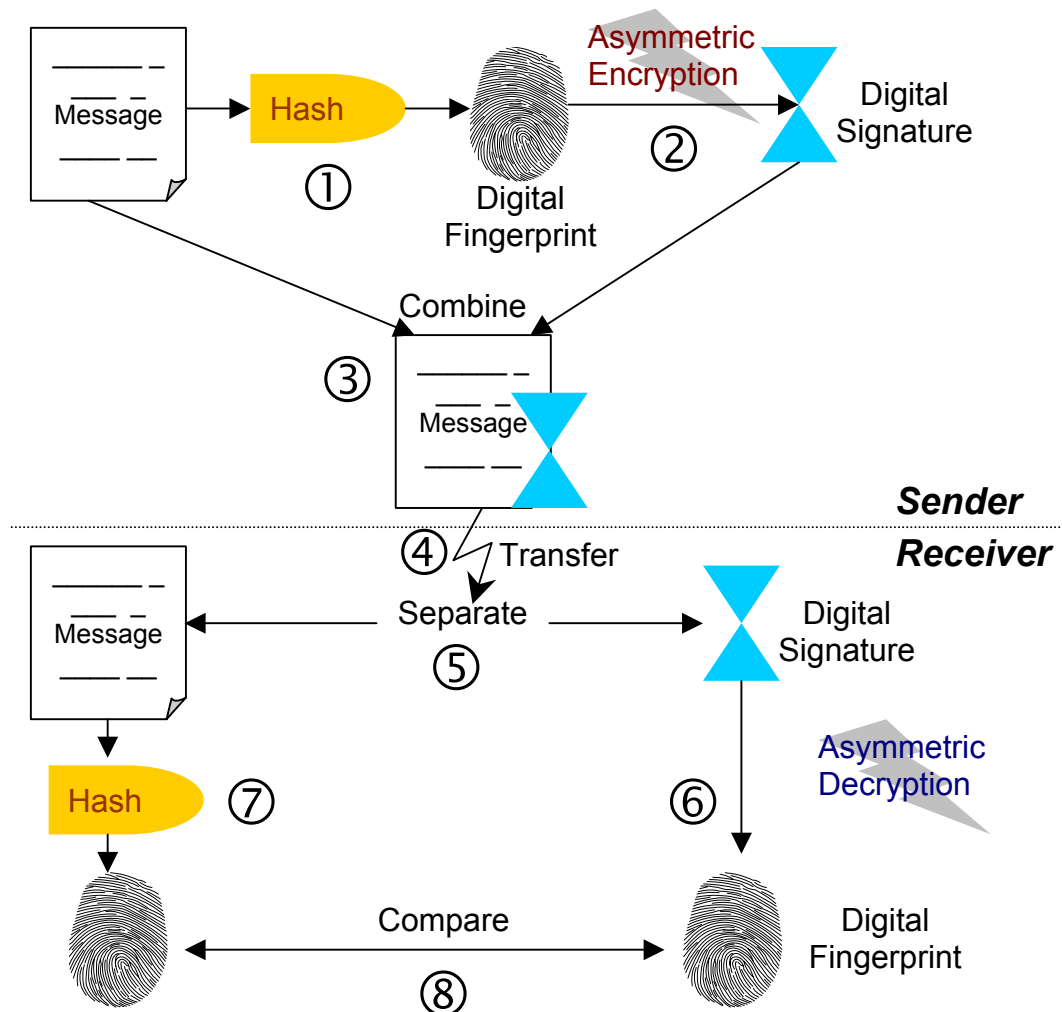


Figure 10 - Digital signature mechanism

At first the message is hashed (1), creating a digital fingerprint which is encrypted using the receiver's public key (2) creating a digital signature. The clear message is combined with the digital signature (3), and the result (an authenticated message) is sent (4). After the reception, the message is separated from the digital signature (5) which is decrypted using the receiver's private key (6). The message is hashed into a "temporary" digital fingerprint (7) which is used to validate the received fingerprint (8). If the message has not been modified during the transfer process, it's authenticated.



2.6. Digital Signature associated with Message Encryption

The digital signature (explained on the page 13) validates if the message has not been corrupted during the transmission (Integrity). To ensure the confidentiality of the message some additional steps must be executed:

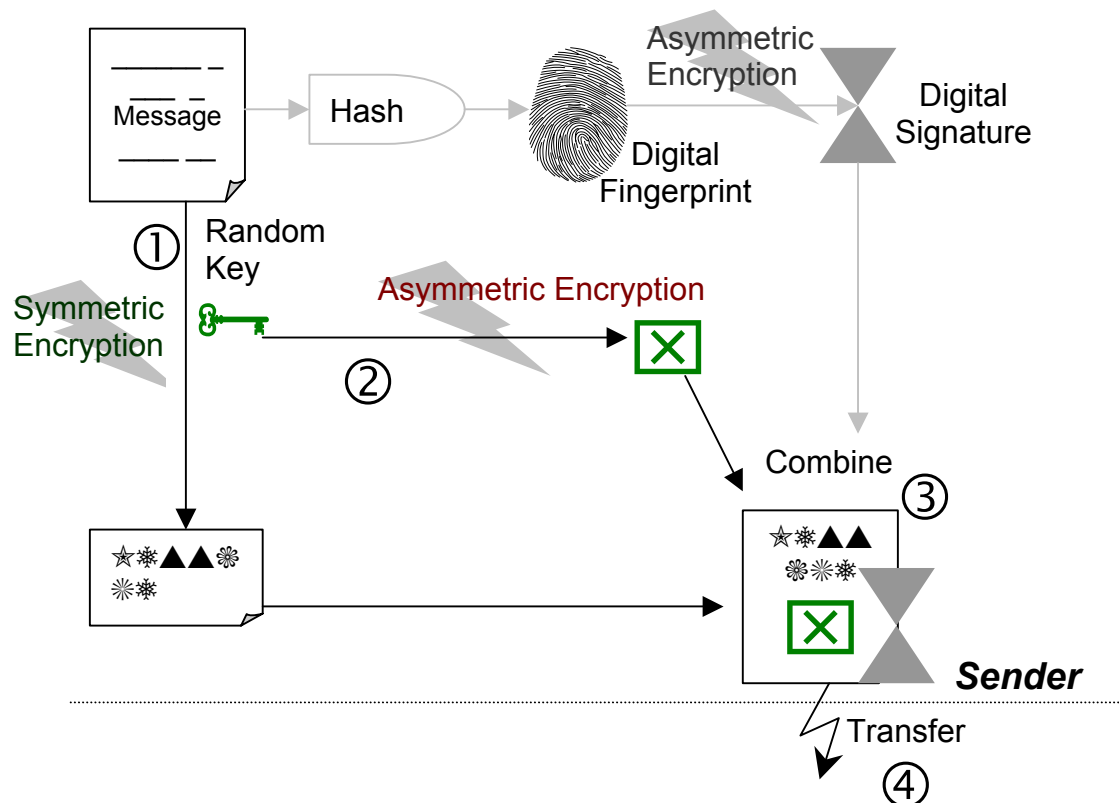


Figure 11 - Digital signature with message encryption

The message is encrypted using a random key (1). This random key is then encrypted using the receiver's public key (2). This encrypted random key will be combined with the digital signature and the encrypted message (3). This package is sent via an unsecured network (4).

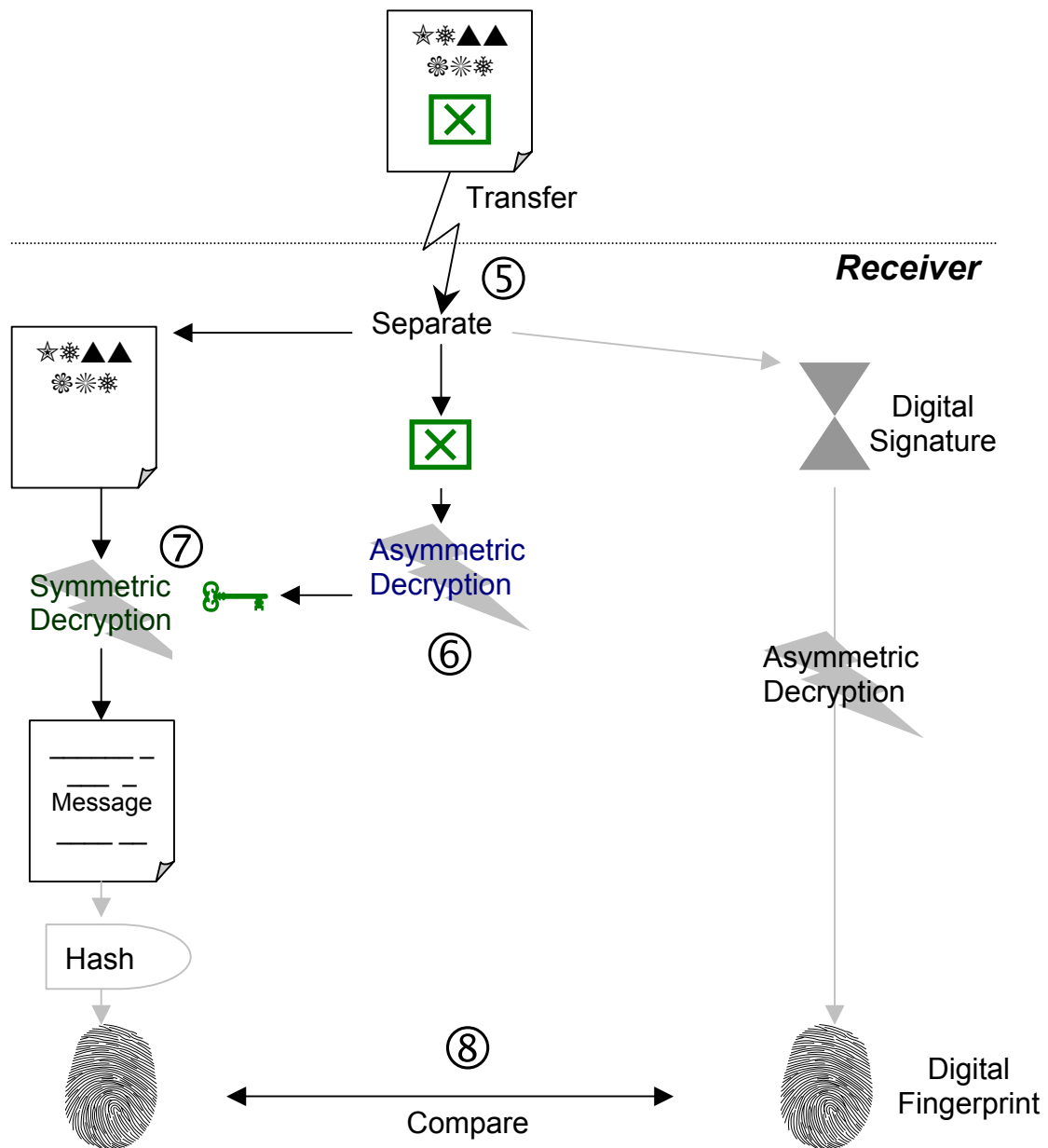


Figure 12 - Digital signature with message decryption

After the reception, the encrypted message and random key are separated from the digital signature (5). The random key is decrypted using the receiver's private key (6). The message is decrypted using the random key (7) and after hashed into a "temporary" digital fingerprint which is used to validate the received fingerprint (8). If the message has not been modified during the transfer process, it's authenticated.



2.7. Summary

Three different formats of messages can be used in public-key cryptosystemsⁱⁱⁱ:

- ⇒ Encrypted message: A symmetric key encrypts the message (as seen on page 10) and the public key encrypts the symmetric key.
- ⇒ Signed message: The message is hashed into a digital fingerprint, which is encrypted into a digital signature (as explained on the page 13) using a private key.
- ⇒ Signed and encrypted message: A combination of the above concepts, in which the message is signed using the private key of the sender and after encrypted using the public key (as explained on the page 14).

ⁱⁱⁱ See bibliography 2, page 10



3. PKI

The public keys must be stored in a directory, to ensure their worldwide availability. As they are accessible via unsecured networks (Internet), an infrastructure must be set-up to allow them to be undoubtedly trusted. This is the main objective of the Public Key Infrastructure. In this chapter we discuss the way this trusted relationship is implemented.

3.1. PKI Entities

CA - Certification Authority

The certification authority is the entity that issues the certificates (see chapter 3.2.2) to the subscriber. It may be on-line (the certificates are obtained via the network infrastructures) - e.g. Verisign - or off-line (the certificates are kept locked in a room and sent by floppy disks using secured transport services.) - e.g. European Sesame Project.

RA - Registration Authority

The Registration Authority is an optional local agent that authenticates the subscriber and who issues requests for certification to the CA on behalf of the subscriber. The RA may be authenticated face-to-face by the CA staff, issued with a certificate, and then trusted to perform face-to-face authentication of the subscribers. A digitally signed message from the RA to the CA will be as good as if the CA had performed face-to-face authentication of the subscriber. One CA may operate several RAs.

Subscriber

Also called a certificate user (or simply subject, later in this text), a subscriber is the entity who has been issued with a certificate, and whose name appears in its subject field.

Relying Party

The relying party is the user receiving the digitally signed information from the subscriber, and who needs to use the PKI to verify the signature.



Repository

The repository holds the certificates and CRLs (see chapter 3.4). It is usually an X.500/LDAP directory, but it could also be a Web site.

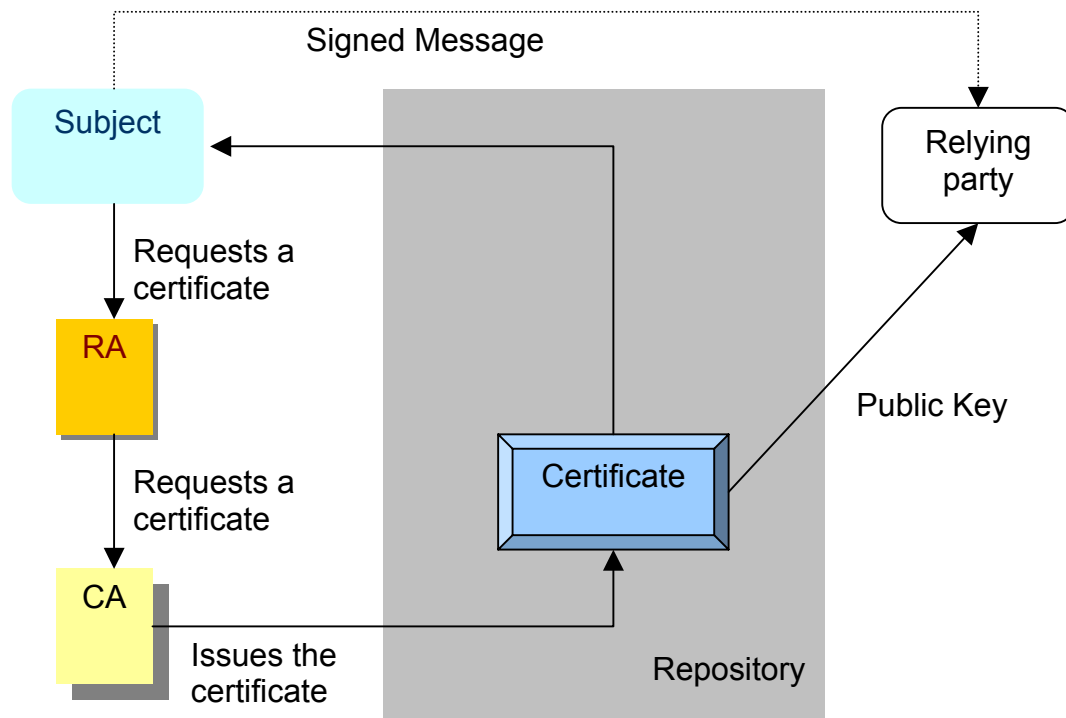


Figure 13 - PKI basic entities and operations

3.2. Certification

Certification is the fundamental function of all PKIs. The certificates provide a secure way of publishing public keys, so that their validity can be trusted.

3.2.1. Subject Certification

The initial procedures used to certify the user's public keys are the most important, since a successful "masquerade" attempt during this phase may be difficult to subsequently detect.

This may be done automatically by "internal" PKIs (runned by a company for its employees), but in the other cases it may require "face-to-face"



Introduction to PKI - Public Key Infrastructure

authentication by either the CA or the RA, and even some "real" (or physical) document validation, like passports or identity cards.

3.2.2. Certificates

A certificate contains (at least) the basic information needed to provide a third-party entity with the subject's public key:

- Subject Identification information
- Subject public Key
- CA Identification Information
- Validity (e.g. time)

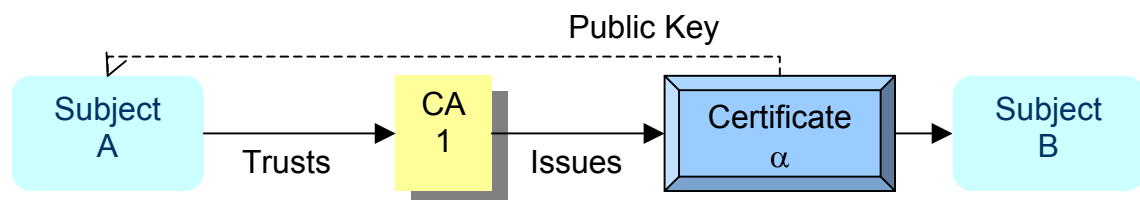


Figure 14 - Certificates

The certificates may be used to identify an *entity* - the identity certificates - or *non-entities*, such as permissions or credentials - the credential certificates.

A *true certificate* is trusted to identify the subject and its public key or credentials and may then be used by other subjects.

3.2.3. Cross Certification

Not all the entities will trust the same CA to hold their certificates. *Cross certification* is used to create the certificate between two CAs. If both CAs trust each other, a cross certificate pair is established. In other cases, only one certificate would be created, and not a pair.

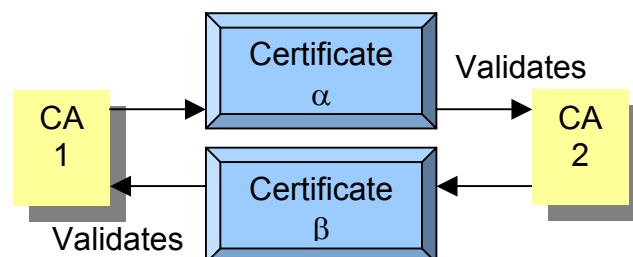


Figure 15 - Cross certificates (pair of trusted CAs)



3.2.4. Certification Path

In a universe composed of several different CAs (and in which not all of them are connected to each other via a cross-certification) an arbitrary number of CAs must validate each other, until a certificate is obtained. This process is called *certification path validation*.

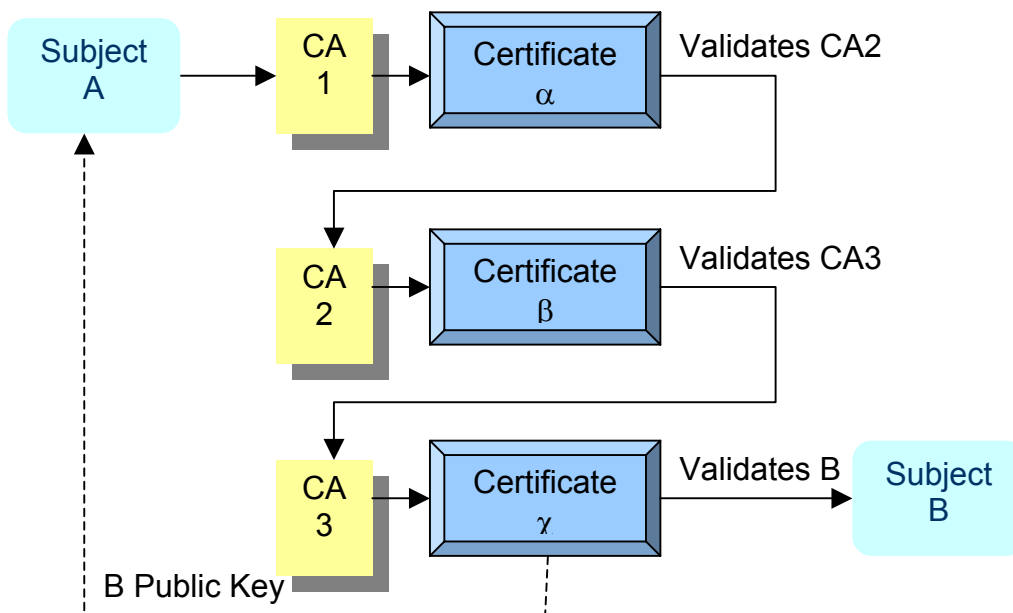


Figure 16 - Certification Path Validation process - example with 3 different CAs.

In the example above, the subject A needs the public key from the subject B. As the subject A does not "trust" the CA 3 (at least not directly), it needs to use the CA1, who "trusts" the CA2, who trusts CA3. This one knows the public key from the subject B that is then sent to the subject A.

3.2.5. CA relationships of a PKI

The CA relationships of a PKI govern its scalability. For a PKI to operate globally, its functions must scale up to a large number of users while keeping the size of the certification paths acceptable^{iv}.

^{iv} It's estimated that in a global web the average certification path length would be between 6 and 7



Introduction to PKI - Public Key Infrastructure

Depending on the general relationships among their subjects, the CAs of a PKI may be arranged within a general hierarchy, top-down hierarchy or a web of trust.

General Hierarchy

Each CA certifies its parent and children, and some extra cross certificates may additionally link the CAs, as shown in the figure below:

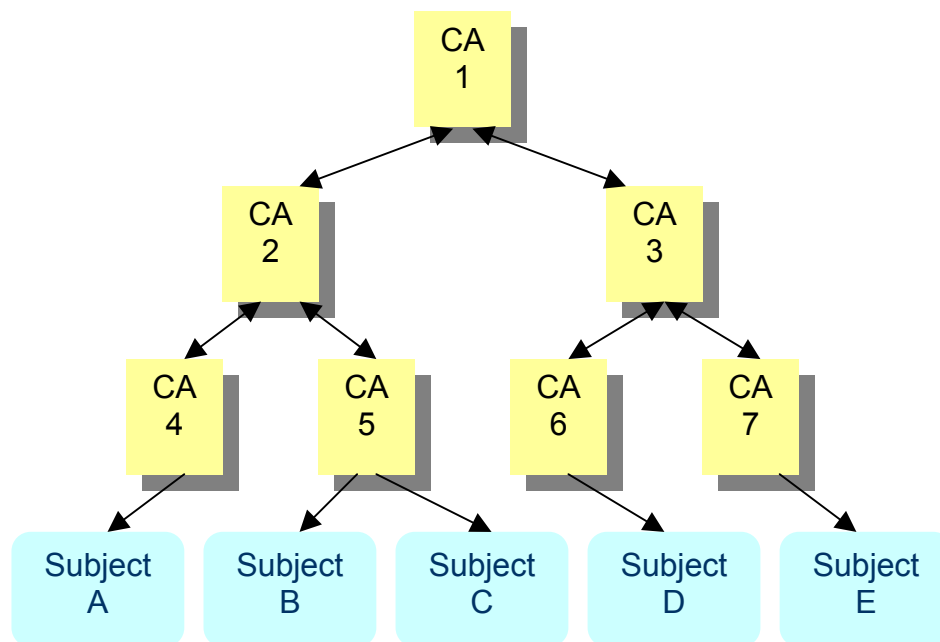


Figure 17 - Example of a 3-level General Hierarchy with cross certificates

In this example, let's suppose that the subject "B" needs to certify a message from the subject "D". It will need to go through the certification path composed by CA5-CA2-CA1-CA3-CA6. But if several subjects from CA5 must certify messages coming from subjects from CA6, a cross-certificate may be established (represented by the dashed line) and then the path is reduced to CA5-CA6.



Top-down Hierarchy

There's a top-level CA and each CA certifies only its children. All users must use the top-level CA as their root CA. This requires all users to obtain a copy of the top-level CA's public key prior to using the PKI:

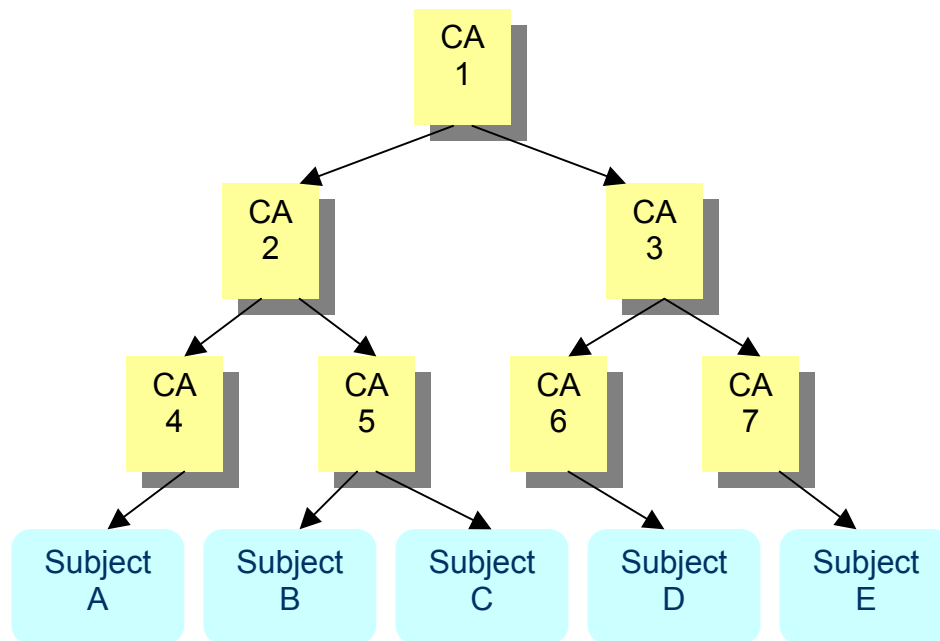


Figure 18 - Example of a 3-level Top-Down Hierarchy

In this example, let's suppose that the subject "B" needs to certify a message from the subject "D". It will need to go through the certification path composed by CA1-CA3-CA6, which is shorter than the previous example (Figure 17 on the page 21).

As all users must fully trust the top-level CA for all purposes, this type of hierarchy may be impractical for a worldwide PKI.

Web of Trust

Some PKIs have no structure at all, depending exclusively on the cross-path certificates between the CAs. This structure may be called a "Web of Trust" - because each CA must base its trust on the certificates of other CAs - and is used by the PGP program (explained later on chapter 4.5, page 30). The users exchange keys and sign each others keys to establish a trust relationship.



3.3. Validation

This is the process that ensures that the certificate information is still valid, as it can change over time. Either the user can ask the CA directly about the validity - every time it's used - or the CA may include a validity period in the certificate. This second alternative is also known as *offline* validation.

3.4. Revocation

Closely related to the validation method, this is the process of informing the users when the information in a certificate is not valid (either because the information has been corrupted or stolen, or simply because some basic user information has changed). This is especially interesting in the absence of online validation approaches, and the most common revocation methods consist in publishing Certification Revocation Lists (CRL). A CRL is a "black" list of revoked certificates that is signed and periodically issued by a CA.

In the initial PKI implementations, there were several problems related to the "time-granularity" problem (wrong information may be given during the time between the certificate has been revoked and has appeared in the CRL) and the size of the CRL. The risk of exposure by the first issue has been reduced by using "Delta-CRL", which is simply a list of changes occurred since the last full CRL has been published. The second issue has been addressed by partitioning the CRL into different lists (for example, one containing the CRL for the end-user subjects, another with the certified CAs).

3.5. Authentication

As explained in the chapter 2.1 (on page 5) In order for the subject to gain access to its private key, it has to possess something (like a smart card or a encrypted key file) and know something (PIN or password) or be something (e.g. a particular fingerprint).



3.6. Keys

3.6.1. Key Pair Models

To increase the security level, different key pairs might exist for different functions, which may be divided into the following categories:

- Non-repudiatable message signing (e.g. e-mail) - Once the message is signed, the subject cannot refuse to be its author.
- Encryption/Decryption functions.
- Authentication only (e.g. LOG ON functions) - Used in cases where a user is not aware of the actual contents of the message that is digitally signed at logon time.

Model	Example	Non-repudiatable	Encryption/Decryption	Access Control
<i>One Key</i>	Verisign (class 1) and PGP	One pair		
<i>Two Keys</i>	Entrust	One pair	One pair	
<i>Three+ Keys</i>	SmartTrust	One pair	One pair	One or many pairs

Table 1 - Examples of key pair models

3.6.2. Key Management

These are the main steps performed in a PKI structure to handle the key pairs:

Key Generation

There are (at least) two different ways of generating the keys (Centralized and Basic authenticated), but the common steps to be performed are:

- Subject identification (see chapter 3.2.1 / page 18)
- CA sends a secret information (normally off-line) to the subject.
- The key pair generation is performed (by the CA or by the subject).
- A connection is established between the subject and the CA (using the secret information to ensure privacy) and either the public key is sent to the CA, either the private key is sent to the subject.



Introduction to PKI - Public Key Infrastructure

- The disadvantage of sending private keys is the loss of the non-repudiation function.
- Sometimes the CA sends the certificate to the user, asking for an acknowledge message.

Storage of Private Keys

To increase the security, the private keys may be stored in SmartCards (something that the subject possesses). The CA root private keys must be strongly protected (if they are compromised the whole PKI is lost) and for ultimate security should ideally be stored in special hardware devices, which are tamperproof, climate proof and that may destroy the key in case of attack. (See references 2 and 3 on page 27)

Revocation of Public Keys

The revocation process has to be easy but also secure:

- If anyone can revoke anyone else's certificate "too" easily, this will lead to Denial of Service (DoS) attacks.
- If the revocation is "too" secure, this may lead "masquerade" attacks to be held before the revocation can be completed.

Some suggested methods are a telephone call from user, validated by a secret token, or a signed message.

See chapter 3.4 on page 23 for more information about this process.

Publication of certificates and CRL

Because the issuing CA signs the certificates and CRLs, they are tamperproof and can be authenticated. As they don't need to be transferred securely, they may then be:

- Published in a LDAP/X.500 directory (explained on the chapter 4, page 28)
- Transferred directly to relying parties (using e-mail or FTP)
- Published on a Web site.

Key Update

As the certificates only have a limited lifetime (typically a year), there's a need for an easy process to update the subject's key pair(s) and issue new certificates periodically. This is far from being an issue and may be fully automated, as the user is already trusted (by a certificate), so he can send a signed message asking for a new certificate.



Introduction to PKI - Public Key Infrastructure

However the update for a root CA key is operationally more difficult, as during the transition (also known as rollover) some users will trust the new key and others still use the old key. This implies that some certificates will be signed by the old CA root key, others with the new CA root key. The solution is to create and publish 3 temporary certificates:

- Old CA root Certificate - signed by the new CA root and with a validity period from date of creation of old certificate to date of expiry of old certificate.
- New CA root Certificate - signed by the old CA root and with a validity period from date of creation of old certificate to date of expiry of old certificate (or until rollover is finished).
- New CA root Certificate - signed by the new CA root. This is the "permanent" one and will be valid until the following update operation is needed.

Backup / Recovery

These functions allow encrypted data to be recovered if a subject loses its private decryption key. The backup is performed by a trusted internal third party (e.g. CA), who keeps a secure copy of the subject's private key. The subject who needs a recovery simply retrieves its private key from this third party (either off-line, using a floppy disk or CD-ROM, either on-line).

The backup is sometimes regarded as a bad thing because it may allow a dubious third party to act as the subject, but corporate environments are accepting it as a good policy to avoid loss of information (for example by using keys for people who already left the company). The organization owns the information, even if the non-repudiation function is not fully guaranteed.

Escrow / Recovery

Technically these functions act similarly to the Backup/Restore, but as this may imply several privacy issues, they are often regarded as a *bad thing*. The trusted third party in this case is a company related to the government or law enforcement agencies, which allow the key recovery to be done by an external organization (such as the police) to retrieve the subject's private key without its (the subject) knowledge or authorization.

References:

1. <http://www.cdt.org/crypto/risks98/>
The risks of key recovery, key escrow and trusted third-party encryption
Interesting report about Key recovery with its inherent risks.



Introduction to PKI - Public Key Infrastructure

2. http://www.chrysalis-its.com/trusted_systems/luna_ca3.htm
Luna® CA³
Example of hardware to securely store root keys.
3. <http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf>
FIPS PUB 140-2
SECURITY REQUIREMENTS FOR CRYPTOGRAPHIC MODULES
This standard specifies the security requirements that will be satisfied by a cryptographic module utilised within a security system protecting sensitive but unclassified information

3.7. Summary : Certificate Lifecycle

The following table gives a short summary of the main activities covered in this chapter:

Activity	CA	RA	Subject	Directory
<i>Verification of Applicant (Registration & Initialization)</i>	Validates (face-to-face or via RA)	Validates (ideally face-to-face)	Provides information to RA and/or CA	-
<i>Certificate Generation</i>	Generates or receive from user	May store a local copy	Receives from CA or generates and sent it.	Stores new certificate
<i>Certificate Publication</i>	Publishes certificate (Intranet, web, floppy)	May store a local copy	-	Makes certificate available
<i>Certificate Revocation</i>	Publish CRL	May inform CA in case of problems and perhaps asks for a new certificate.	Inform CA or RA in case of problems and perhaps asks for a new certificate	Makes CRL available
<i>Certificate Expiration</i>	Update certificate's expiration status	May update the local copy	Perhaps asks for a new certificate	Reflects certificate's expiration status into the database (or simply discards certificate)
<i>Certificate archiving</i>	Keeps (off-line) a copy of the certificate to be used to recall or certify old messages.	May store a local copy	Asks for the certificate if needed.	Clean-up the online certificate.



Table 2 - Summary of basic PKI entities and activities

4. Related technologies

4.1. CMS - Cryptographic Message Syntax

The Cryptographic Message Syntax is used to digitally sign, digest, authenticate, or encrypt arbitrary messages. Its main goal is to define the data structures and processes for digitally signing and encrypting other data structures (also called encapsulation syntax) and it can support a variety of architectures for certificate-based key management, such as the one defined by the PKIX working group.

The complete description of this syntax may be found in the reference 1 below.

References:

1. <http://www.rfc-editor.org/rfc/rfc2630.txt>
Cryptographic Message Syntax - Network Working Group - Type: RFC
Year: 1999
RFC 2630 - CMS Specifications
2. <http://www.rsasecurity.com/rsalabs/pkcs/pkcs-7/>
RSA Laboratories - PKCS #7 - Cryptographic Message Syntax
Standard
CMS standards as implemented by the RSA labs

4.2. SSL

The SSL protocol runs above TCP/IP and below higher-level protocols such as HTTP or IMAP. It allows a server to authenticate itself to a client, allows the client to authenticate itself to the server, and allows both machines to establish an encrypted connection.

Originally developed by Netscape, SSL has been universally accepted on the World Wide Web for authenticated and encrypted communication between clients and servers.

The SSL protocol includes two sub-protocols: the SSL record protocol and the SSL handshake protocol. The SSL record protocol defines the format used to transmit data. The SSL handshake protocol involves using the SSL record



Introduction to PKI - Public Key Infrastructure

protocol to exchange a series of messages between an SSL-enabled server and an SSL-enabled client when they first establish an SSL connection. This exchange of messages is designed to facilitate the following actions:

- Authenticate the server to the client.
- Allow the client and server to select the cryptographic algorithms, or ciphers, that they both support.
- Optionally authenticate the client to the server.
- Use public-key encryption techniques to generate shared secrets.
- Establish an encrypted SSL connection.

References:

1. <http://developer.netscape.com/docs/manuals/security/sslin/contents.htm>
Introduction to SSL

4.3. Secure e-mail / S/MIME

Security services can be added to each communication link along a path, or it can be wrapped around the data being sent, so that it is independent of the communication mechanism. This latter approach is often called "end-to-end" security and it has become a very important topic for users.

Short for Secure Multipurpose Internet Mail Extension - a new version of the MIME protocol that supports encryption of messages - S/MIME is based on RSA's public-key encryption technology (it was originally developed by RSA Data Security, Inc).

Secure messaging, in compliance with the S/MIME standard, is drawing a lot of interest, especially in industries where regulations regarding privacy and security are causing changes. Using PKI, messaging systems can digitally sign messages, encrypt messages or both, providing the authentication, integrity and confidentiality that companies need in an asynchronous world. To enable these functions though, the companies must first enable basic PKI functions, such as issuing and managing key pairs and digital certificates. That has many organizations considering how they will support PKI, either internally or through outsourced services.

References:

1. <http://www.ietf.org/html.charters/smime-charter.html>
S/MIME Mail Security Charter



2. <http://www.ietf.org/rfc/rfc2311.txt>
RFC 2311 - S/MIME Version 2 Message Specification
3. <http://www.imc.org/smime-pgpmime.html>
S/MIME and OpenPGP

4.4. VPN

A virtual private network (VPN) is a private data network that makes use of the public telecommunication infrastructure - instead of owned or leased lines - maintaining privacy through the use of a tunneling protocol and security procedures. The idea of VPN is to give a company the same capabilities at much lower cost by using the shared public infrastructure rather than a private one.

VPNs are an important part of an e-business strategy. Some companies are using VPNs to network remote employees, driving down response times and improving access to business information. Other companies are using VPNs to tie their customers, partners and suppliers into their network as part of an overall e-business strategy.

VPNs require a PKI to authenticate their connection points and, as a result organizations using VPNs are starting to evaluate their overall PKI architectures, because VPN eliminates hard-to-manage modem banks.

References:

- 1) http://searchsystemsmanagement.techtarget.com/originalContent/0,28914_2,sid20_gci758488,00.html
Selecting a VPN solution? Think security first
By Linda Christie, M.A., 18 Apr 2001, searchNetworking

4.5. PGP

Pretty Good Privacy is a product family that enables people to securely exchange messages, and to secure files, disk volumes and network connections with both privacy and strong authentication. PGP is a freely



Introduction to PKI - Public Key Infrastructure

available encryption program that protects the privacy of files and electronic mail, using powerful public key cryptography and working on virtually every platform. It has become a de facto standard for e-mail security.

In an organisation using a PKI with X.509 certificates, it is the job of the CA to issue certificates to users; In an organisation using PGP certificates without a PKI, it is the job of the CA to check the authenticity of all PGP certificates and then sign the good ones.

PGP defines its own PKI built on a "web of trust" (as explained on chapter 3.2.5, page 20).

References:

- 1) <http://www.pgpi.org/>
Home of PGP - Free download and documentation



5. Glossary

#Term	(source) - Definition
Ciphertext	(ϕ) Ciphertext is encrypted text. plaintext is what you have before encryption, and ciphertext is the encrypted result. The term cipher is sometimes used as a synonym for ciphertext, but it more properly means the method of encryption rather than the result.
Decryption	(ϕ) Decryption is the process of converting encrypted data back into its original form, so it can be understood.
Encryption	(ϕ) Encryption is the conversion of data into a form, called a ciphertext, that cannot be easily understood by unauthorized people.
LDAP	<p>(ε) LDAP (Lightweight Directory Access Protocol) is a software protocol for enabling anyone to locate organizations, individuals, and other resources such as files and devices in a network, whether on the public Internet or on a corporate intranet. LDAP is a "lightweight" (smaller amount of code) version of Directory Access Protocol (DAP), which is part of X.500, a standard for directory services in a network. LDAP is lighter because in its initial version it did not include security features.</p> <p>An LDAP directory is organized in a simple "tree" hierarchy consisting of the following levels:</p> <ul style="list-style-type: none">• The root directory (the starting place or the source of the tree), which branches out to• Countries, each of which branches out to• Organizations, which branch out to• Organizational units (divisions, departments, and so forth), which branches out to (includes an entry for)• Individuals (which includes people, files, and shared resources such as printers) <p>An LDAP directory can be distributed among many servers. Each server can have a replicated version of the total directory that is synchronized periodically. An LDAP server is called a Directory System Agent (DSA). An LDAP server that receives a request from a user takes responsibility for the request, passing it to other DSAs as necessary, but ensuring a single coordinated response for the user.</p>
Sniffer	<p>(α) A sniffer is a program that monitors and analyses network traffic, detecting bottlenecks and problems. Using this information, a network manager can keep traffic flowing efficiently.</p> <p>A sniffer can also be used legitimately or illegitimately to capture data being transmitted on a network. A network router reads every packet of data passed to it, determining whether it is intended for a destination within the router's own network or whether it should be passed further along the Internet. A router with a sniffer, however, may be able to read the data in the packet as well as the source and destination addresses.</p>
TCP/IP	(α) TCP/IP (Transmission Control Protocol/Internet Protocol) is the basic communication language or protocol of the Internet. It can also be used as a



European Master in Multimedia Projects

Introduction to PKI - Public Key Infrastructure

	communications protocol in a private network (either an intranet or an extranet). When you are set up with direct access to the Internet, your computer is provided with a copy of the TCP/IP program just as every other computer that you may send messages to or get information from also has a copy of TCP/IP.
X.500	<p>(α) X.500 Directory Service is a standard way to develop an electronic directory of people in an organization so that it can be part of a global directory available to anyone in the world with Internet access. Such a directory is sometimes called a global White Pages directory. The idea is to be able to look up people in a user-friendly way by name, department, or organization. Many enterprises and institutions have created an X.500 directory. Because these directories are organized as part of a single global directory, you can search for hundreds of thousands of people from a single place on the World Wide Web.</p> <p>The X.500 directory is organized under a common "root" directory in a "tree" hierarchy of: country, organization, organizational unit, and person. An entry at each of these levels must have certain attributes; some can have optional ones established locally. Each organization can implement a directory in its own way as long as it adheres to the basic schema or plan. The distributed global directory works through a registration process and one or more central places that manage many directories.</p> <p>In X.500, each local directory is called a Directory System Agent (DSA). A DSA can represent one organization or a group of organizations. The DSAs are interconnected from the Directory Information Tree (DIT). The user interface program for access to one or more DSAs is a Directory User Agent (DUA). DUAs include whois, finger, and programs that offer a graphical user interface. X.500 is implemented as part of the Distributed Computing Environment (DCE) in its Global Directory Service (GDS). The University of Michigan is one of a number of universities that use X.500 as a way to route e-mail as well as to provide name lookup, using the Lightweight Directory Access Protocol (LDAP).</p>

Glossary Sources:

α - SearchNetworking.com

δ - SearchWebServices.com

ϕ - SearchSecurity.com



6. Index

- Access Controls Decision**, 5
- archiving*, 27
- Asymmetric, 10
- Authentication, 5, 23, 24, 38
- Authentication Procedures**, 5
- Backup, 26
- biometrics, 6
- CA, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 31
- CA root, 26
- certificate user, 17
- certificates, 17, 18, 19, 21, 22, 23, 25, 26, 29, 31
- Certification, 17, 18, 19, 20, 23
- Certification Authority, 17
- certification path validation*, 20
- Chaffing and Winnowing, 11
- CMS, 28
- credentials, 19
- CRL, 23, 25, 27
- CRLs, 18, 25
- Cross certification*, 19
- cryptographic checksum, 12
- Cryptographic Message Syntax, 28
- cryptosystems, 10, 16
- Denial of Service, 25
- dictionary attack, 5
- digital fingerprint, 12, 13, 16
- Directory Server**, 9
- Distributed Systems, 7
- DoS, 25
- ECDSA**, 10
- Encrypted message, 16
- Encryption, 10, 11, 24, 32
- entities*, 18, 19, 28
- entity*, 17, 19
- Entrust, 24
- Escrow, 26
- Expiration*, 27
- face-to-face, 17, 18, 27
- fuzzy match, 6
- General Hierarchy, 21
- hardware, 25, 27
- hash, 12
- hierarchy, 21, 22, 32, 33
- Key Generation, 24
- Key Pair, 24
- Key Update, 25
- LDAP, 18, 25, 32, 33
- Lifecycle, 27
- LOG ON functions, 24
- log on software**, 8
- masquerade, 18, 25
- Multiple passwords**, 7
- Non-repudiatable message, 24
- partitioning, 23
- Password Authentication, 7
- Path Validation, 20
- PGP, 22, 24, 30, 31, 37, 38
- PKIX, 28
- Pretty Good Privacy, 30
- private key, 6, 10, 13, 16, 23, 24, 26
- public key, 10, 13, 15, 16, 19, 20, 22, 24, 31
- public keys, 17, 18
- Publication, 25, 27
- RA, 17, 19, 27
- Recovery, 26
- Registration Authority, 17
- relationships, 20, 21
- Relying Party, 17
- Revocation, 23, 25, 27
- RSA**, 10, 11, 12, 28, 29, 38
- S/MIME, 29, 30
- secret-key, 10, 11
- Secure e-mail, 29
- Sesame, 17
- Signed and encrypted message, 16
- Signed message, 16



Introduction to PKI - Public Key Infrastructure

smartcard, 6	Top-down Hierarchy, 22
SmartCards, 25	<i>true certificate</i> , 19
SmartTrust, 24	tunneling protocol, 30
sniffer, 6	unsecured networks, 17
SSL, 28, 29, 37, 38	Validation, 23
subject, 17, 19, 20, 21, 22, 23, 24, 25, 26	Validity, 19
Subject, 18, 19, 24, 27	Verisign, 17, 24
subscriber, 17	virtual private network, 30
Symmetric, 10	VPN, 30
TCP/IP, 28, 32	web of trust, 21, 31
time-granularity, 23	Web of Trust, 22
token, 6, 25	X.500, 18, 25, 32, 33



7. Figures and Tables

<i>Figure 1 - Authentication and Access Control</i>	5
<i>Figure 2 - Multiple Systems, multiple access ?</i>	7
<i>Figure 3 - Multiple Systems, multiple passwords</i>	7
<i>Figure 4 - Password replication</i>	8
<i>Figure 5 - Single log on software</i>	8
<i>Figure 6 - Directory Server</i>	9
<i>Figure 7 - Comparison between symmetric and asymmetric encryption/decryption</i>	10
<i>Figure 8 - Secret-key systems structure</i>	11
<i>Figure 9 - Hashing</i>	12
<i>Figure 10 - Digital signature mechanism</i>	13
<i>Figure 11 - Digital signature with message encryption</i>	14
<i>Figure 12 - Digital signature with message decryption</i>	15
<i>Figure 13 - PKI basic entities and operations</i>	18
<i>Figure 14 - Certificates</i>	19
<i>Figure 15 - Cross certificates (pair of trusted CAs)</i>	19
<i>Figure 16 - Certification Path Validation process - example with 3 different CAs.</i>	20
<i>Figure 17 - Example of a 3-level General Hierarchy with cross certificates</i>	21
<i>Figure 18 - Example of a 3-level Top-Down Hierarchy</i>	22
<i>Table 1 - Examples of key pair models</i>	24
<i>Table 2 - Summary of basic PKI entities and activities</i>	28



8. Bibliography

8.1. Books

1. Technology appraisals
3-days course : "Establishing a Public Key Infrastructure"
6-8 November 2000, by David Chadwick, Andrew Young and Stephen McGibbon

8.2. WEB- General References

2. <http://home.xcert.com/~marcnarc/PKI/thesis/>
By Marc Branchaud, March, 1997
A Survey of Public-Key Infrastructures
3. <http://www.ecommercetimes.com/perl/story/16008.html>
By Mark W. Vigoroso, E-Commerce Times, January 25, 2002
Online Security: Job One for E-Commerce
4. http://searchebusiness.techtarget.com/bestWebLinks/0,289521,sid19_tax282926,00.html
SearchEBusiness.com - Best Web Links - Securing your e-business
5. <http://www.webservicesarchitect.com/content/articles/deJesus01print.asp>
By Edmund De Jesus, Web Services Architect, June 6 2001
Security Implications of Web Services
6. http://www-1.ibm.com/services/strategy/files/IBM_Consulting_Six_signs_that_your_ebusiness_is_trustworthy.pdf
Six Signs that Your e-Business is Trustworthy
Institute for Knowledge Management - April, 2001
7. http://searchsecurity.techtarget.com/originalContent/0,289142,sid14_gci770686,00.html
Govt. should blaze global information warfare trails
SearchSecurity.com - 18 Sep 2001
8. <http://www.networkcomputing.com/1201/1201f1c1.html>
DDoS: Internet Weapons of Mass Destruction
Network Computing - January 8, 2001 - By Brooke Paul
9. http://searchnetworking.techtarget.com/originalContent/0,289142,sid7_gci801362,00.html
Cryptography Basics: Decoding PKI, SSL, PGP
by Michael Bensimon
Friday, December 29, 2000



European Master in Multimedia Projects

Introduction to PKI - Public Key Infrastructure

10. <http://www.cisco.com/warp/public/732/Tech/security/>
Security services - CISCO
11. http://www.rsasecurity.com/solutions/web/whitepapers/AUEB_WP_1100.pdf
RSA e-Security white paper:
Do You Know Who You're Doing e-business with?
12. <http://rr.sans.org/authentic/layered.php>
Layered Authentication - Jeff Parker - May 14, 2001
13. <http://www.pki-page.org/>
The PKI page
This page contains links to various sites and documents which are related to Public Key Infrastructure (PKI) stuff
14. http://searchnetworking.techtarget.com/originalContent/0,289142,sid7_gci801362,00.html
Cryptography Basics: Decoding PKI, SSL, PGP
by Michael Bensimon
Friday, December 29, 2000