Global Information Assurance Certification Paper

An Idiots Guide to Public Key Infrastructure

Mamoor Dewan
Version: 1.4b
27th September 2002

Introduction

The aim of this paper is to provide the reader with an introduction in to the key terms and concepts in the realm of PKI. This will include descriptions and explanations of the various technologies and their interoperation.

The term Public Key Infrastructure (PKI) is used to describe the processes, technologies and practices that are required to provide a secure infrastructure. A PKI should provide the following:

- Authentication: This can be defined as a means of identification. PKI offers this through *digital certificates*.
- Non repudiation: The basis of non-repudiation is that the sender cannot disown any information sent at a later time. Non-repudiation ensures that there is trustworthy means of ensuring ownership of an electronic document. PKI offers non-repudiation through *digital signatures*.
- Confidentiality: This can be defined as the secure transmission of information over networks ensuring that it is not accessed by unauthorised individuals. PKI ensures confidentiality through use of *encryption algorithms*.
- Integrity: The concept of data integrity is that data should not be altered of modified in any way while traversing the network. Integrity of data is ensured by *message hashing*.
- Access Control: The idea of access control is to ensure that only people with the required security privileges are allowed access to information. PKI ensures access control through *public and private key pairs.*

The concepts and technologies that are commonly used to achieve the above functionality are described in the body of this paper.

Digital Signatures

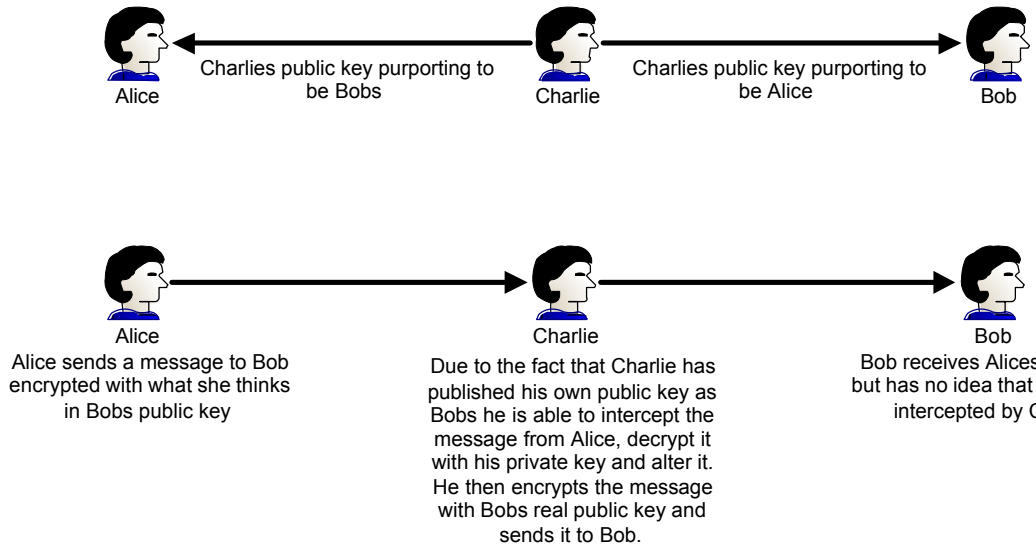
A digital signature is a means of authentication to a message which prevents the message being altered in transit. Digital signatures encrypt a message with the users private signing key. If the signature can be decrypted with the users associated public verification key it will establish the identity of the owner and verify that the message has not been altered since it was signed. The digital signature is created by encrypting the message being sent, therefore each time a different document is sent it will have a different digital signature. For this reason it is not possible to simply cut-and-paste the signature from one communication and append it to another. A digital signature is created from an encryption algorithm that is applied to the entire contents of a message.

A digital signature performs a hashing algorithm on the information to be encrypted, the outcome of this is a fixed sized output file called a message digest. The important thing about the hashing process that it is a one way process, in other words, once the message digest has been produced there is no way of recreating the original information. Also, a small change in the original data will result in a large change in the message digest produced. This ensures that the message cannot be altered in transit, hence guaranteeing integrity and non-repudiation.

However, the reliability of the digital signature depends on the security of the private signing key. If the private signing key is compromised the potential for misuse is high and non-repudiation cannot be maintained. Therefore it is essential that to ensure non-repudiation the private signing key be kept secured.
Another weakness which could potentially limit the usefulness of digital signatures is summarised by the classic man in the middle attack. Take the case of Alice, Bob and Charlie. If we take Charlie to be our "man in the middle" there is nothing stopping him from generating a signing key pair and publishing the public verification part of the key pair to Alice pretending to be Bob; and to Bob pretending to be Alice. He would in effect be positioned in between Alice and Bob and be intercepting the messages passing between them. In the situation when Alice is sending a message to Bob Charlie could do the following:

- Intercept the message.
- Decrypt the message with his private decryption key.
- Alter the message.
- Re-encrypt the message with Bobs public encryption key.
- Sign the message with his own private signing key (which Bob thinks is Alices).
- Send the message to Bob.

Instead of communicating with each other, Alice and Bob are actually communicating with Charlie.



Digital Certificates

As shown in the previous section there is a shortcoming with digital signatures. If the public verification key cannot be trusted there is the potential for malicious activity. An important requirement for a PKI to be useful is the secure distribution of the public keys. When using a public key, whether it is as a sender when encrypting a message, or as a recipient when decrypting a message, you need to be confident that the public key you are using belongs to the correct person. If implementing a small scale PKI public keys could be exchanged face by face, on a floppy disk for example. This becomes unfeasible for any implementation of a PKI which has a large number of users. This is where digital certificates come in to play.

A digital certificate is a guarantee issued by a third party (certification authority) that a person or machine is who they say they are. A digital certificate contains the public keys, name of the person that the certificate is issued to, the certificates CRL distribution point and other associated fields. The digital certificate is authenticated with the digital signature of the certification authority which establishes the accuracy of the public key, and therefore by implication, the authentication and confidentiality of any message that can be decrypted or validated with these public keys.

Using a certificate allows the certification authority to take a users public keys and a host of other important pieces of information and store them in a standard format which can then be signed using a digital signature. Should anyone try to alter the certificate or forge it in any way the client software will detect this because the signature will not validate; and therefore contents of the certificate will not be trusted.

Examples of the fields within a digital certificate are as follow:

- Certificate version number: There are a number of different versions of certificates. The version number of the certificate specifies the fields and formatting of the certificate and therefore how it is to be interpreted by the

applications which make use of it.

- Certificate serial number: This is a unique serial number which is assigned to the certificate. This field is used when the certificate is revoked as a unique reference point.

- Signature algorithm: This field indicates the algorithm that has been used to digitally sign the certificate e.g. RSA with MD5.

- Issuer X.500 Name: This is the name of the issuing organisation in the X.500 naming standard.

- Subject public key information: This field contains information regarding the public key of the owner of the certificate.

- Issuer unique identifier: A unique identifier identifying the issuer.

- Subject unique identifier: A unique identifier identifying the subject.

- Digital signature: This field contains the digital signature for the certificate.

Certificate Revocation Lists

If there is a compromise of a users keys the corresponding certificates need to be revoked. Revocation information needs to be made available to the other users as soon as the compromised certificate is identified; this is done using certificate revocation lists (CRLs). The certificate is revoked using its unique serial number and is placed in a CRL within the directory. This CRL is signed by the CA and is checked every time a certificate is requested from the directory. The certification authority is usually configured to issue CRLs on a periodic basis (e.g every 48 hours), but they are also normally configured to issue a new CRL in the event that a certificate under their control is revoked. This ensures that once the integrity of a key is compromised it is taken out of action as soon as possible.

As you can imagine, for a large scale PKI the CRL list could become very large; this is why CRL distribution points are normally utilised. A CRL distribution point is normally a short list of CRLs (usually less than a thousand certificates) which is checked to ascertain the validity of a certificate. A PKI application will know which CRL distribution point to check by reading the 'CRL distribution point' field of a certificate. This field will indicate the location in the directory where the CRL for the certificate in question will be posted on the event that it is been revoked. The use of CRL distribution points ensures optimal performance of a PKI.

Directories

The role of the directory is to act as a repository for certificates and publicly required information. All public certificates should be published to the directory where they can be retrieved by other users. A directory is a service which contains a global listing of information which different applications can access. The directory provides these main functions:

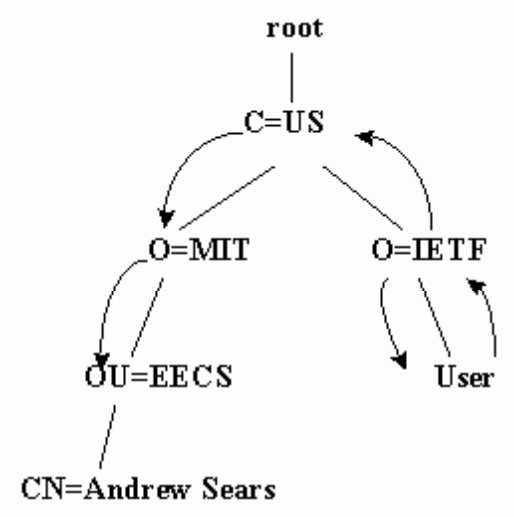
- A repository for information which we wish to distribute.
- Flexibility due to adherence to common standards such as LDAP .

In terms of a PKI each object is represented by an entry in the directory. Each object consists of a series of attributes (email address, phone number, etc) and associated attribute values. When stored in the directory the public components are not protected by encryption, the information is stored in clear text but has been signed by the CA. When a user downloads the information the communication is not secured, but this is not a safety risk as the information is considered to be public. The integrity of the information is however confirmed by the digital signature applied by the CA to the information.

The directory is the most critical part of the PKI as it is here that the certification authority publishes all public certificates and CRLs. It is also the main interface the users of the PKI use to access PKI related information. For this reason it is normally recommended that the directory component of a PKI is designed to be highly available (fault tolerant). This will ensure that in the event of a failure the backup directory the backup will take over and the PKI will be unaffected.

Hierarchy of trust

Within a directory the data is organised in a tree structure. The top of the tree is termed the root. It is a logical connecting point and is not represented by a directory entry. Objects have their own entries in the tree which consist of attributes that provide information about the object represented by the entry.
As mentioned earlier the structure of the directory is hierarchical and therefore each entry in the directory is unique if their path is defined with respect to the root.



[1]

Each entry in the tree can be referred to using its distinguished name (DN). The DN consists of enough of the entities attributes to uniquely identify the entity with respect to the directory. The DN consists of all the superior level attributes of a particular entity. From the diagram above the DN for Andrew Sears is formed by the combination of all the RDNs that are superior to it.

CN = Andrew Sears, OU = EECS, O = MIT, C = US

Trust Models

The CA issues and signs all certificates and acts as the top level trust agent facilitating a third party trust model. Users trust each other because the CA vouches for the authenticity and integrity of the information.

There are two types of trust models:

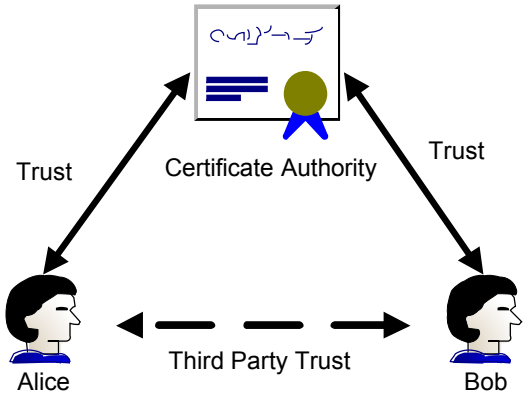
- Direct Trust: This is based on the two entities involved in the trust relationship having an association before the exchange of secure information. It is the responsibility of the two parties involved to ensure that they are happy with the level of trust before the exchange of information commences.

- Third Party Trust: This trust model is used when trust cannot be established on an individual basis. This model becomes complicated when there are many parties involved and therefore a structured trust model needs to be put in to place.

    - Individuals trust each other even if there has been no previous communications because they both posses a relationship with a trusted

[1] Diagram of hierarchy taken from http://itc.mit.edu/rpcp/Pubs/Theses/Andrew/Image33.gif

third party
- The third party is responsible for the trustworthiness of all parties under its trust domain.
- Users implicitly trust any public key from the trusted third party CA



Symmetric & Asymmetric Encryption and PKI

There are two types of keys used when discussing a PKI; symmetric and asymmetric. A symmetric key, as the name suggests is symmetric; that is the same key is used to encrypt and decrypt the information and it is for this reason symmetric key encryption is fast. The downside however is that there is a problem when distributing the keys; how is the symmetric key passed to the recipient ensuring that is it not intercepted in transit?

Asymmetric keys on the other hand work using a key pair; one public key for encryption and another mathematically related but different private key for decryption. Since the process of asymmetric encryption is mathematically intensive it is a lot slower than symmetric encryption, however it does overcome the problem of key distribution. These keys are associated with an entity that needs to authenticate its identity or encrypt data. Each public key is published in a communal data store, usually a directory of some description. Data encrypted with the public key can only be decrypted with the corresponding (and unique) private key. The private key is kept secret and stays with the user, either on the hard disc of their computer or on a token such as a smart card.

PKI uses a combination of the two technologies to overcome the problems of slow performance and key distribution. Symmetric key encryption is used to encrypt the message being sent, but the symmetric key used is encrypted using asymmetric encryption, thereby overcoming the problem of key distribution. An illustrative example of this follows in the next section.
As a result in a PKI you have the following:

- A public encryption key: This is stored in a directory and used when people wish to send you an encrypted message.
- A corresponding private decryption key: This is kept secret and not shared

between users. The private decryption key enables you to decrypt messages sent to you.
- A public verification key: This is also stored in the directory and is used by others to validate your signature.
- A private signing key: This is stored locally on your machine (or on a secure token) and never leaves the machine. It is used to sign documents to provide integrity and non-repudiation.
- A one time session key: This is used to encrypt the actual message to overcome the slow performance of asymmetric encryption.

If you want to send an encrypted mail to a confidant you would encrypt it using their public encryption key. Once this has been performed the only way that someone could decrypt the mail would be to use the associated private decryption key.

By the same token, if you wanted to assure someone that the message came from you, you would encrypt it using you private signing key. If your recipient could successfully decrypt the message (your signature) using your freely available public verification key, they can be reasonably assured that the message came from you.

The one problem with this key pair system is validating the identity of the owner of the public key. This is where digital certificates come in to the equation. A digital certificate proves a person is who they say they are, similar to many widely used proofs of identity such as a driving licence or a passport.

Key Management

As mentioned in the previous section in most PKI implementations there are actually two key pairs involved:

- Signing key pair (private signing and public verification)
- Encryption key pair (public encryption and private decryption)

The public component of the encryption key pair is published in the directory for use by anyone who requires. The decryption key is sent to the client and is stored either on their machine or on a secure token such as a smartcard. Both these keys are usually backed up by the CA. This maintains a key history in the event that a recovery is required.

The signature key pair however is created on the client machine; this is to ensure that non-repudiation can be maintained. If the keys were generated on the CA and then passed to the user the signing key could exist in two locations at the same time, negating non-repudiation. This is the reason that the private signing key is generated on the client machine and never leaves this machine. The public verification key is also created on the client machine; however a backup copy is stored in the directory.

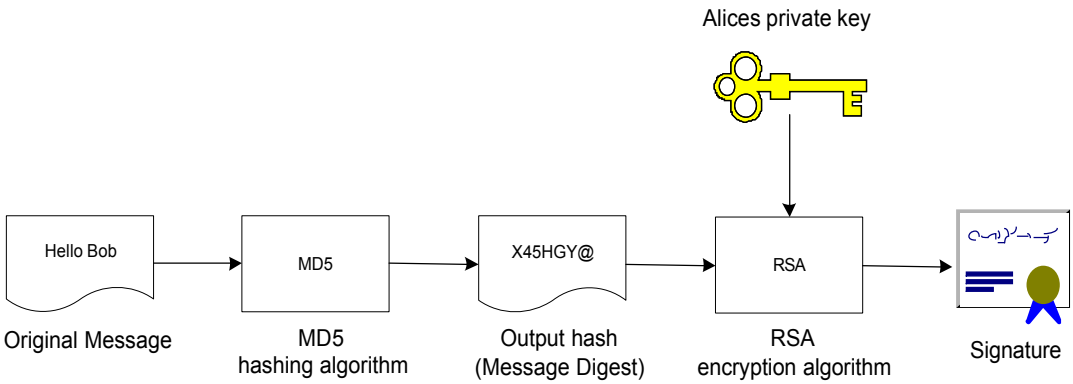
Example

Now that the basic terms have been explained here is an example of how all these concepts fit together to provide the five requirements for data security; Confidentiality, Access Control, Integrity, Authentication, and Non-repudiation.

Consider the case of two users, Alice and Bob. Alice wishes to send a secure communication to Bob.

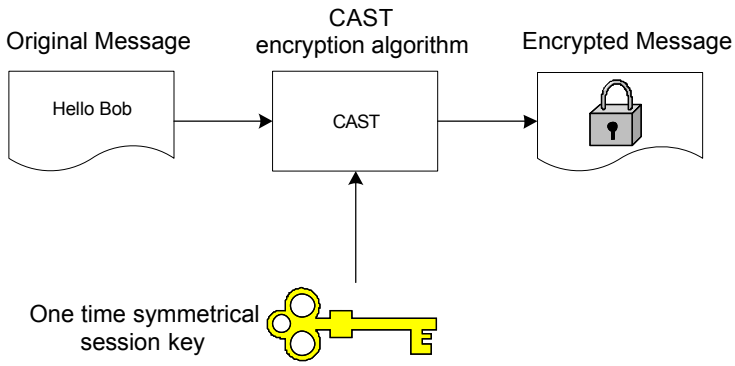
Alice

Firstly Alice must sign her message. She uses a hashing algorithm to create a fixed size hash (message digest) of the message being sent. This process is a one way process i.e. the original message cannot be recreated from the hash. Once the hash has been created she uses her private signing key to sign the message. This creates a unique signature for the message being sent.



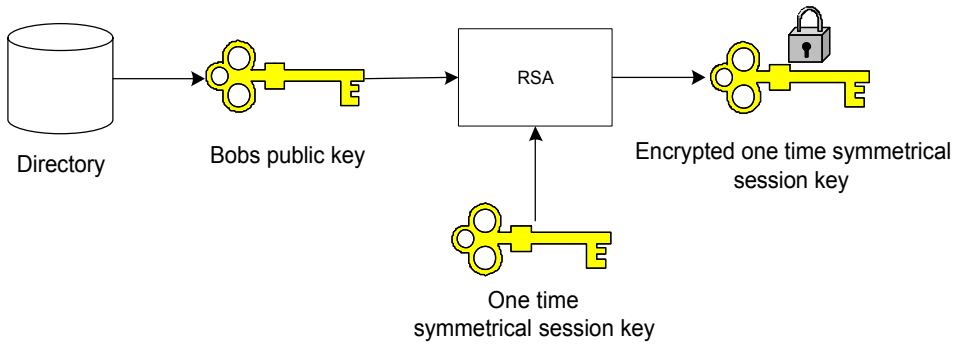
This stage ensures **INTEGRITY**. If the message is altered the hash will change and not validate at the verification stage.

Secondly Alice will encrypt the message with a one time symmetric session key. A one time symmetric session key is used to encrypt the message as symmetric key encryption is a lot faster than asymmetric key encryption.



This stage ensures **CONFIDENTIALITY** as the message is encrypted.

As mentioned earlier, to overcome the problem of key distribution, the one time symmetric session key is encrypted with Bobs public key therefore ensuring that Bob is the only person who can decrypt and use the symmetric one time session key.

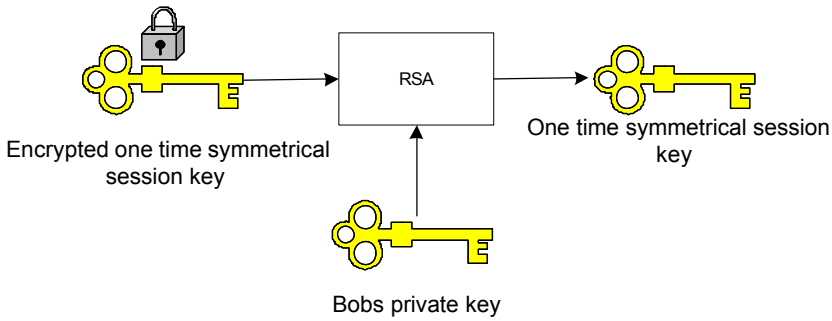


This ensures **ACCESS CONTROL** as only Bob can decrypt the session key.

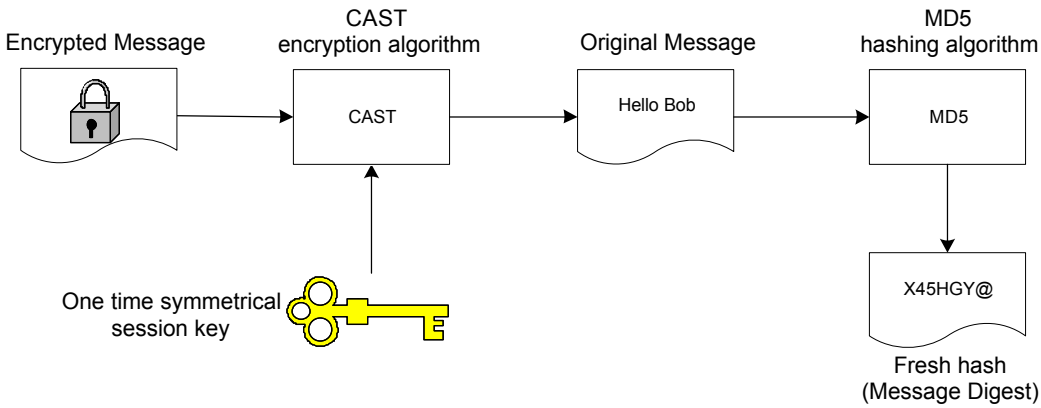
The *signature*, *encrypted message* and the *encrypted session key* are then sent to Bob.

Bob

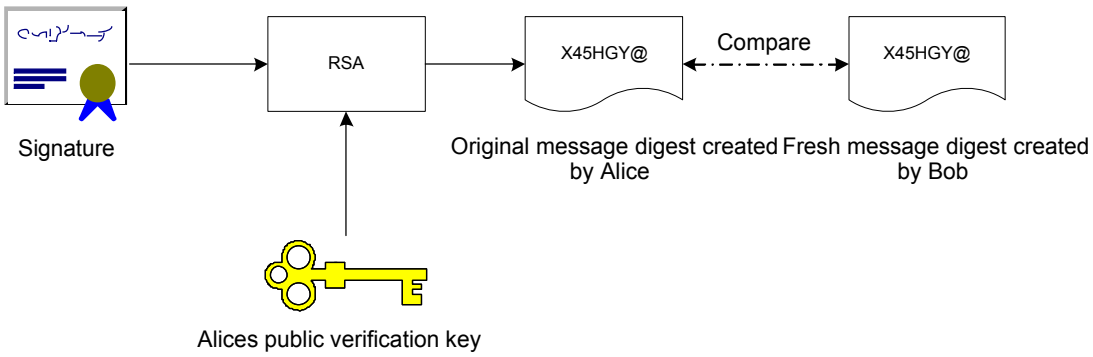
The first thing Bob must do is decrypt the one time symmetrical session key. He uses his private decryption key to obtain the one time symmetrical session key.



Secondly he uses the one time symmetrical session key to decrypt the message and performs a fresh hash on the message to obtain a message digest. This will be used later to check whether the message has been altered in transit.



Lastly Alices signed document is verified. The signature and Alices public key are used to obtain the original hash. This is compared to the fresh hash created by Bob. If these two match the message is valid and has been sent by Alice. This stage ensures **AUTHENTICATION INTEGRITY** and **NON-REPUDIATION.**



Summary

In this paper the fundamental concepts of a PKI were discussed. Key terms such as Digital Signatures, Digital Certificates, Directories and Symmetric and Asymmetric Encryption were discussed and examples were given illustrating their operation.

This report has taken a technological view on PKI; in a real PKI there are many policy and procedural considerations that need to be taken in to account such as certification practice statements and certificate policy documents. These documents address the legal concerns and issues with PKI. It is felt by the author that these areas fall outside the remit of this paper because of their procedural nature and should be investigated by the reader if further information is required.

Generally a PKI offers several benefits over traditional insecured communications including increased privacy, security and assurance when properly implemented. PKI is not a piece of hardware or software, rather it is a set of interconnected components and concepts which together deliver a secure infrastructure. In the end no matter how complicated a PKI is, whether it spans several local offices or different continents, it will remain almost transparent to the end user – exactly how a well designed security infrastructure should be.

References

"PKI Basics Digital Signatures and Public Key Infrastructure (PKI) 101"
URL: http://www.digsigtrust.com/support/pki_basics.html (19/09/2002)

"What is LDAP?" (1999)
URL:
http://www.cfcertification.com/cfdocs/Developing_Web_Applications_with_ColdFusion/16_Connecting_to_LDAP_Directories/dwa16_02.htm (20/09/2002)

Conry-Murray, Andrew "Strategies & Issues: Public Key Infrastructure Nuts and Bolts" (11/05/01)
URL: http://www.networkmagazine.com/article/NMG20011102S0008/1 (19/09/2002)

"Introduction to Public Key Infrastructure" (2002)
URL: http://www.iplanet.com/developer/docs/articles/security/pki.html (12/09/2002)

"How Digital Certificates Work" (2000)
URL:
http://wp.netscape.com/security/techbriefs/certificates/howcerts.html?cp=stbmid
(15/09/2002)

"JTAP Project"
URL: http://www.personal.leeds.ac.uk/~ecldh/lurcis/certs/DigCert.html (21/09/2202)

"Directory Structures" (1999)
URL:
http://www.houseoffusion.com/cfdocs1/Developing_Web_Applications_with_ColdFusion/18_Connecting_to_LDAP_Directories/dwa18_2.htm (20/09/2002)

"Digital Certificates & Encryption"
URL: http://www.enteract.com/~lspitz/digcerts.html (19/09/2002)

"Security: Introduction to Public Key Cryptography (PKI)
URL: http://www.btrade.com/PDF/WP-IntroPKI.pdf

"Entrust /PKI (R5.0) Management" Entrust Technologies, 2000