Public Key Technology Introduction Infrastructure

Govind Singh Tanwar
CSE & IT Department
Govt. Engineering College Bikaner (Raj)

ABSTRACT
Public key technology and digital certificates are emerging as the preferred enablers of strong security for a wide range of e-business messaging applications. In this paper, an overview is provided for public key infrastructure (PKI) and its related technologies: cryptography, digital signature, and digital certificate. Current hot topics of PKI industry such as the trust models, cross-certification, interoperability and standards are described. Commercial PKI products are introduced with a website link to a detail product comparison report. References for further study have also been provided. The target audiences are PKI users, system integrators, software developers, line of business and IT managers, application service or commerce service providers.

Keywords
Public key infrastructure (PKI); Digital Certificates; Digital Signature; Cryptography; Cross-Certification;

1. INTRODUCTION
We are heading toward a world with "a billion connected computers worldwide" (Andrew S. Grove, Intel co-founder and chairman). We are shopping on the Internet, sending credit card numbers, trading stocks, checking bank accounts, and making transactions through the net. We are even signing business contracts through the Internet. In September '98 Bill Clinton signed the first national agreement with the president of Ireland, through the Internet. In the virtual world of Internet, we are doing serious business without face to face communication or any physical contact. Hence, it is becoming urgent for business managers, service providers and solution developers to understand how to secure these information transmissions on the Internet. This paper addresses this need with an overview of the specific technology issues related to Public Key Infrastructure, which is one of the important elements in ensuring secure exchange of information over the Internet.

This paper describes the process and issues of exchanging confidential information through the Internet. It further outlines the procedures for sending a plain text file between two virtual entities, for example, from A to B through the Internet securely. Next, the encryption technologies used to support this approach are explained in detail. The standard concepts involved in this aspect are shown in Figure 1. Plain text is encrypted and signed with *digital signature*, which are based on the public key encryption technology and supported by *digital certificate*. *Digital certificates* are distributed by *public key infrastructure (PKI)*, which involves *trust models* and *cross-certification*. Therefore, overviews of these topics and the way they function are included in section 4. PKI products are described with discussion of standards and interoperability issues. At the end, the paper is summarized with a website link to a PKI application case study and an outline on PKI policy, legal, and operation issues. Glossary is also provided for reader's quick reference.

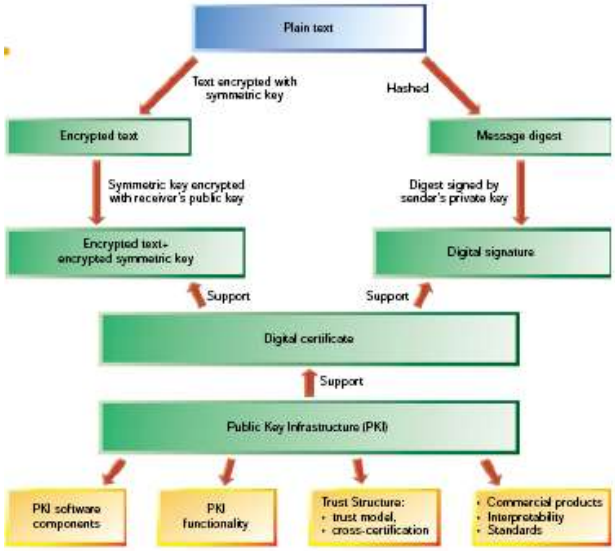

Figure 1. Technical building blocks of secure Internet messaging

2. THE PROBLEM: EXCHANGING CONFIDENTIAL MESSAGES OVER THE INTERNET

Messaging (also called *electronic messaging*) discussed here refers to the exchange of text, images, voice, telex, fax, e-mail, web pages, Electronic Data Interchange (EDI) data and other information over a public data communications network. In order to conduct business transactions over the Internet, electronic messaging must satisfy four properties:

- Confidentiality: No-one apart from the communicating parties can read the details of the transaction
- Data Integrity: No-one can tamper with the details of the information en route
- Authentication: Both communicating parties can positively identify each other
- Non-repudiation: Evidence that a specific transaction has taken place can be provided, so that neither party can deny it

Messaging security is important in electronic commerce, especially in financial systems. In the real world we rely on physical security, for example, we seal the envelope to protect the confidentiality of the mail; we sign the letter to ensure authentication, integrity and non-repudiation. In the world of electronic commerce, a message can be intercepted, read (the *confidentiality* issue) or modified (the *integrity* issue); people may send message under someone else's name (the *authentication* issue) or deny sending message (the non-repudiation issue). These problems need to be addressed for effective messaging in the Internet world.

3. DIGITAL APPROACH - STEPS IN DETAIL

For example, A (sender) has a plain text file that needs to be sent to B (recipient) through the Internet securely. These procedures can be followed:

Figure 2. Sending secure mail through Internet

a) Digital Signature is added to sender A's plain text file
b) Document with signature is compressed
c) Compressed file is encrypted with one-time session key††
d) Session key is encrypted using receiver's public key before attaching to the encrypted file
e) File is converted to ASCII format
f) Message is transmitted through the Internet by sender A

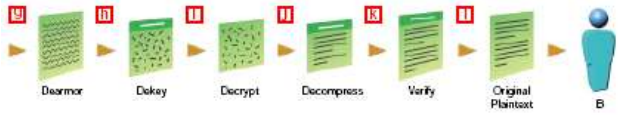

Figure 3. Receiving secure mail from Internet

g) Message is received by recipient B in ASCII format
h) ASCII is converted to original encrypted format
i) One-time session key is recovered using recipient's private key
j) File is decrypted using one-time session key
k) File is decompressed revealing signature and plain text message
l) Signature is verified using sender's public key

The concepts of digital signature (used to ensure integrity, non-repudiation and authenticate the sender), session key (for fast encryption to achieve confidentiality) and public/private key will be discussed in detail in the following sections.

4. TECHNOLOGIES OVERVIEW

There are two basic kinds of encryption algorithms in use today: symmetric key algorithms and asymmetric key algorithms.

4.1 Symmetric Key Algorithms

Symmetric key cryptography, also called "secret-key" cryptography, is a traditional cryptographic paradigm, where the same key is used to encrypt and decrypt the message.
(See figure 4) Each pair of principles must share a "secret" key upon which they have previously agreed. Key lengths can be 40-bit, 56-bit, 128 bit, 256-bit etc. There are well-established standards of symmetric key algorithms:

- DES (Data Encryption Standard)
- IDEA (International Data Encryption Algorithm
- RC2
- RC4
- RC5
- AES

It should be noted that Advanced Encryption Standard (AES) is the symmetric key algorithm currently under development, which is intended to become a standard and will replace DES. AES is selected by a worldwide, peer review process (i.e. no back doors). For more information, see [15].

Figure 4. Symmetric key encryption algorithms

4.1.1 Advantages of symmetric key algorithms

- Provides strong encryption (See Table 1 for the time needed to break to key)
- Computationally simple and fast (Therefore, in the above example, the session key used instep (c) and (j) to encrypt the whole file is a symmetric key)
- Generally, the encrypted file has little data expansion

4.1.2 Disadvantages of symmetric key algorithms

- Key distribution: all parties must first securely exchange an encryption key before using the symmetric algorithm. Keys are not easy to update. This issue is addressed by using the asymmetric algorithm discussed below.
- Because more than one party has the same key, any one can modify the encrypted file, so there is no protection to document integrity, sender authentication or non-repudiation.
- Large number of keys: each user should have different keys for different partners, which results in requiring N*(N-1) /2 keys for N users.

†† The session key is a symmetric key, which is used for a single encryption session and is then discarded.

Average Times needed to search half the symmetric key-space (worst case scenario would be twice as long)					
Key Length (bits)	Individual Attacker	Small Group	Academic Network	Large Company	Military Intelligence Agency
40	weeks	days	hours	milliseconds	microseconds
56	centuries	decades	years	hours	seconds
64	millennia	centuries	decades	days	minutes
80	infeasible	infeasible	infeasible	centuries	centuries
128	infeasible	infeasible	infeasible	infeasibles	millennia

Table 1. Brute force attack

Assumptions are based on 1997 technology:
Individual Attacker: one high-end desktop machine and software *(217 - 224 keys / second)*
Small Group: 16 high-end machines and software *(221 - 224 keys / second)*
Academic Network: 256 high-end machines and software *(225 - 228 keys / second)*
Large Company: $1,000,000 hardware budget *(243 keys / second)*
Military Intelligence Agency: $1,000,000,000 hardware budget and advanced technology *(255 keys / second)*

4.2 Asymmetric Key Algorithms

Asymmetric key cryptography, also called "public-key", is a new cryptographic paradigm. Each user has a pair of keys: a published "public-key" and a secret "private-key". Either one key can be used to encrypt. The encrypted file can only be decrypted by the other key. (See figure 5) This pair of keys is created simultaneously using the same algorithm. The public key is made publicly available (as part of a digital certificate) in a directory of PKI that all parties can access. The private key is never shared with anyone or sent across the Internet. The private key is known
ONLY to the key holder and acts as the identity of the key holder, which makes it possible to grant *authentication* and *non-repudiation*. The private-key cannot be derived from the public-key. Key lengths can be 512-bit, 1024-bit, 2048-bit, etc. RSA is a well-known public key algorithm. Other public key algorithms include:

- Diffie-Hellman
- Elliptic Curve Cryptography

Figure 5. Asymmetric key encryption algorithms

4.2.1 Advantages of asymmetric key algorithms

- Provides strong *confidentiality* - Encryption can be done using receiver's public key and the intended receiver can recover the secret with his/her private key. In the step (d) and (i) of the above example, the session key is encrypted using receiver's public key to be transmitted through the Internet.
- Provides sender *authentication* and *non-repudiation* - A sender can sign using his/her private key. In the step (a) of above example, digital signature is generated by the sender's private key. See below for a detailed explanation on digital signature.
- Encryption key (public-key) is different from the decryption key (private-key), thereby allowing the possibility of a third party verifying the identity of the sender without being able to recover the secret.
- No need for the distribution of secret keys
- Keys are easily changed
- Number of keys = N pairs for N users

4.2.2 Disadvantages of asymmetric key algorithms

- Computationally intensive, which makes 30~100 times slower than the equivalent symmetric key encryption algorithm. (That is why only the session key is encrypted by the public key, not the whole file)
- Slight data packet expansion
- Minimum encryptable data set is large (> 1000 bits)

Symmetric Ciper (Conventional)	40 bits	56 bits	64 bits	80 bits	96 bits	112 bits	120 bits	128 bits
Public Key Asymmetric (RSH,DSA,DH)	274 bits	384 bits	512 bits	1024 bits	1536 bits	2048 bits	2560 bits	3072 bits

Table 2. Key length comparison. In terms of computing power needed to compromise the encryption, 80-bit of symmetric key length is equivalent to 1024-bit of RSA asymmetric key length.

4.3 Digital Signature

As discussed above, *confidentiality*, *authentication* and *non-repudiation* can be achieved using the combination of symmetric and asymmetric encryption algorithms.

4.3.1 Integrity

Digital signature is built to support *integrity*, as well as *authentication* and *non-repudiation*. Using public-key cryptographic algorithms to encrypt messages is computationally slow. So some math algorithm, called hash function††, is used to generate a short, unique representation of the original message, call a *message digest*, which can be encrypted and then used as a digital signature. For different messages, you will have different digital signatures.

4.3.2 How Digital Signature Works

For the case listed above, in step (a), (See figure 6)

1) Sender A copies-and-pastes his/her plain text files into an e-mail note

2) Using special software, he/she obtains a message hash (mathematical summary) of the file
3) He/she then uses the private key to encrypt the hash
4) The encrypted hash becomes the digital signature of the message, i.e. Person A confirms the content of the message by 'signing' it with the private key. (Note that different message will have different digital signature)

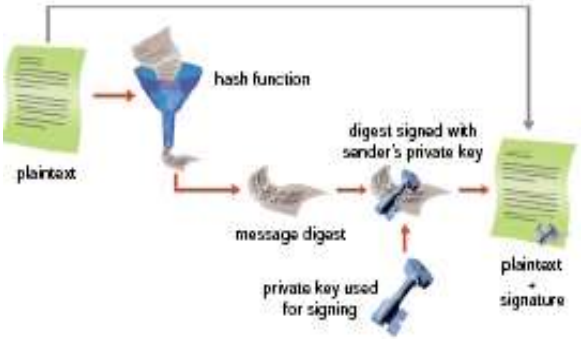

Figure 6. Signing a digital signature

At the other end, person B receives the message, in step (l), (See figure 7)

1) To make sure it's intact and from person A, person B calculates a hash of the received message using the same hash algorithm.
2) Person B then uses person A's public key to decrypt the message hash in the mail and recover the original hash.
3) If the calculated hash matches the original hash, the received message is confirmed to be from sender A without any change during transmission.

Figure 7. Verifying a digital signature

Since the sender encrypts ('signs') the hash with his private key, which is known only to him, this is to confirm the contents of the message and authenticate the identity of the sender of a message (*authentication* and *non-repudiation*). The hash is a unique digest of the message, so it ensures that the original content of the message or document that has been conveyed is unchanged (*integrity*). Additional benefits to the use of a digital signature are that it is easily transportable, cannot be imitated by someone else, and can be automatically *time-stamped.* So far all the issues for messaging security (confidentiality, integrity, authentication and nonrepudiation) have been addressed by using symmetric / asymmetric encryption algorithms and digital signature technology. Message is encrypted by a symmetric key; the symmetric key is encrypted and exchanged using public/private key.

Now how is public key being distributed? *Alternatively, how can person A trust that person B's public key is really from the right person B?* Digital certificate and public key infrastructure (PKI) is developed to take care of this issue.

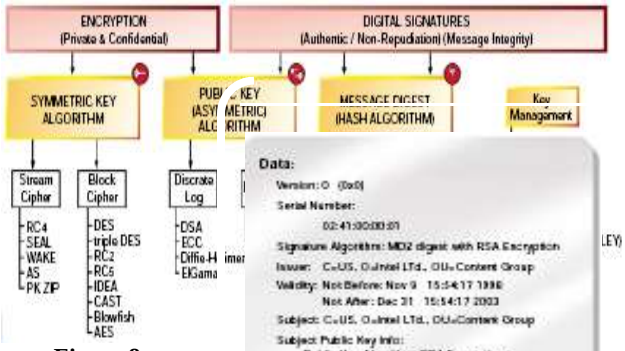

Figure 8. Summary of cryptography standards (Source: Security* Incorporated)

Figure 9. Digital certificate

5. DIGITAL CERTIFICATE AND PKI

5.1 Digital Certificate

As your driver's license or passport is your ID in the real world, a digital certificate is your electronic "identification card" used on Internet. It is issued by a certification authority (CA) and contains,
among other things, *your public key* (see details below). Digital certificates can be kept in CA's registries so those authenticated users can look up other users' public keys. X.509 v3 certificate is a popular standard for public key certificates. PGP v5 is another. As shown in figure 9, an X.509 certificate consists of the following fields:

- The X.509 version number — this identifies which version of the X.509 standard applies to this certificate, which affects what information, can be specified in it. The most current is version 3.
- The serial number of the certificate - the entity (application or person) that creates the certificate is

†† Hashing is the transformation of a string of characters into a usually shorter fixed-length value or key that represents the original string. A minimum change in the original file will result in a totally different hashed file. The same hashing algorithm is shared by the sender (A) and the recipient (B). Some well-known hashing algorithms include MD2, MD4, MD5 and SHA.

responsible for assigning it a unique serial number to distinguish it from other certificates it issues. This information is used in numerous ways; for example when a certificate is revoked, its serial number is placed in a *Certificate Revocation List* or *CRL*.

- Issuer signature algorithm: — identifies the algorithm used by the CA to sign the certificate: hash algorithm (e.g. MD2) + encryption algorithm. (e.g. RSA)
- The unique name of the certificate issuer — the unique name of the entity that signed the certificate. This is normally called a CA. Using the certificate implies trusting the entity that signed this certificate. (Note that in some cases, such as *root* or *top-level* CA certificates, the issuer signs its own certificate)
- The certificate's validity period — the certificate's start date/ time and expiration date/ time; indicates when the certificate will expire.
- Certificate holder's name
- The certificate holder's public key — the public key of the certificate holder, together with an algorithm identifier. This specifies which crypto system the key belongs to and any associated key parameters.
- Issuer unique identifier
- The certificate holder's unique identifier — (or *DN — distinguished name*). This name is intended to be unique across the Internet. A DN consists of multiple subsections and may look like this: CN=Bob Allen, OU=Content Group, O=Intel Ltd., C=US (These refer to the subject's *Common Name*, *Organizational Unit*, *Organization*, and *Country*)
- Extensions: Different CA can add different information in this field. For example, the picture of the user, the authorizations say what you are permitted to do.
- The digital signature of the issuer on the above fields - the signature using the private key of the entity that issued the certificate.

More information of X.509 standard can be found at ITU's "Recommendation X.509 (08/97)" [2] or the Appendix A of NIST's report [1].

Digital certificate provides a standard format of presentation for public keys. These certificates are stored in *trustable* data center and made available on users' request. This function is managed and performed by a Public Key Infrastructure.

5.2 Public Key Infrastructure (PKI)

The main service provided by PKI is to issue digital certificates and to make them widely available and accessible. Just as a telephone book contains a list of names and numbers, a PKI directory lists the public key with an individual or organization. PKIs solve the key management problems: creation, distribution, authentication, and storage of keys. A public key infrastructure consists of (see Figure.10):

- A certificate authority (CA) that issues and verifies digital certificates
- A registration authority (RA) that acts as the verifier for the certificate authority before a digital certificate is issued to a requestor
- Directory Service (DS) where the certificates (with their public keys) are stored and made available (usually in an ITU X.500 standard directory)
- End Entities (EE), which are certificate holders — users, organizations, applications, etc
- Clients that uses the PKI to obtain certificates and validate certificates and signatures;
- A certificate management system

A complete description of the functionality of each of the PKI components can be found in Chapter 2 of NIST's documents [1].

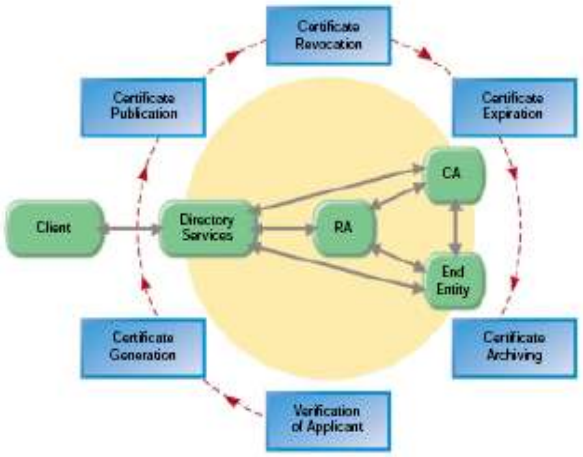

Figure 10. PKI Components and Services

The certificate management system should provide the following service:

- Certificate revocation: canceling a previously issued certificate, creating and publishing Certificate Revocation Lists (CRLs), storing and retrieving certificates and CRLs.
- Certificate expiration and update
- Certificate archiving: to ensure being able to decrypt anything encrypted in the past
- Certificate backup and recovery: to ensure that information is not lost
- Non-repudiation support: key backup and recovery open a loophole in the system. Now a user could claim that because someone else potentially has access to their signing key, they are not completely responsible for such specific transactions, although these transactions were apparently signed by them. It is thus necessary to maintain dual key pairs for every user. The encryption key pair can be backed up and recovered, whereas the signing key pair should never leave the user's possession.
- Time-stamping
- Policy-based certificate validation, e.g. cross-certification

A more detail explanation of PKI functionality can be found at page 3 of NSS' report [12].

5.3 Trust models

The PKI system discussed above gives us the building blocks of security. In this section, we are going to talk about how to use this technology to build up a network of *trust*. In relatively closed systems, such as within a small company, it is easy to trace a certification path back to the root CA. However, users must often communicate with people outside of their corporate environment, including some of whom they have never met, such as vendors, customers, clients, associates, and so on. Establishing a line of trust to those who have not been explicitly trusted by your CA is difficult. Companies follow one or the other *trust model*, which dictates how users will go about establishing certificate validity. There are three different models:

- Direct Trust
- Hierarchical Trust
- A Web of Trust

5.3.1 Direct Trust

Direct trust is the simplest trust model. In this model, a user trusts that a key is valid because he or she knows where it came from. (See figure 11) All crypto-systems use this form of trust in some way. For example, in web browsers, the root Certification Authority keys are directly trusted because they were shipped by the manufacturer. If there is any form of hierarchy, it extends from these directly trusted certificates.

Figure 11. Direct trust

5.3.2 Hierarchical Trust

In a hierarchical system, there are a number of "root" certificates from which trust extends. (See figure 12) Consider it as a big trust "tree." The "leaf" certificate's validity is verified by tracing backward from its certifier, to other certifiers, until a directly trusted root certificate is found. Then, there may be two or more certificates enclosed with the message, forming a hierarchical certificate chain, wherein one certificate testifies to the authenticity of the previous certificate.

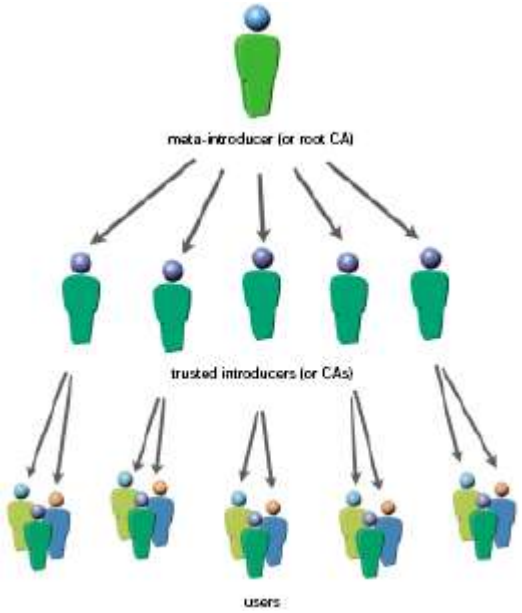

Figure 12. Hierarchical trust

5.3.3 Web of Trust

A web of trust encompasses both of the above models, but also adds the notion that trust is in the eye of the beholder (which is the real-world view) and the idea that more information is better. It is thus a cumulative trust model. (See figure 13) A certificate might be trusted directly, or trusted in some chain going back to a directly trusted root certificate, or by some group of introducers. Any user can validate another user's public key certificate. Any user can act as a certifying authority. PGP is using this web of trust model.

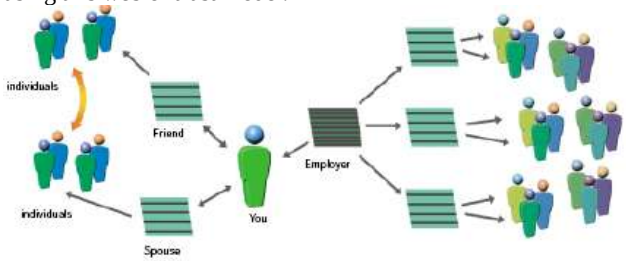

Figure 13. Web of trust

It should be noted that all models above are equally valid in the PKIs as well as in the physical world. The reason for using any special one model is purely dependent on the nature of individual case.

5.4 Cross-certification

The *web of trust* sounds straight forward as discussed above. In practice however it is quite complicated. This will be demonstrated in this section.

For example, two trading partners, each with their own root CA, may want to validate certificates issued by the different root CAs. This requires the CAs' *cross-certification*, which means one CA (e.g. Intel) issues to another CA (e.g. Microsoft*) a certificate that contains its (Microsoft's) public key. Cross-certification refers to two operations. The first operation is the

establishment of a trust relationship between two CAs. CA-Intel and CA-Microsoft exchanges public keys securely. After that, each CA signs the other CA's public key in a certificate, the "cross-certificate". In the second operation, the clients verify a user certificate signed by a cross-certificate CA.

Some operational and legal issues of cross-certification to be addressed are:

- Are the certificate policies under the two CAs comparable in terms of the trust level achieved?
- Are the processes and procedures in the other CA up to the same standards to maintain the trust?
- Who accepts liability?
- What are the implications if one of these CAs wishes to unilaterally cross certify with a third CA?
- Are the certificates from different CAs technically acceptable by the same applications? (See Section 5.2 for further discussion)

These issues are complicated enough, that very few CA cross-certifications exist in the world so far. One example is Thawte*[8], a company providing cross-certifications to other SSL and S/MIME CAs (See 5.1 for more). (Thawte was just acquired by VeriSign* in December, 1999) The government of Canada is also developing the policy setting procedures and criteria for cross-certification with the government CA [9]. This currently is a draft for discussion, but it is a good reference with practical details for implementation of cross-certification.

6. CONCLUSIONS

This paper describes the technical building blocks for messaging security on the Internet and its related technologies. We started with messaging security on Internet, which is encrypted using symmetric key. The symmetric key itself is encrypted by public key to be transmitted through the Internet. The public keys are embedded in digital certificates and available through public key infrastructure...

To implement these in applications like Internet banking and online stock trading, additional factors need to be considered with reference to *policies*, *legal* [13] and *operation.* An example of a PKI case study of the e-tax system for the Australian Tax Office can be found on the Internet [14]. To successfully deploy a PKI, organizations must develop a sound strategy, plan for interoperability, determine how e-business applications will interface with the PKI, and plan for technical staffing requirements.

REFERENCE

[1] NIST Special Publication 800-15, Minimum Interoperability Specifications for PKI Components, Version 1, 1997, http://csrc.nist.gov/pki/documents/mispcv1.doc;

[2] ITU [X.509] Recommendation X.509 (08/97) - Information technology - Open Systems Interconnection - The Directory: Authentication framework, http://www.itu.int/itudoc/itu-t/rec/x/x500up/x509.html

[3] RSA Data Security Inc. "Public Key Cryptography Standards, PKCS#s 1-11", http://www.rsasecurity.com/rsalabs/pkcs/index.html

[4] RSA Frequently Asked Questions About Today's Cryptography,http://www.rsasecurity.com/rsalabs/faq/question.html

[5] Concept definitions, Whatis.com, http://www.whatis.com/

[6] Simson Garfinkel and Gene Spafford, "Web Security & Commerce", (O'Reilly Nutshell, June 1997)

[7] PKI Forum, http://www.pkiforum.org/

[8] Thawte CA cross-certification service white paper, http://www.thawte.com/certs/chained/whitepaper.html

[9] Government of Canada PKI Cross-Certification Methodology and Criteria, http://www.tbs-sct.gc.ca/cio-dpi/pki/documents/x_cert/meth_e.pdf (Draft, September 22, 1999)

[10] Ietf-pkix working group, http://www.imc.org/ietf-pkix/

[11] Dave Kosiur, "Building and Managing Virtual Private Networks", (John Wiley & Sons , October 1998)

[12] NSS Group's evaluation of PKI products, http://www.nss.co.uk/download.htm

[13] Government of Canada PKI policies,http://www.cio-dpi.gc.ca/pki/Documents/documents_e.html

[14] PKI case study "e-tax system for the Australian Tax Office",http://www.baltimore.com/download/baltimore_succes_stories.zip

[15] NIST Advanced Encryption Standard (AES) Development Effort, http://csrc.nist.gov/encryption/aes/aes_home.htm