Global Information Assurance Certification Paper

Implementing PKI: A Real Challenge!!

Juan Valbuena

Version: 1.4b
Option 1
13th February 2003

Abstract

The aim of this paper is to discuss some of the considerations that need to be taken when implementing PKI. The paper will also cover some concepts and terminology related to PKI. Also I have included some points on why a PKI Managed Service is some times more beneficial than an in-house PKI implementation. Also discussed is the importance of the policies that makes PKI implementation trustworthy for their relying parties.

Although it is important to every implementation, it is out of the scope of this paper discuss issues related with Personnel and Roles during the implementation of a PKI, and management protocols between components (i.e. CMP). Also it is assumed that the reader of this paper is familiar with PKI technology.

Introduction

Cryptography has been used for hundreds of years with the main objective of having confidentiality in the transmission of sensitive information across public or insecure pathway. The information was then secured using a secret "word"[1] that only was known by the parties involved. So the receiving party was able to use the same secret "word" to decode the message. The big issue then, was to securely transmit the "word" between the parties, so no one else could get access to it.

It was not before than Whitfield Diffie and Martin Hellman from Stanford University published what is known as the first publication on Public Cryptography, that the idea of having secured communication between two entities without previous exchange of a shared secret ("word") was possible.

In this publication (New Direction in Cryptography) Diffie and Hellman describes the key concept of Public Key Cryptography including digital signatures. One year later (1977), Ronald Rivest, Adi Shamir, and Leonard Adleman (RSA for their Initials) create the first implementation of the Diffie-Hellman algorithm. [2]

That was the origin what is known today as Public Key Cryptography. A concept that has revolutionized the electronic world and has changed the way of doing business. Since then, Public Key Cryptography has been the base for e-business and electronic

[1] The term "Word" will be change later for "Key", that is the term used in Cryptography.
[2] [PKC]

transactions, relying on the fact that two parties can exchange secure communication and authenticate themselves without any previous exchange of secrets of any kind.

Today's electronic world is based on the Internet. That huge network that allows us not only to share information but also to have other types of relationships (i.e. commerce) with third parties, even if we do not know them previously. But, How can that be possible? How can I do business with others that I do not know yet? How can I ensure that the other person is who he/she says he/she is? What is the starting point if I want to participate?

The main objective of this article is to answer those questions and to show some of the considerations that should be taken into account when implementing PKI. I do not intend to cover all of them, but at least the most relevant for usual implementations. But before get into details of any implementation is important to show the key concepts behind Public Key Cryptography.

Before Any Implementation – The Background Information

To understand the term of Public Encryption Infrastructure, we first need to look at some cryptography concepts and terminology.

The first term is cryptology; Cryptology is the practice and study of encryption and decryption of a message (message can be anything, from the text of an email, a document or a file)[3]. In other words, encryption it is the process by a clear text message (message that is readable) is encoded using mathematical means[4] to a bunch of text that is mean less (ciphertext). On the other hand, the Decryption process is the way that that the ciphertext is decoded using the same mathematical means but in opposite way, so it becomes readable again.

To perform this encryption/decryption process we need to introduce the encryption and decryption keys, which are secret values (previously called "words") used to encode and decode the message. These values can have different lengths, which are directly related to the strength of the key and, in fact, in the security of the encoded message.[5]

For example, a 30-bit encryption key has 1,073,741,824 possible variations, which any computer able to perform one million of combinations per second, will guess (crack) the key in just 18 minutes (That's any regular PC), while a 56-bit encryption key will take to the same computer 2,285 years to crack the 56-bit long key (n= length of the key, time=2^n).

[3] [ENT1]
[4] Also called encryption algorithm, or cipher.
[5] [RAIN]

Many encryption algorithms are commonly used today. Most of these can be categorized as symmetric or asymmetric key algorithms.

Symmetric and Asymmetric Cryptography

Symmetric encryption refers to the system that uses a single key (secret key or symmetric key) for both the encryption and decryption operation. In this encryption technique, the challenge is to securely exchange the shared key between the participants of the communication. Even worst, when there is not previous relationship between the participants. Also as the number of participants increase, it becomes problematic the management of the keys, as each one has his/her own key.

The main advantage of this kind of algorithm is its performance, due to the algorithm is not as complex as the one used for asymmetric encryption algorithms (explained later).

Some of the algorithms used today can be seen in the table below together with the key length used for each algorithm. As we can see the key length may vary and it will depend on the algorithm used.

Algorithm	**Key Length**
Data Encryption Standard (DES)[6]	56 bit
Triple Des (3DES)[7]	3 x 56 bit
CAST[8]	40 to 256 bit
River Cipher (RC2. RC4)[9]	Variable
International Data Encryption Algorithm (IDEA)[10]	128 bit
Rijndael[11]	128, 192 or 256 bits
Blowfish[12]	32 bits to 448 bits

Table 1: Algorithms use for symmetric encryption. – Source Entrust Courseware[13] and Juan Valbuena

On the other hand, Public Key Cryptography (PKC), also known as asymmetric encryption uses two different keys, mathematically related to each other that perform opposite functions. What one key encrypts is decrypted with its pair.

[6] [DES]
[7] [3DES]
[8] [CAST128] [CAST256]
[9] [RC2] [RC4]
[10] [IDEA]
[11] [AES]
[12] [BLOW]
[13] [ENT1]

The private key is created first and the public one is generated when applying one-way math function to the private key. It is virtually impossible to determine a private key from the public key[14]. Keeping that in mind, the public portion of the key pair can be freely distributed in a public manner without compromising the private portion, which must be kept secret by its owner. If an encryption is performed using the public key, only the private key can be used to recover the original message.

The main advantage of this kind of algorithm is that there is not need to transmit any secret or shared key between the sender and the recipient and only requires one pair of key per user, but since it uses a more complex algorithms, the performance transforming clear text data into a ciphertext is relatively slow. The key length also varies from 512 to 2048 bits long. Some of the algorithms available today can be seen in the table below.

Algorithm	**Develop by**
RSA [15]	Ron Rivest, Adi Shamir, Leonar Adleman
Digital Signature Algorithm (DSA) [16]	NSA for NIST
Elliptic Curve Digital Signature Algorithm (ECDSA) [17]	Neal Koblitz and Victor Miller
Diffie-Hellman Key Exchange Algorithm [18]	Whitfield Diffie and Martin Hellman

Table2: Asymmetric Algorithms – Source Entrust Courseware and Juan Valbuena

Using these two techniques together, we can securely exchange information and avoid the disadvantages of the two cryptographic systems. First of all, remember that the symmetric cryptography is fast, but the exchanging of the shared key is an issue. And asymmetric cryptography is slow but is possible to secure a channel without previous knowledge of the other party. So, why not secure the shared key using the recipient's public key? That would give us the confidence that only the recipient will be able to decrypt the shared key. Then using the symmetric cryptography with the shared key, we can encrypt and securely transport the message.

That was one of the initial uses of asymmetric cryptography, of facilitate the delivery of keys to be used in symmetric cryptographic functions. [19]

[14] [SANS]
[15] [RSA]
[16] [DSA]
[17] [ECDSA]
[18] [PKC]
[19] [PKIB]

But still one problem remains. How we can ensure that the recipient's public key belongs to the person we think it belongs to? Well, to solve that issue, Diffie and Hellman also work on what is called the digital signature.

To understand digital signatures, we need first need to understand another type of algorithm – The message-digest algorithm.

Message-digest is a method of reducing a message of any length to a string of a fix length, called message-digest or hash, in such a way that it is computationally infeasible to find two messages with the same result or to find a message with a given, predetermined message digest. Some of the most popular message-digest algorithms can been seen in the table below (Table 3). Using this algorithm together with the Private Key, we can obtain what is called the Digital Signature.

Algorithm	**Digest Size**	**Developed by**
Message Digest 5 (MD5)	128 bits	Ron Rives (RSA data Security)
Secure Hash Algorithm (SHA-1)	160 bits	NIST/NSA

Table 3: Message-digest algorithms, Source Entrust Courseware[20]

The digital signature then consists in two simple steps:

1) Apply the message-digest algorithm to the message and obtain the hash.
2) Encrypt the hash using the private key

Considering that the owner securely keeps the private key, no one else could perform such action on the hash with the same result.

The e-Business Requirements

Using a combination of secret key and public key cryptography together with the message-digest algorithm, also called hash function, PKI enables a number of security services including data confidentiality, data integrity, authentication, and non-repudiation support. Let's see what each of those mean.

Confidentiality: It is the actual secrecy of the information as it is passed or stored. Sensitive information should be available only for the people that have the right to see it, and hidden for others.

Integrity: It is the assurance that data has not changed while in transit or over time.

Authentication: It is the proof of identity of a person to ensure that he is who he says he is.

[20] [ENT1]

Non-repudiation: It is the inability of an individual to deny the involvement or association with a performed action.

All of the security services mentioned above must be utilize to maximize the advantages of e-business transactions. PKI provides the means and the infrastructure to effectively deliver these requirements. Thanks to the encryption, PKI provides Confidentiality, and thanks to the digital signature, it provides Strong Authentication, Data Integrity and support for non-repudiation.

Just an Example!

To understand how PKI fulfills these security services, let's see PKI in action:

First of all, the encryption process. We can see in the following example, how a user can send confidential information (let's say an email) to someone else. Assuming that both of them have already a pair of keys (Private and Public).

Anyone who has read anything about PKI in the past should be familiar then with Alice and Bob. Let's use them in our example.

1. Bob will prepare the message (sensitive information) he wants to send to Alice.
2. Bob generates a random key, which will be used to encrypt the message. This random key is a one-time use symmetric key.
3. Bob uses the symmetric key to encrypt the message using any of the symmetric algorithms (DES, 3DES, etc) and gets the ciphertext.
4. Then Bob using Alice's public key encrypts the symmetric key and gets the encrypted key.
5. Put the cipher and the encrypted key together, and then send them to Alice.
6. Alice receives both cipher and encrypted key.
7. Alice using her private key decrypts the encrypted key and gets the symmetric key.
8. And then, Alice using the symmetric key decrypts the cipher and gets the original message.

Few things to highlight:

1) Since the symmetric key was encrypted using Alice's public key, the key can be transported securely through an insecure pathway and only Alice will able to decrypt it.
2) Bob can be confident that only Alice will read message, since he used her public key.
3) No one else will be able to read the message, since Alice is suppose to keep her private key secure and out of the reach of others.

The previous example shows how the public key and the symmetric cryptography can be use to securely transmit any sensitive message.

Now in a second example we can see how PKI using digital signatures can provide Integrity and authentication. Let's say, Alice wants to send back a message to Bob, but at this time the message is not that confidential; but still she needs to prove that she sent it and be able to verify that the message will not change while in transit. So, what she wants to do is to digitally sign the message (based on the concepts of Digital Signature); the process will be as follow:

1. Alice prepares the message to be sent.
2. Alice applies the hash function to the message and gets the hash
3. Alice then uses her private key to encrypt the hash and gets a cipher of the hash.
4. Alice puts together the cipher of the hash together with the original message
5. Alice sends them to Bob.
6. Bob receives the message and he separates the original message from the cipher of the hash.
7. Bob now uses Alice's public key to decrypt the encrypted hash.
8. Using the same Algorithm Alice used to get the hash, bob calculates his own version of the hash from the message he got from Alice.
9. Bob then compares the two hashes (one received from Alice and the one calculated by himself). If they are exactly the same, the message he received is the same that Alice sent.

Few things to highlight:

1) Again, Alice is suppose to keep her private key secure and out of the reach of others
2) Bob, relying on Alice private key, can be confident that the message came from Alice, since he uses Alice's Public key to decrypt the hash. That's authentication.
3) Bob applies the same function as Alice did, and the hash he obtained from the message is the same as the hash sent by Alice. So the message is exactly the same. That's data integrity.

In this second example, how confident is Bob that the message came from Alice depends on how confident he is that the Alice's key really belongs to Alice. This may not be difficult, if Alice and Bob have or have had a direct relationship in the past and the public key has been handed to Bob in a direct way. Otherwise Bob could doubt that the message is coming from Alice.

Using previous examples, we can see that if Alice handles her private key in a trustworthy manner (the private key is kept secured, on not reach of anybody else but herself), Bob can indeed be confident that no one else has access to Alice's private keys, then the message is coming from Alice.

When direct trust is applied to secure communications, it is the sole the responsibility of each of the parties to ensure that they are comfortable with their level of personal trust in the other party. When direct trust is present, key exchanges among individuals with personal relationships provide a powerful mechanism to ensure secure communications.

In a network security solution there is another important form of trust that we need to understand: third-party trust. Entrust Industries defines the Third-Party trust as:

> *"Third-party trust refers to a situation in which two individuals implicitly trust each other even though they have not previously established a personal relationship. In this situation, two individuals implicitly trust each other because they each share a relationship with a common third-party, and that third-party vouches for the trustworthiness of the two people.*
>
> *Third-party trust is a fundamental requirement for any large-scale implementation of a network security product based on Public Key Cryptography. Public Key Cryptography requires access to users' public keys. In a large-scale network, however, it is impractical and unrealistic to expect each user to have previously established relationships with all other users. In addition, because users' public keys must be widely available, the association between a public key and a person must be guaranteed by a trusted third party to prevent masquerading. In effect, users implicitly trust any public key certified by the third-party because their organization owns and operates the third-party certification agent in a secure manner."* [21]

In this type of trust we are relying on a third-party component that is able to provide assurance and guarantee that all the entities under its domain of trust[22], can trust each other without previous relation. (i.e. A trusts B and A trusts C, then B trusts C and C trusts B).

[21] [ENT2]
[22] The Third-party trusts each of the individuals that belong to its domain.

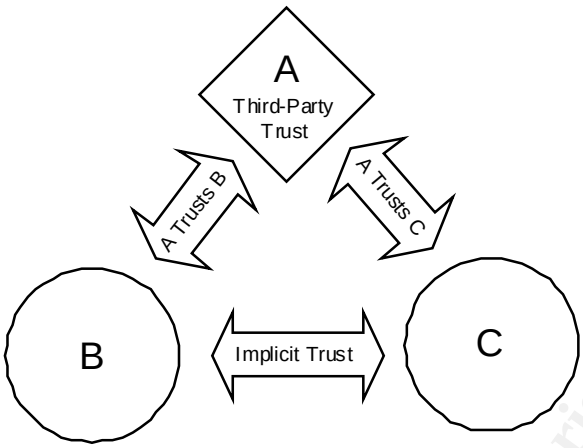

Figure 1: Third-Party Trust Example – Source Juan Valbuena

In PKI, the third-party component is the Certification Authority.

The Certification Authority

A Certification Authority (CA) is a trusted entity whose central responsibility is certifying the authenticity of users. Network user's "electronic identity," issued to him/her by a CA, is that user's proof that the CA trusts him; therefore, through third-party trust, anyone trusting the CA should also trust the user.

The way the CA is able to certify the authenticity of a particular user is by issuing an electronic identity also known as a Digital Certificate. From now on, this Digital Certificate will be the digital identity of the users in the PKI system.

Digital Certificates – X.509

A digital certificate is the ID in the electronic word and it is analogue to any physical ID on use today (i.e. Passport, National ID, Driving License, etc.). it contains the name of the owner, unique serial number or identification number, other user-related information and the owner's Public Key.

To establish trust in the binding between a user's public key and other information (e.g., name) in a certificate, the CA digitally signs the certificate information using its signing private key.

The CA's digital signature provides three important elements of security and trust to the certificate.

1) By definition, a valid digital signature on a certificate is a guarantee of the certificate's integrity.

2) Since the CA is the only entity with access to its signing private key, anyone verifying the CA's signature on the certificate is guaranteed that only that CA could have created the signature.
3) Since only the CA has access to its signing private key, the CA cannot deny signing the certificate (support to non-repudiation).

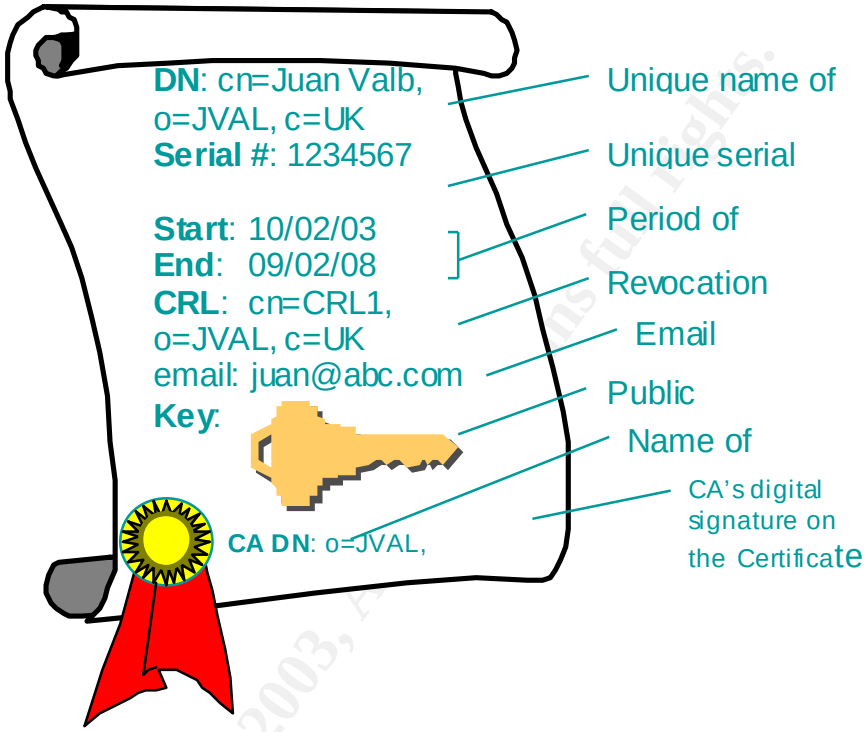

Figure 2. Digital Certificate Example. – Source Juan Valbuena

Why we need a repository?

There is a need to distribute certificates in such a way that other users in the same domain of trust can use them. Regarding our example (see Just an Example!) we assumed that Bob had Alice's public key, but never discussed how he got it. Well, now that we can use our earlier example to explain the need of a certificate distribution point (or repository).

Just before encrypting using Alice's public key (step 4), somehow Bob needs to get Alice's public key, but now he needs to be sure that key belongs to Alice. (Before we assumed Bob was confident about that, but now we assume they even do not know each other).

Then, Bob need to get the Alice's Digital Certificate, vowed by a CA that the certificate is valid and authentic. So Bob needs access to a public repository where he can get it.

Without it, Bob would still need some kind of direct contact (direct trust) with Alice in order to get it. In small organization this could be possible, but in large organizations would be a nightmare.

Now let's say that we have got a public repository for the certificates. So, Bob will need to get Alice's digital certificate that includes Alice's public key. Since the CA has signed Alice's digital certificate (CA that Bob trusts to), Bob can first validate the authenticity of the certificate validating the CA's digital signature, then verify that the certificate is still valid using public revocation information, and if everything looks okay, he can use Alice's public key embedded in Alice's certificate and encrypt the symmetric key, following the steps in our example.

On the other hand, verifying the CA's digital signature on the certificate, other users must also be sure that the certificate is still trustworthy at the time of use. The CA must revoke certificates that are no longer trustworthy.

There are many reasons why a CA needs to revoke a certificate. For example when a user's private key has been compromise (some else has had access to the private key), user's private key is lost, or may be the user just has left the company. In any case, the certificate should not be valid any more. Also, other users that rely on PKI will need a way to be able to determine if a certificate just received is still valid, although it is within its period of validity.

We mentioned earlier that the revocation information could be found in the repository. Well, this information is stored in the Public repository in a list form. Only the unique serial number is included in such a list. This list is better known with the name of Certificate Revocation List (CRL).

So, The CA is responsible for creating the revocation information (CRL) and makes it public to the other users as soon as possible[23].

"The revocation status of a certificate must be checked prior to each use. As a result, any PKI infrastructure must include a certificate revocation system. The CA must be able to securely publish information regarding the status of each certificate in the system".[24]

Another way to distribute revocation information is through Online Certificate Status Protocol (OSCP), where the OCSP responder takes the request from the clients and send them back the status of the certificate (Good, no good, unknown). Usually the OCSP gets the status of the certificates from the CRL (i.e. repository) or the CA itself.

Over the past few years, the consensus in the information technology industry is that the best technology for certificate repositories is provided by directory systems that are

[23] How often the CA publishes the CRL depends on what is stated on the company's policy.
[24] [ENT3]

LDAP (Lightweight Directory Access Protocol)-compliant. LDAP defines the standard protocol to access directory systems. [25]

What a complete PKI Implementation should have?

Based on the PKI overview previously discussed, we can think that the components to set up a PKI implementation are just a few, but other is the truth. One thing that it is true is that a PKI system is composed of Certificates and encryption keys. That's it!. But still we will need the help from other subsystem, components, and management functions to build a real PKI environment, able to provide Confidentiality, Data Integrity and Authentication, and Non-repudiation support and all that with a high level of trust and reliability. [26]

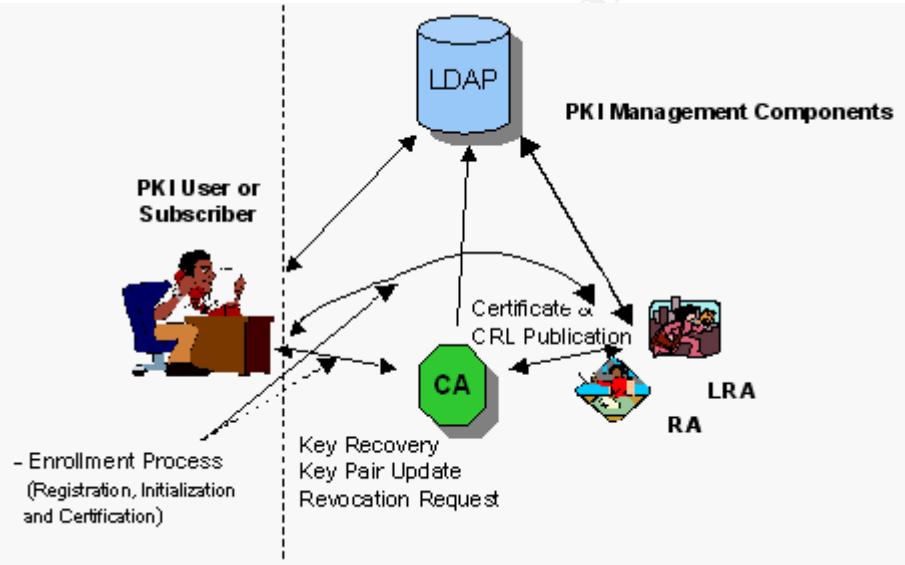

Figure 3: PKIX Architecture Model – Source Juan Valbuena

The Public Key Infrastructure based on X.509 (PKIX) [IETF] & [PKIB] (RFC3280) reflex the most recent version of the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. Figure 3 illustrates an interpretation[27] of this model. Here we can identify the components propose for the standard plus some variations[28]: End Users (subscribers), Certification Authority (CA), Registration Authority (RA), Local Registration Authority (LRA), Repository (LDAP-compliant Directory), and CRL Issuer.

[25] [ENT4]
[26] If the users do not trust their own PKI environment, PKI is useless!!
[27] This interpretation is based on the experience in real implementations.
[28] The standard PKIX contains a CRL issuer and Cross Certification, but does not contain an LRA

PKI Components

In the above picture, we can identify the different components and management functions that are included in a PKI implementation. These are the main parts but not all that might be included. As I stated before, this diagram is based on the standard but with some modification based on personal experience.

The Subscribers are the end-users of the system. But it does mean that they are only people; they can also be servers, workstations or routers. It can be anything that can be identified as a host in the network and that can have a name on a public-key digital certificate. One of the requirements for these subscribers is to enroll into the PKI before they can participate as members of the PKI.

The CA is the main player in a PKI implementation. The CA is responsible for issuing and publishing certificates to the LDAP directory. Also is responsible for including the revocation information in the CRL and publishing them in the LDAP directory. Remember that when the CA issues a certificate, it is binding the subscriber's information with the subscriber's public key.

The Registration Authority RA is another component of the PKI infrastructure. Although some people think that it is an optional component[29], my personal opinion and experience say that the RA plays a very important role on any med-size to large deployments. Basically the RA could be responsible practically for the deployment.

The main functions for a RA are as follow:

- Create an interface for the subscribers to perform day-to-day tasks. (i.e. request for enrollment, Revocation/Recovery operations, and change of details)
- Authenticating the user who is willing to enroll into the PKI system[30].
- Hand to the user the necessary information to proceed with the enrollment.
- Generation of shared secrets to support the initialization process
- Verifying that the subject has possession of the private key being registered.
- Public/Private key-pair generation
- Interact with the CA as an intermediary of the subscriber. (i.e. key compromise notification and key recovery request.)
- Parameter validation of public keys presented for registration.
- In some cases, the RA is responsible to communicate to the user the policies governing the PKI usage (see policies later)

[29] [PKIB]
[30] Remember that the successful authentication of the subscriber prior to enroll him/her/it ensures a trustworthy deployment of a PKI system.

In some cases when the deployment takes place in many different locations, a Local Registration Authority (LRA) should take the role of the RA locally. This LRA will have the same responsibilities as the RA but will not interact with the CA directly. The LRA will identify and authenticate the subscriber, request the digital certificate and the pair of keys to the RA, who will request it to the CA, and perform the tasks listed above.

When companies outsource de PKI service to external providers, the companies usually prefer to keep the Registration Authority inside. So they will perform the functions of the RA or LRA when applicable. This makes sense, since the company using PKI relies on a good deployment of certificates to only those who really belong to the same trust domain and are authorize to get one. Who knows better who is entitled to get one certificate rather than the company's people themselves?

As discussed earlier, the LDAP-compliant directories are the place where the certificates and CRLs are stored. Directories play as well a very important role into the PKI system. In my point of view, I think it is the most important component into a PKI system. Without it, PKI system is useless. Why? Let's see. What could happen if the directory is down for a while? Well, other users will not able to retrieve the public keys from the directory to encrypt to other users, or users will not be able to verify the authenticity of a specific certificate at certain time. Then, PKI will not be operative at all and the whole PKI system will be in the trash. There will be a big chaos until the directory is back on line.

So, that's why the importance to have as many replicas as possible. Can be one per office site, one per city, one per country or maybe one per continent. But the directory should be reachable to the PKI users and other subscribers (servers, routers, etc) all the time for the PKI to work.

PKI Management Functions

In the illustration (Figure 3) we can observe as well some of the management functions that a complete implementation of PKI should consider. This does not mean that all of them will need to be implemented, and it will depend only on the company's requirement.

Some of this management functions are the enrollment process (Registration, Initialization, Certification), and Key Management (Key Pair Recovery, Key Pair Update, and Revocation Request)

The Enrollment Process – Registration, Initialization and, Certification

The Registration process is the process by which the subscriber enrolls into a PKI system. During this step, the subscribers are identified and authenticated. The level of assurance during the registration process will tend to vary following the associated policies, intention of the certificate[31] (authentication, encryption, digitally signature), and the target environment. The RA will take responsibility of this process (LRA when applicable), but the CA could perform it when there is not RA.

Once the identity of the subscriber has been verified in accordance with the applicable policies, the end entity is usually issued on or more shared secret(s) and other identifying information (i.e. reference number and authorization code in entrust implementations) that will then be used for subsequent authentication as the enrollment process continues. It is responsibility of the subscriber to keep this secrets secure until the enrollment process ends. [32]

During the initialization process, the key pair is generated and associated with the subscriber and also associated with the root trust.[33] The key pair (Public/Private) generation usually takes place at the subscriber client system but some times it can also take place at the RA, CA or other secure system (i.e. Hardware Security Module). To support non-repudiation systems, typically the Certificate Policy applicable may dictate that the Private key of the subscriber must be generated at the subscriber system. So the subscriber will have complete control over the key from its generation.

Following the enrollment and the initialization processes, the CA will get the subscriber's information, bind it together with the subscriber's public and digitally sign it issuing the subscriber's digital certificate. This is the certification process.

Although enrollment, initialization and certification are three separate processes, some PKI implementation can combine one or, two in one.

Key Lifecycle Management

One thing we have not mentioned is that the certificate can be use for one or more purposes. It means that a digital certificate can be used only for Authentication, or only for encryption, or maybe both.

As stated by IETF, the usage of the certificate is defined in the *Key Usage* field and it is defined as follow:

---

[31] The Intention of the certificate can be one or more. This is defined in the Policy Certificate, which will be discussed at later stage in this article.

[32] [ENT2]

[33] The Root Trust or Anchor trust is the maximum level of trust, on which the PKI users rely to.

"The key usage extension defines the purpose (e.g., encipherment, signature, certificate signing) of the key contained in the certificate. The usage restriction might be employed when a key that could be used for more than one operation is to be restricted. For example, when an RSA key should be used only to verify signatures on objects other than public key certificates and CRLs, the Digital Signature and/or non-Repudiation bits would be asserted. Likewise, when an RSA key should be used only for key management, the keyEnciphement bit would be asserted."[34]

In the case the certificate (the key pair) is used for Encryption, there is the potential risk that the subscribers loose normal access to the key pair stopping them to decrypt confidential information encrypted previously[35]. So, it is important that a PKI system provides the way to the subscriber to recover the decryption keys, having access to them through time. Usually these encryption decryption key pair is kept by the CA under an automated backup system, easy to recover when the PKI user requests so by a process called key recovery (Usually this process involves the RA).

When the certificate is used for digital signatures, the CA is not able to recover them, since this pair of keys should only be kept under the user control. This may be defined in the security policies that govern PKI to support non-repudiation.

Also, as the certificate is issued with a fixed lifetime, it will eventually expire. Key update systems are required to update the subscriber's key pairs and extend their validity period (usually a new fixed period, same as original issuance). On the other hand the CA also may require updating the key pairs at various times. Some of the reasons for the CA for the updating could be: Changing the algorithm, changing the lifetime of either of the keys, or if the signing key has been revoked.

Since the trust of a PKI environment relies on the security policies in place (i.e. Certificate Policy) and in the good maintenance of the private key by the users, when a private key has been compromise, change of affiliation, name change or the user just has lost access to them (i.e. forgotten password/PIN), the user or RA need to revoke that certificate so will not be any longer valid.

The Hard Decision: In-House or Outsource

One of the key decisions to be made by businesses implementing PKI is whether to operate a private CA or use a third-party CA organization to operate a private CA. The technical aspects of this decision are focused on operational resources such as support

[34] [IETF]

[35] There are many way how the PKI users can loose access to their encryption/decryption keys i.e. Access PIN/password forgotten, smart card lost/stolen, client system crash among others.

facilities, in-house expertise and quality of service. Businesses requiring total control over their infrastructure and seeking to apply the system across an exclusively internal system may consider themselves best served by an in-house solution. But analysts say more businesses are open to outsourcing PKI that in the past, primarily because implementation is complex and it is hard to find and keep in-house IT talent[1]. The mission-critical nature of many PKI implementations requires diligence in the selection and contractual relationship with an outsourced provider.[36]

Implement PKI is not just to install some software on some servers and hand certificates to the end users. It needs planning, internal and external resources, appropriate physical facilities, time, and a good budget.

One thing to consider is that PKI is about trust, so if a company wants to implement a PKI solution for their internal use only, they would prefer rather to do it using internal resources for the planning, design implementation and maintenance. But still no many companies can afford to have the trained and experienced people to do the whole implementation and give the ongoing support.

Implementing PKI can be very expensive, and that's why, some times, it can be more beneficial to outsource it. As I said before, it is not just to install software and hand certificates.

The In-house approach

In an In-house implementation there are few key things to consider that can make a big difference compare with an outsource solution. Here they are:

Secure Facilities: There are some sensitive components in the PKI infrastructure that should be secured all the time. What makes a PKI implementation trustworthy is the way that it is set up, under strict control over the different components and the use of policies during the design and implementation. Also the critical and sensitive components should be securely kept following certain conditions that can make the system secure.

The main component here is the CA. As it is the only one that issues certificates, on which digital signature all the PKI subscribers rely on.

This computer room, where the CA and other PKI-business related equipment are stored, should be a highly secure room, with Physical Access control (i.e Proximity Cards, Smart cards, Biometrics, etc) and logical access control (i.e Firewall system, Intrusion detection systems, strong authentication, etc), 24x7 CCTV in operation with the security guards monitoring, at least two people accessing the room at the same time (no only one) and maximum three or four people have rights to access such a room, Air Conditioning System, flood proof, power resilience, onsite and outside backup system, and disaster recovery plan among others.

[36] [DIC]

Infrastructure: Beside the usual relievable network connectivity, we need to provide a system that is 100% (or as close as possible) available for the users. To provide this high availability solution we need to think about redundant systems. That means that we will need redundant communication links, double connectivity between systems (CA, Directories, Backup servers, etc), and others, always avoiding any single point of failure.

Hardware / Software: For a basic implementation of a PKI system we will need two servers and at least one workstation. That is for one CA and one directory, and the workstation for the RA. For more complex implementation we will need at least 6 to 10 servers and few more workstations. That is for one off-line CA, one On-line CA, Directory and replicas, few more for resilience and so on. The software license will depend on the vendor and the number of seats.

Consulting: When you don't know, ask the experts. During the design, policies definition, pilot implementation, and training of local IT staff among other tasks we will need the help from the experts.

Internal IT Staff: Usually companies do not have any or just small internal help desk team. IT staff will need to be hired and trained to support deployment and provide ongoing support. Usually Companies help desk provide first and second level support.

The Outsource Approach

When looking at Outsource solution, we can understand now why companies are being motivated to outsource PKI systems. The answer can be few hundreds of thousands. Basically all the Secured facilities, Hardware and Software, IT staff for second level support and part of the consulting are taken care of. Third-party companies that provide PKI managed services have the necessary infrastructure, the know-how and the expertise deploying the best-of-the-art solution to other customers.

They even will work with the companies' security and legal people to define the policies and related documents that will dictate how PKI will work.

Verisign[37], a PKI service provider, ensures that the Total Cost of Ownership for the in-House solution split in 5 years is 154US$ per seat per year, and only 60US$ per seat per year when it is outsource. Even if the software is free, in-house solution is far more expensive than in a Managed PKI Solution. [38]

[37] VeriSign, Inc. is a leading provider of digital trust services. Visit www.verisig.com
[38] [CPCPS]

Policies to Implement a PKI solution

As we discussed in previous sections, a PKI implementation is not just hardware, software, digital certificates and pair of keys. But it also includes policies dictating how PKI will be used, the risk management control and business process needed to enable PKI-supported systems and applications.

Keeping in mind the importance of the linkage between the private and public key pair to a subject, meaning that this unique pair is linked to this specific subject and no other, policies must be establish. These policies must define the level of trust that can be placed in a certificate when it is presented to a relying party application. Policies must also define the rules and liabilities of the parties involved in issuing, managing, processing, and supporting certificates. [39]

The role of policy in PKI is critical as it defines the level of risk for relying party applications in a given community of interest. However, PKI policies are in no way mysterious. They in fact are directly related to trust policies already in place in the traditional world.

> *"Use of a PKI to support business processes within a single organization requires no more policy and procedures preparation that that required for any Information Technology (IT) Infrastructure".* [40]

We will include only three documents herein, but it does not mean that they are all. There might be more, regarding the type of PKI implementation, the specific situation, and the intention of each document. Some of these documents are legal bindings that are not included in the general documents (i.e. Certificate Policy, Certificate Practice Statement).

Certificate Policy

The X.509 standard defines a Certificate Policy (CP) as "*a name set of rules that indicates the applicability of a certificate to a particular community and/or class of application with common security requirements*"[41]. In other words, the CP specifies the acceptable uses of digital certificates issued by a CA, and states the contractual liability bounds, and the responsibilities of the CA service provider and its relying parties.

[39] [SDPKI]
[40] [ENT5]
[41] [CPCPS]

A certificate user to help in deciding whether a certificate is sufficiently trustworthy for a particular application may use the CP. Different certificates are issued following different practices and procedures, and may be suitable for different applications and/or propose.

Certificate Policy Statement

A Certification Practice Statement is often confused with a Certificate Policy, but in fact reflects a Certification Authority's statement of practices which should establish conformance with relevant requirements of one or more Certificate Policies, or enable relying parties and subscribers generally to assess the level of trust they may have in the CA and the certificates it issues. Generally it is understood that a CPS contains much greater detail than a Certificate Policy, and may in fact be used to support multiple CPS. In simple terms, one should view the Certificate Policy as the "what I need to do" document, and the Certification Practice Statement as the "how I need to do it" document. [42]

The American Bar Association (ABA) guidelines as "a statement of the practices which a certification authority employs in issuing certificates."

According with the ABA guidelines in 1995 [CPCPS], a CPS may take the form of a declaration by the CA of the details of its trustworthy system and the practices it employs in its operations and support of issuance of a certificate, or it may be a statute of regulation applicable to the CA and covering similar subject matter. Also, whether a CPS is binding on a relying person depends on whether the relying person has knowledge or notice of the CPS.

A CPS is a detailed statement by a CA as to its practices that potentially needs to be understood and consulted by subscribers and certificate users (relying parties). Although the level of detail may vary among CPSs, they will generally be more detailed than certificate policy definitions. Although such detail may be indispensable to adequately disclose, and to make a full assessment of trustworthiness in the absence of accreditation or other recognized quality metrics, a detailed CPS does not form a suitable basis for interoperability between the CAs operated by different organizations. Rather, CP best serve as a vehicle on which to base common interoperability standards and common assurance criteria on an industry wide basis.[43]

PKI Disclosure Statement

PKI Disclosure Statement (PDS) is a document that is being use in some deployments, not with the intention to replace the CP and CPS but with the intention of summarizing what should be important to the subscriber to understand. So, it is kind of small

[42] [CPCPS]
[43] This is the relationship between certificate policy and certificate practice statement

document (not like the CPS that can be more than 100 pages large) that shows to the subscribers what their responsibilities are. This information may be included already in the CPS, but it can be difficult to the user to read all through.

PKI Disclosure Statement (PDS) is an instrument that supplements a CP or CPS by disclosing critical information about the policies and practices of a CA/PKI. A PDS is a vehicle for disclosing and emphasizing information normally covered in detail by associated CP and/or CPS documents. Consequently, a PDS is not intended to replace a CP or CPS.

The following table (Table 4) contains the structure of the proposed PDS according with the PKIX Working Group[44].

STATEMENT TYPES	STATEMENT DESCRIPTIONS
CA contact info	The name, location and relevant contact information for the CA/PKI
Certificate type, validation procedures and usage.	A description of each class/type of certificate issued by the CA corresponding validation procedures, and any restrictions on certificate usage
Reliance limits	The reliance limits, if any.
Obligations of subscribers	The description of, or reference to, the critical subscriber obligations.
Certificate status checking obligations of relying parties	The extent to which relying parties are obligated to check certificate status, and references to further explanation
Limited warranty & disclaimer/Limitation of liability	Summary of the warranty, disclaimers, limitations of liability and any applicable warranty or insurance programs.
Applicable agreements, Certification Practice Statement, Certificate Policy	Identification and references to applicable agreements, CPS, CP and other relevant documents.
Privacy Policy	A description of and reference to the applicable privacy policy.
Refund Policy	A description of and reference to the applicable refund policy
Applicable law and dispute resolution	Statement of the choice of law and dispute resolution mechanisms (anticipated to often include a reference to the ICC's arbitration services)
CA and repository licenses, trust	Summary of any governmental

[44] [PDS]

marks, and audit	licenses, seal programs; and a description of the audit process and if applicable the audit firm.

Table 5: The PDS structure – Source Internet Working Force [45]

Conclusion

During this paper we discussed not only the concepts that PKI involved, but also some other considerations that companies attempting to implement PKI should take into account. One of them was to consider the PKI managed services, which can help companies to fulfill their requirements at lower cost, but still with some security concerns (Letting a third-party company get into your assets and manage them).

Also we discussed the importance of the security policies that surround a PKI system. CP and CPS are key components in establishing the degree of assurance or trust that can be placed in certificates issued by CAs. The planning, development and maintenance of a CP and CPS should be conducted in close cooperation with other organizational units to ensure consistency through the organization.

We also can conclude that implementing PKI is not an easy task, and need a lot of planning, resources and a good budget. But also companies implementing PKI need to ensure that they understand their requirements and how PKI will help them to achieve them.

Some questions remain unanswered, such as who should be involved in a PKI deployment? What are their roles and responsibilities? What are the management protocols between the systems? How I inter-connect two different domains of trust? How long a real implementation would take? How much is going to cost? And who are the main players in the PKI market?

References

[PKC] William Stewart, "PKC History", Public Key Cryptography, February 2003, http://livinginternet.com/?i/is_crypt_pkc_inv.htm, (February 2003)

[ENT1] Entrust Coursewere, Security Technologies for E-Business.

[BGR] Laura Biasci, Lyne Granum, and Frank Rundatz , "Data Encryption Standard" searchSecurity.com, Jan 19, 2001, http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci213893,00.html, (February 2003)

[45] [PDS]

[DES] Kenneth Castelino, "3DES and Encryption", http://kingkong.me.berkeley.edu/~kenneth/courses/sims250/des.html, (February 2003).

[CAST128] C. Adams, Network Working Group, "The CAST-128 Encryption Algorithm", May 1997 http://www.faqs.org/rfcs/rfc2144.html (February 2003)

[CAST256] C. Adams, Network Working Group, "The CAST-256 Encryption Algorithm", http://www.faqs.org/rfcs/rfc2612.html (February 2003).

[RC4] Ronald Rivest, "RC4 Encryption Algorithm", http://www.ncat.edu/~grogans/algorithm_history_and_descriptio.htm, (February 2003)

[RC2] R. Rivest , "A Description of the RC2(r) Encryption Algorithm", March 1998, http://www.ietf.org/rfc/rfc2268.txt (February 2003)

[IDEA] "International Data Encryption Algorithm (IDEA)", Computer Security, Section 4, http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_43.html (February 2003)

[AES] Joan Daemen, Vincent Rijmen, "AES Proposal: Rijndael", 03/09/99, version 2, http://csrc.nist.gov/CryptoToolkit/aes/rijndael/Rijndael.pdf (February 2003)

[BLOW] Bruce Schneier, "The Blowfish Encryption Algorithm", November 1996, http://www.informatik.uni-mannheim.de/informatik/pi4/projects/Crypto/rgp/blowfish/blowfish.html (February 2003)

[SANS] Northcutt, Zeltser, Winters, Kent, Ritchey, "Inside Network Perimeter Security", Sans GIAC, 2003, New Riders Publishing, First Edition, Indianapolis, Indiana, US, page 646.

[RSA] RSA, "PKCS #1 - RSA Cryptography Standard", June 2002, http://www.rsasecurity.com/rsalabs/pkcs/pkcs-1/index.html (February 2003)

[DSA] Tony Panero, "DSA", Marsh 1998, http://home.pacbell.net/tpanero/crypto/dsa.html (February 2003)

[ECDSA] Don Johnson, Alfred Menezes, and Scott Vanstone, "The Elliptic Curve Digital Signature Algorithm (ECDSA)", certicom, http://www.certicom.com/pdfs/whitepapers/ecdsa.pdf (February 2003)

[PKIB] Shashi Kiran, Patricia Loreau, Steve Lloyd, "PKI Basics – A Technical Perspective, November 2002. http://www.pkiforum.org/pdfs/PKI_Basics-A_technical_perspective.pdf (February 2003)

[ENT2] Entrust Industries, "The Concept of Trust in Network Security", August 2000, http://www.entrust.com/resources/pdf/trust.pdf Version 1.2. (February 2003)

[IETF] IETF, Internet X.509 Public Key Infrastructure, http://www.ietf.org/rfc/rfc3280.txt

[BDR] Barbara Depompa Reimiers, "PKI's Are Still Tough To Deploy", April 9, 2001. www.internetweek.com/security/secure040901-1.htm

[DIC] Michael Dickerson, "Implementing PKI", January 2001, http://www.scmagazine.com/scmagazine/2001_01/survey/survey.html (February 2003)

[VERC] Verisign Videconference, "Managed PKI Services vs. In-house PKI Investigative Study - The Real Total Cost of Ownership (TCO)", http://us1.webex.com/verisignevents/onstage/framesets/viewrecording1.php?EventID=213953993

[SDPKI] Sabo & Dzambasow, "PKI Policy White Paper" PKI Forum, March 2001, http://www.pkiforum.org/pdfs/pki_policy.pdf (February 2003)

[ENT5] Sharon Boeyen, "Certificate Policies and Certification Practice Statements", February 1997, http://www.entrust.com/resources/pdf/cps.pdf (February 2003)

[CPCPS] S. Chokhani, "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework", RFC 2527, March 1999. www.ietf.org/rfc/rfc2527.txt?number=2527. (February 2003)

[PDS] S. Santesson, M. Baum, "Internet X.509 Public Key Infrastructure PKI Disclosure Statement", May 2000, http://www.verisign.com/repository/pds.txt. (February 2003)

[RAIN] Rainbow Technologies, "Public Key Infrastructure – Securing the Future of Communications", Rev 1.1 10/27/00, http://www.rainbow.com/partners/documents/PKI_PAPER.pdf, (February 2003)