A Survey of State of the Art in Public Key Infrastructure

NR Rapport nr. 995

Shahrzade Mazaher
Per Røe

August 2003

Tittel/Title:
A survey of state of the art in Public Key Infrastructure

Dato/Date: August
År/Year: 2003
ISBN: **82-539-0502-5**
Publikasjonsnr.:
Publication no.: 995

Forfatter/Author:
Shahrzade Mazaher and Per Røe

Sammendrag/Abstract:
In this report, an overview of the state of the art in Public Key Infrastructure (PKI) is given. Public key cryptography is briefly explained and the infrastructure needed to support it and issues that hinder its widespread usage are discussed. An alternative approach to the problem domain, the Simple Public Key Infrastructure (SPKI), is also shortly described. Some existing implementations are also described and the report concludes with a list of issues that are the major challenges to its broad acceptance.

Emneord/Keywords: Public Key Infrastructure, PKI, Public Key Cryptography, SPKI, X509

Tilgjengelighet/Availability: public

Prosjektnr./Project no.: 995

Satsningsfelt/Research field: Public Key Infrastructure (PKI)

Antall sider/No. of pages: 33

Table Of Content

Introduction

This report is concerned with Public Key Infrastructure (PKI), i.e., the infrastructure needed to support public key cryptography. It will survey the major efforts in the field and point out the major open issues that need to be solved. The remainder of the report is organized as follows. A brief history of public key cryptography is given later in this section. Section 2 shortly explains the principles of public key cryptography and its use. The traditional approaches to the main components that comprise the PKI are described in Section 3 and the issue of privilege management, i.e., management of permissions and credentials, is discussed. Section 4 deals with the issues of interoperability between the various PKI realizations, which differ both in terms of models, architecture, practice, etc. A new approach to PKI is presented in Section 5. Section 6 gives a brief overview of the existing PKI realizations, and Section 7 points to new research directions in PKI.

The concept of Public Key Infrastructure (PKI) dates back to the work done by Diffie and Hellman in cryptography. In their 1976 paper, "New directions in Cryptography" [19], they introduced public key cryptography and claimed that the key management problem was solved. This was done by means of a modified telephone directory, which they called Public File. Instead of entries, each with name, address and phone number, the Public File would contain entries with name, number and public key. To send a confidential message, one would find the recipient's public key by looking him up (by his name) in the Public File, and then encrypt the message with that public key before sending it to the recipient. Only the recipient, presumably, holding the corresponding private key can decrypt the message. As a result of the properties of public key cryptography, the public key need not be kept secret. The difficult problem of key management was solved but an equally difficult problem was introduced, namely, the problem of naming and name management.

In 1978, Kohnfelder, in his bachelor thesis at MIT [39], took up the problem of Public File and its performance in a network setting. To solve this, he proposed to take each entry of the Public File, namely the name and the public key, and digitally sign them. He coined the term **certificate** for this digitally signed version of the entries in the Public File. These certificates could then be distributed to anyone who wanted them.

In the 1980's, ITU (International Telecommunication Union) started an effort about building a directory like the one proposed by Diffie and Hellman. The directory was to cover all the people and devices in the world, and gather all information in one place. The result was a standard, known as X.500 [37], defining all characteristics of such a directory. For authentication purposes, e.g., for granting permission to somebody to change an entry in the directory, a companion standard, X.509, defining a certificate format was produced. A X.509 certificate binds a public key to a **Distinguished Name** (DN), which can be thought of as a pathname into the X.500 directory, and was supposed to be globally unique. For the signing of certificates, the notion of Certification Authority (CA) was introduced. This was supposed to be some trustworthy authority with a public key of his own, publicly available, who would then digitally sign the X.509 certificates.

The Privacy Enhanced Mail (PEM) [38] of ITEF made use of X.509 certificates for the identification of mail recipients. This effort was carried out around 1990. PEM failed, mainly, because of lack of infrastructure; there were no Certification Authorities (CAs) in place to issue X.509 certificates. To provide for certification without CAs, Pretty Good Privacy (PGP) [53] proposed another scheme. PGP allowed any keyholder to sign the key of any other keyholder, i.e., to issue certificates, thus forming a *web of trust*. The assumption was that multiple independent signatures on a certificate would be as trustworthy as the single signature of a CA on the same certificate.

Despite these various initiatives, still, PKI was not commonplace. In late 1990's, three independent initiatives (SDSI[49], SPKI [25], PolicyMaker[12]) started out, all based on the assumption that the PKI

model itself was the problem. What they had in common was the way in which they differed from the traditional PKI model; they used the public key itself as the identifier of the keyholder. Of these more recent efforts, SPKI is still very active and gaining ground.

Lately, in an effort to make PKI more useful, some initiatives such as United State's Federal Bridge [29], have tried to tackle the interoperability problems of the traditional PKI. But, a lot of open issues still need to be addressed.

Public Key Cryptography

Public Key Cryptography has aroused a lot of interest ever since it was introduced in the 1970s. The reason why an infrastructure using public key cryptography is so interesting is that a good PKI makes authentication, encryption and digital signatures possible with the use of non-secret information, namely, the public key.

Traditionally, cryptography has been performed by having the two communicating parties agree on a shared secret before they can start communicating. This secret is usually called a key and is used for both encryption and decryption of messages. Therefore, this kind of cryptography is usually referred to as symmetric cryptography. Since the essence of symmetric cryptography is that only the communicating parties should know the secret key, there is a need for an alternative secure channel for the establishment of the shared secret key between the communicating parties.

Until the 1970's symmetric cryptography was the only form of cryptography available. But, in the mid-1970s asymmetric cryptography was introduced. The major difference between symmetric and asymmetric cryptography is that in asymmetric cryptography there are two keys, called the private and the public key. Each pair of private-public key has the property that messages encrypted with the public key can only be decrypted with the private key, and for all practical purposes, it is impossible to compute the private key from the public key. It is therefore enough that the owner of a private-public key pair keeps its private key secret, while the public key can be made publically available. In this way, all that the communicating parties need to know in order to communicate securely with each other is the other party's public key, which is available to any entity that needs it, and therefore, there is no need for a secure communication channel for exchanging these keys.

Use of Public Key Cryptography

The properties of public key cryptography make it possible to use it both for encryption of messages, signing of messages, and for authentication.

Encryption

Public key cryptography can be used to send encrypted messages in a simple way. In this scheme everybody owns a private-public key pair. To send an encrypted message, all the sender of the message needs is a copy of the receiver's public key, which is publically known. The sender then uses this public key to encrypt the message; this message can be decrypted only by using the corresponding private key which is held by the receiver of the message only. Since the private key, i.e., the secret information, must be known only to its owner, key distribution is easier than with symmetric cryptography.

Authentication

Authentication is also simple with public key cryptography. To authenticate itself to a party, be it a person or a service, the owner of a private-public key pair has to prove that it has access to the private key. To accomplish this, a party that wants to authenticate itself, i.e., the authenticating party, sends to the relying party a random message encrypted with its own private key. The relying party then uses the authenticating party's public key to decrypt the message. In this way, the relying party can be sure that the authenticating party has access to the private key.

Digital Signature

One of the usages of public key cryptography is the digital signing of messages. The digital signature process involves a one-way hash function, i.e., a function with the properties that it is mathematically very

difficult to compute its inverse and that it produces a fixed-length value based on the contents of a message. When digitally signing a message, the hash of the message is generated and then encrypted using the sender's private key. The signature can then be verified using the sender's public key, which is publically available. The receiver then knows that the message is signed using the sender's private key, and that the message has not been altered after the sender has signed it.

An important issue in the use of public key cryptography is trust. The user of a public key must be sure of the identity of the owner of the public key. This is the problem that public key infrastructures try to solve. The usual way of doing this is that the public keys are distributed as a part of a certificate. The certificate contains the public key, but also information that identifies the entity that the public key and the corresponding private key belong to. The information in the certificate is verified and signed by a trusted party, usually called the certification authority (CA).

There are many issues in making such an infrastructure trustable and practically feasible to deploy. These issues and the proposed solutions are the subject of the rest of this report.

Traditional PKI

This section discusses the traditional approach to Public Key Infrastructure (PKI) and the different components that comprise a general purpose PKI. It points out issues of both technical and organizational nature that still have to be tackled in order to achieve a seamless PKI. It also briefly describes the related field of Privilege Management Infrastructure (PMI).

PKI and its Components

A general purpose PKI is very complex and involves issues of different natures. All of them arise from the simple fact that in order to use a public key, one should have an assurance about the authenticity of the key, i.e., a guarantee that the public key in fact belongs to the entity that claims to own it. This guarantee of authenticity is achieved by means of a certificate, i.e., a digitally signed document binding the identity of the keyholder to its public key. Digital certificates stand therefore at the heart of PKI.

The introduction of certificates leads to the question of who is liable for the guarantee of authenticity of a public key? The answer is a Certification Authority (CA). Although conceptually simple, the certificate-CA pair involves quite a few technical and organizational challenges that necessitate the use of an appropriate infrastructure, i.e., a PKI. The main components of PKI are detailed in the remainder of this section.

Digital certificates

As mentioned above, a digital certificate associates an identity with the private-public key pair of the owner of the identity; therefore, the identity must be unique.

The widely used certificate formats are all based on X.509v3 [36]. The basic information contained in a certificate is:

- Subject: the individual or entity being identified by the certificate.
- Public key: the public key of the subject, corresponding to its private key.
- Issuer: the trusted authority that has generated and signed the certificate.
- Serial number: a unique identifier for the certificate.
- Validity period: a date indicating the earliest time the certificate can be used and a date indicating the expiration of the certificate.
- Usage: the description of the usage for which the corresponding private-public key pair is valid.
- Digital signature: the digital signature of the issuer.

Although all based on the same basic format, certificates from different CAs are according to different profiles, use different extensions and ascribe different semantics to the attributes in a certificate.

This situation creates problems with respect to naming. The lack of standards for naming, apart from the defined fields and attributes for encoding names, makes the task of processing names very hard. Actually, the directory and the naming issues are considered by many experts in the field as the major issues in PKI, in general, and as an obstacle to interoperability in particular [14][24].

Certification authorities

The main task of a CA is to issue certificates. In order to do so, a CA needs a private-public key pair for the applicant and means to properly identify the applicant.

The process begins with the user providing the CA with sufficient information about his identity. After a satisfactory verification of the supplied identity credentials, the CA generates a public-private key pair for the user[1], and creates a certificate for the generated public key; the private key must be transferred in a secure way to the applicant who must store it in a secure storage for later use. Currently, most of this process is software supported.

Many systems require the ability to recover a lost key in order to access information previously encrypted by that key. In such cases, the users' private keys are backed up by the CA or a separate key recovery system.

Generally, it is assumed that a CA operates out of a vault where his (private) signing key is very strongly protected. The cost of such a facility is very high and there cannot be very many of them. A scenario in which the users, i.e., those applying for certificates, should present themselves to the CA, with proof of their identities, would be too expensive for the users who might then have to travel a long way. Moreover, the larger the number of certificate-users that a CA must manage, the harder it becomes to verify their identities. In other words, the verification of an identity is more trustworthy when the CA is close to the actual user. To remedy this situation, a *Registration Authority* (RA) was introduced. That is, there are many RAs for each CA such that the users can find one close at hand. The task of the RAs is to verify the identity credentials that the users present and, if approved, start the certification process with a CA.

As mentioned earlier, certificates expire; CAs have therefore the renewal of certificates as part of their task. The process of renewing a certificate is much simpler. The user presents the certificate it has and by proving access to the corresponding private key, his identity can be verified. The certificate can then be renewed immediately if there is no change in the identification information. Another task of the CAs is to revoke certificates when necessary, make the fact known to all possible users of the certificate, and manage the revocation status for all the certificates they have issued. This is discussed in more details in a later section.

Deciding on the length of the key

Cryptography is based on mathematical problems that are extremely hard to solve. Given the proper amount of time and resources, attackers can break any cryptographic key. In general, the longer the key, the harder it is to break; therefore, the choice of the length of he key has very much to do with how long the information to which it is applied should be protected. If the information needs to be protected, e.g., for five years, then the length should be chosen such that, given the current technology, it would take more than five years to break it.

Trust models and certification paths

Users must trust the CA, which includes the ability to verify the CA's signature on the certificate. That requires safe knowledge of the authenticity of the CA's public key.

Ideally, a single CA, trusted by all users, would issue all the certificates. In this case, the users could get the CA's public key in a safe way from the CA and use it for certificate validation. However, in the real world, the model with a single CA is not achievable for both organizational and technical reasons, e.g., different countries having different laws, different needs in different application domains, or technical problems for a single CA to manage a large global population. It is therefore best to have many CAs each being responsible for a subset of the user population.

[1] In principle, the user could himself generate a pair of public-private keys and provide the public key to the CA along with the identity credentials.

The model with multiple CAs while solving the problems involved in the single CA model, introduces problems of its own. That is, users will have to trust many different CAs, and must be able to verify signatures from all of them. The burden of managing multiple CAs is not acceptable for most users; the remedy to this situation is to build trust relationships between different CAs.

Trust relationships between CAs means that CAs not only issue certificates to users and other entities such as application servers and network routers, but, also to other CAs, i.e., a CA will certify the identity of other CAs. In all trust models, the certificate user or *relying party* must have an initial trust in some entity of the model. This initially trusted entity is called the *trust anchor* for that relying party. Each relying party has the public key of his trust anchor.

To validate a received certificate which is not issued by the relying party's trust anchor, the relying party should follow a set of trust relationships from the CA that issued the received certificate to the CA that is his trust anchor. This results in following a chain of certificates, corresponding to the CAs in the set of trust relationships, also called certification path. The length of this path is critical in the performance of a deployed PKI.

Hierarchical model

The most widely used trust model is a strict hierarchy, where the subordinate CAs are certified by the parent CA, but not vice versa. In this model, the root CA of the hierarchy is trusted by all relying parties, i.e., it is the sole trust anchor in the model. It has a self-signed certificate and all relying parties have a copy of its public key. This model has several benefits:

- All certificate paths terminate with the root CA certificate; the length of the certification path depends therefore only on the depth of the hierarchy[2].
- There is only one certification path for each end-entity. This makes it possible to provide a path to the root CA to the relying party by having the end-entities include the certificates of all the CAs on the path along with their own certificates.

The main drawback of this model is that it is not possible for the whole population of certificate users to agree on a single root CA, which will be the common trust anchor.

This model fits best smaller organizations with a hierarchical structure; it has been used in Privacy Enhanced Mail (PEM) and more recently, by the U.S. Department of Defense [33].

Peer-to-peer model

In a peer-to-peer trust model, there is no hierarchy and thereby no root CA and no single trust anchor; any CA can establish a trust relationship to any other peer CA and thus issue a certificate for that CA, i.e., cross-certify the other CA. In such a model, users trust local CAs.

In a fully connected mesh of cross-certifications, where each CA cross-certifies all the other CAs, the certification path is very short, but the proportion of the number of needed cross-certificates to the number of CAs is of the order of n^2. This model is not a very useful one and presents problems with respect to certificate distribution.

A more useful cross-certification model is one that allows longer certification paths and a partially connected mesh. Cross-certification is most useful where subordination cannot be applied and between different trust domains, e.g., different organizations.

[2] In a more general hierarchical model, where the parent CA and each child CA certify each other mutually and the trust anchor of each end-entity is the CA that issued the end-entity's certificate, the length of the certification path depends on the distance of the sender and receiver of a certificate in the hierarchy, and finding the certification path is not as straightforward anymore.

This model also has some drawbacks. One is that since certification paths are longer and traverse several domains, the same level of trust cannot be guaranteed implying decrease in trust in certificates, e.g., some trust domains have a more strict identification process than others. As there might exist several paths between two end-entities, another drawback is the complexity involved in the selection of an optimal certification path. A third drawback is that as new trust relationships are made and new CAs are added to the mesh by existing CAs, it becomes harder to see whether one's trust relationship has been extended to some not fully trusted entity. This brings about the issue of authorization. That is, in addition to PKI establishing identities, there is a need for an authorization system that lets one assign different access rights to one's system to different certificate holders. Authorization is handled by Privilege Management Infrastructure (PMI), discussed later.

An attempt to deploy a cross-certification model is the US *federal bridge* initiative [29]. This is a project where different government agencies can authenticate each others using the bridge CA. They operate with several levels of assurance, and the approach is standard-driven.

Hybrid trust models

A more flexible trust model can be obtained by mixing hierarchical and cross-certification models. For example, enterprises with hierarchical structures can deploy a hierarchical trust model internally, while conducting inter-enterprise business by deploying a cross-certification model. Another example of use of a hybrid model is in dealing with very large hierarchies; some often-used, long certification paths can be optimized by establishing a direct cross-certification link between the two leaf-CAs involved in the path.

Certificate revocation and validation

A certificate is valid only within a period of time indicated in the certificate. But, it can be revoked, i.e., declared invalid by its issuer, before its expiration time. There are different reasons for revoking a certificate, e.g., a key being compromised as a result of a security attack on a system or the owner of the certificate leaving the company that had issued the certificate to him. It is therefore crucial to verify the validity of a received certificate with respect to both its validity period and its revocation status.

Certificate validation is the process that determines whether a certificate can be accepted as being valid. It involves checking different aspects of the certificate:

- The digital signature on the certificate must be verified both to ensure that the certificate has not been tampered with and that the signature is that of the authority that issued it.
- The time at which the certificate is being checked must be within the validity period of the certificate, which is usually one to two years.
- The revocation status of the certificate must be checked to ensure that it is not revoked.
- Syntax and semantics of the certificate must be checked to ensure that the format is right, all the mandatory fields and critical extensions are present. It is important that all critical fields are well understood.
- It must be checked that the use of the certificate use is according to the purpose for which it was created.

As for the revocation process, when notified of some ground requiring the revocation of a certificate, a CA must take action to revoke the certificate and must advise all potential users of the certificate of the fact.

The most widely used mechanism for revocation is based on Certificate Revocation Lists (CRL) as described in [36]. A CA must publish periodically a CRL containing the certificates it has revoked. Each CRL entry consists of the serial number of the revoked certificate along with the revocation date and reason. To ensure the integrity of CRL, it is signed by the CA (or some trusted revocation service). In

order to allow for the use of the freshest release, the CRL contains the date it has been published and a date for the next release.

To check for the revocation status of a certificate, the recipient of a certificate downloads the CRL, checks whether the CRL is up-to-date (not an old copy), checks the CA's signature on the CRL, and lastly, checks whether the certificate's serial number is on the list.

Provided that each received certificate should be validity-checked, this mechanism will easily bring the server providing the CRL service to its knees, creating a denial of service situation. The Online Certificate Status Protocol (OCSP) [46] was designed to ease this situation. OCSP provides more timely information about the revocation status of a certficate and is supposed to be faster than the CRL mechanism. An OCSP service is provided either directly by a CAs or by an authorized responder. For each status request, the service checks the status of the corresponding certificate directly with the CA, and can therefore provide a more timely status than that provided by periodical CRLs. Note that the status returned by the OCSP respomder pertains only to the revocation of the certificate and indicates nothing about the validity status of the certificate, i.e., whether the certificate is still within its validity period.

Certificate policy and certificate practice statement

A CA operates based on a Certificate Policy (CP) and/or Certification Practice Statement (CPS) covering legal and technical aspects of the certificates and the process of issuing them.

X.509 [36] defines a Certificate Policy as:

> "A named set of rules that indicates the applicability of a certificate to a particular community and/or class of application with common security requirements. For example, a particular certificate policy might indicate applicability of a type of certificate to the authentication of electronic data interchange transactions for the trading of goods within a given price range."

This definition is also endorsed by IETF's RFC 2527 [16]. moreover, a CP has a unique identifier associated with it called an Object Identifier (OID), which can be used to refer to a CP, e.g., in a certificate to indicate the policy that applies to it.

The Certification Practice Statement (CPS) is defined by the American Bar Association [4] as follows:

> "A certification practice statement may take the form of a declaration by the certification authority of the details of its trustworthy system and the practices it employs in its operations and in support of issuance of a certificate, or it may be a statute or regulation applicable to the certification authority and covering similar subject matter. It may also be part of the contract between the certification authority and the subscriber. A certification practice statement may also be comprised of multiple documents, a combination of public law, private contract, and/or declarations."

RFC 2527 states that, in summary, the main differences between a CP and a CPS are:

> " (a) Most organizations that operate public or inter-organizational certification authorities will document their own practices in CPSs or similar statements. The CPS is one of the organization's means of protecting itself and positioning its business relationships with subscribers and other entities.
>
> (b) There is strong incentive, on the other hand, for a certificate policy to apply more broadly than to just a single organization. If a particular certificate policy is widely recognized and imitated, it has great potential as the basis of automated certificate acceptance in many systems, including unmanned systems and systems that are manned by people not independently empowered to determine the acceptability of different presented certificates."

There is no general agreement on the role that each of these documents has. In addition, a proposal for a model PKI Disclosure Statement (PDS) [5] has been submitted to IETF, as an Internet Draft in November 1999 ([11]). The PDS defines a model for presenting subsets of information contained in CP and/or CPS, in a more concise and user-friendly way. It is not intended to replace CP and/or CPS, but, rather as a more convenient way to convey the appropriate information. It has even been suggested that a PDS is, in fact, a CP in accordance with the definition in X.509.

Certificate classes

Usually, the receiver of a certificate would like to be able to judge the degree of trust it can put in the certificate, and thereby, in the holder of the certificate. One of the major factors in the degree of trust is the CA's procedure for verifying the identity of a user; another factor is the liability that the CA is willing to assume in case of errors. The recipient of a certificate, the *relying party*, would therefore need some metrics to judge the *quality* of, i.e., the degree of trust that can be put in, a certificate.

The type of information needed to determine the quality of a certificate is described in the CP and/or CPS of a CA. A CP and/or CPS can therefore be used as the basis for determining the certificate's quality. The task of determining the quality of a CP must be done off-line by experts. It is tedious, and it involves reading the document and understanding its technical and legal consequences, and then rating the policy. There is therefore a need for a standard scheme for the classification of the quality of a certificate. This classification should then be made part of the information contained in a certificate. It must also be backed by evidence, from a neutral instance, that the operation of the CA is according to its policy.

There have already been some initiatives in this direction. One class of quality, *qualified certificate*, is defined by the EU directive on electronic signatures. The US *federal bridge* initiative [29] has also defined some classes of quality for certificates, and a CA cross-certifies with the federal bridge at the level that applies to its CP.

Privilege Management Infrastructure

Privilege Management Infrastructure (PMI) was briefly mentioned earlier when trust models were discussed. It manages all aspects of users' rights and privileges by issuing, distributing, revoking, and storing Attribute Certificates (ACs).

Why do we Need an Attribute Certificate

Having identified and built some trust in an entity holding a valid public key certificate, hereafter called PKC, does not mean that the entity should be authorized to have full access to the relying party's system.

Many systems therefore use the identity certificate to perform identity based access control and/or to carry privilege related information in an extension field called *subjectDirectoryAttributes*. But, privileges change frequently and have short lifetimes while a PKC has, relatively, a much longer validity period. Therefore, if a PKC is used for privilege information, changes in the privileges of a certificate holder would result in the revocation of the current valid certificate and the issuance of a new certificate with the new access rights. Furthermore, privileges involve the issue of delegation, i.e., the ability to transfer one's privileges, or a subset of them, to some other certificate holder; there is no adequate support for delegation in a PKC.

A PKC is therefore not suitable for privilege related information, and this is why the idea of having attribute certificates was advanced. The owner of a PKC can thus have many ACs and his rights can change, by issuing new ACs to him and revoking some existing ones, without affecting his PKC.

A separate AC allows also for role-based and rule-based access control, which require information not usually available in PKCs, e.g., an AC can be issued for a role or assign a role to an entity.

PMI Components

The basis of PMI being an attribute certificate, as for the public key certificates, there is a need for an authority to issue the ACs. A CA being the authority that issues certificates binding identities to public keys, does not necessarily mean that the CA should also be the authority for authorization certificates. The authority for authorization is the *Attribute Authority* (AA); note that the two types of authority may be combined in a single entity. The other components of PMI, introduced in X.509, are a *Source of Authority* (SOA) and Attribute Certificate Revocation Lists (ACRLs).

Attribute Certificates

The first AC format was published in ANSI X.9. Version 2 of that standard introduced an extension mechanism and made the owner field to point to either an identity or a specific PKC. ITU-T included it in its X.509 standards, in 1997. An alternative standard, ECMA-219 [21], has been developed by the EuropeanComputer Manufacturers Association (ECMA). This standard describes a model for distributed authentication and access control, in which a trusted third party is responsible for authenticating the entities and providing them priveleges they need for access control, and the corresponding certificate format. SESAME is a non-commercial product implementing the security components and the certificate format of ECMA-219.

The AC is generic enough to allow any attribute to be conveyed. Without limiting the attributes and the extensions that can be included in an AC, it would not be possible to develop interoperable implementations; ANSI, ITU-T and IETF have therefore developed standard attributes and extensions for use in v2 ACs.

ACs are rather short-lived and can be linked to PKCs. The subject of a PKC can thus own many ACs for different privileges in different contexts.

The basic information in ACs is very close to that in public key certificates. The main differences are that an AC does not contain the subject's unique identifier, but instead, it contains attributes and a pointer back to some public key certificate (via its serial number) whose subject is then associated with the privileges defined in the AC.

Attribute authorities

The responsibility of an AA is to delegate privileges to end-entities or other AAs. Of course, in order to do so, it must hold the privileges himself. A privilege that an AA has, has either been delegated to it or it is the source of the privilege itself.

When an AA delegates a privilege it can restrict its further delegation, e.g., not allowing further delegation at all or restricting it to a subset of the possible recipients such as other AAs. AAs may also issue role certificates that associate privileges with a role and certificates to assign a role to an entity.

Though, usually not necessary for short-lived ACs, AAs may have to revoke long-lived ACs. AAs must therefore indicate whether the ACs they issue can be revoked and if so, the place where the revocation information can be obtained. The tasks involved in supporting revocation are issuing revocation notices and publishing Attribute Certificate Revocation List (ACRL). ACRLs are stored in a directory and a separate server can handle revocation status requests.

Source of authority

Basically, a source of authority is to AAs what a root CA is to CAs. That is, it is a trust anchor and the source of all privileges. SOAs are the start of all attribute certificates delegation chains.

Attribute certificate revocation lists

Attribute certificate revocation lists are supported in the same way as the CRLs and the same format, as for CRLs, is used for them.

PKIX

The "Internet X.509 Public Key Infrastructure" working group (PKIX) was established in 1995 to develop Internet standards necessary to support PKIs. The ITU-T Recommendation X.509 is widely accepted as a basis for PKI, and defines certificate formats, their fields, attributes and possible extensions, and some related procedures. The first work item of PKIX was a profile for the X.509 standard.

The X.509 intension is to make the X.509 certificates usable in many different application contexts. As a result, many of the certificate fields and extensions allow for different options, and thus, making it hard for different implementations to interoperate. The profile defined by PKIX restricts the various options to a set useful for Internet.

Other tasks undertaken by PKIX include a profile for X.509 v2 CRL standard, development of operational and management protocols for the PKI-related information, such as the Certificate Management Protocol (CMP)[1] for initializing, certifying, updating and revoking PKI entities, work in the area of certificate policies and certificate practice statement, and time stamping and data certification services, used to build services such as non-repudiation.

It also defines an architectural model for Privilege Management Infrastructure (PMI) and give some advices regarding the implementation of some the PKI components with an emphasis on names and their related topics.

This section briefly overviews the general PKI, discussed in the previous sections, in the context of PKIX.

Public Key Infrastructure

PKIX defines a PKI as:

> "The set of hardware, software, people, policies and procedures needed to create, manage, store, distribute, and revoke PKCs[3] based on public key cryptography."

Furthermore, a PKI is said to comprise five types of components:

- Certification Authorities (CAs) that issue and revoke certificates ;
- Organizational Registration Authorities (ORAs) that vouch for the binding between public keys and the identity of certificate holders (and possibly other attributes);
- End-entities to whom certificates are issued, i.e., certificate holders;
- End-entities that validate received digital signatures, i.e., the relying parties;
- Repositories that store and make available certificates and the corresponding revocation lists.

PKIX also defines an architecture, i.e., the assumed relationships between the identified components, depicted in Figure.1.

[3] Public Key Certificate.

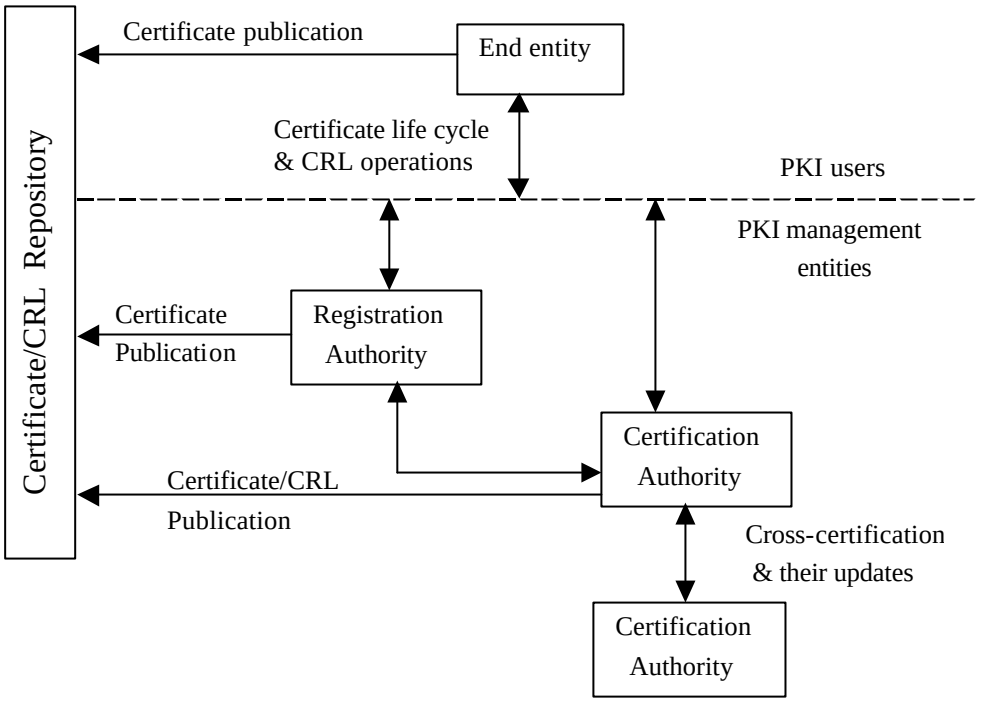

Figure.1. PKIX Architecture

PKIX Functions

General functions of PKIs have been discussed earlier in this section; in what follows a brief overview of them in the context of PKIX is given.

Registration

This is the process whereby an end-entity, the subject of the PKC to be issued, identifies itself to the CA, either directly or via a RA, by providing its name and other needed attributes for the certificate. This information must be validated by the CA (or RA) in accordance with its Certification Practice Statement.

Initialization

The initial values needed to begin communication with a PKI are obtained during the initialization. An end-entity obtaining the certificate or the public key of a CA and/or generating public-private key pair for an end-entity are activities that are covered by initialization.

Certification

The CA issues a PKC for the subject's public key, and returns that certificate to the subject or posts it in a repository.

Key pair recovery

Local policies may require backing up encryption keys or keys used for other purposes such as key exchange. This is in order to be able to recover the key, if lost, and access information previously encrypted by the key if needed. Archiving of the private key may be done by the CA or a separate key recovery system.

Key generation

PKIX allows for an end-entity to generate its own public-private key pair in its local environment and present them to the CA (or RA) along with the other information when registering. Alternatively, the CA (or RA) can generate the key pair that must then be conveyed to the end-entity in a secure way, e.g., in an encrypted file, on a smart card, etc.

Key update

Key pairs must be replaced on a regular basis; this is because either the key has expired or it has been compromised.

In the normal case of key expiration, the transition from the old PKC and its key to a new PKC with a new key must be graceful, especially in the case of updating the key of a CA. This requires suitable mechanisms to support notifications and the switch of the PKCs.

In the case of a key compromise, the transition will not be graceful because of the unplanned nature of the switch. The PKI must support notification of both the invalidity of the previous PKC and the validity and availability of the new PKC. Compromise of the CA's key is a catastrophic event that entails the revocation of the CA's certificate along with all the certificates issued by the compromised CA and its subordinates. Furthermore, there is a need to support out-of-band notification in order to let the users know of the compromise and the subsequent update.

Cross-certification

In PKIX, a cross-certificate is a certificate issued by one CA to another CA. Cross-certification can be accomplished in only one direction or in both directions. Cross-certificates are usually issued across administrative domains, but, they can also be issued within the same administrative domain as well.

Revocation

A PKC, when issued, is expected to be used within its entire validity period; but various circumstances can cause it to become invalid before its expiration time, and thereby revoked.

PKIX proposes two alternative ways to support certificate revocation: the method described in X.509 based on periodically published CRLs, and on-line methods of revocation notification. For the former method, the X.509 v2 CRL format, profiled by PKIX, is used for information about the revocation status of a certificate.

PKIX defines a few protocols that support the on-line checking of certificate status; the most widely used one is the *Online Certificate Status Protocol* (OCSP)[46]. On-line revocation checking reduces the latency between the revocation of a certificate by a CA and the corresponding notification to the end-entities. However, this method imposes new security requirement, i.e., the end-entity must trust the on-line validation service[4] whereas the CRL repository need not be trusted.

Certificate and revocation notice distribution and publication

The PKI is responsible for the distribution of PKCs and the revocation notices. Certificates are distributed either by transferring them to their owners and/or by publishing them in a repository. Revocation notices may be distributed by transmitting them to end-entities, publishing them in a repository and/or forwarding them to an on-line responder.

Privilege Management Infrastructure

PKIX defines a Privilege Management Infrastructure as

[4] The end-entity must validate the signature of the OCSP-server on the received responses.

"The set of hardware, software people, policies and procedures needed to create, manage, store, distribute, and revoke ACs[5]."

Furthermore, a PMI is said to comprise five types of components:

- Attribute Authorities (AAs) that issue and revoke Attribute Certificates (CAs);
- Attribute Certificate Users, i.e., end-entities that process ACs;
- Attribute Certificate Verifiers that check the validity of an AC;
- End-entities requesting actions that need authorization checks to be performed;
- Repositories that store and make available certificates and Certificate Revocation Lists.

As mentioned earlier, the AC is very generic and allows any attribute, which is a hinder in the way of interoperable implementations. PKIX has set out to specify an AC profile for Internet, e-mail, IPSec applications, etc. This profile will constrain many of the options allowed by X.509, e.g., AC chains, like PKC chains, are allowed by X.509, but in order to simplify the implementation, the AC profile recommends that they not be supported.

Operational and Management Protocols

Operational protocols are the transport protocols used to carry certificates, CRLs, and all related management information between the different components of the PKIX architecture. PKIX supports LDAP[13], HTTP[35] , FTP[35] and X.500.

Management protocols support exchange of management requests and information between end-entities and the management entities, of the PKIX architecture, and between the management entities.

Certificate Management Protocol (CMP)[1] and Certificate Management Messages over CMS[6] (CMC)[44] both deal with management message interchange, and Certificate Request Message Format (CRMF)[45] describes the message formats for management requests and responses.

Note that CMP and CMC can be seen as two competing protocols, and thus contributing to the complexity of the interoperability problem between different implementations.

The Online Certificate Status Protocol (OCSP)[46] is a protocol for handling requests and responses about certificate status. It was designed to overcome the limitations of the traditional CRL-based revocation scheme, by providing fast and up-to-date responses to certificate status requests. But OCSP has some shortcomings of its own: it does neither support verification of certificate signatures nor the validation of the chain of certificates on the certification path. To overcome these shortcomings the Simple Certificate Validation Protocol (SCVP)[42] was designed. It allows the use of both trusted and untrusted servers. Trusted servers perform the full validation service for the relying party while untrusted servers provide intermediate certificates, involved in the certification path, for client-based path validation.

Policies

PKIX defines outlines for certificate policies (CP) and certification practice statements (CPS) in RFC 2527[16]. A CP is used to determine whether a certificate can be applied to an application domain; a CPS controls the operation of a CA. This document gives guidance to PKI implementers about the issues that should be covered in CPs and CPSs.

[5] Attribute Certificate.

[6] Cryptographic Message Syntax.

Timestamp and Data Certification services

The time-stamping service, described in [2], provides support for non-repudiation. It is provided by a Time Stamp Authority, a trusted third party, that signs a message to indicate that it existed before a specific time.

The Data Certification Service (DCS)[3], also provided by a trusted third party, verifies the correctness of data submitted to it. That is, it can certify the validity of an entity's signature or possession of data by an entity. As a result, it produces a validation token that can be used as an evidence of the validity of the signature or possession of data at a specific time.

Interoperability

A major issue in the current state of PKI is interoperability between the various existing solutions. Interoperability involves different aspects, namely, technical aspects, policy aspects, business aspects and juridical aspects, as categorized by the Norwegian PKI-Forum report on interoperability [48].

There already exists a large number of CAs issuing certificates, which, while all based on X.509v3 [36], are according to different profiles and use different extensions. How can a certificate issued in one PKI domain be used in other PKI domains? To what extent can a certificate issued in one PKI domain, according to that domain's policy, be trusted in other PKI domains? How can a distinguished name established in one PKI domain be interpreted in another domain, given that the semantics ascribed to the different name attributes in the cetificate are not usually the same in different domains. These are but a few examples of the issues that an interoperability solution has to deal with.

Issues in Interoperability

One of the fundamental issues in interoperability is the acceptance of digital signatures across jurisdictions. To help this situation, many European countries (European Union) as well as USA have passed legislation to promote digital signatures to the same level of credibility as hand-written signatures. Another juridical issue that needs more investigation is concerned with the responsibilities and liability of the CAs and the relying parties.

Policy and business aspects are closely related. There are already standards defining certificate policies and certification practice statements[16][27][28]. Here, one issue is a rating scheme for the certificate policies/certificate practice statements of different PKI domains. The rating will enable a relying party to determine the degree of trust that can be put in a certificate issued under a given policy/practice statement.

The technical aspect of interoperability is the aspect where the issues are best understood. The issues are mainly concerned with protocols and formats for exchange of information between end-entities and PKI services and between PKI services, schemes for sharing information such as certificates and corresponding revocation lists, and models for trust relationships.

Many initiatives addressing the interoperability issues are under way and propose solutions to the technical issues. The solutions to the different technical issues are of different nature.

Issues such as certificate formats, semantics of name attributes and naming conventions, and protocols for the exchange of PKI-related information require more standardization efforts. PKIX has undertaken such an effort for the Internet. The profile defined by PKIX restricts the various options and extensions of the general certificate format of X.509, to a set useful for Internet. PKIX has also developed operational and management protocols for PKI-related information. Another similar initiative is undertaken by the American National Standards Institute (ANSI) that, in its X9F standard, has defined certificate and certificate extension profiles for Banking community. But, still naming related issues pose major challenges to interoperability.

The issue of trust relationship is at the heart of many of the interoperability inititiatives. As mentioned earlier in the section "Trust models and certification paths", a global root CA, to serve as an anchor of trust, for the whole population of certificate-users is not realistic, and different PKI domains might need different trust models depending on the organizational structure of the domain. The major issue is to maintain trust across PKI domains, i.e., to establish trusted paths between CAs in different domains. The main alternative solutions are summarized below. Details on the pros and cons of the different models can be found in [41].

Cross Certification

Cross-certification is one CA issuing a certificate to another CA. Its main goal is to establish a trust relationship between two CAs. Cross-certification can be unilateral or mutual. Note that cross-certification can also be applied within a single domain to help the performance of the PKI. With the cross-certification model, each PKI domain can retain its autonomy.

The Virtual Operation Network (VON) PKI project is a cross-certification initiative where an architecture for PKI interoperability is implemented [30]. This project also uses replicated directories instead of directory chaining.

Bridge CA

In this trust model, the CA called the *Bridge CA* acts as a mediator, i.e., it introduces one organization to another. Instead of bilateraly cross-certifying each other, each organizations enters into a cross-certification arrangement with the Bridge CA under one or more certificate policies. Two organizations have a trusted path through the Bridge CA, where the certificate policies overlap.

An example of such an effort is the American Federal Bridge Certification Authority (FBCA) [29] project where different government agencies can authenticate each others using the bridge CA. FBCA operates with several levels of assurance, and the approach is standard-driven. There is a set of standards that the CA-implementation has to fulfil, especially regarding the format of the issued certificate, as a refinement of the X.509 standard. There is also a standard for the directory services used under way.

Another project related to the FBCA is the EDUCASE - NIH PKI interoperability project [8]. This is a pilot project in PKI interoperability, where the service is signature on a governmental form. The participants of this project are 5 universities that use four different CA implementations. These implementations date from before the start of the project. The architecture used is a Bridge CA, similar to the FBCA project.

Cross-Recognition

This model is being considered by the Asian Pacific Economic Cooperation (APEC) Telecommunications (TEL) Working Group. This model can be shortly described as independent CAs being licensed or audited by a mutually recognized trusted authority. In this way, to authenticate a subject in another PKI domain, a relying party can use a licensed CA in the other domain[6].

This model puts additional burden on the relying party that is now supposed to make the trust decisions. Cross-Recognition is not suitable where a high degree of trust is desirable.

Certificate Trust List

A Certificate Trust List (CTL) contains a list of “trusted CAs”. It is issued by some CA that has signed it. It also contains policy identifiers and other relevant information.

The concept is that the relying party trusting the issuer of the CTL is then allowed to trust all the other CAs on the list. There is a need to establish a well-defined set of criteria to which a CA must adhere in order to become a “trusted CA”.

Accreditation Certificate

This model is considered in the Gatekeeper project of the Australian government [9]. The Gatekeeper Accreditation Certificate (GAC) indicates that a CA is accredited by the Australian government. Each accredited CA will have its public key signed by the GAC, meaning that the subject CA meets the accreditation criteria of the Australian government. A relying party recognizing GAC as a source of trust

can then consider trustworthy all the GAC-accredited CAs. In this model, the Autralian government plays essentially the role of a root CA.

Problem areas

Another noteworthy effort is the PKI challenge set forth by EEMA (The European Forum for Electronic Business) [22]. In this project funded by the European Commission, ten different PKI software providers tested the interoperability of their products.

The reported experiences from the different interoperability projects identify some problem areas with the existing PKI realizations. The main isuues are:

- The directory naming scheme is important for the usefulness of the PKI.
- The different CA-implementations did not interoperate well.
- Different directory servers and different versions of these products did not integrate well.
- Chaining directories was needed, but this is a part of the X.500 standard that is not brought into LDAP. Workarounds where therefore necessary.
- Too complicated certification paths.
- The directory and PKI-implementations in general were unstable.
- PKI interoperability implementations are lacking.

The main conclusion from the EEMA test was that PKI interoperability can be achieved without major problems using a manual process, but the current standards are far too complex.

SPKI

Despite the fact that PKI has been around for already some time, it is not yet commonplace. This is due to the different legal and technical issues discussed in the sections "Traditional PKI" and "Interoperability". The Simple Public Key Infrastructure (SPKI) initiative [25], started in the late 1990's, proposes an alternative approach to solve the problems of the traditional PKI.

Many are of the opinion that it is the PKI model itself that is the problem [17][24]. This is also the basic assumption of SPKI that, in order to manage proper access to services and information, introduces authorization certificates in addition to authentication certificates. Its major criticism of the traditional PKI is that a globally unique distinguished name for every certificate-holder is not realistic and that, in addition, such a global ID would not tell much about the entity it identifies, for authorization purposes. Providing the necessary information about the entities associated with the global IDs in some sort of directory is bound to raise some privacy issues and is not considered realistic. The essence of the SPKI approach is that it uses the public key itself as the unique identifier of the keyholder.

Another similar initiative, also started in the late 1990's, was the SDSI (Simple Distributed Security Infrastructure)[7]. SPKI has adopted SDSI naming scheme as part of its standard.

According to the "SPKI Certificate Theory" document ([25]),

> "The SPKI Working Group has developed a standard form for digital certificates whose main purpose is authorization rather than authentication. These structures bind either names or explicit authorizations to keys or other objects. […]The name and authorization structures can be used separately or together."

SPKI certificates

The SPKI certificates are significantly simpler than the traditional X.509 certificates. Two kinds of certificates are defined, authorization certificates and name certificates. The name certificates are similar to the traditional certificates, since they connect a name to a key, while the authorization certificates connect the key directly to the authorization. That is, the holder of the key is granted the right to the authorization, so the need for Access Control Lists (ACL) is eliminated.

The certificate is formed using LISP-style S-expressions. These are parenthesized expressions where the first element in any S-expression must be a string called the "type" of the expression. The elements in the expression can be coded in ASCII, hex or base64. It is also possible to express general lists.

```
(certificate
  (issuer    (ref    <my-key>    "Bob
Smith"))
  (subject <bob-key>)
  (not-after 1996-03-19_07:00 )
  (tag (*)))
```

Figure 2: An example of an authorization certificate as an S-expression

A SPKI authorization certificate consists of five fields:

1. Issuer: A public key or a hash of a public key that identifies the entity that issued this certificate, or the reserved word "self" for a self-signed certificate.
2. Subject: A public key, or similar, that identifies the entity being spoken about in this certificate.

3. Delegation: A Boolean value that denotes if the holder of the certificate can delegate the authorization to subordinate entities.
4. Authorization: An S-expression that defines what the holder of the certificate is authorized to do.
5. Validity dates: A not-before date and a not-after date that identifies the period in which the certificate is valid.

A SPKI name certificate consists of four fields:

1. Issuer: A public key or a hash of a public key that identifies the entity that issued this certificate, and that has the name in its name space.
2. Name: a byte string.
3. Subject: A public key, or similar that the certificate connects to the name.
4. Validity dates: A not-before date and a not-after date that identifies the period in which the certificate is valid.

In SPKI there aren't any global names; instead, all names are local. It's possible to reference a name in the name space of another entity by using compound names where the first part is the name of the entity that owns the namespace, and the second part is the name, in this name space, itself.

Authorization

The main feature of a SPKI authorization certificate is that it contains a field for authorization in the certificate. Therefore there is no need for the same kind of precise identification of the keyholder as when ACL's are used. Usually it's enough to be able to identify the keyholder if it has abused the authority granted by the certificate. This also permits anonymous certificates where it isn't possible to identify the keyholder. This can be used in situations where privacy is more important than identifying the abuser.

The authorization is given by an S-expression in the certificate. For example can the expression

(tag (ftp (host ftp.nr.no) (dir /pub/project) (* set read write)))

express that the keyholder is authorized to log into ftp.nr.no and get both reading and writing access to the folder /pub/project.

Delegation

In SPKI, it is possible to further delegate the permissions granted by an authorization certificate without involving the owner of the concerned resources. That is, the subject of an authorization certificate can exercise any permission granted by the certificate, and if delegation is allowed by the certificate, delegate that permission or a subset of it to another entity. To achieve this, the certificate holder issues an authorization certificate with the proper permissions to another entity. The authorization to delegate permissions is expressed with a boolean field in the certificate.

Validation

Validation can be done in the same ways as for traditional PKI certificates. Both CRLs and server look-up can be used. If CRLs are used, it is a requirement that the result of the validation is deterministic. In order to achieve this, all CRLs must have a validity period, and only one CRL can be valid at a specific time.

In [40], alternative forms for validity management are discussed. Especially, methods for certificates that only can be used for a specific number of times are examined. This is done by performing online checks

against a server that keeps track of the validity status of certificates and the number of times they have been used.

Existing PKI Solutions

Several PKI implementations and security software using certificates already exist, some commercial, some freeware and some produced by government initiatives. This section gives a brief overview of the most notable of these implementations.

Web-Browsers

Currently, the most widely used application using a PKI solution is the web-browser. The used PKI solution is greatly simplified. The browsers have a pre-installed set of root certificates of trusted CAs and the web-servers they communicate with may have a certificate issued by one of those trusted CAs. Those CAs are mainly commercial ones and have a self-issued root certificate. It is the implementor of the web-browser that has taken the decision of which CAs to trust. Note that, the end-users can install root certificates of other CAs that they trust in their web-browser. Most browsers check the validity of the web-server's certificate with respect to its validity period but do not support certificate revocation. The commonly used security protocol for exchange of information between the web- browsers and servers is the Secure Socket Layer (SSL). When using this protocol, the web-server will first send its certificate to the web-browser that checks the signature using the root certificate of the issuing CA. The browser then uses the web–server's public key to send a one-time key to the web-server. This approach is regarded as pretty secure, but Ellison and Schneider point out some security issues, both for PKI in general, and SSL in [23]. Transport Layer Security (TLS) [18] has evolved from SSL, and is proposed by IETF as a replacement for SSL.

PGP

Pretty Good Privacy (PGP) [53] is an e-mail oriented PKI solution, allowing for encryption and signing of e-mails. In PGP, users generate key pairs for their own use. Instead of a central authority issuing certificates, users can issue certificates for each other. The degree of trust that can be put in a certificate is established through a so-called *web of trust*. The philosophy behind this is that if a large number of users issue certificates for a keyholder, thus guaranteeing its identity, this identity becomes relatively secure and trustable.

GPG

Gnu Privacy Guard (GPG) [32] is a complete and free replacement of PGP, developed by the GNU foundation. It doesn't use any patented algorithms and can therefore be used without any restrictions. GPG uses the same certificates and protocols as PGP since they both follow the OpenPGP standard [14]; a message that is encrypted using GPG can usually be decrypted using PGP and vice versa. PGP/GPG is among the most used systems for signing and encryption of emails.

Governmental PKI implementations

Most countries have a plan for introduction of PKI, and some have already implementations available. There are two different philosophies that dominate. In the one, it's the government that issues the certificates as in Finland, for example, and in the other, private companies issue the certificates, and certificate-users can choose which CA they want to use. The government then has to set up an infrastructure so that the CAs can be certified, and that the certificates from the different CAs can be used interchangeably. Norway and USA both are planning to implement this latter philosophy.

The implementations have so far not got a wide acceptance. Most implementations suffer from few users and few services. The main reason for this is that implementation of services using PKI is often expensive, and since there are few services, the users have no incentive to obtain a certificate.

Below follows a brief description of the status of two of the first governmental PKI implementations, namely, those in Finland and Australia.

Finland

The Finish government was one of the first governments to adopt an electronic ID (eID). In Finland, an identity card with eID and certificates for encryption and signing has existed since 1999. The identity card can also be used as a normal identity card, and is scheduled to replace the normal identity card in autumn 2003. Finland's Public Registration Centre functions as the CA, while it is the police that functions as a RA. The use of the card is still limited. At the beginning of 2003, there were only some thousand citizens that had obtained a card. There are also a limited number of services available. It's expected that the introduction of cheaper card readers and, eventually, more services will help the adoption of the card.

Australia

Australia was one of the first countries to implement a national PKI with the Gatekeeper initiative [31]. The Gatekeeper infrastructure was fully operational in 1998. Gatekeeper functions as a national root CA. It can certify RAs and CAs. But the use of the infrastructure is still limited. After 5 years of operation, the Tax Office is the only department that has fully adopted it for electronic filing of tax returns. One of the main reasons for the limited use is the relatively high cost of implementing a solution that works with the Gatekeeper.

Status of PKI in Norway

In Norway, the government has the position that the government shall encourage the use of private CAs, and that the government later can certify these CAs with respect to public services [26][47]. There have been several pilot projects using electronic signatures within the Norwegian government.

There are two different initiatives for general PKIs in Norway. One initiative is carried out by Zebsign [52], a company, owned by Telenor and the Norwegian Post, that functions as a CA and delivers PKI solutions. The other initiative is BankID [10] that is a cooperation between the Norwegian banks, and where the goal is to deliver an ID that can be used for all electronic bank services, and is common for all the banks. It's also planned that this ID can be used as a general PKI certificate. BankID is supposed to go into production in the first quarter of 2004.

PKI and Mobility

Traditionally, PKI has evolved assuming a rather static environment: a user's private key(s) and other private data were stored, encrypted, on some hard drive always available to that user. But, roaming users have become a fact of life. They move from one place to another, using different devices, both stationary and mobile, with different capabilities (from cell phone, to PDAs, to PCs), and possibly wireless networks. Users should be able to access their keys as needed whether at home or on the move and regardless of the kind of device they are using or the type of network they are connected to.

The limited capabilities of certain devices, the narrow bandwidth of some networks, and the mobility of users themselves, all have consequences for PKI, and have given rise to new research areas. Different initiatives and research projects have already started to address the different issues related to roaming users: WAP Forum has defined a wireless PKI for WAP[50], the American National Standards Institute (ANSI) has defined, as part of its X9F standard for the banking community, short certificates for bandwidth and storage impaired environments, a protocol for certification services in ad hoc wireless networks is described in [51], a project to build and run collaborative groups over ad hoc networks based on public key support is presented in [20], an approach based on online *Credential Servers* is described in [34], to just mention a few. There is still a need to study the full implications of the mobile and wireless context for PKI and try to make the various efforts converge.

Conclusion

For PKI to gain widespread usage, interoperability challenges must be taken up and solutions must be developed; efficient solutions to checking the validity of a certificate in general [54], and processing of certificate revocation lists [43], in particular, are also of great importance.

The SPKI initiative can be seen as an interesting alternative to the traditional PKI since it addresses both authentication and authorization and has adopted a naming scheme that seems viable. Yet, its usability must be proved by extensive testing in a realistic environment. It is only after such experiments that it can be evaluated with respect to ease of use, interoperability between domains, cost of deployment and other important factors.

Another factor playing an important role in a broader acceptance of PKI, is legislation and agreements, on both national and international levels, that support interoperability across PKI domains.

The issues that have been identified as the major obstacles in the way of PKI becoming common place are listed below:

- The naming scheme;
- Interoperability among the different PKI implementations;
- Lack of integration between different directory servers and different versions of these products;
- Chaining directories, which is a part of the X.500 standard but is not supported by LDAP;
- Too complicated certification paths;
- The directory and PKI-implementations in general were unstable.
- A trusted certificate validation service;
- A well suited trust model;
- Viable business models to facilitate cooperation between commercial actors.

In summary, existing PKI realizations support a homogenous environment with a very small number of CAs and exactly one directory. Serious efforts on both technical and legal fronts are needed for PKI to become a part of everyday life.

Acronyms

AA	Attribute Authority
AC	Attribute Certificate
ACRL	Attribute Certificate Revocation List
CA	Certification Authority
CMC	Certificate management Messages over CMS
CMP	Certificate Management Protocol
CMS	Cryptographic Message Syntax
CP	Certification Policy
CRL	Certificate Revocation List
CRMF	Certificate Request Message Format
CPS	Certification Practice Statement
eID	electronic ID
ECMA	European Computer Manufacturers Association
EEMA	The European Forum for Electronic Business
ETSI	The European Telecommunications Standards Institute
FBCA	Federal Bridge Certification Authority
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
ICT	Information and Communication Technology
IETF	The Internet Engineering Task Force
IPSEC	IP Security Protocol
ITU-T	International Telecommunication Union
LDAP	Lightweight Directory Access Protocol
NIST	National Institute of Standards and Technology
OCSP	Online Certificate Status Protocol
ORA	Organizational Registration Authority
PDA	Personal Digital Assistant
PKC	Public Key Certificate
PKI	Public Key Infrastructure
PMI	Privilege Management Infrastructure
RA	Registration Authority
RFC	Request for comment. Document series from the IETF.
SCVP	Simple Certificate Validation Protocol
SDSI	Simple Distributed Security Infrastructure
SOA	Source Of Attribute
SPKI	Simple PKI
SSL	Secure Socket Layer
W3C	The World Wide Web Consortium

References

[1] Adams, C. and Farrell, S. Internet X.509 Public Key Infrastructure Certificate Management Protocols. *IETF PKIX WG, RFC 2510bis*, April 2003.

[2] Adams, C., et al. Internet X.509 Public Key Infrastructure Time Stamp Protocols. *IETF PKIX WG, RFC 3161*, August 2001.

[3] Adams, C., et al. Internet X.509 Public Key Infrastructure Data Certification Server Protocols. *IETF PKIX WG, <draft-ietf-pkix-dcs-xx.txt>*, March 2000.

[4] American Bar Association. Digital Signature Guidelines: Legal Infrastructure for Certification Authorities and Secure Electronic Commerce, August 1996. *http://www.abanet.org/scitech/ec/isc/digital_signature.html*

[5] American Bar Association. PKI Assessment Guidelines. *V0.30, Public Draft forCcomment,* June 2001. *http://www.abanet.org/scitech/ec/isc/pag/pag.html*

[6] APEC. Achieving PKI Interoperability. *Technical Contribution to the APEC TEL WG*. *http://www.apectelwg.org/apecdata/telwg/eaTG/eatf06.html*

[7] A Simple Distributed Security Infrastructure (SDSI), *http://theory.lcs.mit.edu/~cis/sdsi.html*

[8] Alterman, P., Weiser, R., et. Al., Report: EDUCAUSE-NIH PKI Interoperability Pilot Project. *Proceedings of the 1st Annual PKI Research Workshop*, August 2002.

[9] Australian Government. *Gatekeeper project.* *http://www.noie.gov.au/projects/confidence/Securing/Gatekeeper.htm*

[10] BankID. *http://www.bankid.no/*

[11] Baum, M and Santesson, S. Internet X.509 Public Key Infrastructure – PKI Disclosure Statement. *IETF PKIX WG, Internet Draft*, May 2000.

[12] Blaze, M., Feigenbaum, J. and Lacy, J., Decentralized Trust Management. *Proceedings of the 17th Symposium on Security and privacy*, IEEE Computer Society Press, Los Alamitos, 1996.

[13] Boeyen, S., Howes, T. and Richard, P. InternetX.509 Public Key Infrastructure Operational Protocols – LDAPv2. *IETF PKIX WG, RFC 2559*. *http://www.ietf.org/rfc/rfc2559.txt*

[14] Callas J., Donnerhacke L., Finney H. and Thayer R. OpenPGP Message Format, *IETF, RFC 2440*, November 1998.

[15] Chinowsky, B. Workshop Summary. *Proceedings of the 1st Annual PKI Research Workshop*, August 2002.

[16] Chokani, S. and Ford, W., Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework, *IETF PKIX WG, RFC2527*, March 1999.

[17] Clarke, R., The Fundamental Inadequacies of Conventional Public Key Infrastructure". *Proceedings of the European Conference on Information Systems*, Bled, June 2001.

[18] Dierks, T. and Allen, C. The TLS Protocol, *IETF, RFC 2246*, January 1999.

[19] Diffie, w. and Hellman, M.E., New Directions in Cryptography. *IEEE Transactions on Information Theory*, vol. IT-22, No. 6. November 1976.

[20] Dohrmann, S. and Ellison, C., Public key Support for Collaborative Group. *Proceedings of the 1st Annual PKI Research Workshop*, August 2002.

[21] ECMA. Authentication and Privilege Attribute Security Application with related Key Distribution Functions – Part1, 2 and 3. *Standard ECMA-219*. March 1996.

[22] EEMA. *PKI Challenge, The Results – FAQ sheet.* March 2003. https://www.eema.org/pkichallenge/files/pkiC_faq_sheet.pdf,

[23] Ellison, C. and Schneier, B., Ten Risks of PKI: What You're Not Being Told about Public Key Infrastructure. *Computer Security Journal*, v. 16 no. 1, 2000.

[24] Ellison, C., Improvements on Conventional PKI Wisdom. *Proceedings of the 1st Annual PKI Research Workshop*, August 2002.

[25] Ellison, C. et al., SPKI Certificate Theory. *IETF RFC 2693*, September 1999. http://www.ietf.org/rfc/rfc2693.txt

[26] eNorge 2005, Nærings- og handelsdepartementet, May 2002. http://odin.dep.no/nhd/norsk/enorge/p10001876/024101-990129/index-dok000-b-n-a.html

[27] ETSI. Policy Requirements for Certification Authorities Issuing Qualified Certificates. *ETSI Technical Standard TS-101 456v1.2.1*, April 2002.

[28] ETSI. Policy Requirements for Certification Authorities Issuing Public Key Certificates, *ETSI Technical Standard TS-102 042v1.1.1*, April 2002.

[29] Federal Bridge Certification Authority Homepage, http://www.cio.gov/fbca/

[30] Fink, G., Raiszadeh, S. and Dean, T., Experiences Establishing an Experimental International Coalition Public Key Infrastructure. *Proceedings of the 1st Annual PKI Research Workshop*, August 2002.

[31] Gatekeeper. *http://www.noie.gov.au/projects/confidence/Securing/Gatekeeper.htm*

[32] GNU Privacy Guard (GPG). http://www.gnupg.org/

[33] Green, R. M: and Harris, B., United States DoD Public Key Infrastructure: Deploying the PKI Token. *1st Annual PKI Research Workshop*, August 2002.

[34] Gupta, S., Security Characteristics of Cryptographic Mobility solutions. *Proceedings of the 1st Annual PKI Research Workshop*, August 2002.

[35] Housley, R. and Hoffman, P. InternetX.509 Public Key Infrastructure Operational Protocols: FTP and HTTP. *IETF PKIX WG, RFC 2585*, http://www.ietf.org/rfc/rfc2585.txt

[36] ITU-T /ISO, OSI – The Directory: Authentication Framework. *ITU-T X.509 / ISO/IEC 9594-8*, 1997.

[37] ITU-T, The Directory - overview of concepts, models and service. *International Telecommunications Union, X.500 series of Recommendations*, 1993.

[38] Kent, S. T., Internet Privacy Enhanced Mail. *Communications of the ACM*, August 1993.

[39] Kohnfelder, L. M., Towards a Practical Public Key Cryptosystem. *MIT S.B. Thesis*, May 1978.

[40] Kortesniemi, Y. Validity Management in SPKI. *Proceedings of the 1st Annual PKI Research Workshop*, August 2002.

[41] Lloyd, S. et al. CA-CA Interoperability. *White paper*, March 2001. http://www-pkiforum.org

[42] Malpani, A. and Hoffman, P. Simple Certificate Validation Protocol (SCVP). IETF PKIX WG. Work in progress.

[43] Micali, S., NOVOMODO – Scalable Certificate Validation and Simplified PKI Management. *Proceedings of the 1st Annual PKI Research Workshop*, August 2002.

[44] Myers, M., et al. Certificate Management Messages over CMS. *IETF PKIX WG, RFC 2797*. http://www.ietf.org/rfc/rfc2797.txt

[45] Myers, M., et al. Internet X.509 Certificate Request Message Format. *IETF PKIX WG, RFC 2511*. http://www.ietf.org/rfc/rfc2511.txt

[46] Myers, M., Ankney, R., Malpani, A. and Adams, C., X.509 Internet Public Key Infrastructure Online Certificate Status Protocol –OCSP. *IETF PKIX WG, RFC 2560*, June 1999.

[47] PKI-Forum. Strategi for en samfunnsinfrastruktur for elektronisk signatur og elektronisk ID i Norge. June 2002.

[48] Rammeverk for PKI Samtrafikk. *Samtrafikkgruppa, PKI-forum*, December 2002, http://www.handel.no/FileArchive/239/Samtrafikkrapportenv1[1].01.pdf

[49] Rivest, R. L. and Lampson, B., SDSI – A Simple Distributed Security Infrastructure. *Presented at CRYPTO'96 Rump session*, 1996. http:// citeseer.nj.nec.com/rivest96sdsi.html.

[50] WAP Forum. WAP Public Key Infrastructure Definition. http://wwwwapforum.org/.

[51] Yi, S. and Kravets, R., MOCA: Mobile Certificate Authority for Wireless Ad Hoc Networks. *Proceedings of the 2nd Annual PKI Research Workshop*, April 2003.

[52] Zebsign. *http://www.zebsign.no/*

[53] Zimmermann, P. R., *PGP user's Guide*. MIT, October 1994.

[54] Ølnes, J. Trusted Certificate Validation Services – Breaking the PKI Deadlock. October 2002. *White paper from Validsign AS*, Norway. http://www.validsign.com/Library/validation-paper_1_0.pdf