Attribute Certificates in X.509

Toni Nykänen
Helsinki University of Technology
Department of Computer Science and Engineering
Toni.Nykanen@hut.fi

Abstract

This paper is an introduction to Attribute Certificates in X.509 Framework, the Privilege Management Infrastructure (PMI) and its relation to Public Key Infrastructure (PKI), concentrating on the work done by IETF Working Group PKIX, aiming at adapting PKI to the Internet.

Comparison of X.509 PKIX Attribute Certificates and Simple Public Key Infrastructure (SPKI) certificates is presented.

Attribute certification in essence is a way of extending authentication-oriented use of PKI to support tasks related to authorization. Attribute Certificates provide a solution to certify binding of attributes to a given subject.

Applications of Attribute Certificates cover a wide range of topics such as web access control, transport layer security, secured e-mail and integration of legacy systems.

1 Introduction

Public key cryptography reduced the number of keys needed for practical secure communication over an insecure channel. Before the discovery of public key cryptography, each communicating pair of hosts needed a unique key. Thus n hosts communicating with each other would need $(n^2 - n)/2$ keys. Even worse, the keys had to be delivered to hosts via a secure channel, because the same key was used for encryption and decryption [9].

Public key cryptography in turn revealed a problem of effective and trusted publicly available means of distribution of public keys. For this key distribution several methods were proposed, varying greatly in centralization and other characteristics of the service.

For key distribution over an insecure channel to become secure, a signed document, signed by the key distributor, is used. The key distributor acts as a Trusted Third Party (TTP), an entity trusted by communicating peers. Thus, for a communicating peer it is enough to trust the key distributor and to get hold of the public signing key of the key distributor safely. With this, the peer can verify the certificates issued (and signed) by the distributor, and gain access to other peers' public keys safely.

The signed document used to distribute the public keys is known as a *certificate.* A certificate is a document, signed by a Certificate Authority, which binds properties to a subject

that is identified in an appropriate fashion depending on the context and the type of a certificate. In PKIX X.509 identity certificates the subject is a unique name, and the property bound to the name is a public key.

For many applications, the binding that requires a verification from a TTP, is not the binding of the name to a public key, but rather a binding of an identity to a set of attributes. This kind of certification is very useful for example in distributed access control, where a TTP, known as Attribute Authority, issues these bindings, which in turn declare the rights of a subject to specific objects. For an authority, responsible of access control decisions of objects under its control, this kind of certificates ease the overall management burden needed for the access control. There is no need for specific Access Control Lists (ACL), because the certificate, called *Attribute Certificate*, declares the subject's rights to access the particular objects. This especially eases the task of management of ACLs in a system where the objects themselves are responsible of the access control of their own resources. When the ACLs are largely scattered around the numerous objects, the task of keeping them consistent according to the chosen policy becomes very difficult. When using Attribute Certificates, the object only has to verify the certificate given by the subject, and verify that the identity claimed in the Attribute Certificate really is the identity of the subject attempting to access the object. This additional verification of identity is fairly permanent, therefore it is not necessary to be verified for every single access control decision, but at the beginning of each session, if there is such. The validity of an authentication may vary from a single transaction to any long time. A tradeoff between consumption of bandwidth and time, and level of security in pace of required authentications is obvious, known in literature as "Time Of Check To Time Of Use" [17].

2 Terminology and Concepts

2.1 Terminology

- PKI Public Key Infrastructure

  The set of hardware, software, people, policies and procedures needed to create, manage, store, distribute, and revoke Public Key Certificates (PKCs) based on public-key cryptography. [3]

- PMI Privilege Management Infrastructure

  A collection of Attribute Certificates (ACs), with their issuing Attribute Authorities (AAs)s, subjects, relying parties, and repositories, is referred to as Privilege Management Infrastructure [3].

- X.509

  X.509 is an ITU-T (ISO/IEC) Recommendation for a framework for public-key certificates and attribute certificates. Current draft can be found from [22].

- PKIX Internet X.509 Public Key Infrastructure

  An effort launched in October 1995 by an IETF Working group of the same name to deliver a profile for the Internet PKI of X.509 version 3 PKCs and version 2

Certificate Revocation Lists (CRLs) [3]. The PKIX definition of PKI subsumes PMI as well.

- SPKI Simple Public Key Infrastructure

  a kind of PKI that emphasizes on authorization rather than authentication. SPKI certificates bind attributes (describing authorization) to a public key directly [12].

- CA Certification Authority

  An authority trusted by one or more users to create and assign public key certificates. Optionally the CA may create the user's keys. It is important to note that the CA is responsible for the public key certificates during their whole lifetime, not just for assigning them [3].

- AA Attribute Authority (also called AC Issuer)

  An authority trusted by one or more users to create and sign attribute certificate. It is important to note that the AA is responsible for the attribute certificates during their whole lifetime, not just for issuing them [3].

  AA can be any entity in the network having objects in its control. In literature, the AA is often called AC Issuer.

- RA Registration Authority

  An optional entity given responsibility for performing some of the administrative tasks necessary in the registration of subjects, such as: confirming the subject's identity; validating that the subject is entitled to have the values requested in a PKC; and verifying that the subject has possession of the private key associated with the public key requested for a PKC [3].

- Attribute Certificate Verifiers

  Attribute Certificate Verifiers check the validity of an AC and then make use of the result [3].

- PKC Public Key Certificate

  A data structure containing the public key of an End Entity (or CA, or AA) and some other information, which is digitally signed with the private key of the CA which issued it [3]

  In PKIX, PKC is a Public Key Certificate of the type defined in X.509 [22], and profiled in RFC 2459 [18].

- AC Attribute Certificate

  A data structure containing a set of attributes for an end-entity and some other information, which is digitally signed with the private key of the AA which issued it [3].

- Repository

  A database service capable of storing information, such as certificates and CRLs, allowing unauthenticated information retrieval. Repositories include, but are not limited to, directory services [5].

- Client (PKI Client)

  Clients validate digital signatures and certification paths from a known public key of a trusted CA [3].

  A function that uses PKI to obtain certificates and validate certificates and signatures. Client functions are present in CAs and entities. Client functions may also be present in entities that are not certificate holders [5].

- Client (PMI Client)

  PMI Clients request an action for which authorization checks are to be made [3].

- Certificate Holder

  An entity that is named as the subject of a valid certificate [5].

- End Entity

  A certificate subject which uses its private key for purposes other than signing certificates [5].

2.2 Concepts

- Certification binding

  According to Ellison, there are three classes of information that can be bound together by public key certificates or attribute certificates [12]. These are *key*, *name*, and *authorization* (or other attribute). The binding may be indirect: for example, instead of a key, a hash of a key, or a reference to another certificate containing the key, may be used.

  Thus there are three general kinds of certificates, mapping two of the three classes of information together:

  - (authorization<->name)
  - (authorization<->key)
  - (name<->key)

- Trust

  Trust is used and abused relentlessly in literature related to PKI. The problem is that the definitions usually are too vague. A clear distinction can be made between psychological and technological definitions of trust.

  Psychological Trust is not transitive. If it was transitive, say if Alice trusted Bob, and Bob trusted Carol, a person Alice has never even met face-to-face, then by transitivity, Alice would also trust Carol. Computation of transitive closure of trust would then be exhaustive, yet boring, because at the end it would be very likely that everyone in the world would trust everyone. (Effectively rendering this paper and the whole discipline introduced here useless as well.)

  However, when considering trust in a more precise sense, e.g. related to some specific technology, trust may be transitive, but then trust is restricted to a specific aspect. Often the system relies upon trust between the components of the system, and in case of a compromise in trust relations the incident has a wide effect.

Thus, it is hard to give a precise definition of what *trust* is, even in a very limited context.

Nikander [25] discusses the issue of psychological trust and the differences compared to definitions used in a more specific context.

In [24], Nikander presents a more technological definition for trust:

Trust in a principal is a belief that the principal, when asked to perform an action, will act according to a pre-defined description. In particular, this belief implies the belief that the principal will not attempt to harm the requestor independently of the way it fulfills the request.

- Chain of Trust

  A Chain of Trust makes use of transitivity of trust (in this context). For example, a chain is formed in the process of verification of a certificate when the certificate is issued by a non-local Certificate Authority or Attribute Authority. In this chain, each CA trusts a following CA in the chain. Trust between consecutive entries in the chain is formed by a possession of public key of the successor in the chain, which verifies the signatures of the successor.

- Role-based Access Control

  Role-based access control focuses on the users and the jobs users perform. A Role is a collection of rights to objects (actions that can be performed at certain targets. A user may have one or several roles. For access control, role is a kind of intermediary that reduces the amount of information needed in the ACL by decreasing the granularity of access control information. Permissions to objects of the system are not listed for each users separately, but users are given roles and rights of each role are only described once.

- Delegation

  An object that has a permission to access another object may *delegate* a set of its access permissions to that object to a third party.

  Thus delegation, according to this definition, implicitly contains the object's own right to access the another object, not just the right to delegate the right.

3 Motivation

The motivators for the survey are presented here briefly. At first, problems related to authorization issues are listed, and after that two distinct approaches for providing attribute certification are given. The section ends with some current topics concerning the attribute certificates.

3.1 Support for Distributed (Role-based) Access Control

Any access control relying strictly on authentication and enumeration of subjects in ACLs is problematic in distributed environment because of the need to keep the ACLs consistent

and up-to-date, especially when permissions to access objects are granted for very short periods.

Identity certificates are not of much help in distributed access control. PKCs have fairly long lifetimes and PKCs don't support presentation of possibly encrypted attributes bound to the holder of the PKC (for limited support, see [18]), which could be used to declare subject's roles. Due to these limitations identity certificates can only be used to describe holder's identity in Access Control Lists. This doesn't help the burden of administration, since each identity has to be assigned explicit access to the system, and ACLs need to be edited every time permissions are modified.

If part of the Access Control information could be included in some interchangeable and Third Party-signed record, which the subject could present during the authorization, the maintenance of ACLs would become easier. For example, if this signed record declares the subject's membership in a specific group, and the local authorization management maintains the access rights of the groups, this scheme would separate the resolution of subject's group affiliation from the management of the privileges of the group. Even the privileges of the group could be specified in a signed record (a certificate), which would be pointed to from the certificate declaring the subject's group membership [23].

3.2 Support for Delegation of rights

With distributed access control it would be favorable that subjects could delegate their permissions to other subjects with no interaction with the authority during the process of delegation. Thus it would be possible to take the decentralization of authorization a step further. Certainly delegation should be optional, and in some applications it would be beneficial if the delegator was able to further restrict the set of delegatable permissions.

3.3 Means of providing legacy (enterprise) integration

In real life there are numerous computer systems that require traditional password-based authentication (and implicit authorization based on the permissions for the user in the ACL).

If these passwords could be distributed in a safe way such that only the real recipient would be able to decrypt them, then this access control method could be used to "wrap" the old password method. For a user, "Single sign-on" would be an ideal goal. (However, Single sign-on is a tradeoff between usability of the system and security of the system [17].)

3.4 Attribute Certification as a solution

Attribute Certification, in form of X.509 Attribute Certificate profile in PKIX, and as a main function of SPKI certificates, provides solutions to the issues presented above.

With attribute certification, it is possible to use a trusted third party to vouch for a binding from a certain identifier and a set of attributes that can be used to describe e.g. membership to some group. In PKIX the focus has been in certification of one's identity with ordinary

public key certificates, but a profile for attribute certificates has also been developed and the current version of the draft is 5 [15]. Standardization process is still going on, and it is not known when the RFC will be issued.

There is a need for certification of "short-term" attributes to some identity. PKIX and SPKI have taken different approaches to this, varying in centralization and means of identifying an object, where the other one concentrates on unique names and the other one on public keys themselves.

3.5 Ease of deployment

Besides integration to existing PKIs, the solution should require minimal or no changes to existing Internet protocols that could use attributes bound to identities with certificates. The protocols that would benefit from this include TLS, S/MIME and IPSEC [29, 28, 21, 8, 10].

3.6 X.509 (Attribute) Certificates' relation to AAA facilities

Clearly the X.509 PKI, including the PMI model, could be used to implement part of the defined AAA functionality [1]. However, no signs of collaboration between the working groups has surfaced so far. AAA Working Group's main focus is on providing generic means for *Authentication, Accounting* and *Authorization* for network access, whereas PKIX is an attempt to profile the ISO X.509 PKI to the Internet in general. Similarities are obvious.

3.7 Data encoding in Certificates

Standard for data presentation in ISO, and PKIX, is ASN.1, and SPKI has its own Lisp-like S-expressions [11]. But XML is becoming more and more important in the industry. There is an expired draft concerning the XML encoding of SPKI certificates [26], and there has been a lot of discussion about support for XML encoding of PKIX certificates in PKIX mailing lists lately [27].

4 Internet X.509 Public Key Infrastructure (PKIX)

4.1 Overview

The PKIX is based on X.509, an ITU-T (ISO/IEC) standard of Public Key Infrastructure. X.509 is a comprehensive, and largely accepted standard, but in itself of little use. For example, it leaves unspecified many of the important fields of the PKC data structure.

Due to its ancestors' focus on authentication, PKIX as well concentrates mainly on providing means for identification and authentication, with a Public Key Certificate that provides a binding from unique names to public keys.

The Attribute Certificates were included into X.509 in 1997, and a model of Privilege Management Infrastructure was introduced in 1998. Thus even to PKIX, the Attribute Certificates are like an add-on attempt to provide support for authorization requirements.

In the following, types of components involved in PKI, and in PMI, are listed and explained. PKIX profiles for Public Key Certificate and Attribute Certificates are also presented. Emphasis in profiles is on common uses of certificates.

4.2 Components of X.509 PKI

Terminology is explained in section 2.1.

X.509 PKI consists of the following types of components:

- Certification Authorities (CAs)
- Organizational Registration Authorities (ORAs)
- Certificate Holders
- Clients
- Repositories

The interaction of the components is shown in figure 1.

4.2.1 Basic functionality of a PKIX PKI

A short example of an ordinary functionality of a PKIX PKI is probably the best way to bind the different components of a PKI together. The example is modified from an example in [30]:

Suppose Alice wants to communicate safely with Bob, but she doesn't yet have own keys or certificate to use in communication. Suppose also that Alice has already got hold of Bob's public key to avoid redundancy in example:

Alice first generates a public/private key pair using a public key algorithm like RSA. Then she creates a certificate request, which is the certificate just prior to its signing by the Certification Authority. The request contains a unique name of Alice, possibly accompanied with some additional information concerning her. The request also contains Alice's public key. Alice must send this request to a Registration Authority encrypted with RAs public key, or perform this step out of band.

Next, the Registration Authority decides the approval of the request, resulting in an approval or disapproval, and in case the request is approved, the request is forwarded to CA for policy approval and signing. A signed request is Alice's new certificate, which is sent back to her through RA.

Now, as Alice has a valid certificate, Bob is able to decrypt the messages Alice just sent to him encrypted with Bob's public key and signed with Alice's private key. But first Bob

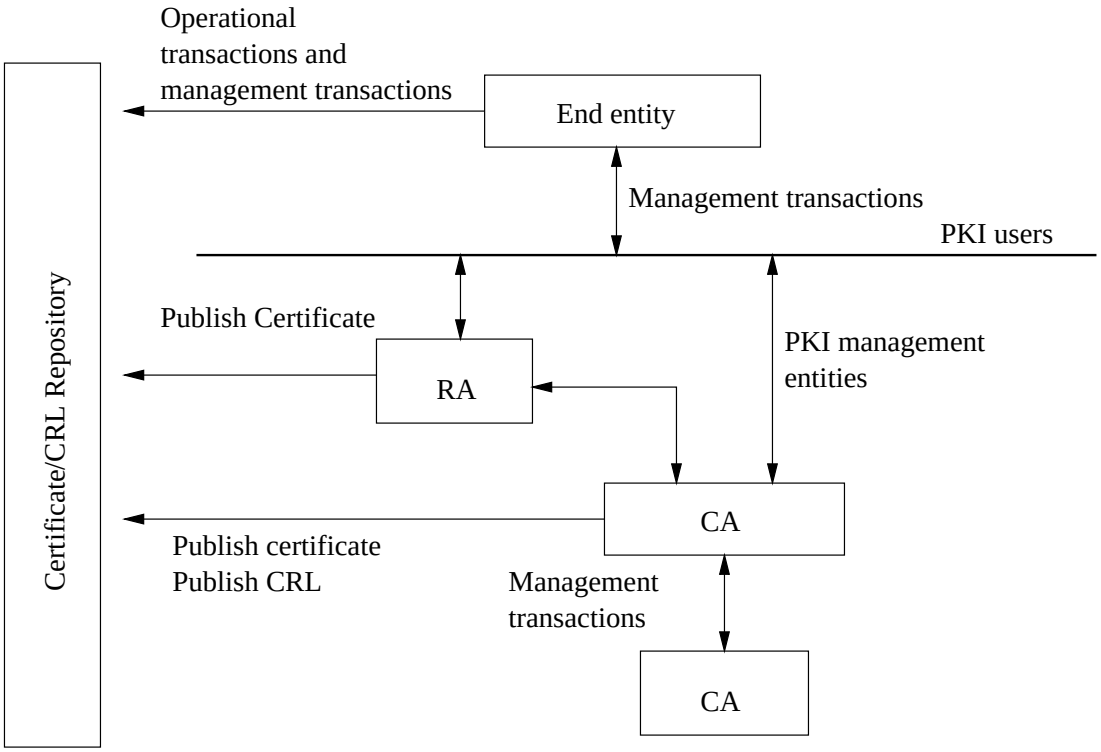

Figure 1: PKI Entities [3] (Drawing is from [30]).

must get Alice's certificate, and verify it. Usually certificates are stored and fetched from a repository.

If Bob and Alice are both on the same CA, then it is easy for Bob to verify Alice's certificate. Bob must have already got hold of CA's public key in a secure way. With that key, Bob can verify Alice's certificate, which is signed by the same CA. However, if Bob and Alice have different CAs, then Bob must ask his CA to find a trusted path to Alice's CA. The trusted path must be such that for each CA the CA acting as a client has already got the serving CA's public key, with which the client may verify the server's signatures. This kind of chains of trust, especially when lengthy, are critical, since one compromised or bad CA can undermine the security of the whole infrastructure.

Compromised CAs, as well as End Entities having a compromised certificate are listed in Certificate Revocation Lists. PKI users in general, when using a certificate, check for the validity of the signature, and acquire a suitably recent CRL and verify that the list doesn't contain the serial number of the certificate being validated. What is a suitably recent CRL depends on the accepted policy [7].

4.3 Components of X.509 PMI

Terminology related to components is presented in section 2.1.

Interaction of components in exchange of attribute certificates is presented in figure 2. The ACs can be "pushed" to AC Verifiers as part of the application protocol which uses ACs

for authorization. This naturally requires changes to the protocols. ACs may also be stored in a repository and retrieved when needed. This model is called the “pull” model.

- Attribute Authorities
- Attribute Certificate Users
- Attribute Certificate Verifiers
- Clients
- Repositories

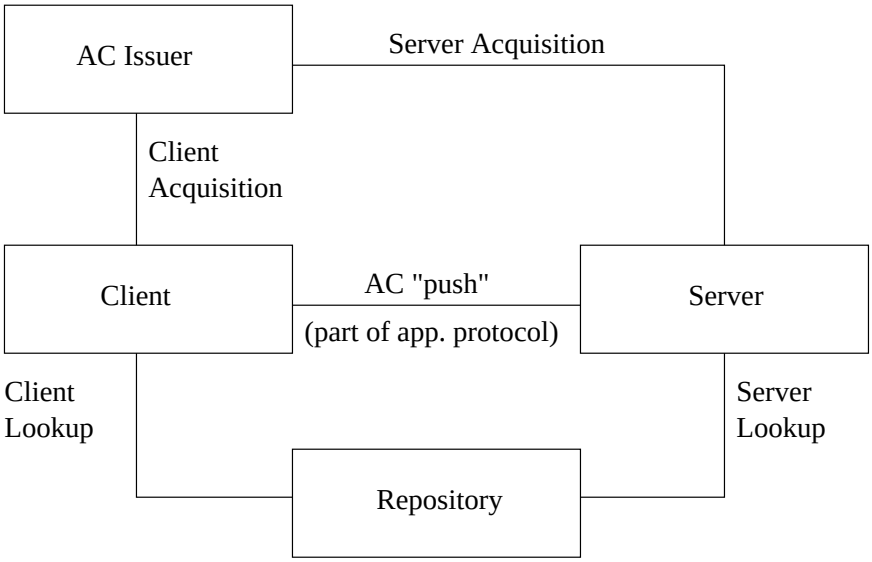

Figure 2: AC Exchanges [3]. (Drawing is from [30].)

The most common use of a PMI is the process of verification of the attributes of a subject. The verifier must first get hold of the AC of the subject (the AC holder). The two ways of doing this are shown in figure 2. In general the “Push” model requires changes in the application protocols, but is more efficient, since there is no need for an additional request for the server to retrieve the AC from the repository. Once the AC verifier has the AC, it checks that the AC signature is valid. Then, if the holder field of the AC states that the holder is identified by a PKC, the validation path (a chain of trust) of the PKC is verified, and after that the PKC itself is checked for validity. The issuer of the AC must be directly trusted by the AC verifier, since the version 6 of the draft [15] does not support delegation. After that the validity time of the AC is checked. After that, basically the AC is valid and the AC verifier may perform any decisions based on the attributes of the AC that belong to the AC holder. (In this example it is taken for granted that the holder of the AC that presented the corresponding PKC really has the claimed identity. This could be proven e.g. by signing the request with the private signing key of public key the pair of which is in the PKC.)

4.4 X.509 Certificate profile

The profile defined by PKIX is presented on general level. The commonly used fields are presented. For more information, see [18].

4.4.1 Fields of X.509 Public Key Certificate

In essence, an X.509 PKI binds a subject's name to subject's public key, as vouched by the name of the issuer, the CA.

The fields in the body of the certificate are presented in table 1.

X.509 PKC	
Field name	Explanation
version	Version of the Certificate; V3 has value of 2.
serialNumber	Assigned by CA. Unique within the PKCs issued by the CA.
issuer	The name of the entity that has signed the PKC, i.e. the CA.
validity	The validity period of this PKC.
subject	Name of the entity whose public key is present in the PKC.
subjectPublicKeyInfo	Subject's public key and algorithm identifier
issuerUniqueID	
subjectUniqueID	
extensions	
signatureAlgorithm	The algorithm used by the CA to sign this PKC.
signatureValue	The result of CA signing the certificate.

Table 1: Fields of X.509 v3 PKC. Optional fields are presented in italics.

Signature Value is the result of CA signing the certificate. With this value, CA certifies the validity of the certificate, including the binding between the name and the public key of the subject.

The Serial Number field is a serial number assigned by the CA. It is unique for each certificate issued by the CA. Thus, name of the CA and the serial number together make a globally unique identifier for the certificate, which can be used to refer to the certificate. For example, the *holder* field (see section 4.5.1) in attribute certificate can use this to refer to AC holder's public key certificate.

Validity tells the time interval during which the CA warrants that it will maintain information about the status of the certificate.

Subject Public Key Info contains the public key associated with the *Subject*, and the identifier of the algorithm with which the key is used.

The Extension field specifies several refinements concerning the use of the certificate. It can be used to:

- Identify the key used in signing of the certificate, in case the CA has several keys.
- Identify the public key of the subject, in case the subject has several keys.

- Describe alternative names for the subject, such as an IP address, DNS name, or an SMTP mail address.
- Restrict the usage of the certificate, such as maximum length for the certificate validation path, or limiting the validation into a specific name space only.
- Declare CRL Distribution point.

4.4.2 Certification Paths and Trust

Each End Entity has a single "most-trusted" CA which is the start of any validation path. For two entities willing to communicate securely using the PKIX they must have a common trusted CA. In X.509 v1 the CAs formed a pure hierarchy, in which each CA trusted a single CA. But X.509 v3 certificates don't have this limitation, and a CA may trust several other CAs. Still, CAs usually form a hierarchy within a single administrative domain. But two different CAs belonging to different domains, and not sharing a common upper-level CA, can establish trust by means of *cross-certification*, a kind of CA certificate, which is an ordinary PKC, where subject is also a CA. (Cross certification may also be used to form normal hierarchies within a single administrative domain as well, but usually these hierarchies are formed out of band.)

4.4.3 Revocation, CRL

Revoked certificates are added to a Certificate Revocation List. The list is distributed in the same way as the certificates. The list is updated periodically according to the local policy. Depending on the policy, it is possible to issue incremental or complete CRLs.

4.4.4 Operational and Management Protocols.

Operational and management protocols are not presented in this paper in detail. Overview of their uses is in [3].

Operational protocols are used in PKI to deliver certificates and CRLs to certificate users [4]. LDAP [6] is one protocol that is used to deliver PKCs and CRLs. LAAP [16] is used to deliver ACs.

Management protocols are used for on-line interaction between PKI users and management entities, for example between a CA and a client system, or between two CAs that cross-certify each other. These protocols are used e.g. to register PKI users or client systems and requests for revocation of certificates [2].

4.5 X.509 Attribute Certificate profile

The following information is presented as stated in the PKIX Attribute Certificate Profile for Authorization draft 5 [15].

4.5.1 Fields of X.509 Attribute Certificate

The fields in the body of the X.509 attribute certificate are presented in table 2.

X.509 AC	
Field name	Explanation
version	Version of the AC; v1 or v2
holder	Holder of the AC.
issuer	The AA that has issued this AC.
serialNumber	Assigned by AA. Unique within the ACs issued by the AA.
attrCertValidityPeriod	The validity period of this AA.
attributes	The “payload”; the attributes bound to the AC holder.
issuerUniqueID	
extensions	
signatureAlgorithm	
signatureValue	

Table 2: Fields of X.509 Attribute Certificate. Optional fields are presented in italics.

The PKIX Attribute Certificate has the same fields describing the signature algorithm and the signature value as the X.509 Public Key Certificate. Version field is also present.

The Holder field describes the holder of the attribute certificate. There are several ways to describe this, the most common of which is a reference to the holder’s PKC via unique serial number. This binds the PKI and the PMI infrastructures together; the certification becomes dependent on both the CA of the PKC and the AC issuer (AA). The draft gives other possibilities for the presentation of the holder also: Entity Name and Object Digest Info. Entity name could be useful if some other authentication than holder’s PKC is used. For example, the authentication may have already been verified in the same session context the verification of authorization is being performed, by e.g. a safe exchange of user name and a password. Then, this user name can be used as a holder of the AC and the privileges or other information presented in the attributes of the AC.

Of course simply presenting an AC and a PKC the AC refers to as its holder is *not* enough for a verifier to believe that the presenter is the subject of the PKC. The presenter must sign the message with the private key the public key pair of which is certified by a CA in the PKC.

Object Digest Info is an interesting way to present holder: it is possible to bind the AC directly to any object by way of a hash calculated of this object and placed in this field. Also, the Object Digest Info makes it possible to bind authorizations directly to public keys. This could be seen as an attempt to provide the “nice” features of SPKI in PKIX. The ability to bind attributes to an arbitrary object really opens new possibilities for PKIX.

So far, only two kinds of digests are defined in the draft: hash of a public key and a hash of a PKC. However, this doesn’t forbid the use of other digests. For example, certifying executable (Java) objects directly via digests could provide useful.

The Issuer links the attribute certificate to the issuer usually by a single GeneralName, which may contain e.g. a DNS name or an IP address of the issuer.

Serial Number identifies the AC uniquely from the other ACs issued by the AA.

Validity Period states the period for which the AA certifies that the binding between the holder and the attributes will be valid.

The Attributes field of the AC can contain any data. Standard types of attributes are the following:

- Service Authentication Information
- Access Identity
- Charging Identity
- Group
- Role
- Clearance

Service Authentication Information can be used for providing legacy applications with needed credentials. Access Identity can be used to identify the AC holder to the AC verifier or the larger system of which the AC verifier is a component. These attributes are usually sensitive information and are thus encrypted.

Charging Identity identifies the AC holder for charging purposes. For example the holder's company may be the charging identity.

Group and Role can be used to present information of AC holder's roles or group memberships.

Clearance is the clearance of the AC holder, associated with security labeling [17].

Extensions field in PKIX attribute certificate can be used to further restrict the applicability of the AC into targets, and to specify distribution points for CRLs, or to declare that no CRL for the certificate is available.

4.5.2 Certification Paths

The draft relies on the assumption that there's a single Attribute Authority that issues all ACs for a particular set of attributes. As long as the attribute sets are disjoint, there may be several authorities, each responsible for its own set of attributes. For example, clearance may be issued by one authority, and group and role memberships by another authority.

4.5.3 Revocation of Attribute Certificates

The Attribute Certificates can be revoked, depending on the chosen policy, in two ways: implicitly, by short validity periods, or in the same way as the PKCs, by using Certificate Revocation Lists, which are issued periodically according to the local policy, and contain the serial numbers of revoked certificates.

4.5.4 Delegation Support in Attribute Certificates

The current draft (number 5) doesn't support delegation, because that would require chains of ACs, which are complex to process and administrate. The validation of AC holder's PKC may have chains, just like in the ordinary case when only authenticity is verified.

However, this restriction applies only to the IETF profile of the PKIX. The ISO standard [22] does support delegation.

4.5.5 Attribute Encryption

The Attribute Encryption can be used to encrypt the attributes when the ACs are transferred insecurely in the network. The encryption is done so that cipher text stealing is prevented: the encrypted data contains the name of the AC Issuer, and the serial number of the AC. These values must be equal to the corresponding value in the body of the AC.

The attributes are encrypted using Cryptographic Message Syntax, EnvelopedData structure [19]. In this structure, the following scheme is possible: for each message, a symmetric key is generated, and this key is then encrypted with each recipient's available public key and included in the EnvelopedData, which then contains the symmetric key encrypted with each recipient's public key, and the actual data which is encrypted with the generated symmetric key.

4.6 Deployment

Xenitellis presents in [30] a good source to look at for free implementations. It gives also a comprehensive list of the (already existing) free software components needed to set up a Certificate Authority. Also, it serves as a comprehensive tutorial into world of PKI.

SSH have products implementing the functionality of a CA and an EE.

However, neither of these sources claim any support for PMI.

5 Simple Public Key Infrastructure (SPKI)

SPKI was developed in response to achieve a PKI the main emphasis of which would be *authorization,* not *authentication* [12]. Initially, the basic SPKI was only concerned of certifying bindings of public keys and attributes. Pretty soon SDSI, Simple Distributed Security Infrastructure [12], a means of defining and using local name spaces, and still being able to identify names in certain namespaces globally, merged with the SPKI, and thus binding between names and public keys became possible as well.

5.1 Principles of SPKI

- Key as global ID

Thanks to mathematics, an ordinary randomly-generated public key of length of 2048 bits (RSA in this example), is globally unique. Therefore it can be used to index a certificate. Also, since the "namespace" of such keys is flat, with no obvious subfields, people are not likely to use plain keys and therefore cannot make faulty guesses of other field's contents (as is possible in case of globally unique names).

- Certificate fields

  An SPKI certificate is very simple: it contains keys of the issuer and subject, information on the delegatability of the authorization (yes or no), validity period, and the issuer's signature. Details can be found from [11].

- Simple authorization

  The SPKI authorization certificates bind permissions to keys directly. Not having to bind permissions to names (or PKCs), and names to keys, is an obvious advantage. There is only a single TTP (or chain of TTPs) to trust in any case.

- Support for delegation

  SPKI authorization certificates support delegation. The delegation can be controlled in a boolean fashion: either the holder is able to delegate the permission or not.

- Storing and providing certificates

  In SPKI, certificates are usually not stored in a global repository. Certificates certifying authority are "pushed" by the keyholder to the verifier. The assumption is that the authorization certificate is of no use to anyone else than the holder and the verifier.

- Fault tolerance

  As described above, the strict hierarchy of PKIX is very sensitive. In case of a malfunction or a compromise of a single CA the part of the PKI in its control is paralyzed, and the communication of these End Entities is prevented or compromised. SPKI addresses this problem by allowing validation of a certificate to have several paths (i.e. there is no hierarchy), and letting the issuer of the certificate to decide what level of trust is acceptable. The construct is called *k-of-n threshold subject* [12, 14].

- Names

  SPKI offers binding of a *local* name to a key.

  Local name is a name that has significance only to the issuer herself. Local names can be chained as follows: Assume keyholder of K_0 defines **(name jim)** to be K_1, and the keyholder of K_1 defines **(name therese)** to be K_2. Now, the keyholder of K_0 can use **(name jim therese)** to refer to K_2.

  The local names can be made "global" by anchoring the outermost name space with the key of the keyholder of that local namespace. In the example above, **(name** K_0 **jim therese)** is globally unique and at the same time meaningful locally, actually referring to key K_2.

5.2 Deployment

So far, SPKI seems to have gained mostly academic interest. [14] has a list of commercial products and available source code of the (partial) implementations.

The development of SPKI seems somewhat stalled: the draft describing the format of the certificate [11] expired in January.

A good resource for SPKI-related material, such as articles, source code, discussion, and comparison to other PKI technologies, is Carl M. Ellison's home page [14].

6 Comparison and Conclusions

In the table below, taken partially from [14], characteristics of both PKIX and SPKI are presented.

	X.509/PKIX	SPKI without SDSI	SPKI with SDSI
CA characteristics	CA hierarchies, possibly cross-certified. Single AA hierarchy.	AA hierarchies: optional k-of-n subjects	Single naming authority
Kind of Identifier	Global, but in practice used as local. Very often a reference to DNS is used as a global identifier.	Global (as long as the keys are properly generated)	Local
Uses	Identity certification. Extended towards authority certification. Authority certification so far is useful only when used together with authenticity certification.	Authority certification: keys to attributes	Identity certification.

Table 3: Characteristics of PKIX and SPKI with and without names.

Two problems in PKIX that initiated the design [13] of SPKI were:

- Centralized and hierarchical key management.

  In PKI every End Entity essentially has only one CA to look for PKCs. The same applies in PMI to AAs [15]. This kind of construct is not in particular very fault-tolerant, and overall it is very static a structure. With PMI, this is strikingly true, since cross-certification is not possible; for a particular attribute, there is only one place to look for the ACs. Thus access control cannot be totally distributed.

  The underlying assumption in PKIX is that a certificate is stored in a repository, where it is fetched on demand.

- Globally unique names

  Trying to find a certificate based on a globally unique name is not always easy, when only part of the name is known. This kind of procedure is also prone to errors, when there is a human involved in the process. (e.g, an Attempt to find Alice Smith from the United States would likely result in hundreds of results). This somewhat prevents the use of directory in itself for an attempt to find a PKC to be used for secure communication.

6.1 Benefits of PKIX

A centralized hierarchy may well be desirable in some applications: when the authority really is an *authority*, such as the government, this kind of construct might well provide a basis for secure electronical commerce and public services, when it comes to authentication of persons, or other entities that have a "natural" name, such as a DNS name.

The Finnish "Electronic Identification" (FINEID, in Finnish "Henkilön Sähköinen Tunnistaminen", HST) is one of the first attempts to establish a widely accepted authority that could be used for public services [20].

This kind of construct doesn't necessary extend well to authorization and related tasks, because it would be desirable that any entity having objects in control would be able to issue authority certificates itself, which is made impossible by the PKIX AC draft with the requirement that there be only a single Attribute Authority for each disjoint set of attributes for an entity in the network. The exclusion of delegation in the draft speaks for this as well.

6.2 Benefits of SPKI

When the usage of the PKI is authorization, then SPKI is conceptually a better choice, because not having to care about names at all simplifies the process of certification and validation remarkably. When there is only one TTP to trust, there is less components that may break up.

Other than that, use of SDSI and local namespaces in a global context would eliminate the problem of having to have a single global namespace. There is one such namespace in today's Internet: the Domain Name System. But it can't be used for everything, and such global names are really not meaningful to a human: SDSI names are.

6.3 Conclusion

It would be safe to say that PKIX is the choice when the application area is biased towards humans, especially in case of public services and such, and when authorization issues are directly related to these. Then, one has to cope with human erraneously selecting a wrong name, only paying attention to a part of the global name that has significance to that person.

When it comes to plain anonymous authorization, especially when the entities are not controlled by humans (being e.g. software agents or so), and especially in case the structure

of the PKI is very dynamic, SPKI certificates would be better choice. The possibility to allow several paths for certificate validation, and having a control on the acceptable level of successively validated paths makes SPKI favorable.

SPKI has been designed in such a way that interoperability with other PKIs has been taken into account (the 5- and 4-tuples, see [12]). The authors have obviously realised that the world won't switch from a PKI to another overnight; and it seems that the world really hasn't.

No matter which PKI becomes dominant, all the hassle about PKIs is basically about moving difficult things from one domain to another. For example, in basic access control, instead of authenticating the user with username and password safely and deciding the authorization based on the ACLs, in the world of PKI (that supports authorization) one has a complex infrastructure that involves issuing and exchange of certificates (either SPKI or PKIX), trusted third parties, public keys of TTPs that have possibly been retrieved out-of-band, and other such components. It is hard to say which one is less secure: keeping ACLs consistent and trusting in the present authentication (which might even include exchange of passwords in cleartext), or trusting in the PKI to function as claimed.

For one, industry seems to have adopted the PKIX-profiled X.509 PKI.

References

[1] IETF Authentication, Authorization and Accounting (AAA) Working Group. *IETF Working Group* <http://www.ietf.org/html.charters/aaa-charter.html>

[2] Carlisle Adams and Stephen Farrell. Internet X.509 Public Key Infrastructure Certificate Management Protocols. *RFC 2510, IETF Network Working Group*, March 1999. <http://www.ietf.org/rfc/rfc2510.txt>

[3] Alfred Arsenault and S. Turner. Internet X.509 Public Key Infrastructure PKIX Roadmap *Work in Progress. Internet-draft 05*, March 2000. [Referred 2 October 2000]. <http://www.ietf.org/internet-drafts/draft-ietf-pkix-roadmap-05.txt>

[4] Sharon Boyen, Tim Howes and Patrick Richard. Internet X.509 Public Key Certificate Operational Protocols - LDAPv2. *RFC2559, IETF Network Working Group*, April 1999. <http://www.ietf.org/rfc/rfc2559.txt>

[5] William Burr, Donna Dodson, Noel Nazario, W. Timothy Polk. MISPC Minimum Interoperability Specification for PKI Components, Version 1 Computer Science Resource Center, NIST September, 1997 <http://csrc.nist.gov/pki/mispc/welcome.html>

[6] David Chadwick. Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv3 *Work in Progress. Internet-draft 03*, September 2000. [Referred 2 October 2000]. <http://www.ietf.org/internet-drafts/draft-ietf-pkix-ldap-v3-03.txt>

[7] Santosh Chokhani. Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practises Framework *RFC 2527, IETF Network Working Group*, March 1999. <http://www.ietf.org/rfc/rfc2527.txt>

[8] Tim Dierks and Christopher Allen. The TLS Protocol Version 1.0 *RFC 2246, IETF Network Working Group*, January 1999. <http://www.ietf.org/rfc/rfc2246.txt>

[9] Whitfield Diffie and Martin Hellman New Directions in Cryptography *IEEE Transactions on Information Theory, pages 644-654*, November 1976

[10] Steve Dousse, Paul Hoffman, Blake Ramsdell, Laurence Lundblade and Lisa Repka. S/MIME Version 2 Message Specification *RFC 2311, IETF Network Working Group*, March 1998

[11] Carl M. Ellison, Bill Frantz, Butler Lampson, Ron Rivest, Brian Thomas and Tatu Ylönen. Simple Public Key Certificate *Expired Internet-draft 6*, July 1999. <http://world.std.com/ cme/spki.txt>

[12] Carl M. Ellison, Bill Frantz, Butler Lampson, Ron Rivest, Brian Thomas and Tatu Ylönen. SPKI Certificate Theory *RFC 2693, IETF Network working group*, September 1999. <http://www.ietf.org/rfc/rfc2693.txt>

[13] Carl M. Ellison. The nature of a useable PKI In *Computer Networks 31, pp. 823-830*, April 1999.

[14] Carl M. Ellison Carl M. Ellison's SPKI home page. Last Modified September 29th 2000. <http://world.std.com/ cme/html/spki.html>

[15] Stephen Farrell and Russell Housley. An Internet Attribute Certificate Profile for Authorization. *Work in Progress. Internet-draft 05*, August 2000 [Referred 2 October 2000]. <http://www.ietf.org/internet-drafts/draft-ietf-pkix-ac509prof-05.txt>

[16] Limited Attribute Certificate Acquisition Protocol. *Work in Progress. expired Internet-draft 01*, July 2000 [Referred 2 October 2000]. <http://www.ietf.org/internet-drafts/draft-ietf-pkix-laap-01.txt>

[17] Dieter Gollman. *Computer Security.* John Wiley and Sons, 1999.

[18] Russell Housley, Warwick Ford, Tim Polk and David Solo. Internet X.509 Public Key Infrastructure Certificate and CRL Profile. *RFC 2459, IETF PKIX Working Group*, January 1999. <http://www.ietf.org/rfc/rfc2459.txt>

[19] Russell Housley. Cryptographic Message Syntax *RFC 2630, IETF Network Working Group*, June 1999. <http://www.ietf.org/rfc/rfc2630.txt>

[20] Väestörekisterikeskus. (Population Register Centre) Henkilön Sähköinen Tunnistaminen (Electronic Identification, FINEID) web pages. <http://www.vaestorekisterikeskus.fi/hstetusivu2.htm> (Finnish) <http://www.vaestorekisterikeskus.fi/fineid.htm> (English)

[21] IETF IP Security Protocol (IPSEC) Working Group. *IETF Working Group* [Referred 11 November 2000] <http://www.ietf.org/html.charters/ipsec-charter.html>

[22] ISO/IEC 9594-8. X.509 draft amendment directory 4th edition. <ftp://ftp.bull.com/pub/OSIdirectory/4thEditionTexts/>

[23] John Linn and Magnus Nyström. Attribute Certification: An Enabling Technology for Delegation and Role-Based Controls in Distributed Environments. In *Proc. 1999 ACM workshop on role-based access control,* Fairfax, VA USA, October 28-29, 1999.

[24] Pekka Nikander. An Architecture for Authorization and Delegation in Distributed Object-Oriented Agent Systems. Doctoral Dissertation. <http://www.tml.hut.fi/ pnr/publications/PhDThesis.pdf>

[25] Pekka Nikander and Kristiina Karvonen. Users and Trust in Cyberspace. In *Cambridge Security Protocols Workshop 2000* Cambridge, United Kingdom, April 3-5, 2000. <http://www.tml.hut.fi/ pnr/publications/cam2000.pdf>

[26] Juha Pääjärvi. XML Encoding of SPKI Certificates. *Work in Progress. expired Internet-draft 00,* March 2000 <http://search.ietf.org/internet-drafts/draft-paajarvi-xml-spki-cert-00.txt>

[27] IETF PKIX Working Group mailing list archive. *IETF Working Group Mailing list archive* [Referred 11 November 2000] <http://www.imc.org/ietf-pkix/>

[28] IETF S/MIME Mail Security (SMIME) Working Group. *IETF Working Group* [Referred 11 November 2000] <http://www.ietf.org/html.charters/smime-charter.html>

[29] IETF Transport Layer Security (TLS) Working Group. *IETF Working Group* [Referred 11 November 2000] <http://www.ietf.org/html.charters/tls-charter.html>

[30] Symeon (Simos) Xenitellis. The Open-source PKI Book A guide to PKIs and Open-source Implementations OpenCA Team Symeon Xenitellis, 1999, 2000 <http://ospkibook.sourceforge.net/docs/OSPKI-2.4.7/OSPKI-html/ospki-book.htm>