Abstract

This document describes one certificate extension to explicitly state the Attribute Certificate (AC) policies that apply to a given Attribute Certificate. The goal of this document is to allow relying parties to perform an additional test when validating an AC, i.e. to assess whether a given AC carrying some attributes can be accepted on the basis of references to one or more specific AC policies.

Conventions Used In This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119].

1. Introduction

When issuing a PKC, a Certificate Authority (CA) can perform various levels of verification with regard to the subject identity. A CA makes its verification procedures, as well as other operational rules it abides by, "visible" through a certificate policy, which may be referenced by a certificate policies extension in the PKC.

The purpose of this document is to define an AC policies extension able to explicitly state the AC policies that apply to a given AC, but not the AC policies themselves.
2. AC Policies Extension Semantics

Attribute Certificates are defined in [RFC3281].

An Attribute Certificate Policy (ACP) is a set of rules that indicates generic rules for registering, verifying, delivering and revoking the attributes contained in a particular Attribute Certificate.

It should thus be noticed that an AA does not necessarily support one single policy. However, for each AC that is delivered it SHALL make sure that the policy applies to all the attributes that are contained in it.

An Attribute Certificate Policy may be used by a certificate user to decide whether or not to trust the attributes contained in a certificate for a particular purpose.

When a certificate contains an AC policies extension, the extension MAY, at the option of the attribute certificate issuer, be either critical or non-critical.

The AC Policies extension MAY be included in an attribute certificate. Like all X.509 certificate extensions [X.509], the AC policies extension is defined using ASN.1 [ASN1].

The AC policies extension is identified by id-pe-acPolicies.

```
id-pe-acPolicies OBJECT IDENTIFIER ::= { id-pe 15 }
```

The AC policies extension includes a list of AC policies recognized by the issuing authority that apply to the attributes included in the attribute certificate.

AC Policies may be defined by any organization with a need. Object identifiers used to identify AC Policies are assigned in accordance with [ITU-T Rec. X660 | ISO/IEC 9834-1].

The presence of this extension in an attribute certificate indicates the AC policies for which the attribute certificate is valid.

An application that recognizes this extension and its content SHALL process the extension regardless of the value of the criticality flag.

If the extension is both flagged non-critical and is not recognized, then the application MAY ignore it.

If the extension is flagged critical or is recognized, it indicates that the attributes contained in the attribute certificate SHALL only be used for the purpose, and in accordance with the rules implied by one of the indicated AC policies. If none of the AC policy identifiers is adequate for the application, then the AC MUST be rejected.
users MUST use the list of AC policies to determine whether it is appropriate to use the attributes contained in that certificate for a particular transaction.

2.1 AC Policy Extension Syntax

The syntax for the AC Policy extension is:

\[
\text{AcPoliciesSyntax ::= SEQUENCE SIZE (1..MAX) OF PolicyInformation}
\]

\[
\text{PolicyInformation ::= SEQUENCE {}
  \text{  policyIdentifier AcPolicyId,}
  \text{  policyQualifiers SEQUENCE SIZE (1..MAX) OF PolicyQualifierInfo OPTIONAL}}
\]

\[
\text{AcPolicyId ::= OBJECT IDENTIFIER}
\]

\[
\text{PolicyQualifierInfo ::= SEQUENCE {}
  \text{  policyQualifierId PolicyQualifierId,}
  \text{  qualifier ANY DEFINED BY policyQualifierId}}
\]

-- policyQualifierIds for Internet policy qualifiers

\[
\text{id-qt OBJECT IDENTIFIER ::= \{ id-pkix 2 \}}
\]

\[
\text{id-qt-acps OBJECT IDENTIFIER ::= \{ id-qt 4 \}}
\]

\[
\text{id-qt-acunotice OBJECT IDENTIFIER ::= \{ id-qt 5 \}}
\]

\[
\text{PolicyQualifierId ::= OBJECT IDENTIFIER \{ id-qt-acps | id-qt-acunotice \}}
\]

-- ACPS pointer qualifier

\[
\text{ACPSuri ::= IA5String}
\]

-- AC user notice qualifier

\[
\text{ACUserNotice ::= UserNotice}
\]

-- UserNotice is defined in [RFC3280]

To promote interoperability, this document RECOMMENDS that policy information terms consist of only an OID. When more than one policy is used, the policy requirements have to be non conflicting, e.g. one policy may refine the general requirements mandated by another policy.

When qualifiers are used with the special policy anyPolicy, they MUST be limited to the qualifiers identified in this section.

This specification defines two policy qualifier types for use by attribute certificate policy writers and attribute certificate issuers. The qualifier types are the ACPS Pointer and AC User Notice qualifiers.
The pointer is in the form of a URI. Processing requirements for this qualifier are a local matter.

The AC User notice is intended for display to a relying party when an attribute certificate is used. The application software SHOULD display the AC user notice of the attribute certificate. The AC user notice is defined in [RFC3280]. It has two optional fields: the noticeRef field and the explicitText field.

The noticeRef field, if used, names an organization and identifies, by number, a particular textual statement prepared by that organization. For example, it might identify the organization’s name and notice number 1. In a typical implementation, the application software will have a notice file containing the current set of notices for the AA; the application will extract the notice text from the file and display it. Messages MAY be multilingual, allowing the software to select the particular language message for its own environment.

An explicitText field includes the textual statement directly in the certificate. The explicitText field is a string with a maximum size of 200 characters.

If both the noticeRef and explicitText options are included in the one qualifier and if the application software can locate the notice text indicated by the noticeRef option, then that text SHOULD be displayed; otherwise, the explicitText string SHOULD be displayed.

2.2 Attribute Certificate Policies

The scope of this document is not the definition of the detailed content of Attribute Certificate policies themselves, therefore specific policies are not defined in this document.

3. Security Considerations

The Attribute Certification Policy defined in this document applies for all the attributes that are included in one AC. AAs shall make sure that the Attribute Certification Policy applies to all the attributes which are included in the attribute certificates they issue.

Attributes may be dynamically grouped in several ACs. It should be observed that since the management of some attributes may be different, different policies may be used by the same AA.

When verifying an Attribute Certificate, a relying party must determine first that the AC was issued by a trusted AA and then has the appropriate policy.
will undermine confidence in the system. If the compromise is detected, all ACs issued by the AC issuer MUST be revoked. Rebuilding after such a compromise will be problematic, so AC issuers are advised to implement a combination of strong technical measures (e.g., tamper-resistant cryptographic modules) and appropriate management procedures (e.g., separation of duties) to avoid such an incident.

Loss of an AC issuer’s private signing key may also be problematic. The AC issuer would not be able to produce revocation status or perform AC renewal. AC issuers are advised to maintain secure backup for signing keys. The security of the key backup procedures is a critical factor in avoiding key compromise.

The availability and freshness of revocation status will affect the degree of assurance that should be placed in a long-lived AC. While long-lived ACs expire naturally, events may occur during its natural lifetime which negate the binding between the AC holder and the attributes. If revocation status is untimely or unavailable, the assurance associated with the binding is clearly reduced.

The binding between an AC holder and attributes cannot be stronger than the cryptographic module implementation and algorithms used to generate the signature. Short key lengths or weak hash algorithms will limit the utility of an AC. AC issuers are encouraged to note advances in cryptology so they can employ strong cryptographic techniques.

If an attribute certificate is tied to the holder’s PKC using the baseCertificateID component of the Holder field and the PKI in use includes a rogue CA with the same issuer name specified in the baseCertificateID component, this rogue CA could issue a PKC to a malicious party, using the same issuer name and serial number as the proper holder’s PKC. Then the malicious party could use this PKC in conjunction with the AC. This scenario SHOULD be avoided by properly managing and configuring the PKI so that there cannot be two CAs with the same name. Another alternative is to tie ACs to PKCs using the publicKeyCert type in the ObjectDigestInfo field. Failing this, AC verifiers have to establish (using other means) that the potential collisions cannot actually occur, for example, the CPSs of the CAs involved may make it clear that no such name collisions can occur.

Implementers MUST ensure that following validation of an AC, only attributes that the issuer is trusted to issue are used in authorization decisions. Other attributes, which MAY be present MUST be ignored. Given that the AA controls PKC extension is optional to implement, AC verifiers MUST be provided with this information by other means. Configuration information is a likely alternative means. This becomes very important if an AC verifier trusts more than one AC issuer.

Francis, Pinkas

4. References

4.1 Normative references
4.2 Informative reference


5. IPR Notice

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF’s procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.
AcPolicies { iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) pkix(7) id-mod(0)
id-mod-ac-policies(26) }

DEFINITIONS IMPLICIT TAGS ::= 

BEGIN

-- EXPORTS ALL --

IMPORTS

-- Imports from RFC 3280 [RFC3280], Appendix A.1

UserNotice, anyPolicy
FROM PKIX1Implicit88 { iso(1) identified-organization(3)
dod(6) internet(1) security(5) mechanisms(5) pkix(7)
id-mod(0) id-pkixl-implicit(19) }

id-pkix, id-pe
FROM PKIX1Explicit88 { iso(1) identified-organization(3)
dod(6) internet(1) security(5) mechanisms(5) pkix(7)
id-mod(0) id-pkixl-explicit(18) };

-- Locally defined OIDs

-- policyQualifierIds for Internet policy qualifiers

id-qt          OBJECT IDENTIFIER ::= { id-pkix 2 }
id-qt-acps     OBJECT IDENTIFIER ::= { id-qt 4 }
id-qt-acunotice OBJECT IDENTIFIER ::= { id-qt 5 }

-- Attributes

id-pe-acPolicies OBJECT IDENTIFIER ::= { id-pe 15 }

AcPoliciesSyntax ::= SEQUENCE SIZE (1..MAX) OF PolicyInformation

PolicyInformation ::= SEQUENCE {
  policyIdentifier    AcPolicyId,
  policyQualifiers    SEQUENCE SIZE (1..MAX) OF
                        PolicyQualifierInfo OPTIONAL}

AcPolicyId ::= OBJECT IDENTIFIER

PolicyQualifierInfo ::= SEQUENCE {
  policyQualifierId    PolicyQualifierId,
  qualifier            ANY DEFINED BY policyQualifierId }

Francis, Pinkas

Internet-Draft            AC Policies extension           December 2003

PolicyQualifierId ::= 
  OBJECT IDENTIFIER { id-qt-acps | id-qt-acunotice }

-- ACPS pointer qualifier

ACPSuri ::= IA5String
AC user notice qualifier

ACUserNotice ::= UserNotice
-- UserNotice is defined in [RFC3280]

END