PKIX Working Group
Internet Draft
Expires September 2002

P. Yee
RSA Security

March 2002

Attribute Certificate Management Messages over CMS
<draft-ietf-pkix-acmc-01.txt>

Status of this Memo

Abstract

This document specifies modifications to the Certificate Management Messages over CMS specification ([CMCbis]) to permit the management of attribute certificates. This document does not stand alone, but must be used in conjunction with [CMCbis]. It is expected that the modifications proposed here will also be used in conjunction with the Attribute Certificate Request Message Format specification ([ACRMF]).

1. Introduction

The key words "MUST", "REQUIRED", "SHOULD", "RECOMMENDED", and "MAY" in this document are to be interpreted as described in [RFC2119].

CMC [CMCbis] specifies the exchanges, structures, and controls for managing public key certificates. This document extends CMC to handle attribute certificates. It profiles CMC, adding new elements as necessary.

CMC supports "Simple PKI Requests" and "Full PKI Requests". The ACMC

specification requires the use of the Full PKI Request form and its corresponding response.

2. Request Modifications

The two primary data structures in CMC are the PKIData content object and the ResponseBody content object. Based on the name of the PKIData structure, one might not think that it is appropriate for attribute certificates; however its content is well-aligned with our purpose. In particular, no changes are required at the top level of either PKIData or ResponseBody.

Within the PKIData structure, the reqSequence (a sequence of TaggedRequest) element is modified in order to carry the ACRMF and other requests. Thus, TaggedRequest becomes:

```
TaggedRequest ::= CHOICE {
    tcr               [0] TaggedCertificationRequest, -- original
    crm               [1] CertRequestMsg,             -- original
    other             [2] ANY DEFINED BY OID -- others including ACRMF
```

Implementations MAY allow requests for both public key and attribute certificates in a single reqSequence.

3. Control Attribute Modifications

CMC specifies a large number of control attributes that can be applied as part of certificate requests. Many of these are inappropriate for attribute certificates. In particular, ACMC only uses the following controls:

Control Attribute	OID	Syntax
dataReturn	id-cmc 4	OCTET STRING
transactionID	id-cmc 5	INTEGER
senderNonce	id-cmc 6	OCTET STRING
recipientNonce	id-cmc 7	OCTET STRING
addExtensions	id-cmc 8	AddExtensions
getCert	id-cmc 15	GetCert
getCRL	id-cmc 16	GetCRL
revokeRequest	id-cmc 17	RevokeRequest
regInfo	id-cmc 18	OCTET STRING

responseInfo	id-cmc 19	OCTET STRING
queryPending	id-cmc 21	OCTET STRING
idConfirmCertAcceptance	id-cmc 24	CMCCertId
cmcStatusInfoExt	id-cmc XX	CMCStatusInfoExt

Additional control attributes are defined: addAttribute, sendTo, and modHandling discussed later.

Control Attribute	OID	Syntax
addAttribute	id-cmc <acmc01>	AddAttribute
sendTo	id-cmc <acmc02>	SendTo
attrModHandling	id-cmc <acmc03>	AttrModHandling

It is possible that a control attribute to support additional retrieval indices for attribute certificates will be added if getCert cannot be suitably modified.

3.1. Data Return Control Attribute

dataReturn, [CMCbis] Section 5.4, is supported without modification by ACMC.

3.2. Transaction ID Control Attribute

transactionID, [CMCbis] Section 5.6, is supported without modification by ACMC.

3.3. Sender Nonce Control Attribute

senderNonce, [CMCbis] Section 5.6, is supported without modification by ACMC.

3.4. Recipient Nonce Control Attribute

recipientNonce, [CMCbis] Section 5.6, is supported without modification by ACMC.

3.5. Add Extensions Control Attribute

The addExtensions control attribute, [CMCbis] Section 5.5, is supported by ACMC. In order to match [ACRMF] messages, the certReferences sequence is additionally allowed to be equal to the attrCertReqId of the AttrCertRequest within an AttrCertReqMsg (see ACRMF, Section 3). Also, when the extensions are being applied to an attribute certificate, the requirement shall be that servers MUST be able to process all extensions defined in [ACPROF].

3.6. Get Certificate Control Attribute

ACMC supports the getCert control attribute ([CMCbis] Section 5.9). Currently, getCert only supports retrieval based upon the issuerName and serialNumber combination. This combination of values suffices for both public key and attribute certificates.

Additional retrieval scenarios are envisaged, as expressed in [CERTHTTP]. Beyond that, attribute certificates have other means by which they can be indexed and retrieved. In particular, retrieval by holder name in conjunction with a particular set of attribute types would be useful.

3.7. Get CRL Control Attribute

The getCRL control attribute ([CMCbis] Section 5.10) is supported as is by ACMC.

3.8. Revoke Request Control Attribute

The revokeRequest control attribute ([CMCbis] Section 5.11)is supported as specified in CMC. Some of the CRLReason codes used, however, are not suitable for use with attribute certificates. In particular, only the unspecified, affliationChanged, superseded, cessationOfOperation, privilegeWithdrawn, or aACompromise values should be used when revoking an attribute certificate.

More generally, this control attribute is not appropriate to employ if the noRevAvail extension is present in the attribute certificate and its value is set to TRUE.

The protocol does not specify which entities are allowed to request the revocation of a certificate.

The revoke request control attribute allows revocation of a public key certificate without having a signature on the request. A password is used for authentication in this case. For attribute certificates, this capability is not supported. If the private signing key is lost, then the public key certificate should be revoked. Attribute certificates that are explicitly linked to the public key certificate being revoked will simply fail to verify.

3.9. Registration Information Control Attribute

The regInfo control attribute ([CMCbis] Section 5.12) is supported as

specified in CMC.

3.10. Response Information Control Attribute

The responseInfo control attribute ([CMCbis] Section 5.12) is supported as specified in CMC.

3.11. Query Pending Control Attribute

The queryPending control attribute ([CMCbis] Section 5.13) is supported as specified in CMC.

3.12. Confirm Certificate Acceptance Control Attribute

The idConfirmCertAcceptance control attribute ([CMCbis] Section 5.14) is supported as specified in CMC.

4. New Control Attributes

4.1. Add Attributes Control Attribute

The addAttributes control attribute is analogous to the addExtensions control attribute.

The Add Attributes control attribute is used by LARAs to specify additional attributes that are to be placed on certificates. This attribute uses the following ASN.1 definition:

```
AddAttributes ::= SEQUENCE

    pkiDataReference            BodyPartID
    certReferences              SEQUENCE OF BodyPartID,
    attributes                  SEQUENCE OF Attribute
}
```

-- pkiDataReference field contains the body part identifier of the embedded request message.

-- certReferences field is a list of references to one or more of the payloads contained within a PKIData. Each element of the certReferences sequence MUST be equal to either the bodyPartID of a TaggedCertificationRequest, the certReqId of the CertRequest within a CertReqMsg, or the attrCertReqId of the AttrCertRequest within an AttrCertReqMsg. By definition, the listed attributes are to be

applied to every element referenced in the certReferences sequence. If a request corresponding to bodyPartID cannot be found, the error badRequest is returned referencing this control attribute.

-- attributes field contains the sequence of attributes to be applied to the referenced certificate requests.

Servers MUST be able to process all attributes defined in [ACPROF]. Servers are not required to be able to process every attribute transmitted using this protocol. Servers are not required to put all LARA-requested attributes into a certificate. Servers are permitted to modify LARA-requested attributes. Servers MUST NOT alter an attribute so as to reverse the meaning of a client-requested attribute. If a certification request is denied due to the inability to handle a requested attribute and a response is returned, the

server MUST return a failInfo attribute with the value of unsupportedAttr.

If multiple Add Attributes statements exist in an enrollment message, the exact behavior is left up to the certificate issuer policy. However it is recommended that the following policy be used. These rules would be applied to individual attribute within an Add Attributes control attribute (as opposed to an "all or nothing" approach).

1. If the conflict is within a single PKIData object, the certificate request would be rejected with an error of badRequest.

2. If the conflict is between different PKIData objects, the outermost version of the attribute would be used (allowing a LARA to override the attribute requested by the end-entity). If the attributes requested by an end-entity are overridden, then the returned status SHALL so indicate (see Section 5).

4.2. Send To Control Attribute

The Send To Control Attribute indicates to the Attribute Authority that a copy of the generated attribute certificate should be sent to the designated recipient. Such a service is useful in cases when the entity for whom the attribute certificate is issued is not the requester.

```
SendTo ::=   GeneralNames
```

GeneralNames is used to specify the recipients of the generated attribute certificate. Note that some forms of GeneralName are not appropriate for receiving attribute certificates without further

specification.

4.3. Attribute Modification Handling Control Attribute

The Attribute Modification Handling Control Attribute allows the requester to specify its permissions for cases where the LARA wishes to change the requested set attributes or their values, or where the

Attribute Authority wishes to issue a set of attributes which differ from those requested. Permissions that may be specified are:

- Attributes to be issued must be exactly as specified (or not at all).
- Attributes to be issued must be according to given profile or policy.
- Attributes types must be as requested, but values may differ (across any subset of attributes).
- Any attributes and values are acceptable.

```
attrModHandling ::= SEQUENCE {
  attrModPermission   AttrModPermission,
  attrModPolicy       OBJECT IDENTIFIER
}

AttrModPermission ::= INTEGER {
  asSpecified         (0),
  byPolicy            (1),
  byType              (2),
  atAADiscretion      (3)
}
```

The Modification Handling control supercedes the Add Attributes control and cannot be further superceded by another instance of this control. If more than one instance of the control appears in a single request, a badRequest CMCFailInfo value MUST be returned to the LARA or end-entity.

When attributes are to be issued according to a given profile or policy, the requester MAY send requested attributes and their value or omit them. If values are supplied, the AA may modify these values within the bounds of the policy. If the attributes are omitted in the request, the AA supplies a permissible set of attributes and values as dictated by the policy.

5. Status Modifications

cMCStatusInfoExt is used to indicate that a request was unsuccessful.

ACMC returns additional status values beyond those specified in [CMC]

Section 5.1.4. The additional status value are encoded using the ExtendedFailInfo field of the cmcStatusInfoExt structure. These relevant values are defined as:

```
  id-cet-acmcFailInfo OBJECT IDENTIFIER ::= { iso(1) identified-organization
    dod(6) internet(1) security(5) mechanisms(5) pkix(7) cet(15) acmcFailInf

  ACMCFailInfo ::= INTEGER {
    unsupportedAttr        (0),
    attrModified           (1),
    policyDoesNotAllow     (2),
    comboNotSupported      (3) }
```

The ACMCFailInfo values mean:

- unsupportedAttr means that the requested attribute was not supported by the recipient AA.

- attrModified indicates that the set of attributes or the attribute values were modified by the AA. This return value is not explicitly fatal, but is meant to alert the requester that one or more modifications were made in the returned attributes. If the Attribute Modification Control is used to signal that attributes are to be set by policy, than this return value MAY be omitted.

- policyDoesNotAllow signals that the prevailing policy under which the attribute certificate is to be issued does not allow the granting of a requested attribute or attribute value; this error value is used in response to the addAttribute control.

- comboNotSupported means that this responder does not support requests for both public key and attribute certificates in one message.

6. Additional Notes

In the Full PKI Response generated when a new attribute certificate is requested, this profile requires that the certificates field of the signedData object MUST contain (at a minimum) the AA's PKC. Other certificates that form the certificate chain for the AA's PKC MAY be included in the certificates field.

Security considerations are not yet discussed in this memo.

7. References

[2459bis] Housley, R., W. Ford, W. Polk, and D. Solo. Work in progress, October 2001. "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", draft-ietf-pkix-new-part1-11.txt.

[ACPROF] Farrell, S. and R. Housley. Work in progress, June 8, 2001. "An Internet Atribute Certificate Profile for Authorization", draft-ietf-pkix-ac509prof-09.txt.

[ACRMF] Yee, P. Work in progress, November 2001. "Attribute Certificate Request Message Format", draft-ietf-pkix-acrmf-00.txt.

[CERTHTTP] Gutmann, P. January 21, 2002. "Certificate Store Access via HTTP", draft-ietf-pkix-certstore-http-02.txt.

[CMCbis] Myers, M., X. Liu, J. Schaad, and J. Weinstein. Work in progress, July 2001. "Certificate Management Messages over CMS", draft-ietf-pkix-rfc2797-bis-01.txt.

[RFC2026] Bradner, S. October 1996. "The Internet Standards Process -- Revision 3", RFC 2026, BCP 9.

[RFC2119] Bradner, S. March 1997. "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, BCP 14.

Appendix A: Object Identifiers

[OIDs go here.]

Appendix B: ASN.1 Module

[ASN.1 goes here.]

Author's Address:

Peter Yee
RSA Security
2955 Campus Drive
Suite 400
San Mateo, California 94403
USA

email: pyee@rsasecurity.com

Full Copyright Statement