INTERNET-DRAFT
Expires in six months

S. Farrell
Baltimore Technologies
David Chadwick
University of Salford
14 July 2000

Limited AttributeCertificate Acquisition Protocol
<draft-ietf-pkix-laap-02.txt>

Status of this Memo

Abstract

The PKIX working group is profiling the use of X.509 attribute certificates. This document specifies a deliberately limited protocol for requesting an attribute certificate from a server. It is intended to be complementary to the use of LDAP for AC retrieval, covering, for example, those cases where use of an LDAP server is not suitable due to the type of authorization model being employed. For many other cases, the use of LDAP is preferred.

Table Of Contents

1. Introduction

<<Comments are in angle brackets like this.>>

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

[ACPROF] specifies the Internet profile of the X.509 attribute certificate (AC) for authorization purposes. This document specifies a deliberately limited protocol for requesting such an AC from a server.

There is clearly a requirement for an AC management protocol (or protocols, like [CMP] and [CMC]). Such management protocols are not specified in this document. There is also a requirement for a specification of an LDAP schema to allow retrieval of ACs from LDAP servers, which is specified in [LDAP-SCHEMA].

In addition to such protocols, which may be more suited to management of long-term or more sensitive (i.e. more "powerful") ACs, there is a requirement for a very simple, explicitly limited AC acquisition protocol. We call this protocol the Limited AC Acquisition Protocol (LAAP).

LAAP consists of a simple request/response protocol encapsulated in [CMP] messages. The entity which issues the request is called the LAAP requestor (LRQ), the entity which issues the response is called the LAAP responder (LRP). The LRQ is typically an AC holder or an AC verifier; the LRP is typically the AC issuer itself.

The situations in which LAAP may be more suitable for use than LDAP include:

- where ACs are very short lived and the latency involved in writing to the LDAP servers is relatively long (e.g. if a complex directory deployment is behind the LDAP server),

- where a least privilege style of AC use is required, the LRP can modulate the AC content based on the context of the LAAP request, for example, if the LRQ is authenticated and the LRP is the AC issuer, then the LRP may choose to include only the minimal set of attributes (administered to be) required by that LRQ,
- where there are potentially numerous ACs, many of which are never actually used during their lifetime (in which case they should only be generated if needed) e.g. many entities have permission to access some data, but only a subset of them actually do access it,
- where ACs contain encrypted attributes and it may not be possible to search the LDAP directory for ACs with attributes of a specific type.

LAAP is only intended to be used for cases where an LRQ wishes to acquire a "current" AC for an entity (possibly itself) leaving almost all details as to the content of the AC to the LRP.

2. LAAP

The LAAP protocol consists of two new message types which are embedded in the PKI Message Body defined in [CMP]. One message (the LAAP Request) is embedded in the GenMsgContent field, the other (the LAAP Response) is embedded in the GenRepContent field. Future specifications MAY enhance the request and/or response types defined here - any such enhancement MUST use a different object identifier to identify the GenMsgContent or GenRepContent.

The one and only feature of this protocol is to request an AC for a particular entity that may be either the LRQ or some other entity. The response is the requested AC or an error.

2.1 Message Types

2.1.1 LAAP Request Message

The request MAY specify the identity of the AC holder (for the third party case), with an optional "profile". A profile is to be interpreted as a bilaterally agreed string, or OID, that may be mapped to a set of AC contents by the LRP. In the third party case, the LRQ, MAY also include some evidence that the AC holder has requested the LRQ to retrieve an AC belonging to the AC holder.

```
LACRequestMessage ::= SEQUENCE {
     holder    Holder OPTIONAL,
     profile   [0] SEQUENCE {
          string          UTF8String,
          oid             OBJECT IDENTIFIER
          } OPTIONAL,
     issuer    [1] AttCertIssuer OPTIONAL,
     acOptions ACOptions OPTIONAL
```

```
    }
```



```
    ACOptions ::= BIT STRING {
      attr-encryption (0),
      proxying        (1),
      object-digests  (2),
      aa-controls     (3)
    }
```

Each field is described below.

"holder": when present this specifies that the LRQ wishes to acquire an AC for this holder. When absent, it means that the LRQ is requesting an AC for itself (the LRP SHOULD use the identity established from whatever authentication is available). The rules for the holder field specified in [ACPROF] apply here (e.g. constrained use of entityName).

"profile": when present this is signals that an AC matching the supplied profile is returned. The definition of profiles is not in scope for this specification and is expected to be a local matter. There are two main uses envisaged for this field:

- Where an LRQ requests its own AC, then the profile field can be used for those entities which require a non-default AC. The typical case here is where a user requests her AC in order to "push" it to a relying party via some protocol (like CMS). In most such cases, the user can use a default AC whose content has been selected for her by an administrator. Occasionally, such users will require a different AC, perhaps for use in some application environment that is seldom used. In such cases the profile field can contain a value provided to the user by the AA administrator. It is often the case that a profile maps well to a role in this scenario.

- When a relying party requests an AC for another entity it needs the AC to contain a set of attributes which will enable the relying party to make a "good" authorization decision. In most such cases, the identity of the relying party will determine (for the AA) the set of attribute types required. However, in cases where the identity of the relying party is not known, or where a single relying party makes "different" types of authorization decision, (say where two applications run from a single account), then the profile allows the relying party to specify which "type" of authorization decision it wishes to make. It is often the case that the profile maps well to an application or function in this scenario.

Where it is desirable that the profile contain a globally unique value, the profile SHOULD use the oid choice. Conformant implementations MUST be able to handle both string and OID profiles.

One possible implementation model which can usefully use the OID choice is for the profile to contain the OID of an LDAP or X.500 object class ([X.501], [RFC2526]) and for the LRP to produce an AC

containing the relevant attribute values specified by that object class.

Note that in all cases where a profile is specified by an LRQ, the resulting AC may or may not meet the LRQ's expectation for ACs which "match" the requested profile. The LRQ MUST check the resulting AC, if it needs to check this "matching". Note also, that in addition to selecting the "attributes" field, an LRP MAY also use the profile to determine other AC fields, e.g. validity or extensions.

"issuer": This field allows the LRQ to specify the AC issuer in case an LRP responds on behalf of more than one AC issuer.

"acOptions": This field allows the LRQ to indicate the set of the optional features from [ACPROF] that the LRQ "supports". Each bit may be set independently by the LRQ to indicate support for one of the optional features. When this field is not present it should be interpreted that all the bits are not set, which has the following meaning:

```
Attribute encryption is not supported
Proxying is not supported
Object digests are not supported
AA controls are not supported
```

Note that if all the optional fields are missing, this means that the minimal LAAP request structure consists of the octets ‘3000’H, an empty ASN.1 sequence. This means "give me my current default AC please and do not use any optional features from [ACPROF]".

2.1.2 LAAP Response Message

The response message consists of an AC (errors are handled at the CMP level).

```
LACResponseMessage ::= AttributeCertificate
```

When an LRQ receives an AC from an LRP it SHOULD verify the AC. In addition the LRQ SHOULD ensure that the AC "matches" the LAAP request issued, i.e. that the holder in the AC matches that in the request (if present). Implementations may of course include additional checks.

The AC in the response MUST conform to the profile specified in [ACPROF] and MUST only make use of the optional features that the LRQ has indicated that it can support.

2.2 Encapsulation in CMP

LAAP requests and responses are carried within a PKIMessage, as defined in [CMP].

```
   PKIMessage ::= SEQUENCE {
      header           PKIHeader,
      body             PKIBody,
      protection   [0] PKIProtection OPTIONAL,
      extraCerts   [1] SEQUENCE SIZE (1..MAX)
                           OF Certificate OPTIONAL
  }
```

2.2.1 PKI Body

The GenMsgContent CHOICE of the PKIBody contains the LAAP request and the GenRepContent contains the LAAP response. Each GenMsgContent and GenRepContent consists of a SEQUENCE OF InfoTypeAndValue. InfoTypeAndValue is an OID and an ANY defined by the OID. There SHALL be only one InfoTypeAndValue for both LAAP requests and responses. Separate OIDs are defined for LAAP requests and responses as follows:

```
   id-laap-req       OBJECT IDENTIFIER ::= { id-laap 1 }
   id-laap-rep       OBJECT IDENTIFIER ::= { id-laap 2 }
```

The ANY field MUST be a LACRequestMessage for a LAAP request, and the ANY field MUST be a LACResponseMessage for a LAAP response.

Errors are handled using the ErrorMsgContent form of PKIBody.

A conformant implementation is NOT REQUIRED to be able to handle any other form of PKIBody. Of course, an LRQ or LRP MAY also handle other forms of PKIBody, e.g. the mandatory profile specified in [CMP].

2.2.2 PKI Header

The fields of the PKIHeader MUST be used as specified in section 3.1.1 of [CMP]. <<may need more specification here.>>

pvno	MUST be ietf-version 3 (2) [CMP2000]
sender	MUST adhere to the restrictions in [ACPROF]. Anonymous requests are allowed provider the holder

field is present and MUST include an empty DN for the sender field.

recipient	MAY be empty DN. Only used for CMP message authentication (e.g. D-H cases)
generalInfo	MUST be absent

All other fields MUST be as specified in [CMP2000].

2.2.3 PKI Protection

Though the PKIMessage construct supports the use of various forms of authentication, the security required for a specific LAAP request or response is not specified here. In order to provide a basic level of interoperability LRPs MUST be able to handle requests authenticated with either the PasswordBasedMac or signature methods described in [CMP] section 3.1.3. LRPs MUST also be able to handle requests which contain no PKIProtection (though they MAY always return an error).

LAAP requestors MUST implement one of PasswordBasedMac, signature or missing.

Algorithms: the defaults are as in CMP.

2.2.4 Additional Public Key Certificates

This field MAY contain a set of public key certificates that MAY be used by the recipient to assist in verification of authentication of the message sender or of an attribute certificate contained in a response.

<<Is this the right conformance specification?>>

2.3 Response Handling

If the LRP provides the AC that the LRQ requested, then a PKIStatus of "granted" is returned.

If the LRP provides an AC that is not exactly what the LRQ requested e.g. the AC grants some privileges but less than the LRQ requested, then a PKIStatus of "grantedWithMods" is returned. It is a local matter for the LRP to decide when to return granted and when to return grantedWithMods.

If the LRP does not return an AC to the LRQ, then a PKIStatus of "rejection" SHALL be returned. Section 2.5.4 describes the PKIFailureInfo that MUST accompany this status message.

Other PKIStatus values MUST NOT be returned to the LRQ.

2.4 Error Handling

If an LRP receives any CMP message which it does not support (e.g. a public key certification request), then it MUST respond with an error containing "rejection" as the PKIStatus, and "badRequest" as the PKIFailureInfo. The status string MAY contain any implementation specific value (though note that this field is intended to be human readable).

For all other error conditions the PKIStatus MUST be "rejection". If the LRP fails to authenticate the LRQ, or no or insufficient authentication information was provided, and the LRP requires

authentication, then a PKIFailureInfo of "badMessageCheck" SHALL be returned.

If the LRP does authenticate the LRQ, but the the LRQ is not authorised to receive the AC, then "notAuthorized" SHALL be returned.

If the LRP detects an incorrectly formatted LACRequestMessage then a PKIFailure of "badDataFormat" SHALL be returned.

If the LRP is unable to return an AC to the LRQ, then a PKIFailure of "badCertId" SHALL be returned.

<<The semantics of this error code are correct and meaningful (i.e. no certificate could be found matching the provided criteria). However the label "badCertId" does not seem to be so correct, since a certificate id was not provided by the LRQ. Therefore we could either define a new PKIFailure code ("noCertFound") or we could ask that the label in [CMP2000] be changed to "noCertFound", leaving the semantics as now, or that [CMP2000] change the semantics of the "badCertId" error to "incorrect certificate id provided and no certificate could be found that matches it".>>

If the LRP suspects that the LRQ has contacted the wrong Attribute Authority then a PKIFailure of "wrongAuthority" MAY be returned in addition to "badCertId".

In addition to the above, the LRP MAY return a statusString for human consumption.

3. Transport Mechanisms

LAAP can be carried via a number of transport mechanisms: either directly over TCP, or by encoding within HTTP.

Both LRQ and LRP implementations MUST support the TCP transport. Either MAY support the HTTP transport.

[CMP] already defines TCP and HTTP transports. These MAY also be used for LAAP. Some changes based on implementation experience have

been developed in [CMP-Tran]]. These changes supercede the equivalent transports defined in [CMP] and MUST be supported by compliant implementations.

LRQs and LRPs are NOT REQUIRED to support polling, as either an AC or an error is expected to be produced immediately in response to a request. This means that even if an LRP does support other forms of [CMP] requests, it cannot use the polling mechanism in response to a LAAP request.

4. Security Considerations

The LRQ MUST verify the AC using the rules given in [ACPROF] before making an authorization decision based on the AC. LAAP (like all such protocols) is vulnerable to denial-of-service attacks, this should be taken into account before deployment. If the LRP is the actual AC issuer, then it should be very careful about handing out ACs in response to unauthenticated requests. One model would be to manage the authentication "strength" required before issuing a given (type of) AC.

5. References

[ACPROF] Farrell, S., Housley, R. "An Internet AttributeCertificate Profile for Authorization", draft-ietf-pkix-acprof-04.txt, July 2000, work-in-progress..

[CMC] M. Myers, X. Liu, J.Schaad, J. Weinstein. "Certificate Management Messages over CMS". RFC 2797. April 2000.

[CMP] Adams, C., Farrell, S., "Internet X.509 Public Key Infrastructure - Certificate Management Protocols", RFC2510. March 1999

[CMP2000] Adams, C., Farrell, S., Update to "Internet X.509 Public Key Infrastructure - Certificate Management Protocols", draft-ietf-pkix-rfc2510bis-01.txt, July 1999, work-in-progress.

[CMP-Tran] Kapoor, A. and Tschalar, R. " Transport Protocols for CMP", draft-ietf-pkix-cmp-transport-protocols-00.txt, June 22 2000, work-in-progress.

[LDAP-SCHEMA] Chadwick, D., "Internet X.509 Public Key Infrastructure Operational Protocols - Additional LDAP Schema for PKIs and PMIs <draft-pkix-ldap-schema-00.txt>, July 2000

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

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

[RFC2459] Housley, R., Ford, W., Polk, T, & Solo, D., "Internet Public Key Infrastructure - X.509 Certificate and CRL

profile", RFC2459.
[RFC2252] Wahl, M., et al., " Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions", RFC2252.
[RFC2256] Wahl, M., "A Summary of the X.500(96) User Schema for use with LDAPv3", RFC2256.
[X.501] ITU-T Recommendation X.501 : Information Technology - Open Systems Interconnection - The Directory: Models, 1993.

Author's Addresses

Stephen Farrell,
Baltimore Technologies,
61/62 Fitzwilliam Lane,
Dublin 2,
IRELAND

tel: +353-1-647-3000
email: stephen.farrell@baltimore.ie

David Chadwick
IS Institute
University of Salford
Salford
England
M5 4WT

Tel: +44 161 295 5351
email: d.w.chadwick@salford.ac.uk

Full Copyright Statement

Appendix A: Object Identifiers

This section lists the object identifiers defined in this specification.

The following object identifiers are inherited from [RFC2459] and [CMP].

```
id-pkix OBJECT IDENTIFIER ::= { iso(1) identified-organization(3)
            dod(6) internet(1) security(5) mechanisms(5) pkix(7) }
   id-mod  OBJECT IDENTIFIER ::= { id-pkix 0 }
   id-it   OBJECT IDENTIFIER ::= { id-pkix 4 }
```

The following new ASN.1 module identifier is defined:

```
   id-mod-laap       OBJECT IDENTIFIER ::= { id-mod <<tbs>> }
   << probably { id-mod 13 }>>
```

The LAAP message types are defined as follows:

```
   id-laap           OBJECT IDENTIFIER ::= { id-it <<tbs>> }
   << probably { id-it 7 } >>
   id-laap-req       OBJECT IDENTIFIER ::= { id-laap 1 }
   id-laap-rep       OBJECT IDENTIFIER ::= { id-laap 2 }
```

Note: The following OID has been assigned that may be used during testing until an official pkix oid is assigned

```
     id-laap OBJECT IDENTIFIER ::= 1.2.826.0.1.3344810.5
```

Appendix B: "Compilable" ASN.1 module

```
   PKIXLaap {iso(1) identified-organization(3) dod(6)
             internet(1) security(5) mechanisms(5) pkix(7) id-mod(0)
             -- temporary, still tbs -- id-mod-laap(13)}

   DEFINITIONS EXPLICIT TAGS ::=
   BEGIN
   -- EXPORTS ALL --

   IMPORTS

      Holder, AttCertIssuer, AttributeCertificate
        FROM
         PKIXAttributeCertificate {iso(1) identified-organization(3)
                     dod(6) internet(1) security(5) mechanisms(5)
                     pkix(7) id-mod(0) id-mod-attribute-cert(12)}

      id-pkix
        FROM
          PKIX1Explicit88 {iso(1) identified-organization(3)
                     dod(6) internet(1) security(5) mechanisms(5)
                     pkix(7) id-mod(0) id-pkix1-explicit-88(1)} ;

       -- this is referenced, but not defined in [CMP]
      id-it   OBJECT IDENTIFIER ::= { id-pkix 4 }

      id-laap           OBJECT IDENTIFIER ::=
                            { id-it -- temporary, still tbs -- 7 }

      -- these OIDs are used as the infoType of the
      -- GenMsgContent and GenRepContent PKIBody fields respectively
```

```
    id-laap-req       OBJECT IDENTIFIER ::= { id-laap 1 }
    id-laap-rep       OBJECT IDENTIFIER ::= { id-laap 2 }

    LACRequestMessage ::= SEQUENCE {
         holder    Holder OPTIONAL,
         profile   [0] SEQUENCE {
              string          UTF8String,
              oid             OBJECT IDENTIFIER
              } OPTIONAL,
         issuer    [1] AttCertIssuer OPTIONAL,
         acOptions ACOptions OPTIONAL
    }
    ACOptions ::= BIT STRING {
         attr-encryption      (0),
         proxying             (1),
         object-digests       (2),
         aa-controls          (3)
    }
    LACResponseMessage ::= AttributeCertificate

END
```


Appendix C: Changes this version / Open Issues

Changes for version 02:

1. Synchronized with latest [ACPROF]
2. Added GenRepContent for LAAP response
3. Added sections on response handling and error codes
4. Added provisional OIDs
5. Decided not to use UDP transport.

Changes for version 01:

1. Synchronized with latest [ACPROF]
2. Samples to separate draft (same as [ACPROF])
3. Removed UDP transport requirement
4. Changed profile from UTF8 to optional OID and/or UTF8
5. Removed holderAuth feature
6. Added "supported options" indicator to request

Changes for version 00:

1. This is the first issue, previously LAAP was specified as part of the AC profile [ACPROF]
2. Changed LAAP so its now encapsulated in [CMP]
3. Added more definition of profile field
4. Added holderAuth field (probably temporarily)
5. Added requirement for UDP transport
6. Added compiled ASN.1 module

Open issues:

1. Register new pkix OIDs
2. What level of authentication to mandate (if any)
3. "badCertID" error code semantics and label to be resolved
4. Are details of CMP encapsulation correct, esp. for 2000 revision?