INTERNET-DRAFT
PKIX WG
Intended Category: Standards Track

Expires on 27 December 2002

D. W. Chadwick
University of Salford
S. Legg
Adacel Technologies
27 June 2002

Internet X.509 Public Key Infrastructure LDAP Schema and Syntaxes for PMIs

<draft-ietf-pkix-ldap-pmi-schema-00.txt>

STATUS OF THIS MEMO

This document is an Internet-Draft and is in full conformance with all the provisions of Section 10 of RFC2026 [1].

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Comments and suggestions on this document are encouraged. Comments on this document should be sent to the PKIX working group discussion list <ietf-pkix@imc.org> or directly to the authors.

ABSTRACT

This document describes LDAP schema features that are needed to support **X.509 Privilege Management Infrastructures. Specifically, X.509** attribute types, object classes, matching rules, attribute value syntaxes and attribute value assertion syntaxes needed for PMIs are defined.

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 [5].

1. Introduction

LDAPv3 [4] servers are a natural repository for X.509 PMI components e.g. attribute certificate attributes, attribute certificate revocation lists and attribute authority entries. This [document/ID/standard] defines the LDAP subschema needed for storing X.509 PMI information in LDAPv3 servers and for accessing this information e.g. searching for it, updating it, and perform comparisons on it.

2. Subschema Publishing

LDAPv3 allows the subschema supported by a server to be published in a subschema subentry. Clients following this profile which support the Search operation containing an extensible matching rule SHOULD use the subschemaSubentry attribute in the root DSE to find the subschemaSubentry, and SHOULD use the matchingRule and matchingRuleUse operational attributes in the subschema subentry in order to determine whether the server supports the various matching rules described below. Servers that support extensible matching SHOULD publish the matching rules they support in the matchingRule and matchingRuleUse operational attributes.

3. PMI Attributes and Syntaxes

LDAP servers MAY store any type of PMI attribute, and LDAP clients MAY request them to be returned by adding them to the Search Request AttributeDescriptionList (either explicitly or implicity via requesting all user attributes).

3.1 Attribute Certificate Attribute

The attributeCertificateAttribute is defined in 17.2.1 of [9]. It is used to hold the attribute certificates of a user. The LDAPspecific encoding for values of this attribute is described in section 3.4.

```
attributeCertificateAttribute  ATTRIBUTE ::= {
  WITH SYNTAX             AttributeCertificate
  EQUALITY MATCHING RULE  attributeCertificateExactMatch
  ID { joint-iso-ccitt(2) ds(5) attributeType(4)
          attributeCertificate(58) } }
```

The corresponding LDAP description is

```
( 2.5.4.58 NAME 'attributeCertificateAttribute'
EQUALITY attributeCertificateExactMatch
SYNTAX 1.2.826.0.1.3344810.7.5 )
```

3.2 Attribute Authority Certificate Attribute

The attribute authority attribute certificate is defined in 17.2.2 of [9]. The aAcertificate attribute holds the privileges of an attribute authority. The LDAPspecific encoding for values of this attribute is

described in section 3.4.

```
aACertificate  ATTRIBUTE ::= {
  WITH SYNTAX             AttributeCertificate
  EQUALITY MATCHING RULE  attributeCertificateExactMatch
  ID { joint-iso-ccitt(2) ds(5) attributeType(4)
          aACertificate(61) } }
```

The corresponding LDAP description is

```
( 2.5.4.61 NAME 'aACertificate'
EQUALITY attributeCertificateExactMatch
SYNTAX 1.2.826.0.1.3344810.7.5 )
```

3.3 Attribute Descriptor Certificate Attribute

The attributeDescriptorCertificate attribute is defined in 17.2.3 of [9]. The certificate is self signed by a source of authority and holds a description of the privilege and its delegation rules. The LDAPspecific encoding for values of this attribute is described in section 3.4.

```
attributeDescriptorCertificate  ATTRIBUTE ::= {
  WITH SYNTAX             AttributeCertificate
  EQUALITY MATCHING RULE  attributeCertificateExactMatch
  ID { joint-iso-ccitt(2) ds(5) attributeType(4)
          attributeDescriptorCertificate (62) } }
```

The corresponding LDAP description is

```
( 2.5.4.62 NAME 'attributeDescriptorCertificate'
EQUALITY attributeCertificateExactMatch
SYNTAX 1.2.826.0.1.3344810.7.5 )
```

3.4 Attribute Certificate Syntax

The LDAP-specific encoding for a certificate value is the octet string that results from BER/DER-encoding an X.509 attribute certificate. The following string states the OID assigned to this syntax:

```
(1.2.826.0.1.3344810.7.5 DESC 'Attribute Certificate' )
```

Servers MUST preserve values in this syntax exactly as given when storing and retrieving them. Transformation of these values between storage and retrieval MUST NOT take place.

3.5 Attribute Certificate Revocation List Attribute

The attributeCertificateRevocationList attribute is defined in section 17.2.4 of [9]. It holds a list of attribute certificates that have been revoked. The LDAP-specific encoding for values of this attribute is described in [2].

```
attributeCertificateRevocationList  ATTRIBUTE ::= {
  WITH SYNTAX             CertificateList
  EQUALITY MATCHING RULE  certificateListExactMatch
  ID { joint-iso-ccitt(2) ds(5) attributeType(4) aCRL(59) } }
```

The corresponding LDAP description is

```
( 2.5.4.59 NAME 'attributeCertificateRevocationList'
EQUALITY certificateListExactMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.9 )
```

3.6 Attribute Authority Certificate Revocation List Attribute

The attribute authority certificate revocation list attribute is defined in section 17.2.5 of [9]. It holds a list of AA certificates that have been revoked. The LDAP-specific encoding for values of this attribute is described in [2].

```
attributeAuthorityRevocationList  ATTRIBUTE ::= {
  WITH SYNTAX             CertificateList
  EQUALITY MATCHING RULE  certificateListExactMatch
  ID { joint-iso-ccitt(2) ds(5) attributeType(4) aARL(63) } }
```

The corresponding LDAP description is

```
( 2.5.4.63 NAME 'attributeAuthorityRevocationList'
EQUALITY certificateListExactMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.9 )
```

3.7 Delegation Path Attribute

The delegation path attribute contains delegation paths, each consisting of a sequence of attribute certificates

```
delegationPath    ATTRIBUTE       ::= {
  WITH SYNTAX     AttCertPath
  ID ( joint-iso-ccitt(2) ds(5) attributeType(4) delPath (73) } )

AttCertPath       ::=     SEQUENCE OF AttributeCertificate
```

The corresponding LDAP description is

```
( 2.5.4.73 NAME 'delegationPath'
SYNTAX 1.2.826.0.1.3344810.7.21 )
```

The following description is copied from X.509 (2000) [9].

"This attribute can be stored in the AA directory entry and would contain some delegation paths from that AA to other AAs. This attribute, if used, enables more efficient retrieval of delegated attribute certificates that form frequently used delegation paths. As such, there are no specific requirements for this attribute to be used and the set

of values that are stored in the attribute is unlikely to represent the complete set of delegation paths for any given AA."

3.8 Delegation Path Syntax

The LDAP-specific encoding for a delegation path value is the octet string that results from the BER/DER-encoding of a sequence of attribute certificates. The following string states the OID assigned to this syntax:

```
( 1.2.826.0.1.3344810.7.21 DESC 'Attribute certificate delegation
 path' )
```

Servers MUST preserve values in this syntax exactly as given when storing and retrieving them.

4 PMI Matching Rules

LDAP servers that support the storage of attributes with the AttributeCertificate syntax MUST support searching for entries containing specific attribute certificates, via the attributeCertificateExactMatch matching rule.

LDAPv3Servers MAY support flexible matching for any attributes with the AttributeCertificate syntax via the attributeCertificateMatch matching rule or any of the matching rules defined for the certificate extensions. LDAPv3 servers SHOULD publish the matching rules that they do support in the matchingRule and matchingRuleUse operational attributes of the subschema subentry. If the server does support flexible matching (either via attributeCertificateMatch or some other matching rule), then the extensibleMatch filter of the Search request MUST be supported. LDAPv3 clients MAY support the extensibleMatch filter of the Search operation, along one or more of the optional elements of attributeCertificateMatch or any of the certificate extension matching rules.

The LDAP-specific (i.e. string) encodings for the assertion syntaxes defined in this document are specified by the Generic String Encoding Rules (GSER) [3]. The ABNF in this document for these assertion syntaxes is provided only as a convenience and is equivalent to the encoding specified by the application of [3]. (The only exception to this is the alternative simple endoding for attributeCertificatExactMatch.) Since the associated ASN.1 types for the assertion syntaxes described here may be extended in future editions of X.509 [9], the provided ABNF should be regarded as a snapshot in time. The LDAP-specific encoding for any extension to a syntax's underlying ASN.1 type can be determined from [3]. In the event that there is a discrepancy between the ABNF in this document and the encoding determined by [3], [3] is to be taken as definitive.

4.1 Attribute Certificate Exact Match

The equality matching rule for all types of attribute with AttributeCertificate syntax is the attributeCertificateExactMatch, This is defined in 17.3.1 of [9]. It is reproduced below for the convenience of the reader (but see Outstanding Issues).

```
      attributeCertificateExactMatch  MATCHING-RULE ::= {
        SYNTAX  AttributeCertificateExactAssertion
        ID      { joint-iso-ccitt(2) ds(5) mr (13)
                    attributeCertificateExactMatch (45) } }

      AttributeCertificateExactAssertion ::= SEQUENCE {
        serialNumber    CertificateSerialNumber,
        issuer          AttCertIssuer }

      CertificateSerialNumber   ::= INTEGER

      AttCertIssuer ::=         [0]     SEQUENCE {
      issuerName                        GeneralNames  OPTIONAL,
      baseCertificateID [0]     IssuerSerial  OPTIONAL,
      objectDigestInfo  [1]     ObjectDigestInfo  OPTIONAL }
-- At least one component shall be present

      IssuerSerial  ::=  SEQUENCE {
        issuer          GeneralNames,
        serial          CertificateSerialNumber,
        issuerUID               UniqueIdentifier OPTIONAL }

      UniqueIdentifier ::= BIT STRING

      ObjectDigestInfo    ::= SEQUENCE {
        digestedObjectType  ENUMERATED {
                publicKey                       (0),
                publicKeyCert           (1),
                otherObjectTypes        (2) },
        otherObjectTypeID       OBJECT IDENTIFIER  OPTIONAL,
        digestAlgorithm         AlgorithmIdentifier,
        objectDigest            BIT STRING }
```

The LDAP definition for the above matching rule is:

```
        ( 2.5.13.45 NAME 'attributeCertificateExactMatch'
         SYNTAX 1.2.826.0.1.3344810.7.6)
```

The syntax definition is:

```
         (1.2.826.0.1.3344810.7.6 DESC 'Attribute certificate exact
          assertion (serial number and issuer details)' )
```

The LDAP-specific encoding of an assertion value of this syntax is a choice between

- the GSER encoding <GSERAttributeCertificateExactAssertion> defined by [3] and
- the simple encoding <SimpleCertificateExactAssertion> defined in [2].

The full syntax is described by the following Augmented BNF [10]:

```
AttributeCertificateExactAssertion =
                           GSERAttributeCertificateExactAssertion /
                           SimpleCertificateExactAssertion

GSERAttributeCertificateExactAssertion = "{" sp acea-serialNumber ","
                                         sp acea-issuer
                                         sp "}"

acea-serialNumber  = id-serialNumber msp CertificateSerialNumber
acea-issuer        = id-issuer       msp AttCertIssuer

id-serialNumber = %x73.65.72.69.61.6C.4E.75.6D.62.65.72
                     ; "serialNumber"
id-issuer       = %x69.73.73.75.65.72 ; "issuer"

AttCertIssuer = "{"    [ sp aci-issuerName ]
                   [ sep sp aci-baseCertificateID ]
                   [ sep sp aci-objectDigestInfo ]
                         sp "}"

At least one of <aci-issuerName>, <aci-baseCertificateID> or
<aci-objectDigestInfo> MUST be present.

aci-issuerName        = id-issuerName        msp GeneralNames
aci-baseCertificateID = id-baseCertificateID msp IssuerSerial
aci-objectDigestInfo  = id-objectDigestInfo  msp ObjectDigestInfo
id-issuerName         = %x69.73.73.75.65.72.4E.61.6D.65
                           ; "issuerName"

GeneralNames = "{" sp GeneralName *( "," sp GeneralName ) sp "}"
GeneralName  = gn-otherName
               / gn-rfc822Name
               / gn-dNSName
               / gn-x400Address
               / gn-directoryName
               / gn-ediPartyName
               / gn-uniformResourceIdentifier
               / gn-iPAddress
               / gn-registeredID

gn-otherName     = id-otherName       ":" OtherName
gn-rfc822Name    = id-rfc822Name      ":" IA5String
gn-dNSName       = id-dNSName         ":" IA5String
gn-x400Address   = id-x400Address     ":" ORAddress
gn-directoryName = id-directoryName   ":" Name
```

```
gn-ediPartyName  = id-ediPartyName    ":" EDIPartyName
gn-iPAddress     = id-iPAddress       ":" OCTET-STRING
gn-registeredID  = gn-id-registeredID ":" OBJECT-IDENTIFIER

gn-uniformResourceIdentifier = id-uniformResourceIdentifier
                               ":" IA5String

id-otherName       = %x6F.74.68.65.72.4E.61.6D.65 ; "otherName"
id-rfc822Name    = %x72.66.63.38.32.32.4E.61.6D.65 ; "rfc822Name"
id-dNSName       = %x64.4E.53.4E.61.6D.65 ; "dNSName"
id-x400Address   = %x78.34.30.30.41.64.64.72.65.73.73
                     ; "x400Address"
id-directoryName = %x64.69.72.65.63.74.6F.72.79.4E.61.6D.65
                     ; "directoryName"
id-ediPartyName  = %x65.64.69.50.61.72.74.79.4E.61.6D.65
                     ; "ediPartyName"
id-iPAddress     = %x69.50.41.64.64.72.65.73.73 ; "iPAddress"
id-registeredId  = %x72.65.67.69.73.74.65.72.65.64.49.64
                     ; "registeredId"

id-uniformResourceIdentifier = %x75.6E.69.66.6F.72.6D.52.65.73.6F.75
                               %x72.63.65.49.64.65.6E.74.69.66.69.65
                               %x72 ; "uniformResourceIdentifier"
gn-id-registeredID = %x72.65.67.69.73.74.65.72.65.64.49.44
                       ; "registeredID"

OtherName  = "{" sp on-type-id "," sp on-value sp "}"
on-type-id = id-type-id msp OBJECT-IDENTIFIER
on-value   = id-value msp Value
id-type-id = %x74.79.70.65.2D.69.64 ; "type-id"
id-value   = %x76.61.6C.75.65       ; "value"

The <Value> rule is defined in [3].

EDIPartyName    = "{" [ sp nameAssigner "," ] sp partyName sp "}"
nameAssigner    = id-nameAssigner msp DirectoryString
partyName       = id-partyName msp DirectoryString
id-nameAssigner = %x6E.61.6D.65.41.73.73.69.67.6E.65.72
                    ; "nameAssigner"
id-partyName    = %x70.61.72.74.79.4E.61.6D.65 ; "partyName"
id-objectDigestInfo   = %x6F.62.6A.65.63.74.44.69.67.65.73.74.49.6E
                          %x66.6F ; "objectDigestInfo"



ObjectDigestInfo = "{"      sp odi-digestedObjectType
                     [ "," sp odi-otherObjectTypeID ]
                       "," sp odi-digestAlgorithm
                       "," sp odi-objectDigest
                           sp "}"

odi-digestedObjectType = id-digestedObjectType msp
```

```
                            DigestedObjectType
odi-otherObjectTypeID  = id-otherObjectTypeID msp OBJECT-IDENTIFIER
odi-digestAlgorithm    = id-digestAlgorithm msp AlgorithmIdentifier
odi-objectDigest       = id-objectDigest msp BIT-STRING

id-digestedObjectType = %x64.69.67.65.73.74.65.64.4F.62.6A.65.63.74
                        %x54.79.70.65 ; "digestedObjectType"
id-otherObjectTypeID  = %x6F.74.68.65.72.4F.62.6A.65.63.74.54.79.70
                        %x65.49.44 ; "otherObjectTypeID"
id-digestAlgorithm    = %x64.69.67.65.73.74.41.6C.67.6F.72.69.74.68
                        %x6D ; "digestAlgorithm"
id-objectDigest       = %x6F.62.6A.65.63.74.44.69.67.65.73.74
                        ; "objectDigest"

DigestedObjectType  = id-publicKey
                    / id-publicKeyCert
                    / id-otherObjectTypes
id-publicKey        = %x70.75.62.6C.69.63.4B.65.79 ; "publicKey"
id-publicKeyCert    = %x70.75.62.6C.69.63.4B.65.79.43.65.72.74
                       ; "publicKeyCert"
id-otherObjectTypes = %x6F.74.68.65.72.4F.62.6A.65.63.74.54.79.70.65
                       %x73 ; "otherObjectTypes"

AlgorithmIdentifier = "{"      sp ai-algorithm
                       [ "," sp ai-parameters ]
                             sp "}"

ai-algorithm  = id-algorithm  msp OBJECT-IDENTIFIER
ai-parameters = id-parameters msp Value
id-algorithm  = %x61.6C.67.6F.72.69.74.68.6D    ; "algorithm"
id-parameters = %x70.61.72.61.6D.65.74.65.72.73 ; "parameters"

IssuerSerial = "{"      sp is-issuer
                    "," sp is-serial
                  [ "," sp is-issuerUID ]
                        sp "}"

is-issuer    = id-issuer    msp GeneralNames
is-serial    = id-serial    msp CertificateSerialNumber
is-issuerUID = id-issuerUID msp UniqueIdentifier

id-serial    = %x73.65.72.69.61.6C          ; "serial"
id-issuerUID = %x69.73.73.75.65.72.55.49.44 ; "issuerUID"

UniqueIdentifier = BIT-STRING
```

4.2 Attribute Certificate Match

Attribute certificate matching rule is defined in section 17.3.2 of [9]. For the convenience of the reader it is reproduced below:

```
attributeCertificateMatch  MATCHING-RULE ::= {
        SYNTAX  AttributeCertificateAssertion
        ID      { joint-iso-ccitt(2) ds(5) mr (13)
                        attributeCertificateMatch (42) }


AttributeCertificateAssertion ::= SEQUENCE {
        holder          [0] CHOICE {
                            baseCertificateID   [0] IssuerSerial,
                            subjectName         [1] GeneralNames
                                } OPTIONAL,
        issuer          [1] GeneralNames OPTIONAL,
        attCertValidity [2] GeneralizedTime OPTIONAL,
        attType         [3] SET OF AttributeType OPTIONAL }
--At least one component of the sequence must be present
```

The LDAP definition of the attributeCertificateMatch matching rule is:

```
( 2.5.13.42 NAME 'attributeCertificateMatch'
    SYNTAX 1.2.826.0.1.3344810.7.7 )
```

The syntax definition is:

```
(1.2.826.0.1.3344810.7.7
    DESC 'Attribute Certificate Assertion' )
```

The LDAP string encoding of an assertion value of this syntax is given by the following ABNF:

```
AttributeCertificateAssertion = "{"    [ sp aca-holder ]
                                  [ sep sp aca-issuer ]
                                  [ sep sp aca-attCertValidity ]
                                  [ sep sp aca-attType ]
                                        sp "}"

aca-holder          = id-holder          msp ACAHolder
aca-issuer          = id-issuer          msp GeneralNames
aca-attCertValidity = id-attCertValidity msp GeneralizedTime
aca-attType         = id-attType         msp SETOFAttributeType

ACAHolder = acah-baseCertificateID / acah-holderName

acah-baseCertificateID = id-baseCertificateID ":" IssuerSerial
acah-holderName        = id-holderName        ":" GeneralNames

id-baseCertificateID = %x62.61.73.65.43.65.72.74.69.66.69.63.61.74
                          %x65.49.44 ; "baseCertificateID"
id-holderName        = %x68.6F.6C.64.65.72.4E.61.6D.65
                          ; "holderName"

SETOFAttributeType = "{" sp AttributeType
```

```
                    *( "," sp AttributeType ) sp "}"
```

The <AttributeType> rule is given in [6].

5 AC Extensions Matching Rules

X.509 defines the following matching rules for matching on various extensions within an attribute certificate.

5.1 Holder Issuer Match

Holder Issuer Match is described in section 17.3.3 of [9]. The string description of the holderIssuerMatch matching rule is:

```
        ( 2.5.13.46 NAME 'holderIssuerMatch'
         SYNTAX 1.2.826.0.1.3344810.7.10)
```

The syntax definition is:

```
        (1.2.826.0.1.3344810.7.10 DESC 'Holder Issuer Assertion' )
```

The ASN.1 for HolderIssuerAssertion is defined in 17.3.3 of [9], as are the semantics of its components.

The LDAP string encoding of an assertion value of this syntax is given by the following ABNF:

```
HolderIssuerAssertion = "{"    [ sp hia-holder ]
                           [ sep sp hia-issuer ]
                                 sp "}"

hia-holder = id-holder msp Holder
hia-issuer = id-issuer msp AttCertIssuer

Holder = "{"    [ sp h-baseCertificateID ]
            [ sep sp h-entityName ]
            [ sep sp h-objectDigestInfo ]
                  sp "}"
```

At least one of <h-baseCertificateID>, <h-entityName> or <h-objectDigestInfo> MUST be present.

```
h-baseCertificateID = id-baseCertificateID msp IssuerSerial
h-entityName        = id-entityName        msp GeneralNames
h-objectDigestInfo  = id-objectDigestInfo  msp ObjectDigestInfo
id-entityName       = %x65.6E.74.69.74.79.4E.61.6D.65 ; "entityName"
```

5.2 Delegation Path Match

Delegation Path Match is described in section 17.3.4 of [9]. The string description of the delegationPathMatch matching rule is:

```
      ( 2.5.13.61 NAME 'delegationPathMatch'
       SYNTAX 1.2.826.0.1.3344810.7.10)
```

The syntax definition is:

```
      (1.2.826.0.1.3344810.7.10 DESC 'DelMatchSyntax' )
```

The ASN.1 for DelMatchSyntax is defined in 17.3.4 of [9], as are the semantics of its components.

The LDAP string encoding of an assertion value of this syntax is given by the following ABNF:

```
DelMatchSyntax = "{" sp dms-firstIssuer ","
                     sp dms-lastHolder
                     sp "}"
dms-firstIssuer = id-firstIssuer msp AttCertIssuer
dms-lastHolder  = id-lastHolder  msp Holder
id-firstIssuer  = %x66.69.72.73.74.49.73.73.75.65.72 ; "firstIssuer"
id-lastHolder   = %x6C.61.73.74.48.6F.6C.64.65.72    ; "lastHolder"
```

5.3 Authority Attribute Identifier Match

Authority Attribute Identifier Match is described in section 15.5.2.4.1 of [9]. The string description of the authAttIdMatch matching rule is:

```
      ( 2.5.13.53 NAME 'authAttIdMatch'
       SYNTAX 1.2.826.0.1.3344810.7.12)
```

The syntax definition is:

```
      (1.2.826.0.1.3344810.7.12 DESC 'Authority Attribute Identifier
       Syntax' )
```

The ASN.1 for AuthorityAttributeIdentifierSyntax is defined in 15.5.2.4 of [9], as are the semantics of its components.

The LDAP string encoding of an assertion value of this syntax is given by the following ABNF:

```
AuthorityAttributeIdentifierSyntax = "{" sp AuthAttId
                                       *( "," sp AuthAttId ) sp "}"

AuthAttId = IssuerSerial
```

5.4 Role Specification Certificate Identifier Match

Role Specification Certificate Identifier match is described in section **15.4.2.1.1 of [9]. The string description of the roleSpecCertIdMatch** Match matching rule is:

```
    ( 2.5.13.54 NAME 'roleSpecCertIdMatch '
    SYNTAX 1.2.826.0.1.3344810.7.13)
```

The syntax definition is:

```
       (1.2.826.0.1.3344810.7.13 DESC 'Role Specification Ceritificate
        Identifier Syntax' )
```

The ASN.1 for RoleSpecCertIdentifierSyntax is defined in 15.4.2.1 of [9], as are the semantics of its components.

The LDAP string encoding of an assertion value of this syntax is given by the following ABNF:

```
RoleSpecCertIdentifierSyntax = "{" sp RoleCertSpecIdentifier
                             *( "," sp RoleCertSpecIdentifier ) sp "}"
RoleCertSpecIdentifier = "{"      sp rsci-roleName
                                "," sp rsci-roleCertIssuer
                                [ "," sp rsci-roleCertSerialNumber ]
                                [ "," sp rsci-roleCertLocator ]
                                      sp "}"
rsci-roleName             = id-roleName msp GeneralName
rsci-roleCertIssuer       = id-roleCertIssuer msp GeneralName
rsci-roleCertSerialNumber = id-roleCertSerialNumber msp
                                CertificateSerialNumber
rsci-roleCertLocator      = id-roleCertLocator msp GeneralName
id-roleName               = %x72.6F.6C.65.4E.61.6D.65 ; "roleName"
id-roleCertIssuer         = %x72.6F.6C.65.43.65.72.74.49.73.73.75.65
                                %x72 ; "roleCertIssuer"
id-roleCertSerialNumber   = %x72.6F.6C.65.43.65.72.74.53.65.72.69.61
                                %x6C.4E.75.6D.62.65.72
                                ; "roleCertSerialNumber"
id-roleCertLocator        = %x72.6F.6C.65.43.65.72.74.4C.6F.63.61.74
                                %x6F.72 ; "roleCertLocator"
```

5.5 Basic Attribute Constraints Match

Basic Attribute Constraints Match is described in section 15.5.2.1.1 of [9]. The string
description of the holderIssuerMatch matching rule is:

```
      ( 2.5.13.55 NAME ' basicAttConstraintsMatch '
      SYNTAX 1.2.826.0.1.3344810.7.14)
```

The syntax definition is:

```
     (1.2.826.0.1.3344810.7.14 DESC 'Basic Attributes Constraints
      Syntax' )
```

The ASN.1 for BasicAttConstraintsSyntax is defined in 15.5.2.1 of [9], as are the semantics of its components.

The LDAP string encoding of an assertion value of this syntax is given by the following ABNF:

```
BasicAttConstraintsSyntax = "{"    [ sp bacm-authority ]
                                [ sep sp bacm-pathLenConstraint ]
                                      sp "}"
bacm-authority         = id-authority         msp BOOLEAN
bacm-pathLenConstraint = id-pathLenConstraint msp INTEGER-0-MAX
id-authority           = %x61.75.74.68.6F.72.69.74.79 ; "authority"
id-pathLenConstraint   = %x70.61.74.68.4C.65.6E.43.6F.6E.73.74.72.61
                             %x69.6E.74 ; "pathLenConstraint"
```

The <BOOLEAN> rule is given in [6].

5.6 Delegated Name Constraints Match

Delegated Name Constraints Match is described in section 15.5.2.2.1 of [9]. The string description of the holderIssuerMatch matching rule is:

```
( 2.5.13.56 NAME ' delegatedNameConstraintsMatch'
 SYNTAX 1.2.826.0.1.3344810.7.15)
```

The syntax definition is:

```
(1.2.826.0.1.3344810.7.15 DESC 'Name Constraints Syntax' )
```

The ASN.1 for NameConstraintsSyntax is defined in 8.4.2.2 of [9], and the semantics of its components when used for delegated name constraints are described in 15.5.2.2.

The LDAP string encoding of an assertion value of this syntax is given in Section 4.2.

5.7 Time Specification Match

Time Specification Match is described in section 15.1.2.1.1 of [9]. The string description of the timeSpecificationMatch matching rule is:

```
( 2.5.13.57 NAME ' timeSpecificationMatch '
 SYNTAX 1.2.826.0.1.3344810.7.16)
```

The syntax definition is:

```
(1.2.826.0.1.3344810.7.16 DESC 'Time Specification' )
```

The ASN.1 for TimeSpecification is defined in 7.2 of [7], as are the semantics of its components.

The LDAP string encoding of an assertion value of this syntax is given by the following ABNF:

```
TimeSpecification = "{"      sp ts-time
```

```
                    [ "," sp ts-notThisTime ]
                    [ "," sp ts-timeZone ]
                          sp "}"

ts-time        = id-time        msp TSTime
ts-notThisTime = id-notThisTime msp BOOLEAN
ts-timeZone    = id-timeZone    msp TimeZone
id-time        = %x74.69.6D.65                      ; "time"
id-notThisTime = %x6E.6F.74.54.68.69.73.54.69.6D.65 ; "notThisTime"
id-timeZone    = %x74.69.6D.65.5A.6F.6E.65          ; "timeZone"

TSTime       = tst-absolute / tst-periodic
tst-absolute = id-absolute ":" AbsoluteTime
tst-periodic = id-periodic ":" Periods
AbsoluteTime = "{"    [ sp at-startTime ]
                 [ sep sp at-endTime ]
                       sp "}"
at-startTime = id-startTime msp GeneralizedTime
at-endTime   = id-endTime   msp GeneralizedTime
id-startTime = %x73.74.61.72.74.54.69.6D.65 ; "startTime"
id-endTime   = %x65.6E.64.54.69.6D.65       ; "endTime"

Periods = "{" [ sp Period *( "," sp Period ) ] sp "}"
Period  = "{"    [ sp p-timesOfDay ]
             [ sep sp p-days ]
             [ sep sp p-weeks ]
             [ sep sp p-months ]
             [ sep sp p-years ]
                   sp "}"

p-timesOfDay  = id-timesOfDay msp DayTimeBands
p-days        = id-days       msp Days
p-weeks       = id-weeks      msp Weeks
p-months      = id-months     msp Months
p-years       = id-years      msp Years
id-timesOfDay = %x74.69.6D.65.73.4F.66.44.61.79 ; "timesOfDay"
id-days       = %x64.61.79.73                   ; "days"
id-weeks      = %x77.65.65.6B.73                ; "weeks"
id-months     = %x6D.6F.6E.74.68.73             ; "months"
id-years      = %x79.65.61.72.73                ; "years"

DayTimeBands = "{" sp DayTimeBand *( "," sp DayTimeBand ) sp "}"
DayTimeBand  = "{"    [ sp dtb-startDayTime ]
                  [ sep sp dtb-endDayTime ]
                        sp "}"
dtb-startDayTime = id-startDayTime msp DayTime
dtb-endDayTime   = id-endDayTime   msp DayTime
id-startDayTime  = %x73.74.61.72.74.44.61.79.54.69.6D.65
                     ; "startDayTime"
id-endDayTime    = %x65.6E.64.44.61.79.54.69.6D.65 ; "endDayTime"
```

```
DayTime = "{"      sp dt-hour
             [ "," sp dt-minute ]
             [ "," sp dt-second ]
                   sp "}"
dt-hour   = id-hour   msp INTEGER ; 0 to 23
dt-minute = id-minute msp INTEGER ; 0 to 59
dt-second = id-second msp INTEGER ; 0 to 59
id-hour   = %x68.6F.75.72       ; "hour"
id-minute = %x6D.69.6E.75.74.65 ; "minute"
id-second = %x73.65.63.6F.6E.64 ; "second"

Days        = days-intDay / days-bitDay / days-dayOf
days-intDay = id-intDay ":" SET-OF-INTEGER
days-bitDay = id-bitDay ":" BitDay
days-dayOf  = id-dayOf  ":" XDayOf
id-intDay   = %x69.6E.74.44.61.79 ; "intDay"
id-bitDay   = %x62.69.74.44.61.79 ; "bitDay"
id-dayOf    = %x64.61.79.4F.66    ; "dayOf"

SET-OF-INTEGER = "{" [ sp INTEGER *( "," sp INTEGER ) ] "}"

BitDay       = BIT-STRING / day-bit-list
day-bit-list = "{" [ sp day *( "," sp day ) ] sp "}"
day          = %x73.75.6E.64.61.79            ; "sunday"
               / %x6D.6F.6E.64.61.79          ; "monday"
               / %x74.75.65.73.64.61.79       ; "tuesday"
               / %x77.65.64.6E.65.73.64.61.79 ; "wednesday"
               / %x74.68.75.72.73.64.61.79    ; "thursday"
               / %x66.72.69.64.61.79          ; "friday"
               / %x73.61.74.75.72.64.61.79    ; "saturday"

XDayOf = xdo-first / xdo-second / xdo-third / xdo-fourth / xdo-fifth

xdo-first  = id-first  ":" NamedDay
xdo-second = id-second ":" NamedDay
xdo-third  = id-third  ":" NamedDay
xdo-fourth = id-fourth ":" NamedDay
xdo-fifth  = id-fifth  ":" NamedDay

NamedDay        = nd-intNamedDays / nd-bitNamedDays
nd-intNamedDays = id-intNamedDays ":" day
nd-bitNamedDays = id-bitNamedDays ":" ( BIT-STRING / day-bit-list )
id-intNamedDays = %x69.6E.74.4E.61.6D.65.64.44.61.79.73
                    ; "intNamedDays"
id-bitNamedDays = %x62.69.74.4E.61.6D.65.64.44.61.79.73
                    ; "bitNamedDays"

Weeks          = weeks-allWeeks / weeks-intWeek / weeks-bitWeek
weeks-allWeeks = id-allWeeks ":" NULL
weeks-intWeek  = id-intWeek  ":" SET-OF-INTEGER
weeks-bitWeek  = id-bitWeek  ":" BitWeek
```

```
id-allWeeks    = %x61.6C.6C.57.65.65.6B.73 ; "allWeeks"
id-intWeek     = %x69.6E.74.57.65.65.6B    ; "intWeek"
id-bitWeek     = %x62.69.74.57.65.65.6B    ; "bitWeek"
BitWeek        = BIT-STRING / week-bit-list
week-bit-list  = "{" [ sp week-bit *( "," sp week-bit ) ] sp "}"
week-bit       = %x77.65.65.6B.31   ; "week1"
                 / %x77.65.65.6B.32 ; "week2"
                 / %x77.65.65.6B.33 ; "week3"
                 / %x77.65.65.6B.34 ; "week4"
                 / %x77.65.65.6B.35 ; "week5"

Months = months-allMonths / months-intMonth / months-bitMonth

months-allMonths = id-allMonths ":" NULL
months-intMonth  = id-intMonth  ":" SET-OF-INTEGER
months-bitMonth  = id-bitMonth  ":" BitMonth
id-allMonths     = %x61.6C.6C.4D.6F.6E.74.68.73 ; "allMonths"
id-intMonth      = %x69.6E.74.4D.6F.6E.74.68    ; "intMonth"
id-bitMonth      = %x62.69.74.4D.6F.6E.74.68    ; "bitMonth"
BitMonth         = BIT-STRING / month-bit-list
month-bit-list   = "{" [ sp month-bit *( "," sp month-bit ) ] sp "}"
month-bit        = %x6A.61.6E.75.61.72.79            ; "january"
                   / %x66.65.62.72.75.61.72.79       ; "february"
                   / %x6D.61.72.63.68                ; "march"
                   / %x61.70.72.69.6C                ; "april"
                   / %x6D.61.79                      ; "may"
                   / %x6A.75.6E.65                   ; "june"
                   / %x6A.75.6C.79                   ; "july"
                   / %x61.75.67.75.73.74             ; "august"
                   / %x22.73.65.70.74.65.6D.62.65.72 ; "september"
                   / %x6F.63.74.6F.62.65.72          ; "october"
                   / %x6E.6F.76.65.6D.62.65.72       ; "november"
                   / %x64.65.63.65.6D.62.65.72       ; "december"

Years = "{" [ sp Year *( "," sp Year ) ] sp "}"
Year  = INTEGER ; must be >= 1000

TimeZone = INTEGER ; -12 to 12
```

The <NULL> rule is given in [6].

5.8 Acceptable Certificate Policies Match

Acceptable Certificate Policies Match is described in section 15.5.2.3.1 of [9]. The string description of the acceptableCertPoliciesMatch matching rule is:

```
( 2.5.13.59 NAME 'acceptableCertPoliciesMatch'
 SYNTAX 1.2.826.0.1.3344810.7.17)
```

The syntax definition is:

```
(1.2.826.0.1.3344810.7.17 DESC 'Acceptable Certificate Policies Syntax)
```

The ASN.1 for AcceptableCertPoliciesSyntax is defined in 15.5.2.3 of [9], as are the semantics of its components.

The LDAP string encoding of an assertion value of this syntax is given by the following ABNF:

```
AcceptableCertPoliciesSyntax = "{" sp CertPolicyId
                                 *( "," sp CertPolicyId ) sp "}"
```

5.9 Attribute Descriptor Match

Attribute Descriptor Match is described in section 15.3.2.2.1 of [9]. The string description of the attDescriptor matching rule is:

```
     ( 2.5.13.58 NAME 'attDescriptor'
      SYNTAX 1.2.826.0.1.3344810.7.18)
```

The syntax definition is:

```
     (1.2.826.0.1.3344810.7.18 DESC 'Attribute Descriptor Syntax')
```

The ASN.1 for AttributeDescriptorSyntax is defined in 15.3.2.2 of [9], as are the semantics of its components.

The LDAP string encoding of an assertion value of this syntax is given by the following ABNF:

```
AttributeDescriptorSyntax = "{"      sp ads-identifier
                                  "," sp ads-attributeSyntax
                               [ "," sp ads-name ]
                               [ "," sp ads-description ]
                                  "," sp ads-dominationRule
                                     sp "}"

ads-identifier      = id-identifier msp AttributeIdentifier
ads-attributeSyntax = id-attributeSyntax msp AttributeSyntax
ads-name            = id-name msp AttributeName
ads-description     = id-description msp AttributeDescription
ads-dominationRule  = id-dominationRule msp PrivilegePolicyIdentifier
id-identifier       = %x69.64.65.6E.74.69.66.69.65.72 ; "identifier"
id-attributeSyntax  = %x61.74.74.72.69.62.75.74.65.53.79.6E.74.61.78
                          ; "attributeSyntax"
id-name             = %x6E.61.6D.65 ; "name"
id-description      = %x64.65.73.63.72.69.70.74.69.6F.6E
                          ; "description"
id-dominationRule   = %x64.6F.6D.69.6E.61.74.69.6F.6E.52.75.6C.65
                          ; "dominationRule"

AttributeSyntax      = OCTET-STRING ; an empty string is not allowed
AttributeIdentifier  = AttributeType
```

```
AttributeName        = UTF8String ; an empty string is not allowed
AttributeDescription = UTF8String ; an empty string is not allowed

PrivilegePolicyIdentifier = "{" sp ppi-privilegePolicy ","
                                sp ppi-privPolSyntax
                                sp "}"

ppi-privilegePolicy = id-privilegePolicy msp PrivilegePolicy
ppi-privPolSyntax   = id-privPolSyntax   msp InfoSyntax
id-privilegePolicy  = %x70.72.69.76.69.6C.65.67.65.50.6F.6C.69.63.79
                        ; "privilegePolicy"
id-privPolSyntax    = %x70.72.69.76.50.6F.6C.53.79.6E.74.61.78
                        ; "privPolSyntax"

PrivilegePolicy = OBJECT-IDENTIFIER

InfoSyntax = is-content / is-pointer
is-content = id-content ":" DirectoryString
is-pointer = id-pointer ":" InfoSyntaxPointer
id-content = %x63.6F.6E.74.65.6E.74 ; "content"
id-pointer = %x70.6F.69.6E.74.65.72 ; "pointer"

InfoSyntaxPointer = "{"      sp isp-name
                       [ "," sp isp-hash ]
                             sp "}"

isp-name = id-name msp GeneralNames
isp-hash = id-hash msp HASH
id-hash  = %x68.61.73.68 ; "hash"
HASH     = "{" sp h-algorithmIdentifier ","
               sp h-hashValue
               sp "}"

h-algorithmIdentifier = id-algorithmIdentifier msp AlgorithmIdentifier
h-hashValue           = id-hashValue           msp BIT-STRING

id-algorithmIdentifier = %x61.6C.67.6F.72.69.74.68.6D.49.64.65.6E.74
                            %x69.66.69.65.72 ; "algorithmIdentifier"
id-hashValue           = %x68.61.73.68.56.61.6C.75.65 ; "hashValue"
```

The <UTF8String> rule is given in [6].

5.10 Source of Authority Match

Note. This rule has not been defined by X.509, but this is perhaps an omission that should be rectified. It is an easy matching rule to define since it has a null syntax i.e. we will be matching on whether the extension is present or not.

Source of Authority Match returns TRUE if an attribute certificate contains an SOA Identifier extension. The SOA Identifier extension is described in section 15.3.2.1 of [9]. The string description of the

sOAIdentifierMatch matching rule is:

```
    ( 2.5.13.x NAME 'sOAIdentifierMatch'
    SYNTAX 1.2.36.79672281.1.5.1)
```

The syntax definition of 1.2.36.79672281.1.5.1 (NULL) is given in [3].

6 PMI Object Classes

The definitions of the PMI directory object classes can be found in section 17.1 of [9]. They are repeated here for the convenience of the reader.

```
pmiUser OBJECT-CLASS ::= {
 -- a privilege holder
        SUBCLASS OF     {top}
        KIND            auxiliary
        MAY CONTAIN     {attributeCertificateAttribute}
        ID { joint-iso-ccitt(2) ds(5) objectClass(6) pmiUser (24) } }


pmiAA OBJECT-CLASS ::= {
 -- an attribute authority
        SUBCLASS OF     {top}
        KIND            auxiliary
        MAY CONTAIN     {aACertificate |
                        attributeCertificateRevocationList |
                        attributeAuthorityRevocationList}
        ID { joint-iso-ccitt(2) ds(5) objectClass(6) pmiAA (25) } }


pmiSOA OBJECT-CLASS ::= {
 -- a PMI Source of Authority
        SUBCLASS OF     {top}
        KIND            auxiliary
        MAY CONTAIN     {attributeCertificateRevocationList |
                        attributeAuthorityRevocationList |
                        attributeDescriptorCertificate}
        ID { joint-iso-ccitt(2) ds(5) objectClass(6) pmiSOA (26) } }

attCertCRLDistributionPt        OBJECT-CLASS ::= {
-- an AC CRL distribution point
   SUBCLASS OF          {top}
   KIND                 auxiliary
   MAY CONTAIN          { attributeCertificateRevocationList |
                                attributeAuthorityRevocationList }
   ID { joint-iso-ccitt(2) ds(5) objectClass(6)
attCertCRLDistributionPts (27) } }

pmiDelegationPath   OBJECT-CLASS        ::= {
-- an object that may contain a delegation path
```

```
        SUBCLASS OF     {top}
        KIND                    auxiliary
        MAY CONTAIN     { delegationPath }
ID { joint-iso-ccitt(2) ds(5) objectClass(6) delegationPath (33) } }

privilegePolicy OBJECT-CLASS    ::= {
-- an object that may contain privilege policy information
        SUBCLASS OF     {top}
        KIND                    auxiliary
        MAY CONTAIN     {privPolicy }
ID { joint-iso-ccitt(2) ds(5) objectClass(6) privilegePolicy (32) } }
```

7. Filter Examples

The following examples are written using the string representation of Search filters defined in [14]. Line-breaks have been added as an aid to readability.

i) To exactly match one attribute certificate using equalityMatch with attributeCertificateExactMatch and GSERAttributeCertificateExactAssertion

```
      (attributeCertificateAttribute={serialNumber 12345 , issuer {
issuerName { directoryName rdnSequence:"O=truetrust ltd, C=GB" })
```

ii) To exactly match one attribute certificate using equalityMatch with attributeCertificateExactMatch and SimpleCertificateExactAssertion

```
      (attributeCertificateAttribute=12345$O=truetrust ltd, C=GB)
```

iii) To match on the serial number of an attribute certificate using extensibleMatch with component matching [13]

```
(attributeCertificateAttribute:componentFilterMatch:=
       item:{ component "serialNumber", rule integerMatch,
value 12345 })
```

iv) To exactly match one attribute certificate using extensibleMatch with component matching

```
(attributeCertificateAttribute:componentFilterMatch:=and:{
item:{ component "serialNumber", rule integerMatch, value 12345 }
item:{ component "issuer.issuerName.directoryName.rdnSequence", rule
distinguishedNameMatch, value "O=truetrust ltd, C=GB"  } })
```

v) To match attribute certificates containing a certain role

To Be Worked Out Later#

8. Security Considerations

This [Internet Draft/Standard] describes the schema for the storage and matching of PMI attributes (attribute certificates, revocation lists etc.) in an LDAP directory server. It does not address the protocol for the retrieval of this information.

LDAP servers SHOULD use authentication and access control methods to protect this information during its storage from unauthorised modification and retrieval. In addition, clients MAY choose to encrypt the attributes in the attribute certificates before storing them in an LDAP server to ensure their confidentiality.

9. References

Normative

[1] Bradner, S. The Internet Standards Process -- Revision 3. RFC 2026 October 1996.

[2] Chadwick, D.W., Legg, S. "Internet X.509 Public Key Infrastructure LDAP Schema and Syntaxes for PKIs" <draft-pkix-ldap-pki-schema-00.txt>, June 2002

[3] S. Legg, "Generic String Encoding Rules", <draft-legg-ldap-gser-XX.txt>, March 2002, a work in progress

[4] J. Sermersheim "Lightweight Directory Access Protocol (v3)" <draft-ietf-ldapbis-protocol-02.txt> July 2001

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

[6] S. Legg, "Common Elements of GSER Encodings", <draft-legg-ldap-gser-abnf-XX.txt>, March 2002, a work in progress

[7] ITU-T Rec. X.520(2000) The Directory: Selected Attribute Types

[9] ITU-T Rec. X.509(2000) The Directory: Authentication Framework

[10] D. Crocker, P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997

Informative

[13] S. Legg, "LDAP & X.500 Component Matching Rules", <draft-legg-ldapext-component-matching-04.txt>, November 2001, a work in progress

[14] Howes, T. "The String Representation of LDAP Search Filters". RFC

2254, December 1997.

10. Intellectual Property 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. [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.

11. Copyright

"AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

12. Authors' Addresses

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

Email: d.w.chadwick@salford.ac.uk

Steven Legg
Adacel Technologies Ltd.
405-409 Ferntree Gully Road,
Mount Waverley,
Victoria, 3149
Australia

Email: steven.legg@adacel.com.au

13. Changes

>From Version 00

i) Added ABNF notation for all of the syntaxes.

ii) Removed the restriction on the syntax of Distribution Point Names.

iii) Removed constraints on IssuerSerial.

iv) Bug detected in X.509 AttributeCertificateExactMatch that will need resolving.

v) Changed the string encodings for non-exact matches to keywords for each component instead of $ separators.

>From Version 01

i) Added and corrected all X.509 PKI schema definitions, since these have been removed from RFC2252-bis.
ii) Changed assertion syntaxes to use the syntax defined by Component Matching Rules
iii) Included all the matching rules for AC extensions

>From Version 02 of <draft-pkix-ldap-schema-02.txt>

i) PKI and PMI schema has been split into separate IDs
ii) Example have been added
iii) Text has been added to mandate that servers must store and retrieve syntaxes containing digital signatures exactly as given.
iv) Text has been removed concerning the use of the ;binary encoding option, as per the decision of the LDAPBIS group.

14. Outstanding Issues

i. There is still a bug in the X.509 AttributeCertificateExactAssertion. It reads:

```
AttributeCertificateExactAssertion ::= SEQUENCE {
        serialNumber    CertificateSerialNumber OPTIONAL,
        issuer          IssuerSerial }
```

OPTIONAL should be removed from the serialNumber. IssuerSerial should be replaced by AttCertIssuer. This ID has assumed that the change will be made.

ii. Should the AttributeType in Attribute Certificate Match allow the LDAP <descr> encoding option for describing attribute type OIDs (i.e. user friendly names instead of object identifiers)? Note that attribute names are not guaranteed to be unique, whereas OIDs are.

iii. The Source of Authority Match is not defined in X.509. Do we prefer compatibility with X.509 and remove it, or get X.509 to add it.

15. Table of Contents