INTERNET-DRAFT D. W. Chadwick
PKIX WG University of Salford
Intended Category: Standards Track S. Legg
Adacel Technologies
26 June 2002

Internet X.509 Public Key Infrastructure LDAP Schema and Syntaxes for PKIs <draft-pkix-ldap-pki-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.

This Internet-Draft expires on 26 December 2002.

ABSTRACT

This document describes LDAP schema features that are needed to support X.509 Public Key Infrastructures. Specifically, X.509 attribute types, object classes, matching rules, attribute value syntaxes and attribute value assertion syntaxes needed for PKIs 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

RFC2587 [8] describes some of the PKI subschema applicable to LDAPv2 [2] servers, specifically the public key certificate related attribute types and object classes that MUST or MAY be supported. RFC 2256 [17] describes some of the PKI related subschema elements for LDAPv3 [4] servers. This [document/ID/standard] supercedes both RFC2587 and RFC 2256 and provides the complete PKI subschema for LDAP v3 [4] servers.

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. PKI Attributes and Syntaxes

3.1 userCertificate Attribute

The userCertificate attribute type contains the public-key certificates a user has obtained from one or more CAs. The LDAPspecific encoding for values of this attribute is described in section 3.3.

```
    ( 2.5.4.36 NAME 'userCertificate'
      EQUALITY certificateExactMatch
      SYNTAX 1.3.6.1.4.1.1466.115.121.1.8 )
```

3.2 cACertificate Attribute

The cACertificate attribute of a CA's directory entry shall be used to store self-issued certificates (if any) and certificates issued to this CA by CAs in the same realm as this CA. The LDAP-specific encoding for values of this attribute is described in section 3.3.

```
    ( 2.5.4.37 NAME 'cACertificate'
EQUALITY certificateExactMatch
      SYNTAX 1.3.6.1.4.1.1466.115.121.1.8 )
```

3.3 Certificate Syntax

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

```
      ( 1.3.6.1.4.1.1466.115.121.1.8 DESC 'A BER and/or DER encoded
public key certificate' )
```

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

Note. The BNF notation in RFC 1778 [12] for "User Certificate" MUST NOT be used. Values in this syntax MUST be transferred as BER and/or DER encoded octets.

3.4 authorityRevocationList Attribute

A value of this attribute is a list of CA certificates that are no longer valid. The LDAP-specific encoding for values of this attribute

is described in section 3.7.

```
    ( 2.5.4.38 NAME 'authorityRevocationList'
EQUALITY certificateListExactMatch
      SYNTAX 1.3.6.1.4.1.1466.115.121.1.9 )
```

3.5 certificateRevocationList Attribute

A value of this attribute is a list of user certificates that are no longer valid. The LDAP-specific encoding for values of this attribute is described in section 3.7.

```
    ( 2.5.4.39 NAME 'certificateRevocationList'
EQUALITY certificateListExactMatch
      SYNTAX 1.3.6.1.4.1.1466.115.121.1.9 )
```

3.6 deltaRevocationList Attribute

This attribute contains a list of revoked certificates (user or CA) that is an addition to a previous certificate revocation list. The LDAP-specific encoding for values of this attribute is described in section 3.7.

```
    ( 2.5.4.53 NAME 'deltaRevocationList'
EQUALITY certificateListExactMatch
      SYNTAX 1.3.6.1.4.1.1466.115.121.1.9 )
```

3.7 Certificate List Syntax

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

```
      ( 1.3.6.1.4.1.1466.115.121.1.9 DESC 'Certificate List' )
```

Servers MUST preserve values in this syntax exactly as given when storing and retrieving them. The BNF notation in RFC 1778 [12] for "Authority Revocation List" MUST NOT be used.

3.8 crossCertificatePair Attribute

The following definition is taken from X.509(2000) [9]. The term forward was used in earlier editions of X.509 for issuedToThisCA and the term reverse was used in earlier editions for issuedByThisCA.

The issuedToThisCA elements of the crossCertificatePair attribute of a CA's directory entry shall be used to store all, except self-issued certificates, issued to this CA. Optionally, the issuedByThisCA elements of the crossCertificatePair attribute, of a CA's directory entry may contain
a subset of certificates issued by this CA to other CAs. If a CA issues a certificate to another CA, and the subject CA is not a subordinate to the issuer CA in a hierarchy, then the issuer CA shall place that certificate in the issuedByThisCA element of the crossCertificatePair attribute
of its own directory entry. When both the issuedToThisCA and the issuedByThisCA elements are present in a single attribute value, issuer name in one certificate shall match the subject name in the other and vice versa, and the subject public key in one certificate shall be capable of verifying the digital signature on the other certificate and

vice versa.

The LDAP-specific encoding for values of this attribute is described in section 3.9.

```
    ( 2.5.4.40 NAME 'crossCertificatePair'
EQUALITY certificatePairExactMatch
      SYNTAX 1.3.6.1.4.1.1466.115.121.1.10 )
```

3.9 Certificate Pair Syntax

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

```
      ( 1.3.6.1.4.1.1466.115.121.1.10 DESC 'Certificate Pair' )
```

Servers MUST preserve values in this syntax exactly as given when storing and retrieving them. The BNF notation in RFC 1778 [12] for "Certificate Pair" MUST NOT be used. Servers must preserve values in this syntax exactly as given when storing and retrieving them.

3.10 PKI Path Attribute

The PKI path attribute is used to store certification paths, each consisting of a sequence of cross-certificates. The LDAP-specific encoding for values of this attribute is described in section 3.11.

```
( 2.5.4.70 NAME 'pkiPath'
      SYNTAX 1.2.826.0.1.3344810.7.19)
```

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

"This attribute can be stored in the CA directory entry and would contain some certification paths from that CA to other CAs. This attribute, if used, enables more efficient retrieval of cross-certificates that form frequently used certification 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 will likely not represent the complete set of forward certification paths for any given CA."

3.11 PKI Path Syntax

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

```
      ( 1.2.826.0.1.3344810.7.19 DESC 'PKI Path' )
```

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

3.12 CPS Attribute

The CPS attribute is used to store a certification authority's certification practice statement.

```
(1.2.826.0.1.3344810.1.1.31 NAME 'cps'
SUBSTR caseIgnoreSubstringsMatch
      SYNTAX 1.3.6.1.4.1.1466.115.121.1.15)
```

3.13 CPS Pointer Attribute

The CPS pointer attribute is used to store a pointer to a certification authority's certification practice statement in the form of a URI.

```
(1.2.826.0.1.3344810.1.1.32 NAME 'cpsPointer'
      SYNTAX 1.3.6.1.4.1.1466.115.121.1.26)
```

3.14 Certificate Policy Attribute

The certificatePolicy attribute is used to store information about a certification authority's certificate policy (either directly or indirectly). The LDAP-specific encoding for values of this attribute is described in section 3.15.

```
( 2.5.4.69 NAME 'certificatePolicy'
EQUALITY objectIdentifierFirstComponentMatch
      SYNTAX 1.2.826.0.1.3344810.7.20)
```

3.15 Certificate Policy Syntax

The LDAP-specific encoding for a certificate policy value is the octet string that results from the BERencoding of a sequence of the policy object identifier and policy information. The following string states the OID assigned to this syntax:

```
      ( 1.2.826.0.1.3344810.7.20 DESC 'CA certificate policy' )
```

3.16 Certificate Policy Pointer Attribute

The CP pointer attribute is used to store a pointer to a certification authority's certificate policy in the form of a URI.

```
(1.2.826.0.1.3344810.1.1.33 NAME 'cpPointer'
      SYNTAX 1.3.6.1.4.1.1466.115.121.1.26)
```

3.17 Supported Algorithms Attribute

This attribute is used to support the selection of an algorithm for use when communicating with a remote end entity using certificates. The LDAP-specific encoding for values of this attribute is described in section 3.17.

```
    ( 2.5.4.52 NAME 'supportedAlgorithms'
EQUALITY objectIdentifierFirstComponentMatch
      SYNTAX 1.3.6.1.4.1.1466.115.121.1.49 )
```

3.18 Supported Algorithm Syntax

The LDAP-specific encoding for a supported algorithm value is the octet string that results from the BER encoding of a SupportedAlgorithm ASN.1 value. The following string states the OID assigned to this syntax:

```
      ( 1.3.6.1.4.1.1466.115.121.1.49 DESC 'Supported Algorithm' )
```

4. Public Key Certificate Matching Rules and Assertion Syntaxes

X.509 [9] supports both equality and flexible certificate matching rules by the server, via the certificateExactMatch and certificateMatch MATCHING-RULEs respectively. (For example, a client may flexibly search for certificates with a particular validity time, key usage, policy or other field.) LDAP servers MUST support the certificateExactMatch matching rule. Clients MAY support certificateExactMatch values for equalityMatch filters. LDAPv3 servers SHOULD support the certificateMatch matching rule. If the server does support flexible matching (either via certificateMatch or some other matching rule), then the extensibleMatch filter of the Search request MUST be supported. Clients MAY support the extensibleMatch filter and one or more of the optional elements of certificateMatch.

The LDAP-specific (i.e. string) encodings for the assertion syntaxes defined in this document are specified by the Generic String Encoding Rules (GSER) [13]. 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 [13]. (The only exception to this is the alternative simple endoding for certificatExactMatch.) 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 [13]. In the event that there is a discrepancy between the ABNF in this document and the encoding determined by [13], [13] is to be taken as definitive.

4.1 Certificate Exact Match

Certificate exact match is defined in 11.3.1 of [9]. The string description of the certificateExactMatch matching rule is:

```
    ( 2.5.13.34 NAME 'certificateExactMatch'
    SYNTAX 1.2.826.0.1.3344810.7.1 )
```

The LDAP syntax definition of the above is:

```
      (1.2.826.0.1.3344810.7.1
DESC 'Certificate Serial Number and Issuer Name' )
```

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

- the GSER encoding defined by [13]<GSERCertificateExactAssertion> and
- the simple encoding defined by <SimpleCertificateExactAssertion>.

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

```
CertificateExactAssertion = GSERCertificateExactAssertion /
                            SimpleCertificateExactAssertion

SimpleCertificateExactAssertion = CertificateSerialNumber "$" LDAPDN
```

<LDAPDN> is a string encoding of a distinguished name as defined in [6].

```
GSERCertificateExactAssertion = "{" sp cea-serialNumber ","
                                sp cea-issuer
                                sp "}"

cea-serialNumber = id-serialNumber msp CertificateSerialNumber
```

```
cea-issuer       = id-issuer       msp Name

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

Name           = id-rdnSequence ":" RDNSequence
id-rdnSequence = %x72.64.6E.53.65.71.75.65.6E.63.65 ; "rdnSequence"

CertificateSerialNumber = INTEGER
```

Note. [14] states that CAs MUST force the serialNumber to be a non-negative integer. Non-conforming CAs MAY issue certificates with serial numbers that are negative, or zero. Certificate users SHOULD be prepared to handle such certificates.

The <sp>, <msp>, <RDNSequence> and <INTEGER> rules are given in [16].

4.2 Certificate Match

Certificate match is defined in 11.3.2 of [9]. The string description of the certificateMatch matching rule is:

```
( 2.5.13.35 NAME 'certificateMatch'
    SYNTAX 1.2.826.0.1.3344810.7.2)
```

The syntax definition is:

```
(1.2.826.0.1.3344810.7.2 DESC 'Certificate Assertion' )
```

The ASN.1 for CertificateAssertion is defined in 11.3.2 of [9], as are the semantics of each of its component types.

The LDAP-specific encoding of an assertion value of this syntax is defined by [13] and described by the following ABNF:

```
CertificateAssertion = "{"    [ sp ca-serialNumber ]
                          [ sep sp ca-issuer ]
                          [ sep sp ca-subjectKeyIdentifier ]
                          [ sep sp ca-authorityKeyIdentifier ]
                          [ sep sp ca-certificateValid ]
                          [ sep sp ca-privateKeyValid ]
                          [ sep sp ca-subjectPublicKeyAlgID ]
                          [ sep sp ca-keyUsage ]
                          [ sep sp ca-subjectAltName ]
                          [ sep sp ca-policy ]
                          [ sep sp ca-pathToName ]
                          [ sep sp ca-subject ]
                          [ sep sp ca-nameConstraints ]
                               sp "}"
```

The <sep> rule is given in [16].

```
ca-serialNumber           = id-serialNumber msp
                               CertificateSerialNumber
ca-issuer                 = id-issuer msp Name
ca-subjectKeyIdentifier   = id-subjectKeyIdentifier msp
                               SubjectKeyIdentifier
ca-authorityKeyIdentifier = id-authorityKeyIdentifier msp
                               AuthorityKeyIdentifier
```

```
ca-certificateValid       = certificateValid msp Time
ca-privateKeyValid        = id-privateKeyValid msp GeneralizedTime
ca-subjectPublicKeyAlgID  = id-subjectPublicKeyAlgID msp
                               OBJECT-IDENTIFIER
ca-keyUsage               = id-keyUsage msp KeyUsage
ca-subjectAltName         = id-subjectAltName msp AltNameType
ca-policy                 = id-policy msp CertPolicySet
ca-pathToName             = id-pathToName msp Name
ca-subject                = id-subject msp Name
ca-nameConstraints        = id-nameConstraints msp
                               NameConstraintsSyntax

id-subjectKeyIdentifier   = %x73.75.62.6A.65.63.74.4B.65.79.49.64.65
                               %x6E.74.69.66.69.65.72
                               ; "subjectKeyIdentifier"
id-authorityKeyIdentifier = %x61.75.74.68.6F.72.69.74.79.4B.65.79.49
                               %x64.65.6E.74.69.66.69.65.72
                               ; "authorityKeyIdentifier"
id-certificateValid       = %x63.65.72.74.69.66.69.63.61.74.65.56.61
                               %x6C.69.64 ; "certificateValid"
id-privateKeyValid        = %x70.72.69.76.61.74.65.4B.65.79.56.61.6C
                               %x69.64 ; "privateKeyValid"
id-subjectPublicKeyAlgID  = %x73.75.62.6A.65.63.74.50.75.62.6C.69.63
                               %x4B.65.79.41.6C.67.49.44
                               ; "subjectPublicKeyAlgID"
id-keyUsage               = %x6B.65.79.55.73.61.67.65 ; "keyUsage"
id-subjectAltName         = %x73.75.62.6A.65.63.74.41.6C.74.4E.61.6D
                               %x65 ; "subjectAltName"
id-policy                 = %x70.6F.6C.69.63.79 ; "policy"
id-pathToName             = %x70.61.74.68.54.6F.4E.61.6D.65
                               ; "pathToName"
id-subject                = %x73.75.62.6A.65.63.74 ; "subject"
id-nameConstraints        = %x6E.61.6D.65.43.6F.6E.73.74.72.61.69.6E
                               %x74.73 ; "nameConstraints"

SubjectKeyIdentifier = KeyIdentifier

KeyIdentifier = OCTET-STRING

AuthorityKeyIdentifier = "{"    [ sp aki-keyIdentifier ]
                           [ sep sp aki-authorityCertIssuer ]
                           [ sep sp aki-authorityCertSerialNumber ]
                                 sp "}"

aki-keyIdentifier       = id-keyIdentifier msp KeyIdentifier
aki-authorityCertIssuer = id-authorityCertIssuer msp GeneralNames

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"
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 [13].

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"

aki-authorityCertSerialNumber = id-authorityCertSerialNumber msp
                                   CertificateSerialNumber

id-keyIdentifier       = %x6B.65.79.49.64.65.6E.74.69.66.69.65.72
                            ; "keyIdentifier"
id-authorityCertIssuer = %x61.75.74.68.6F.72.69.74.79.43.65.72.74.49
                           %x73.73.75.65.72 ; "authorityCertIssuer"

id-authorityCertSerialNumber = %x61.75.74.68.6F.72.69.74.79.43.65.72
                                  %x74.53.65.72.69.61.6C.4E.75.6D.62
                                  %x65.72
                                  ; "authorityCertSerialNumber"

Time                 = time-utcTime / time-generalizedTime
time-utcTime         = id-utcTime         ":" UTCTime
time-generalizedTime = id-generalizedTime ":" GeneralizedTime
id-utcTime           = %x75.74.63.54.69.6D.65 ; "utcTime"
id-generalizedTime   = %x67.65.6E.65.72.61.6C.69.7A.65.64.54.69.6D.65
                          ; "generalizedTime"

KeyUsage           = BIT-STRING / key-usage-bit-list
key-usage-bit-list = "{" [ sp key-usage *( "," sp key-usage ) ] sp "}"

The <key-usage-bit-list> rule encodes the one bits in a KeyUsage value
as a comma separated list of identifiers. The <BIT-STRING> rule is given
in [16].

key-usage = id-digitalSignature
            / id-nonRepudiation
            / id-keyEncipherment
            / id-dataEncipherment
            / id-keyAgreement
            / id-keyCertSign
```

```
            / id-cRLSign
            / id-encipherOnly
            / id-decipherOnly

id-digitalSignature = %x64.69.67.69.74.61.6C.53.69.67.6E.61.74.75.72
                      %x65 ; "digitalSignature"
id-nonRepudiation   = %x6E.6F.6E.52.65.70.75.64.69.61.74.69.6F.6E
                      ; "nonRepudiation"
id-keyEncipherment  = %x6B.65.79.45.6E.63.69.70.68.65.72.6D.65.6E.74
                      ; "keyEncipherment"
id-dataEncipherment = %x64.61.74.61.45.6E.63.69.70.68.65.72.6D.65.6E
                      %x74 ; "dataEncipherment"
id-keyAgreement     = %x6B.65.79.41.67.72.65.65.6D.65.6E.74
                      ; "keyAgreement"
id-keyCertSign      = %x6B.65.79.43.65.72.74.53.69.67.6E
                      ; "keyCertSign"
id-cRLSign          = %x63.52.4C.53.69.67.6E ; "cRLSign"
id-encipherOnly     = %x65.6E.63.69.70.68.65.72.4F.6E.6C.79
                      ; "encipherOnly"
id-decipherOnly     = %x64.65.63.69.70.68.65.72.4F.6E.6C.79
                      ; "decipherOnly"

AltNameType = ant-builtinNameForm / ant-otherNameForm

ant-builtinNameForm = id-builtinNameForm ":" BuiltinNameForm
ant-otherNameForm   = id-otherNameForm   ":" OBJECT-IDENTIFIER

id-builtinNameForm = %x62.75.69.6C.74.69.6E.4E.61.6D.65.46.6F.72.6D
                     ; "builtinNameForm"
id-otherNameForm   = %x6F.74.68.65.72.4E.61.6D.65.46.6F.72.6D
                     ; "otherNameForm"

BuiltinNameForm  = id-rfc822Name
                 / id-dNSName
                 / id-x400Address
                 / id-directoryName
                 / id-ediPartyName
                 / id-uniformResourceIdentifier
                 / id-iPAddress
                 / id-registeredId
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"

CertPolicySet = "{" sp CertPolicyId *( "," sp CertPolicyId ) sp "}"
CertPolicyId  = OBJECT-IDENTIFIER

NameConstraintsSyntax = "{"    [ sp ncs-permittedSubtrees ]
                         [ sep sp ncs-excludedSubtrees ]
```

```
                                sp "}"

ncs-permittedSubtrees = id-permittedSubtrees msp GeneralSubtrees
ncs-excludedSubtrees  = id-excludedSubtrees  msp GeneralSubtrees
id-permittedSubtrees  = %x70.65.72.6D.69.74.74.65.64.53.75.62.74.72
                           %x65.65.73 ; "permittedSubtrees"
id-excludedSubtrees   = %x65.78.63.6C.75.64.65.64.53.75.62.74.72.65
                           %x65.73 ; "excludedSubtrees"

GeneralSubtrees = "{" sp GeneralSubtree
                     *( "," sp GeneralSubtree ) sp "}"
GeneralSubtree  = "{"      sp gs-base
                      [ "," sp gs-minimum ]
                      [ "," sp gs-maximum ]
                           sp "}"

gs-base      = id-base    msp GeneralName
gs-minimum   = id-minimum msp BaseDistance
gs-maximum   = id-maximum msp BaseDistance
id-base      = %x62.61.73.65          ; "base"
id-minimum   = %x6D.69.6E.69.6D.75.6D ; "minimum"
id-maximum   = %x6D.61.78.69.6D.75.6D ; "maximum"
BaseDistance = INTEGER-0-MAX
```

The <OBJECT-IDENTIFIER>, <OCTET-STRING>, <IA5String>, <DirectoryString>, <RelativeDistinguishedName>, <UTCTime>, <GeneralizedTime>, <INTEGER-0-MAX> and <ORAddress> rules are given in [16].

4.3 Certificate Pair Exact Match

Certificate pair exact match is defined in 11.3.3 of [9]. The string description of the certificatePairExactMatch matching rule is:

```
( 2.5.13.36 NAME 'certificatePairExactMatch'
    SYNTAX 1.2.826.0.1.3344810.7.8)
```

The LDAP syntax definition is:

```
(1.2.826.0.1.3344810.7.8
    DESC 'Certificate Pair Exact Assertion' )
```

The ASN.1 for CertificatePairExactAssertion is defined in 11.3.3 of [9], as are the semantics of each of its component types.

The LDAP-specific encoding of an assertion value of this syntax is defined by [13] and described by the following Augmented BNF [10]:

```
CertificatePairExactAssertion = "{"   [ sp cpea-issuedTo ]
                                    [sep sp cpea-issuedBy ]
                                         sp "}"
```

At least one of <cpea-issuedTo> or <cpea-issuedBy> MUST be present.

```
cpea-issuedTo = id-issuedToThisCAAssertion msp
                   CertificateExactAssertion
cpea-issuedBy = id-issuedByThisCAAssertion msp
                   CertificateExactAssertion

id-issuedToThisCAAssertion = %x69.73.73.75.65.64.54.6F.54.68.69.73.43
                               %x41.41.73.73.65.72.74.69.6F.6E
```

```
                                ; "issuedToThisCAAssertion"
id-issuedByThisCAAssertion = %x69.73.73.75.65.64.42.79.54.68.69.73.43
                                %x41.41.73.73.65.72.74.69.6F.6E
                                ; "issuedByThisCAAssertion"
```

4.4 Certificate Pair Match

Certificate pair match is defined in 11.3.4 of [9]. The string description of the certificatePairMatch matching rule is:

```
( 2.5.13.37 NAME 'certificatePairExactMatch'
    SYNTAX 1.2.826.0.1.3344810.7.9)
```

The LDAP syntax definition is:

```
(1.2.826.0.1.3344810.7.9
    DESC 'Certificate Pair Assertion' )
```

The ASN.1 for CertificatePairAssertion is defined in 11.3.4 of [9], as are the semantics of each of its component types.

The LDAP-specific encoding of an assertion value of this syntax is defined by [13] and described by the following Augmented BNF [10]:

```
CertificatePairAssertion = "{"   [ sp cpa-issuedTo ]
                               [sep sp cpa-issuedBy ]
                                    sp "}"
```

At least one of <cpa-issuedTo> and <cpa-issuedBy> MUST be present.

```
cpa-issuedTo = id-issuedToThisCAAssertion msp CertificateAssertion
cpa-issuedBy = id-issuedByThisCAAssertion msp CertificateAssertion
```

5 Certificate Revocation List Matching Rules

X.509[9] defines both equality and flexible matching rules for CRLs, via the certificateListExactMatch and certificateListMatch MATCHING-RULEs respectively. LDAP servers MUST support the certificateListExactMatch matching rule. Clients MAY support certificateListExactMatch values for equalityMatch filters. LDAPv3 servers MAY support the certificateListMatch matching rule. If the server does support flexible matching (either via certificateListMatch or some other matching rule), then the extensibleMatch filter of the Search request MUST be supported. Clients MAY support the extensibleMatch filter and one or more of the optional elements of certificateListMatch.

5.1 Certificate List Exact Match

Certificate List exact match is defined in 11.3.5 of [9]. The string description of the certificateListExactMatch matching rule is:

```
( 2.5.13.38 NAME 'certificateListExactMatch'
    SYNTAX 1.2.826.0.1.3344810.7.3)
```

The syntax definition is:

```
(1.2.826.0.1.3344810.7.3 DESC 'Certificate List Exact Assertion (Issuer
```

```
name, time and distribution point name)' )

The ASN.1 for CertificateListExactAssertion is defined in 11.3.5 of [9],
as are the semantics of each of its component types.

The LDAP-specific encoding of an assertion value of this syntax is
defined by [13] and described
by the following ABNF:

CertificateListExactAssertion = "{"      sp clea-issuer
                                      "," sp clea-thisUpdate
                                    [ "," sp clea-distributionPoint ]
                                      sp "}"

clea-issuer            = id-issuer msp Name
clea-thisUpdate        = id-thisUpdate msp Time
clea-distributionPoint = id-distributionPoint msp
                           DistributionPointName

id-thisUpdate        = %x74.68.69.73.55.70.64.61.74.65
                       ; "thisUpdate"
id-distributionPoint = %x64.69.73.74.72.69.62.75.74.69.6F.6E
                       %x50.6F.69.6E.74 ; "distributionPoint"

DistributionPointName = dpn-fullName / dpn-nameRelativeToCRLIssuer

dpn-fullName                = id-fullName ":" GeneralNames
dpn-nameRelativeToCRLIssuer = id-nameRelativeToCRLIssuer ":"
                              RelativeDistinguishedName

id-fullName                = %x66.75.6C.6C.4E.61.6D.65 ; "fullName"
id-nameRelativeToCRLIssuer = %x6E.61.6D.65.52.65.6C.61.74.69.76.65
                             %x54.6F.43.52.4C.49.73.73.75.65.72
                             ; "nameRelativeToCRLIssuer"
```

5.2 Certificate List Match

Certificate List match is defined in 11.3.6 of [9]. The string description of the certificateListMatch matching rule is:

```
( 2.5.13.39 NAME 'certificateListMatch'
    SYNTAX 1.2.826.0.1.3344810.7.4)
```

The syntax definition is:

```
(1.2.826.0.1.3344810.7.4 DESC 'Certificate List Assertion' )
```

The ASN.1 for CertificateListAssertion is defined in 11.3.6 of [9], as are the semantics of its components.

The LDAP-specific encoding of an assertion value of this syntax is defined by [13] and describedby the following ABNF:

```
CertificateListAssertion = "{"    [ sp cla-issuer ]
                                [ sep sp cla-minCRLNumber ]
                                [ sep sp cla-maxCRLNumber ]
                                [ sep sp cla-reasonFlags ]
                                [ sep sp cla-dateAndTime ]
                                [ sep sp cla-distributionPoint ]
                                [ sep sp cla-authorityKeyIdentifier ]
```

```
                                    sp "}"

cla-issuer       = id-issuer       msp Name
cla-minCRLNumber = id-minCRLNumber msp CRLNumber
cla-maxCRLNumber = id-maxCRLNumber msp CRLNumber
cla-reasonFlags  = id-reasonFlags  msp ReasonFlags
cla-dateAndTime  = id-dateAndTime  msp Time

cla-distributionPoint      = id-distributionPoint msp
                                DistributionPointName
cla-authorityKeyIdentifier = id-authorityKeyIdentifier msp
                                AuthorityKeyIdentifier

id-minCRLNumber = %x6D.69.6E.43.52.4C.4E.75.6D.62.65.72
                    ; "minCRLNumber"
id-maxCRLNumber = %x6D.61.78.43.52.4C.4E.75.6D.62.65.72
                    ; "maxCRLNumber"
id-reasonFlags  = %x72.65.61.73.6F.6E.46.6C.61.67.73 ; "reasonFlags"
id-dateAndTime  = %x64.61.74.65.41.6E.64.54.69.6D.65 ; "dateAndTime"

CRLNumber = INTEGER-0-MAX

ReasonFlags = BIT-STRING
              / "{" [ sp reason-flag
                   *( "," sp reason-flag ) ] sp "}"

reason-flag = id-unused
              / id-keyCompromise
              / id-cACompromise
              / id-affiliationChanged
              / id-superseded
              / id-cessationOfOperation
              / id-certificateHold
              / id-privilegeWithdrawn
              / id-aACompromise

id-unused               = %x75.6E.75.73.65.64 ; "unused"
id-keyCompromise        = %x6B.65.79.43.6F.6D.70.72.6F.6D.69.73.65
                             ; "keyCompromise"
id-cACompromise         = %x63.41.43.6F.6D.70.72.6F.6D.69.73.65
                             ; "cACompromise"
id-affiliationChanged   = %x61.66.66.69.6C.69.61.74.69.6F.6E.43.68
                             %x61.6E.67.65.64 ; "affiliationChanged"
id-superseded           = %x73.75.70.65.72.73.65.64.65.64
                             ; "superseded"
id-cessationOfOperation = %x63.65.73.73.61.74.69.6F.6E.4F.66.4F.70
                             %x65.72.61.74.69.6F.6E
                             ; "cessationOfOperation"
id-certificateHold      = %x63.65.72.74.69.66.69.63.61.74.65.48.6F
                             %x6C.64 ; "certificateHold"
id-privilegeWithdrawn   = %x70.72.69.76.69.6C.65.67.65.57.69.74.68
                             %x64.72.61.77.6E ; "privilegeWithdrawn"
id-aACompromise         = %x61.41.43.6F.6D.70.72.6F.6D.69.73.65
                             ; "aACompromise"
```

6. PKI Object Classes

6.1 PKI user object class

The PKI user object class MAY be used in defining entries for objects

that may be the subject of public-key certificates.

```
( 2.5.6.21 NAME 'pkiUser' SUP top AUXILIARY
MAY userCertificate )
```

6.2 PKI CA object class

The PKI CA object class MAY be used in defining entries for objects that act as certification authorities.

```
( 2.5.6.22 NAME 'pkiCA' SUP top AUXILIARY
MAY ( cACertificate $ certificateRevocationList $
authorityRevocationList $ crossCertificatePair ) )
```

6.3 CRL Distribution Point object class

The CRL Distribution Point object class MAY be used in defining entries for objects which act as CRL Distribution Points

```
( 2.5.6.19 NAME 'cRLDistributionPoint' SUP top STRUCTURAL MUST cn
MAY (certificateRevocationList $ authorityRevocationList $
DeltaRevocationList ) )
```

6.4 Delta CRL object class

The delta CRL object class is used in defining entries for objects that hold delta revocation lists (e.g. CAs, AAs etc.).

```
( 2.5.6.23 NAME 'deltaCRL' SUP top AUXILIARY
MAY deltaRevocationList )
```

6.5 Certificate Policy and CPS object class

The CP CPS object class MAY be used in defining entries for objects that contain certificate policy and / or certification practice information

```
( 2.5.6.30 NAME 'cpCPS' SUP top AUXILIARY MAY ( certificatePolicy $
certificationPracticeStmt ) )
```

6.6 PKI Certification Path object class

The PKI certification path object class MAY be used in defining entries for objects that contain PKI certification paths. It will generally be used in conjunction with entries of structural object class pkiCA.

```
( 2.5.6.31 NAME 'pkiCertPath' SUP top AUXILIARY MAY pkiPath)
```

7. Filter Examples

The following examples are written using the string representation of Search filters defined in [18]. Line-breaks have been added as an aid to readability.

i) To match on the serial number of a PKI certificate using extensibleMatch with component matching

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

ii) To exactly match one certificate using extensibleMatch with certificateExactMatch and GSERCertificateExactAssertion

```
(userCertificate:certificateExactMatch:= {serialNumber 12345 ,
issuer rdnSequence: "O=truetrust ltd, C=GB" } )
```

iii) To exactly match one certificate using equalityMatch with certificateExactMatch and GSERCertificateExactAssertion

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

iv) To exactly match one certificate using equalityMatch with certificateExactMatch and SimpleCertificateExactAssertion

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

v) To exactly match one certificate using extensibleMatch with component matching

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

vi) To match on certificates containing a certain email address as a subjectAltName

```
(userCertificate:componentFilterMatch:=item:{
component "toBeSigned.extensions.*.extnValue.
content.(2.5.29.17).*.rfc822Name",
rule caseIgnoreIA5Match, value "person@email.address.com" })
```

8. Security Considerations

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

LDAP servers SHOULD use access control information to protect the information during its storage. In addition, clients MAY choose to encrypt the attributes in the attribute certificates before storing them in an LDAP server.

9. References

Normative

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

[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] M. Wahl, S. Kille, T. Howes. "Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names", RFC2253, December 1997.

[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

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

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

Informative

[2] Yeong, W., Howes, T., and Kille, S. "Lightweight Directory Access Protocol", RFC 1777, March 1995.

[8] S.Boeyen, T. Howes, P. Richard "Internet X.509 Public Key Infrastructure, LDAPv2 Schema", RFC 2587, June 1999

[12] Howes, T., Kille, S., Yeong, W., Robbins, C., "The String Representation of Standard Attribute Syntaxes", RFC 1778, March 1995

[14] R. Housley, W. Ford, W Polk, D. Solo. "Internet X.509 Public Key Infrastructure - Certificate and CRL Profile" <draft-ietf-pkix-new-part1-08.txt>, July 2001

[17] M. Wahl, "A Summary of the X.500(96) User Schema for use with LDAPv3", RFC 2256, Dec 1997

[18] Howes, T. "The String Representation of LDAP Search Filters". RFC 2254, December 1997.

10. Intellectual Property Notice

11. Copyright

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

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 <draft-pkix-ldap-schema-01.txt>

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 <draft-pkix-ldap-schema-02.txt>

i) Separation in PKI and PMI IDs.
ii) Examples of filters have been added
iii) Text has been added to mandate that servers must store and retrieve many of the syntaxes defined in this ID exactly as given.
iv) The ;binary encoding option has been removed in accordance with work in the LDAPBIS group. A new LDAP-specific encoding has been defined which has exactly the same syntax as the old ;binary encoding.
v) We have obsoleted RFC 2587 and RFC 2256 and copied the relevant schemas into this document.
vi) We have added some new PKI schema appearing for the first time in X.509(2000) e.g. pkiPath

14. Outstanding Issues

i. We need to decide if userSMIMECertificates should also be supported as part of this profile or not.
ii. We have added a CPS attribute and a CPS pointer attribute. These are adapted from the certificationPracticeStmt attribute in the X.509 standard which is a choice of either the CPS or a pointer to it. However our pointer is simply a URI (as in the CPS qualifier extension in the PKIX profile) whereas the X.500 pointer is a GeneralName and an optional hash. Are these changes sensible and acceptable?
iii. We have added a matching rule to the certificatePolicy attribute. No matching rule is defined in X.509, so we have reported this as a defect. Should we stick with the X.509 syntax or create two alternative attributes (a pointer and a policy) as in the CPS case.
iv. We have made the matching rule for supportedAlgorithms as the objectIdentifierFirstComponentMatch. RFC2256 did not specify any matching rule and X.509(2001) specifies a more complex matching rule. Should we align with X.509 or not?
15. Table of Contents