Network Working Group
Request for Comments: 4523
Obsoletes: 2252, 2256, 2587
Category: Standards Track

K. Zeilenga
OpenLDAP Foundation

June 2006

Lightweight Directory Access Protocol (LDAP) Schema Definitions for X.509 Certificates

Status of This Memo

Copyright Notice

Abstract

This document describes schema for representing X.509 certificates, X.521 security information, and related elements in directories accessible using the Lightweight Directory Access Protocol (LDAP). The LDAP definitions for these X.509 and X.521 schema elements replace those provided in RFCs 2252 and 2256.

1. Introduction

This document provides LDAP [RFC4510] schema definitions [RFC4512] for a subset of elements specified in X.509 [X.509] and X.521 [X.521], including attribute types for certificates, cross certificate pairs, and certificate revocation lists; matching rules to be used with these attribute types; and related object classes. LDAP syntax definitions are also provided for associated assertion and attribute values.

As the semantics of these elements are as defined in X.509 and X.521, knowledge of X.509 and X.521 is necessary to make use of the LDAP schema definitions provided herein.

This document, together with [RFC4510], obsoletes RFCs 2252 and 2256 in their entirety. The changes (in this document) made since RFC 2252 and RFC 2256 include:

- addition of pkiUser, pkiCA, and deltaCRL classes;

- update of attribute types to include equality matching rules in accordance with their X.500 specifications;

- addition of certificate, certificate pair, certificate list, and algorithm identifier matching rules; and

- addition of LDAP syntax for assertion syntaxes for these matching rules.

This document obsoletes RFC 2587. The X.509 schema descriptions for LDAPv2 [RFC1777] are Historic, as is LDAPv2 [RFC3494].

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 BCP 14 [RFC2119].

Schema definitions are provided using LDAP description formats [RFC4512]. Definitions provided here are formatted (line wrapped) for readability.

2. Syntaxes

This section describes various syntaxes used in LDAP to transfer certificates and related data types.

2.1. Certificate

(1.3.6.1.4.1.1466.115.121.1.8 DESC 'X.509 Certificate')

A value of this syntax is an X.509 Certificate [X.509, clause 7].

Due to changes made to the definition of a Certificate through time, no LDAP-specific encoding is defined for this syntax. Values of this syntax SHOULD be encoded using Distinguished Encoding Rules (DER) [X.690] and MUST only be transferred using the ;binary transfer option [RFC4522]; that is, by requesting and returning values using attribute descriptions such as "userCertificate;binary".

As values of this syntax contain digitally signed data, values of this syntax and the form of each value MUST be preserved as presented.

2.2. CertificateList

(1.3.6.1.4.1.1466.115.121.1.9 DESC 'X.509 Certificate List')

A value of this syntax is an X.509 CertificateList [X.509, clause 7.3].

Due to changes made to the definition of a CertificateList through time, no LDAP-specific encoding is defined for this syntax. Values of this syntax SHOULD be encoded using DER [X.690] and MUST only be transferred using the ;binary transfer option [RFC4522]; that is, by requesting and returning values using attribute descriptions such as "certificateRevocationList;binary".

As values of this syntax contain digitally signed data, values of this syntax and the form of each value MUST be preserved as presented.

2.3. CertificatePair

```
( 1.3.6.1.4.1.1466.115.121.1.10 DESC 'X.509 Certificate Pair' )
```

A value of this syntax is an X.509 CertificatePair [X.509, clause 11.2.3].

Due to changes made to the definition of an X.509 CertificatePair through time, no LDAP-specific encoding is defined for this syntax. Values of this syntax SHOULD be encoded using DER [X.690] and MUST only be transferred using the ;binary transfer option [RFC4522]; that is, by requesting and returning values using attribute descriptions such as "crossCertificatePair;binary".

As values of this syntax contain digitally signed data, values of this syntax and the form of each value MUST be preserved as presented.

2.4. SupportedAlgorithm

```
( 1.3.6.1.4.1.1466.115.121.1.49
     DESC 'X.509 Supported Algorithm' )
```

A value of this syntax is an X.509 SupportedAlgorithm [X.509, clause 11.2.7].

Due to changes made to the definition of an X.509 SupportedAlgorithm through time, no LDAP-specific encoding is defined for this syntax. Values of this syntax SHOULD be encoded using DER [X.690] and MUST only be transferred using the ;binary transfer option [RFC4522]; that is, by requesting and returning values using attribute descriptions such as "supportedAlgorithms;binary".

As values of this syntax contain digitally signed data, values of this syntax and the form of the value MUST be preserved as presented.

2.5. CertificateExactAssertion

(1.3.6.1.1.15.1 DESC 'X.509 Certificate Exact Assertion')

A value of this syntax is an X.509 CertificateExactAssertion [X.509, clause 11.3.1]. Values of this syntax MUST be encoded using the Generic String Encoding Rules (GSER) [RFC3641]. Appendix A.1 provides an equivalent Augmented Backus-Naur Form (ABNF) [RFC4234] grammar for this syntax.

2.6. CertificateAssertion

(1.3.6.1.1.15.2 DESC 'X.509 Certificate Assertion')

A value of this syntax is an X.509 CertificateAssertion [X.509, clause 11.3.2]. Values of this syntax MUST be encoded using GSER [RFC3641]. Appendix A.2 provides an equivalent ABNF [RFC4234] grammar for this syntax.

2.7. CertificatePairExactAssertion

(1.3.6.1.1.15.3
DESC 'X.509 Certificate Pair Exact Assertion')

A value of this syntax is an X.509 CertificatePairExactAssertion [X.509, clause 11.3.3]. Values of this syntax MUST be encoded using GSER [RFC3641]. Appendix A.3 provides an equivalent ABNF [RFC4234] grammar for this syntax.

2.8. CertificatePairAssertion

(1.3.6.1.1.15.4 DESC 'X.509 Certificate Pair Assertion')

A value of this syntax is an X.509 CertificatePairAssertion [X.509, clause 11.3.4]. Values of this syntax MUST be encoded using GSER [RFC3641]. Appendix A.4 provides an equivalent ABNF [RFC4234] grammar for this syntax.

2.9. CertificateListExactAssertion

(1.3.6.1.1.15.5
DESC 'X.509 Certificate List Exact Assertion')

A value of this syntax is an X.509 CertificateListExactAssertion [X.509, clause 11.3.5]. Values of this syntax MUST be encoded using GSER [RFC3641]. Appendix A.5 provides an equivalent ABNF grammar for this syntax.

2.10. CertificateListAssertion

```
( 1.3.6.1.1.15.6 DESC 'X.509 Certificate List Assertion' )
```

A value of this syntax is an X.509 CertificateListAssertion [X.509, clause 11.3.6]. Values of this syntax MUST be encoded using GSER [RFC3641]. Appendix A.6 provides an equivalent ABNF [RFC4234] grammar for this syntax.

2.11. AlgorithmIdentifier

```
( 1.3.6.1.1.15.7 DESC 'X.509 Algorithm Identifier' )
```

A value of this syntax is an X.509 AlgorithmIdentifier [X.509, Clause 7]. Values of this syntax MUST be encoded using GSER [RFC3641].

Appendix A.7 provides an equivalent ABNF [RFC4234] grammar for this syntax.

3. Matching Rules

This section introduces a set of certificate and related matching rules for use in LDAP. These rules are intended to act in accordance with their X.500 counterparts.

3.1. certificateExactMatch

The certificateExactMatch matching rule compares the presented certificate exact assertion value with an attribute value of the certificate syntax as described in clause 11.3.1 of [X.509].

```
( 2.5.13.34 NAME 'certificateExactMatch'
     DESC 'X.509 Certificate Exact Match'
     SYNTAX 1.3.6.1.1.15.1 )
```

3.2. certificateMatch

The certificateMatch matching rule compares the presented certificate assertion value with an attribute value of the certificate syntax as described in clause 11.3.2 of [X.509].

```
( 2.5.13.35 NAME 'certificateMatch'
     DESC 'X.509 Certificate Match'
     SYNTAX 1.3.6.1.1.15.2 )
```

3.3. certificatePairExactMatch

The certificatePairExactMatch matching rule compares the presented certificate pair exact assertion value with an attribute value of the certificate pair syntax as described in clause 11.3.3 of [X.509].

```
( 2.5.13.36 NAME 'certificatePairExactMatch'
     DESC 'X.509 Certificate Pair Exact Match'
     SYNTAX 1.3.6.1.1.15.3 )
```

3.4. certificatePairMatch

The certificatePairMatch matching rule compares the presented certificate pair assertion value with an attribute value of the certificate pair syntax as described in clause 11.3.4 of [X.509].

```
( 2.5.13.37 NAME 'certificatePairMatch'
     DESC 'X.509 Certificate Pair Match'
     SYNTAX 1.3.6.1.1.15.4 )
```

3.5. certificateListExactMatch

The certificateListExactMatch matching rule compares the presented certificate list exact assertion value with an attribute value of the certificate pair syntax as described in clause 11.3.5 of [X.509].

```
( 2.5.13.38 NAME 'certificateListExactMatch'
     DESC 'X.509 Certificate List Exact Match'
     SYNTAX 1.3.6.1.1.15.5 )
```

3.6. certificateListMatch

The certificateListMatch matching rule compares the presented certificate list assertion value with an attribute value of the certificate pair syntax as described in clause 11.3.6 of [X.509].

```
( 2.5.13.39 NAME 'certificateListMatch'
     DESC 'X.509 Certificate List Match'
     SYNTAX 1.3.6.1.1.15.6 )
```

3.7. algorithmIdentifierMatch

The algorithmIdentifierMatch mating rule compares a presented algorithm identifier with an attribute value of the supported algorithm as described in clause 11.3.7 of [X.509].

```
( 2.5.13.40 NAME 'algorithmIdentifier'
     DESC 'X.509 Algorithm Identifier Match'
     SYNTAX 1.3.6.1.1.15.7 )
```

4. Attribute Types

This section details a set of certificate and related attribute types for use in LDAP.

4.1. userCertificate

The userCertificate attribute holds the X.509 certificates issued to the user by one or more certificate authorities, as discussed in clause 11.2.1 of [X.509].

```
( 2.5.4.36 NAME 'userCertificate'
     DESC 'X.509 user certificate'
     EQUALITY certificateExactMatch
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.8 )
```

As required by this attribute type's syntax, values of this attribute are requested and transferred using the attribute description "userCertificate;binary".

4.2. cACertificate

The cACertificate attribute holds the X.509 certificates issued to the certificate authority (CA), as discussed in clause 11.2.2 of [X.509].

```
( 2.5.4.37 NAME 'cACertificate'
     DESC 'X.509 CA certificate'
     EQUALITY certificateExactMatch
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.8 )
```

As required by this attribute type's syntax, values of this attribute are requested and transferred using the attribute description "cACertificate;binary".

4.3. crossCertificatePair

The crossCertificatePair attribute holds an X.509 certificate pair, as discussed in clause 11.2.3 of [X.509].

```
( 2.5.4.40 NAME 'crossCertificatePair'
     DESC 'X.509 cross certificate pair'
     EQUALITY certificatePairExactMatch
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.10 )
```

As required by this attribute type's syntax, values of this attribute are requested and transferred using the attribute description "crossCertificatePair;binary".

4.4. certificateRevocationList

The certificateRevocationList attribute holds certificate lists, as discussed in 11.2.4 of [X.509].

```
( 2.5.4.39 NAME 'certificateRevocationList'
     DESC 'X.509 certificate revocation list'
     EQUALITY certificateListExactMatch
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.9 )
```

As required by this attribute type's syntax, values of this attribute are requested and transferred using the attribute description "certificateRevocationList;binary".

4.5. authorityRevocationList

The authorityRevocationList attribute holds certificate lists, as discussed in 11.2.5 of [X.509].

```
( 2.5.4.38 NAME 'authorityRevocationList'
     DESC 'X.509 authority revocation list'
     EQUALITY certificateListExactMatch
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.9 )
```

As required by this attribute type's syntax, values of this attribute are requested and transferred using the attribute description "authorityRevocationList;binary".

4.6. deltaRevocationList

The deltaRevocationList attribute holds certificate lists, as discussed in 11.2.6 of [X.509].

```
( 2.5.4.53 NAME 'deltaRevocationList'
     DESC 'X.509 delta revocation list'
     EQUALITY certificateListExactMatch
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.9 )
```

As required by this attribute type's syntax, values of this attribute MUST be requested and transferred using the attribute description "deltaRevocationList;binary".

4.7. supportedAlgorithms

The supportedAlgorithms attribute holds supported algorithms, as discussed in 11.2.7 of [X.509].

```
( 2.5.4.52 NAME 'supportedAlgorithms'
     DESC 'X.509 supported algorithms'
     EQUALITY algorithmIdentifierMatch
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.49 )
```

As required by this attribute type's syntax, values of this attribute MUST be requested and transferred using the attribute description "supportedAlgorithms;binary".

5. Object Classes

This section details a set of certificate-related object classes for use in LDAP.

5.1. pkiUser

This object class is used in augment entries for objects that may be subject to certificates, as defined in clause 11.1.1 of [X.509].

```
( 2.5.6.21 NAME 'pkiUser'
     DESC 'X.509 PKI User'
     SUP top AUXILIARY
     MAY userCertificate )
```

5.2. pkiCA

This object class is used to augment entries for objects that act as certificate authorities, as defined in clause 11.1.2 of [X.509]

```
( 2.5.6.22 NAME 'pkiCA'
     DESC 'X.509 PKI Certificate Authority'
     SUP top AUXILIARY
     MAY ( cACertificate $ certificateRevocationList $
          authorityRevocationList $ crossCertificatePair ) )
```

5.3. cRLDistributionPoint

This class is used to represent objects that act as CRL distribution points, as discussed in clause 11.1.3 of [X.509].

```
( 2.5.6.19 NAME 'cRLDistributionPoint'
     DESC 'X.509 CRL distribution point'
     SUP top STRUCTURAL
     MUST cn
     MAY ( certificateRevocationList $
          authorityRevocationList $ deltaRevocationList ) )
```

5.4. deltaCRL

The deltaCRL object class is used to augment entries to hold delta revocation lists, as discussed in clause 11.1.4 of [X.509].

```
( 2.5.6.23 NAME 'deltaCRL'
     DESC 'X.509 delta CRL'
     SUP top AUXILIARY
     MAY deltaRevocationList )
```

5.5. strongAuthenticationUser

This object class is used to augment entries for objects participating in certificate-based authentication, as defined in clause 6.15 of [X.521]. This object class is deprecated in favor of pkiUser.

```
( 2.5.6.15 NAME 'strongAuthenticationUser'
     DESC 'X.521 strong authentication user'
     SUP top AUXILIARY
     MUST userCertificate )
```

5.6. userSecurityInformation

This object class is used to augment entries with needed additional associated security information, as defined in clause 6.16 of [X.521].

```
( 2.5.6.18 NAME 'userSecurityInformation'
     DESC 'X.521 user security information'
     SUP top AUXILIARY
     MAY ( supportedAlgorithms ) )
```

5.7. certificationAuthority

This object class is used to augment entries for objects that act as certificate authorities, as defined in clause 6.17 of [X.521]. This object class is deprecated in favor of pkiCA.

```
( 2.5.6.16 NAME 'certificationAuthority'
     DESC 'X.509 certificate authority'
     SUP top AUXILIARY
     MUST ( authorityRevocationList $
          certificateRevocationList $ cACertificate )
     MAY crossCertificatePair )
```

5.8. certificationAuthority-V2

This object class is used to augment entries for objects that act as certificate authorities, as defined in clause 6.18 of [X.521]. This object class is deprecated in favor of pkiCA.

```
( 2.5.6.16.2 NAME 'certificationAuthority-V2'
     DESC 'X.509 certificate authority, version 2'
     SUP certificationAuthority AUXILIARY
     MAY deltaRevocationList )
```

6. Security Considerations

General certificate considerations [RFC3280] apply to LDAP-aware certificate applications. General LDAP security considerations [RFC4510] apply as well.

While elements of certificate information are commonly signed, these signatures only protect the integrity of the signed information. In the absence of data integrity protections in LDAP (or lower layer, e.g., IPsec), a server is not assured that client certificate request (or other request) was unaltered in transit. Likewise, a client cannot be assured that the results of the query were unaltered in

transit. Hence, it is generally recommended that implementations make use of authentication and data integrity services in LDAP [RFC4513][RFC4511].

7. IANA Considerations

7.1. Object Identifier Registration

The IANA has registered an LDAP Object Identifier [RFC4520] for use in this technical specification.

```
Subject: Request for LDAP OID Registration
Person & email address to contact for further information:
    Kurt Zeilenga <kurt@OpenLDAP.org>
Specification: RFC 4523
Author/Change Controller: IESG
Comments:
    Identifies the LDAP X.509 Certificate schema elements
     introduced in this document.
```

7.2. Descriptor Registration

The IANA has updated the LDAP Descriptor registry [RFC44520] as indicated below.

```
Subject: Request for LDAP Descriptor Registration
Descriptor (short name): see table
Object Identifier: see table
Person & email address to contact for further information:
    Kurt Zeilenga <kurt@OpenLDAP.org>
Usage: see table
Specification: RFC 4523
Author/Change Controller: IESG

algorithmIdentifierMatch     M 2.5.13.40
authorityRevocationList      A 2.5.4.38 *
cACertificate                A 2.5.4.37 *
cRLDistributionPoint         O 2.5.6.19 *
certificateExactMatch        M 2.5.13.34
certificateListExactMatch    M 2.5.13.38
certificateListMatch         M 2.5.13.39
certificateMatch             M 2.5.13.35
certificatePairExactMatch    M 2.5.13.36
certificatePairMatch         M 2.5.13.37
certificateRevocationList    A 2.5.4.39 *
certificationAuthority       O 2.5.6.16 *
certificationAuthority-V2    O 2.5.6.16.2 *
crossCertificatePair         A 2.5.4.40 *
```

deltaCRL	O	2.5.6.23 *
deltaRevocationList	A	2.5.4.53 *
pkiCA	O	2.5.6.22 *
pkiUser	O	2.5.6.21 *
strongAuthenticationUser	O	2.5.6.15 *
supportedAlgorithms	A	2.5.4.52 *
userCertificate	A	2.5.4.36 *
userSecurityInformation	O	2.5.6.18 *

* Updates previous registration

8. Acknowledgements

This document is based on X.509, a product of the ITU-T. A number of LDAP schema definitions were based on those found in RFCs 2252 and 2256, both products of the IETF ASID WG. The ABNF productions in Appendix A were provided by Steven Legg. Additional material was borrowed from prior works by David Chadwick and Steven Legg to refine the LDAP X.509 schema.

9. References

9.1. Normative References

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

[RFC3641] Legg, S., "Generic String Encoding Rules (GSER) for ASN.1 Types", RFC 3641, October 2003.

[RFC4510] Zeilenga, K., Ed., "Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map", RFC 4510, June 2006.

[RFC4512] Zeilenga, K., "Lightweight Directory Access Protocol (LDAP): Directory Information Models", RFC 4512, June 2006.

[RFC4522] Legg, S., "Lightweight Directory Access Protocol (LDAP): The Binary Encoding Option", RFC 4522, June 2006.

[X.509] International Telecommunication Union - Telecommunication Standardization Sector, "The Directory: Authentication Framework", X.509(2000).

[X.521] International Telecommunication Union - Telecommunication Standardization Sector, "The Directory: Selected Object Classes", X.521(2000).

[X.690] International Telecommunication Union - Telecommunication Standardization Sector, "Specification of ASN.1 encoding rules: Basic Encoding Rules (BER), Canonical Encoding Rules (CER), and Distinguished Encoding Rules (DER)", X.690(2002) (also ISO/IEC 8825-1:2002).

9.2. Informative References

[RFC1777] Yeong, W., Howes, T., and S. Kille, "Lightweight Directory Access Protocol", RFC 1777, March 1995.

[RFC2156] Kille, S., "MIXER (Mime Internet X.400 Enhanced Relay): Mapping between X.400 and RFC 822/MIME", RFC 2156, January 1998.

[RFC3280] Housley, R., Polk, W., Ford, W., and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002.

[RFC3494] Zeilenga, K., "Lightweight Directory Access Protocol version 2 (LDAPv2) to Historic Status", RFC 3494, March 2003.

[RFC3642] Legg, S., "Common Elements of Generic String Encoding Rules (GSER) Encodings", RFC 3642, October 2003.

[RFC4234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.

[RFC4511] Sermersheim, J., Ed., "Lightweight Directory Access Protocol (LDAP): The Protocol", RFC 4511, June 2006.

[RFC4513] Harrison, R. Ed., "Lightweight Directory Access Protocol (LDAP): Authentication Methods and Security Mechanisms", RFC 4513, June 2006.

[RFC4520] Zeilenga, K., "Internet Assigned Numbers Authority (IANA) Considerations for the Lightweight Directory Access Protocol (LDAP)", BCP 64, RFC 4520, June 2006.

Appendix A.

This appendix is informative.

This appendix provides ABNF [RFC4234] grammars for GSER-based [RFC3641] LDAP-specific encodings specified in this document. These grammars where produced using, and relying on, Common Elements for GSER Encodings [RFC3642].

A.1. CertificateExactAssertion

```
CertificateExactAssertion = "{" 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
```

A.2. CertificateAssertion

```
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 "}"

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 = id-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.65.79.49.64.65.6E.74.69.66.69.65.72
     ; 'subjectKeyIdentifier'
id-authorityKeyIdentifier =
     %x61.75.74.68.6F.72.69.74.79.4B.65.79.49.64.65.6E.74.69.66.69.65.72
     ; 'authorityKeyIdentifier'
id-certificateValid = %x63.65.72.74.69.66.69.63.61.74.65.56.61.6C.69.64
     ; 'certificateValid'
id-privateKeyValid = %x70.72.69.76.61.74.65.4B.65.79.56.61.6C.69.64
     ; 'privateKeyValid'
id-subjectPublicKeyAlgID  =
     %x73.75.62.6A.65.63.74.50.75.62.6C.69.63.4B.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.65
     ; '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.74.73
     ; '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
     ;; <Value> as defined in Section 3 of [RFC3641]

id-type-id = %x74.79.70.65.2D.69.64 ; 'type-id'
id-value = %x76.61.6C.75.65 ; 'value'

ORAddress = dquote *SafeIA5Character dquote
SafeIA5Character = %x01-21 / %x23-7F / ; ASCII minus dquote
     dquote dquote ; escaped double quote
dquote = %x22 ; '"' (double quote)

;; Note: The <ORAddress> rule encodes the x400Address component
;; of a GeneralName as a character string between double quotes.
;; The character string is first derived according to Section 4.1
;; of [RFC2156], and then any embedded double quotes are escaped
;; by being repeated. This resulting string is output between
;; double quotes.

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.72.69.74.79.43.65.72.74.49.73.73.75.65.72
     ; 'authorityCertIssuer'

id-authorityCertSerialNumber = %x61.75.74.68.6F.72.69.74.79.43
     %x65.72.74.53.65.72.69.61.6C.4E.75.6D.62.65.72
     ; '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 "}"

;; Note: The <key-usage-bit-list> rule encodes the one bits in
;; a KeyUsage value as a comma separated list of identifiers.

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
     %x75.72.65 ; '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.74.69.66.69.65.72
     ; '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.69.74.74.65.64.53.75.62.74.72.65.65.73
     ; 'permittedSubtrees'
id-excludedSubtrees =
     %x65.78.63.6C.75.64.65.64.53.75.62.74.72.65.65.73
     ; '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
```

A.3. CertificatePairExactAssertion

```
  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
       %x43.41.41.73.73.65.72.74.69.6F.6E ; 'issuedToThisCAAssertion'
  id-issuedByThisCAAssertion = %x69.73.73.75.65.64.42.79.54.68.69.73
       %x43.41.41.73.73.65.72.74.69.6F.6E ; 'issuedByThisCAAssertion'
```

A.4. CertificatePairAssertion

```
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
```

A.5. CertificateListExactAssertion

```
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.50.6F.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'
```

A.6. CertificateListAssertion

```
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.43.68.61.6E.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.70.65.72.61.74.69.6F.6E
        ; 'cessationOfOperation'
   id-certificateHold = %x63.65.72.74.69.66.69.63.61.74.65.48.6F.6C.64
        ; 'certificateHold'
   id-privilegeWithdrawn =
        %x70.72.69.76.69.6C.65.67.65.57.69.74.68.64.72.61.77.6E
        ; 'privilegeWithdrawn'
```

```
   id-aACompromise = %x61.41.43.6F.6D.70.72.6F.6D.69.73.65
        ; 'aACompromise'
```

A.7. AlgorithmIdentifier

```
   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'
```

Author's Address

Kurt D. Zeilenga
OpenLDAP Foundation

EMail: Kurt@OpenLDAP.org

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).