Abstract

This document describes an LDAP schema which can be used to implement a certificate store for X.509 certificates. Specifically, a structural object class for a X.509 certificate is defined. Key fields of a certificate are stored in LDAP attributes so that applications can easily retrieve the certificates needed by using basic LDAP search filters. Multiple certificates for a single entity can be stored and retrieved.

Conventions used in this document
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 [RFC2119].

The following syntax specifications use the augmented Backus-Naur Form (ABNF) as described in [RFC2234].

Schema definitions are provided using LDAPv3 description formats [RFC2252]. Definitions provided here are formatted (line wrapped) for readability.

Table of Contents

1. Introduction .......................... 5
2. Comparison with Values Return Filter Control ........... 5
3. Comparison with component matching approach ............ 6
4. The x509certificate object class and its attribute types 6
4.1 Attributes for mandatory fields of an X.509 certificate 7
4.1.1 X.509 version ......................... 7
4.1.2 Serial number ........................ 7
4.1.3 Signature algorithm ................... 7
4.1.4 Issuer ................................. 8
4.1.5 Validity ............................... 8
4.1.6 Subject ............................... 8
4.1.7 Subject public key info algorithm .......... 9
4.2 Attributes for selected extensions .................. 9
4.2.1 Authority key identifier extension .......... 10
4.2.1.1 Authority key identifier ............ 10
4.2.1.2 Authority cert issuer ............... 10
4.2.1.3 Authority cert serial number ......... 11
4.2.2 Subject key identifier extension .......... 11
4.2.3 Key usage extension .................... 11
4.2.4 Policy information identifier extension .. 12
4.2.5 Subject alternative name extension .......... 12
4.2.5.1 Subject RFC822 name ................. 12
4.2.5.2 Subject DNS name ..................... 13
4.2.5.3 Subject directory name ............... 13
4.2.5.4 Subject Uniform Resource Identifier ... 13
4.2.5.5 Subject IP address ................... 14
4.2.5.6 Subject registered ID ............... 14
4.2.6 Issuer alternative name extension .......... 14
4.2.6.1 Issuer RFC 822 name ................. 14
4.2.6.2 Issuer DNS name ..................... 15
4.2.6.3 Issuer directory name ............... 15
4.2.6.4 Issuer Uniform Resource Identifier ... 15
4.2.6.5 Issuer IP address ................... 16
4.2.6.6 Issuer registered ID ............... 16
4.2.7 Extended key usage extension ............. 16
1. Introduction

A key component in the widespread adoption of a PKI infrastructure is the general availability of public key and their certificates. Today, certificates are often published in an X.500 compliant directory service. These directories are accessed by applications using the LDAP v3 [RFC3377] protocol. An LDAPv3 schema for PKI repository objects is specified in [pkix-ldap-schema], where a set of object classes, attribute types, syntaxes, and extended matching rules are defined. For storing certificates, the "userCertificate" and "cACertificate" attribute types are used. All certificates of an entity are stored as values in these multi-valued attributes. This solution has a serious drawback. In LDAP, the smallest granularity of data access is the attribute. The directory server will therefore always return the full list of certificates of an entry to clients dealing with certificates. If the number of certificates for an entity is large this will result in considerable overhead and burden to the client.

This document proposes to solve this problem by the use of the structural object classes x509userCertificate and x509caCertificate for storing certificates. Each certificate will be stored in a separate entry in the directory. Fields of certificates which are needed to identify a certificate and those which are often used in searching for an appropriate certificate, are extracted from the certificate and stored as attributes of the entry. Applications can thus search for specific certificates with simple LDAP filters. This approach could be named a metadata approach, since data (attributes) about data (certificate) are stored. The use of simple attributes also makes a large scale widely distributed certificate repository service possible by using an indexing service based on The Common Indexing Protocol (CIP) [RFC2651].

This document is one of a set following this approach comprising:

i) the LDAP schema for X.509 public key certificates (this document)
ii) the LDAP schema for X.509 attribute certificates [ldap-ac-schema]
iii) the LDAP schema for X.509 CRLs [ldap-crl-schema]

Two alternative approaches are discussed in the next two sections.

2. Comparison with Values Return Filter Control

In [matchedval] a control has been defined that allows for only a subset of values of a specified attribute to be returned from a matching entry, by defining a filter for the returned values. In this section, this approach is compared with the one proposed in this document.

The major benefit of the Values Return Filter Control is that it does not require any changes to the DIT. While it is a simple matter to modify the DIT in such a way that all certificate information is removed from the entries and placed in the container directly beneath the entries according to the definitions of this specification, it is less simple to simultaneously modify all of the applications that...
depend on certificates being stored in the entry. Thus, it may be desirable to duplicate the certificate information, by having it appear in the entry, as well as in the container beneath the entry for a short period of time, in order to allow for migration of the applications to the new LDAP schema. As in any situation in which information is duplicated, great care must be taken in order to ensure the integrity of the information.

There are several advantages in using the x509certificate object class. No special matching rules are needed to retrieve a specific certificate. Any field in the certificate can be used in the search filter. Even information that doesn’t appear in the certificate can be used in a search filter. It is easier to remove certificates from the DIT, since the entire certificate BER/DER encoding does not have to be supplied in the modify operation. Searches that don’t need extensible matching rules and Values Return Filter Control will perform faster.

Another advantage of the solution proposed here is that it will not be necessary to modify existing server implementations to support this schema. The extended matching rules proposed in [pkix-ldap-schema] would require substantial changes in the servers’ indexing mechanisms. In contrast, servers implementing the x509certificate schema can easily leverage their indexing support for standard LDAPv3 syntaxes.

A CIP based indexing system for a wide scale distributed certificate repository will only be possible by using the solution proposed here.

3. Comparision with component matching approach

[componentmatch] defines a new mechanism for matching in complex syntaxes, by defining generic matching rules that can match any user selected component parts in an attribute value of any arbitrarily complex attribute syntax. We believe that this might be the proper way to solve search problems in the longer term, but that it will take a long time until such ASN.1 based mechanisms will be implemented in LDAP servers and clients. Even if this has happened the mechanism proposed here, will still be usefull in the frame of CIP. A simple and easy to implement mechanism is needed today and this is what this memo wants to provide.

4. The x509certificate object class and its attribute types

The description of all attributes with relevance to fields of an X.509 certificate include a respective reference to [X.509-2000] and to [RFC3280].
4.1 Attributes for mandatory fields of an X.509 certificate

4.1.1 X.509 version

X.509 Version of the encoded certificate (See X.509(2000) 7, RFC3280 4.1.2.1.) or of the CRL.

( 1.3.6.1.4.1.10126.1.5.3.1
  NAME ‘x509version’
  DESC ‘X.509 Version of the certificate, or of the CRL’
  EQUALITY integerMatch
  ORDERING integerOrderingMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
  SINGLE-VALUE )

Values of this attribute may either be 0, 1, or 2 corresponding to X.509 v1, v2 or v3.

4.1.2 Serial number

The serial number is an integer assigned by the CA to each certificate. It is unique for each certificate issued by a given CA (i.e., the issuer name and serial number uniquely identify a certificate). See X.509(2000) 7, RFC3280 4.1.2.2

( 1.3.6.1.4.1.10126.1.5.3.2
  NAME ‘x509serialNumber’
  DESC ‘Unique integer for each certificate issued by a particular CA’
  EQUALITY integerMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 )

4.1.3 Signature algorithm

OID identifying the algorithm used by the CA in signing the certificate (See X.509(2000) 7, RFC3280 4.1.2.3) or the CRL.

( 1.3.6.1.4.1.10126.1.5.3.3
  NAME ‘x509signatureAlgorithm’
  DESC ‘OID of the algorithm used by the CA in signing the CRL or the certificate’
  EQUALITY objectIdentifierMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.38
  SINGLE-VALUE )
4.1.4 Issuer

String representation of the issuer’s distinguished name.

See X.509(2000) 7, RFC3280 4.1.2.4

(1.3.6.1.4.1.10126.1.5.3.4
NAME 'x509issuer'
DESC 'Distinguished name of the entity who has signed and issued the certificate or CRL'
EQUALITY distinguishedNameMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
SINGLE-VALUE )

Values of this attribute type must be encoded according to the syntax given in [RFC2252].

4.1.5 Validity

The "validity" attribute in an X.509 certificate (see X.509(2000) 7, RFC3280 4.1.2.5) consists of an ASN.1 sequence of two timestamps which define the start and end of the certificate’s validity period. The actual values are a string representation of dates in UTC.

The times are represented in string form as defined in [RFC2252].

(1.3.6.1.4.1.10126.1.5.3.5
NAME 'x509validityNotBefore'
DESC 'Date on which the certificate validity period begins'
EQUALITY generalizedTimeMatch
ORDERING generalizedTimeOrderingMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
SINGLE-VALUE )

(1.3.6.1.4.1.10126.1.5.3.6
NAME 'x509validityNotAfter'
DESC 'Date on which the certificate validity period ends, X.509(2000) 7, RFC3280 4.1.2.5'
EQUALITY generalizedTimeMatch
ORDERING generalizedTimeOrderingMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
SINGLE-VALUE )

Note that the field in the certificate may be in UTC or GeneralizedTime format. If in UTC format, the creator of this attribute MUST convert the UTC time into GeneralizedTime format when creating the attribute value.

4.1.6 Subject

String representation of the subject’s distinguished name.
( 1.3.6.1.4.1.10126.1.5.3.7
NAME 'x509subject'
DESC 'Distinguished name of the entity associated with this
public-key, X.509(2000) 7, RFC3280 4.1.2.6'
EQUALITY distinguishedNameMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
SINGLE-VALUE )

Values of this attribute type must be encoded according to the syntax
given in [RFC2253].

4.1.7 Subject public key info algorithm

OID of the algorithm of which the certified public key is an instance of.

( 1.3.6.1.4.1.10126.1.5.3.8
NAME 'x509subjectPublicKeyInfoAlgorithm'
DESC 'OID of the algorithm which this public key is an
instance of, X.509(2000) 7, RFC3280 4.1.2.7'
EQUALITY objectIdentifierMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.38
SINGLE-VALUE )

4.2 Attributes for selected extensions

As this specification intends to only facilitate applications in finding certificates, only those extensions have to be defined that might be searched for. Thus extensions described in [RFC3280] like the following are not dealt with here:

- private key usage period extension
- policy mappings extension
- subject directory attributes extension
- pathLenConstraint of basic constraints extension
- name constraints extensions
- policy constraints extensions
- inhibit any policy extension
- freshest CRL extension
4.2.1 Authority key identifier extension

This attribute identifies the public key to be used to verify the signature on this certificate or CRL. The key may be identified by an explicit key identifier in the keyIdentifier component, by identification of a certificate for the key (giving certificate issuer in the authorityCertIssuer component and certificate serial number in the authorityCertSerialNumber component), or by both explicit key identifier and identification of a certificate for the key.

4.2.1.1 Authority key identifier

( 1.3.6.1.4.1.10126.1.5.3.11
  NAME 'x509authorityKeyIdentifier'
  DESC 'keyIdentifier field of the authorityKeyIdentifier extension, X.509(2000) 8.2.2.1, RFC3280 4.2.1.1'
  EQUALITY octetStringMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.40 SINGLE-VALUE )

4.2.1.2 Authority cert issuer

( 1.3.6.1.4.1.10126.1.5.3.12
  NAME 'x509authorityCertIssuer'
  DESC 'authorityCertIssuer field of the authorityKeyIdentifier extension, X.509(2000) 8.2.2.1, RFC3280 4.2.1.1'
  EQUALITY distinguishedNameMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE )

In this specification, only the "Name" choice, encoded according to [RFC2253], of the "GeneralName" type may be used.
4.2.1.3 Authority cert serial number

( 1.3.6.1.4.1.10126.1.5.3.13
  NAME 'x509authorityCertSerialNumber'
  DESC 'authorityCertSerialNumber field of the
         authorityKeyIdentifier extension, X.509(2000) 8.2.2.1,
         RFC3280 4.2.1.1'
  EQUALITY integerMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
  SINGLE-VALUE )

4.2.2 Subject key identifier extension

This attribute identifies the public key being certified. It enables
distinct keys used by the same subject to be differentiated.

( 1.3.6.1.4.1.10126.1.5.3.14
  NAME 'x509subjectKeyIdentifier'
  DESC 'Key identifier which must be unique with respect to all
         key identifiers for the subject, X.509(2000) 8.2.2.2,
         RFC3280 4.2.1.2'
  EQUALITY octetStringMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.40
  SINGLE-VALUE )

4.2.3 Key usage extension

This attribute defines the purpose (e.g., encipherment, signature,
certificate signing) of the key contained in the certificate.

( 1.3.6.1.4.1.10126.1.5.3.15
  NAME 'x509keyUsage'
  DESC 'Purpose for which the certified public key is used,
         X.509(2000) 8.2.2.3, RFC3280 4.2.1.3'
  EQUALITY caseIgnoreMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

Values of this type are encoded according to the following BNF, so
that each value corresponds to the respective bit in the ASN.1
"KeyUsage" bitstring:

x509keyUsage-value = "digitalSignature" / "nonRepudiation" /
  "keyEncipherment" / "dataEncipherment" / "keyAgreement" /
  "keyCertSign" / "cRLSign" / "encipherOnly" / "decipherOnly"
4.2.4 Policy information identifier extension

This attribute contains OIDs which indicate the policy under which the certificate has been issued and the purposes for which the certificate may be used.

( 1.3.6.1.4.1.10126.1.5.3.16
  NAME 'x509policyInformationIdentifier'
  DESC 'OID which indicates the policy under which the certificate has been issued and the purposes for which the certificate may be used, X.509(2000) 8.2.2.6, RFC3280 4.2.1.5'
  EQUALITY objectIdentifierMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.38
  SINGLE-VALUE )

4.2.5 Subject alternative name extension

The subject alternative name extension allows additional identities to be bound to the subject of the certificate. Separate attribute types are defined for all choices of the ASN.1 type "GeneralName" except for "otherName", "x400Address" and "ediPartyName".

4.2.5.1 Subject RFC822 name

( 1.3.6.1.4.1.10126.1.5.3.17
  NAME 'x509subjectAltNameRfc822Name'
  DESC 'Internet electronic mail address, X.509(2000) 8.3.2.1, RFC3280 4.2.1.7'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

Values of this attribute must be encoded according to the syntax given in [RFC0822].
4.2.5.2 Subject DNS name

( 1.3.6.1.4.1.10126.1.5.3.18
  NAME ‘x509subjectDnsName’
  DESC ‘Internet domain name, X.509(2000) 8.3.2.1, RFC3280 4.2.1.7’
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

Values of this attribute must be encoded as Internet domain names in accordance with [RFC1035].

4.2.5.3 Subject directory name

( 1.3.6.1.4.1.10126.1.5.3.19
  NAME ‘x509subjectDirectoryName’
  DESC ‘Distinguished name, X.509(2000) 8.3.2.1, RFC3280 4.2.1.7’
  EQUALITY distinguishedNameMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 )

Values of this attribute type must be encoded according to the syntax given in [RFC2253].

4.2.5.4 Subject Uniform Resource Identifier

( 1.3.6.1.4.1.10126.1.5.3.20
  NAME ‘x509subjectUniformResourceIdentifier’
  DESC ‘Uniform Resource Identifier for the World-Wide Web, X.509(2000) 8.3.2.1, RFC3280 4.2.1.7’
  EQUALITY caseExactMatch
  SUBSTR caseExactSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

Values of this attribute must be encoded according to the syntax given in [RFC2396].
4.2.5.5 Subject IP address

( 1.3.6.1.4.1.10126.1.5.3.21
 NAME 'x509subjectIpAddress'
 DESC 'Internet Protocol address, X.509(2000) 8.3.2.1, RFC3280 4.2.1.7'
 EQUALITY caseIgnoreMatch
 SUBSTR caseIgnoreSubstringsMatch
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

Values of this attribute type must be stored in the syntax given in Appendix B of [RFC2373].

4.2.5.6 Subject registered ID

( 1.3.6.1.4.1.10126.1.5.3.22
 NAME 'x509subjectRegisteredID'
 DESC 'OID of any registered object, X.509(2000) 8.3.2.1, RFC3280 4.2.1.7'
 EQUALITY objectIdentifierMatch
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.38 )

registeredID is an identifier of any registered object assigned in accordance with ITU-T Rec. X.660.

4.2.6 Issuer alternative name extension

The issuer alternative names extension allows additional identities to be bound to the issuer of the certificate or CRL. Separate attribute types are defined for all choices of the ASN.1 type "GeneralName" except for "otherName", "x400Address" and "ediPartyName".

4.2.6.1 Issuer RFC 822 name

( 1.3.6.1.4.1.10126.1.5.3.23
 NAME 'x509issuerRfc822Name'
 DESC 'Internet electronic mail address, X.509(2000) 8.3.2.2, RFC3280 4.2.1.8'
 EQUALITY caseIgnoreMatch
 SUBSTR caseIgnoreSubstringsMatch
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

Values of this attribute must be encoded according to the syntax given in [RFC0822].
4.2.6.2 Issuer DNS name

( 1.3.6.1.4.1.10126.1.5.3.24
 NAME 'x509issuerDnsName'
 DESC 'Internet domain name, X.509(2000) 8.3.2.2, \[RFC3280\]
 4.2.1.8'
 EQUALITY caseIgnoreMatch
 SUBSTR caseIgnoreSubstringsMatch
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

Values of this attribute must be encoded as Internet domain names in accordance with \[RFC1035\].

4.2.6.3 Issuer directory name

( 1.3.6.1.4.1.10126.1.5.3.25
 NAME 'x509issuerDirectoryName'
 DESC 'Distinguished name, X.509(2000) 8.3.2.2, \[RFC3280\]
 4.2.1.8'
 EQUALITY distinguishedNameMatch
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 )

Values of this attribute type must be encoded according to the syntax given in \[RFC2253\].

4.2.6.4 Issuer Uniform Resource Identifier

( 1.3.6.1.4.1.10126.1.5.3.26
 NAME ‘x509issuerUniformResourceIdentifier’
 DESC 'Uniform Resource Identifier for the World-Wide Web, X.509(2000) 8.3.2.2, \[RFC3280\] 4.2.1.8’
 EQUALITY caseExactMatch
 SUBSTR caseExactSubstringsMatch
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

Values of this attribute must be encoded according to the syntax given in \[RFC2396\].
4.2.6.5 Issuer IP address

( 1.3.6.1.4.1.10126.1.5.3.27
  NAME 'x509issuerIpAddress'
  DESC 'Internet Protocol address, X.509(2000) 8.3.2.2, RFC3280 4.2.1.8'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

Values of this attribute type must be stored in the syntax given in Appendix B of [RFC2373].

4.2.6.6 Issuer registered ID

( 1.3.6.1.4.1.10126.1.5.3.28
  NAME 'x509issuerRegisteredID'
  DESC 'OID of any registered object, X.509(2000) 8.3.2.2, RFC3280 4.2.1.8'
  EQUALITY objectIdentifierMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.38 )

registeredID is an identifier of any registered object assigned in accordance with ITU-T Rec. X.660.

4.2.7 Extended key usage extension

This attribute indicates one or more purposes for which the certified public key may be used, in addition to or in place of the basic purposes indicated in the "x509keyUsage" attribute. These purposes are identified by their OID.

( 1.3.6.1.4.1.10126.1.5.3.30
  NAME 'x509extKeyUsage'
  DESC 'Purposes for which the certified public key may be used (identified by OID), X.509(2000) 8.2.2.4, RFC3280 4.2.1.13'
  EQUALITY objectIdentifierMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.38 )

4.2.8 CRL distribution points extension

This attribute identifies how CRL information for this certificate can be obtained.
In this specification, only the "uniformResourceIdentifier" choice of "distributionPoint.fullName" field is supported. If this attribute exists in an entry, both the "reasons" and "cRLIssuer" fields MUST be absent from the certificate, i.e. the CRL distributed by the distribution point contains revocations for all revocation reasons and the CRL issuer is identical to the certificate issuer.

Values of this attribute must be encoded according to the URI syntax given in [RFC2396].

4.3 Additional attributes

4.3.1 Certificate location

This attribute contains a pointer to the directory entry of a certificate. Thus it is possible to point to the certificate from an, e.g., white pages entry.

( 1.3.6.1.4.1.10126.1.5.4.74
   NAME 'x509certLocation'
   DESC 'Pointer to a x509certificate Entry'
   EQUALITY distinguishedNameMatch
   SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 )

4.3.2 Certificate holder

This attribute contains a pointer to the directory entry of the end entity to which this certificate was issued. Thus it is possible to link a certificate entry in a certificate repository to, e.g., a white pages entry of the subject.

( 1.3.6.1.4.1.10126.1.5.4.75
   NAME 'x509certHolder'
   DESC 'Pointer to the directory entry of the end entity to which this certificate was issued'
   EQUALITY distinguishedNameMatch
   SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 )

4.3.3 Email addresses

The "mail" (0.9.2342.19200300.100.1.3) attribute from [RFC2798] is used to store the subject’s email address. This attribute MUST be populated with the values from a subject alternative name extension
of type rfc822Name if such an extension is present. Legacy applications conforming to [RFC2312] include an "EmailAddress" (1.2.840.113549.1.9.1) attribute in the subject’s distinguished name. If the subject alternative name extension is absent from the certificate, this value MUST be used to populate the "mail" attribute.

4.3.4. X.509 User Certificate
This attribute is used to store the complete certificate. Since it has to be single valued the multi valued attribute userCertificate [pkix-ldap-schema] cannot be used.

( 1.3.6.1.4.1.10126.1.5.4.76
  NAME ’x509userCert’
  DESC ’the complete x.509 user certificate’
  EQUALITY certificateExactMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.8
  SINGLE-VALUE )

4.3.4. X.509 CA Certificate
This attribute is used to store the complete ca certificate. Since it has to be single valued the multi valued attribute caCertificate [pkix-ldap-schema] cannot be used.

( 1.3.6.1.4.1.10126.1.5.4.77
  NAME ’x509caCert’
  DESC ’the complete x.509 CA certificate’
  EQUALITY certificateExactMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.8
  SINGLE-VALUE )
4.4 x509PKC object class

This abstract object class contains the fields of an X.509 user certificate or CA certificate that are used in searches as attributes. It is derived from the abstract objectclass x.509base as specified in [ldap-crl-schema] and is base for the two following object classes.

( 1.3.6.1.4.1.10126.1.5.4.2.3
NAME 'x509PKC'
ABSTRACT
SUP x509base
MUST ( x509serialNumber $ x509validityNotBefore $ x509validityNotAfter $ x509subjectPublicKeyInfoAlgorithm )
MAY ( mail $ x509authorityKeyIdIdentifier $ x509authorityCertIssuer $ x509authorityCertSerialNumber $ x509subjectKeyIdIdentifier $ x509keyUsage $ x509policyInformationIdentifier $ x509subjectRfc822Name $ x509subjectDnsName $ x509subjectDirectoryName $ x509subjectURI $ x509subjectIpAddress $ x509subjectRegisteredID $ x509issuerRfc822Name $ x509issuerDnsName $ x509issuerDirectoryName $ x509issuerURI $ x509issuerIpAddress $ x509issuerRegisteredID $ x509extKeyUsage $ x509cRLDistributionPoint $ x509certHolder ) )

4.4.1. X.509 user Certificate object class

This object class is for storing user certificates.

( 1.3.6.1.4.1.10126.1.5.4.2.4
NAME 'x509userCertificate'
SUP x509PKC
MUST x509userCert
MAY x509subject )

4.4.2. X.509 CA Certificate object class

This object class is for storing CA certificates.

( 1.3.6.1.4.1.10126.1.5.4.2.5
NAME 'x509caCertificate'
SUP x509PKC
MUST ( x509caCert $ x509subject ) )
4.5 x509CertificateHolder object class

This auxiliary object class has an attribute that contains a pointer on an entry with x509Certificate object class. Thus it is possible to link, e.g., an entry of a white pages directory to an entry in a certificate store.

( 1.3.6.1.4.1.10126.1.5.4.2.2
  NAME 'x509CertificateHolder'
  AUXILIARY
  MAY ( x509CertificateLocation ) )

5. DIT Structure and Naming

If the schema presented in this document is used to store certificate information in a directory that contains entries for organizations, persons, services, etc., each certificate belonging to an entity SHOULD be stored as a direct subordinate to the entity’s entry. In this case, these entries MUST be named by a multi-valued RDN formed by the certificate issuer and serial number, as this is the only way to enforce unique RDN under the siblings. This is expressed in the following name form:

| ( 1.3.6.1.4.1.10126.1.5.5.3 |
| NAME "x509PKCNameForm" |
| OC x509PKC |
| MUST ( x509serialNumber $ x509issuer ) ) |

Certificate name form

There are some LDAP implementations that don’t support multi-valued RDNs. These can use following alternative Name Form:

| ( 1.3.6.1.4.1.10126.1.5.5.4 |
| NAME "x509PKCASTNameForm" |
| OC x509PKC |
| MUST x509issuerSerial ) |

The attribute description of x509issuerSerial can be found in [ldap-ac-schema].
For public directories of CAs that, besides the data stored in the certificates, do not hold any additional data about end entities the following DIT structure might be preferable. Entries for certificates are stored directly below the issuing CA’s entry. In this case these entries SHOULD be named by the certificate serial number. This is expressed in the following name form:

```
( 1.3.6.1.4.1.10126.1.5.5.5
  NAME "x509PKCserialNumberNameForm"
  OC x509PKC
  MUST x509serialNumber )
```

certificate serial number name form

Care must be taken when encoding DNs that contain an x509issuer attribute. Such a value is a string representation according to [RFC2253]. These strings contain RFC2253 special characters and must therefore be escaped. For example, the issuer name in a certificate may be:

```
x509issuer: OU=VeriSign Trust Network,OU=(c) 1998 VeriSign\2c Inc. – For authorized use only,OU=Class 1 Public Primary Certification Authority - G2,O=VeriSign\2c Inc.,C=US
```

When used in a DN, this will be appear as:

```
dn: x509serialNumber=123456+x509issuer=OU\3dVeriSign Trust Network \2cOU\3d(c) 1998 VeriSign\5c\2c Inc. - For authorized use only\2cOU\3dClass 1 Public Primary Certification Authority - G2\2cO\3dVeriSign n\5c\2c Inc.\2cC\3dUS, cn=Joe Example,...
```

6. Security Considerations

Attributes of directory entries are used to provide descriptive information about the real-world objects they represent which can be people, organizations, or devices. Most countries have privacy laws regarding the publication of information about people.

Without additional mechanisms such as Operation Signatures [RFC2649] which allow a client to verify the origin and integrity of the data contained in the attributes defined in this document, a client MUST NOT treat this data as authentic. Clients MUST only use - after proper validation - the data which they obtained directly from the certificate. Directory administrators MAY deploy ACLs which limit access to the attributes defined in this document to search filters.

Transfer of cleartext passwords is strongly discouraged where the underlying transport service cannot guarantee confidentiality and may
result in disclosure of the password to unauthorized parties.
In order to protect the directory and its contents, strong authentication MUST have been used to identify the Client when an update operation is requested.

7. Acknowledgements

This document borrows from a number of IETF documents, including [certinfo-schema].

The authors wish to thank Michael Strasser and David Chadwick for their significant contributions to this document.

This work is part of the DFN Project "Ausbau und Weiterbetrieb eines Directory Kompetenzzentrums" funded by the German Ministry of Research (BMBF).

This document has been written in XML according to the DTD specified in RFC2629. xml2rfc has been used to generate an RFC2033 compliant plain text form. The XML source and a HTML version are available on request.

References


Appendix A. Sample directory entries

A sample x509certificate directory entry for an intermediate CA certificate in LDIF format:

dn: x509serialNumber=4903272,EMAILADDRESS=certify@pca.dfn.de,CN=DFN Toplevel Certification Authority,OU=DFN-PCA,OU=DFN-CERT GmbH,O=Deutsches Forschungsnetz,C=DE
objectclass: x509certificate
x509version: 2
x509serialNumber: 4903272
x509issuer: EMAILADDRESS=certify@pca.dfn.de,CN=DFN Toplevel Certification Authority,OU=DFN-PCA,OU=DFN-CERT GmbH,O=Deutsches Forschungsnetz,C=DE
x509validityNotBefore: 20020110170112Z
x509validityNotAfter: 20060110170112Z
x509subject: EMAILADDRESS=ca@daasi.de,OU=DAASI CA,O=DAASI International GmbH,C=DE
x509subjectPublicKeyInfoAlgorithm: 1.2.840.113549.1.1.1
x509basicConstraintsCa: TRUE
x509keyUsage: keyCertSign
x509keyUsage: cRLSign
A sample x509certificate directory entry for an end identity certificate in LDIF format:

dn: x509serialNumber=1581631808272310054353257112721713,EMAILADDRESS=certificate@trustcenter.de,OU=TC TrustCenter for Security in Data Networks GmbH,L=Hamburg,ST=Hamburg,C=DE
objectclass: x509certificate
x509version: 2
Appendix B. Sample searches

This section details how clients should access the certstore. The searches are presented in LDAP URL format.

Retrieve all certificates for an end entity from a certstore using the first DIT structure:

```
ldap:///CN=Norbert%20Klasen,O=DAASI%20International%20GmbH,C=de?
userCertificate?one?(objectClass=x509certificate)
```

Find a certificate in a trustcenter's certstore suitable for sending an encrypted S/MIME message to "norbert.klasen@daasi.de"

```
ldap:///O=TC%20TrustCenter%20for%20Security%20in%20Data%20Networks%20GmbH,L=Hamburg,ST=Hamburg,C=DE?
userCertificate?sub?(&(objectClass=x509certificate)(mail=norbert.klasen@daasi.de)(|(x509keyUsage=keyEncipherment)(x509keyUsage=keyAgreement)(x509extendedKeyUsage=1.3.6.1.5.5.7.3.4)))
```

Find a CA certificate by its "subjectKeyIdentifier" obtained from the "keyIdentifier" field of the "authorityKeyIdentifier" extension in an end entity certificate:

```
ldap://O=TC%20TrustCenter%20for%20Security%20in%20Data%20Networks%20GmbH,L=Hamburg,ST=Hamburg,C=DE?
userCertificate?sub?
```

Find a CA certificate using its "subjectKeyIdentifier" obtained by the "subjectKeyIdentifier" field of the "authorityKeyIdentifier" extension in an end entity certificate:

```
ldap://O=TC%20TrustCenter%20for%20Security%20in%20Data%20Networks%20GmbH,L=Hamburg,ST=Hamburg,C=DE?
userCertificate?sub?
```
ldap://?caCertificate?sub?
(& (objectClass=x509certificate)
(x509subjectKeyIdentifier=\CE6\7A\D9\16\95\4A\E1\12\9F\22\09\6A\43\83\78\25\70\52\E0\19))

Appendix C. Changes from previous Drafts

C.1 Changes in Draft 01

- Included new Attributes x509authorityKeyIdentifier, x509authorityCertissuer, x509authorityCertSerialNumber, x509certificateLocation, x509certificateHolder, and new objectclass x509certificateHolder
- Fixed bug in definition of objectclass x509certificate
- Changed references from RFC 2459 to RFC 3280 and included some respective language in 3.2.
- Changed references from RFC 2251 to RFC 3377 and deleted all references to LDAPv2.
- Deleted ";binary" in examples
- Included new section: Comparison with component matching approach
- Some changes in wording and section titles, and elimination of typos
- Changed order of authors, and one author’s address

C.1 Changes in Draft 01

- abstract object class x509PKC
- aligned to [ldap-ac-schema] and [ldap-crl-schema]