Internet X.509 Public Key Infrastructure Lightweight Directory Access Protocol Schema for X.509 Certificates
draft-ietf-pkix-ldap-pkc-schema-01

Status of this Memo

By submitting this Internet-Draft, I certify that any applicable patent or other IPR claims of which I am aware have been disclosed, and any of which I become aware will be disclosed, in accordance with RFC 3668.

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.

This Internet-Draft will expire on April 25, 2005.

Copyright Notice

Copyright (C) The Internet Society (2004). All Rights Reserved.

Abstract

This document describes a Lightweight Directory Access Protocol schema which can be used to implement a certificate store for X.509 certificates. Specifically, two structural object classes for X.509 user and CA certificates are 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
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 ........................................ 4
2. Comparison with Values Return Filter Control .......... 5
3. Comparison with Component Matching approach .......... 6
4. X.509 certificate object classes ..................... 7
   4.1 X.509 base object class .......................... 7
   4.2 X.509 PKC object class ............................ 7
   4.3 X.509 user certificate object class ............... 8
   4.4 X.509 CA certificate object class ................. 8
   4.5 X.509 PKC extensions auxiliary object class ....... 9
   4.6 X.509 certificate holder object class ............. 9
5. The attribute types of the X.509 certificate object classes . . 9
   5.1 Attributes for mandatory fields of an X.509 certificate . 10
      5.1.1 X.509 version .................................. 10
      5.1.2 Serial number ................................... 10
      5.1.3 Signature algorithm ............................. 10
      5.1.4 Issuer .......................................... 11
      5.1.5 Validity ........................................ 11
      5.1.6 Subject ......................................... 12
      5.1.7 Subject public key info algorithm ............... 12
   5.2 Attributes for selected extensions .................. 12
      5.2.1 Authority key identifier extension ............... 13
      5.2.2 Subject key identifier extension ................ 14
      5.2.3 Key usage extension ............................. 14
      5.2.4 Policy information identifier extension ......... 14
      5.2.5 Subject alternative name extension .............. 15
      5.2.6 Issuer alternative name extension ............... 16
      5.2.7 Basic constraints extension ..................... 18
      5.2.8 Extended key usage extension .................... 19
      5.2.9 CRL distribution points extension ............... 19
   5.3 Additional attributes ............................. 20
      5.3.1 Certificate location ............................ 20
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3.2</td>
<td>Certificate holder</td>
<td>20</td>
</tr>
<tr>
<td>6</td>
<td>DIT structure and naming</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>Security Considerations</td>
<td>22</td>
</tr>
<tr>
<td>8</td>
<td>IANA Considerations</td>
<td>22</td>
</tr>
<tr>
<td>9</td>
<td>Acknowledgments</td>
<td>22</td>
</tr>
<tr>
<td>10</td>
<td>References</td>
<td>23</td>
</tr>
<tr>
<td>10.1</td>
<td>Normative references</td>
<td>23</td>
</tr>
<tr>
<td>10.2</td>
<td>Non-normative references</td>
<td>24</td>
</tr>
<tr>
<td>A</td>
<td>Authors' Addresses</td>
<td>25</td>
</tr>
<tr>
<td>B</td>
<td>Sample directory entries</td>
<td>25</td>
</tr>
<tr>
<td>C</td>
<td>Sample searches</td>
<td>28</td>
</tr>
<tr>
<td>C.1</td>
<td>Changes from previous Drafts</td>
<td>29</td>
</tr>
<tr>
<td>C.2</td>
<td>Changes in draft-klasen-ldap-x509certificate-schema-01</td>
<td>29</td>
</tr>
<tr>
<td>C.3</td>
<td>Changes in draft-klasen-ldap-x509certificate-schema-02</td>
<td>29</td>
</tr>
<tr>
<td>C.4</td>
<td>Changes in draft-klasen-ldap-x509certificate-schema-03</td>
<td>29</td>
</tr>
<tr>
<td>C.5</td>
<td>Changes in draft-ietf-pkix-ldap-pkc-schema-01</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>Changes in draft-ietf-pkix-ldap-pkc-schema-00</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Changes in draft-ietf-pkix-ldap-pkc-schema-01</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>Intellectual Property and Copyright Statements</td>
<td>32</td>
</tr>
</tbody>
</table>
1. Introduction

A key component in the wide-spread adoption of a Public Key Infrastructure is the general availability of public keys 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. Having each certificate stored in a separate entry provides flexibility in structuring the Directory Information Tree. The certificate entries can be stored either below a person entry or below a CA entry as a certificate only repository, as shown in figure 1.

1.) below Person entry:

```
person
/ | \ 
/ | 
| cert1 cert2 cert3
```

2.) below CA cert repository:

```
CA
/ | \ 
/ | 
| certificate repository 
| cert1 cert2 ... cert1008
```

Figure 1: examples of possible DIT-structures
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], which defines a protocol between index servers for exchanging index objects in order to facilitate query routing. The Tagged Index Object format as specified in [RFC2654] was specified to carry directory server information, by collecting the single attribute types and values. By using the schema proposed in this document, index objects can include certificate information in attributes.

If certificates are stored redundantly in person entries and in certificate entries below the person entries, maintainers of repositories MUST make sure that the same certificates are stored in the person entry and the respective certificate entries and keep this consistency. Alternatively, they MUST leave out any certificates in the person entry.

This document is part of a set following this metadata approach comprising:
1. the LDAP schema for X.509 public key certificates (this document)
2. the LDAP schema for X.509 attribute certificates [ldap-ac-schema]
3. the LDAP schema for X.509 CRLs [ldap-crl-schema]

Future documents may be written that use the same method for Qualified certificates as described in [RFC3039] or any other evolving pkix certificate standard. An auxiliary object class for including additional metadata that is not included in the certificate is outside the scope of this document.

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 and consistency 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 rather be possible by using the solution proposed here due to its dependency on attribute values.

3. Comparison with Component Matching approach

[RFC3687] Component matching defines a mechanism for matching against complex syntaxes, by defining generic matching rules that can match against any user selected component parts in an attribute value of any arbitrarily complex attribute syntax. This might prove to be the proper way to solve LDAP search problems in the longer term, but it will take a long time until such ASN.1 based mechanisms are implemented in all LDAP servers and clients. Even when this has happened the mechanism proposed in this document will still be useful
to some applications such as CIP.

A simple and easy to implement mechanism is needed today to search for X.509 attributes.

4. X.509 certificate object classes

The object classes have been designed to form a logical set and be extensible in an orderly way as new PKC/CRL/AC extensions are defined. The methodology is as follows. Every X.509 entry (for a PKC, CRL or AC) is of the x509base abstract object class. There is then an additional abstract object class for each, derived from x509base, which holds the attributes extracted from the basic PKC/AC/CRL ASN.1 structure (excluding all extensions). The PKC object class is then instantiated by two structural object classes for user certificates and for CA certificates. The extensions are added by an additional auxiliary object class.

Thus the inheritance chains for PKCs are:

```
x509base          top
   \       |
     x509PKC  x509PKCext
       \     
         x509caCertificate  x509userCertificate
```

4.1 X.509 base object class

The x509base object class is the abstract object class that is the superior of all of the x.509 entry object classes

( 1.3.6.1.4.1.10126.1.5.4.2.1
  NAME 'x509base'
  ABSTRACT
  MAY x509version )

4.2 X.509 PKC 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 and in name forms. It is derived from the abstract object class
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'
  SUP x509base
  ABSTRACT
  MUST ( x509serialNumber $ x509signatureAlgorithm $ x509issuer $
    x509validityNotBefore $ x509validityNotAfter $ 
    x509subjectPublicKeyInfoAlgorithm )
  MAY ( x509certHolder $ x509issuerSerial ) )

The attribute description of x509issuerSerial can be found in [ldap-ac-schema]

4.3 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
  STRUCTURAL
  MUST userCertificate
  MAY x509subject )

The attribute description of userCertificate can be found in [pkix-ldap-schema]. Although this attribute type is specified as multi-valued it MUST NOT contain more than one certificate if used with this object class.

The attribute type x509subject is specified here as a MAY attribute. Nevertheless if this attribute is not used at least one of the following attributes MUST be filled in: x509subjectRfc822Name, x509subjectDnsName, x509subjectDirectoryName, x509subjectURI, x509subjectIpAddress, or x509subjectRegisteredID.

4.4 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
  STRUCTURAL
  MUST ( caCertificate $ x509subject ) )

The attribute description of caCertificate can be found in [pkix-ldap-schema]. Although this attribute type is specified as
multi-valued it MUST NOT contain more than one certificate if used with this object class.
4.5 X.509 PKC extensions auxiliary object class

The x509PKCext auxiliary object class is used to hold the attributes extracted from the PKC extensions defined in [X.509-2000] and profiled in [RFC3280].

Note. If a PKC holds additional extensions to these, then another auxiliary object class and supporting attributes will need to be defined.

( 1.3.6.1.4.1.10126.1.5.4.2.6
   NAME ’x509PKCext’
   SUP top
   AUXILIARY
   MAY ( x509authorityKeyIdIdentifier $ x509authorityCertIssuer $ x509authorityCertSerialNumber $ x509subjectKeyIdIdentifier $ x509keyUsage $ x509policyInformationIdentifier $ x509subjectRfc822Name $ x509subjectDnsName $ x509subjectDirectoryName $ x509subjectURI $ x509subjectIpAddress $ x509subjectRegisteredID $ x509issuerRfc822Name $ x509issuerDnsName $ x509issuerDirectoryName $ x509issuerURI $ x509issuerIpAddress $ x509issuerRegisteredID $ x509basicConstraintsCa $ x509basicConstraintsPathLen $ x509extKeyUsage $ x509fullCRLDistributionPointURI ) )

4.6 X.509 certificate holder object class

This auxiliary object class has an attribute that contains a pointer to an entry with x509certicate objectclass. Thus it is possible to link, e.g., an entry of a white pages directory to an entry in a certificate store. Such a link points to the opposite direction of the link stored in the attribute type x509certHolder.

( 1.3.6.1.4.1.10126.1.5.4.2.2
   NAME ’x509certificateHolder’
   AUXILIARY
   MAY ( x509certLocation ) )

5. The attribute types of the X.509 certificate object classes

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

5.1.1 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
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
  SINGLE-VALUE )

Values of this attribute may either be 0, 1, 2 or 3 corresponding to X.509 v1, v2, v3, or v4.

5.1.2 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 )

5.1.3 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 )
5.1.4 Issuer

String representation of the certificate or CRL 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'
   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].

5.1.5 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 begin and end of the certificate’s validity period. This sequence has been split up into two separate attributes "x509validityNotBefore" and "x509validityNotAfter". 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'
   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, it MUST be converted into GeneralisedTime format when creating the attribute value.
5.1.6 Subject

String representation of the subject’s distinguished name (see X.509(2000) 7, RFC3280 4.1.2.6).

(1.3.6.1.4.1.10126.1.5.3.7
  NAME ‘x509subject’
  DESC ‘Distinguished name of the entity associated with this public-key’
  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].

5.1.7 Subject public key info algorithm

OID identifying the algorithm associated with the certified public key (see X.509(2000) 7, RFC3280 4.1.2.7).

(1.3.6.1.4.1.10126.1.5.3.8
  NAME ‘x509subjectPublicKeyInfoAlgorithm’
  DESC ‘OID identifying the algorithm associated with the certified public key’
  EQUALITY objectIdentifierMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.38
  SINGLE-VALUE )

5.2 Attributes for selected extensions

As this specification intends to 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
- basic constraints extension
- name constraints extensions
- policy constraints extensions
- inhibit any policy extension
- freshest CRL extension
- authority information access extension
- subject information access extension
5.2.1 Authority key identifier extension

This attribute identifies the public key to be used to verify the signature on this certificate or CRL (see X.509(2000) 8.2.2.1, RFC3280 4.2.1.1). 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.

5.2.1.1 Authority key identifier

( 1.3.6.1.4.1.10126.1.5.3.11
   NAME 'x509authorityKeyIdentifier'
   DESC 'Key Identifier field of the Authority Key Identifier extension'
   EQUALITY octetStringMatch
   SYNTAX 1.3.6.1.4.1.1466.115.121.1.40
   SINGLE-VALUE )

5.2.1.2 Authority cert issuer

( 1.3.6.1.4.1.10126.1.5.3.12
   NAME 'x509authorityCertIssuer'
   DESC 'Authority Cert Issuer field of the Authority Key Identifier extension'
   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.

5.2.1.3 Authority cert serial number

( 1.3.6.1.4.1.10126.1.5.3.13
   NAME 'x509authorityCertSerialNumber'
   DESC 'Authority Cert Serial Number field of the Authority Key Identifier extension'
   EQUALITY integerMatch
   SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
   SINGLE-VALUE )
5.2.2 Subject key identifier extension

This attribute identifies the public key being certified (see X.509(2000) 8.2.2.2, RFC3280 4.2.1.2). It enables distinct keys used by the same subject to be differentiated.

\[
( 1.3.6.1.4.1.10126.1.5.3.14
   \begin{array}{ll}
   \text{NAME} & \text{'x509subjectKeyIdentifier'} \\
   \text{DESC} & \text{'Key identifier which must be unique with respect to all key identifiers for the subject'} \\
   \text{EQUALITY} & \text{octetStringMatch} \\
   \text{SYNTAX} & 1.3.6.1.4.1.1466.115.121.1.40 \\
   \text{SINGLE-VALUE} &
   \end{array}
)\
\]

5.2.3 Key usage extension

This attribute defines the purpose (e.g., encipherment, signature, certificate signing) of the key contained in the certificate (see X.509(2000) 8.2.2.3, RFC3280 4.2.1.3).

\[
( 1.3.6.1.4.1.10126.1.5.3.15
   \begin{array}{ll}
   \text{NAME} & \text{'x509keyUsage'} \\
   \text{DESC} & \text{'Purpose for which the certified public key is used'} \\
   \text{EQUALITY} & \text{caseIgnoreIA5Match} \\
   \text{SYNTAX} & 1.3.6.1.4.1.1466.115.121.1.26 
   \end{array}
)\
\]

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:

\[
x509keyUsage-value = "digitalSignature" / "nonRepudiation" / "keyEncipherment" / "dataEncipherment" / "keyAgreement" / "keyCertSign" / "cRLSign" / "encipherOnly" / "decipherOnly"
\]

5.2.4 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 (see X.509(2000) 8.2.2.6, RFC3280 4.2.1.5).

\[
( 1.3.6.1.4.1.10126.1.5.3.16
   \begin{array}{ll}
   \text{NAME} & \text{'x509policyInformationIdentifier'} \\
   \text{DESC} & \text{'OID which indicates the policy under which the certificate has been issued and the purposes for which}
   \end{array}
)\
\]
the certificate may be used'
EQUALITY objectIdentifierMatch
5.2.5 Subject alternative name extension

The subject alternative name extension allows additional identities to be bound to the subject of the certificate (see X.509(2000) 8.3.2.1, RFC3280 4.2.1.7). Separate attribute types are defined for all choices of the ASN.1 type "GeneralName" except for "otherName", "x400Address" and "ediPartyName".

5.2.5.1 Subject RFC822 name

( 1.3.6.1.4.1.10126.1.5.3.17
   NAME 'x509subjectRfc822Name'
   DESC 'Internet electronic mail address of the entity associated with this public-key'
   EQUALITY caseIgnoreIA5Match
   SUBSTR caseIgnoreIA5SubstringsMatch
   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].

5.2.5.2 Subject DNS name

( 1.3.6.1.4.1.10126.1.5.3.18
   NAME 'x509subjectDnsName'
   DESC 'Internet domain name of the entity associated with this public-key'
   EQUALITY caseIgnoreIA5Match
   SUBSTR caseIgnoreIA5SubstringsMatch
   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].

5.2.5.3 Subject directory name

( 1.3.6.1.4.1.10126.1.5.3.19
   NAME 'x509subjectDirectoryName'
   DESC 'Distinguished name of the entity associated with this public-key'
   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].
5.2.5.4 Subject Uniform Resource Identifier

( 1.3.6.1.4.1.10126.1.5.3.20
  NAME 'x509subjectURI'
  DESC 'Uniform Resource Identifier for the World-Wide Web of the entity associated with this public-key'
  EQUALITY caseExactIA5Match
  SUBSTR caseExactIA5SubstringsMatch
  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].

5.2.5.5 Subject IP address

( 1.3.6.1.4.1.10126.1.5.3.21
  NAME 'x509subjectIpAddress'
  DESC 'Internet Protocol address of the entity associated with this public-key'
  EQUALITY caseIgnoreIA5Match
  SUBSTR caseIgnoreIA5SubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

Values of this attribute type must be stored in the syntax given for IPv4address or IPv6address in Appendix B of [RFC2373].

5.2.5.6 Subject registered ID

( 1.3.6.1.4.1.10126.1.5.3.22
  NAME 'x509subjectRegisteredID'
  DESC 'OID of any registered object identifying the entity associated with this public-key'
  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.

5.2.6 Issuer alternative name extension

The issuer alternative names extension allows additional identities to be bound to the subject of the certificate or CRL (see X.509(2000) 8.3.2.2, RFC3280 4.2.1.8). Separate attribute types are defined for all choices of the ASN.1 type "GeneralName" except for "otherName", "x400Address" and "ediPartyName".
5.2.6.1 Issuer RFC 822 name

( 1.3.6.1.4.1.10126.1.5.3.23
  NAME 'x509issuerRfc822Name'
  DESC 'Internet electronic mail address of the entity who has
       signed and issued the certificate'
  EQUALITY caseIgnoreIA5Match
  SUBSTR caseIgnoreIA5SubstringsMatch
  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].

5.2.6.2 Issuer DNS name

( 1.3.6.1.4.1.10126.1.5.3.24
  NAME 'x509issuerDnsName'
  DESC 'Internet domain name of the entity who has
       signed and issued the certificate'
  EQUALITY caseIgnoreIA5Match
  SUBSTR caseIgnoreIA5SubstringsMatch
  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].

5.2.6.3 Issuer directory name

( 1.3.6.1.4.1.10126.1.5.3.25
  NAME 'x509issuerDirectoryName'
  DESC 'Distinguished name of the entity who has
       signed and issued the certificate'
  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].

5.2.6.4 Issuer Uniform Resource Identifier

( 1.3.6.1.4.1.10126.1.5.3.26
  NAME 'x509issuerURI'
  DESC 'Uniform Resource Identifier for the World-Wide Web
       of the entity who has signed and issued the certificate'
  EQUALITY caseExactIA5Match
  SUBSTR caseExactIA5SubstringsMatch )
Values of this attribute must be encoded according to the syntax given in [RFC2396].

5.2.6.5 Issuer IP address

( 1.3.6.1.4.1.10126.1.5.3.27
   NAME 'x509issuerIpAddress'
   DESC 'Internet Protocol address of the entity who has
         signed and issued the certificate'
   EQUALITY caseIgnoreIA5Match
   SUBSTR caseIgnoreIA5SubstringsMatch
   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].

5.2.6.6 Issuer registered ID

( 1.3.6.1.4.1.10126.1.5.3.28
   NAME 'x509issuerRegisteredID'
   DESC 'OID of any registered object identifying the entity who
         has signed and issued the certificate'
   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.

5.2.7 Basic constraints extension

The basic constraints extension (see X.509(2000) 8.4.2.1, RFC3280 4.2.1.10) identifies whether the subject of the certificate is a CA and the maximum depth of valid certification paths that include this certificate. This can be stored in the following two LDAP attributes:

The attribute x509basicConstraintsCa indicates whether the subject of the certificate is a CA. If the value of this attribute is "TRUE", the certificate MUST be stored in the "caCertificate" attribute.

( 1.3.6.1.4.1.10126.1.5.3.29
   NAME 'x509basicConstraintsCa'
   DESC 'Identifies whether the subject of the certificate is a CA'
   EQUALITY booleanMatch
   SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 )
The attribute x509basicConstraintsPathLen contains the maximum number of non-self-issued intermediate certificates that may follow this certificate in a valid certification path. It is only meaningful, if x509basicConstraintsCa is set to "TRUE".

```
( 1.3.6.1.4.1.10126.1.5.3.33
  NAME 'x509basicConstraintsPathLen'
  DESC 'maximum number of non-self-issued intermediate certificates that may follow this certificate in a valid certification path.'
  EQUALITY integerMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
  SINGLE-VALUE )
```

### 5.2.8 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 (see X.509(2000) 8.2.2.4, RFC3280 4.2.1.13). 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 an OID'
  EQUALITY objectIdentifierMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.38 )
```

### 5.2.9 CRL distribution points extension

This attribute identifies how the full CRL information for this certificate can be obtained (see X.509(2000) 8.6.2.1, RFC3280 4.2.1.14).

```
( 1.3.6.1.4.1.10126.1.5.3.32
  NAME 'x509fullCRLDistributionPointURI'
  DESC 'URI type of DistributionPointName for the full CRL'
  EQUALITY caseExactIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
```

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].

5.3 Additional attributes

5.3.1 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 )

5.3.2 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 )

6. 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 SHOULD 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 two name forms:
(1.3.6.1.4.1.10126.1.5.5.6
NAME 'x509userCertificateNameform')
OC x509userCertificate
MUST ( x509serialNumber $ x509issuer )

( 1.3.6.1.4.1.10126.1.5.5.7
  NAME 'x509caCertificateNameform'
  OC x509caCertificate
  MUST ( x509serialNumber $ x509issuer )
)

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

( 1.3.6.1.4.1.10126.1.5.5.8
  NAME 'x509PKCAltNameForm'
  OC x509PKC
  MUST x509issuerSerial )

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 two name forms:

( 1.3.6.1.4.1.10126.1.5.5.10
  NAME 'x509PKCSerialNumberNameForm'
  OC x509PKC
  MUST x509serialNumber )

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\2c OU\3d Class 1 Public Primary Certification Authority - G2\2cO\3d VeriSign\5c\2c Inc.\2cC\3dUS,cn=Joe Example,...
7. 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, secure authentication MUST have been used to identify the Client when an update operation is requested.

See [RFC2829] for additional information on how to protect sensitive LDAP data.

8. IANA Considerations

This document uses the OIDs below 1.3.6.1.4.1.10126.1.5 to identify the LDAP schema elements described here. This OID was assigned by DAASI International, under its IANA-assigned private enterprise allocation [PRIVATE], for use in this specification.

9. Acknowledgments

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

The authors wish to thank David Chadwick, Russ Housley, Mikhail Sahalayev, Michael Stroeder, and Kurt Zeilenga for their 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).
of Baden-Wuerttemberg, Germany.

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.

10. References

10.1 Normative references


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


10.2 Non-normative references


[RFC3039]  Santesson, S., Polk, T., Barzin, P. and M. Nystrom,
Appendix A. Sample directory entries

A sample x509certificate directory entry for an intermediate CA certificate in LDIF format:
dn: x509serialNumber=1429501,EMAILADDRESS=certify@pca.dfn.de,CN=DFN Top
oplevel Certification Authority,OU=DFN-PCA,OU=DFN-CERT GmbH,O=Deutsc
objectclass: x509base
objectclass: x509PKC
objectclass: x509caCertificate
objectclass: x509PKCext
x509version: 2
x509serialNumber: 1429501
x509issuer: EMAILADDRESS=certify@pca.dfn.de,CN=DFN Toplevel Certification Authority,OU=DFN-PCA,OU=DFN-CERT GmbH,O=Deutsches Forschungsnetz,C=DE
x509validityNotBefore: 20011201121116Z
x509validityNotAfter: 20100131121116Z
x509subject: EMAILADDRESS=certify@pca.dfn.de,CN=DFN Toplevel Certification Authority,OU=DFN-PCA,OU=DFN-CERT GmbH,O=Deutsches Forschungsnetz,C=DE
x509subjectPublicKeyInfoAlgorithm: 1.2.840.113549.1.1.1
x509signatureAlgorithm: 1.2.840.113549.1.1.5
x509basicConstraintsCa: TRUE
x509subjectKeyIdentifier:: Bgv6tfhIeKMgsQs+z6DQxNF/fdA=
x509authorityCertIssuer: EMAILADDRESS=certify@pca.dfn.de,CN=DFN Toplevel Certification Authority,OU=DFN-PCA,OU=DFN-CERT GmbH,O=Deutsches Forschungsnetz,C=DE
x509authorityCertSerialNumber: 1429501
x509keyUsage: keyCertSign
cRLSign
x509policyInformationIdentifier: 1.3.6.1.4.1.11418.300.1.1
caCertificate;binary:: MIIG2jCCBcKgAwIBAgIDFc/9MA0GCSqGSIb3DQEBBQUAMI GaMswQCVDDQVQQEwJERTehM88GA1UEChMYRGGV1dHJNajGZIZvencNajHVu23Nu2XR6MR
YWYAfAYDQVQLEwL4EktQ0VSVCBhBJMRAwgdYDVQQLEdERk4tUEy9BM0yWYDVQQLDey
REtRk4vG9wGBV2Zwq2VydGlmaWNhdGlvdBdXRcb3JpdHhxITATBgkghkig9Gy8QBCW
EEmNlcnRpZnlAcGhHlMmZb5ZTCAfEXYmDEyMExMjEwMTA7QmFaWxMDexMjEwMTAX
ZaMiG3MswQCVDDQVQQEwJERTehM88GA1UEChMYRGGV1dHJNajGZIZvencNajHVu23Nu2XR
R6MYWFAYDQVQLEwL4EktQ0VSVCBhBJMRAwgdYDVQQLEdERk4tUEy9BM0yWYDVQQLDey
QDEyREtRk4vG9wGBV2Zwq2VydGlmaWNhdGlvdBdXRcb3JpdHhxITATBgkghkig9Gy8QBCW
EEmNlcnRpZnlAcGhHlMmZb5ZTCAfEXYmDEyMExMjEwMTA7QmFaWxMDexMjEwMTAX
ZaMiG3MswQCVDDQVQQEwJERTehM88GA1UEChMYRGGV1dHJNajGZIZvencNajHVu23Nu2XR
R6MYWFAYDQVQLEwL4EktQ0VSVCBhBJMRAwgdYDVQQLEdERk4tUEy9BM0yWYDVQQLDey
QDEyREtRk4vG9wGBV2Zwq2VydGlmaWNhdGlvdBdXRcb3JpdHhxITATBgkghkig9Gy8QBCW
EEmNlcnRpZnlAcGhHlMmZb5ZTCAfEXYmDEyMExMjEwMTA7QmFaWxMDexMjEwMTAX
ZaMiG3MswQCVDDQVQQEwJERTehM88GA1UEChMYRGGV1dHJNajGZIZvencNajHVu23Nu2XR
R6MYWFAYDQVQLEwL4EktQ0VSVCBhBJMRAwgdYDVQQLEdERk4tUEy9BM0yWYDVQQLDey
QDEyREtRk4vG9wGBV2Zwq2VydGlmaWNhdGlvdBdXRcb3JpdHhxITATBgkghkig9Gy8QBCW
EEmNlcnRpZnlAcGhHlMmZb5ZTCAfEXYmDEyMExMjEwMTA7QmFaWxMDexMjEwMTAX
ZaMiG3MswQCVDDQVQQEwJERTehM88GA1UEChMYRGGV1dHJNajGZIZvencNajHVu23Nu2XR
R6MYWFAYDQVQLEwL4EktQ0VSVCBhBJMRAwgdYDVQQLEdERk4tUEy9BM0yWYDVQQLDey
QDEyREtRk4vG9wGBV2Zwq2VydGlmaWNhdGlvdBdXRcb3JpdHhxITATBgkghkig9Gy8QBCW
EEmNlcnRpZnlAcGhHlMmZb5ZTCAfEXYmDEyMExMjEwMTA7QmFaWxMDexMjEwMTAX
ZaMiG3MswQCVDDQVQQEwJERTehM88GA1UEChMYRGGV1dHJNajGZIZvencNajHVu23Nu2XR
R6MYWFAYDQVQLEwL4EktQ0VSVCBhBJMRAwgdYDVQQLEdERk4tUEy9BM0yWYDVQQLDey
QDEyREtRk4vG9wGBV2Zwq2VydGlmaWNhdGlvdBdXRcb3JpdHhxITATBgkghkig9Gy8QBCW
EEmNlcnRpZnlAcGhHlMmZb5ZTCAfEXYmDEyMExMjEwMTA7QmFaWxMDexMjEwMTAX
ZaMiG3MswQCVDDQVQQEwJERTehM88GA1UEChMYRGGV1dHJNajGZIZvencNajHVu23Nu2XR
R6MYWFAYDQVQLEwL4EktQ0VSVCBhBJMRAwgdYDVQQLEdERk4tUEy9BM0yWYDVQQLDey
QDEyREtRk4vG9wGBV2Zwq2VydGlmaWNhdGlvdBdXRcb3JpdHhxITATBgkghkig9Gy8QBCW
EEmNlcnRpZnlAcGhHlMmZb5ZTCAfEXYmDEyMExMjEwMTA7QmFaWxMDexMjEwMTAX
ZaMiG3MswQCVDDQVQQEwJERTehM88GA1UEChMYRGGV1dHJNajGZIZvencNajHVu23Nu2XR
R6MYWFAYDQVQLEwL4EktQ0VSVCBhBJMRAwgdYDVQQLEdERk4tUEy9BM0yWYDVQQLDey

A sample x509 certificate directory entry for an end identity certificate in LDIF format:
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=Peter%20Gietz,O=DAASI%20International%20GmbH,C=de?
    userCertificate?one?(objectClass=x509userCertificate)

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

ldap:///ou=DAASI%20CA,o=DAASI%20International%20GmbH,c=DE?
    userCertificate?sub?
    (&(objectClass=x509userCertificate)
        (x509subjectRfc822Name=peter@daasi.de)
        ((x509keyUsage=keyEncipherment)
            (x509extKeyUsage=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:///?caCertificate?sub?
    (&(objectClass=x509caCertificate)(x509subjectKeyIdentifier= %CE6%5C7A%5C16%5C4A%5C12%5C9F%5C22%5C09%5C6A
        %5C43%5C83%5C78%5C25%5C70%5C52%5C0%5C19))

Appendix C. Changes from previous Drafts

C.1 Changes in draft-klasen-ldap-x509certificate-schema-01
   o Included new Attributes x509authorityKeyId, x509authorityCertIssuer, x509authorityCertSerialNumber, x509certificateLocation, x509certificateHolder, and new objectclass x509certificateHolder
   o Fixed bug in definition of objectclass x509certificate
   o Changed references from RFC 2459 to RFC 3280 and included some respective language in 3.2.
   o Changed references from RFC 2251 to RFC 3377 and deleted all references to LDAPv2.
   o Deleted ":binary" in examples
   o Included new section: Comparison with component matching approach
   o Some changes in wording and section titles, and elimination of typos
   o Changed order of authors, and one author’s address

C.2 Changes in draft-klasen-ldap-x509certificate-schema-02
   o abstract object class x509PKC
   o aligned to [ldap-ac-schema] and [ldap-crl-schema]

C.3 Changes in draft-klasen-ldap-x509certificate-schema-03
   o Changed Matching Rules from caseIgnoreMatch to caseIgnoreIA5Match etc.
moved the references to RFC 3280 from the DESC part of the attribute definition to the text
added some additional text about CIP in Introduction
reworded text for x509subjectPublicKeyInfoAlgorithm
changed x509userCert and x509caCert to be inherited from userCertificate and caCertificate respectively
added clarification about x509subject and subject alternative names
added attribute type x509issuerSerial to x509PKC object class
added attribute type x509basicConstraintsCa to x509PKC object class
renamed attribute type x509cRLDistributionPointURI to x509FullcRLDistributionPointURI
devided references in normative and non normative
deleted attribute type mail from x509PKC objectclass
created separate Name Forms for x509userCertificate and x509caCertificate object classes.
changed attribute type x509SerialNumber to MULTI-VALUE
adjusted examples to new schema
Fixed more typos

C.4 Changes in draft-ietf-pkix-ldap-pkc-schema-00
renamed the title and file name
deleted attribute types x509userCert and x509caCert and replaced them with the standard userCertificate and caCertificate attribute types.
 included the description of x509base object class assigning a new OID to it.
separated the extensions attributes from the object class x509PKC and included them into the new auxiliary object class x509PKCext
renamed x509issuerUniformResourceIdentifier and x509subjectUniformResourceIdentifier to x509issuerURI and x509subjectURI respectively.
replaced separate Name Forms for x509userCertificate and x509caCertificate object classes by a single x509PKCNameForm.
 included a super section x509 Schema Object Classes with introductory remarks (inspired by [ldap-ac-schema])
added some additional wording and some ASCII art in the introduction
added some additional wording and some ASCII art in the introduction to the object classes
changed the MUST for using multi-valued RDNs into a SHOULD in section on DIT Structure and Naming
re-ordered the text so that the section on object classes precedes the section on attribute types
 included a reference to RFC 2829 into the security considerations
updated references
- added an IANA considerations section
- added another acknowledgement

C.5 Changes in draft-ietf-pkix-ldap-pkc-schema-01
- changed attribute type x509PolicyInformationIdentifier to MULTI-VALUE
- added new attribute for storing the basic constraints path length and included it into the x509PKCExt object class.
Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The Internet Society (2004). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.