Status of this Memo

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

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.

Copyright (C) The Internet Society (2002). All Rights Reserved.

Abstract

This document forms a certificate profile, based on RFC 3280, for identity certificates issued to physical persons.

The goal of this document is to define a general syntax independent of local legal requirements. The profile is however designed to allow further profiling in order to meet specific local needs.

The profile defines specific conventions for certificates that are qualified within a defined legal framework, named Qualified Certificates. The profile does however not define any legal requirements for such Qualified Certificates.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.
# Table of Contents

1 Introduction ................................................ 3
2 Requirements and Assumptions ................................ 3
2.1 Properties ................................................ 4
2.2 Statement of Purpose ...................................... 5
2.3 Policy Issues ............................................. 5
2.4 Uniqueness of names ...................................... 5
3 Certificate and Certificate Extensions Profile .............. 6
3.1 Basic Certificate Fields .................................. 6
3.1.1 Issuer ................................................ 6
3.1.2 Subject ................................................ 7
3.2 Certificate Extensions .................................... 9
3.2.1 Subject Directory Attributes ............................ 9
3.2.2 Certificate Policies ................................... 10
3.2.3 Key Usage ............................................. 11
3.2.4 Biometric Information ................................ 11
3.2.5 Qualified Certificate Statements ....................... 12
4 Security Considerations .................................... 14
5 References .................................................. 15
6 Intellectual Property Rights .................................. 16
A ASN.1 definitions ........................................... 17
A.1 1988 ASN.1 Module ........................................ 17
A.2 1993 ASN.1 Module ........................................ 20
B A Note on Attributes ......................................... 24
C Example Certificate .......................................... 25
C.1 ASN.1 Structure .......................................... 25
C.1.1 Extensions ............................................. 25
C.1.2 The certificate ......................................... 27
C.2 ASN.1 Dump ................................................ 30
C.3 DER-encoding .............................................. 33
C.4 CA’s public key ............................................ 33
Authors’ Addresses ............................................ 34
Full Copyright Statement ...................................... 35
1 Introduction

This specification is one part of a family of standards for the X.509 Public Key Infrastructure (PKI) for the Internet. It is based on RFC 3280, which defines underlying certificate formats and semantics needed for a full implementation of this standard.

This profile includes specific mechanisms intended for use with Qualified Certificates. The term Qualified Certificates and the assumptions that affects the scope of this document are discussed in Section 2.

Section 3 defines requirements on certificate information content. This profile addresses two fields in the basic certificate as well as five certificate extensions. The certificate fields are the subject and issuer fields. The certificate extensions are subject directory attributes, certificate policies, key usage, a private extension for storage of biometric data and a private extension for storage of statements related to Qualified Certificates. The private extensions are presented in the 1993 Abstract Syntax Notation One (ASN.1), but in conformance with RFC 3280 the 1988 ASN.1 module in Appendix A contains all normative definitions (the 1993 module in Appendix A is informative).

In Section 4, some security considerations are discussed in order to clarify the security context in which the standard may be utilized. Section 5 contains the references.

Appendix A contains all relevant ASN.1 [X.680] structures that are not already defined in RFC 3280. Appendix B contains a note on attributes. Appendix C contains an example certificate. Appendix D contains authors’ addresses and Appendix E contains the IETF Copyright Statement.

It should be noted that this specification does not define the specific semantics of Qualified Certificates, and does not define the policies that should be used with them. That is, this document defines what information should go into Qualified Certificates, but not what that information means. A system that uses Qualified Certificates must define its own semantics for the information in Qualified Certificates. It is expected that laws and corporate policies will make these definitions.

2 Requirements and Assumptions

The term "Qualified Certificate" has been used by the European Commission to describe a certain type of certificates with specific relevance for European legislation. This specification is intended
to support this class of certificates, but its scope is not limited to this application.

Within this standard the term "Qualified Certificate" is used generally, describing a certificate whose primary purpose is to identify a person with high level of assurance, where the certificate meet some qualification requirements defined by an applicable legal framework. The actual mechanisms that decide whether a certificate should or should not be considered to be a "Qualified Certificate" in regard to any legislation are outside the scope of this standard.

Harmonization in the field of identity certificates issued to physical persons, in particular Qualified Certificates, is essential within several aspects that fall outside the scope of RFC 3280. The most important aspects that affect the scope of this specification are:

- Definition of names and identity information in order to identify the associated subject in a uniform way.
- Definition of information which identifies the CA and the jurisdiction under which the CA operates when issuing a particular certificate.
- Definition of key usage extension usage for Qualified Certificates.
- Definition of information structure for storage of biometric information.
- Definition of a standardized way to store predefined statements with relevance for Qualified Certificates.
- Requirements for critical extensions.

2.1 Properties

This profile accommodates profiling needs for Qualified Certificates based on the assumptions that:

- Qualified Certificates are issued by a CA that makes a public statement that the certificate serves the purpose of a Qualified Certificate, as discussed in Section 2.2
- The Qualified Certificate indicates a certificate policy consistent with liabilities, practices and procedures undertaken by the CA, as discussed in 2.3

- The Qualified Certificate is issued to a natural person (living human being).

- The Qualified Certificate contains an identity based on a pseudonym or a real name of the subject.

2.2 Statement of Purpose

This profile defines conventions to declare within a certificate that it serves the purpose of being a Qualified Certificate. This enables the CA to explicitly define this intent.

The function of this declaration is thus to assist any concerned entity in evaluating the risk associated with creating or accepting signatures that are based on a Qualified Certificate.

This profile defines two complementary ways to include this information:

- As information defined by a certificate policy included in the certificate policies extension, and

- As a statement included in the Qualified Certificates Statements extension.

2.3 Policy Issues

Certain policy aspects define the context in which this profile is to be understood and used. It is however outside the scope of this profile to specify any policies or legal aspects that will govern services that issue or utilize certificates according to this profile.

It is however an underlying assumption in this profile that a responsible issuing CA will undertake to follow a publicly available certificate policy that is consistent with its liabilities, practices and procedures.

2.4 Uniqueness of names

Distinguished name is originally defined in X.501 [X.501] as a
representation of a directory name, defined as a construct that identifies a particular object from among the set of all objects. An object can be assigned a distinguished name without being represented by an entry in the Directory, but this name is then the name its object entry could have had if it were represented in the Directory. In the context of qualified certificates, a distinguished name denotes a set of attribute values [X.501] which forms a name that is unambiguous within a certain domain that forms either a real or a virtual DIT (Directory Information Tree)[X.501]. In the case of subject names the domain is assumed to be at least the issuing domain of the CA. The distinguished name MUST be unique for each subject entity certified by the one CA as defined by the issuer name field, during the whole life time of the CA.

3 Certificate and Certificate Extensions Profile

This section defines certificate profiling conventions. The profile is based on the Internet certificate profile RFC 3280 which in turn is based on the X.509 version 3 format. For full implementation of this section implementers are REQUIRED to consult the underlying formats and semantics defined in RFC 3280.

ASN.1 definitions relevant for this section that are not supplied by RFC 3280 are supplied in Appendix A.

3.1 Basic Certificate Fields

This specification provides additional details regarding the contents of two fields in the basic certificate. These fields are the issuer and subject fields.

3.1.1 Issuer

The issuer field SHALL identify the organization responsible for issuing the certificate. The name SHOULD be an officially registered name of the organization.

The identity of the issuer SHALL be specified using an appropriate subset of the following attributes:

domainComponent;
countryName;
stateOrProvinceName;
organizationName;
localityName; and
serialNumber.
Additional attributes MAY be present but they SHOULD NOT be necessary to identify the issuing organization. Attributes present in the issuer field SHOULD be consistent with the laws under which the issuer operates.

A relying party MAY have to consult associated certificate policies and/or the issuer's CPS, in order to determine the semantics of name fields and the laws under which the issuer operates.

3.1.2 Subject

The subject field of a certificate compliant with this profile SHALL contain a distinguished name of the subject (see 2.4 for definition of distinguished name).

The subject field SHALL contain an appropriate subset of the following attributes:

- domainComponent;
- countryName;
- commonName;
- surname;
- givenName;
- pseudonym;
- serialNumber;
- title;
- organizationName;
- organizationalUnitName;
- stateOrProvinceName; and
- localityName.

Other attributes may be present but MUST NOT be necessary to distinguish the subject name from other subject names within the issuer domain.

Of these attributes, the subject field SHALL include at least one of the following:

Choice I: commonName
Choice II: givenName
Choice III: pseudonym

The countryName attribute value specifies a general context in which other attributes are to be understood. The country
attribute does not necessarily indicate the subject’s country of
citizenship or country of residence, nor does it have to indicate
the country of issuance.

Note: Many X.500 implementations require the presence of countryName
in the DIT. In cases where the subject name, as specified in the
subject field, specifies a public X.500 directory entry, the
countryName attribute SHOULD always be present.

The commonName attribute value SHALL, when present, contain a name
of the subject. This MAY be in the subject’s preferred
presentation format, or a format preferred by the CA, or some
other format. Pseudonyms, nicknames and names with spelling other
than defined by the registered name MAY be used. To understand
the nature of the name presented in commonName, complying
applications MAY have to examine present values of the givenName
and surname attributes, or the pseudonym attribute.

Note: Many client implementations presuppose the presence of the
commonName attribute value in the subject field and use this value
to display the subject’s name regardless of present givenName,
surname or pseudonym attribute values.

The surname and givenName attribute types SHALL, if present,
contain the registered name of the subject, in accordance with the
laws under which the CA prepares the certificate. These
attributes SHALL be used in the subject field if the commonName
attribute is not present. In cases where the subject only has a
single name registered, the givenName attribute SHALL be used and
the surname attribute SHALL be omitted.

The pseudonym attribute type SHALL, if present, contain a
pseudonym of the subject. Use of the pseudonym attribute MUST NOT
be combined with use of any of the attributes surname and/or
givenName.

The serialNumber attribute type SHALL, when present, be used to
differentiate between names where the subject field would
otherwise be identical. This attribute has no defined semantics
beyond ensuring uniqueness of subject names. It MAY contain a
number or code assigned by the CA or an identifier assigned by a
government or civil authority. It is the CA’s responsibility to
ensure that the serialNumber is sufficient to resolve any subject
name collisions.

The title attribute type SHALL, when present, be used to store a
designated position or function of the subject within the
organization specified by present organizational attributes in the
subject field. The association between the title, the subject and the organization is beyond the scope of this document.

The organizationName and the organizationalUnitName attribute types SHALL, when present, be used to store the name and relevant information of an organization with which the subject is associated. The type of association between the organization and the subject is beyond the scope of this document.

The stateOrProvinceName and the localityName attribute types SHALL, when present, be used to store geographical information with which the subject is associated. If an organizationName value also is present then the stateOrProvinceName and localityName attribute values SHALL be associated with the specified organization. The type of association between the stateOrProvinceName and the localityName and either the subject or the organizationName is beyond the scope of this document.

Compliant implementations SHALL be able to interpret the attributes named in this section.

3.2 Certificate Extensions

This specification provides additional details regarding the contents of five certificate extensions. These extensions are the subject directory attributes, certificate policies, key usage, private extension for biometric information and private extension for Qualified Certificate statements.

3.2.1 Subject Directory Attributes

The subjectDirectoryAttributes extension MAY contain additional attributes, associated with the subject, as complement to present information in the subject field and the subject alternative name extension.

Attributes suitable for storage in this extension are attributes, which are not part of the subject’s distinguished name, but which MAY still be useful for other purposes (e.g., authorization).

This extension MUST NOT be marked critical.

Compliant implementations SHALL be able to interpret the following attributes:

    dateOfBirth;
placeOfBirth;
gender;
countryOfCitizenship; and
countryOfResidence.

Other attributes MAY be included according to local definitions.

The dateOfBirth attribute SHALL, when present, contain the value of the date of birth of the subject. The manner in which the date of birth is associated with the subject is outside the scope of this document.

The placeOfBirth attribute SHALL, when present, contain the value of the place of birth of the subject. The manner in which the place of birth is associated with the subject is outside the scope of this document.

The gender attribute SHALL, when present, contain the value of the gender of the subject. For females the value "F" (or "f") and for males the value "M" (or "m") have to be used. The manner in which the gender is associated with the subject is outside the scope of this document.

The countryOfCitizenship attribute SHALL, when present, contain the identifier of at least one of the subject’s claimed countries of citizenship at the time that the certificate was issued. If the subject is a citizen of more than one country, more than one country MAY be present. Determination of citizenship is a matter of law and is outside the scope of this document.

The countryOfResidence attribute SHALL, when present, contain the value of at least one country in which the subject is resident. If the subject is a resident of more than one country, more than one country MAY be present. Determination of residence is a matter of law and is outside the scope of this document.

3.2.2 Certificate Policies

The certificate policies extension SHALL contain the identifier of at least one certificate policy which reflects the practices and procedures undertaken by the CA. The certificate policy extension MAY be marked critical.

Information provided by the issuer stating the purpose of the certificate as discussed in Section 2.2 SHOULD be evident through indicated policies.
The certificate policies extension SHOULD include all policy information needed for validation of the certificate. If policy information is included in the QCStatements extension (see 3.2.5), then this information SHOULD also be defined by indicated policies.

Certificate policies MAY be combined with any qualifier defined in RFC 3280.

3.2.3 Key Usage

The key usage extension SHALL be present. Key usage settings SHALL be set in accordance with RFC 3280 definitions. Further requirements on key usage settings MAY be defined by local policy and/or local legal requirements.

The key usage extension MAY be marked critical.

3.2.4 Biometric Information

This section defines an extension for storage of biometric information. Biometric information is stored in the form of a hash of a biometric template.

The purpose of this extension is to provide means for authentication of biometric information. The biometric information that corresponds to the stored hash is not stored in this extension, but the extension MAY include an URI pointing to a location where this information can be obtained. If included, this URI does not imply that this is the only way to access this information.

It is RECOMMENDED that biometric information in this extension is limited to information types suitable for human verification, i.e., where the decision of whether the information is an accurate representation of the subject is naturally performed by a person. This implies a usage where the biometric information is represented by, for example, a graphical image displayed to the relying party, which MAY be used by the relying party to enhance identification of the subject.

This extension MUST NOT be marked critical.

```
bioMetricInfo  EXTENSION ::= {
  SYNTAX BiometricSyntax
  IDENTIFIED BY id-pe-bioMetricInfo }

id-pe-bioMetricInfo OBJECT IDENTIFIER ::= {id-pe 2}

BiometricSyntax ::= SEQUENCE OF BiometricData
```
BiometricData ::= SEQUENCE {
  typeOfBiometricData  TypeOfBiometricData,
  hashAlgorithm        AlgorithmIdentifier,
  biometricDataHash    OCTET STRING,
  sourceDataUri        IA5String OPTIONAL }

TypeOfBiometricData ::= CHOICE {
  predefinedBiometricType    PredefinedBiometricType,
  biometricDataID            OBJECT IDENTIFIER }

PredefinedBiometricType ::= INTEGER { picture(0),
                                      handwritten-signature(1)}

The predefined biometric type picture, when present, SHALL identify
that the source picture is in the form of a displayable graphical
image of the subject. The hash of the graphical image SHALL only be
calculated over the image data excluding any labels defining the
image type.

The predefined biometric type handwritten-signature, when present,
SHALL identify that the source data is in the form of a displayable
graphical image of the subject’s handwritten signature. The hash of
the graphical image SHALL only be calculated over the image data
excluding any labels defining the image type.

3.2.5 Qualified Certificate Statements

This section defines an extension for inclusion of statements
defining explicit properties of the certificate.

A statement suitable for inclusion in this extension MAY be a
statement by the issuer that the certificate is issued as a Qualified
Certificate in accordance with a particular legal system (as
discussed in Section 2.2).

Other statements suitable for inclusion in this extension MAY be
statements related to the applicable legal jurisdiction within which
the certificate is issued. As an example this MAY include a maximum
reliance limit for the certificate indicating restrictions on CA’s
liability.

Each statement SHALL include an object identifier for the statement
and MAY also include optional qualifying data contained in the
statementInfo parameter.

If the statementInfo parameter is included then the object identifier
of the statement SHALL define the syntax and SHOULD define the
semantics of this parameter. If the object identifier does not
define the semantics, a relying party may have to consult a relevant certificate policy or CPS to determine the exact semantics.

This extension may be critical or non-critical. If the extension is critical, this means that all statements included in the extension are regarded as critical.

qcStatements  EXTENSION ::= {
  SYNTAX QCStatements
  IDENTIFIED BY id-pe-qcStatements }

id-pe-qcStatements OBJECT IDENTIFIER ::= { id-pe 3 }

QCStatements ::= SEQUENCE OF QCStatement

QCStatement ::= SEQUENCE {
  statementId   QC-STATEMENT.&Id({SupportedStatements}),
  statementInfo QC-STATEMENT.&Type
  ({SupportedStatements}@statementId) OPTIONAL }

SupportedStatements QC-STATEMENT ::= { qcStatement-1,...}

3.2.5.1 Predefined Statements

This profile includes one predefined object identifier (id-qcs-pkixQCSyntax-v1), identifying conformance with syntax and semantics defined in this profile. This Qualified Certificate profile is referred to as version 1.

qcStatement-1 QC-STATEMENT ::= { SYNTAX SemanticsInformation
  IDENTIFIED BY id-qcs-pkixQCSyntax-v1 }

-- This statement identifies conformance with syntax and semantics defined in this Qualified Certificate profile
-- (Version 1). The SemanticsInformation may optionally contain additional semantics information as specified.

SemanticsInformation ::= SEQUENCE {
  semanticsIdentifier OBJECT IDENTIFIER OPTIONAL,
  nameRegistrationAuthorities NameRegistrationAuthorities
  OPTIONAL } \n  WITH COMPONENTS {..., semanticsIdentifier PRESENT} | \n  WITH COMPONENTS {..., nameRegistrationAuthorities PRESENT})

NameRegistrationAuthorities ::= SEQUENCE SIZE (1..MAX) OF GeneralName

The SemanticsInformation component identified by id-qcs-pkixQCSyntax-v1 MAY contain a semantics identifier and MAY identify
one or more name registration authorities.

The semanticsIdentifier component, if present, SHALL contain an OID, defining semantics for attributes and names in basic certificate fields and certificate extensions. The OID may define semantics for all, or for a subgroup of all present attributes and/or names.

The NameRegistrationAuthorities component, if present, SHALL contain a name of one or more name registration authorities, responsible for registration of attributes or names associated with the subject. The association between an identified name registration authority and present attributes MAY be defined by a semantics identifier OID, by a certificate policy (or CPS) or some other implicit factors.

If a value of type SemanticsInformation is present in a QCStatement then at least one of the fields semanticsIdentifier and nameRegistrationAuthorities must be present, as indicated.

4 Security Considerations

The legal value of a digital signature that is validated with a Qualified Certificate will be highly dependent upon the policy governing the use of the associated private key. Both the private key holder as well as the relying party should make sure that the private key is used only with the consent of the legitimate key holder.

Since the public keys are for public use with legal implications for involved parties, certain conditions should exist before CAs issue certificates as Qualified Certificates. The associated private keys must be unique for the subject, and must be maintained under the subject’s sole control. That is, a CA should not issue a qualified certificate if the means to use the private key is not protected against unintended usage. This implies that the CA have some knowledge about the subject’s cryptographic module.

The CA must further verify that the public key contained in the certificate is legitimately representing the subject.

CAs should not issue CA certificates with policy mapping extensions indicating acceptance of another CA’s policy unless these conditions are met.

Combining the nonRepudiation bit in the keyUsage certificate extension with other keyUsage bits may have security implications and this specification therefore recommends against such practices.

The ability to compare two qualified certificates to determine if
they represent the same physical entity is dependent on the semantics of the subjects' names. The semantics of a particular attribute may be different for different issuers. Comparing names without knowledge of the semantics of names in these particular certificates may provide misleading results.

This specification is a profile of RFC 3280. The security considerations section of that document applies to this specification as well.

5 References


6 Intellectual Property Rights

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF’s procedures with respect to rights in standards-track and standards related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.
A. ASN.1 definitions

As in RFC 3280, ASN.1 modules are supplied in two different variants of the ASN.1 syntax.

Appendix A.1 is in the 1988 syntax, and does not use macros. However, since the module imports type definitions from modules in RFC 3280 which are not completely in the 1988 syntax, the same comments as in RFC 3280 regarding its use applies here as well; i.e., Appendix A.1 may be parsed by an 1988 ASN.1-parser by removing the definitions for the UNIVERSAL types and all references to them in RFC 3280's 1988 modules.

Appendix A.2 is in the 1997 syntax.

In case of discrepancies between these modules, the 1988 module is the normative one.

A.1 1988 ASN.1 Module

PKIXqualified88 {iso(1) identified-organization(3) dod(6)
internet(1) security(5) mechanisms(5) pkix(7) id-mod(0)
id-mod-qualified-cert(??) }

-- Do something about that version!--

DEFINITIONS EXPLICIT TAGS ::= 

BEGIN 

-- EXPORTS ALL --

IMPORTS

GeneralName  
FROM PKIX1Implicit88 {iso(1) identified-organization(3) dod(6)
internet(1) security(5) mechanisms(5) pkix(7) id-mod(0)
id-pkix1-implicit(19)}

AlgorithmIdentifier, DirectoryString, AttributeType, id-pkix, id-pe
FROM PKIX1Explicit88 {iso(1) identified-organization(3) dod(6)
internet(1) security(5) mechanisms(5) pkix(7) id-mod(0)
id-pkix1-explicit(18)};

-- Locally defined OIDs

-- Arc for QC personal data attributes
id-pda  OBJECT IDENTIFIER ::= { id-pkix 9 }
-- Arc for QC statements

Santesson, Nystrom, & Polk [Page 17]
id-qcs  OBJECT IDENTIFIER ::= { id-pkix 11 }

-- Attributes

id-pda-dateOfBirth  AttributeType ::= { id-pda 1 }  
DateOfBirth ::= GeneralizedTime

id-pda-placeOfBirth  AttributeType ::= { id-pda 2 }  
PlaceOfBirth ::= DirectoryString

id-pda-gender  AttributeType ::= { id-pda 3 }  
Gender ::= PrintableString (SIZE(1))  
-- "M", "F", "m" or "f"

id-pda-countryOfCitizenship  AttributeType ::= { id-pda 4 }  
CountryOfCitizenship ::= PrintableString (SIZE (2))  
-- ISO 3166 Country Code

id-pda-countryOfResidence  AttributeType ::= { id-pda 5 }  
CountryOfResidence ::= PrintableString (SIZE (2))  
-- ISO 3166 Country Code

-- Private extensions

-- Biometric info extension

id-pe-biometricInfo OBJECT IDENTIFIER ::= {id-pe 2}

BiometricSyntax ::= SEQUENCE OF BiometricData

BiometricData ::= SEQUENCE {
  typeOfBiometricData  TypeOfBiometricData,
  hashAlgorithm        AlgorithmIdentifier,
  biometricDataHash    OCTET STRING,
  sourceDataUri        IA5String OPTIONAL }

TypeOfBiometricData ::= CHOICE {
  predefinedBiometricType  PredefinedBiometricType,
  biometricDataOid         OBJECT IDENTIFIER }

PredefinedBiometricType ::= INTEGER {
  picture(0), handwritten-signature(1)}

-- QC Statements Extension

id-pe-qcStatements OBJECT IDENTIFIER ::= { id-pe 3}
QCStatements ::= SEQUENCE OF QCStatement

QCStatement ::= SEQUENCE {
    statementId OBJECT IDENTIFIER,
    statementInfo ANY DEFINED BY statementId OPTIONAL}

-- QC statements
id-qcs-pkiQCSyntax-v1 OBJECT IDENTIFIER ::= { id-qcs 1 }

-- This statement identifies conformance with syntax and
-- semantics defined in this Qualified Certificate profile
-- (Version 1). This statement may optionally contain
-- additional semantics information as specified below.

SemanticsInformation ::= SEQUENCE {
    semanticsIdentifier OBJECT IDENTIFIER OPTIONAL,
    nameRegistrationAuthorities NameRegistrationAuthorities OPTIONAL
} -- At least one field shall be present

NameRegistrationAuthorities ::= SEQUENCE SIZE (1..MAX) OF GeneralName

END
A.2 1993 ASN.1 Module

PKIXqualified97 {iso(1) identified-organization(3) dod(6)
   internet(1) security(5) mechanisms(5) pkix(7) id-mod(0)
   id-mod-qualified-cert-97(??) }
-- !Fix the number above!--
DEFINITIONS EXPLICIT TAGS ::= 

BEGIN

-- EXPORTS ALL --

IMPORTS

informationFramework, certificateExtensions, upperBounds,
selectedAttributeTypes, authenticationFramework, upperBounds, id-at
FROM UsefulDefinitions {joint-iso-itu-t(2) ds(5) module(1)
   usefulDefinitions(0) 2 }
ub-name
   FROM UpperBounds upperBounds

authorityKeyIdentifier, subjectKeyIdentifier, keyUsage,
extKeyUsage, privateKeyUsagePeriod, certificatePolicies,
policyMappings, subjectAltName, issuerAltName, basicConstraints,
nameConstraints, policyConstraints, cRLDistributionPoints,
subjectDirectoryAttributes, GeneralName, OTHER-NAME
   FROM CertificateExtensions certificateExtensions

ATTRIBUTE, AttributeType
   FROM InformationFramework informationFramework

title, organizationName, organizationalUnitName, stateOrProvinceName,
localityName, countryName, generationQualifier, dnQualifier, initials,
givenName, surname, commonName, name, serialNumber, pseudonym,
DirectoryString()
   FROM SelectedAttributeTypes selectedAttributeTypes

AlgorithmIdentifier, Extension, EXTENSION
   FROM AuthenticationFramework authenticationFramework

emailAddress
   FROM PKCS-9 {iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
pkcs-9(9) modules(0) pkcs-9(1)}

id-pkix, id-pe, id-domainComponent
   FROM PKIX1Explicit88 { iso(1) identified-organization(3) dod(6)
   internet(1) security(5) mechanisms(5) pkix(7) id-mod(0)
id-pkix1-explicit(18) }

id-pe-authorityInfoAccess, AuthorityInfoAccessSyntax
  FROM PKIX1Implicit88 { iso(1) identified-organization(3) dod(6)
  internet(1) security(5) mechanisms(5) pkix(7) id-mod(0)
  id-pkix1-implicit(19) };

-- Object Identifiers

-- Arc for QC personal data attributes
id-pda  OBJECT IDENTIFIER ::= { id-pkix 9 }

-- Arc for QC statements
id-qcs  OBJECT IDENTIFIER ::= { id-pkix 11 }

-- Private extensions
id-pe-biometricInfo OBJECT IDENTIFIER ::= { id-pe 2 }
id-pe-qcStatements  OBJECT IDENTIFIER ::= { id-pe 3 }

-- Personal data attributes
id-pda-dateOfBirth          AttributeType ::= { id-pda 1 }
id-pda-placeOfBirth         AttributeType ::= { id-pda 2 }
id-pda-gender               AttributeType ::= { id-pda 3 }
id-pda-countryOfCitizenship AttributeType ::= { id-pda 4 }
id-pda-countryOfResidence   AttributeType ::= { id-pda 5 }

-- QC statements
id-qcs-pkixQCSyntax-v1      OBJECT IDENTIFIER ::= { id-qcs 1 }

-- Object Sets

-- The following information object set is defined to constrain the
-- set of legal certificate extensions.
ExtensionSet EXTENSION ::= {
  authorityKeyIdentifier |
  subjectKeyIdentifier |
  keyUsage |
  extKeyUsage |
  privateKeyUsagePeriod |
  certificatePolicies |
  policyMappings |
  subjectAltName |
  issuerAltName |
  basicConstraints |
  nameConstraints |
  policyConstraints |
  cRLDistributionPoints |
  subjectDirectoryAttributes |
authorityInfoAccess | biometricInfo |
qcStatements, ... }

-- The following information object set is defined to constrain the set of attributes applications are required to recognize in distinguished names. The set may of course be augmented to meet local requirements. Note that deleting members of the set may prevent interoperability with conforming implementations.
SupportedAttributes ATTRIBUTE ::= {
countryName | commonName | surname | givenName | pseudonym |
serialNumber | organizationName | organizationalUnitName |
stateOrProvinceName | localityName | title |
emailAddress | domainComponent | dnQualifier,
... -- For future extensions -- }

-- The following information object set is defined to constrain the set of attributes applications are required to recognize in subjectDirectoryAttribute extensions. The set may be augmented to meet local requirements. Note that deleting members of the set may prevent interoperability with conforming implementations.
PersonalDataAttributeSet ATTRIBUTE ::= {
dateOfBirth | placeOfBirth | gender | countryOfCitizenship |
countryOfResidence, ... }

-- Attributes

-- authorityInfoAccess from RFC 3280
authorityInfoAccess ATTRIBUTE ::= {
  WITH SYNTAX AuthorityInfoAccessSyntax
  ID id-pe-authorityInfoAccess }

-- domainComponent from RFC 2247
domainComponent ATTRIBUTE ::= {
  WITH SYNTAX IA5String
  ID id-domainComponent }

dateOfBirth ATTRIBUTE ::= {
  WITH SYNTAX GeneralizedTime
  ID id-pda-dateOfBirth }

placeOfBirth ATTRIBUTE ::= {
  WITH SYNTAX DirectoryString { ub-name }
  ID id-pda-placeOfBirth }

gender ATTRIBUTE ::= {
  WITH SYNTAX PrintableString { size(1) ^ from("M"|"F"|"m"|"f") }
  ID id-pda-gender }
countryOfCitizenship ATTRIBUTE ::= {
    WITH SYNTAX PrintableString (SIZE (2))
    (CONSTRAINED BY { -- ISO 3166 codes only -- })
    ID id-pda-countryOfCitizenship }

countryOfResidence ATTRIBUTE ::= {
    WITH SYNTAX PrintableString (SIZE (2))
    (CONSTRAINED BY { -- ISO 3166 codes only -- })
    ID id-pda-countryOfResidence }

-- Private extensions

-- Biometric info extension

biometricInfo EXTENSION ::= {
    SYNTAX BiometricSyntax
    IDENTIFIED BY id-pe-biometricInfo }

BiometricSyntax ::= SEQUENCE OF BiometricData

BiometricData ::= SEQUENCE {
    typeOfBiometricData TypeOfBiometricData,
    hashAlgorithm AlgorithmIdentifier,
    biometricDataHash OCTET STRING,
    sourceDataUri IA5String OPTIONAL,
    ... -- For future extensions -- }

TypeOfBiometricData ::= CHOICE {
    predefinedBiometricType PredefinedBiometricType,
    biometricDataOid OBJECT IDENTIFIER }

PredefinedBiometricType ::= INTEGER { picture(0),
    handwritten-signature(1)} (picture|handwritten-signature,...)

-- QC Statements Extension

qcStatements EXTENSION ::= {
    SYNTAX QCStatements
    IDENTIFIED BY id-pe-qcStatements }

QCStatements ::= SEQUENCE OF QCStatement

QCStatement ::= SEQUENCE {
    statementId QC-STATEMENT.&id({SupportedStatements}),
    statementInfo QC-STATEMENT.&Type
    {({SupportedStatements}@statementId}) OPTIONAL }

QC-STATEMENT ::= CLASS {

&id OBJECT IDENTIFIER UNIQUE,
&Type OPTIONAL }
WITH SYNTAX {
  [SYNTAX &Type] IDENTIFIED BY &id }

qcStatement-1 QC-STATEMENT ::= { SYNTAX SemanticsInformation
  IDENTIFIED BY id-qcs-pkixQCSyntax-v1}
-- This statement identifies conformance with syntax and
-- semantics defined in this Qualified Certificate profile
-- (Version 1). The SemanticsInformation may optionally contain
-- additional semantics information as specified.

SemanticsInformation ::= SEQUENCE {
  semanticsIdentifier OBJECT IDENTIFIER OPTIONAL,
  nameRegistrationAuthorities NameRegistrationAuthorities OPTIONAL
}(WITH COMPONENTS {..., semanticsIdentifier PRESENT}|
  WITH COMPONENTS {..., nameRegistrationAuthorities PRESENT})

NameRegistrationAuthorities ::= SEQUENCE SIZE (1..MAX) OF GeneralName

-- The following information object set is defined to constrain the
-- set of attributes applications are required to recognize as QCSs.
SupportedStatements QC-STATEMENT ::= {
  qcStatement-1, ... -- For future extensions -- }

END

B. A Note on Attributes

This document defines several new attributes, both for use in the
subject field of issued certificates and in the
subjectDirectoryAttributes extension. A complete definition of these
new attributes (including matching rules), along with object classes
to support them in LDAP-accessible directories, can be found in [PKCS
9].
C. Example Certificate

This section contains the ASN.1 structure, an ASN.1 dump, and the DER-encoding of a certificate issued in conformance with this profile. The example has been developed with the help of the OSS ASN.1 compiler. The certificate has the following characteristics:

1. The certificate is signed with RSA and the SHA-1 hash algorithm
2. The issuer’s distinguished name is O=GMD – Forschungszentrum Informationstechnik GmbH; C=DE
3. The subject’s distinguished name is CN=Petra M. Barzin, O=GMD – Forschungszentrum Informationstechnik GmbH, C=DE
4. The certificate was issued on May 1, 2000 and will expire on November 1, 2000
5. The certificate contains a 1024 bit RSA key
6. The certificate includes a critical key usage extension exclusively indicating non-repudiation
7. The certificate includes a certificate policy identifier extension indicating the practices and procedures undertaken by the issuing CA (object identifier 1.3.36.8.1.1). The certificate policy object identifier is defined by TeleTrust, Germany. It is required to be set in a certificate conformant to the German digital signature law.
8. The certificate includes a subject directory attributes extension containing the following attributes:
   surname: Barzin
   given name: Petra
   date of birth: October, 14th 1971
   place of birth: Darmstadt
   country of citizenship: Germany
   gender: Female
9. The certificate includes a qualified statement private extension indicating that the naming registration authority’s name as "municipality@darmstadt.de".
10. The certificate includes, in conformance with RFC 3280, an authority key identifier extension.

C.1 ASN.1 Structure

C.1.1 Extensions

Since extensions are DER-encoded already when placed in the structure to be signed, they are for clarity shown here in the value notation defined in [X.680].

C.1.1.1 The subjectDirectoryAttributes extension

petrasSubjDirAttrs AttributesSyntax := {
C.1.1.2 The keyUsage extension
petrasKeyUsage KeyUsage ::= {nonRepudiation}

C.1.1.3 The certificatePolicies extension
    policyIdentifier {1 3 36 8 1 1}
}

C.1.1.4 The qcStatements extension
petrasQCStatement QCStatements ::= {
    statementId id-qcs-pkixQCSyntax-v1,
    statementInfo SemanticsInformation : {
        nameRegistrationAuthorities {
            rfc822Name : "municipality@darmstadt.de"
        }
    }
}

C.1.1.5 The authorityKeyIdentifier extension

petrasAKI AuthorityKeyIdentifier ::= {
    keyIdentifier '000102030405060708090A0B0C0D0E0FFEDCBA98'H
}

C.1.2 The certificate

The signed portion of the certificate is shown here in the value notation defined in [X.680]. Note that extension values are already DER encoded in this structure. Some values has been truncated for readability purposes.

{  
    version v3,
    serialNumber 1234567890,
    signature
    {  
        algorithm { 1 2 840 113549 1 1 5 },
        parameters RSAParams : NULL
    },
    issuer rdnSequence :
    {  
        {  
            type { 2 5 4 6 },
            value PrintableString : "DE"
        },
        {  
            type { 2 5 4 10 },
            value UTF8String :
            "GMD - Forschungszentrum Informationstechnik GmbH"
        }
    },
    validity
    {  
        notBefore utcTime : "000501100000Z",
        notAfter utcTime : "001101100000Z"
    },
    subject rdnSequence :
    {  
        {  
            type { 2 5 4 6 },
            value PrintableString : "DE"
        },
        {  
            type { 2 5 4 10 },
            value UTF8String :
            "GMD - Forschungszentrum Informationstechnik GmbH"
        }
    }
}

Santesson, Nystrom, & Polk
type { 2 5 4 6 },
  value PrintableString : "DE"
},
{
  type { 2 5 4 10 },
  value UTF8String :
    "GMD Forschungszentrum Informationstechnik GmbH"
},
{
  
  type { 2 5 4 4 },
  value UTF8String : "Barzin"
},
{
  type { 2 5 4 42 },
  value UTF8String : "Petra"
}
}
},
subjectPublicKeyInfo
{
  algorithm
  {
    algorithm { 1 2 840 113549 1 1 1 },
    parameters RSAParams : NULL
  },
subjectPublicKey '00110000 10000001 10000111 00000010 1000 ...'B
},
extensions
{
  
  extnId { 2 5 29 9 }, -- subjectDirectoryAttributes
  extnValue '305B301006082B06010505070904310413024445300F0 ...'H
  },
  
  extnId { 2 5 29 15 }, -- keyUsage
  critical TRUE,
  extnValue '03020640'H
  },
  
  extnId { 2 5 29 32 }, -- certificatePolicies
  extnValue '3009300706052B24080101'H
  },
  
  extnId { 2 5 29 35 }, -- authorityKeyIdentifier
extnValue '30168014000102030405060708090A0B0C0D0E0FEDCBA98'H
},
{
  extnId { 1 3 6 1 5 5 7 1 3 }, -- qcStatements
  extnValue '302B302906082B06010505070B01301D301B81196D756 ...'H
}
}
C.2 ASN.1 dump

This section contains an ASN.1 dump of the signed portion of the certificate. Some values have been truncated for readability purposes.

TBSCertificate SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 631
  version : tag = [0] constructed; length = 3
    Version INTEGER: tag = [UNIVERSAL 2] primitive; length = 1
      2
  serialNumber CertificateSerialNumber INTEGER: tag = [UNIVERSAL 2] primitive; length = 4
    1234567890
  signature AlgorithmIdentifier SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 13
    algorithm OBJECT IDENTIFIER: tag = [UNIVERSAL 6] primitive; length = 9
      { 1 2 840 113549 1 1 5 }
    parameters OpenType: NULL: tag = [UNIVERSAL 5] primitive; length = 0
      NULL
  issuer Name CHOICE
    rdnSequence RDNSequence SEQUENCE OF: tag = [UNIVERSAL 16] constructed; length = 72
      RelativeDistinguishedName SET OF: tag = [UNIVERSAL 17] constructed; length = 11
        AttributeTypeAndValue SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 9
          type OBJECT IDENTIFIER: tag = [UNIVERSAL 6] primitive; length = 3
            { 2 5 4 6 }
          value OpenType: PrintableString: tag = [UNIVERSAL 19] primitive; length = 2
            "DE"
        RelativeDistinguishedName SET OF: tag = [UNIVERSAL 17] constructed; length = 57
          AttributeTypeAndValue SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 55
            type OBJECT IDENTIFIER: tag = [UNIVERSAL 6] primitive; length = 3
              { 2 5 4 10 }
            value OpenType : UTF8String: tag = [UNIVERSAL 12] primitive; length = 48
              0x474d44202d20506f72736368756e7472756d2049... validity Validity SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 30
      notBefore Time CHOICE

utcTime UTCTime: tag = [UNIVERSAL 23] primitive; length = 13
   000501100000Z

notAfter Time CHOICE
   utcTime UTCTime: tag = [UNIVERSAL 23] primitive; length = 13
   001101100000Z

subject Name CHOICE
   rdnSequence RDNSequence SEQUENCE OF: tag = [UNIVERSAL 16]
      constructed; length = 101
   RelativeDistinguishedName SET OF: tag = [UNIVERSAL 17]
      constructed; length = 11
   AttributeTypeAndValue SEQUENCE: tag = [UNIVERSAL 16]
      constructed; length = 9
      type OBJECT IDENTIFIER: tag = [UNIVERSAL 6] primitive;
      length = 3
      { 2 5 4 6 }
      value OpenType: PrintableString: tag = [UNIVERSAL 19]
      primitive; length = 2
      "DE"

   RelativeDistinguishedName SET OF: tag = [UNIVERSAL 17]
      constructed; length = 55
   AttributeTypeAndValue SEQUENCE: tag = [UNIVERSAL 16]
      constructed; length = 53
      type OBJECT IDENTIFIER: tag = [UNIVERSAL 6] primitive;
      length = 3
      { 2 5 4 10 }
      value OpenType: UTF8String: tag = [UNIVERSAL 12]
      primitive; length = 46
      0x474d4420466f72736368756e67737a656e7472756d20496e...

   RelativeDistinguishedName SET OF: tag = [UNIVERSAL 17]
      constructed; length = 29
   AttributeTypeAndValue SEQUENCE: tag = [UNIVERSAL 16]
      constructed; length = 13
      type OBJECT IDENTIFIER: tag = [UNIVERSAL 6] primitive;
      length = 3
      { 2 5 4 4 }
      value OpenType: UTF8String: tag = [UNIVERSAL 12]
      primitive; length = 6
      0x4261727a696e

   AttributeTypeAndValue SEQUENCE: tag = [UNIVERSAL 16]
      constructed; length = 12
      type OBJECT IDENTIFIER: tag = [UNIVERSAL 6] primitive;
      length = 3
      { 2 5 4 42 }
      value OpenType: UTF8String: tag = [UNIVERSAL 12]
      primitive; length = 5
      0x5065747261

subjectPublicKeyInfo SubjectPublicKeyInfo SEQUENCE: tag = 
   [UNIVERSAL 16] constructed; length = 157
algorithm AlgorithmIdentifier SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 13
algorithm OBJECT IDENTIFIER: tag = [UNIVERSAL 6] primitive;
    length = 9
    { 1 2 840 113549 1 1 1 }
parameters OpenType: NULL: tag = [UNIVERSAL 5] primitive;
    length = 0
NULL
subjectPublicKey BIT STRING: tag = [UNIVERSAL 3] primitive;
    length = 139
0x0030818702818100b8488400d4b6088be48ead459ca19ec717aaf3d1d...
extensions : tag = [3] constructed; length = 233
Extensions SEQUENCE OF: tag = [UNIVERSAL 16] constructed;
    length = 230
Extension SEQUENCE: tag = [UNIVERSAL 16] constructed;
    length = 100
    extnId OBJECT IDENTIFIER: tag = [UNIVERSAL 6] primitive;
        length = 3
        { 2 5 29 9 }
    extnValue OCTET STRING: tag = [UNIVERSAL 4] primitive;
        length = 93
        0x305b301006082b06010505070904310413024445300f06082b060...
extension SEQUENCE: tag = [UNIVERSAL 16] constructed;
    length = 14
    extnId OBJECT IDENTIFIER: tag = [UNIVERSAL 6] primitive;
        length = 3
        { 2 5 29 15 }
critical BOOLEAN: tag = [UNIVERSAL 1] primitive; length = 1
    TRUE
    extnValue OCTET STRING: tag = [UNIVERSAL 4] primitive;
        length = 4
        0x03020640
Extension SEQUENCE: tag = [UNIVERSAL 16] constructed;
    length = 18
    extnId OBJECT IDENTIFIER: tag = [UNIVERSAL 6] primitive;
        length = 3
        { 2 5 29 32 }
    extnValue OCTET STRING: tag = [UNIVERSAL 4] primitive;
        length = 11
        0x3009300706052b24080101
Extension SEQUENCE: tag = [UNIVERSAL 16] constructed;
    length = 31
    extnId OBJECT IDENTIFIER: tag = [UNIVERSAL 6] primitive;
        length = 3
        { 2 5 29 35 }
    extnValue OCTET STRING: tag = [UNIVERSAL 4] primitive;
        length = 24
0x30168014000102030405060708090a0b0c0d0e0f0fedcba98
Extension SEQUENCE: tag = [UNIVERSAL 16] constructed;
length = 57
extnId OBJECT IDENTIFIER: tag = [UNIVERSAL 6] primitive;
length = 8
{ 1 3 6 1 5 5 7 1 3 } 
extnValue OCTET STRING: tag = [UNIVERSAL 4] primitive;
length = 45
0x302b302906082b06010505070b01301d301b81196d756e6963697...

C.3 DER-encoding

This section contains the full, DER-encoded certificate, in hex.

3082030E30820277A00300201020204499602D2300D06092A864886F70D010105
0503004831B03009063550406130244453139303706355040A0C30474DD4420
2D20466F72736368756E67773A656E7472756D2049666F6F7373756469766572
766563686E696204766248301E170D303303353031310500303131033003033504040C06261727A696E30819D300D06092A864886F70D010105003818B003081807282810B8488400D4B608B4E4AD459CA19EC7
17AAF3D1D4EE3ECCA496128A13597D16CC8B85EB37EFCE110C63B01E684E5CF6
32291EAC60FD153C266EAAAC364D4CEA92319F9FBD2D61AD2BFE4EABE17FE67
8341EE52D9A0A8B4EDC07B7ACC7676251404CEEB9994E0CF37BAAE5FB8DEF33B35
FF98BCE77742CE4B12273BD122137FE902015A38E93081E630640603551D09
045D305B3010060B2B60010505070904310412445300F06082B60105050709093301301D46301D6082B60010505070904310412445300F06082B601050507090933013013430
303030350A301706082B60010505070902301BOC58621976737461647430
0E0603551D0F0101FF0040302604030120603551D20040B3009300706052B24
080101301F0603551D203041830160014001020304060708090A0BCD0DE0F
FEFDBA983B30906082B60010505070103042D302B3290602B60010505070B1
301D301B81196D756E69636970616C696794064617273746164746E6530
0D06092A864886F70D01105050003818110048FD14D9AEF691E4321D9AA04CC0
1C12893550CF76F6EBCBE448926B0AE6F904AB89EB58066FF000218AC18D
28C8E2D40FB8C162B725CB0A5778B8574059DEC736223368FC1602A510BC1
EB31E39F3967E6B413D48BC743A0BA19C57FD20F8393E8FEBD8B05CAA5007D
AD36F9D789AEF636A0AC0F93BCB3711B5907

C.4 CA’s public RSA key

This section contains the DER-encoded public RSA key of the CA who signed the example certificate. It is included with the purpose of simplifying verifications of the example certificate.

30818902818100ad1f35964b3674c807b9f8a645d2c8174e514b69a4b46a7382
915abbc44eccede914dae8fccc023abceaa9c53380e641795cb0dda664b872fc10
Authors' Addresses

Stefan Santesson
Microsoft Denmark
Tuborg Boulevard 12
DK-2900 Hellerup
Denmark
EMail: stefans@microsoft.com

Tim Polk
NIST
Building 820, Room 426
Gaithersburg, MD 20899, USA
EMail: wpolk@nist.gov

Magnus Nystrom
RSA Security
Box 10704
S-121 29 Stockholm
Sweden
EMail: magnus@rsasecurity.com
Full Copyright Statement

Copyright (C) The Internet Society (2003). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. In addition, the ASN.1 modules presented in Appendices A and B may be used in whole or in part without inclusion of the copyright notice. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process shall be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns. This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.