Network Working Group
Request for Comments: 3739
Obsoletes: 3039
Category: Standards Track

S. Santesson
Microsoft
M. Nystrom
RSA Security
T. Polk
NIST

March 2004

Internet X.509 Public Key Infrastructure: Qualified Certificates Profile

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Abstract

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

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

The goal of this document is to define a certificate profile that supports the issuance of Qualified Certificates independent of local legal requirements. The profile is however not limited to Qualified Certificates and further profiling may facilitate specific local needs.

Table of Contents

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 [X.509] and [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 affect the scope of this document are discussed in Section 2.

Section 3 defines requirements on certificate information content. This specification provides profiles for two certificate fields: issuer and subject. It also provides profiles for four certificate extensions defined in RFC 3280: subject alternate name, subject directory attributes, certificate policies, and key usage, and it defines two additional extensions: biometric information and qualified certificate statements. The certificate extensions are presented in the 1997 Abstract Syntax Notation One (ASN.1) [X.680], but in conformance with RFC 3280 the 1988 ASN.1 module in Appendix A contains all normative definitions (the 1997 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.

Appendix A contains all relevant ASN.1 structures that are not already defined in RFC 3280. Appendix B contains a note on attributes. Appendix C contains an example certificate.

The appendices sections are followed by the References, Authors Addresses, and the Full Copyright Statement.

1.1. Changes since RFC 3039

This specification obsoletes RFC 3039. This specification differs from RFC 3039 in the following basic areas:

- Some editorial clarifications have been made to introductory sections to clarify that this profile is generally applicable to a broad type of certificates, even if its prime purpose is to facilitate issuance of Qualified Certificates.

- To align with RFC 3280, support for domainComponent and title attributes in subject names are included, and postalAddress is no longer supported.

- To align with actual usage, support for the title attribute in the subject directory attributes extension is no longer supported.

- To better facilitate broad applicability of this profile, some constraints on key usage settings in the key usage extension have been removed.

- A new qc-Statement reflecting this second version of the profile has been defined in Section 3.2.6.1. This profile obsoletes RFC 3039, but the qc-statement reflecting compliance with RFC 3039 is also defined for backwards compatibility.

1.2. Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, [RFC 2119].

2. Requirements and Assumptions

The term "Qualified Certificate" is used by the European Directive on Electronic Signature [EU-ESDIR] to refer to a specific type of certificates, with appliance in European electronic signature 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 a high level of assurance, where the certificate meets some qualification requirements defined by an applicable legal framework, such as the European Directive on Electronic Signature [EU-ESDIR]. The actual mechanisms that decide whether a certificate should or should not be considered a "Qualified Certificate" in regard to any legislation are outside the scope of this standard.

Harmonization in the field of identity certificates issued to natural 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 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 Section 2.3.

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

- The Qualified Certificate contains a name which may be either based on the real name of the subject or a pseudonym.

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 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 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 a set of all objects. The distinguished name MUST be unique for each subject entity certified by the one CA as defined by the issuer name field, for 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 section 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 distinguished name of the issuer SHALL be specified using an appropriate subset of the following attributes:

```
domainComponent;
countryName;
stateOrProvinceName;
organizationName;
localityName; and
serialNumber.
```

The domainComponent attribute is defined in [RFC 2247], all other attributes are defined in [RFC 3280] and [X.520].

Additional attributes MAY be present, but they SHOULD NOT be necessary to identify the issuing organization.

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.

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.
```

The domainComponent attribute is defined in [RFC 2247], all other attributes are defined in [RFC 3280] and [X.520].

Other attributes MAY also be present; however, the use of other attributes MUST NOT be necessary to distinguish one subject name from another subject name. That is, the attributes listed above are sufficient to ensure unique subject names.

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 be used in the subject field if neither the commonName attribute nor the pseudonym attribute is present. In cases where the subject only has a givenName, 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 is also 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 section provides additional details regarding the contents of four certificate extensions defined in RFC 3280: Subject Alternative Name, Subject directory attributes, Certificate policies, and Key usage. This section also defines two additional extensions: biometric information and qualified certificate statements.

3.2.1. Subject Alternative Name

If the subjectAltName extension is present, and it contains a directoryName name, then the directoryName MUST follow the conventions specified in section 3.1.2 of this profile.

3.2.2. Subject Directory Attributes

The subjectDirectoryAttributes extension MAY be present and MAY contain additional attributes associated with the subject, as a 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 date of birth is defined in the GeneralizedTime format and SHOULD specify GMT 12.00.00 (noon) down to the granularity of seconds, in order to prevent accidental change of date due to time zone adjustments. For example, a birth date of September 27, 1959 is encoded as "19590927120000Z". Compliant certificate parsing applications SHOULD ignore any time data and just present the contained date without any time zone adjustments.

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 the certificate was issued. If more than one country of citizenship is specified, each country of citizenship SHOULD be specified through a separate, single-valued countryOfCitizenship attribute. 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 more than one country of residence is specified, each country of residence SHOULD be specified through a separate, single-valued countryOfResidence attribute. Determination of residence is a matter of law and is outside the scope of this document.

3.2.3. Certificate Policies

The certificate policies extension SHALL be present and 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 MUST include all policy information needed for certification path validation. If policy related statements are included in the QCStatements extension (see 3.2.6), then these statements SHOULD also be contained in the identified policies.

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

3.2.4. 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 SHOULD be marked critical.

3.2.5. Biometric Information

This section defines an OPTIONAL 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 a means for the authentication of biometric information. The biometric information that corresponds to the stored hash is not stored in this extension, but the extension MAY include a URI (sourceDataUri) that references a file containing this information.

If included, the URI MUST use the HTTP scheme (http://) [HTTP/1.1] or the HTTPS scheme (https://) [RFC 2818]. Since the fact that identifying data is being checked may itself be sensitive information, those deploying this mechanism may also wish to consider using URIs which cannot be easily tied by outsiders to the identities of those whose information is being retrieved.

Use of the URI option presumes that the data encoding format of the file content is determined through means outside the scope of this specification, such as file naming conventions and metadata inside the file. Use of this URI option does not imply that it is the only way to access this information.

It is RECOMMENDED that biometric information in this extension be 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)} (picture|handwritten-signature,...)
```

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 be calculated over the whole referenced image file.

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 be calculated over the whole referenced image file.

3.2.6. Qualified Certificate Statements

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

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 }

-- NOTE: This extension does not allow to mix critical and
-- non-critical Qualified Certificate Statements. Either all
-- statements must be critical or all statements must be
-- non-critical.

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,...}
```

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.

3.2.6.1. Predefined Statements

The certificate statement (id-qcs-pkixQCSyntax-v1), identifies conformance with requirements defined in the obsoleted RFC 3039 (Version 1). This statement is thus provided for identification of old certificates issued in conformance with RFC 3039. This statement MUST NOT be included in certificates issued in accordance with this profile.

This profile includes a new qualified certificate statement (identified by the OID id-qcs-pkixQCSyntax-v2), identifying conformance with requirements defined in this profile. This Qualified Certificate profile is referred to as version 2, while RFC 3039 is referred to as version 1.

```
qcStatement-1 QC-STATEMENT ::= { SYNTAX SemanticsInformation
    IDENTIFIED BY id-qcs-pkixQCSyntax-v1 }
--  This statement identifies conformance with requirements
--  defined in RFC 3039 (Version 1). This statement may
--  optionally contain additional semantics information as
--  specified below.

qcStatement-2 QC-STATEMENT ::= { SYNTAX SemanticsInformation
    IDENTIFIED BY id-qcs-pkixQCSyntax-v2 }
--  This statement identifies conformance with requirements
--  defined in this Qualified Certificate profile
--  (Version 2). This statement may optionally contain
--  additional semantics information as specified below.

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 SementicsInformation 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 where the statementID component is set to id-qcs-pkix-QCSyntax-v1 or id-qcs-pkix-QCSyntax-v2, then at least one of the semanticsIdentifier or nameRegistrationAuthorities fields must be present, as indicated. Note that the statementInfo component need not be present in a QCStatement value even if the statementID component is set to id-qcs-pkix-QCSyntax-v1 or id-qcs-pkix-QCSyntax-v2.

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 has 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 depending on the context in which the certificate is to be used. Applications validating electronic signatures based on such certificates should determine whether the present key usage combination is appropriate for their use.

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.

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 (Normative)

```
PKIXqualified88 {iso(1) identified-organization(3) dod(6)
    internet(1) security(5) mechanisms(5) pkix(7) id-mod(0)
    id-mod-qualified-cert(31) }

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
   id-qcs  OBJECT IDENTIFIER ::= { id-pkix 11 }

   -- Personal data 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

   -- Certificate 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)}
       (picture|handwritten-signature)
   -- QC Statements Extension
   -- NOTE: This extension does not allow to mix critical and
   -- non-critical Qualified Certificate Statements. Either all
   -- statements must be critical or all statements must be
   -- non-critical.

   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-pkixQCSyntax-v1   OBJECT IDENTIFIER ::= { id-qcs 1 }
   --  This statement identifies conformance with requirements
   --  defined in RFC 3039 (Version 1). This statement may
   --  optionally contain additional semantics information as specified
   --  below.

   id-qcs-pkixQCSyntax-v2   OBJECT IDENTIFIER ::= { id-qcs 2 }
   --  This statement identifies conformance with requirements
   --  defined in this Qualified Certificate profile
   --  (Version 2). This statement may optionally contain
   --  additional semantics information as specified below.

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

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

   END
```

A.2. 1997 ASN.1 Module (Informative)

```
   PKIXqualified97 {iso(1) identified-organization(3) dod(6)
       internet(1) security(5) mechanisms(5) pkix(7) id-mod(0)
       id-mod-qualified-cert-97(35) }

   DEFINITIONS EXPLICIT TAGS ::=

   BEGIN

   -- EXPORTS ALL --
   IMPORTS

   informationFramework, certificateExtensions, selectedAttributeTypes,
       authenticationFramework, upperBounds, id-at
       FROM UsefulDefinitions {joint-iso-itu-t(2) ds(5) module(1)
       usefulDefinitions(0) 3 }

   ub-name
       FROM UpperBounds upperBounds

   GeneralName
       FROM CertificateExtensions certificateExtensions

   ATTRIBUTE, AttributeType
       FROM InformationFramework informationFramework

   DirectoryString
       FROM SelectedAttributeTypes selectedAttributeTypes

   AlgorithmIdentifier, Extension, EXTENSION
       FROM AuthenticationFramework authenticationFramework

   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
   id-qcs  OBJECT IDENTIFIER ::= { id-pkix 11 }

   -- 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 }

   -- Certificate extensions

   id-pe-biometricInfo         OBJECT IDENTIFIER ::= { id-pe 2 }
   id-pe-qcStatements          OBJECT IDENTIFIER ::= { id-pe 3 }
   -- QC statements

   id-qcs-pkixQCSyntax-v1      OBJECT IDENTIFIER ::= { id-qcs 1 }
   id-qcs-pkixQCSyntax-v2      OBJECT IDENTIFIER ::= { id-qcs 2 }

   -- Personal data attributes

   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 }

   -- Certificate 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
   -- NOTE: This extension does not allow to mix critical and
   -- non-critical Qualified Certificate Statements. Either all
   -- statements must be critical or all statements must be
   -- non-critical.

   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 requirements
       --  defined in RFC 3039 (Version 1). This statement
       --  may optionally contain additional semantics information
       --  as specified below.

   qcStatement-2 QC-STATEMENT ::= { SYNTAX SemanticsInformation
       IDENTIFIED BY id-qcs-pkixQCSyntax-v2}
       --  This statement identifies conformance with requirements
       --  defined in this Qualified Certificate profile
       --  (Version 2). This statement may optionally contain
       --  additional semantics information as specified below.

   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 |
       qcStatement-2 , ... -- 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 [RFC 2985].

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 (using the syntax specified in [RFC 2253]): O=GMD - Forschungszentrum Informationstechnik GmbH, C=DE

3. The subject’s distinguished name is (using the syntax specified in [RFC 2253]): GN=Petra+SN=Barzin, O=GMD - Forschungszentrum Informationstechnik GmbH, C=DE

4. The certificate was issued on 1 February, 2004 and will expire on 1 February, 2008

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.

8. The certificate includes a subject directory attributes extension containing the following attributes:

```
date of birth:         October, 14th 1971
place of birth:        Darmstadt
country of citizenship:Germany
gender:                Female
```

9. The certificate includes a qualified statement certificate extension indicating that the naming registration authority's name is "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

```
certSubjDirAttrs AttributesSyntax ::= {
    {
        type id-pda-countryOfCitizenship,
        values {
            PrintableString : "DE"
        }
    },
    {
        type id-pda-gender,
        values {
            PrintableString : "F"
        }
    },
    {
        type id-pda-dateOfBirth,
        values {
            GeneralizedTime : "197110141200Z"
        }
    },
    {
        type id-pda-placeOfBirth,
        values {
            DirectoryString : utf8String : "Darmstadt"
        }
    }
}
```

C.1.1.2. The keyUsage Extension

```
   certKeyUsage KeyUsage ::= {nonRepudiation}
```

C.1.1.3. The certificatePolicies Extension

```
   certCertificatePolicies CertificatePoliciesSyntax ::= {
       {
           policyIdentifier {1 3 36 8 1 1}
       }
   }
```

C.1.1.4. The qcStatements Extension

```
   certQCStatement QCStatements ::= {
       {
           statementId   id-qcs-pkixQCSyntax-v2,
           statementInfo SemanticsInformation : {
               nameRegistrationAuthorities {
                   rfc822Name : "municipality@darmstadt.de"
               }
           }
       }
   }
```

C.1.1.5. The authorityKeyIdentifier Extension

```
   certAKI 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 have been truncated for readability purposes.

```
   certCertInfo CertificateInfo ::= {
     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 :
           }
         }
       },
     validity
     {
       notBefore utcTime : "040201100000Z",
       notAfter utcTime :  "080201100000Z"
     },
     subject rdnSequence :
       {
         {
           {
             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 '30818902818100DCE74CD5...0203010001'H
     },
     extensions
     {
       {
         extnId { 2 5 29 9 },  -- subjectDirectoryAttributes
         extnValue '305B301006082B0601050507090...7374616474'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 '30168014000102030405060708090A0B0C0D0E0FFEDCBA98'H
       },
       {
         extnId { 1 3 6 1 5 5 7 1 3 }, -- qcStatements
         extnValue '302B302906082B06010505070B0...4742E6465 '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.

```
CertificateInfo SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 633
  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
  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 } -- id-at-countryName
          value PrintableString
            "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 } -- id-at-organizationName
          value UTF8String
            "GMD Forschungszentrum Informationstechnik GmbH"
  validity Validity SEQUENCE: tag = [UNIVERSAL 16]
  constructed; length = 30
    notBefore Time CHOICE
      utcTime UTCTime: tag = [UNIVERSAL 23] primitive; length = 13
        040201100000Z
    notAfter Time CHOICE
      utcTime UTCTime: tag = [UNIVERSAL 23] primitive; length = 13
        080201100000Z
  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 } -- id-at-countryName
          value PrintableString
            "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 } -- id-at-organizationName
          value UTF8String
            "GMD Forschungszentrum Informationstechnik GmbH"
      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 } -- id-at-surname
          value UTF8String
            "Barzin"
        AttributeTypeAndValue SEQUENCE: tag = [UNIVERSAL 16]
        constructed; length = 12
          type OBJECT IDENTIFIER: tag = [UNIVERSAL 6]
          primitive; length = 3
            { 2 5 4 42 } -- id-at-givenName
          value UTF8String
            "Petra"
  subjectPublicKeyInfo SubjectPublicKeyInfo SEQUENCE:
  tag = [UNIVERSAL 16] constructed; length = 159
    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 } -- rsaEncryption
      parameters OpenType
        NULL
    subjectPublicKey BIT STRING: tag = [UNIVERSAL 3]
    primitive; length = 141
      0x0030818902818100dce74cd5a1d55aeb01cf5ecc20f3c3fca787...
  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 } -- id-ce-subjectDirectoryAttributes
        extnValue OCTET STRING: tag = [UNIVERSAL 4]
        primitive; length = 93
          0x305b301006082b0601050507090431041302444530 0f06082...
      Extension SEQUENCE: tag = [UNIVERSAL 16]
      constructed; length = 14
        extnId OBJECT IDENTIFIER: tag = [UNIVERSAL 6]
          primitive; length = 3
          { 2 5 29 15 } -- id-ce-keyUsage
        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 } -- id-ce-certificatePolicies
        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 } -- id-ce-authorityKeyIdentifier
        extnValue OCTET STRING: tag = [UNIVERSAL 4]
        primitive; length = 24
          0x301680140001020304050607080900a0b0c0d0e0ffedcba98
      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 } -- id-pe-qcStatements
        extnValue OCTET STRING: tag = [UNIVERSAL 4]
        primitive; length = 45
          0x302b302906082b06010505070b02301d301b81196d756e696...
```

C.3 DER-encoding

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

```
30820310 30820279 A0030201 02020449 9602D230 0D06092A 864886F7 0D010105
05003048 310B3009 06035504 06130244 45313930 37060355 040A0C30 474D4420
2D20466F 72736368 756E6773 7A656E74 72756D20 496E666F 726D6174 696F6E73
74656368 6E696B20 476D6248 301E170D 30343032 30313130 30303030 5A170D30
38303230 31313030 3030305A 3065310B 30090603 55040613 02444531 37303506
0355040A 0C2E474D 4420466F 72736368 756E6773 7A656E74 72756D20 496E666F
```

```
726D6174 696F6E73 74656368 6E696B20 476D6248 311D300C 06035504 2A0C0550
65747261 300D0603 5504040C 06426172 7A696E30 819F300D 06092A86 4886F70D
01010105 0003818D 00308189 02818100 DCE74CD5 A1D55AEB 01CF5ECC 20F3C3FC
A787CFCB 571A21AA 8A20AD5D FF015130 DE724E5E D3F95392 E7BB16C4 A71D0F31
B3A9926A 8F08EA00 FDC3A8F2 BB016DEC A3B9411B A2599A2A 8CB655C6 DFEA25BF
EDDC73B5 94FAA0EF E595C612 A6AE5B8C 7F0CA19C EC4FE7AB 60546768 4BB2387D
5F2F7EBD BC3EF0A6 04F6B404 01176925 02030100 01A381E9 3081E630 64060355
1D09045D 305B3010 06082B06 01050507 09043104 13024445 300F0608 2B060105
05070903 31031301 46301D06 082B0601 05050709 01311118 0F313937 31313031
34313230 3030305A 30170608 2B060105 05070902 310B0C09 4461726D 73746164
74300E06 03551D0F 0101FF04 04030206 40301206 03551D20 040B3009 30070605
2B240801 01301F06 03551D23 04183016 80140001 02030405 06070809 0A0B0C0D
0E0FFEDC BA983039 06082B06 01050507 0103042D 302B3029 06082B06 01050507
0B02301D 301B8119 6D756E69 63697061 6C697479 40646172 6D737461 64742E64
65300D06 092A8648 86F70D01 01050500 03818100 8F8C80BB B2D86B75 F4E21F82
EFE0F20F 6C558890 A6E73118 8359B9C7 8CE71C92 0C66C600 53FBC924 825090F2
95B08826 EAF3FF1F 5917C80B B4836129 CFE5563E 78592B5B B0F9ACB5 2915F0F2
BC36991F 21436520 E9064761 D932D871 F71FFEBD AD648FA7 CF3C1BC0 96F112D4
B882B39F E1A16A90 AE1A80B8 A9676518 B5AA7E97
```

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.

```
   30818902818100c88f4bdb66f713ba3dd7a9069880e888d4321acb53cda7fcdf
   da89b834e25430b956d46a438baa6798035af30db378424e00a8296b012b1b24
   f9cf0b3f83be116cd8a36957dc3f54cbd7c58a10c380b3dfa15bd2922ea8660f
   96e1603d81357c0442ad607c5161d083d919fd5307c1c3fa6dfead0e6410999e
   8b8a8411d525dd0203010001
```

References

Normative References

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

[RFC 2247] Kille, S., Wahl, M., Grimstad, A., Huber R. and S. Sataluri, "Using Domains in LDAP/X.500 Distinguished Names", RFC 2247, January 1998.

[RFC 2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[RFC 2985] Nystrom, M. and B. Kaliski, "PKCS #9: Selected Object Classes and Attribute Types Version 2.0", RFC 2985, November 2000.

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

[X.509] ITU-T Recommendation X.509 (2000) | ISO/IEC 9594-8:2001, Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks

[X.520] ITU-T Recommendation X.520 (2001) | ISO/IEC 9594-6:2001, Information Technology - Open Systems Interconnection - The Directory: Selected Attribute Types, 2001.

[X.680] ITU-T Recommendation X.680 (2002) | ISO/IEC 8824-1:2002), Information Technology - Abstract Syntax Notation One, 2002.

[ISO 3166] ISO 3166-1:1997, Codes for the representation of names of countries, 1997.

[HTTP/1.1] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

Informative References

[X.501] ITU-T recommendation X.501 (2001) | ISO/IEC 9594-2:2001, Information Technology - Open Systems Interconnection - The Directory: Models, 2001.

[EU-ESDIR] Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures, 1999.

[RFC 2253] Wahl, M., Kille, S. and T. Howes, "Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names", RFC 2253, December 1997.

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

Intellectual Property

Acknowledgement

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