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

S. Santesson
AddTrust
W. Polk
NIST
P. Barzin
SECUDE
M. Nystrom
RSA Security

January 2001

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

Copyright (C) The Internet Society (2001). All Rights Reserved.

Abstract

This document forms a certificate profile for Qualified Certificates, based on RFC 2459, for use in the Internet. The term Qualified Certificate is used to describe a certificate with a certain qualified status within applicable governing law. Further, Qualified Certificates are issued exclusively 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.

It is important to note that the profile does not define any legal requirements for 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

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 2459, which defines underlying certificate formats and semantics needed for a full implementation of this standard.

The standard profiles the format for a specific type of certificates named 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 information content in Qualified Certificates. 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 2459 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 Qualified Certificates are assumed to be utilized. Section 5 contains the references.

Appendix A contains all relevant ASN.1 [X.680] structures that are not already defined in RFC 2459. 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 more generally, describing the format for a certificate whose primary purpose is identifying a person with high level of assurance in public non-repudiation services. The actual mechanisms that will 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 Qualified Certificates is essential within several aspects that fall outside the scope of RFC 2459. 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

A Qualified Certificate as defined in this standard is assumed to have the following properties:

- The certificate is 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 certificate indicates a certificate policy consistent with liabilities, practices and procedures undertaken by the CA, as discussed in 2.3

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

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

2.2 Statement of Purpose

For a certificate to serve the purpose of being a Qualified Certificate, this profile assumes that the CA will have to include in the certificate information that explicitly defines this intent.

The function of this information 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 assumed that the 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 a profile for Qualified Certificates. The profile is based on the Internet certificate profile RFC 2459 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 2459.

ASN.1 definitions relevant for this section that are not supplied by RFC 2459 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:

countryName;
commonName;
surname;
givenName;
pseudonym;
serialNumber;
organizationName;
organizationalUnitName;
stateOrProvinceName
localityName and
postalAddress.

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

title;
dateOfBirth;
placeOfBirth;
gender;
countryOfCitizenship; and
countryOfResidence.

Other attributes MAY be included according to local definitions.

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 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 2459.

3.2.3 Key Usage

The key usage extension SHALL be present. If the key usage nonRepudiation bit is asserted then it SHOULD NOT be combined with any other key usage , i.e., if set, the key usage non-repudiation SHOULD be set exclusively.

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)} (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 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 defined statements related to Qualified Certificates.

A typical 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 }
          (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 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 2459. The security considerations section of that document applies to this specification as well.

5 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 2459] Housley, R., Ford, W., Polk, W. and D. Solo, "Internet X.509 Public Key Infrastructure: Certificate and CRL Profile", RFC 2459, January 1999.

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

[X.501] ITU-T Recommendation X.501: Information Technology - Open Systems Interconnection - The Directory: Models, June 1993.

[X.509] ITU-T Recommendation X.509: Information Technology - Open Systems Interconnection - The Directory: Authentication Framework, June 1997.

[X.520] ITU-T Recommendation X.520: Information Technology - Open Systems Interconnection - The Directory: Selected Attribute Types, June 1993.

[X.680] ITU-T Recommendation X.680: Information Technology - Abstract Syntax Notation One, 1997.

[ISO 3166] ISO Standard 3166: Codes for the representation of names of countries, 1993.

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 2459, 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 2459 which are not completely in the 1988 syntax, the same comments as in RFC 2459 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 2459’s 1988 modules.

Appendix A.2 is in the 1993 syntax. However, since the module imports type definitions from modules in RFC 2459 which are not completely in the 1993 syntax, the same comments as in RFC 2459 regarding its use applies here as well; i.e., Appendix A.2 may be parsed by an 1993 ASN.1-parser by removing the UTF8String choice from the definition of DirectoryString in the module PKIX1Explicit93 in RFC 2459. Appendix A.2 may be parsed "as is" by an 1997 ASN.1 parser, however.

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-88(10) }

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-88(2)}

AlgorithmIdentifier, DirectoryString, Attribute, AttributeType,
    id-pkix, id-pe, id-at
    FROM PKIX1Explicit88 {iso(1) identified-organization(3) dod(6)
    internet(1) security(5) mechanisms(5) pkix(7) id-mod(0)
    id-pkix1-explicit-88(1)};

-- 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 }

-- Attributes

id-at-serialNumber           AttributeType ::= { id-at 5 }
SerialNumber ::=             PrintableString (SIZE(1..64))

id-at-postalAddress          AttributeType ::= { id-at 16 }
PostalAddress ::=            SEQUENCE SIZE (1..6) OF DirectoryString

id-at-pseudonym              AttributeType ::= { id-at 65 }
Pseudonym ::=                DirectoryString

domainComponent              AttributeType ::=
                             { 0 9 2342 19200300 100 1 25 }
DomainComponent ::=          IA5String

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)}
    (picture|handwritten-signature)

-- 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-pkixQCSyntax-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 {
    semanticsIndentifier        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

```
PKIXqualified93 {iso(1) identified-organization(3) dod(6)
    internet(1) security(5) mechanisms(5) pkix(7) id-mod(0)
    id-mod-qualified-cert-93(11) }
```

```
DEFINITIONS EXPLICIT TAGS ::=

BEGIN

-- EXPORTS ALL --

IMPORTS

authorityKeyIdentifier, subjectKeyIdentifier, keyUsage,
    extendedKeyUsage, privateKeyUsagePeriod, certificatePolicies,
    policyMappings, subjectAltName, issuerAltName, basicConstraints,
    nameConstraints, policyConstraints, cRLDistributionPoints,
    subjectDirectoryAttributes, authorityInfoAccess, GeneralName,
    OTHER-NAME
    FROM PKIX1Implicit93 {iso(1) identified-organization(3) dod(6)
    internet(1) security(5) mechanisms(5) pkix(7) id-mod(0)
    id-pkix1-implicit-93(4)}

id-pkix, AlgorithmIdentifier, ATTRIBUTE, Extension, EXTENSION,
    DirectoryString{}, ub-name, id-pe, id-at, id-at-commonName,
    id-at-surname, id-at-countryName, id-at-localityName,
    id-at-stateOrProvinceName, id-at-organizationName,
    id-at-organizationalUnitName, id-at-givenName, id-at-dnQualifier,
    pkcs9email, title, organizationName, organizationalUnitName,
    stateOrProvinceName, localityName, countryName,
    generationQualifier, dnQualifier, initials, givenName, surname,
    commonName, name
    FROM PKIX1Explicit93 {iso(1) identified-organization(3) dod(6)
    internet(1) security(5) mechanisms(5) pkix(7) id-mod(0)
    id-pkix1-explicit-93(3)};

-- Object Identifiers

-- Externally defined OIDs
id-at-serialNumber  OBJECT IDENTIFIER ::= { id-at 5}
id-at-postalAddress OBJECT IDENTIFIER ::= { id-at 16 }
id-at-pseudonym     OBJECT IDENTIFIER ::= { id-at 65 }
id-domainComponent  OBJECT IDENTIFIER ::= { 0 9 2342 19200300 100 1 25 }

-- 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 }

-- Private extensions
```

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

-- Personal data attributes
id-pda-dateOfBirth          OBJECT IDENTIFIER ::= { id-pda 1 }
id-pda-placeOfBirth         OBJECT IDENTIFIER ::= { id-pda 2 }
id-pda-gender               OBJECT IDENTIFIER ::= { id-pda 3 }
id-pda-countryOfCitizenship OBJECT IDENTIFIER ::= { id-pda 4 }
id-pda-countryOfResidence   OBJECT IDENTIFIER ::= { 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. Note that this set is an
-- extension of the ExtensionSet defined in RFC 2459.
ExtensionSet EXTENSION ::= {
    authorityKeyIdentifier |
    subjectKeyIdentifier |
    keyUsage |
    extendedKeyUsage |
    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, and that
-- this set is an extension of the SupportedAttributes set in RFC 2459.

SupportedAttributes ATTRIBUTE ::= {
    countryName | commonName | surname | givenName | pseudonym |
    serialNumber | organizationName | organizationalUnitName |
    stateOrProvinceName | localityName | postalAddress |
    pkcs9email | 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 ::= {
    title | dateOfBirth | placeOfBirth | gender | countryOfCitizenship |
    countryOfResidence, ... }

-- Attributes

-- serialNumber from X.520
serialNumber ATTRIBUTE ::= {
    WITH SYNTAX PrintableString (SIZE(1..64))
    ID          id-at-serialNumber }

-- postalAddress from X.520
postalAddress ATTRIBUTE ::= {
    WITH SYNTAX SEQUENCE SIZE (1..6) OF DirectoryString { 30 }
    ID          id-at-postalAddress }

-- pseudonym from (forthcoming) X.520)
pseudonym ATTRIBUTE ::= {
    WITH SYNTAX DirectoryString { ub-name }
    ID          id-at-pseudonym }

-- 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. In the interest of conformity, they have been defined here using the ASN.1 ATTRIBUTE definition from RFC 2459, which is sufficient for the purposes of this document, but greatly simplified in comparison with ISO/ITU’s definition. 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 2459, 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 ::= {
    {
        type id-pda-countryOfCitizenship,
        values {
            PrintableString : "DE"
        }
    },
    {
        type id-pda-gender,
        values {
            PrintableString : "F"
        }
    },
    {
        type id-pda-dateOfBirth,
        values {
            GeneralizedTime : "197110140000Z"
        }
    },
    {
        type id-pda-placeOfBirth,
        values {
            DirectoryString : utf8String : "Darmstadt"
        }
    }
}
```

C.1.1.2 The keyUsage extension

```
   petrasKeyUsage KeyUsage ::= {nonRepudiation}
```

C.1.1.3 The certificatePolicies extension

```
   petrasCertificatePolicies CertificatePoliciesSyntax ::= {
       {
           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"
        }
      },
      {
        {
          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 ’30168014000102030405060708090A0B0C0D0E0FFEDCBA98’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 has 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
               0x474d44202d20466f72736368756e67737a656e7472756d2049...
     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
               0x474d4420466f72736368756e67737a656e7472756d20496e66...
         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
             0x30168014000102030405060708090a0b0c0d0e0ffedcba98
```

```
        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.

```
3082030E30820277A0030201020204499602D2300D06092A864886F70D010105
05003048310B3009060355040613024445313930370603550040A0C30474D4420
2D20466F72736368756E67737A656E7472756D20496E666F726D6174696F6E73
746563686E696B20476D6248301E170D3030303530313130303030305A170D30
30313130313130303030305A3065310B300906035504061302444531373035 06
0355040A0C2E474D4420466F72736368756E67737A656E7472756D20496E666F
726D6174696F6E73746563686E696B20476D6248311D300C060355042A0C0550
65747261300D06035504040C064261727A696E30819D300D06092A864886F70D
010101050003818B0030818702818100B8488400D4B6088BE48EAD459CA19EC7
17AAF3D1D4EE3ECCA496128A13597D16CC8B85EB37EFCE110C63B01E684E5CF6
32291EAC60FD153C266EAAC36AD4CEA92319F9BFDD261AD2BFE41EAB4E17FE67
8341EE52D9A0A8B4DEC07B7ACC76762514045CEE9994E0CF37BAE05F8DE33B35
FF98BCE77742CE4B12273BD122137FE9020105A381E93081E630640603551D09
045D305B301006082B06010505070904310413024445300F06082B0601050507
090331031301463 01D06082B060105050709013111180F31393731313031343 0
30303030305A301706082B06010505070902310B0C094461726D737461647430
0E0603551D0F0101FF040403020640301206 03551D20040B3009300706052B24
080101301F0603551D230418301680140001020304050607 08090A0B0C0D0E0F
FEDCBA98303906082B0601050507010304 2D302B302906082B06010505070B01
301D301B81196D756E69636970616C697479406461726D73746164742E646530
0D06092A864886F70D01010505000381810048FD14D9AFE961E4321D9AA40CC0
1C12893550CF76FBECBDE448926B0AE6F904AB89E7B5F808666FB007218AC18D
28CE1E2D40FBF8C16B275CBA0547D7885B74059DEC736223368FC1602A510BC1
EB31E39F3967BE6B413D48BC743A0AB19C57FD20F3B393E8FEBD8B05CAA5007D
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
915abbc44eccede914dae8fcc023abcea9c53380e641795cb0dda664b872fc10
9f9bbb852bf42d994f634c681608e388dce240b558513e5b60027bd1a07cef9c
9b6db37c7e1f1abd238eed96e4b669056b260f55e83f14e6027127c9deb3ad18
afcd3f8a5f5bf50203010001
```

Authors’ Addresses

Stefan Santesson
AddTrust AB
P.O. Box 465
S-201 24 Malmo
Sweden

EMail: stefan@addtrust.com

Tim Polk
NIST
Building 820, Room 426
Gaithersburg, MD 20899, USA

EMail: wpolk@nist.gov

Petra Barzin
SECUDE - Sicherheitstechnologie Informationssysteme GmbH
Landwehrstrasse 50a
D-64293 Darmstadt
Germany

EMail: barzin@secude.com

Magnus Nystrom
RSA Security AB
Box 10704
S-121 29 Stockholm
Sweden

EMail: magnus@rsasecurity.com

Full Copyright Statement

Copyright (C) The Internet Society (2001). 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. 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 must 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.

Acknowledgement

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