PKI applications (C2)

Standards and protocols

Pascal Steichen (MSSI-uni.lu) - 12/01/2008

1. "Implementing" the directive (1999/93/EC)

- European Electronic Signature Standardization Initiative (EESSI)

  Industry and European standardization bodies, within the frame of the ICTSB, have been requested by the European Commission to analyze in a coherent manner, the needs for standardization activities in support of essential legal requirements as stated in the Directive in relation to electronic signatures products and services to be made available to the market. The assessment of available standards and current initiatives at global and regional level, both in formal standardization bodies and industry consortia, did identify gaps and the need for any additional standardization initiatives in all relevant forms, such as standards, specifications, agreements, workshops or any other form of consensus building. On the basis of this analysis, a work programme has been defined and implemented.

  It is for Industry and European Standardization bodies to set up the implementation framework, compliant with the minimal legal framework stated by the Directive, which answers business needs and brings the full advantage of the legal recognition of the electronic signature in support of the development of an open electronic commerce environment.

  Although several standardization initiatives in the area of authentication had already been launched by standards bodies and industry fora at national, regional and international levels, it was ascertained that they lacked the necessary consistency and coherence for validity and cross-recognition.

  To remedy this, the European ICT Standards Board, with the support of the European Commission, has launched an initiative bringing together industry and public authorities, experts and other market players: the European Electronic Signature Standardization Initiative (EESSI).

  EESSI seeks to identify under a common approach the needs for standardization activities in support of the Directive’s requirements, and to monitor the implementation of the work programme.

  EESSI has been anxious to ensure that three main principles were adhered to:

  - effective involvement of all parties concerned with the broad subject area of electronic signatures;
  - openness and transparency of the mechanisms used and of the initiatives taken;
  - encouragement of global, internationally-accepted solutions whilst avoiding duplication of work.

- CEN (Comité Européen de Normalisation) / ISSS (Information Society Standardization System) E-sign workshop
- ETSI (European Telecommunications Standards Institute) TC (Technical Committee) / ESI (Electronic Signatures and Infrastructures)

- European Commission

  COMMISSION DECISION of 14 July 2003 on the publication of reference numbers of generally recognised standards for electronic signature products in accordance with Directive 1999/93/EC of the European Parliament and of the Council:

  List of generally recognised standards for electronic signature products that Member States shall presume are in compliance with the requirements laid down in

  1. Annex II of Directive 1999/93/EC (Requirements for certification-service-providers issuing qualified certificates):
     - CWA 14167-1 (March 2003): security requirements for trustworthy systems managing certificates for electronic signatures — Part 1: System Security Requirements
     - CWA 14167-2 (March 2002): security requirements for trustworthy systems managing certificates for electronic signatures — Part 2: cryptographic module for CSP signing operations — Protection Profile (MCSO-PP)
  2. Annex III of Directive 1999/93/EC (Requirements for secure signature-creation devices):
     - CWA 14169 (March 2002): secure signature-creation devices

1.1. CEN/ISSS E-sign workshop

Successful implementation of European Directive 1999/93/EC on a Community framework for electronic signatures [Dir.1999/93/EC] requires standards for services, processes, systems and products related to electronic signatures as well as guidance for conformity assessment of such services, processes, systems and products.

In 1999 the European ICT Standards Board, with the support of the European Commission, undertook an initiative bringing together industry and public authorities, experts and other market players, to create the European Electronic Signature Standardisation Initiative (EESSI).

Within this framework the CEN/ISSS Workshop on Electronic Signatures and the ETSI TC on Electronic Signatures and Infrastructures (ETSI/ESI) were entrusted with the execution of a work programme in support of the implementation of the Directive1999/93/EC and the development of a European electronic signature infrastructure.

- CWA 14167 (Multipart) - Security Requirements for Trustworthy Systems Managing Certificates for Electronic Signatures
  - CWA 14167-1 - Security Requirements for Trustworthy Systems Managing Certificates for Electronic Signatures - Part 1: System Security Requirements
  - CWA 14167-2 - Security Requirements for Trustworthy Systems Managing Certificates for Electronic Signatures - Part 2: Cryptographic Module for CSP signing operations with backup - Protection profile (CMCSOB-PP)
  - CWA 14167-3 - Security Requirements for Trustworthy Systems Managing Certificates for Electronic Signatures - Part 3: Cryptographic module for CSP key generation services - Protection profile (CMCKG-PP)
  - CWA 14167-4 - Security Requirements for Trustworthy Systems Managing Certificates for Electronic Signatures - Part 4: Cryptographic module for CSP signing operations - Protection profile - CMCSO PP
- CWA 14169 - Secure Signature-creation devices "EAL 4+"
- CWA 14170 - Security requirements for signature creation applications
- CWA 14171 - General guidelines for electronic signature verification
- CWA 14172 (Multipart) - EESSI Conformity Assessment Guidance

    - CWA 14172-1 - EESSI Conformity Assessment Guidance - Part 1: General introduction
    - CWA 14172-2 - EESSI Conformity Assessment Guidance - Part 2: Certification Authority services and processes
    - CWA 14172-3 - EESSI Conformity Assessment Guidance - Part 3: Trustworthy systems managing certificates for electronic signatures
    - CWA 14172-4 - EESSI Conformity Assessment Guidance - Part 4: Signature-creation applications and general guidelines for electronic signature verification
    - CWA 14172-5 - EESSI Conformity Assessment Guidance - Part 5: Secure signature-creation devices
    - CWA 14172-6 - EESSI Conformity Assessment Guidance - Part 6: Signature-creation device supporting signatures other than qualified
    - CWA 14172-7 - EESSI Conformity Assessment Guidance - Part 7: Cryptographic modules used by Certification Service Providers for signing operations and key generation services
    - CWA 14172-8 - EESSI Conformity Assessment Guidance - Part 8: Time-stamping Authority services and processes

- CWA 14355 - Guidelines for the implementation of Secure Signature-Creation Devices
- CWA 14365 (Multipart) - Guide on the Use of Electronic Signatures

    - CWA 14365-1 - Guide on the Use of Electronic Signatures - Part 1: Legal and Technical Aspects
    - CWA 14365-2 - Guide on the Use of Electronic Signatures - Part 2: Protection Profile for Software Signature Creation Devices

- CWA 14890 (Multipart) - Application Interface for smart cards used as Secure Signature Creation Devices

    - CWA 14890-1 - Application Interface for smart cards used as Secure Signature Creation Devices - Part 1: Basic requirements
    - CWA 14890-2 - Application Interface for smart cards used as Secure Signature Creation Devices - Part 2: Additional Services

1.2. ETSI TC/ESI

- ETSI TS 101 733 V1.7.3 - CMS Advanced Electronic Signatures (CAdES)
- ETSI TS 101 862 V1.3.3 - Qualified Certificate profile
- ETSI TS 101 456 V1.4.3 - Policy requirements for certification authorities issuing qualified certificates
- ETSI TS 102 042 V1.3.4 - Policy requirements for certification authorities issuing public key certificates
- ETSI TS 101 861 V1.3.1 - Time stamping profile
- ETSI TS 101 903 V1.3.2 - XML Advanced Electronic Signatures (XAdES)
- ETSI TR 102 605 V1.1.1 - Registered E-Mail
- ...

- ETSI TR 102 044 V1.1.1 - Requirements for role and attribute certificates
- ETSI TS 102 023 V1.2.1 - Policy requirements for time-stamping authorities
- ETSI SR 002 176 V1.1.1 - Algorithms and Parameters for Secure Electronic Signatures
- ETSI TR 102 153 V1.1.1 - Pre-study on certificate profiles
- ETSI TR 102 045 V1.1.1 - Signature policy for extended business model
- ETSI TS 102 158 V1.1.1 - Policy requirements for Certification Service Providers issuing attribute certificates usable with Qualified certificates
- ETSI TR 102 272 V1.1.1 - ASN.1 format for signature policies
- ETSI TS 102 280 V1.1.1 - X.509 V.3 Certificate Profile for Certificates Issued to Natural Persons
- ETSI TR 102 046 V1.2.1 - Maintenance report
- ETSI TR 102 317 V1.1.1 - Process and tool for maintenance of ETSI deliverables
- ETSI TR 102 040 V1.3.1 - International Harmonization of Policy Requirements for CAs issuing Certificates
- ETSI TR 102 047 V1.2.1 - International Harmonization of Electronic Signature Formats
- ETSI TS 102 176-1 V2.0.0 - Algorithms and Parameters for Secure Electronic Signatures; Part 1: Hash functions and asymmetric algorithms
- ETSI TS 102 176-2 V1.2.1 - Algorithms and Parameters for Secure Electronic Signatures; Part 2: Secure channel protocols and algorithms for signature creation devices
- ETSI TR 102 438 V1.1.1 - Application of Electronic Signature Standards in Europe
- ETSI TS 102 231 V2.1.1 - Provision of harmonized Trust-service status information
- ETSI TR 102 458 V1.1.1 - Mapping Comparison Matrix between the US Federal Bridge CA Certificate Policy and the European Qualified Certificate Policy (TS 101 456)
- ETSI TR 102 437 V1.1.1 - Guidance on TS 101 456 (Policy Requirements for certification authorities issuing qualified certificates)
- ETSI TS 102 734 V1.1.1 - Profiles of CMS Advanced Electronic Signatures based on TS 101 733 (CAdES)
- ETSI TS 102 904 V1.1.1 - Profiles of XML Advanced Electronic Signatures based on TS 101 903 (XAdES)
- ETSI TR 102 572 V1.1.1 - Best Practices for handling electronic signatures and signed data for digital accounting
- ETSI TS 102 573 V1.1.1 - Policy requirements for trust service providers signing and/or storing data for digital accounting

1.3. Examples

- CWA 14167-1 - Security Requirements for Trustworthy Systems Managing Certificates for Electronic Signatures - Part 1: System Security Requirements
- ETSI TS 101 456 - Policy requirements for certification authorities issuing qualified certificates
- ETSI TS 102 042 - Policy requirements for certification authorities issuing public key certificates
- CWA 14355 - Guidelines for the implementation of Secure Signature-Creation Devices

1.3.1. CWA 14167-1

Security Requirements for Trustworthy Systems Managing Certificates for Electronic Signatures - Part 1: System Security Requirements

Annex II of [Dir.1999/93/EC] provides the requirements for a Certificate Service Provider (CSP) issuing Qualified Certificates (QCs). This CWA principally concentrates on providing all the technical security requirements for the Trustworthy Systems (TWSs) a CSP needs to deploy.

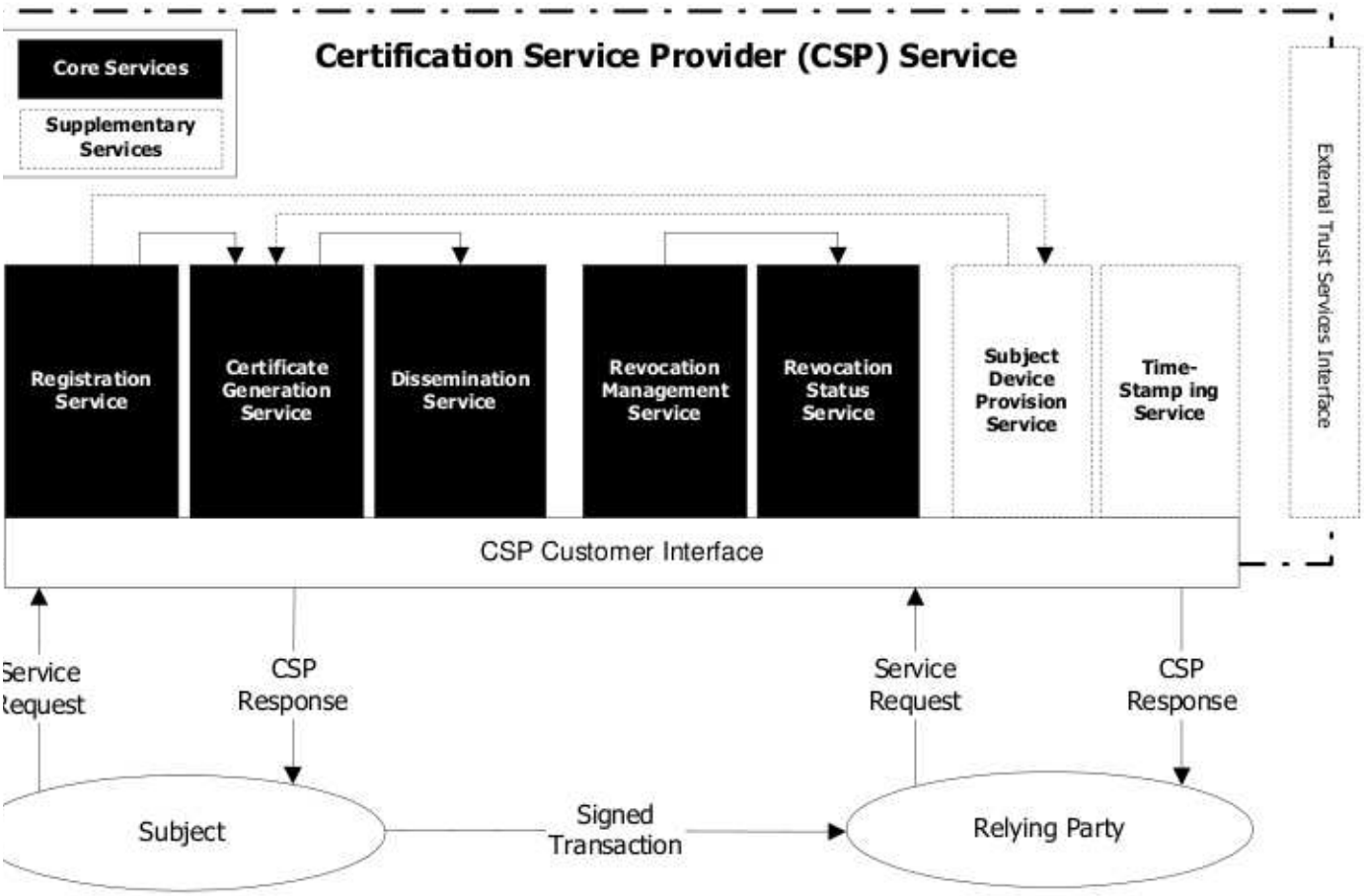
According to Annex II of Dir.1999/93/EC, CSPs must:

> "use trustworthy systems and products which are protected against modification and which must ensure the technical and cryptographic security of the processes supported by them".

Non-Qualified Certificates (NQCs) used for Electronic Signatures may require less security provisions when compared to QCs and therefore this CWA caters for both and indicates the areas where differentiation is required.

- Overall architecture:

  A CSP's logical architecture is shown in the figure below, and can be seen to facilitate the production and use of a signed transaction from the Subject to a Relying Party. This figure illustrates both mandatory and optional services along with the CSP's interfaces to its Subjects, Relying Parties and to any external Trust Services.



- Security levels:

  The certificates produced by a CSP fall into the following categories:

1. Non-Qualified Certificates (NQCs):

    - Used for Electronic Signatures, meeting [Dir.1999/93/EC], Article 5.2
    - Used for Electronic Signatures in internal tasks of the TWS (Trustworthy System)
2. Qualified Certificates (QCs):
    - Used for Advanced Electronic Signatures (AES) which are created by a Secure-Signature-Creation Device (SSCD), meeting Dir.1999/93/EC, Article 5.1
    - Used for Advanced Electronic Signatures (AES) which are created by a Signature-Creation Device (SCDev)

- Security Requirements
    - General Security Requirements
        - Management
        - Systems & Operations
        - Identification & Authentication
        - System Access Control
        - Key Management
        - Accounting & Auditing
        - Archiving
        - Backup & Recovery

- Security Requirements (cont'd)
    - Core Services Security Requirements
        - General
        - Registration Service
        - Certificate Generation Service
        - Dissemination Service
        - Certificate Revocation Management Service
        - Certificate Revocation Status Service
    - Supplementary Services Security Requirements
        - Time-Stamping Service
        - Subject Device Provision Service.

1.3.2. ETSI TS 101 456

Policy requirements for certification authorities issuing qualified certificates

A certificate policy is a "named set of rules that indicates the applicability of a certificate to a particular community and/or class of application with common security requirements".

The policy requirements are defined in the present document in terms of certificate policies. These certificate policies are for qualified certificates, as defined the Directive [1], and hence are called qualified certificate policies. Certificates issued in accordance with the present document include a certificate policy identifier which can be used by relying parties in determining the certificates suitability and trustworthiness for a particular application. The present document specifies two qualified certificate policies:

1. a qualified certificate policy for qualified certificates issued to the public, requiring use of secure signature-creation devices;

   NOTE 1: The exact meaning of public is left to interpretation within the context on national legislation. A CA may be considered to be issuing qualified certificates to the public if the certificates are not restricted to uses governed by voluntary agreements under private law among participants.

2. a qualified certificate policy for qualified certificates issued to the public.

1.3.3. ETSI TS 102 042

Policy requirements for certification authorities issuing public key certificates

The present document includes options for supporting the same level of quality by certification authorities issuing qualified certificates (as required article 5.1 of the Electronic Signature Directive 1999/93/EC [16]) but "normalized" for wider applicability and for ease of alignment with other similar specifications and standards from other sources and institutions. Through such harmonization the quality level set by the Electronic Signature Directive can become embodied in more widely recognized and accepted specifications.

The policy requirements are defined in terms of three reference certificates policies and a framework from which CAs can produce a certificate policy targeted at a particular service.

1. An **extended Normalized Certificate Policy (NCP+)** which offers
    - the same quality as that offered by the Qualified Certificate Policy (QCP) as defined in TS 101 456
    - but without the legal constraints implied by the Electronic Signature Directive (1999/93/EC)
    - and, instead of requiring the use of a Secure Signature Creation, requires the use of a secure user device.

   NOTE 2: The certificate policy NCP+ is particularly suited to the support of Advanced Electronic Signatures, as defined by the Electronic Signature Directive (1999/93/EC), for human beings as well as legal entities since the use of a secure user device provides confidence that the signing key remains under the sole control of the signatory.

2. A **Normalized Certificate Policy (NCP)** which offers
    - the same quality as that offered by the Qualified Certificate Policy (QCP) as defined in TS 101 456

- but without the legal constraints implied by the Electronic Signature Directive (1999/93/EC)
- and without requiring the use of a Secure Signature Creation Device.

NOTE 1: The certificate policy NCP is particularly suited to the support of Advanced Electronic Signatures, as defined by the Electronic Signature Directive (1999/93/EC), for legal entities if they use physical means to provide reasonable confidence that the signing key remains under their sole control.

3. A **Lightweight Certificate Policy (LCP)** which incorporates less demanding policy requirements (e.g. physical presence).

1.3.4. CWA 14355

Guidelines for the implementation of Secure Signature-Creation Devices

- SSCD Types:
  1. SSCD Type 1: SCD/SVD-pair generation
  2. SSCD Type 2: Signature-creation
  3. SSCD Type 3: Both SCD/SVD-pair generation and signature-creation

2. X.509 Public Key Certificate - RFC 3280

A 'certificate' is a digitally signed, structured message that asserts an association between specific data and a particular public key. An 'identity certificate' is then a particular class of certificate that associates a particular identifier with a particular public key.

The dominant standard at present is the family of CCITT X.500 standards, in particular X.509 (X.509 (1988, 1997) 'The Directory - Authentication Framework', Volume VIII of CCITT Blue Book, pages 48-81, CCITT/ITU, 1988, 1997). The current version of X.509 is number 3, usually referred to as X.509v3, which was finalised in 1997. A set of standards, dubbed PKIX (Internet X.509 Public Key Infrastructure), enables use of X.509.

> Carl M. Ellison describes the history this way: "the X.500 proposal was published in the late 1980s. It was to be a global directory of named entities. To tie a public key to some node or sub-directory of that structure, the X.509 certificate was defined. The Subject of such a certificate was a path name indicating a node in the X.500 database - a so-called 'Distinguished Name'. The X.500 dream has effectively died but the X.509 certificate has lived on. The distinguished name took the place of a person's name and the certificate was called an 'identity certificate', assumed to bind an identity to a public key ...". In short, X.509 was the hammer that came to hand when the nail was discovered.

Structure of an X.509 certificate (RFC 3280):

```
Certificate ::= SEQUENCE {
     TBSCertificate ::= SEQUENCE {
             version,                (default v1)
             serialNumber,
             signature,              (algorithm)
             issuer,                 (DN, e.g. c=LU,o=Uni)
             validity,
             subject,                (DN, c=LU,o=Uni,cn=Pascal Steichen)
             subjectPublicKeyInfo,   (public key and algorithms used)
             issuerUniqueID,         (optional)
             subjectUniqueID,        (optional)
             extensions
     }
     signatureAlgorithm,             (algorithm)
     signatureValue                  (CA's signature)
     }
```

2.1. Certificate Extensions

```
Extension ::= SEQUENCE {
      extnID,        (identifier)
      critical,      (criticality - boolean)
      extnValue      (valeur)
}
```

The extensions defined for X.509 v3 certificates provide methods for associating additional attributes with users or public keys and for managing a certification hierarchy. The X.509 v3 certificate format also allows communities to define private extensions to carry information unique to those communities. Each extension in a certificate is designated as either critical or non-critical. A certificate using system MUST reject the certificate if it encounters a critical extension it does not recognize; however, a non-critical extension MAY be ignored if it is not recognized. The following sections present recommended extensions used within Internet certificates and standard locations for information. Communities may elect to use additional extensions; however, caution ought to be exercised in adopting any critical extensions in certificates which might prevent use in a general context.

The **standard extensions** can be grouped as follows:

1. informations on keys
2. informations on the certificat usages

3. user and CA attributes
4. co-certification constraints

2.1.1. informations on keys

- Authority Key Identifier

  The authority key identifier extension provides a means of identifying the public key corresponding to the private key used to sign a certificate. This extension is used where an issuer has multiple signing keys (either due to multiple concurrent key pairs or due to changeover). The identification MAY be based on either the key identifier (the subject key identifier in the issuer's certificate) or on the issuer name and serial number.

```
AuthorityKeyIdentifier ::= SEQUENCE {
    keyIdentifier             [0] KeyIdentifier           OPTIONAL,
    authorityCertIssuer       [1] GeneralNames            OPTIONAL,
    authorityCertSerialNumber [2] CertificateSerialNumber OPTIONAL }
```

- Subject Key Identifier

  The subject key identifier extension provides a means of identifying certificates that contain a particular public key. To facilitate certification path construction, this extension MUST appear in all conforming CA certificates, that is, all certificates including the basic constraints extension (see below) where the value of cA is TRUE. The value of the subject key identifier MUST be the value placed in the key identifier field of the Authority Key Identifier extension (see above) of certificates issued by the subject of this certificate.

  This extension MUST NOT be marked critical.

- Key Usage

  The key usage extension defines the purpose (e.g., encipherment, signature, certificate signing) of the key contained in the certificate. The usage restriction might be employed when a key that could be used for more than one operation is to be restricted. For example, when an RSA key should be used only to verify signatures on objects other than public key certificates and CRLs, the digitalSignature and/or nonRepudiation bits would be asserted. Likewise, when an RSA key should be used only for key management, the keyEncipherment bit would be asserted.

  This extension MUST appear in certificates that contain public keys that are used to validate digital signatures on other public key certificates or CRLs. When this extension appears, it SHOULD be marked critical.

```
KeyUsage ::= BIT STRING {
       digitalSignature        (0),
       nonRepudiation          (1),
       keyEncipherment         (2),
       dataEncipherment        (3),
       keyAgreement            (4),
       keyCertSign             (5),
       cRLSign                 (6),
       encipherOnly            (7),
       decipherOnly            (8)
       }
```

  Bits in the KeyUsage type are used as follows:
  - The digitalSignature bit is asserted when the subject public key is used with a digital signature mechanism to support security services other than certificate signing (bit 5), or CRL signing (bit 6). Digital signature mechanisms are often used for entity authentication and data origin authentication with integrity.
  - The nonRepudiation bit is asserted when the subject public key is used to verify digital signatures used to provide a non-repudiation service which protects against the signing entity falsely denying some action, excluding certificate or CRL signing. In the case of later conflict, a reliable third party may determine the authenticity of the signed data. Further distinctions between the digitalSignature and nonRepudiation bits may be provided in specific certificate policies.
  - The keyEncipherment bit is asserted when the subject public key is used for key transport. For example, when an RSA key is to be used for key management, then this bit is set.
  - The dataEncipherment bit is asserted when the subject public key is used for enciphering user data, other than cryptographic keys.
  - The keyAgreement bit is asserted when the subject public key is used for key agreement. For example, when a Diffie-Hellman key is to be used for key management, then this bit is set.
  - The keyCertSign bit is asserted when the subject public key is used for verifying a signature on public key certificates. If the keyCertSign bit is asserted, then the cA bit in the basic constraints extension (see below) MUST also be asserted.
  - The cRLSign bit is asserted when the subject public key is used for verifying a signature on certificate revocation list (e.g., a CRL, delta CRL, or an ARL). This bit MUST be asserted in certificates that are used to verify signatures on CRLs.
  - The meaning of the encipherOnly bit is undefined in the absence of the keyAgreement bit. When the encipherOnly bit is asserted and the keyAgreement bit is also set, the subject public key may be used only for enciphering data while performing key agreement.
  - The meaning of the decipherOnly bit is undefined in the absence of the keyAgreement bit. When the decipherOnly bit is asserted and the keyAgreement bit is also set, the subject public key may be used only for deciphering data while performing key agreement.

- Extended Key Usage

  This extension indicates one or more purposes for which the certified public key may be used, in addition to or in place of the basic purposes indicated in the key usage extension. In general, this extension will appear only in end entity certificates. This extension is defined as follows:

```
ExtKeyUsageSyntax ::= SEQUENCE SIZE (1..MAX) OF KeyPurposeId

KeyPurposeId ::= OBJECT IDENTIFIER

id-kp OBJECT IDENTIFIER ::= < id-pkix 3 >

id-kp-serverAuth             OBJECT IDENTIFIER ::= < id-kp 1 >
  -- TLS WWW server authentication
  -- Key usage bits that may be consistent: digitalSignature,
  -- keyEncipherment or keyAgreement

id-kp-clientAuth             OBJECT IDENTIFIER ::= < id-kp 2 >
  -- TLS WWW client authentication
  -- Key usage bits that may be consistent: digitalSignature
  -- and/or keyAgreement

id-kp-codeSigning             OBJECT IDENTIFIER ::= < id-kp 3 >
  -- Signing of downloadable executable code
  -- Key usage bits that may be consistent: digitalSignature

id-kp-emailProtection         OBJECT IDENTIFIER ::= < id-kp 4 >
  -- E-mail protection
  -- Key usage bits that may be consistent: digitalSignature,
  -- nonRepudiation, and/or (keyEncipherment or keyAgreement)

id-kp-timeStamping            OBJECT IDENTIFIER ::= < id-kp 8 >
  -- Binding the hash of an object to a time
  -- Key usage bits that may be consistent: digitalSignature
  -- and/or nonRepudiation

id-kp-OCSPSigning             OBJECT IDENTIFIER ::= < id-kp 9 >
  -- Signing OCSP responses
  -- Key usage bits that may be consistent: digitalSignature
  -- and/or nonRepudiation
```

- CRL Distribution Points

  The CRL distribution points extension identifies how CRL information is obtained. The extension SHOULD be non-critical, but this profile RECOMMENDS support for this extension by CAs and applications.

```
    CRLDistributionPoints ::= SEQUENCE SIZE (1..MAX) OF DistributionPoint

    DistributionPoint ::= SEQUENCE {
        distributionPoint       [0]     DistributionPointName OPTIONAL,
        reasons                 [1]     ReasonFlags OPTIONAL,
        cRLIssuer               [2]     GeneralNames OPTIONAL }

    DistributionPointName ::= CHOICE {
        fullName                [0]     GeneralNames,
        nameRelativeToCRLIssuer [1]     RelativeDistinguishedName }

    ReasonFlags ::= BIT STRING {
         unused                  (0),
         keyCompromise           (1),
         cACompromise            (2),
         affiliationChanged      (3),
         superseded              (4),
         cessationOfOperation    (5),
         certificateHold         (6),
         privilegeWithdrawn      (7),
         aACompromise            (8) }
```

2.1.2. informations on the certificat usages

- Certificate Policies

  The certificate policies extension contains a sequence of one or more policy information terms, each of which consists of an object identifier (OID) and optional qualifiers. Optional qualifiers, which MAY be present, are not expected to change the definition of the policy.

- Policy Mappings

  This extension is used in CA certificates. It lists one or more pairs of OIDs; each pair includes an

issuerDomainPolicy and a subjectDomainPolicy. The pairing indicates the issuing CA considers its issuerDomainPolicy equivalent to the subject CA's subjectDomainPolicy.

The issuing CA's users might accept an issuerDomainPolicy for certain applications. The policy mapping defines the list of policies associated with the subject CA that may be accepted as comparable to the issuerDomainPolicy.

This extension MAY be supported by CAs and/or applications, and it MUST be non-critical.

2.1.3. user and CA attributes

- Subject Alternative Name

  The subject alternative names extension allows additional identities to be bound to the subject of the certificate. Defined options include an Internet electronic mail address, a DNS name, an IP address, and a uniform resource identifier (URI). Other options exist, including completely local definitions. Multiple name forms, and multiple instances of each name form, MAY be included. Whenever such identities are to be bound into a certificate, the subject alternative name (or issuer alternative name) extension MUST be used.

  Because the subject alternative name is considered to be definitively bound to the public key, all parts of the subject alternative name MUST be verified by the CA.

```
SubjectAltName ::= GeneralNames

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

GeneralName ::= CHOICE {
     otherName                       [0]     OtherName,
     rfc822Name                      [1]     IA5String,
     dNSName                         [2]     IA5String,
     x400Address                     [3]     ORAddress,
     directoryName                   [4]     Name,
     ediPartyName                    [5]     EDIPartyName,
     uniformResourceIdentifier       [6]     IA5String,
     iPAddress                       [7]     OCTET STRING,
     registeredID                    [8]     OBJECT IDENTIFIER }
```

- Issuer Alternative Names

  This extension is used to associate Internet style identities with the certificate issuer. Issuer alternative names MUST be encoded as in Subject Alternative Name.

  Where present, this extension SHOULD NOT be marked critical.

- Subject Directory Attributes

  The subject directory attributes extension is used to convey identification attributes (e.g., nationality) of the subject. The extension is defined as a sequence of one or more attributes. This extension MUST be non-critical.

2.1.4. co-certification constraints

- Basic Constraints

  The basic constraints extension identifies whether the subject of the certificate is a CA and the maximum depth of valid certification paths that include this certificate.

```
BasicConstraints ::= SEQUENCE {
     cA                      BOOLEAN DEFAULT FALSE,
     pathLenConstraint       INTEGER (0..MAX) OPTIONAL }
```

- Name Constraints

  The name constraints extension, which MUST be used only in a CA certificate, indicates a name space within which all subject names in subsequent certificates in a certification path MUST be located. Restrictions apply to the subject distinguished name and apply to subject alternative names. Restrictions apply only when the specified name form is present. If no name of the type is in the certificate, the certificate is acceptable.

- Policy Constraints

  The policy constraints extension can be used in certificates issued to CAs. The policy constraints extension constrains path validation in two ways. It can be used to prohibit policy mapping or require that each certificate in a path contain an acceptable policy identifier.

2.2. Example

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1424 (0x590)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=LU, O=LuxTrust s.a, CN=LuxTrust Normalised CA
        Validity
            Not Before: Mar  2 11:07:48 2007 GMT
            Not After : Mar  2 11:07:48 2012 GMT
        Subject: C=LU, ST=Luxembourg, L=Luxembourg, O=Ministry of the Economy and Foreign Trade, OU=CASES Lu
CN=*.cases.lu/emailAddress=pst@cases.lu
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (2048 bit)
                Modulus (2048 bit):
                    00:da:34:1c:18:83:0e:96:24:54:0a:c9:58:95:85:
                    2d:65:26:3f:1d:f6:c7:a0:6f:4c:0a:6e:af:ac:1c:
                    0b:b1:4c:e7:ec:d5:4b:73:e6:9c:67:de:87:e5:cd:
                    da:9d:e2:e8:1b:fe:13:27:65:6b:ab:66:d3:f3:1a:
                    e0:92:00:f6:89:de:8d:f0:e4:d3:1a:15:da:44:f0:
                    d8:0e:cb:61:f6:d8:5f:f1:65:ae:c5:f9:63:7b:8f:
                    9e:8d:5c:72:d5:60:be:05:04:1e:14:35:c9:87:0f:
                    fc:fd:e2:e1:3c:7b:a4:2f:de:f3:e9:b9:42:73:b1:
                    52:9d:c8:e8:27:77:af:e3:03:68:44:7c:fd:cc:19:
                    3d:13:41:1b:c3:df:2b:2f:63:76:9b:67:b6:ed:69:
                    6b:77:0b:e8:03:97:10:a1:5b:e9:4c:7e:b2:7c:aa:
                    d2:8b:96:57:da:1a:aa:58:8f:ac:fd:3b:96:57:77:
                    2b:94:52:fb:ea:0b:ab:52:d5:51:39:07:26:f5:3f:
                    67:43:57:6c:30:6c:e0:4a:af:74:c2:ed:98:27:67:
                    36:19:f0:e5:f8:ec:21:43:0f:ec:6e:89:3b:ca:67:
                    9c:e8:6f:fa:39:b1:c4:11:b1:44:14:a0:10:96:cb:
                    1d:07:ca:cf:fd:28:08:2c:15:41:d1:9e:80:d9:b1:
                    72:25
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints: critical
                CA:FALSE
            Authority Information Access:
                OCSP - URI:http://ocsp.luxtrust.lu
                CA Issuers - URI:http://ca.luxtrust.lu/LTNCA.crt

            X509v3 Certificate Policies:
                Policy: 1.3.171.1.1.2.2.1
                  User Notice:
                    Explicit Text: LuxTrust Server Certificate.Not supported by SSCD, Key Generation by Subs
and CPS on http://repository.luxtrust.lu
                  CPS: http://repository.luxtrust.lu
```

```
                  User Notice:
                    Explicit Text: LuxTrust Server Certificate.Not supported by SSCD, Key Generation by Subs
and CPS on http://repository.luxtrust.lu
                  CPS: http://repository.luxtrust.lu
                Policy: 0.4.0.2042.1.3

            Netscape Cert Type:
                SSL Server
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment, Data Encipherment
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
            X509v3 Authority Key Identifier:
                keyid:CE:FE:46:9D:63:2F:89:FD:F2:38:16:25:D8:F1:6C:DE:47:F8:CE:C1

            X509v3 CRL Distribution Points:
                URI:http://crl.luxtrust.lu/LTNCA.crl

            X509v3 Subject Key Identifier:
                66:2B:FD:C1:89:E5:54:1B:87:A8:3C:5B:BC:20:84:33:CE:57:FF:40
    Signature Algorithm: sha1WithRSAEncryption
        3a:c3:89:1a:8d:c0:17:35:ac:9c:73:23:33:4f:b4:cc:9d:5f:
        08:38:ed:cb:af:86:64:67:66:61:ff:de:66:55:c6:31:c9:ff:
        eb:75:bd:51:d6:24:af:e6:14:cb:91:92:0b:0b:ec:96:39:8f:
        fc:5d:7a:fe:d4:4d:89:92:5e:f6:45:89:5d:bc:e0:4e:0b:9b:
        1f:e1:4a:41:3d:59:3e:d0:65:08:44:58:bf:f3:eb:78:d4:7d:
        c0:15:cd:8e:7c:9b:b3:af:39:8d:cb:8d:4c:bc:e1:f0:ef:7d:
        52:03:11:af:a5:d0:4d:d0:2a:ff:9d:63:b1:d8:5b:ad:1b:ba:
        9d:c0:14:b5:68:33:d5:6e:40:cb:c8:72:26:ef:f7:95:0a:e2:
        3f:18:01:dd:95:22:02:ea:37:08:eb:13:48:64:40:54:2b:10:
        06:80:cd:31:ef:1f:b6:2c:24:dc:6f:3f:64:07:84:a2:d0:9f:
        2d:97:71:92:f7:19:93:55:92:6f:60:05:88:35:ce:49:e0:ba:
        53:38:31:22:53:48:3f:94:7b:5d:5d:29:75:92:d6:2a:69:7c:
        40:33:35:bb:c4:6e:f4:ba:27:d5:95:46:2d:f2:17:e1:d6:36:
        62:06:fc:e8:51:eb:42:ac:86:65:bf:8d:7d:31:7a:37:96:34:
        4f:82:f3:17
-----BEGIN CERTIFICATE-----
MIIFgjCCBGqgAwIBAgICBZAwDQYJKoZIhvcNAQEFBQAwRTELMAkGA1UEBhMCTFUx
FTATBgNVBAoTDEx1eFRydXN0IHMuYTEfMB0GA1UEAxMWTHV4VHJ1c3QgTm9ybWFs
aXNlZCBDQTAeFw0wNzAzMDIxMTA3NDhaFw0xMjAzMDIxMTA3NDhaMIG4MQswCQYD
VQQGEwJMVTETMBEGA1UECBMKTHV4ZW1ib3VyZzETMBEGA1UEBxMKTHV4ZW1ib3Vy
ZzEyMDAGA1UEChMpTWluaXN0cnkgb2YgdGhlIEVjb25vbXkgYW5kIEZvcmVpZ24g
VHJhZGUxGTAXBgNVBAsTEENBU0VTIEx1eGVtYm91cmcxEzARBgNVBAMUCiouY2Fz
ZXMubHUxGzAZBgkqhkiG9w0BCQEWDHBzdEBjYXNlcy5sdTCCASIwDQYJKoZIhvcN
AQEBBQADggEPADCCAQoCggEBANo0HBiDDpYkVArJWJWFLWUmPx32x6BvTApur6wc
C7FM5+zVS3PmnGfeh+XN2p3i6Bv+Eydla6tm0/Ma4JIA9onejfDk0xoV2kTw2A7L
```

3. IETF - Internet X.509 Public Key Infrastructure (PKIX)

PKIX standardisation areas:

It describes the basic certificate fields and the extensions to be supported for the Certificates and the Certificate Revocation Lists. Then, it talks about the basic and extended Certificate Path Validation. Finally, it covers the supported cryptographic algorithms.

- Management protocols.

  Management protocols are the protocols that are required to support on–line interactions between PKI user and management entities. The possible set of functions that can be supported by management protocols is

  - registration of entity, that takes place prior to issuing the certificate,
  - initialisation, for example generation of key-pair,
  - certification, the issuance of the certificate,
  - key–pair recovery, the ability to recover lost keys,
  - key–pair update, when the certificate expires and a new key–pair and certificate have to be generated,
  - revocation request, when an authorised person advices the CA to include a specific certificate into the revocation list,
  - cross-certification, when two CAs exchange information in order to generate a cross–certificate.

  The PKIX standard first discusses the assumptions and restrictions of the protocols. Then, it provides the data structures used for the PKI management messages and defines the functions that conforming implementations must carry out. Finally, it describes a simple protocol for transporting PKI messages.

- Operational protocols.

  The operational protocols are:

- required to deliver certificates and CRLs to certificate–using client systems.

  There is an emphasis to have a variety of distribution mechanisms for the certificates and the CRLs, using for example, LDAP, HTTP and FTP. For example, the retrieval of the CRL by a merchant to check whether a certificate is valid, constitutes an operational protocol.

  Currently they describe how LDAP, FTP and HTTP can be used as operational protocols, as well how online validation (OCSP) should be implemented.

- Certificate policies and Certificate Practice Statements.

  The Certificate Policies and the Certificate Practice Statements are recommendations of documents that will describe the obligations and other rules with regard the usage of the Certificate.

  The purpose of this document is to establish a clear relationship between certificate policies and CPSs, and to present a framework to assist the writers of certificate policies or CPSs with their tasks. In particular, the framework identifies the elements that may need to be considered in formulating a certificate policy or a CPS. The purpose is not to define particular certificate policies or CPSs, per se.

- Time–stamping and data–certification/validation services.

  The time–stamping services define a trusted third–party that creates time stamp tokens in order to indicate that a datum existed at a particular point in time. The data certification and validation services provide certification of possesion of data and claim of possesion of data, and validation of digitally signed documents and certificates.

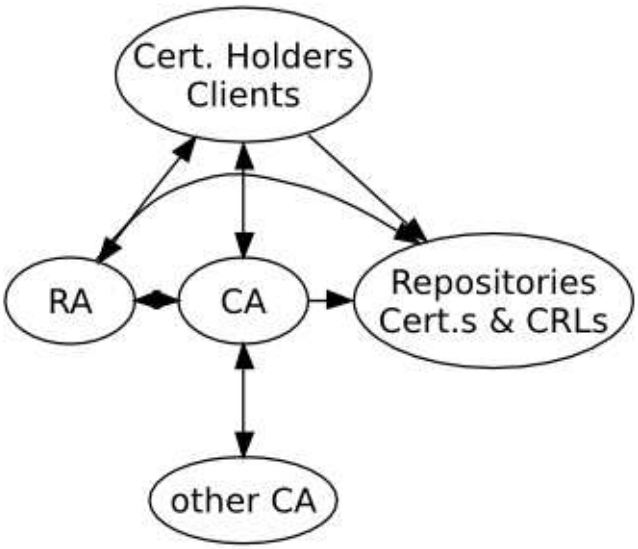
The relevant Request For Comments (RFC) documents are depicted in the following table

Subject	RFC
Certificate and Certificate Revocation List (CRL) Profile	**RFC 3280**
Certificate Management Protocol (CMP)	RFC 4210
Operational protocols	RFC 3494 (LDAP), RFC 2585, **RFC 2560 (OCSP)**
Certificate Policy and Certification Practices Framework	**RFC 3647**
Time–stamping and data–certification services	**RFC 3628**, RFC 3161

3.1. PKI - Public-Key Infrastructure

A PKI is a set of hardware, software, people, policies and procedures needed to create, manage, store, distribute and revoke PKCs based on public–key cryptography.

A PKI consists of five types of components.



Type of component	Description
Certification Authorities (CAs)	to issue and revoke PKCs
Registration Authorities (RAs)	to vouch for the binding between public keys and certificate holder identities and other attributes
Certificate holders (end entities, EE)	to sign and encrypt digital documents
Clients (end entities, EE)	to validate digital signatures and their certification path from a known public key of a trusted CA
Repositories	to store and make available certificates and Certificate Revocation Lists (CRLs)

The End–entity, using management transactions, sends its certificate request to the Registration Authority for approval. If it is actually approved, it is forwarded to the Certification Authority for signing. The Certification Authority verifies the certificate request and if it passes the verification, it is signed and the Certificate is produced. To public the Certificate, the CA sends it to Certificate Repository for collection from the End–entity.

The diagram shows that the End–entity can communicate directly with the CA. According to the PKIX

recommendations, it is possible to implement the functionality within the CA. Although it is a bit confusing, the diagram shows all possible communications, regardless of the implementation decisions.

Additionally, both the CA and RA are shown to deliver Certificates to the repository. Depending on the implementation, one of the two is chosen.

For the issue of the revocation of the certificates, a similar course with the generation of the Certificates is taken. The End–entity asks the RA to have its Certificate revoked, the RA decides and possibly forwards it to the CA, the CA updates the revocation list and publishes it on the CRL repository.

Finally, the End–entities can check the validity of a specific Certificate using an operational protocol.

The following diagram shows the relationship between the entities defined above in terms of the PKI management operations. The letters in the diagram indicate "protocols" in the sense that a defined set of PKI management messages can be sent along each of the lettered lines.

At a **high level**, the set of operations for which management messages are defined can be grouped as follows.

- CA establishment

  When establishing a new CA, certain steps are required (e.g., production of initial CRLs, export of CA public key).

- End entity initialization

  This includes importing a root CA public key and requesting information about the options supported by a PKI management entity.

- Certification

  Various operations result in the creation of new certificates:

  - initial registration/certification

    This is the process whereby an end entity first makes itself known to a CA or RA, prior to the CA issuing a certificate or certificates for that end entity. The end result of this process (when it is successful) is that a CA issues a certificate for an end entity’s public key, and returns that certificate to the end entity and/or posts that certificate in a public repository. This process may, and typically will, involve multiple "steps", possibly including an initialization of the end entity’s equipment. For example, the end entity’s equipment must be securely initialized with the public key of a CA, to be used in validating certificate paths. Furthermore, an end entity typically needs to be initialized with its own key pair(s).

  - key pair update

    Every key pair needs to be updated regularly (i.e., replaced with a new key pair), and a new certificate needs to be issued.

  - certificate update

    As certificates expire, they may be "refreshed" if nothing relevant in the environment has changed.

  - CA key pair update

    As with end entities, CA key pairs need to be updated regularly; however, different mechanisms are required.

  - cross-certification request

    A "cross-certificate" is a certificate in which the subject CA and the issuer CA are distinct and SubjectPublicKeyInfo contains a verification key (i.e., the certificate has been issued for the subject CA’s signing key pair).

    One CA requests issuance of a cross-certificate from another CA. For the purposes of this standard, the following terms are defined. When it is necessary to distinguish more finely, the following terms may be used: a cross-certificate is called an "inter-domain cross-certificate" if the subject and issuer CAs belong to different administrative domains; it is called an "intra-domain cross-certificate" otherwise.

  - cross-certificate update

    Similar to a normal certificate update, but involving a cross-certificate.

- Certificate/CRL discovery operations

  Some PKI management operations result in the publication of certificates or CRLs:

  - certificate publication

    Having gone to the trouble of producing a certificate, some means for publishing it is needed. The "means" defined in PKIX MAY involve methods (LDAP, for example) as described in [RFC2559], [RFC2585] (the "Operational Protocols" documents of the PKIX series of specifications).

- CRL publication

  As for certificate publication.

- Recovery operations

  Some PKI management operations are used when an end entity has "lost" its PSE:

  - key pair recovery

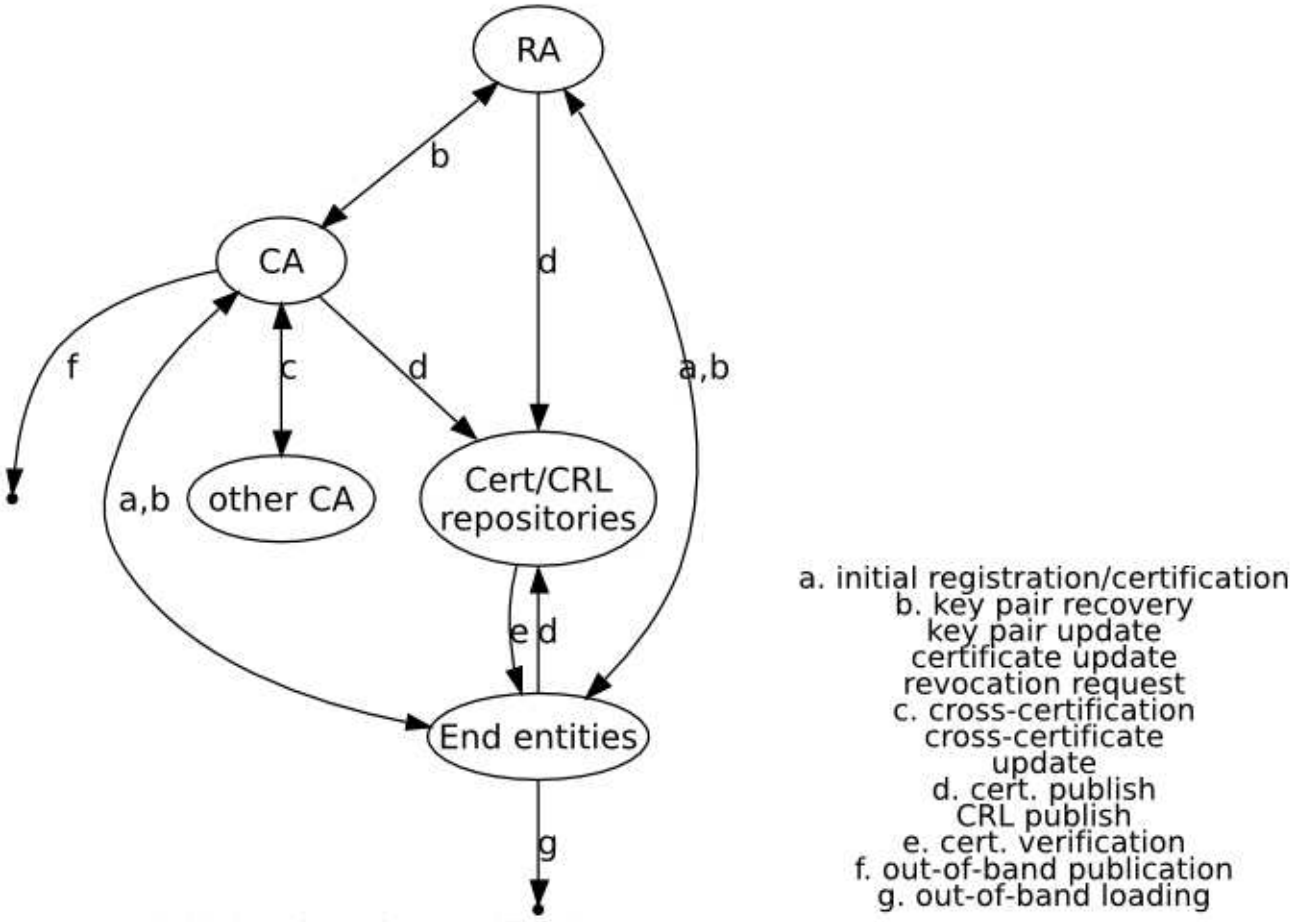
    As an option, user client key materials (e.g., a user's private key used for decryption purposes) MAY be backed up by a CA, an RA, or a key backup system associated with a CA or RA. If an entity needs to recover these backed up key materials (e.g., as a result of a forgotten password or a lost key chain file), a protocol exchange may be needed to support such recovery.

- Revocation operations

  Some PKI operations result in the creation of new CRL entries and/or new CRLs:

  - revocation request

    An authorized person advises a CA of an abnormal situation requiring certificate revocation.



4. Certificate revocation list (CRL) - RFC 3280

- A complete CRL: lists all unexpired certificates, within its scope, that have been revoked for one of the revocation reasons covered by the CRL scope.
- A delta CRL: only lists those certificates, within its scope, whose revocation status has changed since the issuance of a referenced complete CRL.

  The referenced complete CRL is referred to as a base CRL. The scope of a delta CRL MUST be the same as the base CRL that it references.

```
CertificateList ::= SEQUENCE {
      TBSCertList ::=    SEQUENCE  {
              version,         (optional)
              signature,       (algorithm)
```

```
                issuer,
                thisUpdate,
                nextUpdate,     (optional)
                revokedCertificates       SEQUENCE OF SEQUENCE {
                        userCertificate,        (certificate serial number),
                        revocationDate,
                        crlEntryExtensions      (optional)
                },              (optional)
                crlExtensions,  (optional)
        }
        signatureAlgorithm,     (algorithm)
        signatureValue,
}
```

4.1. CRL Entry Extensions

```
    reasonCode ::= < CRLReason >

    CRLReason ::= ENUMERATED {
         unspecified             (0),
         keyCompromise           (1),
         cACompromise            (2),
         affiliationChanged      (3),
         superseded              (4),
         cessationOfOperation    (5),
         certificateHold         (6),
         removeFromCRL           (8),
         privilegeWithdrawn      (9),
         aACompromise           (10) }
```

4.2. Basic Certificate Processing

The basic path processing actions to be performed for a certificate validation:

1. The certificate was signed with the working_public_key_algorithm using the working_public_key and the working_public_key_parameters.
2. The certificate validity period includes the current time.
3. At the current time, the certificate is not revoked and is not on hold status.

   This may be determined by obtaining the appropriate CRL status information, or by out-of-band mechanisms.

4. The certificate issuer name is the working_issuer_name.

5. Online Certificate Status Protocol (OCSP) - RFC 2560

In lieu of or as a supplement to checking against a periodic CRL, it may be necessary to obtain timely information regarding the revocation status of a certificate (e.g. high-value funds transfer or large stock trades). -> **OCSP**

The Online Certificate Status Protocol (OCSP) enables applications to determine the (revocation) state of an identified certificate. OCSP may be used to satisfy some of the operational requirements of providing more timely revocation information than is possible with CRLs and may also be used to obtain additional status information. An OCSP client issues a status request to an OCSP responder and suspends acceptance of the certificate in question until the responder provides a response.

This protocol specifies the data that needs to be exchanged between an application checking the status of a certificate and the server providing that status.

- An OCSP request contains the following data:
  - protocol version
  - service request
  - target certificate identifier
  - optional extensions
- Upon receipt of a request, an OCSP Responder determines if:
  1. the message is well formed
  2. the responder is configured to provide the requested service and
  3. the request contains the information needed by the responder

If any one of the prior conditions are not met, the OCSP responder produces an error message; otherwise, it returns a definitive response.

OCSP responses can be of various types. An OCSP response consists of a response type and the bytes of the actual response. There is one basic type of OCSP response that MUST be supported by all OCSP servers and clients. The rest of this section pertains only to this basic response type.

- All definitive response messages SHALL be digitally signed. The key used to sign the response MUST belong to one of the following:
  - the CA who issued the certificate in question

    - a Trusted Responder whose public key is trusted by the requester
    - a CA Designated Responder (Authorized Responder)

      who holds a specially marked certificate issued directly by the CA, indicating that the responder may issue OCSP responses for that CA

- A definitive response message is composed of:
    - version of the response syntax
    - name of the responder
    - responses for each of the certificates in a request
    - optional extensions
    - signature algorithm OID
    - signature computed across hash of the response
- The response for each of the certificates in a request consists of:
    - target certificate identifier
    - certificate status value
    - response validity interval
    - optional extensions
- This specification defines the following definitive response indicators for use in the certificate status value:
    - good

      The "good" state indicates a positive response to the status inquiry. At a minimum, this positive response indicates that the certificate is not revoked, but does not necessarily mean that the certificate was ever issued or that the time at which the response was produced is within the certificate's validity interval. Response extensions may be used to convey additional information on assertions made by the responder regarding the status of the certificate such as positive statement about issuance, validity, etc.

    - revoked

      The "revoked" state indicates that the certificate has been revoked (either permanantly or temporarily (on hold)).

    - unknown

      The "unknown" state indicates that the responder doesn't know about the certificate being requested.

- Functional Requirements :
    - Certificate Content (via AuthorityInfoAccess extension)

      In order to convey to OCSP clients a well-known point of information access, CAs SHALL provide the capability to include the AuthorityInfoAccess extension in certificates that can be checked using OCSP. Alternatively, the accessLocation for the OCSP provider may be configured locally at the OCSP client.

      CAs that support an OCSP service, either hosted locally or provided by an Authorized Responder, MUST provide for the inclusion of a value for a uniformResourceIndicator (URI) accessLocation and the OID value id-ad-ocsp for the accessMethod in the AccessDescription SEQUENCE.

      The value of the accessLocation field in the subject certificate defines the transport (e.g. HTTP) used to access the OCSP responder and may contain other transport dependent information (e.g. a URL).

    - Signed Response Acceptance Requirements

      Prior to accepting a signed response as valid, OCSP clients SHALL confirm that:
        1. The certificate identified in a received response corresponds to that which was identified in the corresponding request;
        2. The signature on the response is valid;
        3. The identity of the signer matches the intended recipient of the request.
        4. The signer is currently authorized to sign the response.
        5. The time at which the status being indicated is known to be correct (thisUpdate) is sufficiently recent.
        6. When available, the time at or before which newer information will be available about the status of the certificate (nextUpdate) is greater than the current time.

6. Certificate Policy and Certification Practices Framework - RFC 3647

- Certificate policy (CP)

  The X.509 standard defines a CP as "a named set of rules that indicates the applicability of a certificate to a particular community and/or class of application with common security requirements".

  An X.509 Version 3 certificate may identify a specific applicable CP, which may be used by a relying party to decide whether or not to trust a certificate, associated public key, or any digital signatures verified using the public key for a particular purpose.

  When a certification authority issues a certificate, it is providing a statement to a certificate user (i.e., a relying party) that a particular public key is bound to the identity and/or other attributes of a particular entity (the certificate subject, which is usually also the subscriber). The extent to which the relying party should rely

on that statement by the CA, however, needs to be assessed by the relying party or entity controlling or coordinating the way relying parties or relying party applications use certificates. Different certificates are issued following different practices and procedures, and may be suitable for different applications and/or purposes.

- CPs typically fall into two major categories:
  - some CPs "indicate the applicability of a certificate to a particular community"

    These CPs set forth requirements for certificate usage and requirements on members of a community. For instance, a CP may focus on the needs of a geographical community, such as the ETSI policy requirements for CAs issuing qualified certificates.

    Also, a CP of this kind may focus on the needs of a specific vertical-market community, such as financial services.

  - the second category "indicate the applicability of a certificate to a ... class of application with common security requirements"

    These CPs identify a set of applications or uses for certificates and say that these applications or uses require a certain level of security. They then set forth PKI requirements that are appropriate for these applications or uses. A CP within this category often sets requirements appropriate for a certain "level of assurance" provided by certificates, relative to certificates issued pursuant to related CPs. These levels of assurance may correspond to "classes" or "types" of certificates.

  A CP is represented in a certificate by a unique number called an "Object Identifier" (OID).

- CPs also constitute a basis for an audit, accreditation, or another assessment of a CA (cross-certification).

  Each CA can be assessed against one or more certificate policies or CPSs that it is recognized as implementing. When one CA issues a CA-certificate for another CA, the issuing CA must assess the set of certificate policies for which it trusts the subject CA (such assessment may be based upon an assessment with respect to the certificate policies involved). The assessed set of certificate policies is then indicated by the issuing CA in the CA-certificate. The X.509 certification path processing logic employs these CP indications in its well-defined trust model.

- Certification Practice Statement (CPS)

  The term certification practice statement (CPS) is defined as: "A statement of the practices which a certification authority employs in issuing certificates".

  A CPS establishes practices concerning lifecycle services in addition to issuance, such as certificate management (including publication and archiving), revocation, and renewal or re-keying.

  - "A certification practice statement may take the form of a declaration by the certification authority of the details of its trustworthy system and the practices it employs in its operations and in support of issuance of a certificate ..."

    This form of CPS is the most common type, and can vary in length and level of detail.

    Some PKIs may not have the need to create a thorough and detailed statement of practices. For example, the CA may itself be the relying party and would already be aware of the nature and trustworthiness of its services. In other cases, a PKI may provide certificates providing only a very low level of assurances where the applications being secured may pose only marginal risks if compromised. In these cases, an organization establishing a PKI may only want to write or have CAs use a subscriber agreement, relying party agreement, or agreement combining subscriber and relying party terms, depending on the role of the different PKI participants. In such a PKI, that agreement may serve as the only "statement of practices" used by one or more CAs within that PKI. Consequently, that agreement may also be considered a CPS and can be entitled or subtitled as such.

    Likewise, since a detailed CPS may contain sensitive details of its system, a CA may elect not to publish its entire CPS. It may instead opt to publish a CPS Summary (or CPS Abstract). The CPS Summary would contain only those provisions from the CPS that the CA considers to be relevant to the participants in the PKI (such as the responsibilities of the parties or the stages of the certificate lifecycle). A CPS Summary, however, would not contain those sensitive provisions of the full CPS that might provide an attacker with useful information about the CA’s operations. Throughout this document, the use of "CPS" includes both a detailed CPS and a CPS Summary (unless otherwise specified).

  - CPSs do not automatically constitute contracts and do not automatically bind PKI participants as a contract would (dual-purpose however possible).

    Where a document serves the dual purpose of being a subscriber or relying party agreement and CPS, the document is intended to be a contract and constitutes a binding contract to the extent that a subscriber or relying party agreement would ordinarily be considered as such. Most CPSs, however, do not serve such a dual purpose. Therefore, in most cases, a CPS’s terms have a binding effect as contract terms only if a separate document creates a contractual relationship between the parties and that document incorporates part or all of the CPS by reference. Further, if a particular PKI employs a CPS Summary (as opposed to the entire CPS), the CPS Summary could be incorporated into any applicable subscriber or relying party agreement.

    In the future, a court or applicable statutory or regulatory law may declare that a certificate itself is a

document that is capable of creating a contractual relationship, to the extent its mechanisms designed for incorporation by reference (such as the Certificate Policies extension and its qualifiers) indicate that terms of its use appear in certain documents. In the meantime, however, some subscriber agreements and relying party agreements may incorporate a CPS by reference and therefore make its terms binding on the parties to such agreements.

- The main differences between CPs and CPSs can be summarized as follows:
  1. A PKI uses a CP to establish requirements that state what participants within it must do. A single CA or organization can use a CPS to disclose how it meets the requirements of a CP or how it implements its practices and controls.
  2. A CP facilitates interoperation through cross-certification, unilateral certification, or other means. Therefore, it is intended to cover multiple CAs. By contrast, a CPS is a statement of a single CA or organization. Its purpose is not to facilitate interoperation (since doing so is the function of a CP).
  3. A CPS is generally more detailed than a CP and specifies how the CA meets the requirements specified in the one or more CPs under which it issues certificates.
- In addition to populating the certificate policies extension with the applicable CP object identifier, a certification authority may include, in certificates it issues, a reference to its certification practice statement.

6.1. Recommended CP or CPS outline

In order to comply with the RFC, the drafters of a compliant CP or CPS are strongly advised to adhere to the following outline:

1. INTRODUCTION
    1. Overview
    2. Document name and identification
    3. PKI participants
        1. Certification authorities
        2. Registration authorities
        3. Subscribers
        4. Relying parties
        5. Other participants
    4. Certificate usage
        1. Appropriate certificate uses
        2. Prohibited certificate uses
    5. Policy administration
        1. Organization administering the document
        2. Contact person
        3. Person determining CPS suitability for the policy
        4. CPS approval procedures
    6. Definitions and acronyms
2. PUBLICATION AND REPOSITORY RESPONSIBILITIES
    1. Repositories
    2. Publication of certification information
    3. Time or frequency of publication
    4. Access controls on repositories
3. IDENTIFICATION AND AUTHENTICATION
    1. Naming
        1. Types of names
        2. Need for names to be meaningful
        3. Anonymity or pseudonymity of subscribers
        4. Rules for interpreting various name forms
        5. Uniqueness of names
        6. Recognition, authentication, and role of trademarks
    2. Initial identity validation
        1. Method to prove possession of private key
        2. Authentication of organization identity
        3. Authentication of individual identity
        4. Non-verified subscriber information
        5. Validation of authority
        6. Criteria for interoperation
    3. Identification and authentication for re-key requests
        1. Identification and authentication for routine re-key
        2. Identification and authentication for re-key after revocation
    4. Identification and authentication for revocation request
4. CERTIFICATE LIFE-CYCLE OPERATIONAL REQUIREMENTS
    1. Certificate Application
        1. Who can submit a certificate application
        2. Enrollment process and responsibilities
    2. Certificate application processing
        1. Performing identification and authentication functions
        2. Approval or rejection of certificate applications

3. Time to process certificate applications
3. Certificate issuance
    1. CA actions during certificate issuance
    2. Notification to subscriber by the CA of issuance of certificate
4. Certificate acceptance
    1. Conduct constituting certificate acceptance
    2. Publication of the certificate by the CA
    3. Notification of certificate issuance by the CA to other entities
5. Key pair and certificate usage
    1. Subscriber private key and certificate usage
    2. Relying party public key and certificate usage
6. Certificate renewal
    1. Circumstance for certificate renewal
    2. Who may request renewal
    3. Processing certificate renewal requests
    4. Notification of new certificate issuance to subscriber
    5. Conduct constituting acceptance of a renewal certificate
    6. Publication of the renewal certificate by the CA
    7. Notification of certificate issuance by the CA to other entities
7. Certificate re-key
    1. Circumstance for certificate re-key
    2. Who may request certification of a new public key
    3. Processing certificate re-keying requests
    4. Notification of new certificate issuance to subscriber
    5. Conduct constituting acceptance of a re-keyed certificate
    6. Publication of the re-keyed certificate by the CA
    7. Notification of certificate issuance by the CA to other entities
8. Certificate modification
    1. Circumstance for certificate modification
    2. Who may request certificate modification
    3. Processing certificate modification requests
    4. Notification of new certificate issuance to subscriber
    5. Conduct constituting acceptance of modified certificate
    6. Publication of the modified certificate by the CA
    7. Notification of certificate issuance by the CA to other entities
9. Certificate revocation and suspension
    1. Circumstances for revocation
    2. Who can request revocation
    3. Procedure for revocation request
    4. Revocation request grace period
    5. Time within which CA must process the revocation request
    6. Revocation checking requirement for relying parties
    7. CRL issuance frequency (if applicable)
    8. Maximum latency for CRLs (if applicable)
    9. On-line revocation/status checking availability
    10. On-line revocation checking requirements
    11. Other forms of revocation advertisements available
    12. Special requirements re key compromise
    13. Circumstances for suspension
    14. Who can request suspension
    15. Procedure for suspension request
    16. Limits on suspension period
10. Certificate status services
    1. Operational characteristics
    2. Service availability
    3. Optional features
    4. End of subscription
    5. Key escrow and recovery
    6. Key escrow and recovery policy and practices
    7. Session key encapsulation and recovery policy and practices

5. FACILITY, MANAGEMENT, AND OPERATIONAL CONTROLS

1. Physical controls
    1. Site location and construction
    2. Physical access
    3. Power and air conditioning
    4. Water exposures
    5. Fire prevention and protection
    6. Media storage
    7. Waste disposal
    8. Off-site backup
2. Procedural controls
    1. Trusted roles
    2. Number of persons required per task
    3. Identification and authentication for each role
    4. Roles requiring separation of duties
3. Personnel controls
    1. Qualifications, experience, and clearance requirements
    2. Background check procedures

3. Training requirements
4. Retraining frequency and requirements
5. Job rotation frequency and sequence
6. Sanctions for unauthorized actions
7. Independent contractor requirements
8. Documentation supplied to personnel

4. Audit logging procedures
    1. Types of events recorded
    2. Frequency of processing log
    3. Retention period for audit log
    4. Protection of audit log
    5. Audit log backup procedures
    6. Audit collection system (internal vs. external)
    7. Notification to event-causing subject
    8. Vulnerability assessments
5. Records archival
    1. Types of records archived
    2. Retention period for archive
    3. Protection of archive
    4. Archive backup procedures
    5. Requirements for time-stamping of records
    6. Archive collection system (internal or external)
    7. Procedures to obtain and verify archive information
6. Key changeover
7. Compromise and disaster recovery
    1. Incident and compromise handling procedures
    2. Computing resources, software, and/or data are corrupted
    3. Entity private key compromise procedures
    4. Business continuity capabilities after a disaster
8. CA or RA termination

6. TECHNICAL SECURITY CONTROLS

1. Key pair generation and installation
    1. Key pair generation
    2. Private key delivery to subscriber
    3. Public key delivery to certificate issuer
    4. CA public key delivery to relying parties
    5. Key sizes
    6. Public key parameters generation and quality checking
    7. Key usage purposes (as per X.509 v3 key usage field)
2. Private Key Protection and Cryptographic Module Engineering Controls
    1. Cryptographic module standards and controls
    2. Private key (n out of m) multi-person control
    3. Private key escrow
    4. Private key backup
    5. Private key archival
    6. Private key transfer into or from a cryptographic module
    7. Private key storage on cryptographic module
    8. Method of activating private key
    9. Method of deactivating private key
    10. Method of destroying private key
    11. Cryptographic Module Rating
3. Other aspects of key pair management
    1. Public key archival
    2. Certificate operational periods and key pair usage periods
4. Activation data
    1. Activation data generation and installation
    2. Activation data protection
    3. Other aspects of activation data
5. Computer security controls
    1. Specific computer security technical requirements
    2. Computer security rating
6. Life cycle technical controls
    1. System development controls
    2. Security management controls
    3. Life cycle security controls
7. Network security controls
8. Time-stamping

7. CERTIFICATE, CRL, AND OCSP PROFILES

1. Certificate profile
    1. Version number(s)
    2. Certificate extensions
    3. Algorithm object identifiers
    4. Name forms
    5. Name constraints
    6. Certificate policy object identifier
    7. Usage of Policy Constraints extension

        8. Policy qualifiers syntax and semantics
        9. Processing semantics for the critical Certificate Policies extension
    2. CRL profile
        1. Version number(s)
        2. CRL and CRL entry extensions
    3. OCSP profile
        1. Version number(s)
        2. OCSP extensions

8. COMPLIANCE AUDIT AND OTHER ASSESSMENTS

    1. Frequency or circumstances of assessment
    2. Identity/qualifications of assessor
    3. Assessor’s relationship to assessed entity
    4. Topics covered by assessment
    5. Actions taken as a result of deficiency
    6. Communication of results

9. OTHER BUSINESS AND LEGAL MATTERS

    1. Fees
        1. Certificate issuance or renewal fees
        2. Certificate access fees
        3. Revocation or status information access fees
        4. Fees for other services
        5. Refund policy
    2. Financial responsibility
        1. Insurance coverage
        2. Other assets
        3. Insurance or warranty coverage for end-entities
    3. Confidentiality of business information
        1. Scope of confidential information
        2. Information not within the scope of confidential information
        3. Responsibility to protect confidential information
    4. Privacy of personal information
        1. Privacy plan
        2. Information treated as private
        3. Information not deemed private
        4. Responsibility to protect private information
        5. Notice and consent to use private information
        6. Disclosure pursuant to judicial or administrative process
        7. Other information disclosure circumstances
    5. Intellectual property rights
    6. Representations and warranties
        1. CA representations and warranties
        2. RA representations and warranties
        3. Subscriber representations and warranties
        4. Relying party representations and warranties
        5. Representations and warranties of other participants
    7. Disclaimers of warranties
    8. Limitations of liability
    9. Indemnities
    10. Term and termination
        1. Term
        2. Termination
        3. Effect of termination and survival
    11. Individual notices and communications with participants
    12. Amendments
        1. Procedure for amendment
        2. Notification mechanism and period
        3. Circumstances under which OID must be changed
    13. Dispute resolution provisions
    14. Governing law
    15. Compliance with applicable law
    16. Miscellaneous provisions
        1. Entire agreement
        2. Assignment
        3. Severability
        4. Enforcement (attorneys’ fees and waiver of rights)
        5. Force Majeure
    17. Other provisions

7. Time-Stamping Authorities (TSAs) - RFC 3628

In creating reliable and manageable digital evidence it is necessary to have an agreed upon method of associating time data to transaction so that they might be compared to each other at a later time. The quality of this evidence is based on creating and managing the data structure that represent the events and the quality of the parametric data points that anchor them to the real world. In this instance this being the time data and how it was applied.

A typical transaction is a digitally signed document, where it is necessary to prove that the digital signature from the signer was applied when the signer’s certificate was valid.

A timestamp or a time mark (which is an audit record kept in a secure audit trail from a trusted third party) applied to a digital signature value proves that the digital signature was created before the date included in the time-stamp or time mark.

To prove the digital signature was generated while the signer’s certificate was valid, the digital signature must be verified and the following conditions satisfied:

1. the time-stamp (or time mark) was applied before the end of the validity period of the signer’s certificate,
2. the time-stamp (or time mark) was applied either while the signer’s certificate was not revoked or before the revocation date of the certificate.

The electronic time stamp is gaining interest from the business sector as an important component of electronic signatures. It is also featured by the ETSI Electronic Signature Format standard (TS 101 733) or Electronic Signature Formats for long term electronic signatures (RFC 3126), built upon the Time-Stamp Protocol (RFC 3161). Agreed minimum security and quality requirements are necessary in order to ensure trustworthy validation of long-term electronic signatures.

The European Directive 1999/93/EC defines certification service provider as "an entity or a legal or natural person who issues certificates or provides other services related to electronic signatures". One example of a certification-service-provider is a Time-Stamping Authority.

7.1. Time-Stamp Token

The TSA shall ensure that time-stamp tokens are issued securely and include the correct time. In particular:

1. The time-stamp token shall include an identifier for the time-stamp policy;
2. Each time-stamp token shall have a unique identifier;
3. The time values the TSU uses in the time-stamp token shall be traceable to at least one of the real time values distributed by a UTC(k) laboratory.

   NOTE: The Bureau International des Poids et Mesures (BIPM) computes UTC on the basis of its local representations UTC(k) from a large ensemble of atomic clocks in national metrology institutes and national astronomical observatories round the world. The BIPM disseminates UTC through its monthly Circular T. This is available on the BIPM website (www.bipm.org) and it officially identifies all those institutes having recognized UTC(k) time scales.

4. The time included in the time-stamp token shall be synchronized with UTC within the accuracy defined in this policy and, if present, within the accuracy defined in the time-stamp token itself;
5. If the time-stamp provider’s clock is detected as being out of the stated accuracy then time-stamp tokens shall not be issued.
6. The time-stamp token shall include a representation (e.g., hash value) of the datum being time-stamped as provided by the requestor;

7. The time-stamp token shall be signed using a key generated exclusively for this purpose.

   NOTE: A protocol for a time-stamp token is defined in RFC 3631 and profiled in TS 101 861.

   NOTE: In the case of a number of requests at approximately the same time, the ordering of the time within the accuracy of the TSU clock is not mandated.

8. The time-stamp token shall include further :
   - where applicable, an identifier for the country in which the TSA is established;
   - an identifier for the TSA;
   - an identifier for the unit which issues the time-stamps.

A time-stamping service supports assertions of proof that a datum existed before a particular time. A TSA may be operated as a Trusted Third Party (TTP) service, though other operational models may be appropriate, e.g., an organization might require a TSA for internal time-stamping purposes.

Non-repudiation services require the ability to establish the existence of data before specified times. This protocol may be used as a building block to support such services.

- The TSA is REQUIRED:
  - to use a trustworthy source of time.
  - to include a trustworthy time value for each time-stamp token.
  - to include a unique integer for each newly generated time-stamp token.
  - to produce a time-stamp token upon receiving a valid request from the requester, when it is possible.
  - to include within each time-stamp token an identifier to uniquely indicate the security policy under which the token was created.
  - to only time-stamp a hash representation of the datum

    i.e., a data imprint associated with a one-way collision resistant hash-function uniquely identified by an OID.

  - to examine the OID of the one-way collision resistant hash-function and to verify that the hash value length is consistent with the hash algorithm.

- The TSA is REQUIRED (cont'd):
  - not to examine the imprint being time-stamped in any way (other than to check its length, as specified in the previous bullet).
  - not to include any identification of the requesting entity in the time-stamp tokens.
  - to sign each time-stamp token using a key generated exclusively for this purpose and have this property of the key indicated on the corresponding certificate.
  - to include additional information in the time-stamp token, if asked by the requester using the extensions field, only for the extensions that are supported by the TSA. If this is not possible, the TSA SHALL respond with an error message.

7.2. Time-Stamp Protocol (TSP) - RFC 3161 - Request Format

```
TimeStampReq ::= SEQUENCE {
     version                      INTEGER < v1(1) >,
     messageImprint               MessageImprint,
       --a hash algorithm OID and the hash value of the data to be
       --time-stamped
     reqPolicy             TSAPolicyId               OPTIONAL,
     nonce                 INTEGER                   OPTIONAL,
     certReq               BOOLEAN                   DEFAULT FALSE,
     extensions            [0] IMPLICIT Extensions   OPTIONAL }

MessageImprint ::= SEQUENCE   {
     hashAlgorithm                AlgorithmIdentifier,
     hashedMessage                OCTET STRING }
```

The messageImprint field SHOULD contain the hash of the datum to be time-stamped. The hash is represented as an OCTET STRING. Its length MUST match the length of the hash value for that algorithm (e.g., 20 bytes for SHA-1 or 16 bytes for MD5).

The reqPolicy field, if included, indicates the TSA policy under which the TimeStampToken SHOULD be provided.

The nonce, if included, allows the client to verify the timeliness of the response when no local clock is available. The nonce is a large random number with a high probability that the client generates it only once (e.g., a 64 bit integer). In such a case the same nonce value MUST be included in the response, otherwise the response shall be rejected.

If the certReq field is present and set to true, the TSA’s public key certificate that is referenced by the ESSCertID identifier inside a SigningCertificate attribute in the response MUST be provided by the TSA in the certificates field from the SignedData structure in that response. That field may also contain other certificates.

If the certReq field is missing or if the certReq field is present and set to false then the certificates field from the SignedData structure MUST not be present in the response.

The time-stamp request does not identify the requester, as this information is not validated by the TSA. In situations where the TSA requires the identity of the requesting entity, alternate identification/authentication means have to be used.

7.3. Time-Stamp Protocol (TSP) - RFC 3161 - Response Format

```
TimeStampResp ::= SEQUENCE {
     status                    PKIStatusInfo,
     timeStampToken            TimeStampToken   OPTIONAL  }

PKIStatusInfo ::= SEQUENCE {
     status         PKIStatus,
     statusString PKIFreeText        OPTIONAL,
     failInfo       PKIFailureInfo   OPTIONAL }
```

When the status contains the value zero or one, a TimeStampToken MUST be present. When status contains a value other than zero or one, a TimeStampToken MUST NOT be present. One of the following values MUST be contained in status:

```
PKIStatus ::= INTEGER {
     granted                 (0),
     -- when the PKIStatus contains the value zero a TimeStampToken, as
        requested, is present.
     grantedWithMods         (1),
      -- when the PKIStatus contains the value one a TimeStampToken,
        with modifications, is present.
     rejection               (2),
     waiting                 (3),
     revocationWarning       (4),
      -- this message contains a warning that a revocation is
      -- imminent
     revocationNotification (5)
      -- notification that a revocation has occurred }
```

8. Public-Key Cryptography Standards (PKCS)

	Version	Name	Comments
PKCS#1	2.1	RSA Cryptography Standard	See RFC 3447. Defines the format of RSA encryption.
PKCS#3	1.4	Diffie-Hellman Key Agreement Standard	A cryptographic protocol that allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel.
PKCS#5	2.0	Password-based Encryption Standard	See RFC 2898 and PBKDF2.
PKCS#6	1.5	Extended-Certificate Syntax Standard	Defines extensions to the old v1 X.509 certificate specification. Obsoleted by v3 of the same.
PKCS#7	1.5	Cryptographic Message Syntax Standard	See RFC 2315. Used to sign and/or encrypt messages under a PKI. Used also for certificate dissemination (for instance as a response to a PKCS#10 message). Formed the basis for S/MIME, which is now based on RFC 3852, an updated Cryptographic Message Syntax Standard (CMS).
PKCS#8	1.2	Private-Key Information Syntax Standard	
PKCS#9	2.0	Selected Attribute Types	Defines selected attribute types for use in PKCS #6 extended certificates, PKCS #7 digitally signed messages, PKCS #8 private-key information, and PKCS #10 certificate-signing requests.
PKCS#10	**1.7**	**Certification Request Standard**	**See RFC 2986. Format of messages sent to a certification authority to request certification of a public key. See certificate signing request.**
PKCS#11	2.20	Cryptographic Token Interface (Cryptoki)	An API defining a generic interface to cryptographic tokens (see also Hardware Security Module).
PKCS#12	1.0	Personal Information Exchange Syntax Standard	Defines a file format commonly used to store private keys with accompanying public key certificates, protected with a password-based symmetric key.
PKCS#13	–	Elliptic Curve Cryptography Standard	(Under development)
PKCS#14	–	Pseudo-random Number Generation	(Under development)
PKCS#15	1.1	Cryptographic Token Information Format Standard	Defines a standard allowing users of cryptographic tokens to identify themselves to applications, independent of the application's Cryptoki implementation (PKCS #11) or other API. RSA has relinquished IC-card-related parts of this standard to ISO/IEC 7816-15.[1]

9. Bibliographic references

- What do you need to know about the person with whom you are doing business? (Carl M. Ellison)
- EESSI
- CEN CWA on electronic signatures
- COMMISSION DECISION of 14 July 2003 on the publication of reference numbers of generally recognised standards for electronic signature products in accordance with Directive 1999/93/EC of the European Parliament and of the Council
- ETSI Electronic Signatures and Infrastructures deliverables
- ETSI Standards Making Process
- Réglementations techniques dans le monde des ICP (infrastructures à clé publique) - OLAS
- CPs et CPS de LuxTrust
- Public-Key Cryptography Standards (PKCS) (RSA laboratories)
- PKCS (wikipedia)