

ASN.1: Cryptographic files

Zdeněk Říha





ASN.1 Grammar

- To understand the structure (what is the meaning of particular fields) we need ASN.1 grammar

```
CertificateList ::= SEQUENCE {  
  tbsCertList      TBSCertList,  
  signatureAlgorithm AlgorithmIdentifier,  
  signatureValue   BIT STRING }
```

```
TBSCertList ::= SEQUENCE {  
  version          Version OPTIONAL,  
                  -- if present, MUST be v2  
  signature        AlgorithmIdentifier,  
  issuer           Name,  
  thisUpdate      Time,  
  nextUpdate      Time OPTIONAL,  
  revokedCertificates SEQUENCE OF SEQUENCE {  
    userCertificate CertificateSerialNumber,  
    revocationDate  Time,  
    crlEntryExtensions Extensions OPTIONAL  
                  -- if present, MUST be v2  
  } OPTIONAL,  
  crlExtensions   [0] EXPLICIT Extensions OPTIONAL  
                  -- if present, MUST be v2  
}
```

ASN.1 – RSA keys



```
RSAPublicKey ::= SEQUENCE {
    modulus          INTEGER, -- n
    publicExponent   INTEGER -- e
}

--
-- Representation of RSA private key with information for the CRT algorithm.
--
RSAPrivateKey ::= SEQUENCE {
    version          Version,
    modulus          INTEGER, -- n
    publicExponent   INTEGER, -- e
    privateExponent  INTEGER, -- d
    prime1           INTEGER, -- p
    prime2           INTEGER, -- q
    exponent1        INTEGER, -- d mod (p-1)
    exponent2        INTEGER, -- d mod (q-1)
    coefficient       INTEGER, -- (inverse of q) mod p
    otherPrimeInfos  OtherPrimeInfos OPTIONAL
}
```

 RSA.key

Source:
PKCS#1



ASN.1 – RSA padding

- PKCS#1 v1.5

- $m = 0x00 \parallel 0x01 \parallel 0xFF \dots 0xFF \parallel 0x00 \parallel T$
- Where T is defined as DER encoding of

```
DigestInfo ::= SEQUENCE {  
    digestAlgorithm AlgorithmIdentifier,  
    digest OCTET STRING  
}
```

- In practice:

```
MD2:      (0x)30 20 30 0c 06 08 2a 86 48 86 f7 0d 02 02 05 00 04 10 || H.  
MD5:      (0x)30 20 30 0c 06 08 2a 86 48 86 f7 0d 02 05 05 00 04 10 || H.  
SHA-1:    (0x)30 21 30 09 06 05 2b 0e 03 02 1a 05 00 04 14 || H.  
SHA-256:  (0x)30 31 30 0d 06 09 60 86 48 01 65 03 04 02 01 05 00 04 20 || H.  
SHA-384:  (0x)30 41 30 0d 06 09 60 86 48 01 65 03 04 02 02 05 00 04 30 || H.  
SHA-512:  (0x)30 51 30 0d 06 09 60 86 48 01 65 03 04 02 03 05 00 04 40 || H.
```



ASN.1 – RSA signature

- RSA signature is the number $s = m^d \text{ mod } n$

```
ASN.1 Editor - Opening File: postsignature_tsa_tsu1.der
File View Tools Help
(0,1818) SEQUENCE
+ (4,1538) SEQUENCE
- (1546,13) SEQUENCE
  (1548,9) OBJECT IDENTIFIER : : '1.2.840.113549.1.1.11'
  (1559,0) NULL
  (1561,257) BIT STRING UnusedBits: 0 : '7BA3DA2079DA32BC74B858B5ED5028EC4880D631D09B1758A1304491DBF5DE6A'
File Name: C:\Documents and Settings\Administrator\Plocha\PKI\postsignature_tsa_tsu1.der
Size: 1822 (bytes)
```



ASN.1 – signature OIDs

RSA Encryption ¹	1.2.840.113549.1.1.1
RSASSA-PKCS1_v15 with SHA1	1.2.840.113549.1.1.5
RSASSA-PSS	1.2.840.113549.1.1.10 (PKCS #1 Version 2.1)
RSASSA-PKCS1_v15 with SHA224	1.2.840.113549.1.1.14
RSASSA-PKCS1_v15 with SHA256	1.2.840.113549.1.1.11
RSASSA-PKCS1_v15 with SHA384	1.2.840.113549.1.1.12
RSASSA-PKCS1_v15 with SHA512	1.2.840.113549.1.1.13

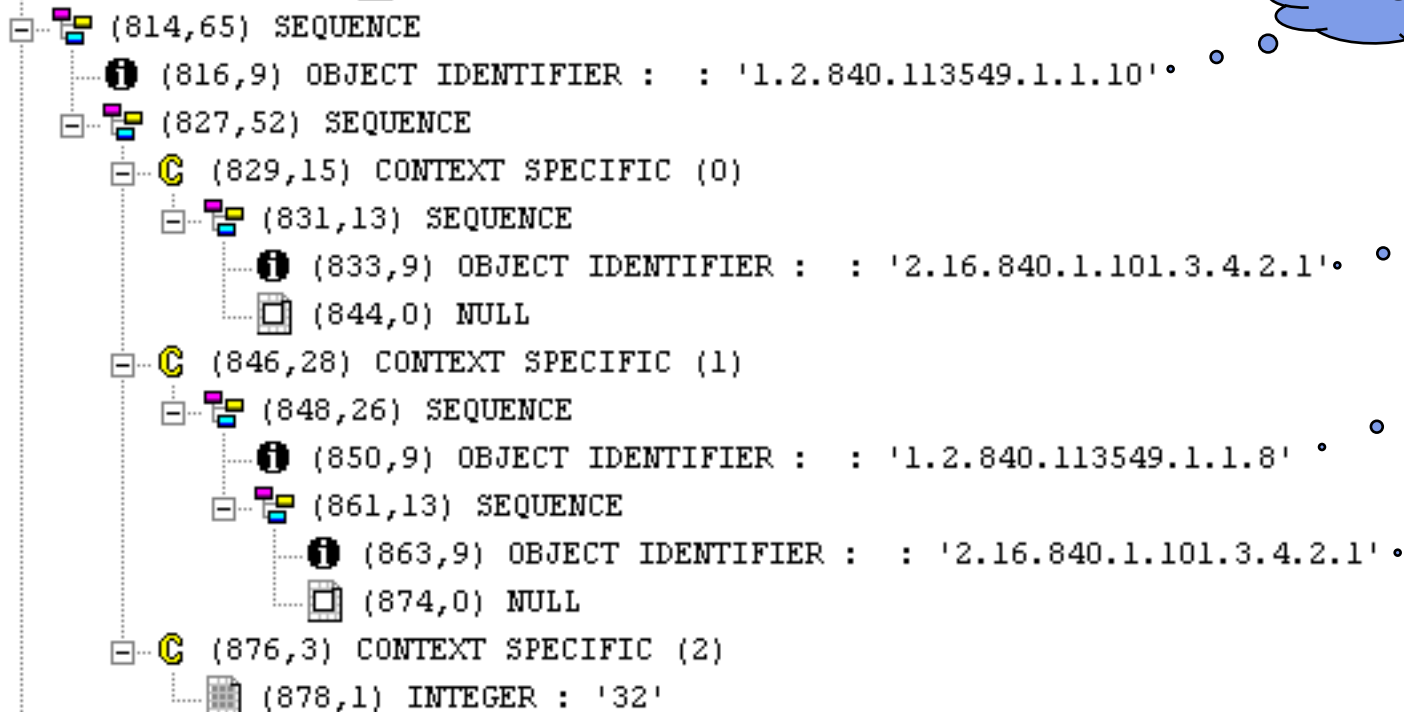
ASN.1 – RSA PSS params



```
RSASSA-PSS-params ::= SEQUENCE {  
    hashAlgorithm      [0] HashAlgorithm      DEFAULT sha1,  
    maskGenAlgorithm   [1] MaskGenAlgorithm   DEFAULT mgf1SHA1,  
    saltLength         [2] INTEGER           DEFAULT 20,  
    trailerField       [3] TrailerField       DEFAULT trailerFieldBC  
}
```

Source:
PKCS#1

```
TrailerField ::= INTEGER { trailerFieldBC(1) }
```



RSASSA-PSS

SHA256

MGF1

SHA256



ASN.1 – DSA keys

```
Dsa-Parms ::= SEQUENCE {  
    p          INTEGER,  
    q          INTEGER,  
    g          INTEGER }
```

Source:
RFC 5480

```
DSAPublicKey ::= INTEGER -- public key, Y
```

DSAPrivateKey is an INTEGER, usually denoted as X

```
ASN1_SEQUENCE_cb(DSAPrivateKey, dsa_cb) = {  
    ASN1_SIMPLE(DSA, version, LONG),  
    ASN1_SIMPLE(DSA, p, BIGNUM),  
    ASN1_SIMPLE(DSA, q, BIGNUM),  
    ASN1_SIMPLE(DSA, g, BIGNUM),  
    ASN1_SIMPLE(DSA, pub_key, BIGNUM),  
    ASN1_SIMPLE(DSA, priv_key, BIGNUM)  
} ASN1_SEQUENCE_END_cb(DSA, DSAPrivateKey)
```

Source:
OpenSSL

ASN.1 Editor - Opening File: ca_dsa_3072.key

```
File View Tools Help
```

(0,1214) SEQUENCE

- (4,1) INTEGER : '0'
- (7,385) INTEGER : '00D1817B0239DFCCA78268BB9B57EFFE70119102A611D6E553'
- (396,21) INTEGER : '00D6422767C29597287C6CF9EAC71BA0B4B864FF51'
- (419,384) INTEGER : '0958FA358A7A0EF8E9B1E1D0A255A25821159130566BFF2F'
- (807,385) INTEGER : '00C4E08EC8CE183F6BC79FEAE6B09456FE4B61C727D83C70'
- (1196,20) INTEGER : '3AD05ADEFD96EA52CC915E0EBE411B9B94ADD3DA'

 DSA.key



ASN.1 – DSA signature

```
Dss-Sig-Value ::= SEQUENCE {  
    r      INTEGER,  
    s      INTEGER }
```

Source:
RFC 5480

The screenshot shows the ASN.1 Editor interface for the file 'ca_dsa_3072_sha1.crt'. The main window displays a tree view of the ASN.1 structure:

- (0,1681) SEQUENCE (highlighted)
 - (4,1617) SEQUENCE
 - (1625,9) SEQUENCE
 - (1627,7) OBJECT IDENTIFIER : dsaWithShal : '1.2.840.10040.4.3'
 - (1636,47) BIT STRING UnusedBits: 0
 - (1639,44) SEQUENCE
 - (1641,20) INTEGER : '64CA41FEA8CBA7E9282D215BC60BF4FECD198858'
 - (1663,20) INTEGER : '1B78E8B76423099D9D897F59066A813E93C3A7A1'

The status bar at the bottom indicates: File Name: C:\Documents and Settings\Administrator\Plocha\PKI\gen_sod\keys\ca_dsa Size: 1685 (bytes)



ASN.1 – DSA - OIDs

```
-- DSA with SHA-1
-- Parameters are ABSENT

id-dsa-with-sha1 OBJECT IDENTIFIER ::= {
    iso(1) member-body(2) us(840) x9-57(10040) x9algorithm(4) 3 }

-- DSA with SHA-224
-- Parameters are ABSENT

id-dsa-with-sha224 OBJECT IDENTIFIER ::= {
    joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101)
    csor(3) algorithms(4) id-dsa-with-sha2(3) 1 }

-- DSA with SHA-256
-- Parameters are ABSENT

id-dsa-with-sha256 OBJECT IDENTIFIER ::= {
    joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101)
    csor(3) algorithms(4) id-dsa-with-sha2(3) 2 }
```



ASN.1 – ECDSA keys

```
ECParameters ::= SEQUENCE {  
    version      INTEGER{ecpVer1(1)} (ecpVer1),  
    fieldID      FieldID{{FieldTypes}},  
    curve        Curve,  
    base         ECPoint,  
    order        INTEGER,  
    cofactor    INTEGER OPTIONAL,  
    ...  
}
```

```
Curve ::= SEQUENCE {  
    a      FieldElement,  
    b      FieldElement,  
    seed   BIT STRING OPTIONAL  
}
```

```
ECPoint ::= OCTET STRING -- Elliptic curve point
```

```
ECPrivateKey{CURVES:IOSet} ::= SEQUENCE {  
    version      INTEGER { ecPrivkeyVer1(1) } ( ecPrivkeyVer1 ),  
    privateKey   OCTET STRING, • • •  
    parameters   [0] Parameters{{IOSet}} OPTIONAL, INTEGER  
    publicKey    [1] BIT STRING OPTIONAL  
}
```

```
SubjectPublicKeyInfo ::= SEQUENCE {  
    algorithm      AlgorithmIdentifier{{ECPKAlgorithms}},  
    subjectPublicKey BIT STRING • • • ECPoint  
}
```


ASN.1 – ECDSA signatures



```
ec-signature-value ::= SEQUENCE {  
    r    INTEGER,  
    s    INTEGER  
}
```

Source:
RFC 5480

1.2.840.10045.4.1 - ecdsa-with-SHA1

The screenshot shows the ASN.1 Editor interface for the file 'Switzerland.crt'. The main window displays a tree view of the certificate's structure. The root is a SEQUENCE (0,1059) containing several elements. The element at (948,7) is an OBJECT IDENTIFIER with the value '1.2.840.10045.4.1', which is highlighted in blue. Below it is a BIT STRING (957,104) with UnusedBits: 0. The next element is a SEQUENCE (960,101) containing two INTEGER values: one at (962,49) with the value '00FEEB445183C58A9055C8EC17926AB1135D7234F540A4486951E73967FC60C2D6D86B6230FF081ED34FEC3251FCDE5C4D' and another at (1013,48) with the value '0A555CA2359A949C0F68C56BF7B72C1AD77108825B8053783A32F00BF685A2785EEECB5A1673A6ED6577A1B59560C4A4'.

File Name: C:\zriha\data\CSCA_certificates\Switzerland.crt Size: 1063 (bytes)

ASN.1 – ECDSA signature OLD



ECDSA with SHA1	1.2.840.10045.1 (ANSI X9.62)
ECDSA with SHA1	1.2.840.10045.4.1 (ANSI X9.62)
ECDSA with SHA224	1.2.840.10045.4.3.1 (ANSI X9.62)
ECDSA with SHA256	1.2.840.10045.4.3.2 (ANSI X9.62)
ECDSA with SHA384	1.2.840.10045.4.3.3 (ANSI X9.62)
ECDSA with SHA512	1.2.840.10045.4.3.4 (ANSI X9.62)
ECDSA with SHA1	0.4.0.127.0.7.4.1.1 (BSI)
ECDSA with SHA224	0.4.0.127.0.7.4.1.2 (BSI)
ECDSA with SHA256	0.4.0.127.0.7.4.1.3 (BSI)
ECDSA with SHA384	0.4.0.127.0.7.4.1.4 (BSI)
ECDSA with SHA512	0.4.0.127.0.7.4.1.5 (BSI)



ASN.1 - certificates

```
Certificate ::= SEQUENCE {  
    tbsCertificate      TBSCertificate,  
    signatureAlgorithm AlgorithmIdentifier,  
    signatureValue     BIT STRING }
```

```
TBSCertificate ::= SEQUENCE {  
    version            [0] EXPLICIT Version DEFAULT v1,  
    serialNumber      CertificateSerialNumber,  
    signature         AlgorithmIdentifier,  
    issuer            Name,  
    validity          Validity,  
    subject           Name,  
    subjectPublicKeyInfo SubjectPublicKeyInfo,  
    issuerUniqueID   [1] IMPLICIT UniqueIdentifier OPTIONAL,  
                    -- If present, version MUST be v2 or v3  
    subjectUniqueID [2] IMPLICIT UniqueIdentifier OPTIONAL,  
                    -- If present, version MUST be v2 or v3  
    extensions       [3] EXPLICIT Extensions OPTIONAL  
                    -- If present, version MUST be v3  
}
```

```
Version ::= INTEGER { v1(0), v2(1), v3(2) }  
CertificateSerialNumber ::= INTEGER
```

Source:
RFC 5280



ASN.1 – certificates - pubkey

```
SubjectPublicKeyInfo ::= SEQUENCE {  
    algorithm      AlgorithmIdentifier,  
    subjectPublicKey BIT STRING }
```

Source:
RFC 5280

```
AlgorithmIdentifier ::= SEQUENCE {  
    algorithm      OBJECT IDENTIFIER,  
    parameters    ANY DEFINED BY algorithm OPTIONAL }
```

```
(293,418) SEQUENCE  
├── (297,13) SEQUENCE  
│   ├── (299,9) OBJECT IDENTIFIER : rsaEncryption : '1.2.840.113549.1.1.1'  
│   └── (310,0) NULL  
└── (312,399) BIT STRING UnusedBits: 0  
    ├── (317,394) SEQUENCE  
    │   ├── (321,385) INTEGER : '00A4A6BEDFA5969EE5647114F3E610CAB822C7B21098E6156CE073CCA6DA511E8F9AB6A1BD1DA64ED6B05'  
    │   └── (710,3) INTEGER : '65537'
```




ASN.1 – certificates - times

```
Validity ::= SEQUENCE {  
    notBefore      Time,  
    notAfter       Time }
```

Source:
RFC 5280

```
Time ::= CHOICE {  
    utcTime          UTCTime,  
    generalTime      GeneralizedTime }
```

- Until 2049: UTCTime
 - YYMMDDHHMMSSZ
- From 2050: GeneralizedTime
 - YYYYMMDDHHMMSSZ

 CSCA_CZE.crt

```
┌─── (172,30) SEQUENCE  
│   ├── (174,13) UTC TIME : '090113000000Z'  
│   └── (189,13) UTC TIME : '240413000000Z'
```



ASN.1 – certificates - names

```
Name ::= CHOICE { -- only one possibility for now --  
    rdnSequence  RDNSequence }
```

```
RDNSequence ::= SEQUENCE OF RelativeDistinguishedName
```

```
RelativeDistinguishedName ::=  
    SET SIZE (1..MAX) OF AttributeTypeAndValue
```

```
AttributeTypeAndValue ::= SEQUENCE {  
    type      AttributeType,  
    value     AttributeValue }
```

```
AttributeType ::= OBJECT IDENTIFIER
```

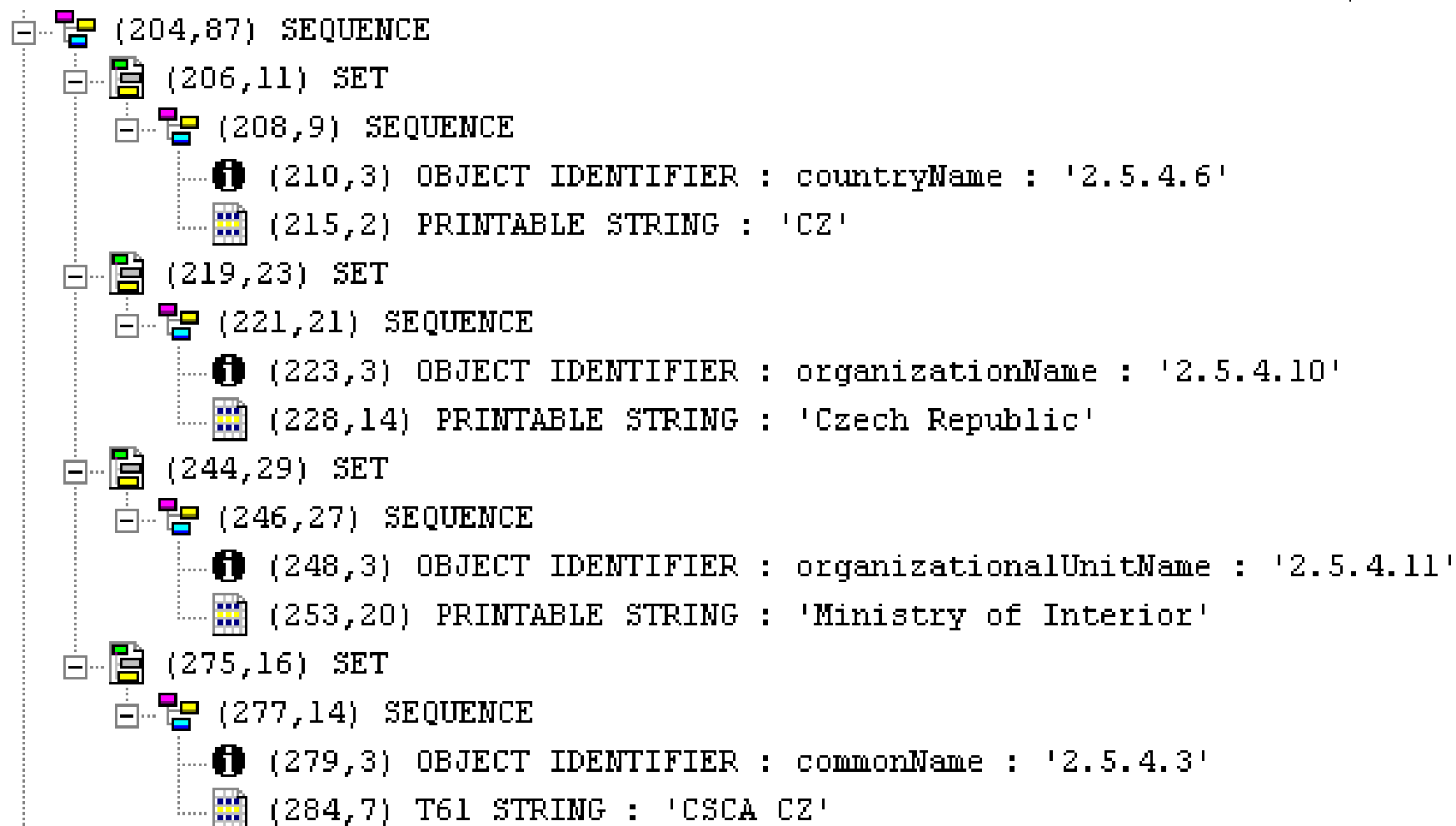
```
AttributeValue ::= ANY -- DEFINED BY AttributeType
```

```
DirectoryString ::= CHOICE {  
    teletexString      TeletexString (SIZE (1..MAX)),  
    printableString   PrintableString (SIZE (1..MAX)),  
    universalString    UniversalString (SIZE (1..MAX)),  
    utf8String         UTF8String (SIZE (1..MAX)),  
    bmpString          BMPString (SIZE (1..MAX)) }
```

Source:
RFC 5280



ASN.1 – certificate - names



ASN.1 – certificate - names



```
commonName ATTRIBUTE ::= {
    SUBTYPE OF      name
    WITH SYNTAX     DirectoryString {ub-common-name}
    ID              id-at-commonName }
```

```
DirectoryString { INTEGER : maxSize } ::= CHOICE {
    teletexString TeletexString (SIZE (1..maxSize)),
    printableString PrintableString (SIZE (1..maxSize)),
    bmpString BMPString (SIZE (1..maxSize)),
    universalString UniversalString (SIZE (1..maxSize)),
    uTF8String UTF8String (SIZE (1..maxSize)) }
```

```
countryName ATTRIBUTE ::= {
    SUBTYPE OF      name
    WITH SYNTAX     CountryName
    SINGLE VALUE    TRUE
    ID              id-at-countryName }
```

```
CountryName ::= PrintableString (SIZE(2))
```

```
-- id-at-objectClass
-- id-at-aliasedEntryName
-- id-at-encryptedAliasedEntryName
id-at-knowledgeInformation
id-at-commonName
-- id-at-encryptedCommonName
id-at-surname
-- id-at-encryptedSurname
id-at-serialNumber
-- id-at-encryptedSerialNumber
id-at-countryName
```

```
OBJECT IDENTIFIER ::= {id-at 0}
OBJECT IDENTIFIER ::= {id-at 1}
OBJECT IDENTIFIER ::= {id-at 1 2}
OBJECT IDENTIFIER ::= {id-at 2}
OBJECT IDENTIFIER ::= {id-at 3}
OBJECT IDENTIFIER ::= {id-at 3 2}
OBJECT IDENTIFIER ::= {id-at 4}
OBJECT IDENTIFIER ::= {id-at 4 2}
OBJECT IDENTIFIER ::= {id-at 5}
OBJECT IDENTIFIER ::= {id-at 5 2}
OBJECT IDENTIFIER ::= {id-at 6}
```

Source:
ITU-T X.520

ASN.1 – certificate - names



id-at-localityName
-- *id-at-encryptedLocalityName*
id-at-collectiveLocalityName
-- *id-at-encryptedCollectiveLocalityName*
id-at-stateOrProvinceName
-- *id-at-encryptedStateOrProvinceName*
id-at-collectiveStateOrProvinceName
-- *id-at-encryptedCollectiveStateOrProvinceName*
id-at-streetAddress
-- *id-at-encryptedStreetAddress*
id-at-collectiveStreetAddress
-- *id-at-encryptedCollectiveStreetAddress*
id-at-organizationName
-- *id-at-encryptedOrganizationName*
id-at-collectiveOrganizationName
-- *id-at-encryptedCollectiveOrganizationName*
id-at-organizationalUnitName
-- *id-at-encryptedOrganizationalUnitName*
id-at-collectiveOrganizationalUnitName
-- *id-at-encryptedCollectiveOrganizationalUnitName*
id-at-title
-- *id-at-encryptedTitle*
id-at-description
-- *id-at-encryptedDescription*
id-at-searchGuide
-- *id-at-encryptedSearchGuide*
id-at-businessCategory
-- *id-at-encryptedBusinessCategory*
id-at-postalAddress
-- *id-at-encryptedPostalAddress*
id-at-collectivePostalAddress
-- *id-at-encryptedCollectivePostalAddress*
id-at-postalCode
-- *id-at-encryptedPostalCode*
id-at-collectivePostalCode
-- *id-at-encryptedCollectivePostalCode*

OBJECT IDENTIFIER ::= {id-at 7}
OBJECT IDENTIFIER ::= {id-at 7 2}
OBJECT IDENTIFIER ::= {id-at 7 1}
OBJECT IDENTIFIER ::= {id-at 7 1 2}
OBJECT IDENTIFIER ::= {id-at 8}
OBJECT IDENTIFIER ::= {id-at 8 2}
OBJECT IDENTIFIER ::= {id-at 8 1}
OBJECT IDENTIFIER ::= {id-at 8 1 2}
OBJECT IDENTIFIER ::= {id-at 9}
OBJECT IDENTIFIER ::= {id-at 9 2}
OBJECT IDENTIFIER ::= {id-at 9 1}
OBJECT IDENTIFIER ::= {id-at 9 1 2}
OBJECT IDENTIFIER ::= {id-at 10}
OBJECT IDENTIFIER ::= {id-at 10 2}
OBJECT IDENTIFIER ::= {id-at 10 1}
OBJECT IDENTIFIER ::= {id-at 10 1 2}
OBJECT IDENTIFIER ::= {id-at 11}
OBJECT IDENTIFIER ::= {id-at 11 2}
OBJECT IDENTIFIER ::= {id-at 11 1}
OBJECT IDENTIFIER ::= {id-at 11 1 2}
OBJECT IDENTIFIER ::= {id-at 12}
OBJECT IDENTIFIER ::= {id-at 12 2}
OBJECT IDENTIFIER ::= {id-at 13}
OBJECT IDENTIFIER ::= {id-at 13 2}
OBJECT IDENTIFIER ::= {id-at 14}
OBJECT IDENTIFIER ::= {id-at 14 2}
OBJECT IDENTIFIER ::= {id-at 15}
OBJECT IDENTIFIER ::= {id-at 15 2}
OBJECT IDENTIFIER ::= {id-at 16}
OBJECT IDENTIFIER ::= {id-at 16 2}
OBJECT IDENTIFIER ::= {id-at 16 1}
OBJECT IDENTIFIER ::= {id-at 16 1 2}
OBJECT IDENTIFIER ::= {id-at 17}
OBJECT IDENTIFIER ::= {id-at 17 2}
OBJECT IDENTIFIER ::= {id-at 17 1}
OBJECT IDENTIFIER ::= {id-at 17 1 2}

Source:
ITU-T X.520



Certificate profiles

- For particular areas/purposes there exist certificate profiles which prescribe what kind of attributes will be used in Names
- E.g. for electronic passports ICAO Doc. 9303 states:

The following Attributes SHOULD be used:

- country (country codes SHALL follow the format of two letter country codes, specified in [R16], *ISO/IEC 3166, Codes for the representation of names of countries and their subdivisions — 1997.*).
- organization;
- organizational-unit;
- common name.

Additionally some countries MAY use:

- serial number.

Source:
ICAO Doc. 9303



ASN.1 – certificates – v3

```
UniqueIdentifier ::= BIT STRING
```

```
Extensions ::= SEQUENCE SIZE (1..MAX) OF Extension
```

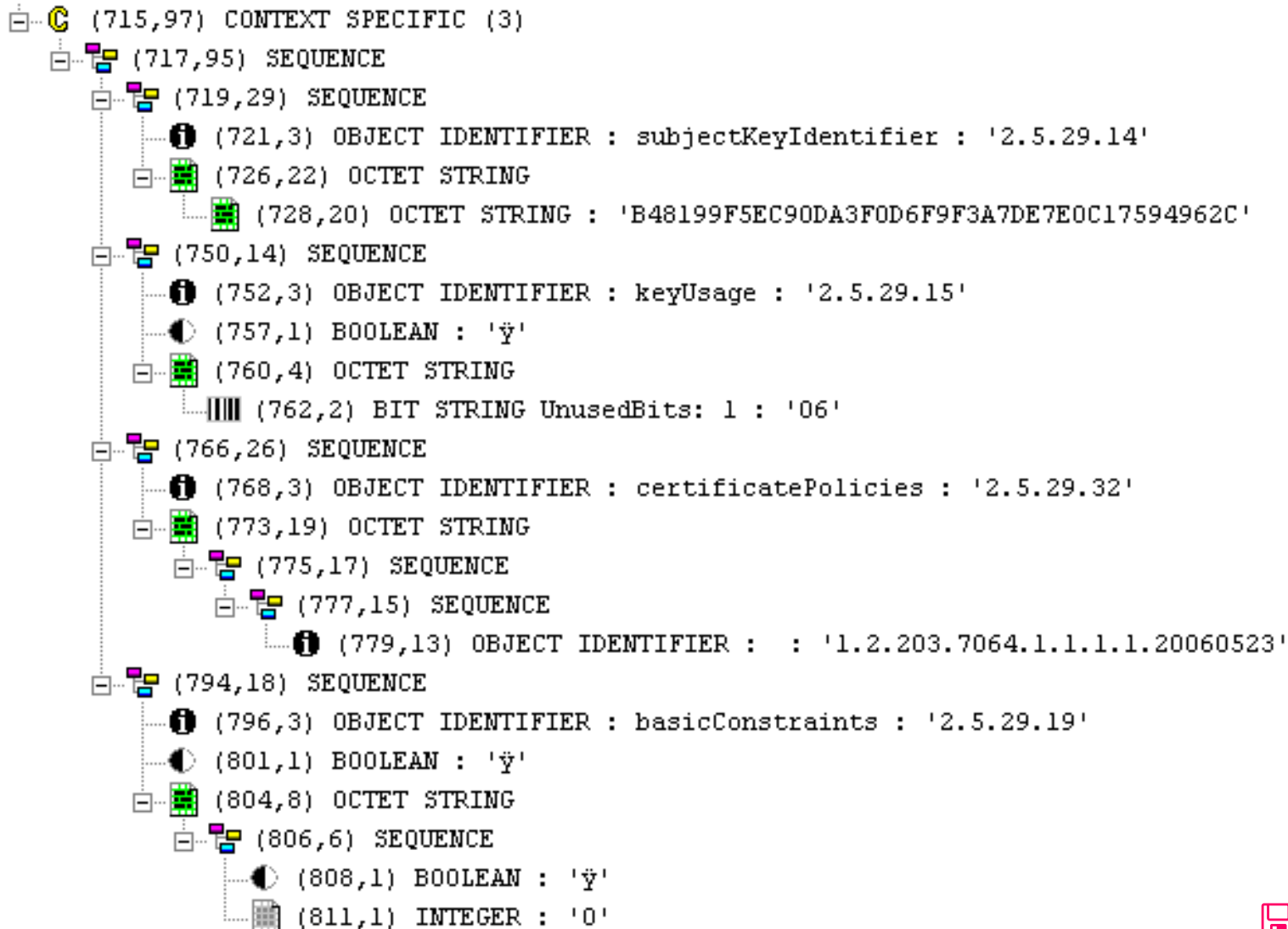
```
Extension ::= SEQUENCE {  
    extnID      OBJECT IDENTIFIER,  
    critical    BOOLEAN DEFAULT FALSE,  
    extnValue   OCTET STRING  
                -- contains the DER encoding of an ASN.1 value  
                -- corresponding to the extension type identified  
                -- by extnID  
}
```

Source:
RFC 5280

- Critical x non-critical extensions



ASN.1 – certs – extensions





X509v3 cert extensions

- Authority Key Identifier
 - Identification of the issuing CA
 - Non critical

```
id-ce-authorityKeyIdentifier OBJECT IDENTIFIER ::= { id-ce 35 }
```

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

```
KeyIdentifier ::= OCTET STRING
```

- Similarly “Subject Key Identifier”

Source:
RFC 5280



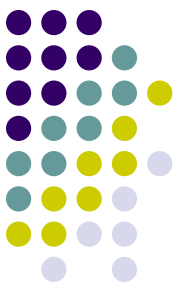
X509v3 cert extensions

- Key Usage
 - Restrictions of the use of the key

```
id-ce-keyUsage OBJECT IDENTIFIER ::= { id-ce 15 }
```

```
KeyUsage ::= BIT STRING {  
    digitalSignature          (0),  
    nonRepudiation           (1), -- recent editions of X.509 have  
                                -- renamed this bit to contentCommitment  
    keyEncipherment          (2),  
    dataEncipherment         (3),  
    keyAgreement             (4),  
    keyCertSign              (5),  
    cRLSign                  (6),  
    encipherOnly             (7),  
    decipherOnly             (8) }
```

Source:
RFC 5280



X509v3 cert extensions

- Extended Key Usage
 - Purposes of the certified key

```
id-ce-extKeyUsage OBJECT IDENTIFIER ::= { id-ce 37 }
ExtKeyUsageSyntax ::= SEQUENCE SIZE (1..MAX) OF KeyPurposeId
KeyPurposeId ::= OBJECT IDENTIFIER
anyExtendedKeyUsage OBJECT IDENTIFIER ::= { id-ce-extKeyUsage 0 }
```

```
id-kp OBJECT IDENTIFIER ::= { id-pkix 3 }
id-kp-serverAuth OBJECT IDENTIFIER ::= { id-kp 1 }
id-kp-clientAuth OBJECT IDENTIFIER ::= { id-kp 2 }
id-kp-codeSigning OBJECT IDENTIFIER ::= { id-kp 3 }
id-kp-emailProtection OBJECT IDENTIFIER ::= { id-kp 4 }
id-kp-timeStamping OBJECT IDENTIFIER ::= { id-kp 8 }
id-kp-OCSPSigning OBJECT IDENTIFIER ::= { id-kp 9 }
```

X509v3 cert extensions



```
id-ce-certificatePolicies OBJECT IDENTIFIER ::= { id-ce 32 }

anyPolicy OBJECT IDENTIFIER ::= { id-ce-certificatePolicies 0 }

certificatePolicies ::= SEQUENCE SIZE (1..MAX) OF PolicyInformation

PolicyInformation ::= SEQUENCE {
    policyIdentifier      CertPolicyId,
    policyQualifiers     SEQUENCE SIZE (1..MAX) OF
                        PolicyQualifierInfo OPTIONAL }

CertPolicyId ::= OBJECT IDENTIFIER

PolicyQualifierInfo ::= SEQUENCE {
    policyQualifierId    PolicyQualifierId,
    qualifier            ANY DEFINED BY policyQualifierId }

-- policyQualifierIds for Internet policy qualifiers

id-qt          OBJECT IDENTIFIER ::= { id-pkix 2 }
id-qt-cps      OBJECT IDENTIFIER ::= { id-qt 1 }
id-qt-unotice  OBJECT IDENTIFIER ::= { id-qt 2 }

PolicyQualifierId ::= OBJECT IDENTIFIER ( id-qt-cps | id-qt-unotice )

Qualifier ::= CHOICE {
    cpsSuri          CPSuri,
    userNotice      UserNotice }

CPSuri ::= IA5String

UserNotice ::= SEQUENCE {
    noticeRef        NoticeReference OPTIONAL,
    explicitText     DisplayText OPTIONAL }

NoticeReference ::= SEQUENCE {
    organization     DisplayText,
    noticeNumbers    SEQUENCE OF INTEGER }

DisplayText ::= CHOICE {
    ia5String        IA5String          (SIZE (1..200)),
    visibleString    VisibleString      (SIZE (1..200)),
    bmpString        BMPString          (SIZE (1..200)),
    utf8String       UTF8String         (SIZE (1..200)) }
```

- Certificate Policies
 - Policy relevant for the issue and use of the certificate
 - Preferably only an OID

Source:
RFC 5280



X509v3 cert extensions

- Subject Alternative Name
- Issuer Alternative Name
- “Internet style identities”
 - Email
 - DNS name
 - IP address
 - URL
- Must be verified by CA



X509v3 cert extensions

- Basic Constraints
- Is Subject a CA?
- Max. length/depth of the certificate chain/path
 - A pathLenConstraint of zero indicates that no non-self-issued intermediate CA certificates may follow in a valid certification path.

```
id-ce-basicConstraints OBJECT IDENTIFIER ::= { id-ce 19 }
```

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



X509v3 cert extensions

- Name Constraints
- Only for CA certificates
- “indicates a name space within which all subject names in subsequent certificates in a certification path MUST be located”

```
id-ce-nameConstraints OBJECT IDENTIFIER ::= { id-ce 30 }

NameConstraints ::= SEQUENCE {
    permittedSubtrees      [0]      GeneralSubtrees OPTIONAL,
    excludedSubtrees      [1]      GeneralSubtrees OPTIONAL }

GeneralSubtrees ::= SEQUENCE SIZE (1..MAX) OF GeneralSubtree

GeneralSubtree ::= SEQUENCE {
    base                    GeneralName,
    minimum                 [0]      BaseDistance DEFAULT 0,
    maximum                 [1]      BaseDistance OPTIONAL }

BaseDistance ::= INTEGER (0..MAX)
```

Source:
RFC 5280



X509v3 cert extensions

- Policy Constraints
- Must be critical
- For CA certificates
- Constraints path validation
 - Prohibit policy mapping (or)
 - Require acceptable policy OID in each certificate

```
id-ce-policyConstraints OBJECT IDENTIFIER ::= { id-ce 36 }
```

```
PolicyConstraints ::= SEQUENCE {  
    requireExplicitPolicy          [0] SkipCerts OPTIONAL,  
    inhibitPolicyMapping           [1] SkipCerts OPTIONAL }
```

```
SkipCerts ::= INTEGER (0..MAX)
```

Source:
RFC 5280



X509v3 cert extensions

- CRL Distribution Points
- How to obtain CRL

```
id-ce-cRLDistributionPoints OBJECT IDENTIFIER ::= { id-ce 31 }
```

```
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) }
```

Source:
RFC 5280



ASN.1 – certificate request

```
CertificationRequest ::= SEQUENCE {  
  certificationRequestInfo CertificationRequestInfo,  
  signatureAlgorithm AlgorithmIdentifier,  
  signature BIT STRING  
}
```

```
CertificationRequestInfo ::= SEQUENCE {  
  version INTEGER { v1(0) },  
  subject Name,  
  subjectPKInfo SubjectPublicKeyInfo,  
  attributes [0] Attributes  
}
```

```
Attributes ::= SET OF Attribute
```

```
Attribute ::= SEQUENCE {  
  type ATTRIBUTE.&id({IOSet}),  
  values SET SIZE(1..MAX) OF ATTRIBUTE.&Type({IOSet}@type)  
}
```

Source:
RFC 5280

ASN.1 - CRL



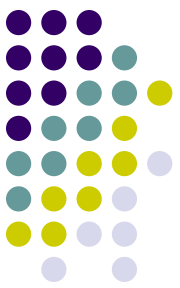
```
CertificateList ::= SEQUENCE {
    tbsCertList      TBSCertList,
    signatureAlgorithm AlgorithmIdentifier,
    signatureValue   BIT STRING }

```

```
TBSCertList ::= SEQUENCE {
    version          Version OPTIONAL,
                    -- if present, MUST be v2

    signature        AlgorithmIdentifier,
    issuer           Name,
    thisUpdate       Time,
    nextUpdate       Time OPTIONAL,
    revokedCertificates SEQUENCE OF SEQUENCE {
        userCertificate CertificateSerialNumber,
        revocationDate Time,
        crlEntryExtensions Extensions OPTIONAL
                    -- if present, version MUST be v2
    } OPTIONAL,
    crlExtensions   [0] EXPLICIT Extensions OPTIONAL
                    -- if present, version MUST be v2
}

```



ASN.1 – PKCS#7 / CMS

```
ContentInfo ::= SEQUENCE {  
    contentType ContentType,  
    content [0] EXPLICIT ANY DEFINED BY contentType }
```

```
ContentType ::= OBJECT IDENTIFIER
```

```
SignedData ::= SEQUENCE {  
    version CMSVersion,  
    digestAlgorithms DigestAlgorithmIdentifiers,  
    encapContentInfo EncapsulatedContentInfo,  
    certificates [0] IMPLICIT CertificateSet OPTIONAL,  
    crls [1] IMPLICIT RevocationInfoChoices OPTIONAL,  
    signerInfos SignerInfos }
```

```
DigestAlgorithmIdentifiers ::= SET OF DigestAlgorithmIdentifier
```

```
EncapsulatedContentInfo ::= SEQUENCE {  
    eContentType ContentType,  
    eContent [0] EXPLICIT OCTET STRING OPTIONAL }
```

```
SignerInfos ::= SET OF SignerInfo
```



ASN.1 - PKCS#7 / CMS

```
SignerInfo ::= SEQUENCE {  
    version CMSVersion,  
    sid SignerIdentifier,  
    digestAlgorithm DigestAlgorithmIdentifier,  
    signedAttrs [0] IMPLICIT SignedAttributes OPTIONAL,  
    signatureAlgorithm SignatureAlgorithmIdentifier,  
    signature SignatureValue,  
    unsignedAttrs [1] IMPLICIT UnsignedAttributes OPTIONAL }
```

```
SignerIdentifier ::= CHOICE {  
    issuerAndSerialNumber IssuerAndSerialNumber,  
    subjectKeyIdentifier [0] SubjectKeyIdentifier }
```

```
SignedAttributes ::= SET SIZE (1..MAX) OF Attribute
```

```
UnsignedAttributes ::= SET SIZE (1..MAX) OF Attribute
```

```
Attribute ::= SEQUENCE {  
    attrType OBJECT IDENTIFIER,  
    attrValues SET OF AttributeValue }
```

```
AttributeValue ::= ANY
```

```
SignatureValue ::= OCTET STRING
```

PKCS#7 Sample

France.p7s



ASN.1 Editor - Opening File: France.p7s

File View Tools Help

```
(39,139) SEQUENCE
├── (181,967) CONTEXT SPECIFIC (0)
└── (1152,453) SET
    ├── (1156,449) SEQUENCE
    │   ├── (1160,1) INTEGER : '1'
    │   ├── (1163,63) SEQUENCE
    │   │   ├── (1165,58) SEQUENCE
    │   │   │   ├── (1167,43) SET
    │   │   │   │   ├── (1169,41) SEQUENCE
    │   │   │   │   │   ├── (1171,3) OBJECT IDENTIFIER : commonName : '2.5.4.3'
    │   │   │   │   │   ├── (1176,34) PRINTABLE STRING : 'Country Signing CA FRA RSA3072SHA1'
    │   │   │   │   │   └── (1212,11) SET
    │   │   │   │   │       ├── (1214,9) SEQUENCE
    │   │   │   │   │       │   ├── (1216,3) OBJECT IDENTIFIER : countryName : '2.5.4.6'
    │   │   │   │   │       │   └── (1221,2) PRINTABLE STRING : 'fr'
    │   │   │   │   └── (1225,1) INTEGER : '2'
    │   │   └── (1228,9) SEQUENCE
    │   │       ├── (1230,5) OBJECT IDENTIFIER : sha1 : '1.3.14.3.2.26'
    │   │       └── (1237,0) NULL
    │   └── (1239,93) CONTEXT SPECIFIC (0)
    │       ├── (1241,24) SEQUENCE
    │       │   ├── (1243,9) OBJECT IDENTIFIER : contentType : '1.2.840.113549.1.9.3'
    │       │   └── (1254,11) SET
    │       │       ├── (1256,9) OBJECT IDENTIFIER : data : '1.2.840.113549.1.7.1'
    │       │       └── (1267,28) SEQUENCE
    │       │           ├── (1269,9) OBJECT IDENTIFIER : signingTime : '1.2.840.113549.1.9.5'
    │       │           └── (1280,15) SET
    │       │               ├── (1282,13) UTC TIME : '061204101915Z'
    │       │               └── (1297,35) SEQUENCE
    │       │                   ├── (1299,9) OBJECT IDENTIFIER : messageDigest : '1.2.840.113549.1.9.4'
    │       │                   └── (1310,22) SET
    │       │                       ├── (1312,20) OCTET STRING : 'F65EC9C78EA67FDD4DF868DC4BA5FFE4F025DA18'
    │       └── (1334,13) SEQUENCE
    │           ├── (1336,9) OBJECT IDENTIFIER : : '1.2.840.113549.1.1.10'
    │           └── (1347,0) SEQUENCE
    │               └── (1349,256) OCTET STRING : '3AA6264B6731CCC3CF0D1CCB424830A03F403D7E3D842F51F9034EBF7FA9E63379029A8F36E0AE5829391F6343E0D84C858'
```

File Name: C:\Documents and Settings\Administrator\plocha\pki_files\France.p7s

Size: 1609 (bytes)



ASN.1 – PKCS#8

```
-- Private-key information syntax
```

```
PrivateKeyInfo ::= SEQUENCE {  
    version Version,  
    privateKeyAlgorithm AlgorithmIdentifier,  
    privateKey PrivateKey,  
    attributes [0] Attributes OPTIONAL }
```

```
Version ::= INTEGER {v1(0)} (v1,...)
```

```
PrivateKey ::= OCTET STRING
```

```
Attributes ::= SET OF Attribute
```

```
-- Encrypted private-key information syntax
```

```
EncryptedPrivateKeyInfo ::= SEQUENCE {  
    encryptionAlgorithm AlgorithmIdentifier,  
    encryptedData EncryptedData  
}
```

```
EncryptedData ::= OCTET STRING
```