

CMS - Signed Data

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

DigestAlgorithmIdentifiers ::= SET OF DigestAlgorithmIdentifier

SignerInfos ::= SET OF SignerInfo

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

ContentType ::= OBJECT IDENTIFIER

SignerInfo ::= SEQUENCE {
    version CMSVersion,
    sid SignerIdentifier,
    digestAlgorithm DigestAlgorithmIdentifier,
    signedAttrs [0] IMPLICIT SignedAttributes OPTIONAL,
    signatureAlgorithm SignatureAlgorithmIdentifier,
    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
```

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

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

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

UniqueIdentifier ::= BIT STRING

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

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

Extension ::= SEQUENCE {
    extnID             OBJECT IDENTIFIER,
    critical           BOOLEAN DEFAULT FALSE,
    extnValue          OCTET STRING }

Name ::= CHOICE {
    RDNSequence }
RDNSequence ::= SEQUENCE OF RelativeDistinguishedName
RelativeDistinguishedName ::=
    SET OF AttributeTypeAndValue

AttributeTypeAndValue ::= SEQUENCE {
    type              AttributeType,
    value             AttributeValue }
AttributeType ::= OBJECT IDENTIFIER
AttributeValue ::= ANY DEFINED BY AttributeType
```

PKCS#1 - RSA

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

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
}
```

PKCS#3 - DH:

```
DHParameter ::= SEQUENCE {
    prime INTEGER, -- p
    base  INTEGER, -- g
    privateValueLength INTEGER OPTIONAL }
}
```

PKCS#8 - private keys:

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

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

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

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