付録２－２ 相互認証証明書等の方式（ASN.1構造とオブジェクト識別子）

１　Explicitly Tagged Module

```
MOJCrossAndLinkCertExplicit { 1 2 392 100300 1 4 51 }
DEFINITIONS EXPLICIT TAGS ::=
BEGIN

-- EXPORTS ALL --

IMPORTS
        authorityKeyIdentifier, subjectKeyIdentifier, keyUsage,
        certificatePolicies, policyMappings,
        basicConstraints, authorityInfoAccess, crlDistributionPoints
        FROM MOJCrossAndLinkCertImplicit { 1 2 392 100300 1 4 52 };


Certificate  ::=  SIGNED { TBSCertificate }

TBSCertificate  ::=  SEQUENCE {
   version                [0]   Version,
   serialNumber                 CertificateSerialNumber,
   signature                    AlgorithmIdentifier,
   issuer                       Name,
   validity                     Validity,
   subject                      Name,
   subjectPublicKeyInfo         SubjectPublicKeyInfo,
   extensions             [3]   Extensions  }

Version  ::=  INTEGER { v3(2) }

CertificateSerialNumber ::= INTEGER

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

Time ::= CHOICE {
        utcTime         UTCTime
}

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

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

```
Extension ::= SEQUENCE {
   extnId            EXTENSION.&id ({ExtensionSet}),
   critical          BOOLEAN DEFAULT FALSE,
   extnValue         OCTET STRING }

ExtensionSet  EXTENSION  ::= { authorityKeyIdentifier |
                               subjectKeyIdentifier |
                               keyUsage |
                               certificatePolicies |
                               policyMappings |
                               basicConstraints |
                               authorityInfoAccess |
                               crlDistributionPoints }

EXTENSION ::= CLASS {
   &id          OBJECT IDENTIFIER UNIQUE,
   &ExtnType }
WITH SYNTAX  {
   SYNTAX               &ExtnType
   IDENTIFIED BY        &id }

SIGNED { ToBeSigned } ::= SEQUENCE {
     toBeSigned  ToBeSigned,
     algorithm   AlgorithmIdentifier,
     signature   BIT STRING
}

AlgorithmIdentifier  ::=  SEQUENCE {
   algorithm            ALGORITHM-ID.&id({SupportedAlgorithms}),
   parameters           ALGORITHM-ID.&Type({SupportedAlgorithms}
                                          { @algorithm}) OPTIONAL }

ALGORITHM-ID ::= CLASS {
     &id    OBJECT IDENTIFIER UNIQUE,
     &Type  OPTIONAL
}
WITH SYNTAX { OID &id [PARMS &Type] }

SupportedAlgorithms ALGORITHM-ID ::= { ..., -- extensible
                                       rsaPublicKey |
                                       rsaSHA-256 }

rsaPublicKey ALGORITHM-ID ::= { OID rsaEncryption PARMS NULL }

rsaSHA-256 ALGORITHM-ID ::= { OID sha256WithRSAEncryption PARMS NULL }
```

```
pkcs-1 OBJECT IDENTIFIER ::= {
     iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) 1 }

rsaEncryption OBJECT IDENTIFIER ::=  { pkcs-1 1 }

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

Sha256WithRSAEncryption OBJECT IDENTIFIER  ::=  { pkcs-1 11 }

AttributeTypeAndValue ::= SEQUENCE {
        type    ATTRIBUTE.&id ({SupportedAttributes}),
        value   ATTRIBUTE.&Type ({SupportedAttributes}{@type})}

Name ::= CHOICE {
        rdnSequence  RDNSequence
}

RDNSequence ::= SEQUENCE OF RelativeDistinguishedName

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

ID ::= OBJECT IDENTIFIER

ATTRIBUTE ::= CLASS {
        &Type,
        &id     OBJECT IDENTIFIER UNIQUE }
WITH SYNTAX {
        WITH SYNTAX &Type
        ID          &id }

SupportedAttributes ATTRIBUTE ::= {
        commonName | countryName | organizationName | organizationalUnitName }

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

countryName ATTRIBUTE ::= {
        WITH SYNTAX        PrintableString (SIZE (2))
        ID                 id-at-countryName }
```

```
organizationName ATTRIBUTE ::= {
        WITH SYNTAX        DirectoryString {ub-organization-name}
        ID                 id-at-organizationName }

organizationalUnitName ATTRIBUTE ::= {
        WITH SYNTAX        DirectoryString {ub-organizational-unit-name}
        ID                 id-at-organizationalUnitName }

id-at OBJECT IDENTIFIER ::= {joint-iso-ccitt(2) ds(5) 4}

id-at-commonName              OBJECT IDENTIFIER ::= {id-at 3}
id-at-countryName             OBJECT IDENTIFIER ::= {id-at 6}
id-at-organizationName        OBJECT IDENTIFIER ::= {id-at 10}
id-at-organizationalUnitName  OBJECT IDENTIFIER ::= {id-at 11}

DirectoryString { INTEGER:maxSize } ::= CHOICE {
        printableString         PrintableString (SIZE (1..maxSize)),
        utf8String              UTF8String (SIZE(1..maxSize))
}

ub-common-name               INTEGER ::= 64
ub-organization-name         INTEGER ::= 64
ub-organizational-unit-name  INTEGER ::= 64

END
```

2　Implicitly Tagged Module

```
MOJCrossAndLinkCertImplicit { 1 2 392 100300 1 4 52 }

DEFINITIONS IMPLICIT TAGS ::=
BEGIN

--EXPORTS ALL --

IMPORTS
        Name, CertificateSerialNumber, DirectoryString, EXTENSION
        FROM MOJCrossAndLinkCertExplicit { 1 2 392 100300 1 4 51 };


authorityKeyIdentifier EXTENSION ::= {
        SYNTAX          AuthorityKeyIdentifier
        IDENTIFIED BY   id-ce-authorityKeyIdentifier }

AuthorityKeyIdentifier ::= SEQUENCE {
    keyIdentifier               [0] KeyIdentifier,
```

```
    authorityCertIssuer         [1] GeneralNames,
    authorityCertSerialNumber   [2] CertificateSerialNumber }

KeyIdentifier ::= OCTET STRING

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

GeneralName ::= CHOICE {
        directoryName               [4] Name,
        uniformResourceIdentifier   [6] IA5String
}

subjectKeyIdentifier EXTENSION ::= {
        SYNTAX          SubjectKeyIdentifier
        IDENTIFIED BY   id-ce-subjectKeyIdentifier }

SubjectKeyIdentifier ::= KeyIdentifier

keyUsage EXTENSION ::= {
        SYNTAX KeyUsage
        IDENTIFIED BY id-ce-keyUsage }

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

certificatePolicies EXTENSION ::= {
        SYNTAX  CertificatePoliciesSyntax
        IDENTIFIED BY id-ce-certificatePolicies }

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

PolicyInformation ::= SEQUENCE {
        policyIdentifier   CertPolicyId }

CertPolicyId ::= OBJECT IDENTIFIER
```

```
policyMappings EXTENSION ::= {
     SYNTAX   PolicyMappingsSyntax
     IDENTIFIED  BY  id-ce-policyMappings }

PolicyMappingsSyntax ::= SEQUENCE SIZE (1..MAX) OF SEQUENCE {
     issuerDomainPolicy        CertPolicyId,
     subjectDomainPolicy       CertPolicyId }

basicConstraints EXTENSION ::= {
        SYNTAX  BasicConstraintsSyntax
        IDENTIFIED BY id-ce-basicConstraints }

BasicConstraintsSyntax ::= SEQUENCE {
        cA            BOOLEAN DEFAULT FALSE }

authorityInfoAccess EXTENSION ::= {
        SYNTAX  AuthorityInfoAccessSyntax
        IDENTIFIED BY id-pe-authorityInfoAccess }

AuthorityInfoAccessSyntax  ::=
        SEQUENCE SIZE (1..MAX) OF AccessDescription

AccessDescription  ::=  SEQUENCE {
        accessMethod          OBJECT IDENTIFIER,
        accessLocation        GeneralName  }

crlDistributionPoints EXTENSION ::= {
        SYNTAX  CRLDistPointsSyntax
        IDENTIFIED BY id-ce-cRLDistributionPoints }

CRLDistPointsSyntax ::= SEQUENCE SIZE (1..MAX) OF DistributionPoint

DistributionPoint ::= SEQUENCE {
        distributionPoint       [0]     DistributionPointName }

DistributionPointName ::= CHOICE {
        fullName                [0]     GeneralNames }

id-ce   OBJECT IDENTIFIER ::= {joint-iso-ccitt(2) ds(5) 29}

id-ce-subjectKeyIdentifier         OBJECT IDENTIFIER ::= {id-ce 14}
id-ce-keyUsage                     OBJECT IDENTIFIER ::= {id-ce 15}
id-ce-basicConstraints             OBJECT IDENTIFIER ::= {id-ce 19}
id-ce-cRLDistributionPoints        OBJECT IDENTIFIER ::= {id-ce 31}
id-ce-certificatePolicies          OBJECT IDENTIFIER ::= {id-ce 32}
id-ce-policyMappings               OBJECT IDENTIFIER ::= {id-ce 33}
```

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

id-pkix OBJECT IDENTIFIER ::=
  { iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) pkix(7) }

id-pe   OBJECT IDENTIFIER ::= { id-pkix 1 }

id-pe-authorityInfoAccess OBJECT IDENTIFIER ::= { id-pe 1 }

id-ad   OBJECT IDENTIFIER ::= { id-pkix 48 }

id-ad-ocsp      OBJECT IDENTIFIER ::= { id-ad 1 }

END
```