```
    subject      [5] Name                  OPTIONAL,
    publicKey    [6] SubjectPublicKeyInfo,
    extensions   [9] Extensions
}

ProofOfPossession ::= CHOICE {
    signature         [1] POPOSigningKey }

POPOSigningKey ::= SEQUENCE {
    algorithmIdentifier   AlgorithmIdentifier,
    signature             BIT STRING }

GeneralName ::= CHOICE {
        directoryName               [4] Name
}

registeredCorporationInfo EXTENSION ::= {
        SYNTAX  RegisteredCorporationInfoSyntax
        IDENTIFIED BY id-registeredcert-pe-registeredCorporationInfo }

RegisteredCorporationInfoSyntax ::= SEQUENCE {
  corporateName              [0] EXPLICIT DirectoryString{ub-corporate-name},
  corporateAddress           [2] EXPLICIT DirectoryString{ub-corporate-address},
  representativeDirectorName  [3] EXPLICIT DirectoryString
                                          {ub-representative-director-name},
  representativeDirectorTitle [4] EXPLICIT DirectoryString
                                          {ub-representative-director-title}
}

id-registeredcert OBJECT IDENTIFIER ::= { 1 2 392 100300 1 }

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

id-registeredcert-pe-registeredCorporationInfo OBJECT IDENTIFIER ::= {
     id-registeredcert-pe 3 }

ub-corporate-name                INTEGER ::= 128
ub-corporate-address             INTEGER ::= 128
ub-representative-director-name  INTEGER ::= 126
ub-representative-director-title  INTEGER ::= 128

END
```

付録２　電子証明書の方式（ASN.１構造とオブジェクト識別子）

１　Explicitly Tagged Module

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

-- EXPORTS ALL --

IMPORTS
       authorityKeyIdentifier, subjectKeyIdentifier, keyUsage,
       privateKeyUsagePeriod, certificatePolicies,
       basicConstraints, authorityInfoAccess, jCertificatePolicies,
       registrar, registeredCorporationInfo, crlDistributionPoints
       FROM MOJCorpCertImplicit { 1 2 392 100300 1 4 2 };


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
```