CMS 長期署名プロファイル

2005 年 8 月 10 日

次世代電子商取引推進協議会（ECOM）

まえがき

長期署名フォーマットの仕様が、次のドキュメントで定義されている。

- RFC3126, “Electronic Signature Formats for long term electronic signatures” (ETSI TS 101 733 V.1.2.2(2000-12)の長期署名フォーマットと同等)
- ETSI TS 101 733 V1.5.1(2003-12), “Electronic Signature Formats”
- draft-pinkas-smime-cades-00.txt(2005-7), “CMS Advanced Electronic Signatures (CAdES)”

上記標準では、多くの選択的な定義が含まれており、システムを実現するに当たっては、そのサブセットのみをサポートすれば電子署名文書の長期保存が可能となる。電子商取引実証推進協議会（ECOM）報告書「電子署名文書長期保存に関するガイドライン（H14 年 3 月)」では、必要なサブセットを規定する推奨プロファイルの概要を紹介している。

本プロファイル案では、ECOM の推奨するプロファイルを更に詳細化し、電子署名文書の長期保存を可能とするシステムにとって必要な項目を明確にする。１章ではプロファイルの詳細を説明し、２章にプロファイルを表としてまとめる。

1. 長期署名プロファイル詳細

本プロファイル案では、最新の“CMS Advanced Electronic Signatures (CAdES)”の仕様をベースとし、効果的（真正性の長期保証が可能）かつ効率的（必要最低限のデータやタイムスタンプを利用）なプロファイルを検討する。

1.1. 署名フォーマット

基本となる電子署名文書の形式として、署名ポリシの有無により、図 1 に示す BES（Basic Electronic Signature）と図 2 に示す EPES（Explicit Policy Electronic Signature）の２通りを利用できる。

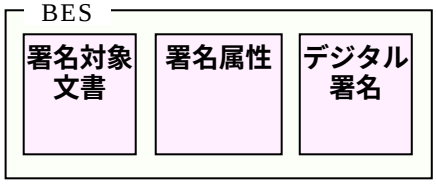

図 1 BES

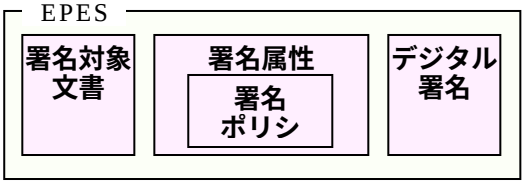

図 2 EPES

署名ポリシとは、署名者と検証者がデジタル署名を有効とみなすための、署名の生成と検証に関する一連の規則を定めるものであり、"ETSI TR 102 272 V1.1.1(2003.12) ： Electronic Signatures and Infrastructures (ESI);ASN.1 format for signature policies"及び"RFC3125 : Electronic Signature"に規定されている。

電子署名文書の形式は次の仕様に基づいている。

- Cryptographic Message Syntax (CMS : RFC3852)
- Enhanced Security Services (ESS : RFC2634)

（1）General syntax

電子署名文書形式の General syntax は、CMS(RFC3852)に定義されているとおり。

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

    ContentType ::= OBJECT IDENTIFIER
```

（2）Data content type

Data content type は、CMS(RFC3852)に定義されているとおり。

```
    id-data OBJECT IDENTIFIER ::= { iso(1) member-body(2)
                                    us(840) rsadsi(113549) pkcs(1) pkcs7(7) 1 }
```

（3）Signed-data content type

Signed-data content type は、CMS(RFC3852)に定義されているとおり。

```
    id-signedData OBJECT IDENTIFIER ::= { iso(1) member-body(2)
                                          us(840) rsadsi(113549) pkcs(1) pkcs7(7) 2 }
```

（4）SingedData type

SignedData の構文は、CMS(RFC3852)に定義されているとおり。

```
    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
```

- SignedData のバージョンは**３である必要はない。**
  - 次のいずれかを満たす場合はバージョン３
    - certificates 属性がありバージョン１の属性証明書があり、バージョン２の属性

証明書がない

- ✧ encapsulated content type が id-data 以外
- ✧ いずれかの SignerInfo がバージョン３

➢ ただし、certificates 属性があり異なるタイプの証明書が存在するか、crls 属性があり異なるタイプのＣＲＬが存在する場合を除く。

（５）EncapusulatedContentInfo type

EncapusulatedContentInfo type は、CMS(RFC3852)に定義されているとおり。

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

ContentType ::= OBJECT IDENTIFIER
```

- 長期保存のためには、SignedData に eContent を含めておくか、別途保存・管理しておくことを推奨する。
- アーカイブタイムスタンプ生成の時には必ず eContent を対象に含めなければならない。

（６）SingerInfo type

SignerInfo type は、CMS(RFC 3852)に定義されているとおり。

```
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
```

- 署名者一人につき、ひとつの SignerInfo が対応し、署名が並列に複数添付される場合には、複数の SingerInfo が作られる。
- SignerInfo の**<u>バージョンは問わない</u>**
  - ➢ SignerIdentifier が issuerAndSerialNumber ならば１
  - ➢ SignerIdentifier が subjectKeyIdentifier ならば３

- 長期署名では、少なくとも SignedAttributes に次の値を格納していなければならない。
  - ContentType
  - MessageDigest
  - SigningCertificate

① Message digest の算出プロセス

CMS(RFC3852)に定義されているとおり。

② Message signature の生成プロセス

CMS(RFC3852)に定義されているとおり。

③ Message signature の検証プロセス

CMS(RFC3852)に定義されているものを本書で拡張したもの。

署名検証プロセスでは、必ず ESS Signing Certificate 属性、あるいは Other Signing Certificate 属性を利用して正しいことが確認された署名者公開鍵を用いる。

（7）必須の CMS 属性

次の属性は署名データの signed attribute 中に存在しなければならない。

① Content type

構文は、CMS(RFC3852)に定義されているとおり。

```
id-contentType OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                        rsadsi(113549) pkcs(1) pkcs9(9) 3 }

ContentType ::= OBJECT IDENTIFIER
```

- ContentType 属性は、signed attribute 内にただ 1 つだけ存在しなければならない。

② Message digest

構文は、CMS(RFC3852)に定義されているとおり。

```
Id-messageDigest OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                          rsadsi(113549) pkcs(1) pkcs9(9) 4 }

MessageDigest ::= OCTET STRING
```

- MessageDigest 属性の値は、SignerInfo の DigestAlgorithm を利用して encapContentInfo eContent OCTET STRING の ASN.1 TLV の V から求めたもの。

MessageDigest 属性は、signed attribute 内にただ 1 つだけ存在しなければならない。

③ Signing certificate 属性

signed-data には、ESS signing certificate あるいは Other signing certificate のどちらか一方のみの signing certificate 属性を含めなければならない。この属性値は、”simple substitution 攻撃” と “re-issue 攻撃”を防ぐために用いる。

A) ESS signing certificate 属性の定義

ESS signing certificate 属性は ESS(RFC2634)に基づく。ESS signing certificate は signed attribute でなければならない。

signing certificate 属性または次項の Other signing certificate 属性は必ず存在しなければならず、属性値が空であってはならない。署名検証のための証明書はこの属性値から得なければならない。署名検証ポリシ(Signature Validation Policy)でここに他の証明書が存在することを規定していれば、signing certificate 属性に信頼点までのすべての証明書を含むこともある。ESSCertID は issuerSerial フィールドを含まなければならない。

SignerInfo の issuerAndSerialNumber は issuerSerial フィールドと整合が取れていなければならない。署名検証は ESSCertID で特定された証明書を利用して行う必要がある。もしも証明書のハッシュ値が署名検証用の証明書とマッチしないものであった場合は、署名は無効であるとみなさねばならない。

また、policy information フィールドは利用しない。

```
SigningCertificate ::=  SEQUENCE {
                certs        SEQUENCE OF ESSCertID,
                policies     SEQUENCE OF PolicyInformation OPTIONAL  --利用しない
}

id-aa-signingCertificate OBJECT IDENTIFIER ::= { iso(1)member-body(2) us(840)
                                  rsadsi(113549) pkcs(1) pkcs9(9) smime(16) id-aa(2) 12 }

ESSCertID ::=  SEQUENCE {
                certHash        Hash,
                issuerSerial    IssuerSerial OPTIONAL
}

Hash ::= OCTET STRING -- SHA1 hash of entire certificate

IssuerSerial ::= SEQUENCE {
                Issuer          GeneralNames,
                serialNumber    CertificateSerialNumber
    }
```

B) Other signing certificate 属性の定義

SHA-1 以外のハッシュアルゴリズムを利用できることを除いて、ESS SigningCertificate の定義と同じ。

```
id-aa-ets-otherSigCert OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                        rsadsi(113549) pkcs(1) pkcs9(9) smime(16) id-aa(2) 19 }

OtherSigningCertificate ::=   SEQUENCE {
            certs        SEQUENCE OF OtherCertID,
            policies     SEQUENCE OF PolicyInformation OPTIONAL  --利用しない
    }

OtherCertID ::= SEQUENCE {
            otherCertHash            OtherHash,
            issuerSerial             IssuerSerial OPTIONAL
    }

OtherHash ::= CHOICE {
            sha1Hash   OtherHashValue,  -- SHA-1 の場合、ここに格納
            otherHash   OtherHashAlgAndValue
    }

OtherHashValue ::= OCTET STRING

OtherHashAlgAndValue ::= SEQUENCE {
            hashAlgorithm   AlgorithmIdentifier,
            hashValue       OtherHashValue
  }
```

※ESS SigningCertificate 属性、Other signing certificate 属性、いずれも検証時には処理できなければならない。

（8）EPES に対する必須属性

① Signature policy identifier

EPES は署名ポリシに対するリファレンスを持たなければならない。signature policy identifier は signed attribute でなければならない。

```
id-aa-ets-sigPolicyId OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs9(9) smime(16) id-aa(2) 15 }

SignaturePolicyIdentifier ::= CHOICE{
            SignaturePolicyId           SignaturePolicyId,
            SignaturePolicyImplied      SignaturePolicyImplied }

SignaturePolicyId ::= SEQUENCE {
            sigPolicyIdentifier   SigPolicyId,
            sigPolicyHash         SigPolicyHash,
            sigPolicyQualifiers   SEQUENCE SIZE (1..MAX) OF
               SigPolicyQualifierInfo      OPTIONAL
    }

SignaturePolicyImplied ::= NULL

SigPolicyId ::= OBJECT IDENTIFIER

SigPolicyHash ::= OtherHashAlgAndValue

SigPolicyQualifierInfo ::= SEQUENCE {
            sigPolicyQualifierId   SigPolicyQualifierId,
            sigQualifier             ANY DEFINED BY sigPolicyQualifierId
    }

SigPolicyQualifierId ::=    OBJECT IDENTIFIER
id-spq-ets-uri OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs9(9) smime(16) id-spq(5) 1 }

SPuri ::= IA5String

id-spq-ets-unotice OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs9(9) smime(16) id-spq(5) 2 }

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

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

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

（9）オプションの CMS 属性

次にあげる属性は、本書で定義する署名データに出現しても良い属性であり、本属性が存在することを理由として、構築時や検証時にエラーとしてはならない。

① Signing time

構文は、CMS(RFC3852)に定義されているとおりだが、長期署名では UTCTime でなく GeneralizedTime（YYYYMMDDHHMMSSZ と表記。秒の端数は含めない）の使用を推奨している。

```
id-signingTime OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                                    rsadsi(113549) pkcs(1) pkcs9(9) 5 }

SigningTime ::= Time

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

- SigningTime 属性は、signed attribute 内に複数存在してはならない。

② Countersignature

countersignature は unsigned attribute でなければならない。

```
id-countersignature OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                                rsadsi(113549) pkcs(1) pkcs9(9) 6 }

Countersignature ::= SignerInfo
```

※unsigned attribute に含まれる countersignature は、生成時期に制約がない（つまり ES-A 生成後の署名に対しても countersignature を添付可能と考えられる）。また長期署名フォーマットの適用も可能と考えられる。countersignature の対象となっている署名における archiveTimeStamp の対象については注意を要する。

（１０）オプションの ESS 属性

次にあげる属性は、本書で定義する署名データに含めても良い属性であり、本属性が存在することを理由として、構築時や検証時にエラーとしてはならない。

① content reference 属性

content reference 属性は signedAttribute。

```
id-aa-contentReference    OBJECT IDENTIFIER ::= { iso(1) member-body(2)
                              us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 10 }

ContentReference ::= SEQUENCE {
            contentType                 ContentType,
            signedContentIdentifier    ContentIdentifier,
           originatorSignatureValue   OCTET STRING }
```

② Content Identifier 属性

Contentidentifier は signed attribute。

```
id-aa-contentReference    OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 10 }

ContentReference ::= SEQUENCE {
            contentType ContentType,
            signedContentIdentifier ContentIdentifier,
            originatorSignatureValue OCTET STRING }
```

③　Content Hints 属性

```
ContentHints ::= SEQUENCE {
            contentDescription UTF8String (SIZE (1..MAX)) OPTIONAL,
                contentType ContentType }

id-aa-contentHint OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
            rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 4}
```

（１１）他のオプション属性

次にあげる属性は、本書で定義する署名データに含めても良い属性であり、本属性が存在することを理由として、構築や検証時にエラーとしてはならない。

①　Commitment Type Indication 属性

commitmentTypeIndication は signedAttribute である。

```
id-aa-ets-commitmentType OBJECT IDENTIFIER ::= { iso(1) member-body(2)
                        us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 16}

CommitmentTypeIndication ::= SEQUENCE {
            commitmentTypeId            CommitmentTypeIdentifier,
            commitmentTypeQualifier     SEQUENCE SIZE (1..MAX) OF
                                                CommitmentTypeQualifier      OPTIONAL
    }

CommitmentTypeIdentifier ::= OBJECT IDENTIFIER

CommitmentTypeQualifier ::= SEQUENCE {
            commitmentTypeIdentifier   CommitmentTypeIdentifier,
            qualifier                  ANY DEFINED BY  commitmentTypeIdentifier
    }
```

```
id-cti-ets-proofOfOrigin OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) cti(6) 1}
id-cti-ets-proofOfReceipt OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) cti(6) 2}
id-cti-ets-proofOfDelivery OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) cti(6) 3}
id-cti-ets-proofOfSender OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) cti(6) 4}
id-cti-ets-proofOfApproval OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) cti(6) 5}
id-cti-ets-proofOfCreation OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) cti(6) 6}
```

Proof of origin	署名者がその文書を生成したこと、承認したこと、そして送信したことを示す。
Proof of receipt	署名者がその文書を受け取ったことを示す。
Proof of delivery	TSP（信頼できるサービスプロバイダ）がその文書を受信者がアクセスできる状態でローカルなストアにおいたことを提示したことを示す。
Proof of sender	その提示をしたエンティティがその文書を送信したことを示す。（生成したのではなくても良い）
Proof of approval	署名者がその文書を承認したことを示す。
Proof of creation	署名者がその文書を生成したことを示す。（承認したり送信したりする必要はない）

② Signer Location 属性

Signer Location は signedAttribute。

```
id-aa-ets-signerLocation OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                        rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 17}

SignerLocation ::= SEQUENCE {
                -- 少なくとも次のどれか１つが必
            countryName             [0] DirectoryString      OPTIONAL,
                -- X.500 の Coutry 名
            localityName            [1] DirectoryString      OPTIONAL,
                -- X.500 の locality 名
            postalAdddress          [2] PostalAddress        OPTIONAL
    }

PostalAddress ::= SEQUENCE SIZE(1..6) OF DirectoryString
```

③ Signer Attributes 属性

signer-attributes は signed attribute。

```
id-aa-ets-signerAttr OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                        rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 18}

SignerAttribute ::= SEQUENCE OF CHOICE {
            claimedAttributes       [0]  ClaimedAttributes,
            certifiedAttributes     [1]  CertifiedAttributes
    }

ClaimedAttributes ::= SEQUENCE OF Attribute

CertifiedAttributes ::= AttributeCertificate
```

④ Content Time-Stamp 属性

content time-stamp は signed attribute。

```
id-aa-ets-contentTimestamp OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                        rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 20}

ContentTimestamp::= TimeStampToken
```

（１２）複数署名のサポート

① 並列署名（Independent Signatures）

並列署名は、同一の文書に対して並行して複数人による複数の署名を添付する場合に用いる。個々の署名は独立であり、これを実現するためには複数の SignerInfo を利用する。SignerInfo 毎に独立して長期署名フォーマットを適用できる。

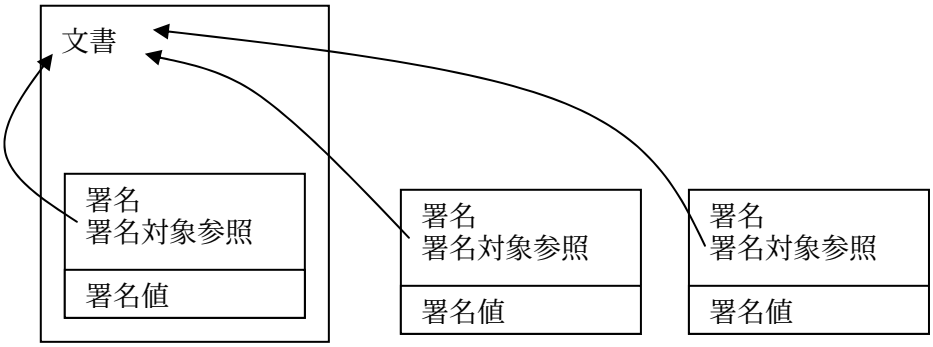

図 3　並列署名

② 直列署名（Embedded Signatures）

直列署名は、署名に対して署名を重ねていく場合に用いる。これを実現するためには、counterSignature 属性を利用する。countersignature は、生成時期に制約がない（つまり ES-A 生成後の署名に対しても countersignature を添付可能と考えられる）。また長期署名フォーマットの適用も可能と考えられる。countersignature の対象となっている署名における archeveTimeStamp の対象については注意を要する。

countersignature に対する長期署名フォーマットの適用については、署名対象が本文（図中の文書）ではなく、そのハッシュ値のみの情報を含む署名値であるため、その有効性について注意深く検討する必要がある。本プロファイルでは countersignature の長期署名フォーマットの適用については規定しない。

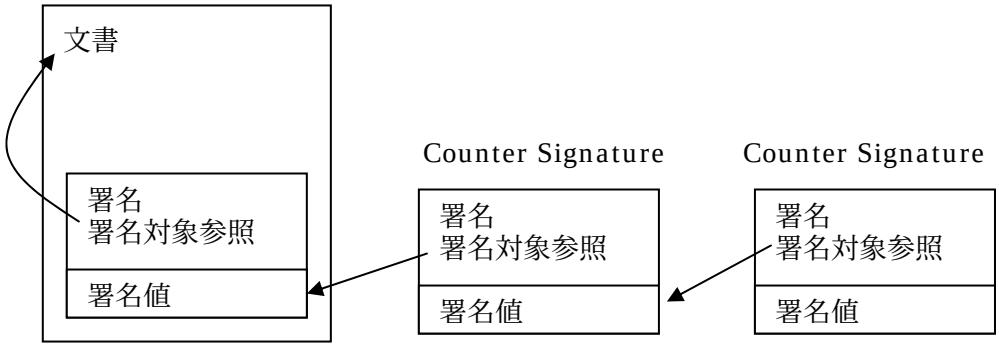

図 4　直列署名

1.2. ES-T（Electronic Signature Time-stamped)

ES-T は、デジタル署名の存在時刻を確定するために、電子署名文書中の署名値（CMS の SignerInfo の SignatureValue）に対して TSA から取得したタイムスタンプトークンを追加したものである。署名値は電子文書のハッシュ値をもとに計算されるため、署名値から生成したタイムスタンプトークンは、署名の存在時刻とともに、電子データの存在時刻も証明することとなる。

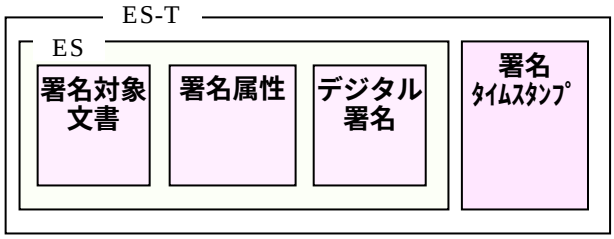

図 5　ES-T

ひとつの署名に対していくつかの異なった TSA からタイムスタンプトークンを取得して格納しても良い。複数の署名が添付されている場合、個々の署名値に対してそれぞれタイムスタンプトークンを取得してもよいし、ある署名についてのみタイムスタンプトークンを取得してもよい。

Signature Timestamp 属性の OID は次の値である。

```
id-aa-signatureTimeStampToken OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 14}
```

Signature Timestamp 属性の値は、ASN.1 形式の SignatureTimeStampToken である。

```
SignatureTimeStampToken ::= TimeStampToken
```

TimeStampToken の messageImprint フィールドの値は、signedData の SignerInfo 内の signature フィールドの値である。

TimeStampToken は、RFC3161 で定義される。

SignatureTimestamp の検証情報（認証パス及び失効情報、ES の検証情報に準じる）は次のいずれかに格納する。

1) タイムスタンプトークン内の certificates と crls
2) ES の検証情報と同じ場所（Complete validation reference data と Extended validation data）
3) タイムスタンプトークン内の unsigned attribute(Extended validation data 形式)

※構築時は、1)または 3)を推奨、検証時は 1)～3)全てに対応することを必須とする。

1.3. ES-C（Complete validation reference data)

ES-C は、ES-T に対してデジタル署名の検証の際に利用する認証パス上の全ての公開鍵証明書（ただし署名者の公開鍵証明書を除く）とそれぞれの公開鍵証明書の CRL や OCSP レスポンスなどの失効情報（署名者の公開鍵証明書の失効情報を含む）に対するリファレンス情報を追加したものである。

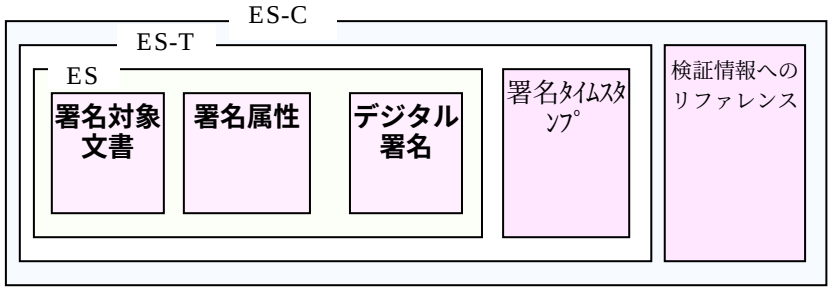

図 6　ES-C

complete validation reference data 付の電子署名は、署名の検証に必要なすべてのデータ（証明書及び失効情報）を備える電子署名文書である。

Complete validation reference data の最小構成は次のとおり：

- Signature Timestamp 属性
- Complete Certificate Refs
- Complete Revocation Refs

Complete validation reference data は次の情報を含む X-Long validation data を構成してもよい(将来、検証プロセスがこれらのデータにアクセスできなくなることに備えるため)：

- Complete Certificate Values　**←必須とする。**
- Complete Revocation Values　**←必須とする。**

Complete validation reference data はまた次の情報を含む Extended validation data を構成してもよい(将来の CA の危殆化に備えることと、検証データの完全性を確保するため)：

- ES-C Timestamp(ES-X Type1 の場合に存在)　**←利用しない（無視してかまわない)。**
- Time-Stamped Certificates and CRLs references(ES-X Type2 の場合に存在)　**←利用しない（無視してかまわない)。**

① Complete Certificate Refs 属性の定義

Complete Certificate Refs 属性は unsigned attribute である。Complete Certificate Refs 属性は ES の検証に用いる署名者の証明書に至るすべての CA の証明書を参照する。（ただし署名者の証明書への参照は含まない）

この属性は１署名につき一つだけ含む。

注記１：署名者の証明書は signing certificate 属性で参照される。

注記２：署名タイムスタンプの認証パスを含んでも良い。

Complete Certificate Refs 属性の OID は次のとおりである。

```
id-aa-ets-certificateRefs OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 21}
```

Complete Certificate Refs 属性は ASN.1 構文の CompleteCertificateRefs を値として持つ。

```
CompleteCertificateRefs ::= SEQUENCE OF OtherCertID
```

OtherCertID には IssuerSerial を含まねばならない。certHash は参照される証明書のハッシュ値とマッチしなければならない。

② Complete Revocation Refs 属性の定義

Complete Revocation Refs 属性は unsigned attribute である。この属性は１署名に対して一つだけ存在する。この属性は、ES-C を検証するために必要な署名者及び CA の証明書に対する CRL あるいは OCSP レスポンスのすべてを参照する。

注記：署名タイムスタンプの認証パスに対する失効情報を格納しても良い。

Complete Revocation Refs 属性の OID は次のとおり。

```
id-aa-ets-revocationRefs OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                        rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 22}
```

Complete Revocation Refs 属性は ASN.1 構文の CompleteRevocationRefs を値として持つ。

```
CompleteRevocationRefs ::= SEQUENCE OF CrlOcspRef

CrlOcspRef ::= SEQUENCE {
        crlids          [0] CRLListID       OPTIONAL,
        ocspids         [1] OcspListID      OPTIONAL,
        otherRev        [2] OtherRevRefs    OPTIONAL
    }
```

CompleteRevocationRefs は signing certificate に対する CrlOcspRef を必ず１つ持たなければならず、CompleteCertificateRefs 属性の中の各 OtherCertID に対して１つずつ CrlOcspRef を持たなければならない。２番目以降の CrlOcspRef の順番は、対応する OtherCertID の順番と同じでなければならない。信頼しているＣＡの証明書を除く証明書パス上のすべての証明書に対して、CRLListID、OcspListID、OtherRevRefs のうち、少なくとも一つを含めなければならない。CRL あるいは OCSP レスポンス以外の失効情報は利用しない。

```
CRLListID ::=   SEQUENCE {
            crls          SEQUENCE OF CrlValidatedID}

CrlValidatedID ::=   SEQUENCE {
            crlHash                 OtherHash,
            crlIdentifier           CrlIdentifier OPTIONAL}

CrlIdentifier ::= SEQUENCE {
            crlissuer               Name,
            crlIssuedTime           UTCTime,
            crlNumber               INTEGER OPTIONAL
    }

OcspListID ::=   SEQUENCE {
            ocspResponses         SEQUENCE OF OcspResponsesID}

OcspResponsesID ::=   SEQUENCE {
            ocspIdentifier            OcspIdentifier,
            ocspRepHash               OtherHash    OPTIONAL
    }

OcspIdentifier ::= SEQUENCE {
            ocspResponderID     ResponderID,
              -- As in OCSP response data
            producedAt        GeneralizedTime
              -- As in OCSP response data
    }
```

crlValidatedID を作成する際、crlHash は、署名を含む CRL の完全な DER エンコードされたデータに対して計算する。crlIdentifier は、通常、他の情報によって CRL が推測できないときに存在する。

crlIdentifier は、CRL を特定するためのものであり、発行者名と発行時刻（CRL が含む"thisUpdate"が示す時刻）を利用している。

crlListID 属性は、unsigned attribute である。CRL が Delta CRL であれば、complete revocation list には CRL の集合に対する参照が含まれねばならない

OcspIdentifier は OSCP レスポンスを特定するもので、発行者名と発行時刻（OCSP レスポンスに含まれる"producedAt"が示す時刻）を用いる。同時刻に発行された OCSP を区別するには、OcspResponceID に含まれるレスポンスのハッシュ値を用いる。

注記１：署名タイムスタンプの失効情報を含めてよい。

```
OtherRevRefs ::= SEQUENCE {
            otherRevRefType       OtherRevRefType
            otherRevRefs          ANY DEFINED BY otherRevRefType
    }

OtherRevRefType ::= OBJECT IDENTIFIER
```

1.4. Extended Validation Data（ES-X）

ES-X は、将来の CA の危殆化に備えたり、検証データの完全性を確保したり入手困難になることに備えるために、ES-C を拡張するものである。

ES-X には、ES-X Long（図 4-8）、ES-X Type1（図 4-9）、ES-X Type2（図 4-10）の 3 通りのフォーマットが用意される。本プロファイルでは、**ES-X Long の使用のみを認める。**

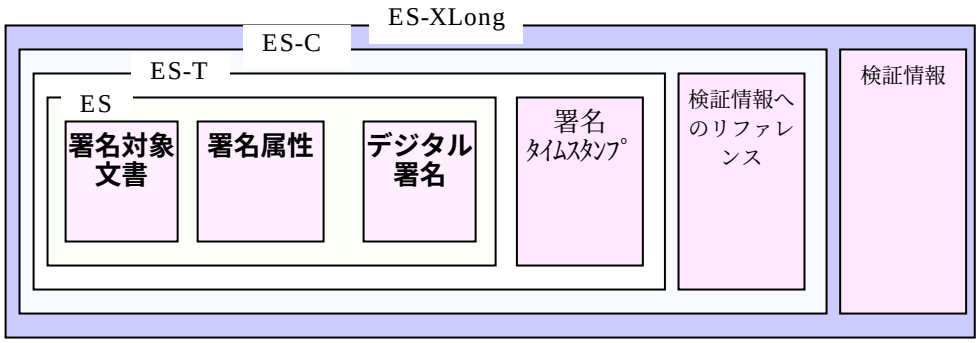

図 7　ES-X Long

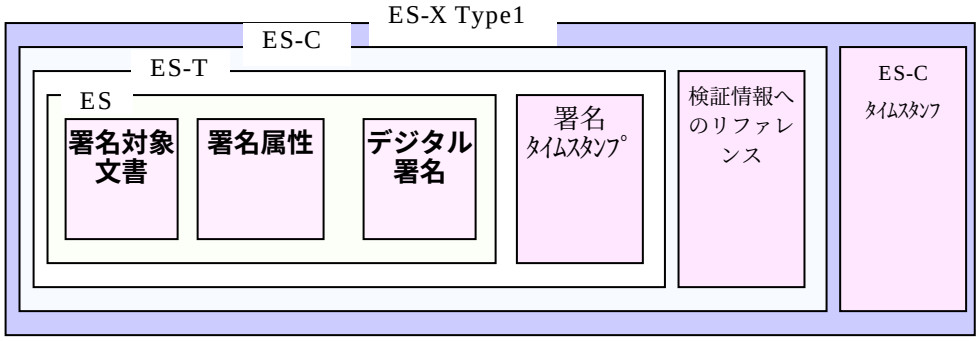

図 8　ES-X Type1（使用しない）

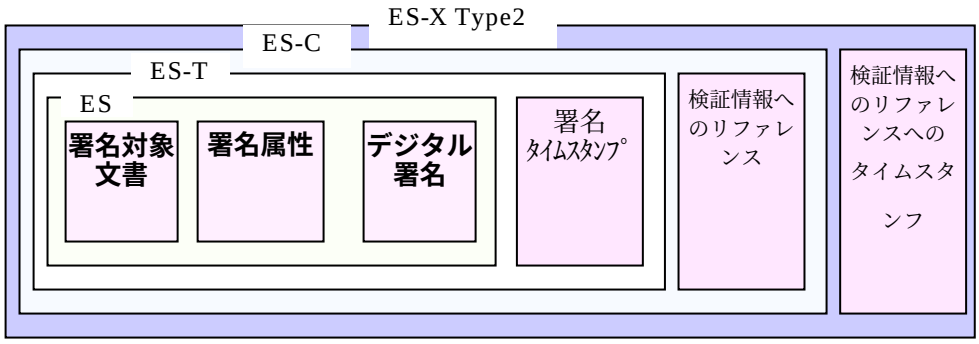

図 9　ES-X Type2（使用しない）

ES-X Long は検証情報そのものを電子署名文書内に抱え込むフォーマットである。ES-X Type1 は、ES-C 全体に対するタイムスタンプを取得して追加するもの、ES-X Type2 は、検証情報へのリファレンスのみに対するタイムスタンプを取得して追加するものである。

検証情報のリファレンスには検証情報のハッシュ値が含まれるが、そのとき用いるハッシュ関数が脆弱化するケースを想定すると、リファレンスをタイムスタンプの対象とするのでは、リファレンスと検証情報そのものとの対応関係を証明することができなくなる。更に、検証情報そのもの（特に中間のサブ CA の公開鍵証明書や失効情報など）の消失に備えるには、検証情報そのものを保持しておく必要がある。

長期保存のためには、電子文書、デジタル署名、タイムスタンプ、検証情報全体をタイムスタンプや耐タンパな仕組みで保護する必要がある。長期署名フォーマットでは、この後に述べるアーカイブタイムスタンプによって保護する。つまり、適当な時期にアーカイブタイムスタンプを追加することによって、ES-C タイムスタンプも検証情報リファレンスへのタイムスタンプも必要なくなり、重要なのは検証情報そのものを確保しておくことである。

ES-X Long は保護対象となる全てのデータを格納するフォーマットに相当する。電子署名文書の長期保存を可能とするシステムを実装するためには、ES-X Long のみをサポートすればよい。

① Certificate Values 属性の定義

Certificate Values 属性は unsigned attribute である。この属性は 1 署名につき 1 つだけ存在する。この属性により、CompleteCertificationRefs が参照する証明書および**署名者の証明書**を保持する。**（署名者証明書をここに含めるのは、格納必須である場所が他に指定されていないため、SingedData の Certificates などではアーカイブタイムスタンプの対象とはならず、保護されないため）**

注意：Attribute Certificate が利用されるときは、この構造が用いられるのではなく、signer-attributes 属性が用いられる。

Certificate Values 属性を示す OID は次のとおり。

```
id-aa-ets-certValues OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                        rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 23}
```

Certificate Values 属性は値として次の ASN.1 構文で表される CertificateValues を取る。

```
CertificateValues ::=   SEQUENCE OF Certificate
```

Certificate の定義は RFC3280 と ITU-T Recommendation X.509 を参照のこと。

② Revocation Values 属性の定義

Revocation Values 属性は unsigned attribute である。この属性は 1 署名につき 1 つだけ存在する。この属性は、CompleteRevocationRefs 属性で参照される CRL と OCSP レスポンスの値を保持する。

Revocation Values 属性の OID は次のとおり。

```
id-aa-ets-revocationValues OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                        rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 24}
```

Revocation Values 属性は値として次の ASN.1 構文で表される RevocationValues を取る。

```
RevocationValues ::=   SEQUENCE {
        crlVals          [0] SEQUENCE OF CertificateList     OPTIONAL,
        ocspVals         [1] SEQUENCE OF BasicOCSPResponse   OPTIONAL,
        otherRevVals     [2] OtherRevVals                    OPTIONAL  ←利用しない
}
```

```
OtherRevVals ::= SEQUENCE {
        otherRevValType      OtherRevValType,
        otherRevVals         ANY DEFINED BY otherRevValType
}

OtherRevValType ::= OBJECT IDENTIFIER
```

Other revocation values **←利用しない。**

CertificateList の定義は、RFC3280 と ITU-T Recommendation X.509 を参照のこと。

BasicOCSPResponse の定義は、RFC2560 を参照のこと。

③ ES-C Time-Stamp 属性の定義 **←利用しない。無視してかまわない。**

④ Time-Stamped Certificates and CRLs 属性の定義 **←利用しない。無視してかまわない。**

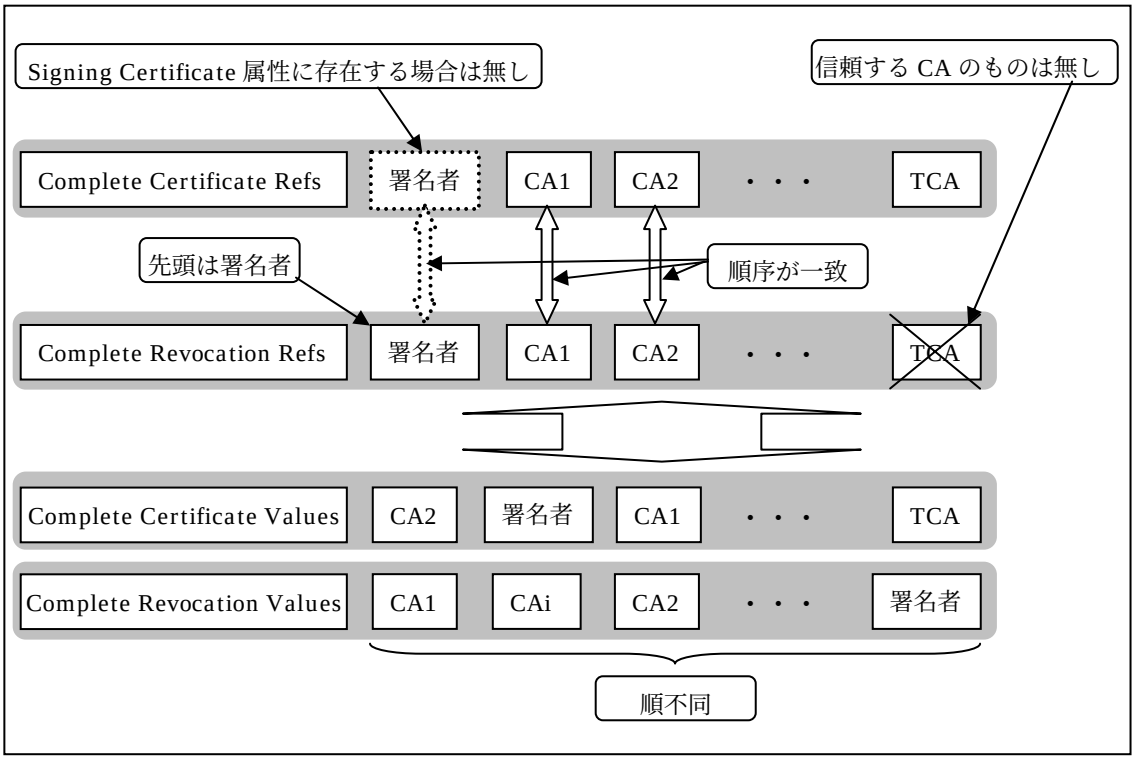

図 10 Complete Validation Reference Data と Validation Values との関係

1.5. ES-A（Archive Validation Data ）

電子署名の検証可能期間を極めて長くしようとしたとき、タイムスタンプの署名の危殆化や TSA の証明書の有効期限切れが発生しうるため、タイムスタンプの署名を複数回重ねることが要求されることがある。このとき、archive time-stamp 属性が用いられる。このタイムスタンプは期間をおいて繰り返し付与される。

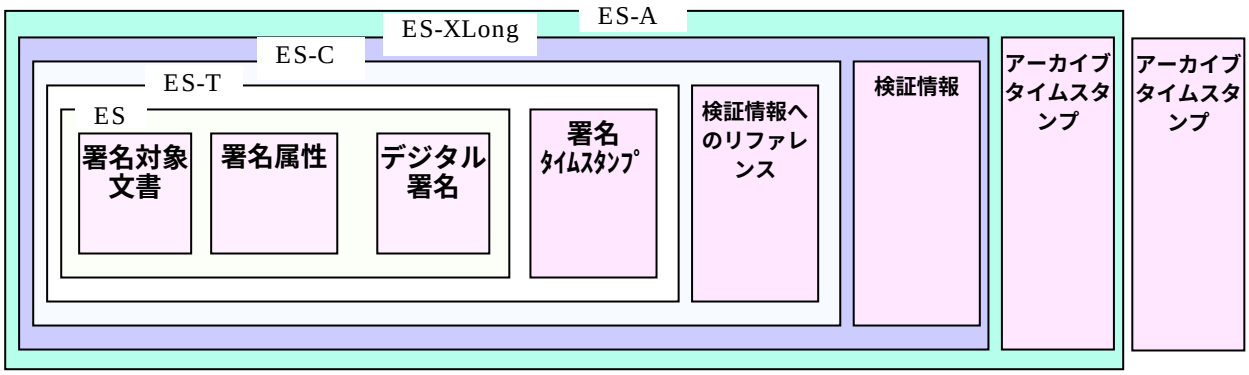

図 11　ES-A

① Archive Time-Stamp 属性の定義

Archive Time-Stamp は署名対象文書と署名全体に対するタイムスタンプである。Certificate Values と Revocation Values 属性がない場合、タイムスタンプをとる前にこれらの属性を加えなければならない。Archive Time-Stamp 属性は、unsigned attribute である。この属性は、１署名に対して、時間の経過や複数の TSA から得ることにより複数添付できる。

Archive Time-Stamp 属性の OID は次のとおりである。

```
id-aa-ets-archiveTimestamp OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                                   rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 27}
```

Archive Time-Stamp 属性の値として、次の ASN.1 構文の ArchiveTimeStampToken が入る。

```
ArchiveTimeStampToken ::= TimeStampToken
```

TimeStampToken の messageImprint の値は、次のデータをこの順に結合した値（ただし、タイプと長さを除いたもの）のハッシュ値である。

- encapContentInfo eContent OCTET STRING;
- signedAttributes;
- signature field within SignerInfo;
- SignatureTimeStampToken attribute;

- CompleteCertificateRefs attribute;
- CompleteRevocationData attribute;
- CertificateValues attribute
  (まだこの値を確保していなければ、ES-A を作る際に確保しなければならない。)
- RevocationValues attribute
  (まだこの値を確保していなければ、ES-A を作る際に確保しなければならない。)
- ESCTimeStampToken attribute if present; **←利用しない。**
- TimestampedCertsCRLs attribute if present; **←利用しない。**
- any previous ArchiveTimeStampToken attributes.

TimeStampToken に関しては、RFC3161 を参照のこと。
タイムスタンプは、オリジナルの署名よりも強いアルゴリズム（あるいは長い鍵）を利用するのが適当である。

アーカイブタイムスタンプの検証情報は次のいずれかに格納することが考えられる。
1) タイムスタンプトークン内の certificates と crls
2) タイムスタンプトークン内の unsigned attribute(Extended validation data 形式)
本プロファイルでは、構築時には 1)に格納することを推奨し、検証時には 1),2)を処理できることを必須とする。

なお、draft-pinkas-smime-cades-00.txt や ETSI TS 101 733 V1.5.1 では、アーカイブタイムスタンプの取得対象が異なる。つまり、**TimeStampToken** の messageImprint の値は、次のデータをこの順に結合した値（ただし、タイプと長さを除いたもの）のハッシュ値である。

- SignedData 内の encapContentInfo
- もしも存在した場合は、SignedData 内の Certificates と crls
- すべての署名属性、非署名属性を含む SignerInfo の全てのデータ

ところが、最後の項目を対象とするとした場合、各アーカイブタイムスタンプの検証時に、そのアーカイブタイムスタンプが対象とした情報が確定できない場合がある。例えば、contersignature が後から添付された場合、countersignature にアーカイブタイムスタンプが付与された場合などがそれに当たる。従って、本プロファイルでは、タイムスタンプ対象を明確に定める旧仕様（RFC3126 や TSI TS 101 733 V1.4.0 以前）に基づいた仕様を採用することとし、新版の仕様は対象外とする。

2. CMS 長期署名プロファイル案まとめ

	CAdES BES	CAdES EPES	CAdES ES-T	CAdES ES-C	CAdES ES-X Long	CAdES ES-A
SignedAttributes	○	○	○	○	○	○
ContentType	○	○	○	○	○	○
MessageDigest	○	○	○	○	○	○
SigningTime	△	△	△	△	△	△
SigningCertificate	○	○	○	○	○	○
SignaturePolicyIdentifier	×	○	△2	△2	△2	△2
ContentReference	△	△	△	△	△	△
ContentIdentifier	△	△	△	△	△	△
ContentHints	△	△	△	△	△	△
CommitmentTypeIndication	△	△	△	△	△	△
SignerLocation	△	△	△	△	△	△
SignerAttribute	△	△	△	△	△	△
ContentTimeStamp	△	△	△	△	△	△
UnsignedAttribute	△	△	○	○	○	○
CounterSignature	△	△	△	△	△	△
SignatureTimeStamp	×	×	○	○	○	○
CompleteCertificateRefs	×	×	×	○	○	○
CompleteRevocationRefs	×	×	×	○	○	○
AttributeCertificateRefs	×	×	×	△	△	△
AttributeRevocationRefs	×	×	×	△	△	△
CertificateValues	×	×	×	×	○	○
RevocationValues	×	×	×	×	○	○
ES-C TimeStamp	×	×	×	×	×	×
TimeStampedCertsAndCrls	×	×	×	×	×	×
ArchiveTimeStamp	×	×	×	×	×	○

○：必須要素

△：オプション要素

SignaturePolicyIdentifier 要素は必須

×：不要（あってはならない要素）

	ETSI TS 101 733 V 1.4.0 以前 (RFC 3126)	ETSI TS 101 733 V 1.5.1
アーカイブタイムスタンプの計算対象	○	― ※1

※1：計算方法に不確定な要素があり、現バージョンのプロファイルでは対象外とする。