



**Agència Catalana
de Certificació**

Guia de sintaxi i formats de signatura
electrònica - Part 1: Sintaxi de Missatge
Criptogràfic

Informació general

Control documental

Projecte:	N/A
Entitat de destinació:	Públic
Títol:	Guia de sintaxi i formats de signatura electrònica
Codi de referència:	ACC_Guia_002-1
Versió:	1.0
Data:	
Fitxer:	Guia formats CMS v1r0.doc
Eina/es d'edició:	Word 2002
Autor/s:	Àrea d'assessorament i recerca de CATCert, amb la col·laboració de l'Institut de Dret i Tecnologia de la Universitat Autònoma de Barcelona
Resum:	

Estat formal

Preparat per:	Revisat per:	Aprovat per:
Nom: AiR/IDT	Nom: Director AiR	Nom: Director General

Control de versions

Versió	Parts que canvien	Descripció del canvi	Data
1.0	Tot	Creació del document	

Índex

1. Introducció	6
1.1 Objecte i abast	6
1.2 Contingut	6
1.3 Referències	6
2. La sintaxi CMS de signatura electrònica	8
2.1 Introducció	8
2.2 La creació de la signatura electrònica	10
2.2.1 Detalls de codificació	10
2.2.2 El procés de càlcul del resum del document a signar.....	11
2.2.3 El procés de generació de la signatura	11
2.3 La verificació de la signatura electrònica	12
3. L'estructura de la signatura electrònica CMS	13
3.1 La informació del contingut encapsulat dins de la signatura electrònica	16
3.2 Els certificats digitals dins de la signatura electrònica	17
3.3 La informació de revocació dins de la signatura electrònica	18
3.4 La informació del signatari	19
3.5 Els atributs bàsics de la signatura electrònica	21
3.5.1 El tipus de contingut signat	21
3.5.2 El resum criptogràfic del document signat.....	22
4. Els atributs opcionals de la signatura electrònica CMS	23
4.1 La data i l'hora de la signatura	23
4.2 La contrasignatura	24
4.3 Els atributs addicionals de la signatura electrònica CMS per correu electrònic segur (S/MIME)	25
4.3.1 El certificat emprat per signar	26
4.3.2 La identificació del contingut signat	27
4.3.3 Les pistes sobre el contingut signat.....	28
4.3.4 La referència al contingut signat	29
5. Els atributs addicionals de la signatura electrònica CMS derivats de la Directiva europea (CADES)	30
5.1 El certificat emprat per signar (definició alternativa)	30
5.2 L'identificador de la política de signatura electrònica	31

5.3	La indicació del tipus de compromís del signatari	34
5.4	La localització del signatari	35
5.5	Els atributs del signatari	36
5.6	El segell de data i hora sobre el contingut	37
5.7	El segell de data i hora sobre la signatura	37
5.8	Les referències completes dels certificats	38
5.9	Les referències completes de la informació de revocació de certificats.....	38
5.10	Les referències completes dels certificats d'atributs	41
5.11	Les referències completes de la informació de revocació d'atributs.....	41
5.12	El segell de data i hora sobre la signatura completa.....	42
5.13	El segell de data i hora sobre les referències de certificats i revocacions	43
5.14	Els valors dels certificats	43
5.15	Els valors de les revocacions	44
5.16	El segell de data i hora d'arxiu	46
6.	<i>Els formats de la signatura electrònica CMS</i>	47
6.1	La signatura electrònica bàsica (CADES-BES).....	47
6.2	La signatura electrònica amb política explícita (CADES-EPES)	48
6.3	La signatura electrònica amb segell de data i hora (CADES-T).....	49
6.4	La signatura electrònica amb referències completes de dades de validació (CADES-C)	50
6.5	La signatura electrònica amb dades completes de validació (CADES-X Long)	52
6.6	La signatura electrònica amb referències completes de dades de validació i segellada (CADES-X Type 1)	54
6.7	La signatura electrònica amb referències completes i segellades de dades de validació (CADES-X Type 2)	55
6.8	La signatura electrònica amb dades completes de validació i segellada (CADES- X Long Type 1)	57
6.9	La signatura electrònica amb dades completes i segellades de validació (CADES-X Long Type 2).....	59
6.10	La signatura electrònica d'arxiu (CADES-A).....	60
	<i>Annex. La sintaxi de la signatura electrònica en CMS</i>	63

1. Introducció

1.1 Objecte i abast

Aquesta guia té per objecte la presentació de la sintaxi de missatge criptogràfic i el seu ús en relació amb la signatura electrònica, així com els seus formats i atributs per a la producció de signatures electròniques amb valor legal.

1.2 Contingut

Aquesta Guia té els següents continguts:

1. Introducció (aquesta secció).
2. La sintaxi CMS de signatura electrònica.
3. L'estructura de la signatura electrònica CMS.
4. Els atributs opcionals de la signatura electrònica CMS.
5. Els atributs addicionals de la signatura electrònica CMS derivats de la Directiva europea (CADES).
6. Els formats de signatura electrònica CMS.
7. Annex.

1.3 Referències

- Directiva 99/93/CE, del Parlament Europeu i del Consell de 13 de desembre de 1999, per la que s'estableix un marc comunitari per a la signatura electrònica.
- ETSI TS 101 733 v1.6.3 (2005-09). Electronic Signatures and Infrastructures (ESI); CMS Advanced Electronic Signatures (CADES).
- IETF RFC 2311, S/MIME Version 2 Message Specification, 1998.
- IETF RFC 2315, PKCS #7: Cryptographic Message Syntax, Version 1.5, 1998.
- IETF RFC 2560, X.509 Internet Public Key Infrastructure, Online Certificate Status Protocol – OCSP, 1999.
- IETF RFC 2630, Cryptographic Message Syntax, 1999.
- IETF RFC 2634, Enhanced Security Services for S/MIME, 1999.
- IETF RFC 2985, PKCS #9: Selected Object Classes and Attribute Types, Version 2.0, 2000.
- IETF RFC 3126, Electronic Signature Formats for Long Term Electronic Signatures, 2001.

-
- IETF RFC 3161, Internet X.509 Public Key Infrastructure, Time-Stamp Protocol (TSP), 2001.
 - IETF RFC 3280, Internet X.509 Public Key Infrastructure, Certificate and Certificate Revocation List (CRL) Profile, 2002.
 - IETF RFC 3281, An Internet Attribute Certificate Profile for Authorization, 2002.
 - IETF RFC 3369, Cryptographic Message Syntax (CMS), 2002.
 - IETF RFC 3850, Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Certificate Handling, 2004.
 - IETF RFC 3851, Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification, 2004.
 - IETF RFC 3852, Cryptographic Message Syntax (CMS), 2004.
 - IETF RFC 3855, Transporting Secure/Multipurpose Internet Mail Extensions (S/MIME) Objects in X.400, 2004.
 - ITU-T Recommendation X.208, Specification of Abstract Syntax Notation One, 1988.
 - ITU-T Recommendation X.209, Specification of Basic Encoding Rules for Abstract Syntax Notation One, 1988.
 - ITU-T Recommendation X.509, The Directory – Authentication Framework, 1988.
 - ITU-T Recommendation X.680, Information Technology – Abstract Syntax Notation One: Specification of Basic Notation, 1997.
 - ITU-T Recommendation X.690, Information Technology – Encoding Rules for Abstract Syntax Notation One: Specification of Basic Encoding Rules, of Canonical Encoding Rules and Distinguished Encoding Rules, 1997.
 - Llei 59/2003, de 19 de desembre, de signatura electrònica.
 - RSA Labs PKCS #6: Extended-Certificate Syntax Standard, Version 1.5, 1993.
 - RSA Labs PKCS #7: Cryptographic Message Syntax Standard, Version 1.5, 1993.
 - RSA Labs PKCS #9: Selected Attribute Types, Version 1.1, 1993.

2. La sintaxi CMS de signatura electrònica

2.1 Introducció

Un dels sistemes més emprats per a la representació de la signatura electrònica, en la qual es va basar inicialment¹, és la Sintaxi de Missatge Criptogràfic (Cryptographic Message Syntax o CMS), que s'utilitza per signar digitalment, produir resums criptogràfics, autenticar, o xifrar qualsevol contingut d'un missatge o document electrònic, mitjançant la Notació de Sintaxi Abstracta Número 1 (ASN.1) i les normes de codificació bàsica (BER) o diferenciada (DER).

L'especificació CMS prové de l'especificació PKCS#7, publicada inicialment en 1991 i, posteriorment en la seva versió 1.5 pels Laboratoris RSA el 1993², a partir de treballs previs de l'IETF orientats a la privacitat del correu electrònic³, i de la UIT-T, orientats a un marc de treball d'autenticació del Directori dins del model de sistemes oberts d'informació (OSI)⁴.

Així mateix, l'especificació PKCS#9, publicada en la seva versió 1.1 pels Laboratoris RSA el 1993, va descriure els atributs de la signatura electrònica especificada a l'especificació PKCS#7.

La Sintaxi de Missatge Criptogràfic (CMS), posteriorment, ha passat a l'IETF per a la seva normalització, que ha publicat els següents documents:

- RFC 2630, l'any 1999.
- RFC 3369, l'any 2002.
- RFC 3852, l'any 2004 (versió actualment vigent).

La Sintaxi de Missatge Criptogràfic és un dels formats més emprats, en l'actualitat, per representar signatures electròniques. Algunes de les aplicacions populars que es basen en aquest format són el correu electrònic segur S/MIME, les aplicacions ofimàtiques Microsoft Office o Adobe PDF.

Així mateix, CMS pot ser emprat per representar signatures electròniques avançades o reconegudes, incorporant les informacions addicionals definides a l'especificació ETSI TS 101733, en desenvolupament de la Directiva europea de signatura electrònica.

¹ Avui existeix un altre format genèric per a la representació de la signatura electrònica, basat en l'ús del llenguatge XML.

² PKCS#7 ha estat també publicat per l'IETF com RFC 2315.

³ Vegeu les especificacions tècniques RFC 1421 a 1424 (Privacy-Enhanced Mail)..

⁴ Vegeu les Recomanacions UIT-T de la sèrie X.500, en especial la Recomanació X.509.

L'especificació CMS descriu una sintaxi d'encapsulament⁵ per a la protecció de les dades (continguts), mitjançant signatura digital i xifratge, i que permet múltiples embolcalls; és a dir, que un sobre d'encapsulament pot estar inclòs dins d'un altre⁶.

La sintaxi CMS també permet incorporar qualsevol atribut, com per exemple la data de creació de la signatura, que se signarà juntament amb el contingut del missatge, i altres atributs que, tot i no ser signats, posteriorment s'associaran amb una signatura, com per exemple una contrasignatura o un segell de data i hora.

La sintaxi CMS pot suportar diferents arquitectures basades en gestió de claus de certificats, com per exemple la definida pel grup de treball PKIX de l'IETF, basada en l'ús de certificats, amb base en la Recomanació X.509v3 de la UIT-T.

Així mateix, la sintaxi CMS és suficient per suportar diferents tipus de contingut, mitjançant el que l'especificació tècnica defineix com un "contingut de protecció", anomenat `ContentInfo`, que encapsula un únic tipus de contingut identificat, que com ja hem indicat pot proporcionar encapsulaments addicionals.

L'especificació defineix sis tipus de contingut: "dades", "dades signades", "dades dins d'un sobre", "dades resumides", "dades xifrades" i "dades autenticades", mentre que altres especificacions tècniques poden definir altres tipus addicionals de contingut de protecció.

Una implementació que es dugui a terme d'acord amb aquesta especificació, ha d'implementar de forma obligatòria el contingut de protecció (`ContentInfo`), així com els següents tipus de continguts: "dades", "dades signades" i "dades dins d'un sobre", i opcionalment es poden implementar altres tipus de continguts.

El següent identificador d'objecte identifica el tipus "informació de contingut de protecció":

```
id-ct-contentInfo OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
rsadsi(113549) pkcs(1) pkcs9(9) smime(16) ct(1) 6 }
```

La sintaxi CMS relaciona un identificador de tipus de contingut amb un contingut concret, mitjançant la sintaxi ASN.1 pel tipus `ContentInfo`:

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

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

⁵ En aquest document emprarem els termes "encapsulament" o "embolcall" com a sinònims.

⁶ D'aquesta manera, hom pot signar digitalment dades encapsulades prèviament.

Els camps del tipus `ContentInfo` tenen els següents significats:

- `contentType` indica el tipus del contingut associat i és un identificador d'objecte, és a dir, una cadena única de nombres enters assignats per una autoritat que defineix el tipus de contingut.
- `content` és el contingut associat i el tipus de contingut únicament pot estar determinat pel camp `contentType`.

El tipus de contingut per “dades,” “dades signades”, “dades dins d’un sobre”, “dades resumides”, “dades xifrades” i “dades autenticades” es defineixen en l’especificació de la CMS, ponent-se definir, com ja s’ha comentat, altres tipus de continguts en altres especificacions⁷.

En aquest sentit, un exemple d’especificació que ha definit un tipus de contingut específic és l’especificació tècnica RFC 3161, que defineix el tipus de contingut, i la corresponent estructura, que representa un segell de data i hora (`TimeStampToken`).

2.2 La creació de la signatura electrònica

2.2.1 Detalls de codificació

En general, els valors de la signatura es generen emprant ASN.1, d’acord amb la Recomanació de la UIT-T X.208 de 1988, i es codifiquen emprant les Normes de Codificació Bàsica BER, d’acord amb la Recomanació de la UIT-T X.209 de 1988, i normalment es representen com cadenes d’octets. Cal tenir en compte, també, que bastants sistemes de missatgeria permeten transmetre qualsevol cadena d’octets d’una manera fiable, però altres no, i que l’especificació CMS no proporciona mecanismes per codificar cadenes d’octets per a les transmissions fiables en els entorns esmentats.

Com filosofia general de disseny, cada tipus de contingut permet el seu tractament en “un sol pas” emprant les Normes de Codificació Bàsica (BER) de longitud indeterminada. Aquest tipus d’operació és especialment eficient si el contingut és llarg, es troba emmagatzemat en comptes o prové de la sortida d’un altre procés.

La operació en un sol pas, però, presenta un inconvenient important quan s’empren les Normes de Codificació Diferenciada (DER), ja que no es coneixen abans de l’operació les longituds dels diferents components.

En qualsevol cas, els atributs signats dins del tipus de contingut “dades signades” i els atributs autenticats dins del tipus de contingut “dades autenticades” han de ser

⁷ Sempre que no es codifiquin com CHOICE.

transmesos codificats en DER per assegurar que els destinataris puguin verificar un contingut amb un o més atributs no reconeguts. Per tant, els atributs signats i els autenticats són els únics tipus de dades emprats a la CMS que requereixen codificació DER.

2.2.2 El procés de càlcul del resum del document a signar

El procés de càlcul del resum del document es pot produir sobre el contingut que s'ha de signar o sobre el contingut i els atributs a signar. En qualsevol cas, l'entrada inicial per al procés de càlcul del resum del document és el valor del contingut encapsulat que s'ha de signar. En concret l'entrada inicial és la cadena d'octets del camp `eContent` sobre el que s'aplica el procés de signatura. Tan sols els octets que conté el valor `eContent OCTET STRING` s'incorporen a l'algorisme de resum del document, no la seva etiqueta o la longitud de la cadena d'octets.

El resultat del procés de càlcul del resum del document depèn de si tenim el camp `signedAttr`. Quan no tenim aquest camp, el resultat és el resum del contingut, tal i com s'ha descrit abans. Quan tenim aquest camp, el resultat és el resum del document de la totalitat de la codificació en DER del valor `SignedAttrs` que trobem en el camp `signedAttrs`.

Donat que el valor `SignedAttrs`, quan el tinguem, ha de contenir els atributs `content-type` i `message-digest` corresponent al contingut signat (indicat al camp `eContent` indicat anteriorment), aquests valors també són indirectament inclosos en el resultat. Per al càlcul del resum criptogràfic es du a terme una codificació a banda del camp `signedAttrs`.

Quan no tenim el camp `signedAttrs`, tan sols formen part del càlcul del resum criptogràfic els octets del camp `eContent`, que corresponen a la informació de contingut encapsulat de les dades signades (per exemple el contingut d'un fitxer que se signa). Això suposa un avantatge, i és que no cal conèixer la longitud del contingut abans del procés de generació de signatura.

Tot i que l'etiqueta i la longitud del contingut no s'inclouen en el procés de resum criptogràfic, es troben protegits per la pròpia naturalesa de l'algorisme de resum, ja que no és pràctic calcular un parell de documents diferents, de qualsevol longitud, que tinguin el mateix resum.

2.2.3 El procés de generació de la signatura

Els elements per la generació del procés de la signatura inclouen el resultat del procés de càlcul del resum del document i la clau privada del signatari. Els detalls de la generació de la signatura dependran de l'algorisme de xifratge emprat. El identificador d'objecte, juntament amb qualsevol paràmetre, que especifiqui l'algorisme de signatura emprat pel signatari es troba en el camp

`signatureAlgorithm`. El valor de la signatura generat pel signatari ha d'estar codificat mitjançant una cadena d'octets, dins del camp `signature`.

2.3 La verificació de la signatura electrònica

Els elements necessaris per dur a terme el procés de verificació de la signatura inclouen el resultat del procés de càlcul de resum del document i la clau pública del signatari.

En relació amb la clau pública del signatari, el destinatari la pot obtenir per qualsevol mitjà, però és preferible que l'aconsegueixi d'un certificat obtingut del camp `certificates` de l'estructura `SignedData`. La selecció i validació de la clau pública del signatari pot estar basada en la validació de la ruta de certificació, així com en altre context extern. Els detalls de la verificació de la signatura dependran de l'algorisme de signatura emprat.

En relació amb el resum, el destinatari no ha de confiar en cap valor de resum de document calculat pel remitent, sinó que l'ha de calcular d'acord amb les indicacions de la secció anterior, en especial per al tractament correcte dels atributs signats.

En aquest cas, perquè la signatura sigui vàlida, el valor del resum criptogràfic calculat pel destinatari ha de ser el mateix que el valor de l'atribut anomenat resum del missatge (`messageDigest`), que com hem vist és un atribut signat de la informació del signatari.

També resulta necessari que, en cas que la informació del signatari inclogui atributs signats, el valor de l'atribut `content-type` coincideixi amb el valor `eContentType` corresponent a la informació del contingut encapsulat de les dades signades.

3. L'estructura de la signatura electrònica CMS

L'estructura de la signatura electrònica CMS s'anomena “dades signades” (`SignedData`) i és un dels tipus de continguts de protecció a que ens hem referit anteriorment.

Aquest tipus de contingut anomenat “dades signades” consisteix en un contingut de qualsevol tipus i zero o més valors de signatura⁸. Qualsevol nombre de signataris poden signar en paral·lel qualsevol tipus de contingut.

L'aplicació típica del tipus de contingut “dades signades” representa la signatura digital d'un signatari sobre el tipus de contingut “dades”, bé es trobi encapsulat dins de la pròpia estructura “dades signades” o a fora. Una altra aplicació típica proporciona certificats i llistes de certificats revocats (CRLs) a terceres persones, sense incloure cap signatura de cap dada concreta.

El procés mitjançant el qual es construeix el contingut de les “dades signades” consta dels següents passos⁹:

1. Per a cada signatari es “calcula” un resum criptogràfic (un valor de hash) del contingut amb un algorisme de resum específic elegit pel signatari. Si el signatari signa qualsevol altra informació diferent al contingut (generalment atributs de la signatura), aleshores el resum criptogràfic i la informació afegida es “resumeixen” amb l'algorisme de resum escollit pel signatari, i el resultat esdevé el “resum criptogràfic” que serà emprat per produir la signatura digital. Aquest procés permet que diferents signataris signin el mateix contingut amb atributs diferents.
2. Cada signatari signa digitalment el resum criptogràfic amb la seva clau privada.
3. El valor de la signatura i altra informació específica de cada signatari es recull en un camp anomenat `signerInfo`. Els certificats i CRLs corresponents a cada signatari, i també els certificats i les CRLs que no corresponen a cap signatari, es recullen en aquest pas.
4. L'algorisme de resum i el valor del camp `signerInfo` de tots els signataris es recullen conjuntament amb el contingut dins el camp `SignedData`, com es defineix més endavant.

El destinatari del document signat ha de calcular de forma independent el resum criptogràfic del document. Aquest resum i la clau pública del signatari serveixen per verificar el valor de la signatura. La clau pública del signatari es pot referir mitjançant

⁸ El cas de zero signatures s'empra freqüentment senzillament per enviar els nostres certificats a una persona, i es tracta d'un mètode que no s'ha d'emprar mai en signatura electrònica amb valor legal, per motius evidents.

⁹ Més endavant s'aporten detalls addicionals sobre la codificació i els procediments.

un nom únic d'emissor juntament amb un número de sèrie específic d'aquest emissor, o bé mitjançant un identificador de clau de subscriptor que identifiqui de forma única el certificat que conté la clau pública. Així mateix, el certificat del signatari es pot incloure en el camp previst a tal efecte en l'estructura "SignedData".

El següent identificador d'objecte identifica el tipus de contingut "dades signades", i presenta la següent estructura:

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

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

Els camps de tipus SignedData tenen els següents significats:

- `version` indica el número de versió de la sintaxi CMS emprada. El valor apropiat dependrà dels certificats, del tipus de contingut encapsulat (`eContentType`), i de la informació dels signataris (`SignerInfo`).
- `digestAlgorithms` és un recull d'identificadors d'algorisme de resum criptogràfic. El camp és obligatori, i hi pot haver un nombre indeterminat d'identificadors d'elements, incloent-hi zero. Cadascun identifica l'algorisme de resum, juntament amb qualsevol altre paràmetre associat, emprat per un o més signataris. Té com objectiu llistar els algorismes de resum de documents utilitzats per qualsevol dels signataris, en qualsevol ordre, per facilitar la verificació de la signatura en un "sol pas".

Cal tenir en compte que les implementacions poden donar errors a l'hora de validar signatures que utilitzin un algorisme que no estigui inclòs en aquest recull.

- `encapContentInfo` és un camp obligatori que conté el contingut signat, i consisteix en un identificador de tipus de contingut (`EncapsulatedContentInfo`) i el contingut en sí mateix, estructurat o no, en funció de la semàntica definida per aquest tipus.
- `certificates`, que és un camp opcional, és un recull de certificats, d'acord amb el tipus `CertificateChoices`. L'objectiu d'aquest conjunt de certificats és

que sigui suficient per construir la ruta de certificació que existeix entre una autoritat de certificació arrel i tots els signataris indicats al camp `signerInfos`. Hi pot haver més certificats dels necessaris, i hi pot haver certificats suficients que continguin les rutes de certificació des de dos o més autoritats de certificació independents. Tanmateix, també hi pot haver menys certificats dels necessaris, si s'espera que els destinataris disposin d'una manera alternativa d'obtenir els certificats necessaris (per exemple, des d'un conjunt de certificats dels quals es disposa prèviament)¹⁰.

Es pot incloure el certificat del signatari, tot i que aquest recull de certificats és genèric per a tots els signataris del document que es relacionen dins del camp `signerInfos`. Cas d'emprar certificats d'atribut, cal que no siguin de la versió1, ja que estan obsolets.

- `crls`, que també és un camp opcional, és un recull d'informació sobre les revocacions dels certificats, d'acord amb el tipus `RevocationInfoChoices`. L'objectiu és que reculli informació suficient per determinar si els certificats que hi ha en el camp `certificates` són vàlids, però aquesta correlació entre certificats i revocacions a la pròpia signatura no és imprescindible. La llista de revocació de certificats (`CertificatesRevocationList`) és la principal font d'informació sobre l'estat de revocació (o suspensió) dels certificats. Hi pot haver més CRLs de les necessàries, o menys de les necessàries, de forma que és possible que l'aplicació que verifiqui una signatura electrònica hagi de tenir la capacitat d'obtenir llistes en línia.
- `signerInfos`, que és un camp obligatori, incorpora un recull d'informació sobre cada signatari del document, d'acord amb el tipus estructurat anomenat `SignerInfos`. Hi pot haver qualsevol nombre de signataris, incloent-hi zero. El fet que apareguin diferents signataris no implica cap flux concret de signatura, ni aporta cap informació sobre l'ordre de les signatures.

Com que cada signatari pot utilitzar una tècnica de signatura digital diferent i futures especificacions de CMS podrien actualitzar o canviar la sintaxi CMS, totes les implementacions han d'oferir un tractament adequat de les versions de `SignerInfo` no implementades. A més, ja que totes les implementacions no suportaran tots els algorismes de signatura possibles, han d'oferir un tractament adequat dels algorismes de signatura no implementats quant se'ls trobin, per exemple, indicant el tipus d'error corresponent, o les limitacions de la implementació de CMS.

¹⁰ Aquest és el cas, per exemple, d'algunes aplicacions de creació de signatura electrònica en l'àmbit de la telefonia mòbil, per tal d'estalviar ample de banda i en funció de les limitacions d'espai del dispositiu mòbil.

3.1 La informació del contingut encapsulat dins de la signatura electrònica

El tipus estructurat `EncapsulatedContentInfo` representa el contingut signat, mitjançant la següent sintaxi:

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

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

Els camps del tipus `EncapsulatedContentInfo` tenen els significats següents.

- `eContentType` és un identificador d'objecte, que especifica de forma única el tipus de contingut. Per exemple, és habitual que el tipus de contingut sigui `id-dades`, amb OID 1.2.840.113549.1.7.1.
- `eContent`, que és un camp opcional, és el contingut en sí mateix representat mitjançant una cadena d'octets¹¹. Per exemple, podria ser un text com ara "Sol·licito permís per a la crema de restes de jardineria", o un fitxer en Microsoft Word.

Resulta habitual que l'`eContentType` sigui el tipus de contingut "dades", que es refereix a qualsevol cadena d'octets, com pot ser un text en ASCII; la interpretació del contingut és responsabilitat de l'aplicació que gestiona les dades signades, ja que aquestes cadenes no cal que tinguin cap estructura interna (tot i que poden contenir la seva pròpia definició de ASN.1 o una altra estructura).

El següent identificador d'objecte identifica al tipus de contingut "dades":

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

L'especificació tècnica de correu electrònic segur d'Internet (S/MIME) empra aquest tipus de contingut ("id-data") per identificar el contingut signat o xifrat, que es codifica en MIME. L'ús d'aquest identificador de contingut s'especifica en l'especificació tècnica RFC 2311 per a S/MIME v2 i en l'especificació tècnica RFC 3851 per a S/MIME v3.1.

¹¹ Sense que sigui necessari que estigui codificat en DER.

El tipus de contingut “dades” normalment s’encapsula dins del tipus de contingut “dades signades”, “dades dins d’un sobre”, “dades resumides”, “dades xifrades” o “dades autenticades”.

En el cas d’una signatura electrònica, el tipus de contingut “dades” es pot emprar per representar la informació signada, bé a dins de la signatura o bé per referir-se a un document extern per al qual es genera una signatura electrònica.

El tipus de contingut “dades” és la forma més recomanada per referir-se a documents signats, sent obligatori el seu ús, com hem indicat anteriorment, en aplicacions de correu electrònic segur, que incorporen signatures electròniques.

Quan no es fa constar el camp `eContent` dins del tipus `EncapsulatedContentInfo`, llavors parlem de “signatures externes” (freqüentment referides per les expressions angleses “dettached signatures” o “external signatures”).

Quan hi ha signatures externes, el contingut que se signa no s’inclou de cap manera dins del `EncapsulatedContentInfo`. Tot i això, el valor de la signatura es calcula igualment i el valor del camp `eContentType` s’assigna com si el valor del camp `eContent` hi fos present igualment¹².

En el cas, indesitjable¹³, que no hi hagi cap signatari, el valor del tipus `EncapsulatedContentInfo` que se “signa” esdevé irrellevant. En aquest cas, el tipus de contingut dins del `EncapsulatedContentInfo` que se signa ha de ser necessàriament `id-data`, i el camp `eContent` s’ha d’ometre.

3.2 Els certificats digitals dins de la signatura electrònica

El tipus estructurat `CertificateSet` proporciona un conjunt de certificats, mitjançant la següent sintaxi:

```
CertificateSet ::= SET OF CertificateChoices  
  
CertificateChoices ::= CHOICE {  
    certificate Certificate,  
    extendedCertificate [0] IMPLICIT ExtendedCertificate, -- Obsolete
```

¹² L’únic problema important que pot generar això és que ningú no faci una associació entre el document extern signat i la pròpia signatura, ja que en aquest cas no es podrà determinar què va ser signat, i aleshores la signatura no tindria valor.

¹³ Cas que, com veurem posteriorment, es troba prohibit en relació amb la creació de signatures electròniques amb efecte legal.

```
v1AttrCert [1] IMPLICIT AttributeCertificateV1,-- Obsolete
v2AttrCert [2] IMPLICIT AttributeCertificateV2,
other [3] IMPLICIT OtherCertificateFormat }
```

```
OtherCertificateFormat ::= SEQUENCE {
    otherCertFormat OBJECT IDENTIFIER,
    otherCert ANY DEFINED BY otherCertFormat }
```

L'objectiu del tipus és que sigui suficient per mostrar la ruta¹⁴ de certificació que existeix entre una autoritat de certificació arrel i tots els certificats de signatari amb el qual està associat. Es pot donar el cas que hi hagi més certificats dels necessaris, o menys.

El tipus `CertificateChoices` pot contenir un certificat extens PKCS #6, un certificat X.509, un certificat d'atributs X.509 versió 1, un certificat d'atributs X.509 versió 2, o qualsevol altre format de certificat.

Els certificats extensos PKCS #6 i els certificats d'atributs X.509 versió 1 es consideren obsolets, però tot i que es recomana no utilitzar-los, cal que les aplicacions els interpretin per garantir la compatibilitat amb aplicacions més antigues.

L'especificació CMS preveu l'ús del camp opcional `OtherCertificateFormat` per suportar qualsevol altre format de certificat, que pugui aparèixer en el futur.

3.3 La informació de revocació dins de la signatura electrònica

El tipus estructurat `RevocationInfoChoices` proporciona un conjunt d'informacions alternatives d'estats de revocació, mitjançant la següent sintaxi:

```
RevocationInfoChoices ::= SET OF RevocationInfoChoice
```

```
RevocationInfoChoice ::= CHOICE {
    crl CertificateList,
    other [1] IMPLICIT OtherRevocationInfoFormat }
```

```
OtherRevocationInfoFormat ::= SEQUENCE {
```

¹⁴ Algunes aplicacions poden imposar límits superiors en la longitud de la ruta de certificació (com per exemple un màxim de 4 certificats), o requerir que hi hagi certa relació entre els subscriptors i els emissors de certificats dins d'una ruta de certificació.

```
otherRevInfoFormat OBJECT IDENTIFIER,  
otherRevInfo ANY DEFINED BY otherRevInfoFormat }
```

L'objectiu és que el tipus contingui informació suficient per determinar si els certificats o els certificats d'atributs amb els que està associat estan revocats. Es pot donar el cas de que hi hagi més informació d'estats de revocació de la necessària, o per contra, menys de la necessària.

Les fonts principals d'obtenció d'informació sobre estats de revocació són les Llistes de certificats revocats (CRLs) X.509, però hi pot haver altres fonts d'informació de revocació de certificats de diferent format, que s'inclouran en el tipus `OtherRevocationInfoFormat`, sense que calgui modificar l'especificació CMS (per exemple Online Certificate Status Protocol – OCSP¹⁵).

El camp `CertificateList` pot contenir una llista de revocació de certificats (CRL), una llista de revocació d'autoritats de certificació (ARL), una llista "Delta CRL", o una llista de revocació de certificats d'atributs, ja que totes aquestes llistes comparteixen una sintaxi comú.

La definició de `CertificateList` és la mateixa que aquella especificada en la Recomanació X.509, perfilada posteriorment per diverses RFC de l'IETF.

3.4 La informació del signatari

El tipus estructurat `SignerInfo` representa la informació de cada signatari, mitjançant la següent sintaxi:

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

¹⁵ L'especificació tècnica ETSI TS 101 733 ha establert un atribut específic per descriure les referències als mecanismes de revocació OCSP, que recomana emprar enlloc del tipus `OtherRevocationInfoFormat`. Per a la resta de mecanismes possibles de revocació que es puguin oferir en el futur, es continua recomanant l'ús del tipus `OtherRevocationInfoFormat`.

```
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
```

Els camps del tipus `SignerInfo` tenen els següents significats:

- `version` indica el número de versió de la sintaxi CMS emprada. Si a `SignerIdentifier` tenim l'emissor del certificat del signatari i el corresponent número de sèrie (`issuerAndSerialNumber`) aleshores la versió ha de ser 1, mentre que si hi tenim un identificador de clau del subscriptor (`subjectKeyIdentifier`), aleshores la versió ha de ser 3.
- `sid` especifica el certificat del signatari (i per tant la clau pública del signatari). El destinatari necessita la clau pública del signatari per verificar la signatura. El `SignerIdentifier` permet dues alternatives per especificar la clau pública del signatari. La primera, `issuerAndSerialNumber`, identifica el certificat del signatari mitjançant el nom de l'emissor del certificat del signatari i el número de sèrie d'aquest certificat; l'altra, el `subjectKeyIdentifier` identifica el certificat del signatari mitjançant un identificador de clau de subscriptor.

Quan es fa referència a un certificat X.509, el identificador de la clau coincideix amb el valor de l'extensió `subjectKeyIdentifier` del certificat del signatari. Quan es fa referència a altres formats de certificats (per exemple, un certificat d'atributs), els documents que especifiquen el format del certificat i el seu ús d'acord amb l'especificació CMS han d'incloure detalls de com relacionar el identificador de la clau del signatari amb el camp corresponent del certificat. Les implementacions han de suportar el tractament de les dues formes del tipus estructurat `SignerIdentifier`.

Quan es genera un `SignerIdentifier`, les implementacions han de poder oferir suport a una sola de les seves opcions (`issuerAndSerialNumber` o `subjectKeyIdentifier`), i sempre emprar la mateixa, o poden emprar les

dues de forma arbitrària. En qualsevol cas, `subjectKeyIdentifier` s'ha d'emprar per referir-se a una clau pública continguda en un certificat que no sigui X.509.

- `digestAlgorithm` identifica l'algorisme de resum del document, i qualsevol paràmetre associat emprat pel signatari. El resum del document es calcula sobre el contingut que s'ha de signar, o sobre el contingut juntament amb els atributs. L'algorisme de resum del document ha d'estar entre els llistats en el camp `digestAlgorithms` del `SignerData` corresponent, ja que les implementacions podrien fallar al validar signatures que es calculin amb un algorisme de resum que no estigui inclòs en aquest conjunt.
- `signedAttrs` és un recull d'atributs signats i, tot i que es defineix com un camp opcional, sempre s'ha d'incloure quan el tipus de contingut del valor `EncapsulatedContentInfo` que s'ha de signar no és "dades" (`id-data`).

Si aquest camp existeix, ha de contenir com a mínim dos tipus d'atributs: un del tipus `content-type` que tingui com a valor el mateix que el del contingut encapsulat que s'ha de signar (`EncapsulatedContentInfo`), i un del tipus `message-digest`, que tingui com a valor el resum criptogràfic del contingut.

- `signatureAlgorithm` identifica l'algorisme de la signatura i qualsevol paràmetre associat emprat pel signatari per generar la signatura digital.
- `signature` és el resultat de la generació de la signatura digital, emprant el resum del document i la clau privada del signatari. Els detalls concrets de la signatura dependran de l'algorisme de signatura emprat.
- `unsignedAttrs` és un recull d'atributs no signats, i és un camp opcional. Un exemple n'és la contrasignatura, que després analitzarem.

Els camps del tipus `SignedAttribute` i `UnsignedAttribute`, tenen els següents significats:

- `attrType` ens indica el tipus d'atribut i és un identificador d'objecte.
- `attrValues` és un conjunt de valors que conté l'atribut. El tipus de cada valor del conjunt es pot determinar de forma unívoca per `attrType` i s'hi poden posar restriccions respecte al nombre d'elements del conjunt.

3.5 Els atributs bàsics de la signatura electrònica

3.5.1 El tipus de contingut signat

El tipus d'atribut `content-type` especifica el tipus de contingut que es signa, i sempre és un atribut signat; per tant, sempre que tinguem atributs signats, un d'ells ha de ser l'atribut `content-type` dins de `signerInfos`, i el seu valor ha de

correspondre amb el valor del camp `eContentType` que trobem dins del camp obligatori `encapContentInfo` de les dades signades.

Els següents identificadors d'objecte identifiquen l'atribut `content-type`:

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

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

Encara que a la sintaxi CMS qualsevol atribut es defineix com un conjunt de valors (SET OF `AttributeValue`), el tipus de contingut (`content-type`) ha de tenir un valor únic i ha de ser diferent a zero.

3.5.2 El resum criptogràfic del document signat

L'atribut resum criptogràfic (`MessageDigest`) especifica el resum criptogràfic de la cadena d'octets que es signa del camp `eContent` i que trobem dins del camp `encapContentInfo` de les dades signades. Ha de ser sempre un atribut signat i es calcula emprant l'algorisme de resum del signatari del document.

Sempre que tinguem qualsevol atribut signat, com a mínim un ha de ser l'atribut signat "resum criptogràfic".

Els següents identificadors d'objecte identifiquen l'atribut "resum criptogràfic"

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

```
MessageDigest ::= OCTET STRING
```

Encara que a la sintaxi CMS qualsevol atribut es defineix com un conjunt de valors (SET OF `AttributeValue`), el resum criptogràfic (`messageDigest`) ha de tenir un valor únic i ha de ser diferent a zero.

4. Els atributs opcionals de la signatura electrònica CMS

4.1 La data i l'hora de la signatura electrònica

El tipus atribut “data i hora de signatura” (`SigningTime`) especifica el moment en què, suposadament, el signatari ha realitzat el procés de signar. Ha de ser sempre un atribut signat.

El següent identificador d'objecte identifica l'atribut “data i hora de signatura”:

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

```
SigningTime ::= Time
```

```
Time16 ::= CHOICE {
    utcTime UTCTime,
    generalizedTime GeneralizedTime }
```

Les dates que van de l'1 de gener de 1950 al 31 de desembre de 2049 (ambdues incloses), han d'estar codificades segons `UTCTime`. Les dates que quedin fora d'aquest rang, han d'estar codificades segons `GeneralizedTime`.

Els temps basats en `UTCTime` han d'expressar-se d'acord amb Temps Universal Coordinat (GMT – Hora Meridià de Greenwich), ha d'incloure els segons, encara que el seu nombre sigui zero, i tenir el següent format `YYMMDDHHMMSSZ`, posant zeros a hores, minuts i segons, quan vulguem representar mitjanit (`YYMMDD000000Z`). La interpretació del segle està implícita: si “any” (YY) és més gran o igual a 50, l'any s'ha d'interpretar com 19YY, i quan és més petit que 50, com 20YY.

Els temps basats en `GeneralizedTime` també s'han d'expressar d'acord amb Temps Universal Coordinat i han d'incloure els segons encara que el seu valor sigui zero (però mai les fraccions de segons), seguin el model `YYYYMMDDHHMMSSZ`.

L'especificació CMS no imposa cap requeriment pel que fa a la data i hora de signatura que consti dins de `SigningTime`, i la seva acceptació queda en mans del destinatari.

¹⁶ La definició de “Time” és d'especificada en la versió X.509 de 1997.

Encara que a la sintaxi CMS qualsevol atribut es defineix com un conjunt de valors (SET OF AttributeValue), la data i l'hora de la signatura (SigningTime) ha de tenir un valor únic i ha de ser diferent a zero.

4.2 La contrasignatura

L'atribut "contrasignatura" (Countersignature) especifica una o més signatures sobre la cadena d'octets de la signatura en un valor de SignerInfo dins el tipus "dades signades". El resum criptogràfic es calcula sobre la cadena d'octets, sense incloure la seva longitud o la seva etiqueta.

L'atribut "contrasignatura" signa en sèrie altres signatures, i ha de ser un atribut no signat.

El següent identificador d'objecte, identifica l'atribut "contrasignatura":

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

```
Countersignature ::= SignerInfo
```

Els valors de l'atribut "contrasignatura" per a les signatures ordinàries, tenen els mateixos significats que els valors de SignerInfo, menys en els següents casos:

- El camp Atributs Signats signedAttributes no pot contenir un atribut content-type, ja que no tenim un contingut diferent per una contrasignatura.
- El camp signedAttributes ha de contenir un atribut de resum criptogràfic, si aquest conté qualsevol altre atribut.

Un atribut de contrasignatura està definit com un conjunt de valors d'atribut (SET OF AttributeValue), i ha de contenir un o més valors.

Una contrasignatura, com que conté el tipus SignerInfo, pot contenir per sí mateixa un atribut de contrasignatura, i arribar a tenir així llargues sèries de signatures (empaquetament de signatures), que poden ser emprades per establir fluxos de signatures ordenades.

4.3 Els atributs addicionals de la signatura electrònica CMS per correu electrònic segur (S/MIME)

S/MIME (*Secure/Multipurpose Internet Mail Extensions*) és una ampliació del protocol de les extensions de correu electrònic d'Internet (MIME), que s'utilitza per incorporar-hi els següents serveis de seguretat:

- Autenticació.
- Integritat de missatges.
- Irrefutabilitat d'origen (*non-repudiation of origin*), emprant signatures electròniques.
- Confidencialitat de dades.

Tot i que S/MIME s'utilitza típicament en entorns de correu electrònic d'Internet, es pot emprar amb qualsevol mecanisme de transport que sigui capaç de tractar dades en format MIME, com per exemple el protocol HTTP pel web. Un altre exemple el trobem a l'RFC 3855, que especifica com emprar X.400 pel transport de missatges S/MIME.

Addicionalment, S/MIME es pot emprar en sistemes de transferència automàtica de missatges que signen documents generats mitjançant programari sense intervenció humana.

S/MIME es troba especificat en els següents estàndards vigents de l'IETF¹⁷:

- RFC 3850, 2004: Processament dels certificats.
- RFC 3851, 2004: Especificació de missatges S/MIME.

A més dels serveis bàsic de S/MIME, l'especificació RFC 2634 ofereix els següents serveis avançats de seguretat S/MIME:

- Rebuts de correu electrònic signats electrònicament.
- Etiquetes de seguretat, per autorització i control d'accés.
- Llistes segures de correu electrònic.
- Protecció del certificat emprat per signar.

Alguns d'aquests serveis avançats de seguretat requereixen l'addició de determinats atributs a la signatura electrònica, que presenten a continuació.

¹⁷ No s'inclouen les versions anteriors que han quedat obsoletes.

4.3.1 El certificat emprat per signar

L'atribut "certificat emprat per signar" (*SigningCertificate*) ha estat dissenyat per prevenir atacs de substitució i reexpedició de certificats, així com per permetre l'ús en la verificació de la signatura electrònica d'un conjunt restringit de certificats d'autorització.

La definició de *SigningCertificate* és la següent:

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

```
SigningCertificate ::= SEQUENCE {
    certs          SEQUENCE OF ESSCertID,
    policies       SEQUENCE OF PolicyInformation OPTIONAL }
```

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

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

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

El primer certificat identificat a la seqüència d'identificadors de certificats (*certs*) ha de ser el certificat a emprar per verificar la signatura electrònica. La codificació del camp *ESSCertID* per aquest certificat ha d'incloure obligatòriament el camp *issuerSerial*¹⁸. El certificat identificat és emprat durant el procés de verificació de la signatura electrònica, de forma que en cas que el resum (*hash*) del certificat no correspongui amb el certificat emprat per verificar la signatura, aquesta ha de ser considerada invàlida.

Si dins de *certs* hi ha més d'un certificats, el segon i següents certificats determinen el conjunt de certificats d'autorització que són utilitzats durant la validació

¹⁸ Aquest camp es podria ometre si altres restriccions de programari garantissin la presència del camp *issuerAndSerialNumber* dins de *SignerInfo*.

de la signatura. Resulta important notar que aquests certificats d'autorització poden ser certificats de clau pública o certificats d'atributs¹⁹. El camp `issuerSerial` del camp `ESSCertID` ha de ser present per aquests certificats²⁰.

Quan es crea un `ESSCertID`, el camp `certHash` es calcula sobre el certificat complet, incloent-hi la seva signatura, codificat en DER.

Per una altra banda, quan es codifica el camp `IssuerSerial`, `serialNumber` és el número de sèrie que identifica de forma única el certificat. En cas de certificats de clau pública, el camp `issuer` ha de contenir només el nom de l'emissor del certificat, codificat mitjançant el subtipus `directoryName` del tipus `GeneralNames`; mentre que en cas de certificats d'atributs, el camp `issuer` ha de coincidir amb el contingut del camp `issuer` del certificat d'atribut.

La seqüència d'informació de política (`PolicyInformation`) identifica les polítiques de certificats que el signatari manifesta apliquen al certificat, i d'acord amb les quals s'hauria de confiar en la signatura electrònica que es verifica. D'aquesta forma, aquest valor suggereix quina política hauria de ser emprada pel tercer destinatari de la signatura per construir la cadena de certificats necessària per validar la signatura.

L'atribut `SigningCertificate` sempre ha de ser un atribut signat.

Encara que a la sintaxi CMS qualsevol atribut es defineix com un conjunt de valors (`SET OF AttributeValue`), el certificat emprat per signar (`SigningCertificate`) ha de tenir un valor únic i ha de ser diferent a zero.

4.3.2 La identificació del contingut signat

La identificació del contingut (`ContentIdentifier`) és un atribut que conté una cadena d'octets que identifica de forma única el contingut.

La definició de `ContentIdentifier` és la següent:

```
id-aa-contentIdentifier OBJECT IDENTIFIER ::= { iso(1) member-body(2)
    us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 7 }
```

¹⁹ En general, no es recomana incloure en aquest atribut més que el certificat emprat per signar. En particular, l'especificació tècnica ETSI TS 101733, que identifica els requisits de signatura electrònica avançada d'acord amb la Directiva europea, recomana incloure els certificats d'atribut en un camp especial (`SignerAttributes`), en forma de rol certificat.

²⁰ Tot i que l'especificació IETF RFC 2634 considera aquest camp com opcional, d'acord amb l'especificació ETSI TS 101733 resulta obligatori per signatures avançades d'acord amb la Directiva europea.

`ContentIdentifier ::= OCTET STRING`

Un exemple d'ús d'aquest atribut és la sol·licitud d'un rebut de recepció de correu electrònic S/MIME, que l'utilitza per identificar el contingut signat en relació amb el qual es demana l'acusament de rebuda (`signedContentIdentifier`).

El contingut mínim d'aquest camp hauria de ser la concatenació d'informació específica d'identificació de l'usuari, una cadena `GeneralizedTime` i un número aleatori.

L'atribut `ContentIdentifier` sempre ha de ser un atribut signat.

Encara que a la sintaxi CMS qualsevol atribut es defineix com un conjunt de valors (`SET OF AttributeValue`), la identificació del contingut (`ContentIdentifier`) ha de tenir un valor únic i ha de ser diferent a zero.

4.3.3 Les pistes sobre el contingut signat

Les pistes sobre el contingut (`ContentHints`) és un atribut que descriu el contingut signat que es troba a l'interior d'un missatge multicapa; és a dir, un missatge que embolcalla altres missatges, com per exemple un correu electrònic que transporta un altre correu electrònic signat²¹.

El funcionament d'aquest atribut té sentit en el context de l'embolcall triple S/MIME, que es presenta en una altra secció d'aquest document.

La definició de `ContentHints` és la següent:

```
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 }
```

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

Els missatges que contenen un objecte `SignedData` que embolcalla un objecte `EnvelopedData`, que per tant fa confidencial el tipus de contingut de l'objecte interior (embolcall triple S/MIME) sempre haurien d'incloure l'atribut `ContentHints`, excepte quan el tipus de contingut sigui `data`.

²¹ Aquest és el cas de les llistes segures de correu electrònic signat.

Alguns tipus de contingut de missatges poden imposar o impedir la inclusió de l'atribut `ContentHints`: per exemple, quan es xifra un rebut signat emprant un objecte `EnvelopedData`, resulta obligatori crear un objecte `SignedData` que embolcalli l'objecte xifrat, incloent-hi un atribut signat `ContentHints` amb el camp `contentType` amb valor `id-ct-receipt`.

El camp `contentDescription` es pot emprar per proporcionar informació que el destinatari emprarà per seleccionar missatges protegits per al seu processament, com per exemple l'assumpte d'un missatge de correu electrònic. Si aquest camp, que és opcional, es troba establert, aleshores l'atribut hauria d'aparèixer a l'objecte `SignedData` que embolcalla l'objecte `EnvelopedData`, enlloc de a l'objecte `SignedData` interior.

4.3.4 La referència al contingut signat

La referència al contingut signat (`ContentReference`) és un enllaç des d'unes "dades signades" (`SignedData`) a unes altres.

Es pot emprar per enllaçar una resposta al missatge de correu electrònic original, per exemple, o per incorporar unes dades signades per referència dins d'unes altres dades.

Les primeres dades signades han d'incloure un atribut signat `ContentIdentifier`, i les segones dades signades s'enllacen amb les primeres incorporant un atribut signat `ContentReference` que conté el tipus de contingut (`ContentType`), identificador de contingut (`ContentIdentifier`) i el valor de la signatura de les primeres dades signades.

La definició de `ContentReference` és la següent:

```
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 }
```

5. Els atributs addicionals de la signatura electrònica CMS derivats de la Directiva europea (CAAdES)

Addicionalment als elements i continguts de la signatura electrònica que ja hem presentat, resulta necessari presentar l'especificació de l'ETSI TS 101733, que determina els formats de signatura electrònica d'acord amb la Directiva europea de signatura electrònica.

Aquesta especificació, que s'inscriu dins del mandat de normalització tècnica de la signatura electrònica realitzada per la Comissió Europea als organismes de normalització europeus, sota la direcció i supervisió de la Iniciativa Europea de Normalització de la Signatura Electrònica (EESSI), descriu configuracions específiques i elements addicionals de la signatura electrònica, amb els següents objectius:

1. Assegurar el compliment dels requisits jurídics de la signatura electrònica avançada i, en el seu cas, de la signatura electrònica reconeguda.
2. Garantir la possibilitat de validar legalment la signatura electrònica, fins i tot durant llargs terminis temporals.
3. Especificar els casos d'ús del segellament de data i hora dels continguts signats, les signatures electròniques i la informació amb valor d'evidència associada a les mateixes.

5.1 El certificat emprat per signar (definició alternativa)

El primer atribut addicional definit en compliment de la Directiva europea és una definició alternativa de l'atribut `SigningCertificate` presentat anteriorment.

Amb una sintaxi força similar, l'objectiu d'aquesta definició alternativa és permetre l'ús d'algorismes de resum diferents de SHA-1.

La definició de `OtherSigningCertificate` és la següent:

```
id-aa-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 }
```

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

```
OtherHash ::= CHOICE {
    sha1Hash           OtherHashValue,
    otherHash           OtherHashAlgAndValue }
```

```
OtherHashValue ::= OCTET STRING
```

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

En el moment de la redacció d'aquest document, existeix una proposta de modificació de la sintaxi de l'atribut `SigningCertificate` per corregir aquesta carència (*ESS Update: Adding CertID Algorithm Agility*), tot i que emprava una sintaxi diferent a la indicada a l'especificació de l'ETSI.

Probablement aquest atribut `OtherSigningCertificate` serà substituït pel nou atribut `SigningCertificateV2`, i per aquest motiu, no se'n recomana l'ús.

5.2 L'identificador de la política de signatura electrònica

Com veurem posteriorment, algunes signatures electròniques identifiquen de forma explícita la política d'acord amb la qual han estat creades o cal que siguin verificades.

Aquesta identificació es produeix mitjançant l'ús d'un atribut específic, que sempre ha de ser un atribut signat.

La definició de `SignaturePolicyIdentifier` és la següent:

```
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 {  
    sigPolicyId          SigPolicyId,  
    sigPolicyHash        SigPolicyHash,  
    sigPolicyQualifiers  SEQUENCE SIZE (1..MAX) OF  
                        SigPolicyQualifierInfo OPTIONAL}
```

```
SignaturePolicyImplied ::= NULL
```

El camp `sigPolicyId` conté un identificador d'objecte que identifica de forma única la versió específica de la política de signatura electrònica que resulta aplicable. La sintaxi d'aquest camp és la següent:

```
SigPolicyId ::= OBJECT IDENTIFIER
```

El camp `sigPolicyHash`, que és opcional, conté l'identificador de l'algorisme i el valor del resum criptogràfic de la política de signatura electrònica, com a forma de protecció de la integritat de la política. Quan el valor del resum es desconeix, aleshores aquest camp es pot posar a zero.

```
SigPolicyHash ::= OtherHashAlgAndValue
```

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

```
OtherHashValue ::= OCTET STRING
```

Un identificador de política de signatura pot ser qualificat amb informació addicional, d'acord amb la sintaxi i semàntica associada al corresponent identificador d'objecte del camp `sigPolicyQualifierId`. La sintaxi dels qualificadors de política de signatura electrònica és la següent:

```
SigPolicyQualifierInfo ::= SEQUENCE {  
    sigPolicyQualifierId  SigPolicyQualifierId,
```

sigQualifier ANY DEFINED BY sigPolicyQualifierId }

L'especificació ETSI defineix un conjunt de qualificadors de política de signatura:

- spuri, que conté la referència URI a la política de signatura electrònica.
- sp-user-notice, que conté un avís d'usuari que s'hauria de mostrar quan es valida la signatura electrònica.

La definició i sintaxi d'aquests qualificadors és la següent:

```
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)) }
```

5.3 La indicació del tipus de compromís del signatari

L'atribut `CommitmentTypeIndication` permet a un signatari explicitar a un verificador que, signant les dades, mostra un tipus de compromís del signatari.

Aquesta indicació es pot produir mitjançant l'ús d'un atribut específic, que en aquest cas ha de ser un atribut signat, o mitjançant la seva inclusió dins d'una política de signatura electrònica.

La definició de `CommitmentTypeIndication` és la següent:

```
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 }
```

L'especificació ETSI defineix els següents tipus de compromís:

```
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}
```

Aquests tipus de compromís tenen el següent significat:

- Prova d'origen (*proof of origin*) indica que el signatari reconeix haver creat, aprovat i enviat el missatge.
- Prova de recepció (*proof of reception*) indica que el signatari reconeix haver rebut el contingut del missatge.
- Prova de lliurament (*proof of delivery*) indica que el signatari²² que inclou aquesta indicació ha lliurat el missatge en un magatzem local accessible pel receptor del missatge.
- Prova d'enviament (*proof of sender*) indica que el signatari ha enviat el missatge (però no necessàriament que l'ha creat).
- Prova d'aprovació (*proof of approval*) indica que el signatari ha aprovat el contingut del missatge.
- Prova de creació (*proof of creation*) indica que el signatari ha creat el missatge (però no necessàriament que l'ha aprovat o enviat).

5.4 La localització del signatari

L'atribut `SignerLocation` especifica un mnemònic en relació amb una adreça associada amb el signatari, que es troba a una localització geogràfica particular, com per exemple una ciutat.

Aquest mnemònic es troba enregistrat al país en el qual es troba localitzat el signatari i s'utilitza en la provisió del servei postal (d'acord amb la Recomendació ITU-T F.1).

La definició de `SignerLocation` és la següent:

²² Típicament es tracta d'un intermediari, com per exemple un proveïdor de correu electrònic.

```
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 { -- com a mínim, n'hi haurà un
    countryName      [0]    DirectoryString OPTIONAL,
                        -- As used to name a Country in X.500
    localityName     [1]    DirectoryString OPTIONAL,
                        -- As used to name a locality in X.500
    postalAddress    [2]    PostalAddress OPTIONAL }
```

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

L'atribut `SignerLocation` és sempre un atribut signat.

5.5 Els atributs del signatari

L'atribut `SignerAttributes` especifica atributs addicionals del signatari, com per exemple un rol.

Aquests atributs poden ser de dos tipus:

- Atributs al·legats pel signatari.
- Atributs del signatari certificats.

La definició de `SignerAttributes` és la següent:

```
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
```

L'atribut `SignerAttributes` és sempre un atribut signat.

5.6 El segell de data i hora sobre el contingut

L'atribut `ContentTimeStamp` conté el segellament de la data i l'hora del contingut abans de signar-lo, demostrant que existia el contingut abans de ser signat.

La definició de `ContentTimeStamp` és la següent:

```
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
```

El valor del camp `messageImprint` del `TimeStampToken`, com es descriu a l'especificació IETF RFC 3161, serà el valor del resum criptogràfic del camp `eContent`, dins del tipus `encapContentInfo` de l'objecte `SignedData`.

L'atribut `ContentTimeStamp` és sempre un atribut signat.

5.7 El segell de data i hora sobre la signatura

L'atribut `SignatureTimeStampToken` conté el segellament de la data i l'hora del valor de la signatura (d'un signatari concret), demostrant que la signatura existia en un moment concret del temps. Poden existir diverses instàncies d'aquest atribut en relació amb la mateixa signatura, de diverses Entitats de Segellament de Data i Hora.

La definició de `SignatureTimeStampToken` és la següent:

```
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 }
```

```
SignatureTimeStampToken ::= TimeStampToken
```

El valor del camp `messageImprint` del `TimeStampToken`, com es descriu a l'especificació IETF RFC 3161, serà el valor del resum criptogràfic del camp `signature` de l'objecte `SignedData`.

L'atribut `SignatureTimeStampToken` és sempre un atribut no signat pel signatari.

5.8 Les referències completes dels certificats

L'atribut `CompleteCertificateReferences` conté referències del conjunt de certificats d'Entitat de Certificació que han estat emprats per validar una signatura electrònica, amb exclusió del certificat del signatari; és a dir, indica la cadena de certificats considerar vàlida per verificar la signatura²³.

Només pot existir una instància d'aquest atribut dins de la signatura electrònica.

La definició de `CompleteCertificateReferences` és la següent:

```
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 }
```

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

Com es pot veure a la definició del tipus `OtherCertID`, les referències als certificats són els resums criptogràfics corresponents.

L'atribut `CompleteCertificateReferences` és sempre un atribut no signat pel signatari.

5.9 Les referències completes de la informació de revocació de certificats

L'atribut `CompleteRevocationReferences` conté referències del conjunt de llistes de revocació de certificats o de respostes OCSP que ha estat emprat en la validació dels certificats de la cadena de certificació corresponent a la signatura electrònica.

Només pot existir una instància d'aquest atribut dins de la signatura electrònica.

Aquest atribut es pot emprar per demostrar que el verificador de la signatura electrònica ha aplicat la diligència deguda en relació amb la comprovació de la signatura electrònica, i que serà capaç de recuperar aquesta informació des del lloc en que es trobi emmagatzemada.

La definició de `CompleteRevocationReferences` és la següent:

²³ Els valors dels certificats no s'inclouen en aquest atribut, però existeix un altre atribut específic que permet guardar-los.

```
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 }
```

```
CompleteRevocationRefs ::= SEQUENCE OF CrlOcspRef
```

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

L'atribut ha de contenir un al menys `CrlOcspRef` pel certificat emprat per signar, seguit de tants `CrlOcspRef` com certificats es trobin indicats a l'atribut `CompleteCertificateRefs`, i en el mateix ordre.

Haurien d'existir tantes referències a la informació de revocació com certificats formen part de la cadena de certificació emprada per validar la signatura, amb excepció del certificat arrel²⁴.

```
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 }
```

²⁴ La verificació del certificat arrel es realitza per altres mecanismes diferents a la informació de revocació que s'emet l'Entitat de Certificació arrel a si mateixa. Típicament els mecanismes per confiar en el certificat arrel són aliens a la pròpia xarxa, i es produeixen "fora de línia".

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

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

`crlListID` conté totes les referències (resums criptogràfics) a les llistes de revocació de certificats necessàries per validar la signatura electrònica. Les llistes es poden guardar o ser recuperades posteriorment d'Internet.

Quan es crea el `CrlValidatedID`, el resum criptogràfic de la CRL es calcula sobre tota la CRL, incloent-hi la seva signatura, codificada en DER.

Opcionalment, el tipus `CrlIdentifier` identifica cada llista de revocació de certificats mitjançant el nom del seu emissor i el moment de la seva emissió, que ha de correspondre amb el camp `thisUpdate` de la CRL i, quan es troba present, amb el camp `crlNumber` de la CRL corresponent. Aquest camp hauria de ser present excepte quan les informacions que conté es trobin enregistrades en un altre lloc.

En cas que la llista identificada sigui una CRL Delta – que només conté una part de la informació total de revocació – aleshores cal incloure també la resta de CRLs, de forma que tota la informació de revocació sigui referida²⁵.

`OcspListID` conté totes les referències a les respostes OCSP de validació dels certificats.

El tipus `OcspIdentifier` identifica cada resposta OCSP mitjançant el nom del seu emissor i el seu moment d'emissió, que ha de correspondre amb el moment indicat al camp `producedAt` de la resposta OCSP corresponent.

Ja que pot resultar necessari distingir entre dues respostes OCSP rebudes dins del mateix segon, opcionalment es pot emprar el resum criptogràfic de les respostes.

L'atribut també es pot emprar per identificar les referències d'informació de revocació de cadenes de certificació d'Entitats de Segellament de Data i Hora que emeten segells de data i hora criptogràfics.

En aquest cas, l'atribut serà afegit al tipus `SignedData` del segell de temps corresponent.

²⁵ El nombre necessari de CRLs dependrà de la política de publicació de les llistes que apliqui l'Entitat de Certificació. Si per exemple emet una CRL completa una dia cada setmana, i la resta de dies emet CRLs Delta, aleshores caldrà registrar referències a totes les CRL Delta fins arribar a la CRL completa, que serà també inclosa.

L'atribut `CompleteRevocationReferences` és sempre un atribut no signat pel signatari.

5.10 Les referències completes dels certificats d'atributs

L'atribut `AttributeCertificateReferences` conté referències del conjunt de certificats d'Entitat d'Atributs que han estat emprats per validar un certificat d'atributs del signatari; és a dir, indica la cadena de certificats considerar vàlida per verificar l'atribut certificat²⁶.

Aquest atribut s'utilitza, conseqüentment, únicament quan la signatura electrònica conté certificats d'atributs, per exemple, en un camp `CertifiedRole`.

Només pot existir una instància d'aquest atribut dins de la signatura electrònica.

La definició de `CompleteCertificateReferences` és la següent:

```
id-aa-ets-attrCertificateRefs OBJECT IDENTIFIER ::= { iso(1) member-body(2)
    us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 44 }
```

```
AttributeCertificateRefs ::= SEQUENCE OF OtherCertID
```

Com es pot veure a la definició del tipus `OtherCertID`, les referències als certificats són els resums criptogràfics corresponents.

L'atribut `AttributeCertificateReferences` és sempre un atribut no signat pel signatari.

5.11 Les referències completes de la informació de revocació d'atributs

L'atribut `AttributeRevocationReferences` conté referències del conjunt de llistes de revocació de certificats o de respostes OCSP que ha estat emprat en la validació dels certificats d'atributs del signatari.

Aquest atribut es pot emprar per demostrar que el verificador de la signatura electrònica ha aplicat la diligència deguda en relació amb la comprovació dels certificats d'atributs, i que serà capaç de recuperar aquesta informació des del lloc en que es trobi emmagatzemada.

²⁶ Els valors dels certificats d'atributs no s'inclouen en aquest atribut, però existeix un altre atribut específic que permet guardar-los.

Aquest atribut s'utilitza, conseqüentment, únicament quan la signatura electrònica conté certificats d'atributs, per exemple, en un camp `CertifiedRole`, i només en cas que els certificats d'atributs siguin revocables²⁷.

Només pot existir una instància d'aquest atribut dins de la signatura electrònica.

La definició de `AttributeCertificateReferences` és la següent:

```
id-aa-ets-attrRevocationRefs OBJECT IDENTIFIER ::= { iso(1) member-body(2)
    us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 45 }
```

```
AttributeRevocationRefs ::= SEQUENCE OF CrlOcspref
```

L'atribut `AttributeRevocationReferences` és sempre un atribut no signat pel signatari.

5.12 El segell de data i hora sobre la signatura completa

L'atribut `ESCTimeStampToken` conté el segellament de data i hora de la signatura electrònica completa, per protegir-la d'un possible compromís de la clau de l'Entitat de Certificació.

Poden existir diverses instàncies d'aquest atribut dins de la signatura electrònica.

La definició de `ESCTimeStampToken` és la següent:

```
id-aa-ets-escTimeStamp OBJECT IDENTIFIER ::= { iso(1) member-body (2)
    us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 25 }
```

```
ESCTimeStampToken ::= TimeStampToken
```

El valor del camp `messageImprint` del segell de data i hora ha de resultar de la concatenació de les següents dades:

- La cadena d'octets del camp `SignatureValue` dins de `SignerInfo`.
- L'atribut `SignatureTimeStampToken`, o una marca de data i hora operada per una Entitat de Marcatge de Data i Hora.
- L'atribut `CompleteCertificateReferences`.

²⁷ Algunes Entitats d'Atributs emeten certificats d'atributs que no són revocables, ja que el seu termini de vigència és breu.

-
- L'atribut `CompleteRevocationReferences`.

L'atribut `ESCTimeStampToken` és sempre un atribut no signat pel signatari.

5.13 El segell de data i hora sobre les referències de certificats i revocacions

L'atribut `TimestampedCertsCRLs` conté el segellament de data i hora de les referències als certificats i a la informació de revocació de la signatura electrònica, per protegir la signatura de determinats compromisos de l'Entitat de Certificació.

Poden existir diverses instàncies d'aquest atribut dins de la signatura electrònica.

La definició de `TimestampedCertsCRLs` és la següent:

```
id-aa-ets-certCRLTimestamp OBJECT IDENTIFIER ::= { iso(1) member-body (2)
    us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 26 }
```

```
TimestampedCertsCRLs ::= TimeStampToken
```

El valor del camp `messageImprint` del segell de data i hora ha de resultar de la concatenació de les següents dades:

- L'atribut `CompleteCertificateReferences`.
- L'atribut `CompleteRevocationReferences`.

L'atribut `TimestampedCertsCRLs` és sempre un atribut no signat pel signatari.

5.14 Els valors dels certificats

L'atribut `CertificateValues` conté els valors dels certificats a que es refereix l'atribut `CompleteCertificateReferences`, amb excepció dels valors dels certificats d'atributs, que s'inclouen a l'atribut `SignerAttributes`.

Només pot existir una instància d'aquest atribut dins de la signatura electrònica.

La definició de `CertificateValues` és la següent:

```
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 }
```

CertificateValues ::= SEQUENCE OF Certificate

Aquest atribut pot incloure la informació dels certificats corresponents a una Entitat de Segellament de Data i Hora que ha subministrat segells de data i hora criptogràfics, quan aquests certificats no han estat inclosos dins de la signatura del segell de data i hora. En aquest cas, cas l'atribut serà afegit al SignedData del segell de data i hora corresponent.

L'atribut CertificateValues és sempre un atribut no signat pel signatari.

5.15 Els valors de les revocacions

L'atribut RevocationValues conté els valors de les llistes de revocació de certificats i de les respostes OCSP a que es refereix l'atribut CompleteRevocationReferences.

Només pot existir una instància d'aquest atribut dins de la signatura electrònica.

La definició de RevocationValues és la següent:

```
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 }
```

```
RevocationValues ::= SEQUENCE {
    crlVals          [0] SEQUENCE OF CertificateList OPTIONAL,
    ocspsVals        [1] SEQUENCE OF BasicOCSPResponse OPTIONAL,
    otherRevVals     [2] OtherRevVals OPTIONAL }
```

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

```
OtherRevValType ::= OBJECT IDENTIFIER
```

El tipus BasicOCSPResponse es defineix a la RFC 2560, de la següent forma:

```
BasicOCSPResponse ::= SEQUENCE {
    tbsResponseData  ResponseData,
```

```
signatureAlgorithm  AlgorithmIdentifier,  
signature          BIT STRING,  
certs             [0] EXPLICIT SEQUENCE OF Certificate OPTIONAL }
```

```
ResponseData ::= SEQUENCE {  
    version          [0] EXPLICIT Version DEFAULT v1,  
    responderID     ResponderID,  
    producedAt      GeneralizedTime,  
    responses       SEQUENCE OF SingleResponse,  
    responseExtensions [1] EXPLICIT Extensions OPTIONAL }
```

```
ResponderID ::= CHOICE {  
    byName          [1] Name,  
    byKey           [2] KeyHash }
```

```
KeyHash ::= OCTET STRING -- SHA-1 hash of responder's public key  
           (excluding the tag and length fields)
```

```
SingleResponse ::= SEQUENCE {  
    certID          CertID,  
    certStatus     CertStatus,  
    thisUpdate     GeneralizedTime,  
    nextUpdate     [0] EXPLICIT GeneralizedTime OPTIONAL,  
    singleExtensions [1] EXPLICIT Extensions OPTIONAL }
```

```
CertStatus ::= CHOICE {  
    Good           [0] IMPLICIT NULL,  
    Revoked       [1] IMPLICIT RevokedInfo,  
    unknown       [2] IMPLICIT UnknownInfo }
```

```
RevokedInfo ::= SEQUENCE {  
    revocationTime GeneralizedTime,  
    revocationReason [0] EXPLICIT CRLReason OPTIONAL }
```

```
UnknownInfo ::= NULL
```

Aquest atribut pot incloure la informació de revocació dels certificats corresponents a una Entitat de Segellament de Data i Hora que ha subministrat segells de data i hora criptogràfics, quan aquests certificats no han estat inclosos dins de la signatura del segell de data i hora. En aquest cas, cas l'atribut serà afegit al SignedData del segell de data i hora corresponent.

L'atribut `RevocationValues` és sempre un atribut no signat pel signatari.

5.16 El segell de data i hora d'arxiu

L'atribut `ArchiveTimestampToken` conté el segellament de data i hora de diversos camps del `SignedData`. Si els atributs `CertificateValues` i `RevocationValues` no es troben presents, aleshores caldrà afegir-los abans de calcular el segell de data i hora d'arxiu.

Poden existir diverses instàncies d'aquest atribut dins de la signatura electrònica, i de fet el nombre d'aparicions d'aquest atribut es trobarà lligat a la durada de la signatura electrònica, ja que caldrà re-segellar la signatura per garantir-ne la validesa criptogràfica al llarg del temps.

La definició de `ArchiveTimestampToken` és la següent:

```
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 }
```

```
ArchiveTimeStampToken ::= TimeStampToken
```

El valor del camp `messageImprint` del segell de data i hora ha de resultar de la concatenació de les següents dades:

- L'element `encapContentInfo` del `SignedData`.
- Els elements `Certificates` i `crls` del `SignedData`, quan es trobin presents.
- Totes les dades del `SignerInfo`, incloent-hi tots els atributs, signats i no signats.

L'atribut `ArchiveTimeStamp` és sempre un atribut no signat pel signatari.

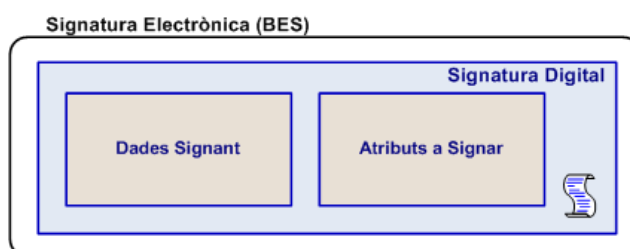
6. Els formats de la signatura electrònica CMS

La signatura electrònica és un objecte digital que, com hem vist en les seccions anteriors, presenta moltes opcions de configuració diferents: en funció de les diferents necessitats identificades.

6.1 La signatura electrònica bàsica (CADES-BES)

La signatura electrònica bàsica és el format de signatura electrònica que compleix els mínims exigits per la normativa legal de signatura electrònica derivada de la Directiva europea.

El següent gràfic mostra l'estructura de la signatura electrònica bàsica:



La signatura electrònica bàsica, representada en sintaxi CMS, es troba formada pels següents elements:

1. Les dades signades per l'usuari, com per exemple un document electrònic	Obligatori
2. El tipus de contingut signat: <code>ContentType</code>	Obligatori
3. El resum criptogràfic del missatge: <code>MessageDigest</code>	Obligatori
4. El certificat emprat per signar, en qualsevol de les dues definicions especificades: <code>ESSSigningCertificate</code> o <code>OtherSigningCertificate</code>	Obligatori
5. La data i hora al·legada de la signatura: <code>SigningTime</code>	Opcional
6. Les pistes sobre el contingut signat: <code>ContentHints</code>	Opcional
7. La identificació del contingut: <code>ContentIdentifier</code>	Opcional
8. La referència als continguts: <code>ContentReference</code>	Opcional
9. La indicació del tipus de compromís:	Opcional

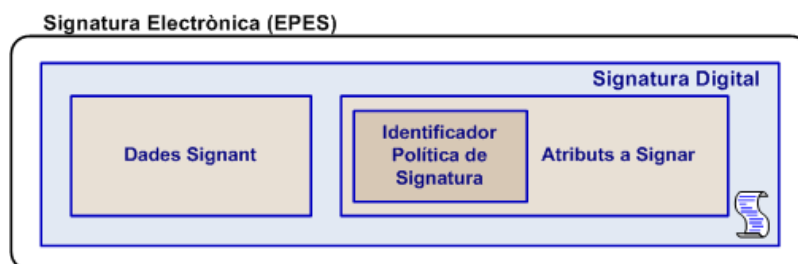
CommitmentTypeIndication	
10. La localització del signatari: <i>SignerLocation</i>	Opcional
11. Els atributs del signatari: <i>SignerAttributes</i>	Opcional
12. El segell de data i hora sobre el contingut: <i>ContentTimestamp</i>	Opcional
13. Contrasignatura: <i>Countersignature</i>	Opcional

6.2 La signatura electrònica amb política explícita (CADES-EPES)

La signatura electrònica amb política explícita afegeix, a la política bàsica, la indicació explícita de la política de signatura electrònica que resulta aplicable a la creació i verificació de la signatura electrònica.

En aquest cas resulta obligatori aplicar les normes indicades a la política per considerar una signatura vàlidament creada o verificada.

El següent gràfic mostra l'estructura de la signatura electrònica amb política explícita:



La signatura electrònica amb política explícita, representada en sintaxi CMS, es troba formada pels següents elements:

1. Les dades signades per l'usuari, com per exemple un document electrònic	Obligatori
2. El tipus de contingut signat: <i>ContentType</i>	Obligatori
3. El resum criptogràfic del missatge: <i>MessageDigest</i>	Obligatori
4. El certificat emprat per signar, en qualsevol de les dues definicions especificades: <i>ESSSigningCertificate</i> o <i>OtherSigningCertificate</i>	Obligatori

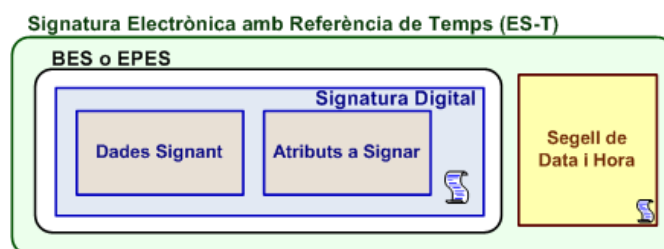
5. La data i hora al·legada de la signatura: <i>SigningTime</i>	Opcional
6. Les pistes sobre el contingut signat: <i>ContentHints</i>	Opcional
7. La identificació del contingut: <i>ContentIdentifier</i>	Opcional
8. La referència als continguts: <i>ContentReference</i>	Opcional
9. La indicació del tipus de compromís: <i>CommitmentTypeIndication</i>	Opcional
10. La localització del signatari: <i>SignerLocation</i>	Opcional
11. Els atributs del signatari: <i>SignerAttributes</i>	Opcional
12. El segell de data i hora sobre el contingut: <i>ContentTimestamp</i>	Opcional
13. Contrasignatura: <i>Countersignature</i>	Opcional
14. Identificació de la política de signatura: <i>SignaturePolicyIdentifier</i>	Obligatori

6.3 La signatura electrònica amb segell de data i hora (CADES-T)

La signatura electrònica amb segell de data i hora afegeix a una signatura electrònica un segell de data i hora, amb la finalitat de garantir l'existència de la signatura abans de la data i hora corresponent.

Aquesta garantia de data i hora es pot aportar mitjançant un element addicional a la signatura, corresponent a un segell criptogràfic de data i hora, o mitjançant una marca de data i hora, que es gestiona a banda de la signatura.

El següent gràfic mostra l'estructura de la signatura electrònica amb segell de data i hora:



La signatura electrònica amb segell de data i hora, representada en sintaxi CMS, es troba formada pels següents elements:

1. Les dades signades per l'usuari, com per exemple un document electrònic	Obligatori
2. El tipus de contingut signat: <code>ContentType</code>	Obligatori
3. El resum criptogràfic del missatge: <code>MessageDigest</code>	Obligatori
4. El certificat emprat per signar, en qualsevol de les dues definicions especificades: <code>ESSSigningCertificate</code> o <code>OtherSigningCertificate</code>	Obligatori
5. La data i hora al·legada de la signatura: <code>SigningTime</code>	Opcional
6. Les pistes sobre el contingut signat: <code>ContentHints</code>	Opcional
7. La identificació del contingut: <code>ContentIdentifier</code>	Opcional
8. La referència als continguts: <code>ContentReference</code>	Opcional
9. La indicació del tipus de compromís: <code>CommitmentTypeIndication</code>	Opcional
10. La localització del signatari: <code>SignerLocation</code>	Opcional
11. Els atributs del signatari: <code>SignerAttributes</code>	Opcional
12. El segell de data i hora sobre el contingut: <code>ContentTimestamp</code>	Opcional
13. Contrasignatura: <code>Countersignature</code>	Opcional
14. Identificació de la política de signatura: <code>SignaturePolicyIdentifier</code>	Opcional ²⁸
15. Segell de data i hora de la signatura: <code>SignatureTimeStampToken</code>	Obligatori ²⁹

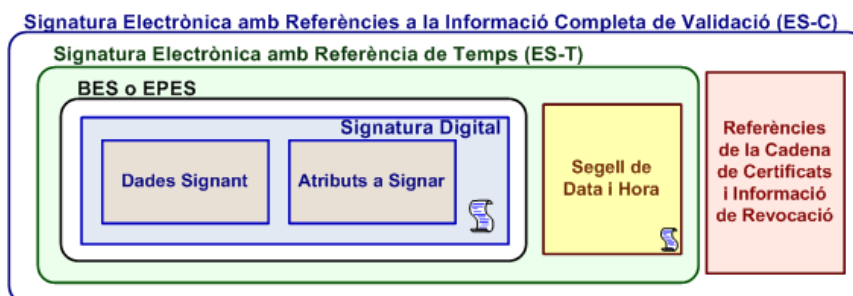
6.4 La signatura electrònica amb referències completes de dades de validació (CADES-C)

La signatura electrònica amb referències completes de dades de validació – també anomenada signatura electrònica completa – afegeix a una signatura electrònica les referències a les dades necessàries per validar la signatura, tot i que, de fet, no inclou les pròpies informacions de certificats ni de revocació.

²⁸ Depèn del tipus de signatura segellada.

²⁹ Només és obligatori quan s'utilitza segellament criptogràfic de data i hora, enlloc de marcatge de data i hora.

El següent gràfic mostra l'estructura de la signatura electrònica amb referències completes de dades de validació:



La signatura electrònica amb referències completes de dades de validació, representada en sintaxi CMS, es troba formada pels següents elements:

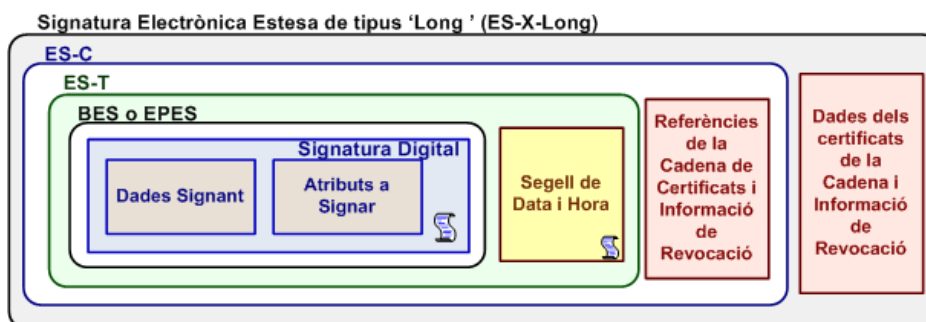
1. Les dades signades per l'usuari, com per exemple un document electrònic	Obligatori
2. El tipus de contingut signat: <code>ContentType</code>	Obligatori
3. El resum criptogràfic del missatge: <code>MessageDigest</code>	Obligatori
4. El certificat emprat per signar, en qualsevol de les dues definicions especificades: <code>ESSSigningCertificate</code> o <code>OtherSigningCertificate</code>	Obligatori
5. La data i hora al·legada de la signatura: <code>SigningTime</code>	Opcional
6. Les pistes sobre el contingut signat: <code>ContentHints</code>	Opcional
7. La identificació del contingut: <code>ContentIdentifier</code>	Opcional
8. La referència als continguts: <code>ContentReference</code>	Opcional
9. La indicació del tipus de compromís: <code>CommitmentTypeIndication</code>	Opcional
10. La localització del signatari: <code>SignerLocation</code>	Opcional
11. Els atributs del signatari: <code>SignerAttributes</code>	Opcional
12. El segell de data i hora sobre el contingut: <code>ContentTimestamp</code>	Opcional
13. Contrasignatura: <code>Countersignature</code>	Opcional

14. Identificació de la política de signatura: SignaturePolicyIdentifier	Opcional ³⁰
15. Segell de data i hora de la signatura: SignatureTimeStampToken	Obligatori ³¹
16. Referències completes de certificats: CompleteCertificateReferences	Obligatori
17. Referències completes de revocació: CompleteRevocationReferences	Obligatori

6.5 La signatura electrònica amb dades completes de validació (CADES-X Long)

La signatura electrònica amb dades completes de validació afegeix a la signatura amb referències completes de validació els valors dels certificats i de les informacions de revocació, de forma que la signatura ja conté en si mateixa totes les dades necessàries per validar la signatura, sense dependre de tercers sistemes o dipòsits que emmagatzemin aquestes informacions.

El següent gràfic mostra l'estructura de la signatura electrònica amb dades completes de validació:



La signatura electrònica amb dades completes de validació, representada en sintaxi CMS, es troba formada pels següents elements:

1. Les dades signades per l'usuari, com per exemple un document electrònic	Obligatori
--	------------

³⁰ Depèn del tipus de signatura segellada.

³¹ Només és obligatori quan s'utilitza segellament criptogràfic de data i hora, enlloc de marcatge de data i hora.

2. El tipus de contingut signat: <code>ContentType</code>	Obligatori
3. El resum criptogràfic del missatge: <code>MessageDigest</code>	Obligatori
4. El certificat emprat per signar, en qualsevol de les dues definicions especificades: <code>ESSSigningCertificate</code> o <code>OtherSigningCertificate</code>	Obligatori
5. La data i hora al·legada de la signatura: <code>SigningTime</code>	Opcional
6. Les pistes sobre el contingut signat: <code>ContentHints</code>	Opcional
7. La identificació del contingut: <code>ContentIdentifier</code>	Opcional
8. La referència als continguts: <code>ContentReference</code>	Opcional
9. La indicació del tipus de compromís: <code>CommitmentTypeIndication</code>	Opcional
10. La localització del signatari: <code>SignerLocation</code>	Opcional
11. Els atributs del signatari: <code>SignerAttributes</code>	Opcional
12. El segell de data i hora sobre el contingut: <code>ContentTimestamp</code>	Opcional
13. Contrasignatura: <code>Countersignature</code>	Opcional
14. Identificació de la política de signatura: <code>SignaturePolicyIdentifier</code>	Opcional ³²
15. Segell de data i hora de la signatura: <code>SignatureTimeStampToken</code>	Obligatori ³³
16. Referències completes de certificats: <code>CompleteCertificateReferences</code>	Obligatori
17. Referències completes de revocació: <code>CompleteRevocationReferences</code>	Obligatori
18. Valors de certificats: <code>CertificateValues</code>	Obligatori
19. Valors de revocació: <code>RevocationValues</code>	Obligatori

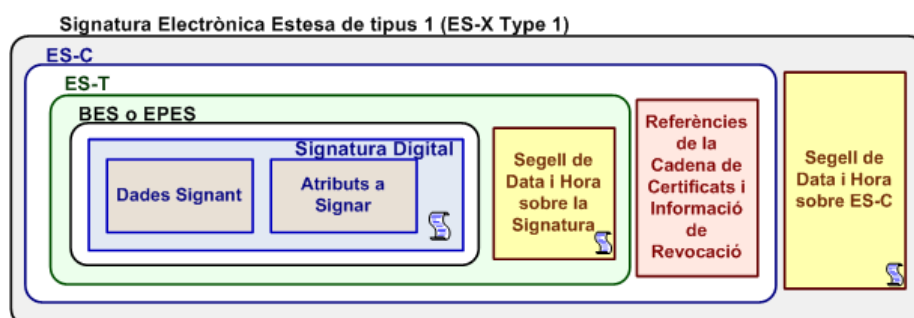
³² Depèn del tipus de signatura segellada.

³³ Només és obligatori quan s'utilitza segellament criptogràfic de data i hora, enlloc de marcatge de data i hora.

6.6 La signatura electrònica amb referències completes de dades de validació i segellada (CAES-X Type 1)

La signatura electrònica amb referències completes de dades de validació i segellada – també anomenada signatura electrònica extensa de tipus 1 – afegeix a la signatura amb referències completes de dades de validació un segell de data i hora sobre aquesta, a l'objecte de protegir la integritat i garantir l'existència de les informacions de la signatura, en especial dels atributs no signats pel signatari.

El següent gràfic mostra l'estructura de la signatura electrònica amb referències completes de dades de validació i segellada:



La signatura electrònica amb referències completes de dades de validació i segellada, representada en sintaxi CMS, es troba formada pels següents elements:

1. Les dades signades per l'usuari, com per exemple un document electrònic	Obligatori
2. El tipus de contingut signat: <code>ContentType</code>	Obligatori
3. El resum criptogràfic del missatge: <code>MessageDigest</code>	Obligatori
4. El certificat emprat per signar, en qualsevol de les dues definicions especificades: <code>ESSSigningCertificate</code> o <code>OtherSigningCertificate</code>	Obligatori
5. La data i hora al·legada de la signatura: <code>SigningTime</code>	Opcional
6. Les pistes sobre el contingut signat: <code>ContentHints</code>	Opcional
7. La identificació del contingut: <code>ContentIdentifier</code>	Opcional
8. La referència als continguts: <code>ContentReference</code>	Opcional
9. La indicació del tipus de compromís: <code>CommitmentTypeIndication</code>	Opcional

10. La localització del signatari: <i>SignerLocation</i>	Opcional
11. Els atributs del signatari: <i>SignerAttributes</i>	Opcional
12. El segell de data i hora sobre el contingut: <i>ContentTimestamp</i>	Opcional
13. Contrasignatura: <i>Countersignature</i>	Opcional
14. Identificació de la política de signatura: <i>SignaturePolicyIdentifier</i>	Opcional ³⁴
15. Segell de data i hora de la signatura: <i>SignatureTimeStampToken</i>	Obligatori ³⁵
16. Referències completes de certificats: <i>CompleteCertificateReferences</i>	Obligatori
17. Referències completes de revocació: <i>CompleteRevocationReferences</i>	Obligatori
18. Segell de data i hora de la signatura electrònica completa: <i>ESCTimeStampToken</i>	Obligatori

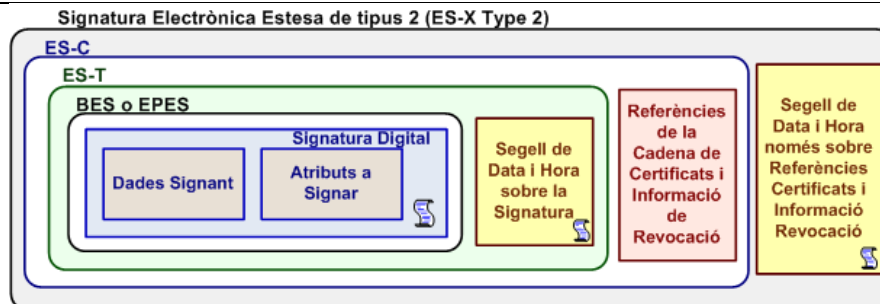
6.7 La signatura electrònica amb referències completes i segellades de dades de validació (CAES-X Type 2)

La signatura electrònica amb referències completes i segellades de dades de validació – també anomenada signatura electrònica extensa de tipus 2 – afegeix a la signatura amb referències completes de dades de validació un segell de data i hora sobre les referències de dades de validació, i no sobre la resta d'elements de la signatura, a l'objecte de protegir la integritat i garantir l'existència de d'aquestes referències.

El següent gràfic mostra l'estructura de la signatura electrònica amb referències completes i segellades de dades de validació:

³⁴ Depèn del tipus de signatura segellada.

³⁵ Només és obligatori quan s'utilitza segellament criptogràfic de data i hora, enlloc de marcatge de data i hora.



La signatura electrònica amb referències completes i segellades de dades de validació, representada en sintaxi CMS, es troba formada pels següents elements:

1. Les dades signades per l'usuari, com per exemple un document electrònic	Obligatori
2. El tipus de contingut signat: <code>ContentType</code>	Obligatori
3. El resum criptogràfic del missatge: <code>MessageDigest</code>	Obligatori
4. El certificat emprat per signar, en qualsevol de les dues definicions especificades: <code>ESSSigningCertificate</code> o <code>OtherSigningCertificate</code>	Obligatori
5. La data i hora al·legada de la signatura: <code>SigningTime</code>	Opcional
6. Les pistes sobre el contingut signat: <code>ContentHints</code>	Opcional
7. La identificació del contingut: <code>ContentIdentifier</code>	Opcional
8. La referència als continguts: <code>ContentReference</code>	Opcional
9. La indicació del tipus de compromís: <code>CommitmentTypeIndication</code>	Opcional
10. La localització del signatari: <code>SignerLocation</code>	Opcional
11. Els atributs del signatari: <code>SignerAttributes</code>	Opcional
12. El segell de data i hora sobre el contingut: <code>ContentTimestamp</code>	Opcional
13. Contrasignatura: <code>Countersignature</code>	Opcional
14. Identificació de la política de signatura: <code>SignaturePolicyIdentifier</code>	Opcional ³⁶
15. Segell de data i hora de la signatura:	Obligatori ³⁷

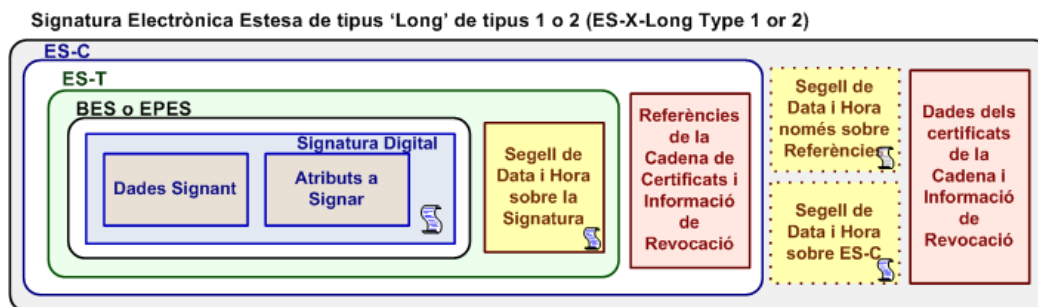
³⁶ Depèn del tipus de signatura segellada.

SignatureTimeStampToken	
16. Referències completes de certificats: CompleteCertificateReferences	Obligatori
17. Referències completes de revocació: CompleteRevocationReferences	Obligatori
18. Segell de data i hora de sobre les referències de certificats i revocacions: TimestampedCertsCRLs	Obligatori

6.8 La signatura electrònica amb dades completes de validació i segellada (CADES-X Long Type 1)

La signatura electrònica amb dades completes de validació i segellada afegeix a la signatura amb dades completes de validació un segell de data i hora sobre aquesta, a l'objecte de protegir la integritat i garantir l'existència de les informacions de la signatura, en especial dels atributs no signats pel signatari.

El següent gràfic mostra l'estructura de la signatura electrònica amb dades completes de validació i segellades:



La signatura electrònica amb dades completes de validació i segellada, representada en sintaxi CMS, es troba formada pels següents elements:

1. Les dades signades per l'usuari, com per exemple un document electrònic	Obligatori
2. El tipus de contingut signat: <i>ContentType</i>	Obligatori
3. El resum criptogràfic del missatge: <i>MessageDigest</i>	Obligatori

³⁷ Només és obligatori quan s'utilitza segellament criptogràfic de data i hora, enlloc de marcatge de data i hora.

4. El certificat emprat per signar, en qualsevol de les dues definicions especificades: <code>ESSSigningCertificate</code> o <code>OtherSigningCertificate</code>	Obligatori
5. La data i hora al·legada de la signatura: <code>SigningTime</code>	Opcional
6. Les pistes sobre el contingut signat: <code>ContentHints</code>	Opcional
7. La identificació del contingut: <code>ContentIdentifier</code>	Opcional
8. La referència als continguts: <code>ContentReference</code>	Opcional
9. La indicació del tipus de compromís: <code>CommitmentTypeIndication</code>	Opcional
10. La localització del signatari: <code>SignerLocation</code>	Opcional
11. Els atributs del signatari: <code>SignerAttributes</code>	Opcional
12. El segell de data i hora sobre el contingut: <code>ContentTimestamp</code>	Opcional
13. Contrasignatura: <code>Countersignature</code>	Opcional
14. Identificació de la política de signatura: <code>SignaturePolicyIdentifier</code>	Opcional ³⁸
15. Segell de data i hora de la signatura: <code>SignatureTimeStampToken</code>	Obligatori ³⁹
16. Referències completes de certificats: <code>CompleteCertificateReferences</code>	Obligatori
17. Referències completes de revocació: <code>CompleteRevocationReferences</code>	Obligatori
18. Segell de data i hora de la signatura electrònica completa: <code>ESCTimeStampToken</code>	Obligatori
19. Valors de certificats: <code>CertificateValues</code>	Obligatori
20. Valors de revocació: <code>RevocationValues</code>	Obligatori

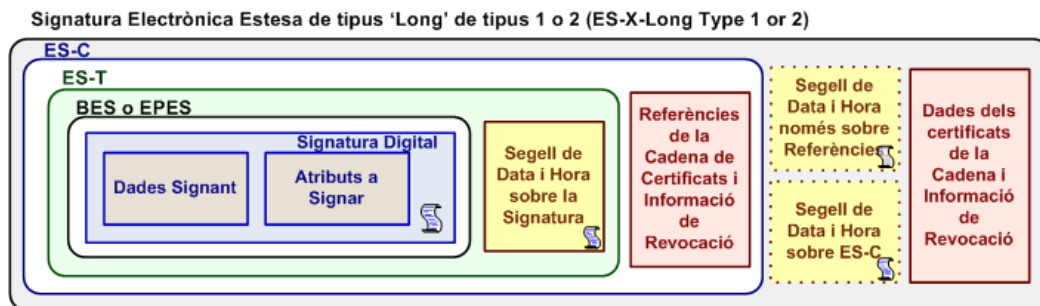
³⁸ Depèn del tipus de signatura segellada.

³⁹ Només és obligatori quan s'utilitza segellament criptogràfic de data i hora, enlloc de marcatge de data i hora.

6.9 La signatura electrònica amb dades completes i segellades de validació (CADES-X Long Type 2)

La signatura electrònica amb dades completes i segellades de validació afegeix a la signatura amb dades completes de validació un segell de data i hora sobre les referències de dades de validació, i no sobre la resta d'elements de la signatura, a l'objecte de protegir la integritat i garantir l'existència de d'aquestes referències.

El següent gràfic mostra l'estructura de la signatura electrònica amb dades completes i segellades de validació:



La signatura electrònica amb dades completes i segellades de validació, representada en sintaxi CMS, es troba formada pels següents elements:

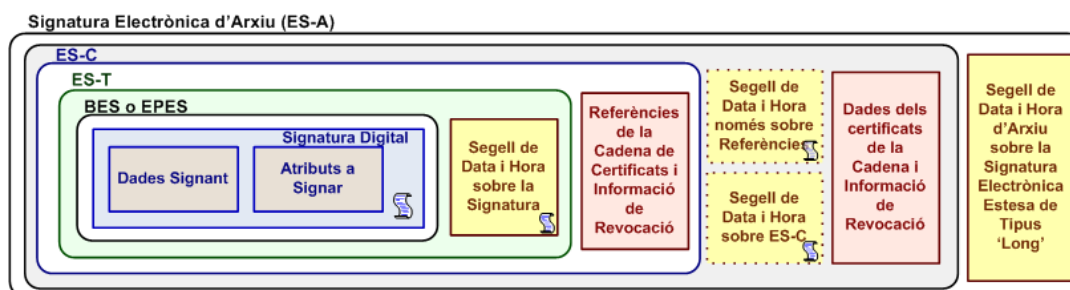
1. Les dades signades per l'usuari, com per exemple un document electrònic	Obligatori
2. El tipus de contingut signat: <code>ContentType</code>	Obligatori
3. El resum criptogràfic del missatge: <code>MessageDigest</code>	Obligatori
4. El certificat emprat per signar, en qualsevol de les dues definicions especificades: <code>ESSSigningCertificate</code> o <code>OtherSigningCertificate</code>	Obligatori
5. La data i hora al·legada de la signatura: <code>SigningTime</code>	Opcional
6. Les pistes sobre el contingut signat: <code>ContentHints</code>	Opcional
7. La identificació del contingut: <code>ContentIdentifier</code>	Opcional
8. La referència als continguts: <code>ContentReference</code>	Opcional
9. La indicació del tipus de compromís: <code>CommitmentTypeIndication</code>	Opcional
10. La localització del signatari: <code>SignerLocation</code>	Opcional

11. Els atributs del signatari: <code>SignerAttributes</code>	Opcional
12. El segell de data i hora sobre el contingut: <code>ContentTimestamp</code>	Opcional
13. Contrasignatura: <code>Countersignature</code>	Opcional
14. Identificació de la política de signatura: <code>SignaturePolicyIdentifier</code>	Opcional ⁴⁰
15. Segell de data i hora de la signatura: <code>SignatureTimeStampToken</code>	Obligatori ⁴¹
16. Referències completes de certificats: <code>CompleteCertificateReferences</code>	Obligatori
17. Referències completes de revocació: <code>CompleteRevocationReferences</code>	Obligatori
18. Segell de data i hora de sobre les referències de certificats i revocacions: <code>TimestampedCertsCRLs</code>	Obligatori
19. Valors de certificats: <code>CertificateValues</code>	Obligatori
20. Valors de revocació: <code>RevocationValues</code>	Obligatori

6.10 La signatura electrònica d'arxiu (CADES-A)

La signatura electrònica d'arxiu afegeix a diversos formats de signatura electrònica extensa un segell de data i hora sobre la signatura.

El següent gràfic mostra l'estructura de la signatura electrònica d'arxiu:



⁴⁰ Depèn del tipus de signatura segellada.

⁴¹ Només és obligatori quan s'utilitza segellament criptogràfic de data i hora, enlloc de marcatge de data i hora.

La signatura electrònica d'arxiu, representada en sintaxi CMS, es troba formada pels següents elements:

1. Les dades signades per l'usuari, com per exemple un document electrònic	Obligatori
2. El tipus de contingut signat: <code>ContentType</code>	Obligatori
3. El resum criptogràfic del missatge: <code>MessageDigest</code>	Obligatori
4. El certificat emprat per signar, en qualsevol de les dues definicions especificades: <code>ESSSigningCertificate</code> o <code>OtherSigningCertificate</code>	Obligatori
5. La data i hora al·legada de la signatura: <code>SigningTime</code>	Opcional
6. Les pistes sobre el contingut signat: <code>ContentHints</code>	Opcional
7. La identificació del contingut: <code>ContentIdentifier</code>	Opcional
8. La referència als continguts: <code>ContentReference</code>	Opcional
9. La indicació del tipus de compromís: <code>CommitmentTypeIndication</code>	Opcional
10. La localització del signatari: <code>SignerLocation</code>	Opcional
11. Els atributs del signatari: <code>SignerAttributes</code>	Opcional
12. El segell de data i hora sobre el contingut: <code>ContentTimestamp</code>	Opcional
13. Contrasignatura: <code>Countersignature</code>	Opcional
14. Identificació de la política de signatura: <code>SignaturePolicyIdentifier</code>	Opcional ⁴²
15. Segell de data i hora de la signatura: <code>SignatureTimeStampToken</code>	Obligatori ⁴³
16. Referències completes de certificats: <code>CompleteCertificateReferences</code>	Obligatori
17. Referències completes de revocació: <code>CompleteRevocationReferences</code>	Obligatori
18. Segell de data i hora de la signatura electrònica completa:	Obligatori ⁴⁴

⁴² Depèn del tipus de signatura segellada.

⁴³ Només és obligatori quan s'utilitza segellament criptogràfic de data i hora, enlloc de marcatge de data i hora.

ESCTimeStampToken	
19. Segell de data i hora de sobre les referències de certificats i revocacions: <code>TimestampedCertsCRLs</code>	Obligatori ⁴⁵
20. Valors de certificats: <code>CertificateValues</code>	Obligatori
21. Valors de revocació: <code>RevocationValues</code>	Obligatori
22. Segell de data i hora d'arxiu: <code>ArchiveTimeStamp</code>	Obligatori

⁴⁴ Només és obligatori quan la signatura electrònica d'arxiu es construeix sobre la signatura electrònica CAdES-X Long Type 1.

⁴⁵ Només és obligatori quan la signatura electrònica d'arxiu es construeix sobre la signatura electrònica CAdES-X Long Type 2.

Annex. La sintaxi de la signatura electrònica en CMS

A continuació s'exposa la sintaxi CMS completa, integrant els diferents camps que permeten les diferents especificacions, incloent-hi S/MIME i les extensions de la normativa europea de signatura electrònica avançada.

Definició general del tipus de contingut de protecció

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

ContentType ::= OBJECT IDENTIFIER
```

Estructura de la signatura electrònica (dades signades)

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

SignedData: Algorismes de resum criptogràfic

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

```
DigestAlgorithmIdentifier ::= AlgorithmIdentifier
```

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

SignedData: Contingut signat

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

SignedData: Certificats digitals a emprar

Conjunt de certificats

```
CertificateSet ::= SET OF CertificateChoices
```

```
CertificateChoices ::= CHOICE {  
    certificate Certificate,  
    extendedCertificate [0] IMPLICIT ExtendedCertificate, -- Obsolete  
    v1AttrCert [1] IMPLICIT AttributeCertificateV1, -- Obsolete  
    v2AttrCert [2] IMPLICIT AttributeCertificateV2,  
    other [3] IMPLICIT OtherCertificateFormat }
```

Definició de certificat⁴⁶

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

```
TBSCertificate ::= SEQUENCE {  
    version [0] Version DEFAULT v1,  
    serialNumber CertificateSerialNumber,  
    signature AlgorithmIdentifier,  
    issuer Name,  
    validity Validity,  
    subject Name,  
    subjectPublicKeyInfo SubjectPublicKeyInfo,  
    issuerUniqueID [1] IMPLICIT UniqueIdentifier OPTIONAL,
```

⁴⁶ Extreta de l'especificació tècnica RFC 3280.

```

-- If present, version MUST be v2 or v3
subjectUniqueID [2] IMPLICIT UniqueIdentifier OPTIONAL,
-- If present, version MUST be v2 or v3
extensions      [3] 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 }
```

Definició de certificat extens (obsolet)

```
ExtendedCertificate ::= SEQUENCE {
  extendedCertificateInfo ExtendedCertificateInfo,
  signatureAlgorithm      SignatureAlgorithmIdentifier,
```

signature Signature }

```
ExtendedCertificateInfo ::= SEQUENCE {  
    version CMSVersion,  
    certificate Certificate,  
    attributes UnauthAttributes }
```

```
Signature ::= BIT STRING
```

Definició de certificat d'atributs, versió 1 (obsolet)

```
AttributeCertificateV1 ::= SEQUENCE {  
    acInfo AttributeCertificateInfoV1,  
    signatureAlgorithm AlgorithmIdentifier,  
    signature BIT STRING }
```

```
AttributeCertificateInfoV1 ::= SEQUENCE {  
    version AttCertVersionV1 DEFAULT v1,  
    subject CHOICE {  
        baseCertificateID [0] IssuerSerial,  
        -- associated with a Public Key Certificate  
        subjectName [1] GeneralNames },  
        -- associated with a name  
    issuer GeneralNames,  
    signature AlgorithmIdentifier,  
    serialNumber CertificateSerialNumber,  
    attCertValidityPeriod AttCertValidityPeriod,  
    attributes SEQUENCE OF Attribute,  
    issuerUniqueID UniqueIdentifier OPTIONAL,  
    extensions Extensions OPTIONAL }
```

```
AttCertVersionV1 ::= INTEGER { v1(0) }
```

Definició de certificat d'atributs, versió 2⁴⁷

⁴⁷ Només s'aporta la definició general, extreta de l'especificació tècnica RFC 3281.

AttributeCertificateV2 ::= AttributeCertificate

```
AttributeCertificate ::= SEQUENCE {
    acinfo          AttributeCertificateInfo,
    signatureAlgorithm AlgorithmIdentifier,
    signatureValue  BIT STRING }
```

```
AttributeCertificateInfo ::= SEQUENCE {
    version          AttCertVersion -- version is v2,
    holder           Holder,
    issuer           AttCertIssuer,
    signature        AlgorithmIdentifier,
    serialNumber     CertificateSerialNumber,
    attrCertValidityPeriod AttCertValidityPeriod,
    attributes       SEQUENCE OF Attribute,
    issuerUniqueID  UniqueIdentifier OPTIONAL,
    extensions       Extensions     OPTIONAL }
```

Definició d'altres formats de certificats

```
OtherCertificateFormat ::= SEQUENCE {
    otherCertFormat OBJECT IDENTIFIER,
    otherCert ANY DEFINED BY otherCertFormat }
```

SignedData: Informacions de revocació

Conjunt d'informacions de revocació

RevocationInfoChoices ::= SET OF RevocationInfoChoice

```
RevocationInfoChoice ::= CHOICE {
    crl CertificateList,
    other [1] IMPLICIT OtherRevocationInfoFormat }
```

Definició de llista de certificats⁴⁸

```
CertificateList ::= SEQUENCE {
    tbsCertList          TBSCertList,
    signatureAlgorithm  AlgorithmIdentifier,
    signature            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] Extensions OPTIONAL
                        -- if present, MUST be v2
```

Definició d'altres formats d'informació de revocació

```
OtherRevocationInfoFormat ::= SEQUENCE {
    otherRevInfoFormat OBJECT IDENTIFIER,
    otherRevInfo ANY DEFINED BY otherRevInfoFormat }
```

SignedData: Informació dels signataris

Conjunt de signataris

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

⁴⁸ Extreta de l'especificació tècnica RFC 3280.

Definició d'informació de signatari

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

Identificació del signatari

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

```
IssuerAndSerialNumber ::= SEQUENCE {  
    issuer Name,  
    serialNumber CertificateSerialNumber }
```

```
SubjectKeyIdentifier ::= OCTET STRING
```

Algorisme de resum criptogràfic emprat

```
DigestAlgorithmIdentifier ::= AlgorithmIdentifier
```

Atributs de la signatura CMS (signats o no)

```
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
```

```
ContentType ::= OCTET STRING
```

```
MessageDigest ::= OCTET STRING
```

```
SigningTime ::= Time
```

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

```
Countersignature ::= SignerInfo
```

Atributs addicionals de la signatura ESS

```
SigningCertificate ::= SEQUENCE {  
    certs          SEQUENCE OF ESSCertID,  
    policies       SEQUENCE OF PolicyInformation OPTIONAL }
```

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

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

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

```
ContentIdentifier ::= OCTET STRING
```

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

contentType ContentType }

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

Atributs addicionals d'acord amb la Directiva europea

OtherSigningCertificate ::= SEQUENCE {
 certs SEQUENCE OF OtherCertID,
 policies SEQUENCE OF PolicyInformation OPTIONAL }

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

OtherHash ::= CHOICE {
 sha1Hash OtherHashValue,
 otherHash OtherHashAlgAndValue }

OtherHashValue ::= OCTET STRING

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

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

SignaturePolicyId ::= SEQUENCE {
 sigPolicyId 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

SPuri ::= IA5String

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)) }

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

CommitmentTypeIdentifier ::= OBJECT IDENTIFIER

CommitmentTypeQualifier ::= SEQUENCE {

```
commitmentTypeIdentifier  CommitmentTypeIdentifier,  
qualifier                 ANY DEFINED BY commitmentTypeIdentifier }
```

```
SignerLocation ::= SEQUENCE { -- com a mínim, n'hi haurà un  
  countryName      [0]    DirectoryString OPTIONAL,  
                    -- As used to name a Country in X.500  
  localityName     [1]    DirectoryString OPTIONAL,  
                    -- As used to name a locality in X.500  
  postalAddress    [2]    PostalAddress OPTIONAL }
```

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

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

```
ClaimedAttributes ::= SEQUENCE OF Attribute
```

```
CertifiedAttributes ::= AttributeCertificate
```

```
ContentTimestamp ::= TimeStampToken
```

```
SignatureTimeStampToken ::= TimeStampToken
```

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

```
CompleteRevocationRefs ::= SEQUENCE OF CrlOcspRef
```

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

```
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 }

AttributeCertificateRefs ::= SEQUENCE OF OtherCertID

AttributeRevocationRefs ::= SEQUENCE OF CrlOcspRef

ESCTimeStampToken ::= TimeStampToken

TimestampedCertsCRLs ::= TimeStampToken

CertificateValues ::= SEQUENCE OF Certificate

RevocationValues ::= SEQUENCE {
    crlVals                [0] SEQUENCE OF CertificateList OPTIONAL,
    ocspVals               [1] SEQUENCE OF BasicOCSPResponse OPTIONAL,
    otherRevVals           [2] OtherRevVals OPTIONAL}

OtherRevVals ::= SEQUENCE {
```

```
OtherRevValType  OtherRevValType,  
OtherRevVals     ANY DEFINED BY OtherRevValType }
```

```
OtherRevValType ::= OBJECT IDENTIFIER
```

```
BasicOCSPResponse ::= SEQUENCE {  
  tbsResponseData  ResponseData,  
  signatureAlgorithm AlgorithmIdentifier,  
  signature         BIT STRING,  
  certs             [0] EXPLICIT SEQUENCE OF Certificate OPTIONAL }
```

```
ResponseData ::= SEQUENCE {  
  version           [0] EXPLICIT Version DEFAULT v1,  
  responderID       ResponderID,  
  producedAt        GeneralizedTime,  
  responses         SEQUENCE OF SingleResponse,  
  responseExtensions [1] EXPLICIT Extensions OPTIONAL }
```

```
ResponderID ::= CHOICE {  
  byName           [1] Name,  
  byKey            [2] KeyHash }
```

```
KeyHash ::= OCTET STRING -- SHA-1 hash of responder's public key  
(excluding the tag and length fields)
```

```
SingleResponse ::= SEQUENCE {  
  certID           CertID,  
  certStatus       CertStatus,  
  thisUpdate       GeneralizedTime,  
  nextUpdate       [0] EXPLICIT GeneralizedTime OPTIONAL,  
  singleExtensions [1] EXPLICIT Extensions OPTIONAL }
```

```
CertStatus ::= CHOICE {  
  good             [0] IMPLICIT NULL,  
  revoked          [1] IMPLICIT RevokedInfo,  
  unknown          [2] IMPLICIT UnknownInfo }
```

```
RevokedInfo ::= SEQUENCE {  
    revocationTime          GeneralizedTime,  
    revocationReason [0]   EXPLICIT CRLReason OPTIONAL }
```

```
UnknownInfo ::= NULL
```

```
ArchiveTimeStampToken ::= TimeStampToken
```

Algorisme i valor de la signatura digital del signatari

```
SignatureAlgorithmIdentifier ::= AlgorithmIdentifier
```

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