وزارة المواصلات والاتصالات
MINISTRY OF TRANSPORT AND COMMUNICATIONS

Electronic Signature Formats Standards

Version : 1.0

Author : Qatar Public Key Infrastructure Section

Document Classification : PUBLIC

Published Date: August 2018

Q-CERT

Document Information

Date	Version	Observations
05/08/2018	1.0	Qatar National PKI Team

Content

1. Overview

The development of electronic transactions is subject to the existence of guarantees on the security of data transmissions. In this respect, the electronic signature is a security element to protect the data and the electronic transactions, thus it guarantees the identity of the signer and the integrity of the data. Several electronic signature formats have been developed and standardized to support the different types of electronic data to be signed.

2. Introduction

2.1. Objective of the document

The purpose of this document is to provide an overview of the different formats and standardized structures of the electronic signature.

This document provides a source of reference information for the successful selection of the format and structure of the electronic signature for the implementation of applications based on electronic signatures.

2.2. Audience

This document is intended for developers wishing to implement electronic signature creation and verification applications, all organizations and users that intend to use Electronic Signatures. It can also serve anybody wanting to deepen on the aspects of electronic signature.

3. Basic electronic Signature Formats

An electronic signature is a set of structured data in electronic form attached or logically linked to an electronic document and used to authenticate the signatory.

The electronic signature optionally contains information such as the document to be signed, the set of certificates constituting the chain of trust, the revocation lists corresponding to the certificates, so-called signed attributes, a signature value and so-called unsigned attributes (see Figure 1).

- The signed attributes are:
  - The hash of the electronic document to sign.
  - The content type of the document to sign.
  - Other attributes related to the signer such as email address, certificate identifier, etc.
- The signature value is the result of the encryption applied to all the attributes related to the signature (the hash of the electronic document to be signed and / or the hash of the set of signed attributes).
- Unsigned attributes are additional information that are not used to calculate the signature value.

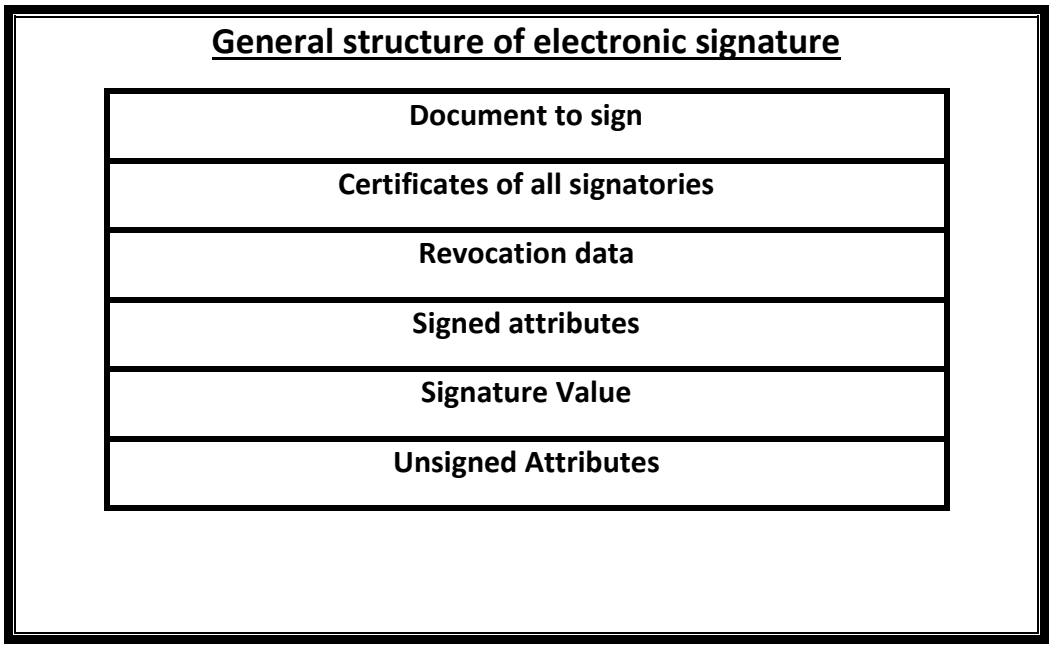

Figure1: **General Structure of Electronic Signature**

An electronic signature can exist in several formats. These formats will be presented in the following subsections.

3.1. Format CMS/PKCS#7

CMS [**RFC 5652**], an **IETF** standard, defines a syntax for signing or encrypting any electronic content. The syntax of CMS is based on **ASN.1** encoding which is a formal notation for describing structured data in an abstract way.

The **CMS** is derived from **PKCS#7** version 1.5 [**RFC 2315**].

PKCS#7 version 1.5 was developed outside the **IETF**; it was originally published as a technical note of "RSA Laboratories" in November 1993. Since that time, the **IETF** has taken responsibility for the development and implementation of CMS format.

The **PKCS#7** specification is used to attach an electronic signature to a data block. It defines a syntax for the protection of messages with an electronic signature encoded in **ASN.1**. **PKCS#7** does not support the inclusion of specific information such as: signing time, signatory information, type of document to sign.

CMS Signature Syntax

The presentation of the CMS signature in ASN.1 is as follows:

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

Figure 2: **SignedData structure**

The fields of type ***SignedData*** have the following meanings:

- **version:** is the syntax version number. If no attribute certificates are present in the certificates field, the encapsulated content type is *id-data*, and all of the elements of SignerInfos are version 1, then the value of version shall be 1. Alternatively, if attribute certificates are present, the encapsulated content type is other than *id-data*, or any of the elements of SignerInfos are version 3, then the value of *version* shall be 3.
- **digestAlgorithms:** is a collection of message digest algorithm identifiers. There may be any number of elements in the collection, including zero. Each element identifies the message digest algorithm, along with any associated parameters, used by one or more signer. The collection is intended to list the message digest algorithms employed by all of the signers, in any order, to facilitate one-pass signature verification.
- **encapContentInfo:** is the signed content, consisting of a content type identifier and the content itself.
- **certificates:** is a collection of certificates. It is intended that the set of certificates be sufficient to contain chains from a recognized "root" or "top-level certification authority" to all of the signers in the *signerInfos* field. There may be more certificates than necessary, and there may be certificates sufficient to contain chains from two or more independent top-level certification authorities. There may also be fewer certificates than necessary, if it is expected that recipients have an alternate means of obtaining necessary certificates (e.g., from a previous set of certificates).
- **crls:** is a collection of certificate revocation lists (CRLs). It is intended that the set contain information sufficient to determine whether or not the certificates in the certificates field are valid, but such correspondence is not necessary. There may be more CRLs than necessary, and there may also be fewer CRLs than necessary.
- **signerInfos:** is a collection of per-signer information. There may be any number of elements in the collection, including zero.

The *SignerInfo* field is used to encapsulate the data identifying a signatory. The presentation of the *SignerInfo* signature in ASN.1 is as follows:

SignerInfo Syntax

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

Figure 3: ***SignerInfo* Syntax**

The fields of type ***SignerInfo*** have the following meanings:

- **version:** is the syntax version number. If the *SignerIdentifier* is the CHOICE *issuerAndSerialNumber*, then the version shall be 1. If the *SignerIdentifier* is *subjectKeyIdentifier*, then the version shall be 3.
- **sid:** specifies the signer's certificate (and thereby the signer's public key). The signer's public key is needed by the recipient to verify the signature. *SignerIdentifier* provides two alternatives for specifying the signer's public key. The *issuerAndSerialNumber* alternative identifies the signer's certificate by the issuer's distinguished name and the certificate serial number; the *subjectKeyIdentifier* identifies the signer's certificate by the X.509 *subjectKeyIdentifier* extension value.
- **digestAlgorithm** identifies the message digest algorithm, and any associated parameters, used by the signer. The message digest is computed on either the content being signed or the content together with the signed attributes. The message digest algorithm should be among those listed in the *digestAlgorithms* field of the associated *SignerData*.
- **signedAttributes** is a collection of attributes that are signed. The field is optional, but it must be present if the content type of the *EncapsulatedContentInfo* value being signed is not id-data. Each *SignedAttribute* in the SET must be DER encoded.
- **signatureAlgorithm** identifies the signature algorithm, and any associated parameters, used by the signer to generate the digital signature.
- **signature** is the result of digital signature generation, using the message digest and the signer's private key.
- **unsignedAttributes** is a collection of attributes that are not signed. The field is optional.

3.2. S/MIME Format

S/MIME [5751] is a method of securing e-mail exchanges to ensure the confidentiality and non-repudiation of e-mail messages. It allows to electronically sign MIME documents, mainly included in an email.

S/MIME was originally developed by **RSA** Data Security. Ratified in July 1999 by the **IETF**, **S/MIME** has become a standard.

S/MIME adds other MIME content types (MIME entities) that are used for the electronic signature. There are two ways to encode a signed email:

3.2.1. S/MIME Clear-signed Message

The signed messages, of the MIME type, are consisted of two parts:

- The clear original message in the MIME entity "**text/plain**".
- The signature in **PKCS#7** (**S/MIMEv2**) or CMS (**S/MIMEv3**) format encoded in base64 in the MIME entity "**application/pkcs7-signature**".

```
From: "User1" <user1@domain.qa>
To: "User2" <user2@domain.qa>
Subject: S/MIME message
Date: Tue, 24 Jul 2018 01:38:36 +0100
MIME-Version: 3.2
Content-Type: multipart/signed;
     protocol="application/pkcs7-signature";
     micalg=SHA2; boundary="ABCDE"
--ABCDE
Content-Type: text/plain
This is my plain text letter.
--ABCDE
Content-Type: application/pkcs7-signature
Content-Transfer-Encoding: base64
MIAGCSqGSIb3DQEHAqCAMIACAQExCzAJBgUrDgMCGgUAMI
ggJfoAMCAQICDnW7AAAAAjGqNhNz5/78MA0GCSqGSIb3DQ
MA4GA1UECBMHSGFtYnVyZzEQMA4GA1UEBxMHSGFtYnVyZz
[...]
--ABCDE—
```

Figure 4: **S/MIME Clear-signed Message**

The "**S/MIME Clear-signed Message**" signature allows non-**S/MIME** compatible software to read the contents of the original message.

3.2.2. S/MIME Opaque-signed Message

The original message is included in **PKCS#7** or **CMS** format, encoded in base64, in the **MIME** entity "**application/pkcs7-mime**".

```
From: "User1" <user1@domain.qa>
To: "User2" <user2@domain.qa>
Subject: S/MIME message
10
Date: Tue, 24 Jul 2018 01:53:36 +0100
MIME-Version: 3.2
Content-Type: application/pkcs7-mime;
smime-type=signed-data
Content-Transfer-Encoding: base64
MIAGCSqGSIb3DQEHAqCAMIACAQExCzAJBgUrDgMCGgUAMI
bTogIkhlaW5yaWNoIE1vc2VyIiA8aC5tb3Nlci5qdW5AbW
c2VyLmp1bkBtb3NlcndhcmUuYXQ+DQpTdWJqZWN0OiBNSU
[...]
```

Figure5: **S/MIME Opaque-signed Message**

The "S/MIME Opaque-signed Message" signature does not allow non-S/MIME-compatible software to read the content of the original message.

3.3. XMLDsig

XMLDsig [**RFC 3275**] is defined by **W3C** and **IETF**. It uses **XML** as syntax to carry all the data forming the signature. As a result, it benefits from all the flexibility that **XML** offers in terms of structuring data. Producing an **XML** signature amounts to producing an **XML** document according to syntax rules defined by a **W3C** recommendation presented in [**RFC 3275**].

XMLDsig Syntax

```
<Signature ID?>
   <SignedInfo>
     <CanonicalizationMethod/>
     <SignatureMethod/>
     (<Reference URI? >
       (<Transforms>)?
       <DigestMethod>
       <DigestValue>
     </Reference>)+
   </SignedInfo>
   <SignatureValue>
  (<KeyInfo>)?
  (<Object ID?>)*
 </Signature>
```

Figure 6**: XMLDsig Syntax (Regular expression)**

The fields of XMLDsig signature have the following meanings:

- **SignedInfo**: (required element) is the information that is actually signed. Note that the algorithms used in calculating the *SignatureValue* are also included in the signed information while the *SignatureValue* element is outside *SignedInfo*.
- **CanonicalizationMethod:** is the algorithm that is used to canonicalize the *SignedInfo* element before it is digested as part of the signature operation.
- **SignatureMethod:** is the algorithm that is used to convert the canonicalized *SignedInfo* into the *SignatureValue*. It is a combination of a digest algorithm and a key dependent algorithm and possibly other algorithms such as padding, for example RSA-SHA2. The algorithm names are signed to resist attacks based on substituting a weaker algorithm.
- **Reference:** Each Reference element includes the digest method and resulting digest value calculated over the identified data object. It may also include transformations that produced the input to the digest operation. A data object is signed by computing its digest value and a signature over that value. The signature is later checked via reference and signature validation.
- **SignatureValue:** The *SignatureValue* element contains the actual value of the digital signature; it is always encoded using base64 [MIME].
- **KeyInfo**: indicates the key to be used to validate the signature. Possible forms for identification include certificates, key names, and key agreement algorithms and information. *KeyInfo* is optional for two reasons. First, the signer may not wish to reveal key information to all document processing parties. Second, the information may be known within the application's context and need not be represented explicitly. Since *KeyInfo* is outside of *SignedInfo*, if the signer wishes to bind the keying information to the signature, a Reference can easily identify and include the *KeyInfo* as part of the signature.

3.4. PDF Signature

The International Organization for Standardization published the PDF signature in 2008 [**ISO 32000-1**] and identifies ways in which an electronic signature can be incorporated into a **PDF** document to authenticate the author of the document and validate the integrity of the content of the document. This standard is based on the **CMS** specification.

4. Advanced electronic signature formats

4.1. CAdES Format

The **CAdES** format [**ETSI 319 122-1** | **ETSI 319 122-2**] is developed by **ETSI**; it is an extension of the **CMS** format. It complies with Directive 1999/93 / EC of the European Parliament and of the Council of 13 December 1999 on the Community Framework for Electronic Signatures. **CADES** defines advanced electronic signatures that remain valid for long periods.

CAdES specifies profiles that describe the structure of the advanced electronic signature. These profiles are:

- **CAdES-BES** (Basic Electronic Signature)
- **CAdES-EPES** (Electronic Signature with explicit signature policy)
- **CAdES-T** (Time Stamped Electronic Signature)
- **CAdES-C** (Electronic Signature with complete validation data)
- **CAdES-X** (Electronic Signature with Extended Validation Data)
- **CAdES-X-L** (Electronic Signature with Extended Validation Data Incorporated for the Long Term)
- **CAdES-A** (Electronic signature with validation data for archiving)

These profiles will be presented in Section **7** entitled "Advanced Electronic Signature Profiles".

4.2. XAdES Format

The **XAdES** format [ETSI TS 101 903] is developed by **ETSI**, it is an extension of the IETF / W3C **XMLDSig** Signature format, by defining **XML** formats for advanced electronic signatures that remain valid for long periods of time and are in accordance with Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on the Community framework for signatures.

This standard adds the following profiles to XMLDS**ig**:

- **XAdES-BES** (Basic Electronic Signature)
- **XAdES-EPES** (Electronic signature with explicit signature policy)
- **XAdES-T** (timestamped electronic signature)
- **XAdES-C** (Electronic signature with complete validation data)
- **XAdES-X** (Electronic signature with extended validation data)
- **XAdES-X-L** (Electronic signature with extended validation data incorporated for the long term)
- **XAdES-A** (Electronic Signature with Validation Data for Archiving)

These profiles will be presented in Section **7** entitled "Advanced Electronic Signature Profiles".

4.3. PAdES Format

In April 2016, ETSI published a new standard (updated one) for advanced signatures for PDF documents that meets the requirements of the European Directive on a Community Framework for Electronic Signatures (Directive 1999/93 / EC). The new standard is the result of adaptations and extensions to ISO 32000-1 that are carried out by ETSI in collaboration with PDF experts. The standard is named PAdES [ETSI EN 319 142-1 | ETSI EN 319 142-2].

PAdES defines the following profiles:

- PAdES-BES (Basic Electronic Signature) which is similar to CAdES-BES.
- PAdES-EPES (Electronic Signature with explicit signature policy) which is similar to CAdES-EPES.
- PAdES-T (Time Stamped Electronic Signature) which is similar to CAdES-T.

5. Electronic Signature Types

5.1. Electronic Signature Types for XML format (XML and XAdES)

5.1.1. Detached Signature

An XML signature is called "detached signature" if it is neither enveloping nor enveloped (see Figure 7). The signature is over content external to the Signature element, and can be identified via a URI or transform. Consequently, the signature is "detached" from the content it signs. This definition typically applies to separate data objects, but it also includes the instance where the Signature and data object reside within the same XML document but are sibling elements.

There are two types of detached signature:

- The signed resource and the document containing the XML signature are fully detached (remote resource).
- The signed resource and the signature are on the same XML document but in two different levels of encapsulation.

5.1.2. Enveloping Signature

An XML signature is called "enveloping signature" if the root of the signed content is a descendant of the root of the signature (see Figure 7). The signature is over content found within an Object element of the signature itself. The Object (or its content) is identified via a Reference (via a URI fragment identifier or transform). XMLDSIG defines a specific Object container for this type of signature.

5.1.3. Enveloped Signature

An XML signature is called "enveloped signature" if the root of the signature is a descendant of the root of the signed XML content (see Figure 7). The signature is over the XML content that contains the signature as an element. The content provides the root XML document element. Obviously, enveloped signatures must take care not to include their own value in the calculation of the ***SignatureValue***. This type of signature is useful when the signature is transported by a protocol itself based on XML.

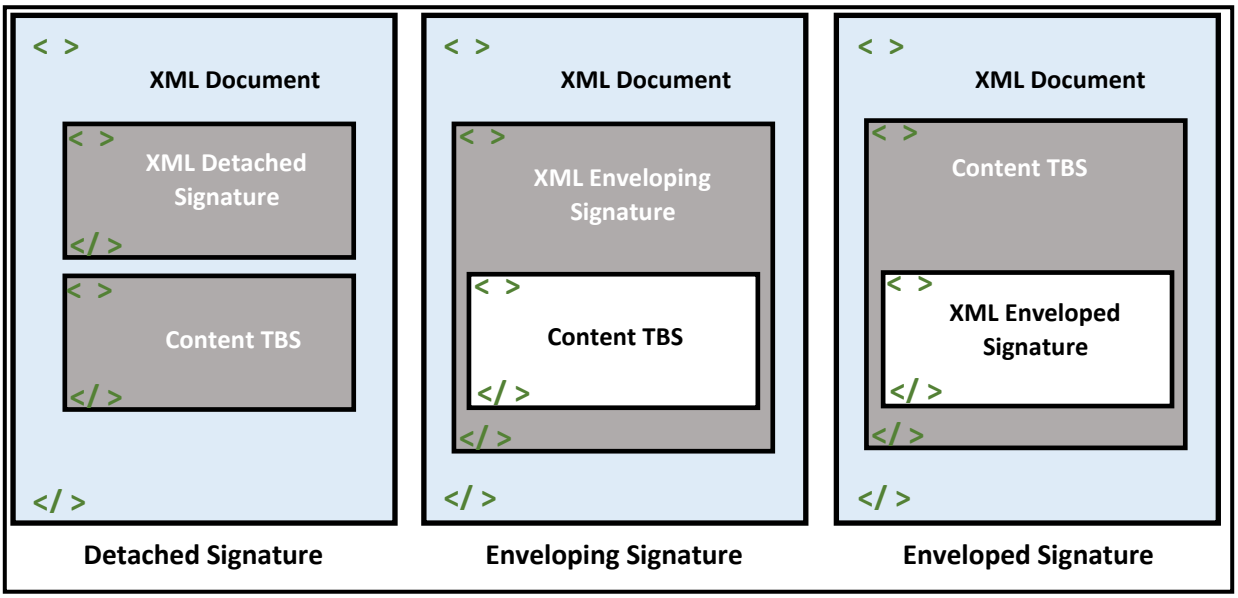

Figure7: **XML Signature Types**

5.2. Electronic Signature Types for CMS format (CMS and CAdES)

5.2.1. Attached Signature

An electronic signature is called "attached signature" if it includes the signed content.

5.2.2. Detached Signature

An electronic signature is called "detached signature" if the signed content is not included into the signature file.

6. Multiple signatures

6.1. Co-signature

A co-signature is used when all parties agree on the signed content. The validity of already existing signatures is not verified or confirmed when applying a new one. In the structure of the electronic signature, the co-signature appears as an additional structure (structure of information relating to the second signer). The structures of the all signer information are placed at the same level as that of the first signer.

6.2. Counter signature

By using a counter-signature, the signatory confirms the content to be signed as well as the previous signature. The counter-signature appears in the *counter-signature* field in the unsigned attributes section. It has the same structure as the previous signature.

7. Advanced Electronic Signature Profiles

7.1. Overview

The advanced electronic signature profiles differ from each other in the number and types of attributes contained in the signed attributes section and the unsigned attributes section in the electronic signature structure.

The **CAdES** and **XAdES** standards have similar concepts, but they differ in representation and coding: the **CAdES** standard uses the **ASN.1** data representation, while the **XAdES** standard uses the **XML** data structures to represent the necessary information. The formats and profiles of these two standards are also similar.

This section provides an overview of the different advanced electronic signature profiles.

7.2. Basic Electronic Signature Profile (CAdES-BES/XAdES-BES)

This profile of signature adds to the **CMS** / **XMLDISIG** signature signed attributes such as ***signing-time***, ***signer-location***, ***commitment-type***, and unsigned attributes such as the ***countersignature*** attribute. Some signed and unsigned attributes are optional while others are mandatory (See Tables in **APPENDIX III** and **APPENDIX IV**).

7.3. Electronic signature with Explicit Signature Policy Profile (CAdES-EPES/XAdES-EPES)

This profile of signature adds to the (**CAdES-BES** / **XAdES-BES**) format the signed attribute ***signature-policy-identifier*** (See Figure 8). **CAdES-EPES** / **XAdES-EPES** signatures are built on **CAdES-BES** / **XAdES-BES** signatures by adding one ***signature-policy-identifier*** attribute.

The signature policy is a set of rules related to the creation and validation of the electronic signature on which the signatory and the verifier agree. The policy can be explicitly identified in the signature structure by using the ***signature-policy-identifier*** attribute.

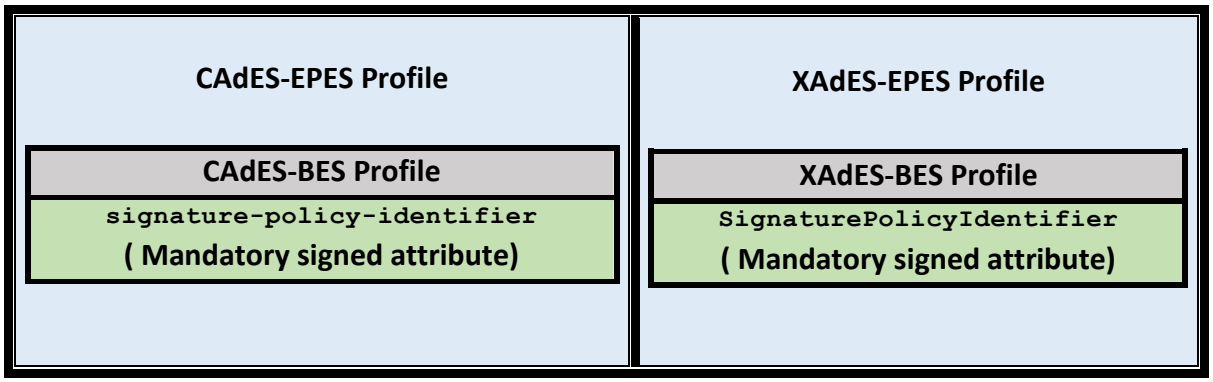

Figure 8: **CAdES-EPES / XAdES-EPES Profile**

7.4. Electronic Signature with Time Stamp Profile (CAdES-T / XAdES-T)

An electronic signature with Time Stamp adds the time stamp in the format **CAdES-BES** / **XAdES-BES** or **CAdES-EPES** / **XAdES-EPES**, which is the initial step of a long-term validity of the signature. This shall be an unsigned attribute (see Figure 9). The time stamp is applied to the value of the electronic signature (The signature-time-stamp attribute shall encapsulate one time-stamp token computed on the electronic signature value for a specific signer).

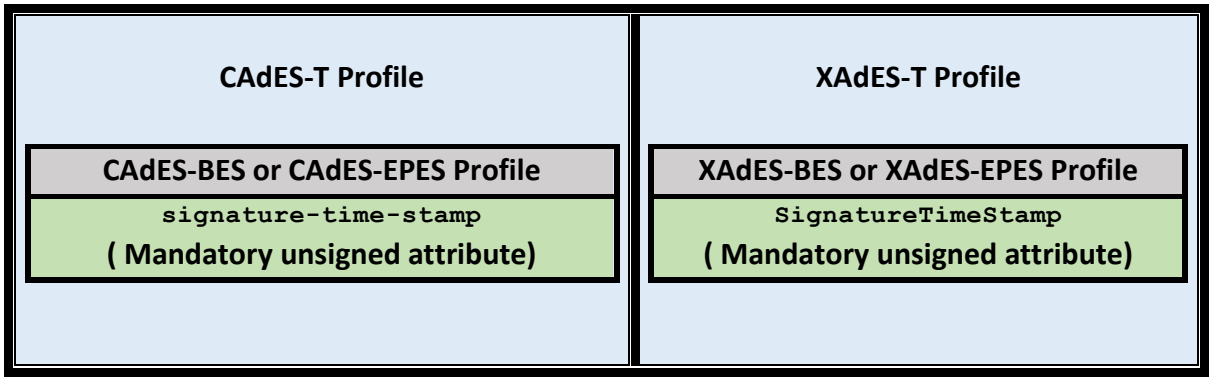

Figure 9: **CAdES-T / XAdES-T Profile**

7.5. Electronic Signature with Complete validation data Profile (CAdES-C / XAdES-C)

An electronic signature with complete validation data (**CAdES-C** / **XAdES-C**) adds to the (**CAdES-T** / **XAdES-T**) format in the unsigned attributes section the following elements (see Figure 10):

- `complete-certificate-references` (**CAdES-C**).
- `CompleteCertificateRefsV2` (**XAdES-C**).
- `complete-revocation-references` (**CAdES-C**).
- `CompleteRevocationsRefsV2 (XAdES-C)`

The revocation status can be got from a **CRL** or an **OCSP** responder. Following are the criteria to take into consideration when implementing both solutions:

• An **OCSP** response is small (size), however a reliable way to query the server has to be implemented. The verifier has to deal smoothly with all kinds of exceptions.

• When using the **CRL validation way, it is not required that the CRL(s) is to be downloaded from an online server**. The verifier has to insure getting the freshest CRL in case the SVA is running in an offline environment. The CRL can be bulky at times which may affect the speed of the validator.

An OCSP response shall be incorporated into the signature either by using the encoding of the *`OCSPResponse`*type or the *`BasicOCSPResponse`* type (As defined in **ETSI EN 319 122-1 V1.1.1 clause 4.8.2**). Certificate Revocation Lists (**CRLs**) shall be as defined in **IETF RFC 5280**.

This format is useful in situations where the information is archived by an external source, such as a trusted third party.

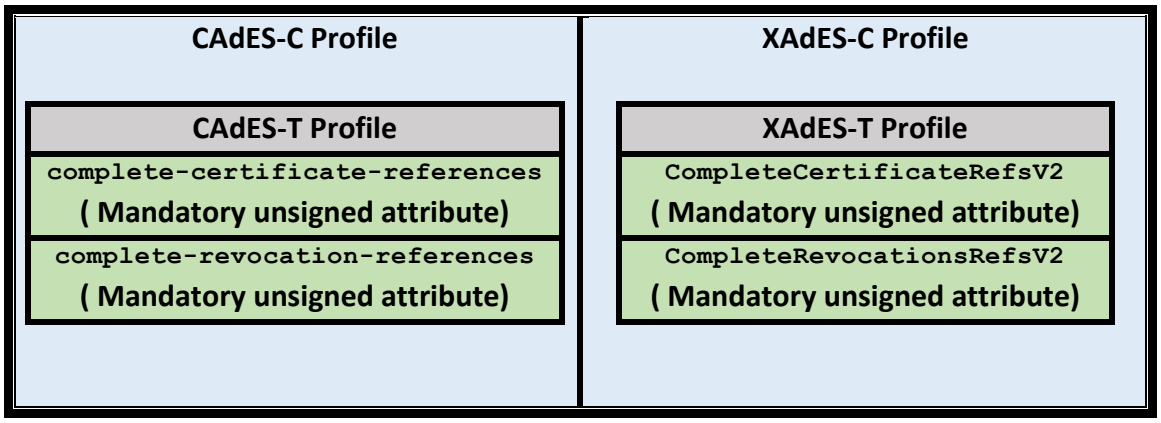

Figure 10: **CAdES-C / XAdES-C Profile**

7.6. Electronic Signature with eXtended validation data Profile (CAdES-X / XAdES-X)

Some parts of the **CAdES-C** / **XAdES-C** profile may be time-stamped to provide more additional protection to the associated electronic signature.

This profile can have two types:

7.6.1. Electronic Signature with eXtended validation data Profile Type 1 (CAdES-X Type 1 / XAdES-X Type 1)

A time stamp on the entire profile (**CAdES-C** / **XAdES-C**) is applied:

- **CAdES-X type 1** signatures are built on **CAdES-C** signatures by adding one or more **CAdES-C-Timestamp** attributes each. The ***CAdES-C-time-stamp*** attribute shall be an unsigned attribute. The **CAdES-C-time-stamp** attribute shall encapsulate one time-stamp token covering the signature, the signature timestamp, the ***complete-certificate-references*** attribute and ***complete-revocation-references*** attribute (see figure 11).
- **XAdES-X type 1** signatures are built on **XAdES-C** signatures by adding one or more ***SigAndRefsTimeStampV2*** qualifying properties containing one or more electronic time-stamps. The ***SigAndRefsTimeStampV2*** qualifying property shall be an unsigned qualifying property qualifying the signature. The ***SigAndRefsTimeStampV2*** qualifying property shall encapsulate electronic time-stamps on the digital signature value, the signature time-stamp, if present, and the **XAdES** qualifying properties containing references to validation data (see figure 11).

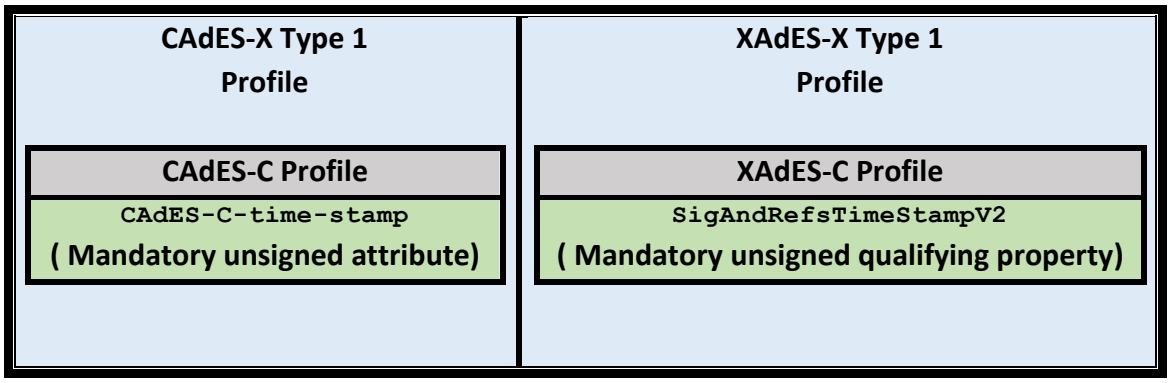

Figure 11: **CAdES-X Type 1 / XAdES-X Type 1 Profile**

7.6.2. Electronic Signature with eXtended validation data Profile Type 2 (CAdES-X Type 2 / XAdES-X Type 2)

- **CAdES-E-X type 2** signatures are built on **CAdES-C** signatures by adding one or more time-***stamped-certs-crls-references*** attributes each. The ***time-stamped-certs-crls-references*** attribute shall be an unsigned attribute. The ***time-stamped-certs-crls-references*** attribute shall encapsulate one time-stamp token of the **complete-certificate-references** attribute and the ***complete-revocation-references*** attribute (see Figure 12).
- **XAdES-X type 2** signatures are built on **XAdES-C** signatures by adding one or more ***RefsOnlyTimeStampV2*** qualifying properties. The ***RefsOnlyTimeStampV2*** qualifying property shall be an unsigned qualifying property qualifying the signature. The ***RefsOnlyTimeStampV2*** qualifying property shall encapsulate electronic time-stamps on the **XAdES** qualifying properties containing references to validation data (see Figure 12).

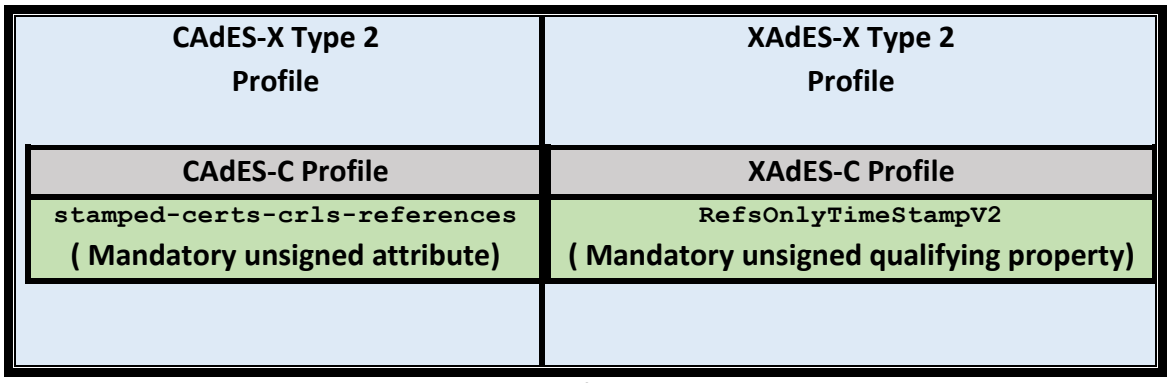

Figure 12: **CAdES-X Type 2 / XAdES-X Type 2 Profile**

7.7. Electronic Signature with eXtended Long validation data Profile (CAdES-X-L / XAdES-X-L)

This profile of electronic signature adds validation data (certificate values and revocation data) to the electronic signature structure, when this validation data will not be stored elsewhere for long term.

7.7.1. CAdES-X-L Profile

CAdES-X-L signatures are built on **CAdES-X** signature by adding attributes that contain certificates and revocation values (see Figure 13):

- The ***certificate-values*** attribute shall be an unsigned attribute. The ***certificate-values*** attribute:
  - Shall contain the values of the certificates referenced within ***complete-certificate-references***, ***attribute-certificate-references***, and the ***signingcertificate-reference*** attributes, which are not stored ***SignedData.certificates***. Certificate values within ***SignedData.certificates*** should not be included.
  - No other certificates shall be included.
- The ***revocation-values*** attribute shall be an unsigned attribute. The ***revocation-values*** attribute:
  - Shall contain the elements corresponding to the references in ***complete-revocation-references*** and ***attribute-revocation-references***, which are not stored in ***SignedData.crls***.
  - No other element shall be included.

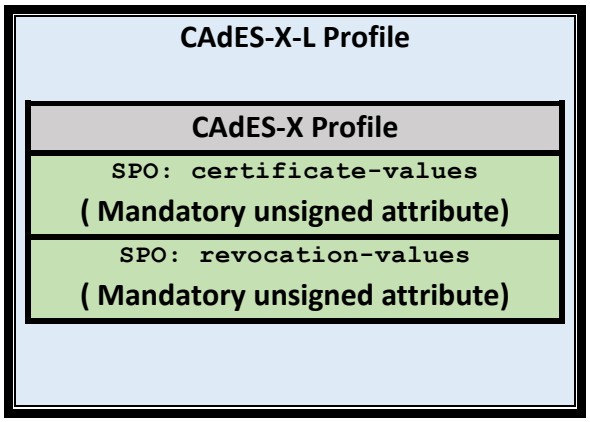

Figure 13: **CAdES-X-L Profile**

7.7.2. XAdES-X-L Profile

XAdES-X-L signatures are built on **XAdES-X** signatures by adding qualifying properties that contain certificates and revocation values (see Figure 14):

- The ***CertificateValues*** qualifying property shall be an unsigned qualifying property qualifying the signature. The ***CertificateValues*** qualifying property:
    1- Shall contain the certificate of the trust anchor, if such certificate does exist and if it is not present within the ***ds:KeyInfo***. If this certificate is present within the ***ds:KeyInfo***, it should not be included.
    2- Shall contain the **CA** certificates within the signing certificate path that are not present within the ***ds:KeyInfo***. The certificates present within ***ds:KeyInfo*** element should not be included.
    3- Shall contain the signing certificate if it is not present within the ***ds:KeyInfo***. If this certificate is present within the ***ds:KeyInfo***, it should not be included.
    4- Shall contain certificates used to sign revocation status information (e.g. **CRL**s or **OCSP** responses) of certificates in 1, 2, and 3, and certificates within their respective certificate paths that are not present in the signature. Certificate values present within the signature, including certificate values within the revocation status information themselves should not be included.
    5- Shall not contain **CA** certificates that pertain exclusively to the certificate paths of certificates used to sign attribute certificates or signed assertions within ***SignerRoleV2***, or electronic time-stamps.
    6- May contain a set of certificates used to validate any countersignature incorporated into the **XAdES** signature that are not present in other elements of the **XAdES** signature or its countersignatures. This set may include any of the certificates listed in 1, 2, 3 and 4 referred to signing certificates of countersignatures instead of the signing certificate of the **XAdES** signature. The certificates present elsewhere in the **XAdES** signature or its countersignatures should not be included.

- The ***RevocationValues*** qualifying property shall be an unsigned qualifying property that qualifies the signature. The ***RevocationValues*** qualifying property:
    1- Shall contain revocation values corresponding to **CA** certificates within the signing certificate path if they are not present within the ***ds:KeyInfo***. It shall not contain a revocation value for the trust anchor. The revocation values present within ***ds:KeyInfo*** element should not be included.
    2- Shall contain a revocation value for the signing certificate if it is not present within the ***ds:KeyInfo***. If it is present within ***ds:KeyInfo*** element, it should not be included.
    3- May contain revocation values corresponding to certificates used to sign **CRL**s or **OCSP** responses of 1 and 2, and certificates within their respective certificate paths. The revocation values present within ***ds:KeyInfo*** element should not be included.

4- Shall not contain revocation values corresponding to **CA** certificates that pertain exclusively to the certificate paths of certificates used to sign attribute certificates or signed assertions within ***SignerRoleV2***, or electronic time-stamps. And
5- May contain revocation values corresponding to the signing certificate of any countersignature incorporated into the **XAdES** signature as well as to the **CA** certificates in its certificate path. This set may include any of the revocation values listed in 1, 2, and 3 referred to signing certificates of countersignatures instead of the signing certificate of the **XAdES** signature. However, those revocation values among the aforementioned ones that are already present in other elements of the **XAdES** signature should not be included.

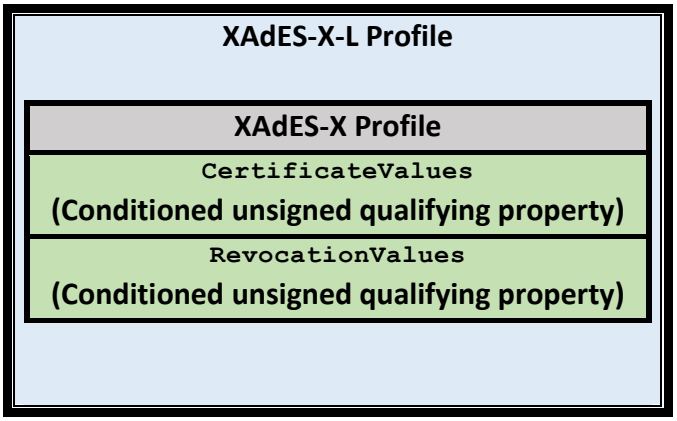

Figure 14: **XAdES-X-L Profile**

7.8. Electronic Signature with Archive validation data Profile (CAdES-A / XAdES-A)

In order to mitigate the risk of weak signature algorithms, an additional time stamp can be added to the **CAdES-X-L** / **XAdES-X-L** signature structure. The purpose of this extra element is to tackle the long-term availability and integrity of the validation material. **CAdES-A** / **XAdES-A** adds in **CAdES-X-L** / **XAdES-X-L** profile in the unsigned attributes section the following elements (see Figure 15):

- **CAdES-A** signatures are built on **CAdES-T**, **CAdES-C**, **CAdES-X** (of Type 1 and of Type 2) and **CAdES-X-L** signatures by adding ***archive-time-stamp-v3*** attribute. The ***archive-time-stamp-v3*** attribute shall be an unsigned attribute. The **archive-time-stamp-v3** attribute shall be a time-stamp token of the signed document and the signature, including signed attributes, and all other essential components of the signature as protected by the ***ats-hash-index-v3*** attribute.
- **XAdES-A** signatures are built on **XAdES-T**, **XAdES-C**, **XAdES-X** (of Type 1 and of Type 2) and **XAdES-X-L** signatures by adding ***ArchiveTimeStamp*** qualifying property. The ***ArchiveTimeStamp*** qualifying property shall be an unsigned qualifying property qualifying the signature. The ***ArchiveTimeStamp*** qualifying property shall encapsulate electronic time-stamps computed on all the data objects incorporated into the **XAdES** signature at the time of generating each electronic time-stamp.

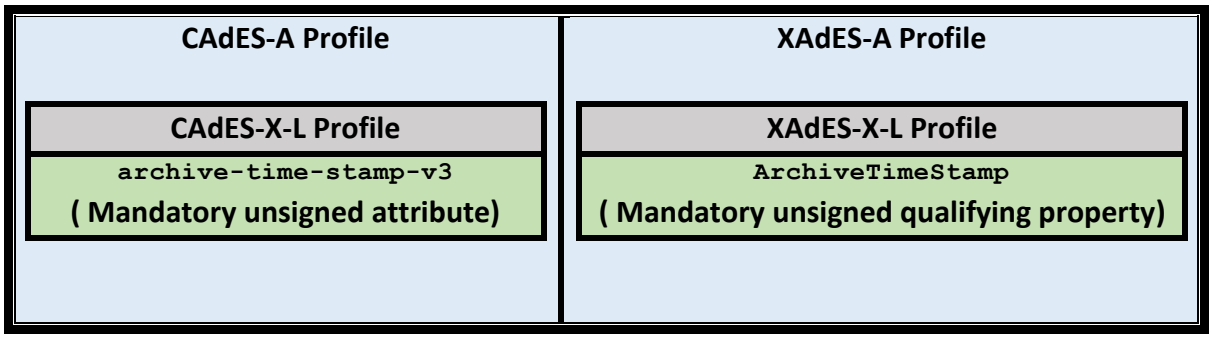

Figure 15: **CAdES-A / XAdES-A Profile**

APPENDIX I

ACRONYMS

ASN.1: Abstract Syntax Notation One
CAdES: CMS Advanced Electronic Signature
CAdES-A: CAdES with Archive validation data
CAdES-BES: CAdES Basic Electronic Signature
CAdES-C: CAdES with Complete validation data
CAdES-EPES: CAdES Explicit Policy Electronic Signature
CAdES-T: CAdES with Time Stamp
CadES-X-L: CadES with Extended Long validation data
CAdES-X: CAdES with eXtended validation data
CMS: Cryptographic Message Syntax
CRL: Certificate Revocation List
HTTP: HyperText Transfer Protocol
IETF: The Internet Engineering Task Force
ETSI: European Telecommunications Standards Institute
OCSP: Online Certificate Status Protocol
PDF: Portable Document Format
PAdES: PDF Advanced Electronic Signature
PKCS: Public-Key Cryptography Standards
RFC: Request For Comments
RSA: Rivest Shamir Adelmen
S/MIME: Secure Multipurpose Mail Extension
URI: Uniform Resource Identifier
W3C: World Wide Web Consortium
XAdES: XML Advanced Electronic Signatures
XML: eXtensible Markup Language
XMLDSig: XML Digital Signature

APPENDIX II

REFERENCES

- [**RFC**] RFC 5652, **Cryptographic Message Syntax (CMS)**, September 2009.
- [**RFC**] RFC 2315, **PKCS #7: Cryptographic Message Syntax Version 1.5**, March 1998.
- [**RFC**] RFC 5751, **Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification**, January 2010.
- [**RFC**] RFC 3275, **(Extensible Markup Language) XML-Signature Syntax and Processing**, March 2002
- [**ISO**] ISO 32000-1, **PDF (Portable Document Format), version 1.7**, July 2008.
- [**ETSI**] ETSI 319 122-1, **Electronic Signatures and Infrastructures (ESI); CAdES digital signatures; Part 1: Building blocks and CAdES baseline signatures,** April 2016.
- [**ETSI**] ETSI 319 122-2, **Electronic Signatures and Infrastructures (ESI); CAdES digital signatures; Part 2: Extended CAdES signatures,** April 2016.
- [**ETSI**] ETSI 319 132-1, **Electronic Signatures and Infrastructures (ESI); XAdES digital signatures; Part 1: Building blocks and XAdES baseline signatures**, April 2016.
- [**ETSI**] ETSI 319 132-2, **Electronic Signatures and Infrastructures (ESI); XAdES digital signatures; Part 2: Extended XAdES signatures**, April 2016.
- [**ETSI**] ETSI 319 142-1, **Electronic Signatures and Infrastructures (ESI); PAdES digital signatures; Part 1: Building blocks and PAdES baseline signatures**, April 2016.
- [**ETSI**] ETSI 319 142-2, **Electronic Signatures and Infrastructures (ESI); PAdES digital signatures; Part 2: Extended PAdES signatures**, April 2016.
- [ETSI] ETSI EN 319 162-1, **Electronic Signatures and Infrastructures (ESI); Associated Signature Containers (ASiC); Part 1: Building blocks and ASiC baseline containers,** April 2016.

APPENDIX III

CAdES-BES, CAdES-EPES, CAdES-T, CAdES-C, CAdES-X, CADES-X-L and CAdES-A Signatures

Signature fields/Attributes/Services	CAdES-BES	CAdES-EPES	CAdES-T	CAdES-C	CAdES-X	CAdES-X-L	CAdES-A
content-type	shall be present	shall be present	shall be present	shall be present	shall be present	shall be present	shall be present
message-digest	shall be present	shall be present	shall be present	shall be present	shall be present	shall be present	shall be present
Service: protection of signing certificate	shall be provided	shall be provided	shall be provided	shall be provided	shall be provided	shall be provided	shall be provided
SPO: ESS signingcertificate	conditioned presence	conditioned presence	conditioned presence	conditioned presence	conditioned presence	conditioned presence	conditioned presence
SPO: ESS signingcertificate-v2	conditioned presence	conditioned presence	conditioned presence	conditioned presence	conditioned presence	conditioned presence	conditioned presence
signing-time	may be present	may be present	may be present	may be present	may be present	may be present	may be present
commitment-type-indication	may be present	may be present	may be present	may be present	may be present	may be present	may be present
content-hints	may be present	may be present	may be present	may be present	may be present	may be present	may be present
mime-type	may be present	may be present	may be present	may be present	may be present	may be present	may be present
signer-location	may be present	may be present	may be present	may be present	may be present	may be present	may be present
signer-attributes-v2	may be present	may be present	may be present	may be present	may be present	may be present	may be present
countersignature	may be present	may be present	may be present	may be present	may be present	may be present	may be present
content-time-stamp	may be present	may be present	may be present	may be present	may be present	may be present	may be present

signature-policy-identifier	*	**shall be present**	**may be present**	**may be present**	**may be present**	**may be present**	**may be present**
signature-policy-store	*	**conditioned presence**	**conditioned presence**	**conditioned presence**	**conditioned presence**	**conditioned presence**	**conditioned presence**
content-reference	**may be present**	**may be present**	**may be present**	**may be present**	**may be present**	**may be present**	**may be present**
content-identifier	**may be present**	**may be present**	**may be present**	**may be present**	**may be present**	**may be present**	**may be present**
signature-time-stamp	*	*	**shall be present**	**shall be present**	**shall be present**	**shall be present**	**shall be present**
complete-certificate-references	*	*	*	**shall be present**	**shall be present**	**shall be present**	*
complete-revocation-references	*	*	*	**shall be present**	**shall be present**	**shall be present**	*
attribute-certificatereferences	*	*	*	**conditioned presence**	**conditioned presence**	**conditioned presence**	*
attribute-revocationreferences	*	*	*	**conditioned presence**	**conditioned presence**	**conditioned presence**	*
CAdES-C-timestamp	*	*	*	*	**shall be present** in X-Type1	**shall be present** in X-L-Type1	*
time-stamped-certs-crlsreferences	*	*	*	*	**shall be present** in X-Type2	**shall be present** in X-L-Type2	*
Service: certificate values in longterm validation	*	*	*	*	*	**shall be provided**	**shall be provided**
SPO: SignedData.certificates	*	*	*	**may be present**	**may be present**	**may be present**	**conditioned presence**
SPO: certificate-values	*	*	*	*	*	**shall be present**	*
Service: revocation values in longterm validation	*	*	*	*	*	**shall be provided**	**shall be provided**
SPO: SignedData.crls.crl	*	*	*	**may be present**	**may be present**	**may be present**	**conditioned presence**

SPO: SignedData.crls.other	*	*	*	**may be present**	**may be present**	**may be present**	**conditioned presence**
SPO: revocation-values	*	*	*	*	*	**shall be present**	*
archive-time-stamp-v3	*	*	*	*	*	*	**shall be provided**

Note: The requirements shall be implemented, on the attributes and certain signature fields, for each signature level are expressed in the table. The values that can appear in columns are:

- **shall be present**: means that the attribute or signature field is mandatory, and shall be as specified in the document referenced in **APPENDIX II** "**References**".
- **shall not be present**: means that the attribute or signature field shall not be part of the structure.
- **may be present**: means that the attribute or signature field is not mandatory but can be included to add extra information to the structure, and shall be as specified in the document referenced in **APPENDIX II** "**References**".
- **shall be provided**: means that the service identified shall be provided as further specified in the SPO-related rows. This value only appears in rows that contain requirements for services.
- **conditioned presence**: means that the presence of the item identified is conditioned as per the requirement(s) specified in the document referenced in **APPENDIX II** "**References**".
- "*****": means that the attribute or the signature field (service) should not be present (provided) in the corresponding level.

APPENDIX IV

XAdES-BES, XAdES-EPES, XAdES-T, XAdES-C, CAdES-X, XADES-X-L and XAdES-A Signatures

Elements/Qualifying properties/Services	XAdES-BES	XAdES-EPES	XAdES-T	XAdES-C	XAdES-X	XAdES-X-L	XAdES-A
SigningTime	may be present	may be present	may be present	may be present	may be present	may be present	may be present
SigningCertificateV2	conditioned presence	conditioned presence	conditioned presence	conditioned presence	conditioned presence	conditioned presence	conditioned presence
CommitmentTypeIndication	may be present	may be present	may be present	may be present	may be present	may be present	may be present
DataObjectFormat	may be present	may be present	may be present	may be present	may be present	may be present	may be present
SignatureProductionPlaceV2	may be present	may be present	may be present	may be present	may be present	may be present	may be present
SignerRoleV2	may be present	may be present	may be present	may be present	may be present	may be present	may be present
CounterSignature	may be present	may be present	may be present	may be present	may be present	may be present	may be present
AllDataObjectsTimeStamp	may be present	may be present	may be present	may be present	may be present	may be present	may be present
IndividualDataObjectsTimeStamp	may be present	may be present	may be present	may be present	may be present	may be present	may be present
SignaturePolicyIdentifier	*	shall be present	may be present	may be present	may be present	may be present	may be present
SignaturePolicyStore	*	conditioned presence	conditioned presence	conditioned presence	conditioned presence	conditioned presence	conditioned presence
SignatureTimeStamp	*	*	shall be present	shall be present	shall be present	shall be present	shall be present
CertificateValues	*	*	*	*	*	conditioned presence	conditioned presence
AttrAuthoritiesCertValues	*	*	*	*	*	conditioned presence	conditioned presence
RevocationValues	*	*	*	*	*	conditioned presence	conditioned presence

AttributeRevocationValues	*	*	*	*	*	conditioned presence	conditioned presence
Service: incorporation of validation data for electronic time-stamps	*	*	*	*	*	*	shall be provided
SPO: TimeStampValidationData	*	*	*	*	*	*	conditioned presence
SPO: certificate and revocation values embedded in the electronic time-stamp itself	*	*	*	*	*	*	conditioned presence
ArchiveTimeStamp	*	*	*	*	*	*	shall be present
RenewedDigest	*	*	*	*	*	*	conditioned presence
CompleteCertificateRefsV2	*	*	*	shall be present	shall be present	shall be present	*
AttributeCertificateRefsV2	*	*	*	conditioned presence	conditioned presence	conditioned presence	*
CompleteRevocationRefs	*	*	*	shall be present	shall be present	shall be present	*
AttributeRevocationRefs	*	*	*	conditioned presence	conditioned presence	conditioned presence	*
RefsOnlyTimeStampV2	*	*	*	*	shall be present in X-Type2	shall be present in X-L-Type2	*
SigAndRefsTimeStampV2	*	*	*	*	shall be present in X-Type1	shall be present in X-L-Type1	*

Note: The requirements shall be implemented, on the attributes and certain signature fields, for each signature level are expressed in the table. The values that can appear in columns are:

- **shall be present**: means that the attribute or signature field is mandatory, and shall be as specified in the document referenced in **APPENDIX II** "**References**".
- **shall not be present**: means that the attribute or signature field shall not be part of the structure.
- **may be present**: means that the attribute or signature field is not mandatory but can be included to add extra information to the structure, and shall be as specified in the document referenced in **APPENDIX II** "**References**".

- **shall be provided**: means that the service identified shall be provided as further specified in the SPO-related rows. This value only appears in rows that contain requirements for services.
- **conditioned presence**: means that the presence of the item identified is conditioned as per the requirement(s) specified in the document referenced in **APPENDIX II** "**References**".
- "*": means that the attribute or the signature field (service) should not be present (provided) in the corresponding level.

APPENDIX V

PAdES-BES and PAdES-EPES signatures

Attribute /Field/Service	PAdES-BES	PAdES-EPES
content-type	shall be present	shall be present
message-digest	shall be present	shall be present
signing-certificate reference	shall be provided	shall be provided
signer-attributes-v2	may be present	may be present
content-time-stamp	may be present	may be present
signature-time-stamp	may be present	may be present
commitment-type-indication	Conditioned presence	Conditioned presence
entry with key M in the Signature Dictionary	may be present	may be present
entry with key Location in the Signature Dictionary	may be present	may be present
entry with key Reason in the Signature Dictionary	Conditioned presence	Shall not be present
entry with key Filter in the Signature Dictionary	shall be present	shall be present
entry with key ByteRange in the Signature Dictionary	shall be present	shall be present
entry with key SubFilter in the Signature Dictionary	shall be present	shall be present
entry with key Contents in the Signature Dictionary	shall be present	shall be present
entry with key Name in the Signature Dictionary	may be present	may be present
entry with key ContactInfo in the Signature Dictionary	may be present	may be present
entry with key Cert in the Signature Dictionary	Shall not be present	Shall not be present
signature-policy-identifier	Shall not be present	shall be present

APPENDIX VI

Associated Signature Containers (ASiC)

- **Concept**

Associated Signature Containers (ASiC) specifies the use of container structures to bind together one or more signed objects with either advanced electronic signatures or timestamp tokens into one single digital container.

- **Basic container structure**

The **ASiC** is a data container holding a set of file objects and associated digital signatures and/or time assertions using the ZIP format. Any **ASiC** container has an internal structure including:

  - A **root folder**, for all the container content possibly including folders reflecting the content structure.
  - A "**META-INF**" folder, in the root folder, for files containing metadata about the content, including associated signature or time assertion files.

- **Container Types**

1- The first type is **ASiC Simple** (**ASiC-S**) that associates one single file object with either:
   - One **signature file**.
   - One **time assertion file**.

   This type of container can also include a file named "**mimetype**" specifying the media type. This type of container allows to add at a later time additional signatures signing the aforementioned file object and additional **ASiCArchiveManifest** files to protect long term time-stamp tokens.

2- The second type is **ASiC Extended** (**ASiC-E**), a container that associates one or more file objects with either:
   - One or more XAdES signatures present within one or more signature files and optionally one or more ERS within one or more time assertion files.
   - One or more CAdES signatures present within one or more CAdES object files and/or one or more time assertions within one or more time assertion files.

A **signature file** can contain either:	A **time assertion** file can contain either:
• One **CAdES** object. • One or more **XAdES** signatures.	• One time-stamp token • One Evidence Record