NATIONAL SECURITY AUTHORITY

P.O.BOX 16 Budatínska 30, 850 07 Bratislava 57
Slovak Republic
Tel.: +421 2 6869 1111, Fax: +421 2 6382 4005, e-mail: podatelna@nbusr.sk

Bratislava 1.03.2012
Ref.: 2071/2012/IBEP/OEP-001

Experts of technical group on e-procedures
under the Services Directive and
Anneli Andresson-Bourgey

EUROPEAN COMMISSION
Internal Market and Services DG
Services
Services I
B-1049 Bruxelles / Brussel - Belgium
Tel. (32-2) 296.71.19

Subject: Clarification of the answers on the questions raised by ETSI ESI STF and proposals for ETSI ESI STF members under M/460 according to STF questions provided on Brussels technical workshop in the Context of the Implementation of the Service Directive.

Dear all,
I would like to clarify position of the NSA (which is competent body for electronic signature in Slovakia) related with the answers on the questions raised by ETSI ESI STF people and for that reason I have prepared many easy understandable pictures. It is mainly about the questions which we discussed on previous technical group meetings on e-procedures under the Services Directive and especially on CD 2011/130/EU implementation.

Many already identified and known problems which must be fixed in ETSI ESI standards are for the members of ETSI ESI known for many years but are still open and not solved. There are new and new procedures or rules are defined by ESI in ETSI documents instead of fixing a main old incorrect ones. Some of them are in the following text and some were presented on meetings like requirements to define CAdES, PAdES and XAdES unsigned attribute/element where the document of machine processable signature policy will be included for AdES long-term validation formats. For that reason many of them were fixed in CD 2011/130/EU to achieve interoperability. Presently we can see in ETSI ESI drafts the status of work and how editors are solving or also ignoring some problems on freely accessible drafts available e.g. on:
http://docbox.etsi.org/ESI/Open/Latest_Drafts/

Summarization of some questions and ESI editors' intents:

1. In the long-term validation a verifier must use only **stable status** information of the certificate validity (in CRL or OCSP). It means, for validation there must be used only information which was **updated after the signature creation time** of the signer certificate validity. In CRL or OCSP the time when the update happened is indicated in the field thisUpdate. ESI editor incorrectly requires for the long-term validation the information which was updated before the signature creation time, what means a **later CRL or OCSP could change the signer certificate status**.
2. CAdES does not require DER encoding of the whole signature as mandatory and CD 2011/130/EU requires DER only for specific fields and also pointing directly to annex (ETSI TS 101 733 V1.8.1 Annex K) where are fixed some mistakes of the normative part of

ETSI TS 101 733. ESI editor requires in the long-term profiles DER encoding of the whole signature what will cause problems in standard situations which we are **not able to solve.** Editor of CAdES defines totally a new attribute for the long-term validation where the same problem remains and he only defines another way of processing of the same rules which are already implemented in the present archive time-stamp. For that reason a new attribute of CAdES **creates new barriers to** interoperability and requires expensive development instead of defining a new OID for identification of archive time-stamp where are used unique rules as defined in ETSI TS 101 733 V1.8.1 Annex K and in present applications such easy correction can be realized very quickly.

3. Presently many AdES signatures protect by hash values only the **content** of the signed document without any information of interpretation or processing of signed document. For that reason such information are predicted by the other not standardized or protected way where we could have dangerous interoperability problems. It means it is expected that a checker determines the format of Signed Document (SD) then analyses it in order to check that the document is conformant to this format. Practically it is not possible in many situations because SD formats do not have e.g. a standard header for the document identification and it is possible to have many formats which look the same e.g. based on TXT document (ASCI or Unicode or XML, HTML, PDF, MIME… all are TXT SD), binary documents like pictures, formats which use ZIP containers (DOCX, ODF, JAVA), audio or video which are also in many cases without identification header inside SD. For that reason it is crucial that the type of interpretation and processing of SD will be protected by digital signature in signed attributes and such type of SD will be provided for a higher level where the SD is used as trusted information from the signature validation level. For better imagination and understanding of this practical problem I have included into the ZIP in attachment "sd-sca-sva.zip" documents of types ASCII, XML, PDF/A, and TIFF without the type extension in file name and you could **try to read it or process it**. :o)
4. The protection of the signed document can be realized in interoperable way only when the type of protection is automatically detected by verification application and such protection is correctly checked. CD 2011/130/EU defines a minimal format of AdES where such protection is used in a standard way and is automatically identifiable in other applications. For that reason in the interoperable solution there must not be used special rules which require the usage of additional data located in the signed document. Such rules are not useful for interoperability because additional data are processed in a non-standard way usually defined for particular implementation for specific systems or document formats. ETSI ESI defines ASiC-S for interoperability where rules are unique. Example of non-interoperable solution is ASiC-E format where ESI makes a useful work and describes many techniques of how some specific implementations of some types of documents have defined special rules for signature usage only for specific type of the document. Present ASiC-E does not contain warning that ASiC-E is not intended for interoperability because standard AdES application must implement specific additional rules where when such rules are not implemented the signature validation is not able to validate the document integrity and only the document containing references to other document is verified. Such type of signature is also defined on national level as Integrity signature where the text document contains references to many electronic files and it is up to the user how such references are processed because the signature protects only the content of the text document. The example is available on page: http://lockitin.webnode.sk/ http://lockitin.webnode.sk/produkty/create-integrity-signature/ and defined in the standard http://www.nbusr.sk/en/electronic-signature/approved-formats/index.html

Detailed information:

1 AdES creation and long-term validation

The present document defines unique conditions for the AdES creation and validation to achieve the interoperability and the long-term signature validation e.g. in EU. These AdES profiles are intended for a wide interoperability usage and for that reason the signed document must be protected by hash value directly included in signed attributes or signed elements. It means the technique where the hash value of the signed document is protected on the next level of the separate hash computation, like **ds:Manifest** does, is out of scope of these profiles because the rules of the hash computation on the next level are specific for particular application and implementation in specific usage case. The hash computation required by a particular application for the next level separate hash computation can be ignored in many standard applications where CAdES or XAdES are implemented as described in e.g. http://www.w3.org/TR/xmldsig-core/#sec-o-Manifest and for that reason such usage is also restricted in CD2011/130/EU Annex Table 1 in row "ds: Reference URI M One reference to every original data object to be signed (URIs can point to external object as well), + reference to SignedProperties element".

The following examples provide a simple view of main differences between an interoperable signature format and non-interoperable signatures.

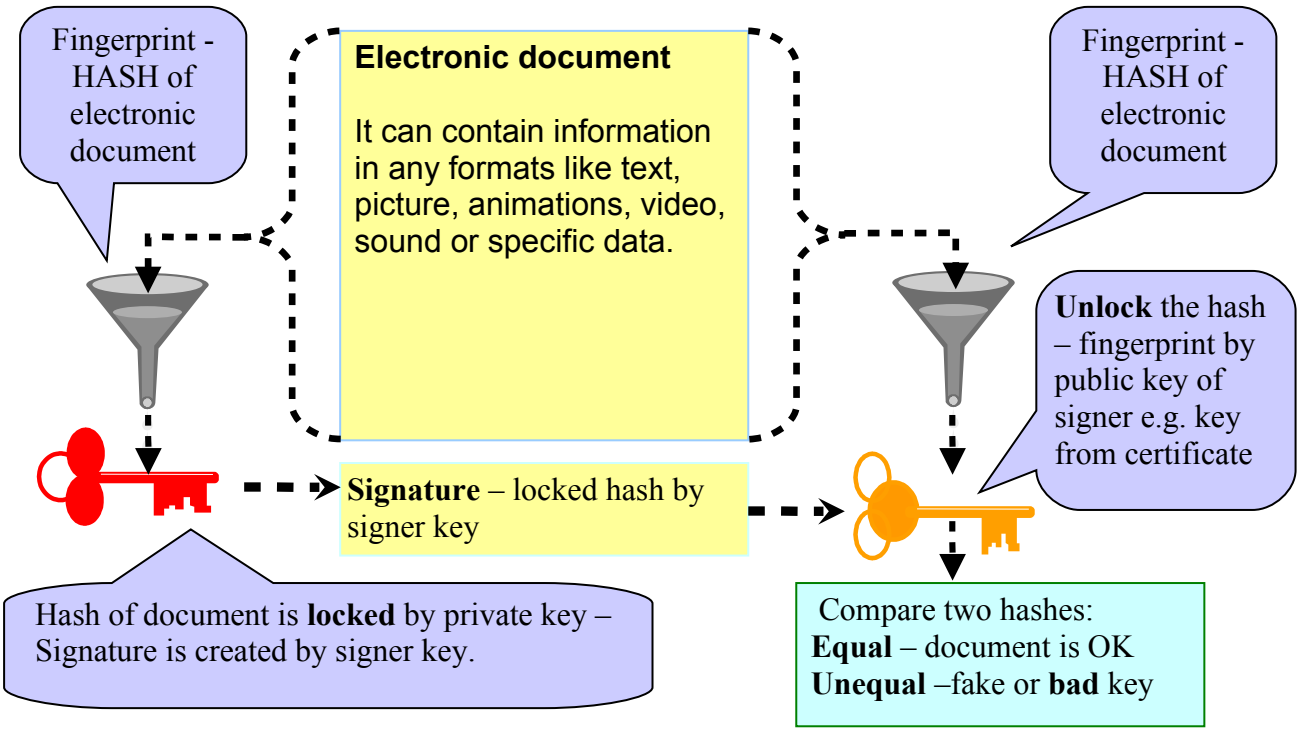


Figure 1: Interoperable signature format where the hash is computed in a standard way

Non-interoperable signatures define particular rules how the hash computation must be performed in particular applications for specific use case. Such rules are ignored in standard signature formats and for that reason the result in standard signature application does not have the computation and checking of hash of additional electronic documents **what causes dangerous security problems. When the interoperability is required then only ASiC-S must be used because ASiC-E requires additional processing of signed document, what could be a security problem**.
ASiC-E and ds:Manifest can be correctly processed only in specific applications which are able to manage additional rules which are not common in present interoperable applications as shown in the example ETSI TS 102 918 V1.2.1 (2012-02) B.3 Example of ASiC-E with CAdES "/META-INF/ASiCmanifest1.xml" containing the hash of file1.xml and file2.xml.

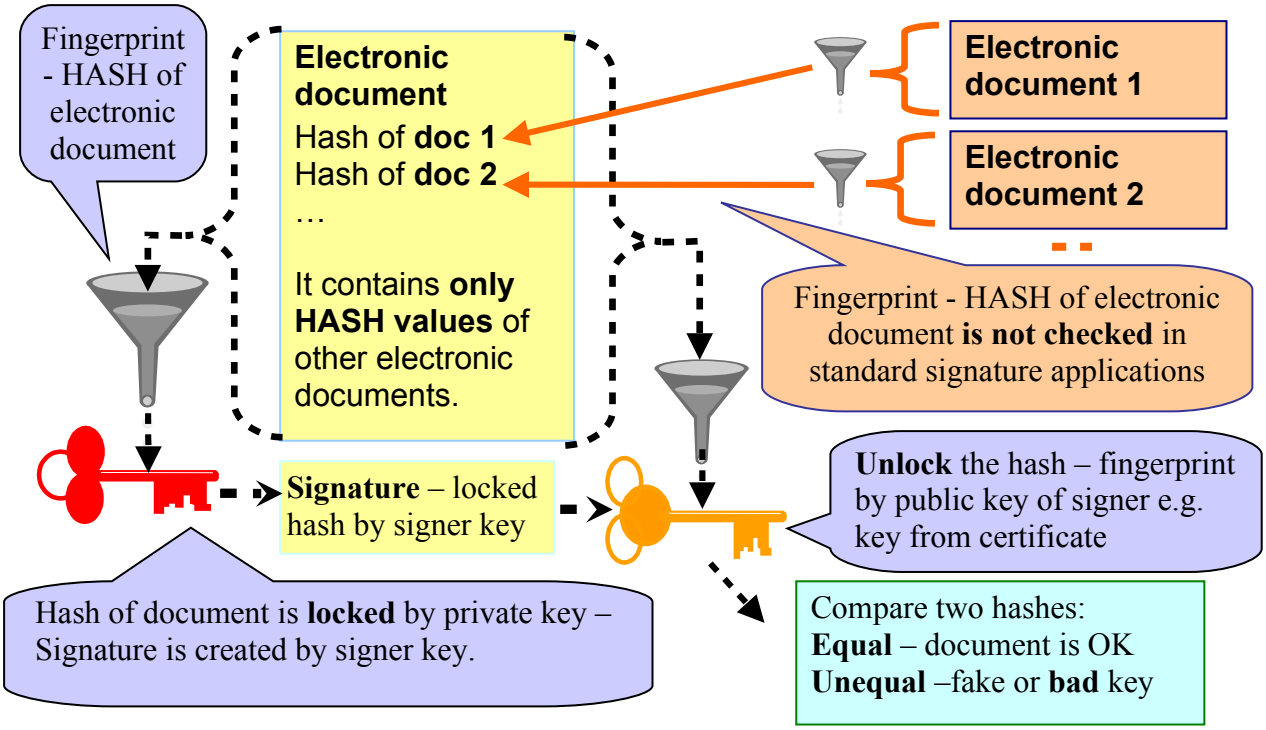


Figure 2: Non-interoperable signature where a signed document points to another documents

This chapter describes conditions based on TS 101 733 CMS Advanced Electronic Signatures (CAdES). The other AdES formats like TS 102 778 PDF Advanced Electronic Signature (PAdES) as well as TS 101 903 XML Advanced Electronic Signatures (XAdES) can use the same rules adapted according to their signature formats.
When the usage of the signed document and the signature are expected for the long-term perspective then there must exist provable evidence that a particular object was existing in a particular time. The object which can be used as a proof of evidence (PoE) of particular object existence in the past is e.g. an archive timestamp (ATS) or time mark where at least two types of information are present: the protection of data and the protection of time when such protection of data was realized. PoE protects from the usage of fake objects created e.g. now with algorithms broken and now claiming that the object was created in the past.

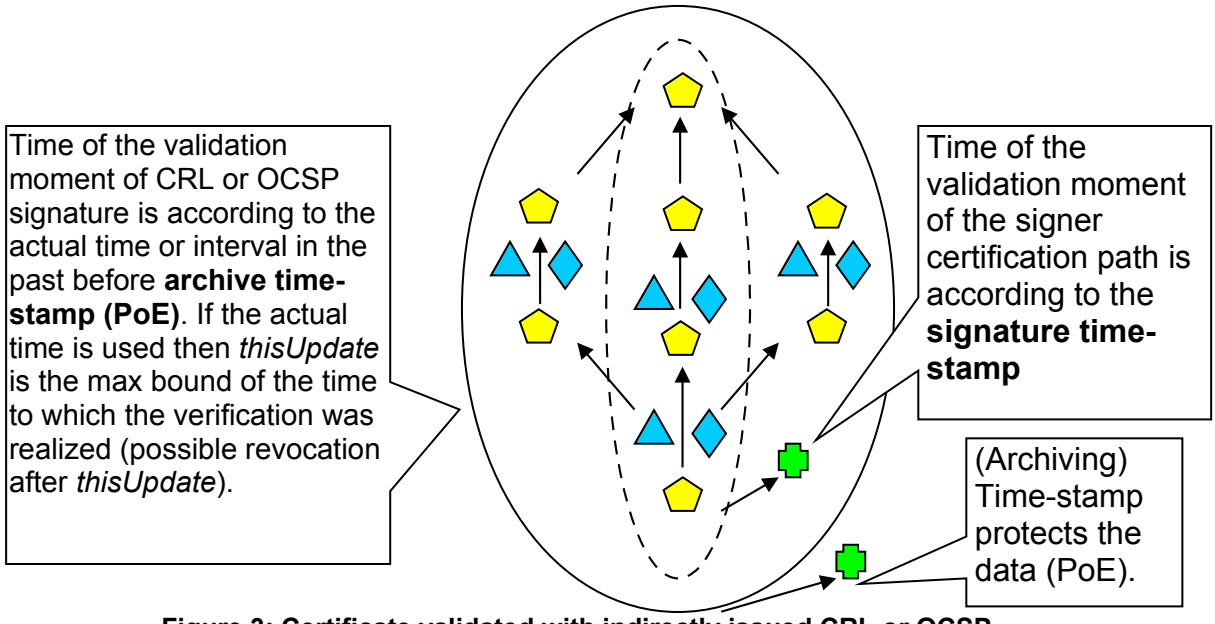


Figure 3: Certificate validated with indirectly issued CRL or OCSP

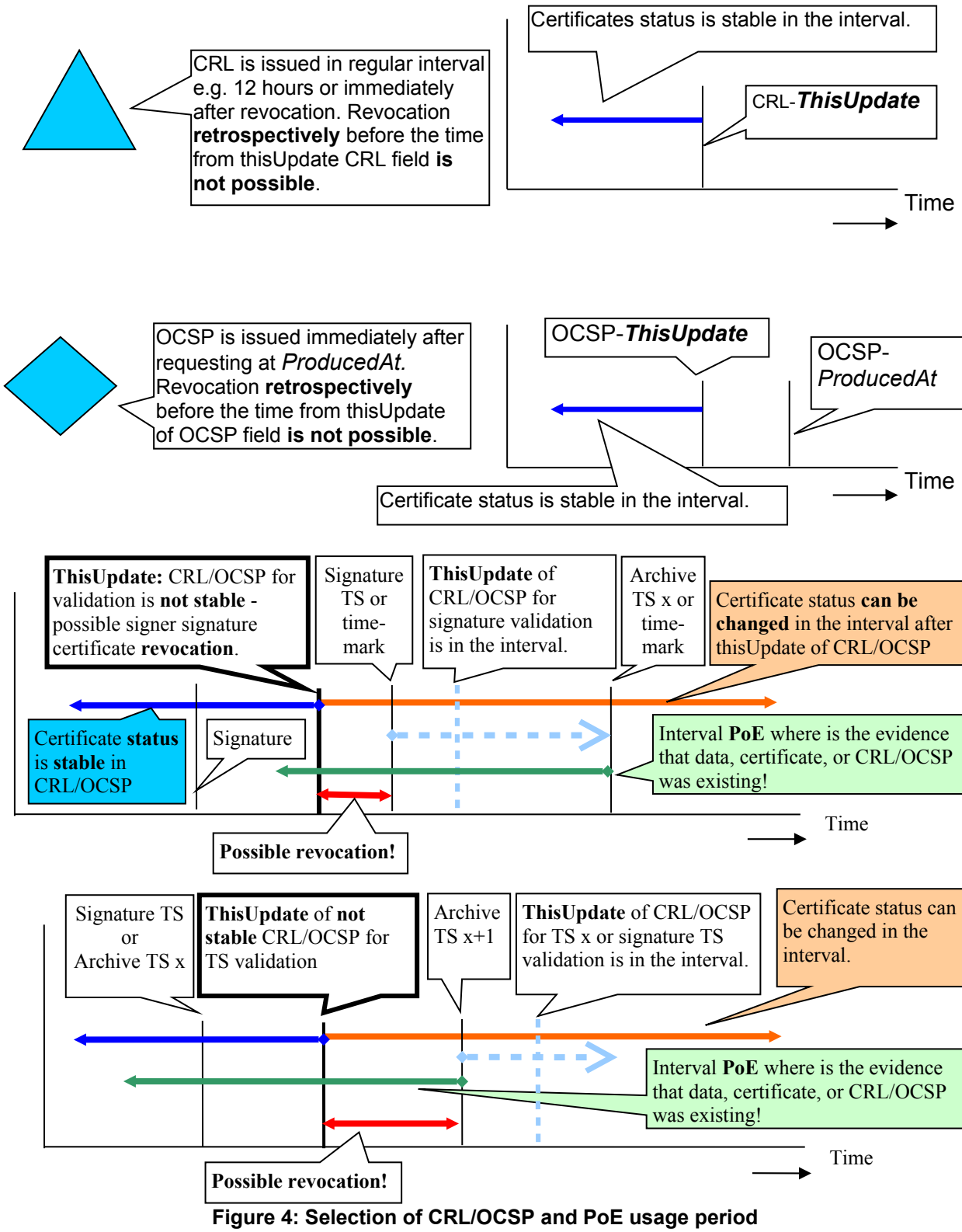


Figure 4: Selection of CRL/OCSP and PoE usage period

The long-term validation must use only information which was **updated after the time to which we validate** the certificate. Information updated before that time is not stable and **later** could be **changed**. Selection of CRL or OCSP response must be according to ***thisUpdate*** time which must be greater than the signature verification time (signature time-stamp or signature time-mark). Any new certificate revocation time must be after ***thisUpdate*** time of the latest CRL or OCSP. It means

before the time value thisUpdate of CRL or OCSP the new revocation must not be realized as a basic rule where backward revocation is not permitted.

Signature must not be intended only for protection of a content of electronic documents without any information about the type of interpretation and processing of the electronic document.
It means it is expected that a checker of the document content determines the format of the signed document according to additional information protected by signature then analyses the document in order to check that the document is conformant to this format.

Without additional information determining the format of the signed document according to analyses of the document is practically not possible in many situations because formats of signed documents do not have e.g. a standard header for identification and it is possible to have many formats which look the same e.g. based on TXT document (ASCI or Unicode or XML, HTML, PDF, MIME… all are TXT documents), binary documents like pictures, formats which use ZIP containers (DOCX, ODF, JAVA), audio or video which are also in many cases without identification header inside the signed document.

For that reason it is crucial that **the type of interpretation and processing** of the signed document will be protected by digital signature in signed attributes and such type of the signed document will be provided for a higher level as trusted information from the signature validation level.

1.1 The identification and processing of the signed document type.

1.1.1 CAdES

When the *id-contentType* OID *id-data (1.2.840.113549.1.7.1)* of *SignerInfo - SignedAttributes* (RFC 5652) is not able to identify uniquely the visualization type of signed data then the signed attribute *id-aa-contentHint* (RFC 2634, RFC 5035, RFC 5911) is recommended to be included and contains the field *contentDescription* with the text *Content-Type:* which specifies the MIME type (RFC 2045) of visualization and optional *name* parameter (or *Content-Disposition: attachment; filename*) which contains the proposed file name of signed data in additional processing of signed data e.g. in export and provides a hint of the possible file extension name.

An example of *ContentHints* usage defined in ETSI TS 101 733 V1.8.3 (2011-01) paragraph 5.10.3 content-hints Attribute:

```
Attribute SEQUENCE {
 attrType OBJECT IDENTIFIER 1.2.840.113549.1.9.16.2.4 id-aa-contentHint
 attrValues SET {
  ContentHints SEQUENCE {
   contentDescription UTF8String 'MIME-Version: 1.0
    Content-Type: text/plain; charset=UTF-8; name="Document.txt"
    Content-Disposition: attachment; filename="Document.txt"'
   contentType OBJECT IDENTIFIER 1.2.840.113549.1.7.1 id-data
  }
 }
}
```

1.1.2 XAdES

Interoperable QES based on XAdES signatures must contain the signed element *DataObjectFormat* which must contain visualization type identification in *MimeType* element of data to which the reference element points after all reference transformation were applied in order to achieve a unique visualization. When signed data are intended to be exported from XML signature or the signature is detached then it is strongly recommended that *DataObjectFormat* contains at least one <*xades:Description*> element which contains *Content-Type* MIME header field to allow a unique signed data identification for visualization or processing and to provide a hint of the possible file name and extension name in *name* parameter. The element <*xades:Description*> shall be used to

indicate the encoding of the data, in accordance with the rules defined in RFC 2045; see an example of structured contents and MIME.

The purpose of the MIME *Content-Type* field is to describe the data being signed fully enough that the receiving user agent can pick an appropriate agent or mechanism to present the data to the user, or otherwise deal with the data in an appropriate manner. Examples of usage *DataObjectFormat* ETSI TS 101 903 and <*xades:Description*>::

```
<xades:DataObjectFormat ObjectReference="...">
  <xades:Description>
    Content-Type: text/plain; charset=UTF-8; name="Document.txt"
  </xades:Description>
  <xades:MimeType>text/plain</xades:MimeType>
 </xades:DataObjectFormat>
```

The `ds:Reference` element must not reference to a ds:Manifest to achieve the interoperability as defined in http://www.w3.org/TR/xmldsig-core/#sec-Manifest because the rules how the hash computation is realized in the manifest are defined in a particular application and might not be checked in other applications. It happens because other applications are not able to detect additional rules for the separate hash computation.

1.2 CAdES archive timestamp hash calculation

Each archive-time-stamp attribute must be included in a new *UnsignedAttributes-Attribute*, it means the SET of *UnsignedAttributes-Attribute-attrValues* MUST contain only **one** archive-time-stamp attribute.

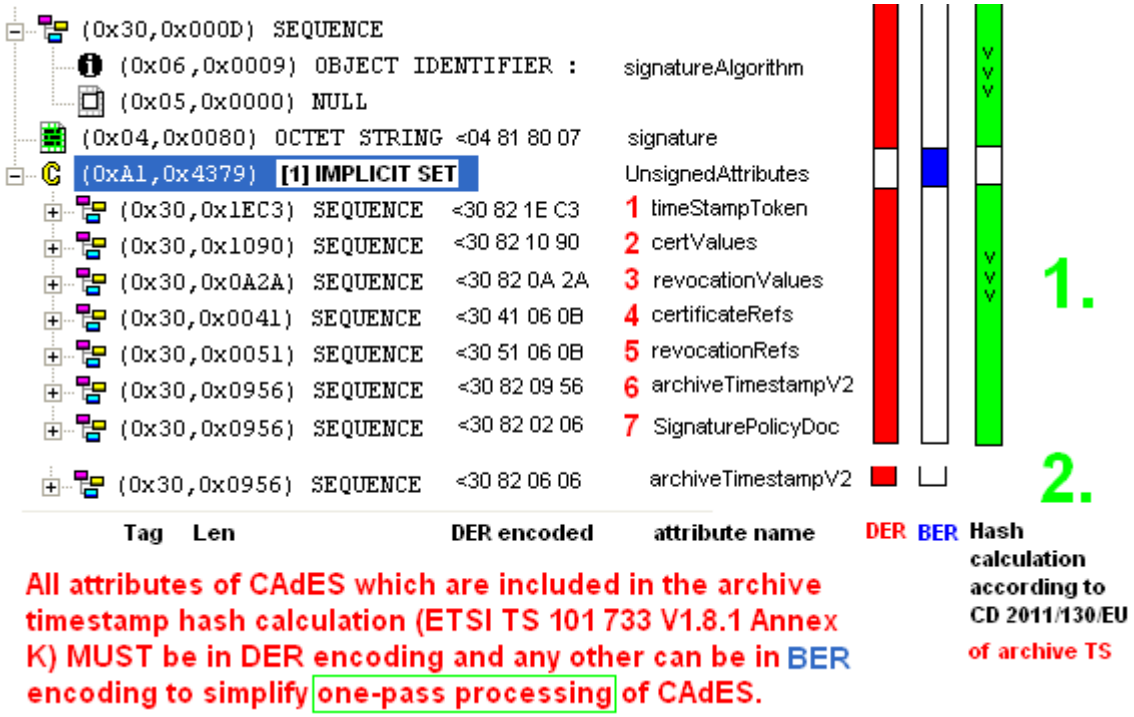
```
UnsignedAttributes ::= SET SIZE (1..MAX) OF Attribute
  Attribute ::= SEQUENCE {
    attrType OBJECT IDENTIFIER,
    attrValues SET OF AttributeValue }
```

The hash value calculation for the time-stamp MUST be according to rules defined in ETSI TS 101 733 V1.8.1 **Annex K Time-stamp hash calculation** and the values used in the hash calculations MUST be used from DER encoded signature without any DER modifications. It means the hash value is calculated over the DER fields directly without the order modification (SET, SET OF), the ASN.1 type or ASN.1 length modification (of DER Type, Length and Value). The field UnsignedAttributes ::= SET SIZE (1..MAX) OF is according to CD 2011/130/EU BER encoded and for that reason MUST not be ordered when at least one archive time-stamp is present.
For that reason the encoding of unsignedAttrs [1] IMPLICIT SET must be BER (Basic Encoding Rules) where the hash of attributes included in unsignedAttrs [1] IMPLICIT SET is computed in the same order of attributes as the unsigned attributes were included in the SET and the application must not change the order of unsigned attributes and must not include, into hash computation, the type and length of SET - unsignedAttrs [1] IMPLICIT SET.

1.3 DER and BER according to CD 2011/130/EU

A new archive time-stamp will be included into the signature according to CD 2011/130/EU where the DER is required only in fields which are used for archive timestamp hash calculation defined in ETSI TS 101 733 V1.8.1 **Annex K Time-stamp hash calculation**. Creation and also verification of ATS does not require any reordering of attributes or including the SET into the hash calculation.

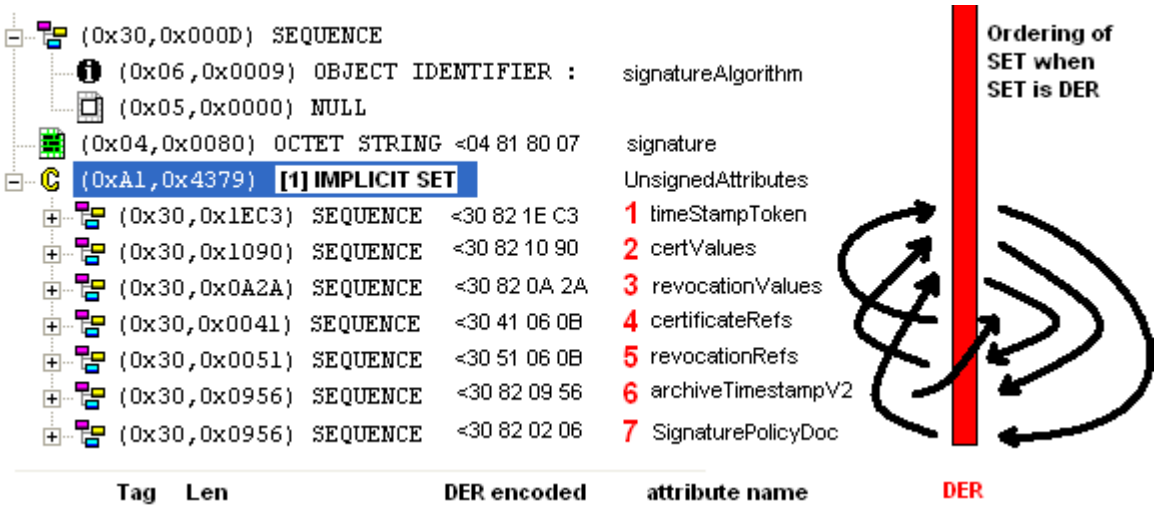
ATS creation/validation:
1. Create the archive hash value of attributes as indicated in the green column.
2. Get a time-stamp from TS authority and add a new unsigned archive time-stamp attribute.



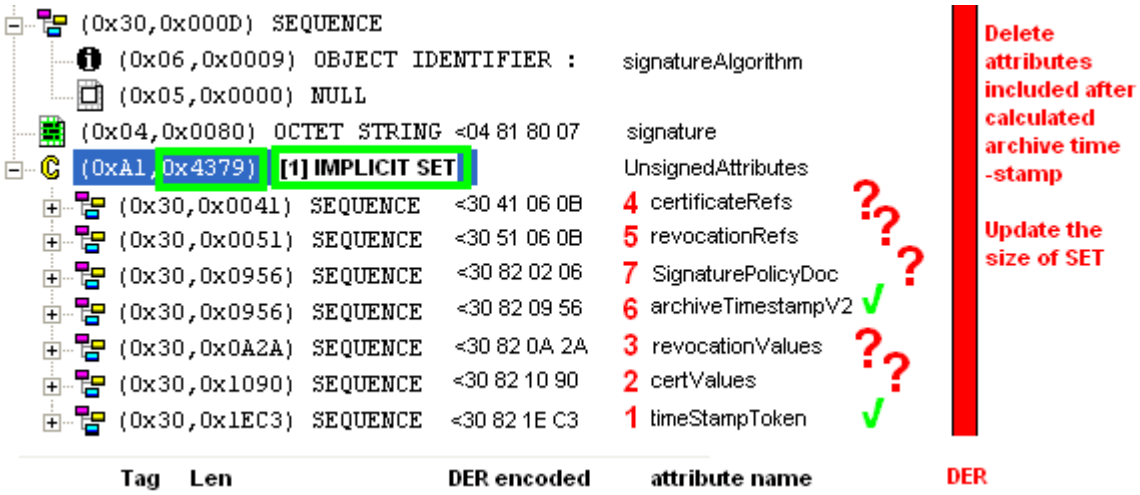
1.4 Requirements for the whole DER of signature cases complications which are unsolvable in standard situations.

Validation (creation) of archive time-stamp included in the signature according to incorrect definition which fails in standard situations as it is incorrectly defined in clause 6.4.1 archive-time-stamp Attribute Definition of ETSI TS 101 733 V1.8.3 (2011-01). The definition requires the hash calculation **"- all data elements in the SignerInfo sequence including all signed and unsigned attributes."** where also SET OF unsigned attributes is included and when SET OF is required to be **DER** then unsigned attributes must be before the hash computation **ordered** and unsigned attributes, which were included into the signature in time or after the time of actually calculated archive timestamp, must be **deleted**, what is a insuperable obstacle because we are not able to detect for many attributes **when** some attributes **were included** in unsigned attributes.

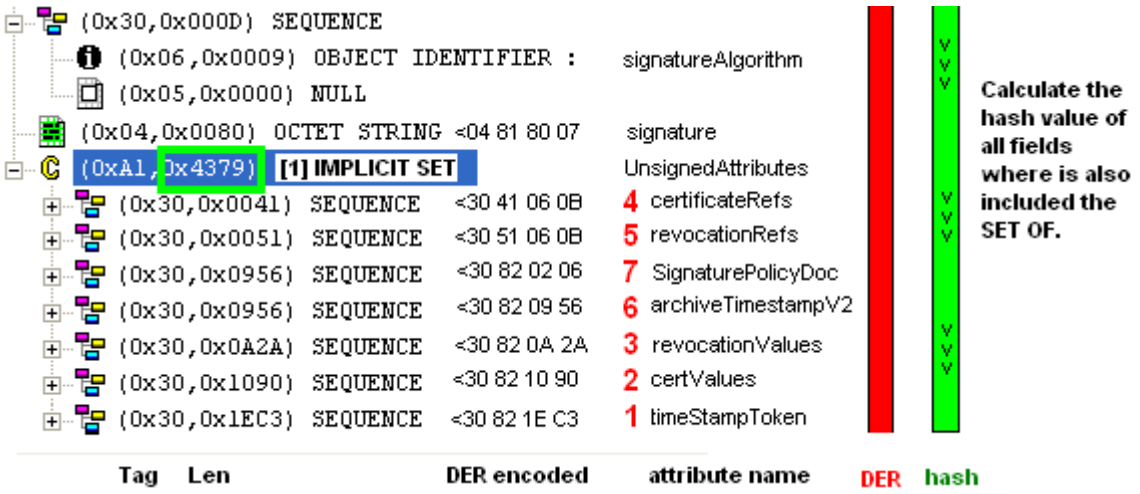
ATS creation/validation:
1. Attributes in SET must be ordered when SET is DER encoded – ordering of unsigned attributes according to the binary encoding of each attribute.



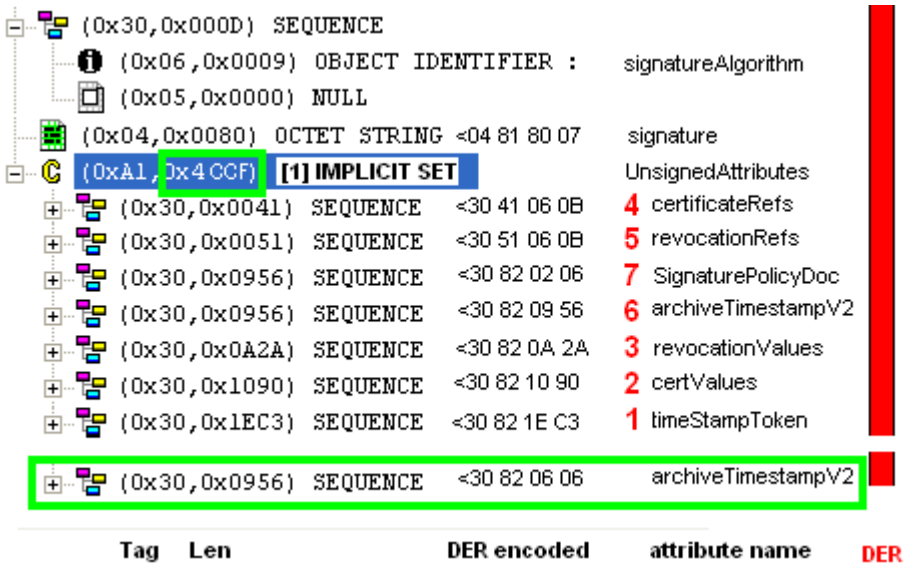
2. **Delete** attributes which were included after the verified archive timestamp. E.g. delete later archive time-stamps or attributes included after verified archive timestamp and **update the length** value of **SET**. It is **unclear** which attributes instead of time-stamp was included after the calculated archive time-stamp because many attributes **do not have time value** when they were included in unsigned attributes.



3. Calculate the archive hash value of attributes as indicated in the green column.

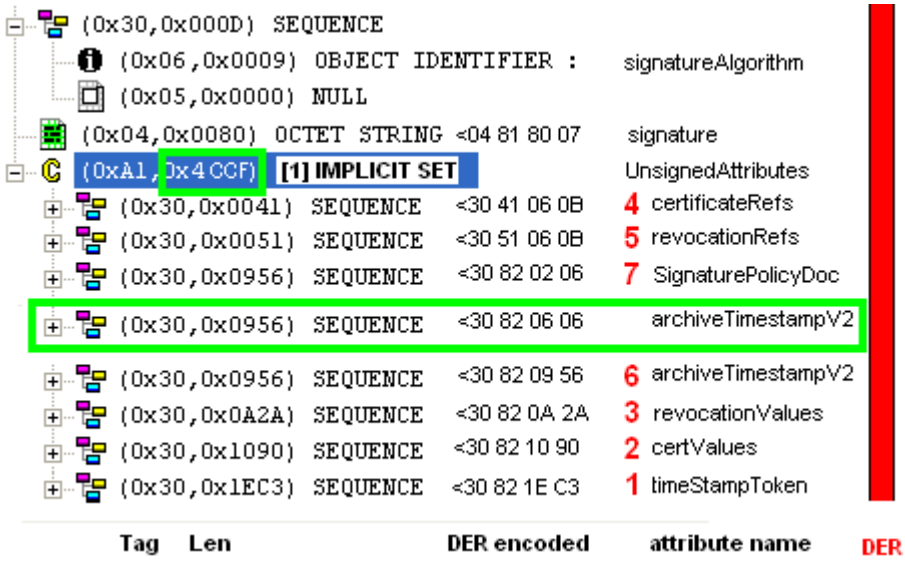


4. Get a time-stamp from TS authority and add a new unsigned archive time-stamp attribute and update the size of SET.



Add a new archive time -stamp and update the size of a SET of unsigned attributes.

5. The attributes in SET must be ordered when DER is required.



Atributes of SET, when DER is required, must be ordered.

The same complicated steps are used when the archive time-stamps are validated.

Ing. Peter Rybár
peter.rybar@nbusr.sk
Information Security and Electronic Signature Department
National Security Authority