**Improvement of the signature usage and validation**

The e-Signature Validation remote Plugtest is the best place to check what is used and expected in validation and also in the past validation (a signature created in the past and validated now, hereinafter called as the past validation). At the ETSI ESI meetings were many times declared that for the past validation the validation rules must be based on `thisUpdate` field of CRL or OCSP and the time value in `thisUpdate` field of CRL or OCSP response must be after the signature creation time. It means the status of the certificate was updated after e.g. the signature time stamp. Definitions of the past validation in ETSI TS 102 853 are incorrect. Info about TS 102 853 [http://webapp.etsi.org/workprogram/Report_WorkItem.asp?WKI_ID=41188](http://webapp.etsi.org/workprogram/Report_WorkItem.asp?WKI_ID=41188) and rules according to ITU -T and ISO standards, ignored by TS 102 853 editor, are summarised on the web [http://lockit.webnode.sk/](http://lockit.webnode.sk/) in presentation Electronic signature - simply, long-term, safely and in accordance with Commission Decision 2011/130/EU.

The rules defined in ITU-T and ISO standards can be clarified for the past validation in ETSI ESI according to experiences with validations e.g. the usage of INDETERMINATE is unclear and confusing for common signature users and for the user must be provided an info about what to do when the signature does not contain some objects which are essential for final validation status. In this case correct information like an interval in which the certificate was valid according to used CRL or OCSP can be shown for the user.

The validation must be split to three levels:

- a proof of existence (PoE) is the first level,
- a signer certificate validity is the second level, and
- the integrity check of the signed document with a signature is the third level.

The concept of revocation `freshness` in ETSI TS 102 853 is absurd and incorrect for the past validation because it allows to use CRL or OCSP issued and updated before the signing time, what means, the signature with the revoked certificate will be valid also when the revocation time was before the signature time stamp time, and this is a crucial mistake in TS 102 853.

In each level we must use only directly trusted data (e.g. included in a trusted list) or data protected by an acceptable evidence of existence which guarantees that objects existed before a specific time and only such data can be used in validation as correct data.

- The first level is the creation of the list of proof of existence (PoE is also validated) of all objects in a signature which are validated by a key from a particular certificate (objects like: a signer signature of ES, signature of each CRL, signature of each certificate, signature of each TS or signature of each OCSP response). The PoE, included in the item in the list, is the date/time when the object existed based on the time in which we have trusted evidence like a time-stamp or external evidence of the signature like a time mark or secure evidence in the archive or we have evidence as is defined for delivery service in eIDAS. When the PoE does not exist or is not entered by a verifier and the certificate of the object is not expired or revoked, we use the time value `thisUpdate` of present (actual) CRL or OCSP.
• The second level is the signer certificate validation where we will use the most recent certificate status (OCSP or CRL) which can contain the status of a signer certificate based on the PoE from the list, \textit{thisUpdate} field, certificate.validity.\texttt{notBefore}, certificate.validity.\texttt{notAfter}, CRL.\texttt{expiredCertsOnCRL} and OCSP.\texttt{ArchiveCutoff} or positive statement OCSP[certificate].\texttt{CertHash} (see conditions for CRL or OCSP in Table 8 or 9).

1. When a result of the \textbf{certificate status is not revoked} in used CRL or OCSP responses: A green period of the certificate validity ends at a date/time value of thisUpdate and after that time the status of the certificate is unknown (we do not have a sibyl crystal ball) – unknown period ends in the end of the certificate validity.

2. When a result of the \textbf{certificate status is revoked}

Diagram:

- Certificate status
  - Valid
  - Unknown in used CRL or OCSP

- time

- thisUpdate of CRL or OCSP

- Certificate validity.
  - notBefore
  - notAfter
The green – a valid certificate status period ends at the revocation date/time. The value of \textit{thisUpdate} is not used because it is in a red interval where the certificate is revoked. When the red - revoked certificate status period is used with an optional reason \texttt{certificateHold}, applications in the past validation systems have a \textbf{big problem}:

- When an archive format is created in the period when the certificate is revoked with the optional reason certificateHold, the \textit{signature is forever invalid}.
- When the previous signature archive form is updated in the time when the certificate becomes valid (status revoked with the optional reason certificateHold was removed from CRL and CA database) the \textit{same signature becomes forever valid}.

- The third level is a digital signature validation based on public key from the signer certificate. In this level the integrity of the signature and the signed document is evaluated and protected meta-data by signature of the signed document are collected for the user of the signed document to guarantee a unique processing and interpretation of information stored in the signed document.

Finally when the time, as an \textbf{evidence of existence of signature}, is in the green interval (of the second level), the signature is valid; when the evidence is in the red interval, the signature is invalid and when the evidence is in the yellow interval, the user is informed: Validation must get a new CRL or OCSP which is updated after trusted evidence of the time of the signature existence where the update is indicated in thisUpdate field of OCSP or CRL.
What is important, is, when the timestamp or the external evidence are missing, the application must provide intervals achieved at the second level of validation (the interval based on the certificate validity according to CRL or OCSP) for the user (as a technically correct interval in which the user must decide when the signature was created – to find out some PoE) and it is up to the relying party to decide or find out in which part of intervals the evidence of the signature existence will be acceptable. The indeterminate status is only internal, technical status and users would be lost without additional supporting guidance. The indeterminate status must not be present for the user.

As you can see the validation does not use the field `nextUpdate` because the time value in `nextUpdate` is only a wish (expectation) when a desired action could happen, and it must not be used for technical validation. Only true information must be provided. The concept of revocation freshness in ETSI TS 102 853 for the past validation is a kind of illusion, bluff, lying and such deceiving algorithm must not be used as a European Standard.

See Annex A Table 8 or Table 9
Formats of certificate revocation list and confirming the status and validity of certificates (pdf, 348 kB)

**Table 8 Status according to CRL**

1. if \( \text{certificate.notBefore} < \text{CRL.thisUpdate} \) and
   \( (\text{CRL.expiredCertsOnCRL} <= \text{certificate.notAfter}) \) and \( (0 < \text{CRL.expiredCertsOnCRL}) \) or
   \( (\text{CRL[thisUpdate} <= \text{certificate.notAfter}) \) and \( (0 = \text{CRL.expiredCertsOnCRL}) \) then
   if certificate is not in CRL then
     VALID
   else
     INCOMPLETE VERIFICATION: waiting for a new CRL
2. if ControlTime <= CRL.thisUpdate then
   VALID
3. else
   if ControlTime < CRL[certificate].revocationDate then
     VALID
   else
     INVALID
4. else
   INCOMPLETE AUTOMATIC VERIFICATION: a request to CA for CRL which can contain the status of the certificate being verified.

Where:
- If `CRL.expiredCertsOnCRL` is not present in the CRL extension, then its value is 0, otherwise the value is defined according to ITU-T X.509 (08/2005).
- `CRL.thisUpdate` is the time before (and including) which the information about the certificate status is always stable and permanent.
- `Certificate.notBefore` is the time since when it is possible to use the certificate and its status can be indicated in CRL.
- `Certificate.notAfter` is the time after which the certificate status in CRL cannot be changed anymore but can be indicated.
- `CRL[certificate].revocationDate` is the date of the certificate revocation being indicated in CRL.
The proof of existence (PoE) can be included in validation also as OCSP with Nonce extension. The OCSP Nonce extension [https://tools.ietf.org/html/rfc6960#section-4.4.1](https://tools.ietf.org/html/rfc6960#section-4.4.1) will be used as a store of the hash value of "time-stamped" object as the PoE, and OCSP response field *producedAt* [https://tools.ietf.org/html/rfc6960#section-2.4](https://tools.ietf.org/html/rfc6960#section-2.4) will provide the time value, equivalent to the time-stamp time value, of existence of the object whose hash value is stored in the OCSP Nonce extension. The indication that the time-stamp is implemented over OCSP protocol and the proof of existence (PoE) is included in the OCSP nonce field can be realised by including a signed and/or unsigned attribute/element in the document signature.

Nonce ::= OCTET STRING

ProofOfExistence ::= SEQUENCE {
    poEType  PoEType DEFAULT poESignerSignatureBinOctets,
    poE MessageImprint
}

PoEType ::= INTEGER { poESignerSignatureBinOctets(0), poEAnyData(1),
    poESignedData(2), poEArchivedSignature(3) }

MessageImprint ::= SEQUENCE { -- MessageImprint is defined in RFC 3161
    hashAlgorithm                AlgorithmIdentifier,
    hashedMessage                OCTET STRING
}

-- Signed or unsigned attribute used as index of OCSP/CRL PoE with data
id-PoECRLorOCSPIndex OBJECT IDENTIFIER ::= { 1.3.6.1.4.1.38655.1.7 }  
-- if ETSI adopt it then OID can be mapped to ETSI OID tree

PoECRLorOCSPIndex ::= SEQUENCE {
    poEHashIndex MessageImprint, -- hash index of OCSP or CRL
    poEDataType PoEType DEFAULT poEArchivedSignature,
    poEData OCTET STRING OPTIONAL
}

Component poEHashIndex contains a hash value as an index of OCSP or CRL used as the proof of existence (PoE).

When poEHashIndex is an index of CRL and it is a signed attribute, then PoECRLorOCSPIndex is the PoE of the signature which was created after the time of CRL creation (thisUpdate of CRL).

When poEDataType contains poESignerSignatureBinOctets or poEArchivedSignature then PoECRLorOCSPIndex is an unsigned attribute. When poEDataType contains poESignedData then PoECRLorOCSPIndex is a signed attribute.
When `poEDataType` contains `poE ArchivedSignature` then `poEData` contains ATSHashIndexV2 defined for CMS signature (see Draft EN 319 122-1) and OCSP response identified by `poEHashIndex` contains in a nonce extension the hash of time-stamped data as are defined for archive time-stamp version 3 in CMS signature standard.

When `poEDataType` contains `poESignerSignatureBinOctets` then OCSP response identified by `poEHashIndex` contains in a nonce extension `ProofOfExistence` with the hash of time-stamped data as are defined for the signature time-stamp in CMS signature standard.

When `poEDataType` contains `poESignedData` then `poEData` may contain ATSHashIndexV2 defined for CMS signature (see Draft EN 319 122-1) and OCSP response identified by `poEHashIndex` contains in a nonce extension `ProofOfExistence` with the hash of time-stamped data as are defined for the content time-stamp in CMS signature standard.

The time-stamp (TS) can be implemented as OCSP with a nonce containing the hash of time-stamped data without any modification of OCSP protocol on a signer or server side. Modified are only signature creation applications, which add `ProofOfExistence` containing a hash of data which will be time-stamped in a Nonce extension of OCSP request. The nonce is defined in RFC 6960 as: "The nonce cryptographically binds a request and a response to prevent replay attacks".

It means we can use it because OCSP guarantees that the value in request and response will be the same.

When `poEType` is `poESignerSignatureBinOctets`, it can be used for any digital signatures which are validated with a public key included in a certificate whose status OCSP response provides together with a hash value of digital signature in the OCSP Nonce extension.

It is a significant improvement and simplification of signature validation because immediately we have the status of a certificate and also the proof of existence of a digital signature in the time value of OCSP `producedAt` field without any modification of already available OCSP services.

An incorrect and absurd concept of revocation freshness in ETSI TS 102 853, for the past validation, causes that some editors of ETSI ESI are not able to understand some requirements of REGULATION (EU) No. 910/2014.

Static Grace Period is not used anymore for validation according to REGULATION (EU) No. 910/2014 Article 24(3): "If a qualified trust service provider issuing qualified certificates decides to revoke a certificate, it shall register such revocation in its certificate database and publish the revocation status of the certificate in a timely manner, and in any event within 24 hours after the receipt of the request. The revocation shall become effective immediately upon its publication".

The period during which a verifier must wait, is a dynamic interval from one second to 24 hours (maximum).

The regulation uses a mechanism based on a certificate database, see point (k) of Article 24(2): "in case of qualified trust service providers issuing qualified certificates, establish and keep updated a certificate database."
According to standards based on X.509 the issuing process of the certificate status e.g. of CRL or OCSP indicates in the field `thisUpdate` the date and time when the certificate database was locked and updated.

The value of `thisUpdate` field, of making (producing) CRL or OCSP response, is a crucial time event because any request of revocation received after the time of lock (the value in `thisUpdate` of making CRL or OCSP response) is stored in the certificate database with the revocation time value after the time value from `thisUpdate` field of making CRL or OCSP response (thisUpdate is also internal value in the certificate database and is only increased according to a real time and is used as a value of the certificate revocation time). It means a backward revocation is not possible and is forbidden.

The time when the revocation becomes effective immediately upon its publication is indicated in the field `thisUpdate` of e.g. CRL or OCSP.

According to such legal and technical conditions the interval which is one second to 24 hours (maximum) is the value of `thisUpdate` of CRL (or OCSP), published after a request of revocation, minus (-) the time when the request of revocation was submitted to CA. CA must be able, according to the regulation, to produce such CRL or OCSP where `thisUpdate` is in 24 hours' time interval (shorter is better).

It means the interval in which the application waits for a final certificate status is from one second to 24 hours maximum (the status of a certificate in question is final and will never be changed when `thisUpdate` is after the control time).

How long the application will wait for the final status, depends on used technology.

The worst case is CRL or when OCSP is based on CRL, where the value in `thisUpdate` field of OCSP is the same as the value in `thisUpdate` of the source CRL. When OCSP is based on the certificate database then `thisUpdate` is limited on the technological limitation of the database engine speed.

The positive OCSP in the `certHash` extension also provides an important assurance for a user that the status of a real certificate is known in the certificate database.

**Proposal of visual indication of electronic signature usage in transition from a printed electronic document to the electronic document signed electronically**

Now we have CMS, PDF and XML signature profiles.

For the easy transition from a printed electronic document, which is signed by hand, to the electronic document which is signed electronically, we can define a new visual convention for indication that the electronic document is/was signed electronically. It can be realised by entering more than one `at-sign`, @.

The `at-sign`, @, can be used as an indication that the electronic document is signed electronically. The at-sign, @, can be included at least in one field of the electronic document, where a handwritten signature is expected in a printed paper document.
Information with a history about the at-sign, @, is on the web page http://en.wikipedia.org/wiki/At_sign

Example:

**Paper document**

Jørgen Friis  
Vice President  
Standards Enabling Services  
ETSI

Name:  
Title:  
Date:  
Signature: 

Vladimir Janský  
Authorized to act as Director of SIBEP

Name:  
Title:  
Date:  
Signature: 

**Electronic document**

The at-sign, @, can be used as an indication that the electronic document is signed electronically. The at-sign, @, can be included at least in one field of the electronic document, where a handwritten signature is expected in a paper document.

Jørgen Friis  
Chief Services Officer

Name:  
Title:  
Date:  
Signature: 

Mario Italy  
director

Name:  
Title:  
Date:  
Signature: 

Tuesday, 11 November 2014

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