NATIONAL SECURITY AUTHORITY

Version 3.0

Formats of certificate revocation list and confirming the status and validity of certificates

17 January 2010

This English version of the Slovak document No. 535/2009/IBEP/OEP-001 is for reference purposes only. In case of conflict between the English translation and the original Slovak version, the Slovak version shall prevail and supersedes the English translation as the original version. Therefore, only the NSA Deliverables published by NSA in their original language shall be used for evaluation of products and technical judgement.

NATIONAL SECURITY AUTHORITY

Information Security and Electronic Signature Department

Budatinska 30, 850 07 Bratislava 57

http://www.nbusr.sk/

e-mail: info@nbusr.sk

Content

1 Introduction

In the Qualified Electronic Signatures verification (hereinafter referred to as QES) the basic assumption is to verify correctly the validity of the qualified certificate and the certificates for the maintenance of QES (hereinafter referred to as maintenance certificates, which are used for the qualified electronic signature verification). In order to verify the certificate status unambiguously, it is necessary to define unambiguous rules for the content of data included in CRL and OCSP. Pursuant to the Act No.215/2002 Coll. on Electronic Signature and on the amendment and supplementing of certain acts as amended, particularly pursuant to the Act No.214/2008 Coll.:

- Certification Authority and Accredited Certification Authority (hereinafter referred to as ACA) pursuant to Article 14 (1) letter i), point 4 publish the certificate revocation list (CRL); pursuant to Article 15 (4) the certification authority supplies the information about the certificate status by providing CRL according to Article 8 containing all certificates whose validity was earlier revoked while the certificate whose validity was earlier revoked must be at least once included in the list pursuant to Article 8 and Article 15 (5). If ACA is able to set up the technical conditions, the certification authority provides the information about the certificate status also in the form of confirmation of the qualified certificate existence and validity (positive OCSP).
- The responsibility of ACA in providing the accredited certification services pursuant to Article 14 (3) letter e) is to send out lists of issued qualified certificates and CRL to the National Security Authority (hereinafter referred to as the NSA); the format, method and periodicity of sending out these lists are stipulated by the regulation issued by the NSA.
- The NSA pursuant to Article 10 letter m) maintains the list of all issued qualified certificates together with the information about their validity being sent according to Article 14 (3) letter e) and provides that information.

2 Scope

The NSA issues the present standard pursuant to Article 3 (1), Article 4 (2) and Article 8 (7) of the Decree No.131/2009 Coll. on the format, content and administration of certificates and qualified certificates and the format, periodicity and method of issuing a list of revoked qualified certificates (on certificates and qualified certificates) and pursuant to Article 10 (2) letter j) of the Act No.215/2002 Coll. The present standard is issued for purposes of ensuring the unambiguous determination of the qualified certificate and maintenance certificate status; also for purposes of processing the data being sent to the NSA and the unified method of providing the information from obtained data to the public. The document technically specifies required properties and sets of used protocols in order to create the unified environment for providing the required services to the public.

The need of long-term verification of the qualified electronic signature validity is solved by the Act No.214/2008 Coll. by the amendatory act on ES which requires that ACA sends out issued qualified certificates and maintenance certificates to the NSA and the NSA is required to provide the information about the validity of these certificates for a long time. The present document specifies the ACA method of sending out these data to the NSA and the NSA method of providing this information to the public.

In addition to certificates issued by the NSA, the NSA also provides the information about the validity of expired qualified certificates and expired maintenance certificates issued by CA being accredited by the NSA in the form of positive OCSP response with three possible state indicators:

1. **valid**;

2. **invalid** from the time and date and possibly with the reason of revocation, if this reason was provided by ACA to the NSA;

3. **unknown**, if the certificate issued by ACA is not expired and ACA has not provided information about the certificate revocation to the NSA.

3 References

References to documents defining used types and procedures.

[1] ETSI TS 101 733 Electronic Signature Formats (CAdES)

[2] ETSI TR 102 272 ASN.1 format for signature policies

[3] RFC 5280 X.509 PKI Certificate and Certificate Revocation List 5-2008

[4] RFC 3739 Qualified Certificates Profile 3-2004

[5] ETSI TS 101 862 Qualified Certificate Profile

[6] RFC 5652 Cryptographic Message Syntax 9-2009

[7] RFC 3161 Time-Stamp Protocol (TSP) 8-2001

[8] RFC 2560 X.509 PKI Online Certificate Status Protocol 8-1999

[9] NSA Qualified Electronic Signature Formats

[10] ITU-T RECOMMENDATION X.509 (08/2005) | ISO/IEC 9594-8:2005

[11] ETSI TS 102 280 X.509 V.3 Cert. Profile for Cert. Issued to Natural Persons

[12] Common PKI COMMON PKI SPECIFICATIONS FOR INTEROPERABLE APPLICATIONS FROM T7 & TELETRUST

[13] ETSI TS 101 456 Policy requirements for cert. authorities issuing qualified cert.

[14] ETSI TS 102 042 Policy requirements for cert. authorities issuing public key cert.

[15] ETSI TS 102 231 Provision of harmonized Trust-service status information

[16] RFC 2560 X.509 PKI Online Certificate Status Protocol 6-1999

[17] ISO/IEC 7064 Data processing - Check character systems 2003

[18] RFC 3548 The Base16, Base32, and Base64 Data Encodings 7-2003

[19] ISO/IEC 3166 Codes for the representation of countries

[20] RFC 3647 Internet X.509 PKI Certificate Policy and Certification Practices Framework

[21] RFC 2822 Internet Message Format 4-2001

[22] RFC 2046 MIME Part Two- Media Types 11-1996

[23] RFC 3629 UTF-8, a transformation format of ISO 10646 11- 2003

[24] RFC 2585 Operational Protocols: FTP and HTTP 5- 1999

[25] NSA Decree No.131/2009 Coll. on the format, content and administration of certificates and qualified certificates and the format, periodicity and method of issuing a list of revoked qualified certificates (on certificates and qualified certificates)

[26] NSA Decree No.136/2009 Coll. on the method and procedure of using the electronic signature in business and administrative relations

[27] NSA Formats of certificates and qualified certificates

4 Abbreviations

ACA	Accredited Certification Authority
AdES	Advanced Electronic Signature
ASCII	American Standard Code for Information Interchange
ASN.1	Abstract Syntax Notation 1
BASE64	Type of content coding in MIME
CA	Certification Authority
CBC	Cipher-block chaining
CMS	Cryptographic Message Syntax
CAdES	CMS Advanced Electronic Signature
CRL	Certificate Revocation List
DER	Distinguished Encoding Rules (for ASN.1)
ESS	Enhanced Security Services (enhances CMS)
HTML	Hypertext Markup Language
HTTP	Hyper Text Transfer Protocol
ISO	International Organization for Standardization
Common PKI	COMMON PKI SPECIFICATIONS FOR INTEROPERABLE APPLICATIONS
MIME	Multipurpose Internet Mail Extensions
Maintenance certificates	It is a set of certificates which are used for verification of AdES based on qualified certificate e.g. ACA, a timestamp certificate. Maintenance certificates are issued according to rules defined in the national legislation which ensures for the verifier the equivalent level of assurance as is defined for the qualified certificate.
NSA	National Security Authority of the Slovak Republic
OCSP	Online Certificate Status Protocol
OID	Object Identifier
PKIX	Internet X.509 Public Key Infrastructure
SMTP	Simple Mail Transfer Protocol
TSP	Time Stamp Protocol
URL	Uniform Resource Locator
UTF-8	Transformation format of ISO 10646
Act on ES	Act No.215/2002 Coll. on Electronic Signature and on the amendment and supplementing of certain acts as amended
QES	Qualified Electronic Signature

5 CRL Format

Certification authorities accredited in Slovakia must pursuant to the Act on ES issue CRL in which they give the status of issued qualified certificates and maintenance certificates. CRL must contain at least items whose content is specified by this profile.

Table 1 Basic format of the certificate revocation list - CRL

	Record in ASN.1	**Short description**
1.	`CertificateList ::= SEQUENCE {`	
2.	`tbsCertList TBSCertList,`	DER coded data, signed by CA.

3.	`signatureAlgorithm`
	`AlgorithmIdentifier,`
		An identifier of the signing algorithm and its parameters, if the signing algorithm requires parameters. The algorithm is used by the certification authority for signing *tbsCertList*.

4.	`signatureValue BIT STRING`
	`}`
		CRL signature.

Table 2 TBSCertList - signed block

	Record in ASN.1	**Short description**
1.	`TBSCertList ::= SEQUENCE {`	

2.	`version Version OPTIONAL,`
	`-- if present, MUST be v2`
		CRL version must be v2 (value 1), because CRL for qualified certificates must contain *crlExtensions* and may contain *crlEntryExtensions*.

3.	`signature AlgorithmIdentifier,`	Exactly the same content as in Table 1, line 3.
4.	`issuer Name,`	A name of CRL (CA) issuer. Requirements for the name *issuer* are the same as defined in the NSA document "Formats of certificates and qualified certificates", Table 2, line 5.
5.	`thisUpdate Time,`	Date and time when CRL was issued. The format is the same as defined in the NSA document "Formats of certificates and qualified certificates" Table 2, line 6.
6.	`nextUpdate Time OPTIONAL,`	Date and time of subsequent CRL. Each CRL **must contain** this item. CRL can be issued even before the time in *nextUpdate*, but it cannot be issued after that time. CA issues CRL with the time value in *nextUpdate* which is the same or later than the time in the latest issued CRL. The format is the same as in Table 2, line 5.
7.	`revokedCertificates SEQUENCE OF SEQUENCE {`	The certificate revocation list. This item is missed out, if any certificate is not revoked.

8.	`userCertificate`
	`CertificateSerialNumber,`
		A serial number of the revoked certificate.

9.	`revocationDate Time,`	Date and time of the certificate revocation. The format is the same as in Table 2, line 5.

10.	`crlEntryExtensions Extensions OPTIONAL --if present, MUST be v2`
	`} OPTIONAL,`
		A non-empty list of extensions which more exactly specify the certificate revocation.

11.	`crlExtensions [0] EXPLICIT`
	`Extensions OPTIONAL`
	`-- if present, MUST be v2 }`
		A non-empty list of CRL extensions.

Table 3 CRL types according to CA which revokes certificates

	Type	Short description
1.	"*direct*" CRL	In direct CRL the item *DName* of the CRL issuer is identical with the item *DName* of the issuer of revoked certificates.
2.	"*indirect*" CRL	In indirect CRL *DName* of the CRL issuer is different than *DName* of the issuer of revoked certificates. This CRL contains the extension *issuingDistributionPoint* with the item *indirectCRL* = TRUE. The certificate can refer to indirect CRL if it contains the certificate extension *CRLDistributionPoints* with the item *distributionPoint* containing the item *cRLIssuer.* The item *cRLIssuer* contains *DName* which is different from the name *subject* in CA certificate which belongs to the certificate of the issuer.

Certificates in the sub-tree of the Root NSA certificate must contain the first certificate extension *CRLDistributionPoints* to direct CRL.
Issuing of indirect CRL is forbidden because of ambiguous certification path creation for verification of CRL signature and because of possible frauds (when the false CA can change the status of the certificate validity being issued by other CA) with the exception:

a. Only indirect CRL issued by the NSA or indirect CRL verified by the certificate being issued by CA which issued certificates whose validity is verified by indirect CRL, are permitted.
b. The same problem occurs with the OCSP response (RFC 2560) which is not usually verified directly by CA certificate used for verification of issued certificates which are verified by OCSP (OCSP is verified by the certificate issued by CA and thus OCSP is not issued directly by CA) and therefore the rules which apply to the issuance of indirect CRL defined in the previous point 1 equally apply to the issuance of the OCSP response.

Table 4 crlExtensions

	CRL extensions	OID and necessity of the item in CRL	Short description	Critical

1.	`IssuerAltNames`	`{2 5 29 18}`
		`Only if contained in CA certificate.`
			Alternative (technical) name of the certificate issuer: e.g. OtherName, e-mail, DNS name, IP address, URI, etc.	SHOULD NOT

2.	`CRLNumber`	`{2 5 29 20}`
		`Must be in CRL.`
			Positive sequential number of CRL that is incremented by one during the issuance. Max size 20 BYTE. $1 \le$ CRLNumber $\le 2^{159}$	MUST NOT

3.	`DeltaCRLIndicator`	`{2 5 29 27}`
		`Not recommended.`
			It indicates that it is delta-CRL. Delta-CRL issuing is not recommended.	MUST

4.	`IssuingDistributionPoint`	`{2 5 29 28}`
		`Must be in CRL.`
			It determines the group of revoked certificates included in CRL (User/CA), the reason, if it is indirect CRL or how and where from it is possible to obtain CRL. It must contain HTTP address. It can contain LDAP but in this case it must contain the internet address of LDAP server too. In order to	MUST

			verify the validity of qualified certificates and maintenance certificates, CA must issue one complete CRL (non-segmented) containing all revoked user certificates and CA certificates issued by CA.	
5.	AuthorityKeyIdentifier	{2 5 29 35} Must be in CRL.	Public key identifier *keyIdentifier* of CA that issued CRL. It is recommended to fill out *authorityCertSerialNumber* too.	MUST NOT

Table 5 crlEntryExtensions – extensions of revoked certificates in CRL

	Extensions of certificate list in CRL	OID and necessity of the item in CRL	Short description	Critical
1.	ReasonCode	{2 5 29 21} Only if the reason is known.	The reason of the certificate revocation.	MUST NOT
2.	HoldInstructionCode	{2 5 29 23} Must not be used.	The identifier which indicates why the certificate was in the status *Hold.* (Only with delta CRL. Delta is not permitted).	MUST NOT
3.	InvalidityDate	{2 5 29 24} May be present.	It indicates the time when the private key was suspected of compromising.	MUST NOT
4.	CertificateIssuer	{2 5 29 29} Indirect CRL must contain. Direct CRL must not contain.	It is used only in indirect CRL to determine the issuer name of the revoked certificate if the issuer name is different than the CRL issuer name. The value *GeneralNames* must contain only one *directoryName* from the subject DName of the issuer certificate of the revoked certificate. If the first item *crlEntryExtensions* does not contain *Certificatelssuer,* then the CRL issuer becomes the issuer of the revoked certificate until the item with the first *CertificateIssuer* occurs in sequence. Then the following items will have the previous issuer until the next item with *Certificatelssuer* occurs in sequence. Thus the list is sorted according to *Certificatelssuer* and *Certificatelssuer* of certificates issued by the same issuer is found only in the first record of the revoked certificates.	MUST

6 OCSP Format

Certification authorities accredited in the Slovak Republic pursuant to Act on Electronic Signature may issue the OCSP response in which the status of issued qualified certificates and maintenance certificates is determined. The OCSP response must contain at least items which are listed in this profile and whose content is specified by this profile.

6.1 A format of a request for obtaining the certificate status

Table 6 OCSP request

	Record in ASN.1	**Requirements extending RFC 2560**
1.	`OCSPRequest ::= SEQUENCE {`	
2.	`tbsRequest TBSRequest,`	

3.	`optionalSignature [0] EXPLICIT`
	`Signature OPTIONAL }`
		The item `optionalSignature` must not be required for obtaining the OCSP response.

4.	`TBSRequest ::= SEQUENCE {`	
5.	`Version [0] EXPLICIT Version DEFAULT v1,`	

6.	`requestorName [1] EXPLICIT GeneralName`
	`OPTIONAL,`
		The item `requestorName` must not be required for obtaining the OCSP response.

7.	`requestList SEQUENCE OF Request,`	

8.	`requestExtensions [2] EXPLICIT Extensions`
	`OPTIONAL }`	

9.	`Signature ::= SEQUENCE {`
	`signatureAlgorithm AlgorithmIdentifier,`
	`signature BIT STRING,`
	`certs [0] EXPLICIT SEQUENCE OF Certificate`
	`OPTIONAL}`	

10.	`Version  ::= INTEGER  {  v1(0) }`	
11.	`Request  ::= SEQUENCE {`	
12.	`reqCert CertID,`	

13.	`singleRequestExtensions [0] EXPLICIT`
	`Extensions OPTIONAL }`	

14.	`CertID ::= SEQUENCE {`	
15.	`hashAlgorithm AlgorithmIdentifier,`	The algorithm must be only from the set of algorithms which are considered to be secure in a particular period. This set of algorithms is published by decree and is automatically verifiable by signature policy being published by the NSA which was valid in the period when the request was sent.

16.	`issuerNameHash OCTET STRING,`
	`-- Hash of  Issuer's DN`	

17.	`issuerKeyHash  OCTET STRING,`
	`-- Hash of  Issuers public key`	

18.	`serialNumber   CertificateSerialNumber}`	

In communication with the NSA the OCSP request must be in DER coding and sent only through HTTP or SMTP protocols at the address which is published by the NSA on its web page. The MIME format and type for the OCSP request are defined in Annex C, Table 10 and line 6.

6.2 A format of the OCSP response

Table 7 OCSP response

	Record in ASN.1	Requirements extending RFC 2560

1.	`BasicOCSPResponse ::= SEQUENCE {`
	`tbsResponseData ResponseData,`
	`signatureAlgorithm AlgorithmIdentifier,`
	`signature BIT STRING,`	

2.	`certs [0] EXPLICIT SEQUENCE OF`
	`Certificate OPTIONAL }`
		A response must contain the certificate which is used for the OCSP verification.

3.	`ResponseData ::= SEQUENCE {`	
4.	`version [0] EXPLICIT Version DEFAULT v1,`	
5.	`responderID ResponderID,`	
6.	`producedAt GeneralizedTime,`	Time of signing the OCSP response.
7.	`responses SEQUENCE OF SingleResponse,`	

8.	`responseExtensions [1] EXPLICIT`
	`Extensions OPTIONAL }`	

9.	`ResponderID ::= CHOICE {`
	`byName [1] Name,`
	`byKey [2] KeyHash }`
		The item `byName` must be present in the OCSP response.

10.	`SingleResponse ::= SEQUENCE {`	
11.	`certID CertID,`	
12.	`certStatus CertStatus,`	
13.	`thisUpdate GeneralizedTime,`	If the response is about the expired certificate, then the item must contain time and date of the certificate expiration or a later value but it cannot be later than `thisUpdate` from the latest CRL which could have contained the serial number of the certificate being verified. If the latest CRL which can contain the certificate status has a value in `thisUpdate` which is earlier than is the expiration time of the certificate being verified, in `OCSP-SingleResponse-thisUpdate` the value must be `CRL-thisUpdate.` OCSP provided by the NSA must return the status of the certificate issued by the accredited CA in `certStatus` **`unknown`**, if the certificate has not expired yet.

14.	`nextUpdate [0]EXPLICIT GeneralizedTime`
	`OPTIONAL,`
		This item is not present when the certificate is expired.

15.	`singleExtensions [1]EXPLICIT Extensions`
	`OPTIONAL }`
		It must contain [12] `CertHash` from the certificate whose status is provided.

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

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

18.	`UnknownInfo ::= NULL`
	`--this can be replaced with an enumeration`	

CertHash (positive statement) – an extension defined for the type *SingleResponse* in *singleExtensions*:

```
Definition of the extension CertHash adopted from Common PKI [12] Optional
SigG-Profile:

Common PKI Object Identifiers
      id-commonpki OBJECT IDENTIFIER ::= {1 3 36 8 }
      id-commonpki-at OBJECT IDENTIFIER ::= {id-commonpki 3}
      id-commonpki-at-certHash OBJECT IDENTIFIER ::= {id-commonpki-at 13}

Common PKI PRIVATE EXTENSIONS
CertHash ::= SEQUENCE {
      hashAlgorithm AlgorithmIdentifier,
            -- The identifier of the algorithm that has been used the
            -- hash value below.
      certificateHash OCTET STRING
            -- The hash over DER-encoding of the entire PKC
      }     -- or AC (i.e. NOT a hash over tbsCertificate).
```

The OCSP response must be in the format *BasicOCSPResponse* in DER coding and each *SingleResponse* in *singleExtensions* must contain the extension *CertHash* (positive statement).

7 A format and method of data provision from ACA to the NSA

7.1 A format of provided qualified certificates and maintenance certificates

ACA issues qualified certificates in the format X.509 [11] and to the public provides the information about the validity in the CRL form [11] and optionally in the OCSP form [8]. ACA pursuant to Act No. 215/2002 Coll. must send the lists of issued and revoked qualified certificates (Article 14 (3) letter e)) to the NSA while the NSA stipulates the format, method and periodicity of their sending. Received qualified certificates must not be published by the NSA; just the information about their validity is allowed to be provided. ACA sends issued qualified certificates in the format *Certificate* in DER coding to the NSA.

A format according to X.509:

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

ACA shall send encoded lists of issued qualified certificates within 8 days after the issuance of the certificate to the e-mail address of the NSA which is released for this purpose by the NSA on the web page of the NSA. Prior to sending ACA shall store issued qualified certificates in DER coding according to ASN.1 for X.509 in the attachment of the e-mail according to MIME [21] in BASE64 coding [18] and in such way ACA shall create `multipart/mixed` signed S/MIME message. This message shall be encrypted by public key from the NSA certificate which is published on the web page of the NSA for the purpose of e-mail encryption containing qualified certificates and maintenance certificates. Encrypted message shall be in the CMS format [6] and algorithms AES 256bit in CBC mode shall be used for encryption. Encrypted message shall be encoded to MIME electronic mail and shall be sent to e-mail address of the NSA.

After receiving the electronic mail the NSA shall decrypt this message and verify gradually the integrity of received qualified certificates and maintenance certificates which are issued by the NSA to ACA. The certificate validity shall not be verified by the NSA.
According to integrity verification and content of sent qualified certificates and maintenance certificates the NSA shall store to its database only qualified certificates and maintenance certificates which are issued by NSA accredited certification authority in compliance with the NSA standard on formats of certificates and qualified certificates [27]. Other types of certificates shall not be stored in the database.
According to signature verification and content of delivered certificates the NSA shall create the **integrity signature** which shall contain all received certificates and in the item NOTICE the status indicators shall be the following:

- OK – verified, inserted into the NSA database;
- NO – not verified, not inserted into the NSA database – the reason, e.g. incorrect format, not permitted algorithm, not permitted short key length, not issued by accredited CA;
- DP – duplicate, already inserted into the NSA database.

The NSA sends the integrity signature by e-mail to ACA as the receipt of certificate acceptance. If ACA does not receive the receipt of acceptance within 5 days after the list was sent, it shall repeat to send the certificates and if again it does not obtain the receipt of acceptance by e-mail from the NSA, it shall contact the NSA through the NSA registry.

7.2 A format of providing the change of the certificate validity

ACA sends the lists of revoked qualified certificates and maintenance certificates to the electronic mail address of the NSA, which is published for that purpose on the web page of the NSA. If ACA revokes issued qualified certificates or issued maintenance certificates, then it is obliged to send CRL containing this revocation to the NSA within 8 days after the change was published by using the e-mail attachment. The message of electronic mail contains DER coded CRL according to ASN.1 being defined in X.509 in the attachment according to MIME [21] in BASE64 coding [18] and in such way it creates `multipart/mixed` signed S/MIME message.

The NSA sends to ACA by means of electronic mail the integrity signature from received CRLs as a receipt of CRLs acceptance which shall contain all received CRLs and in the item NOTICE the status indicators shall be the following:

- OK – verified, inserted into the database;
- NO – not verified, not inserted into the NSA database – the reason, e.g. incorrect format, not permitted algorithm, not permitted short key length, not issued by accredited CA;
- DP – duplicate, already inserted into the NSA database.

If ACA does not receive the receipt of acceptance within 5 days after the list was sent, it shall repeat to send the CRL and if again it does not obtain the receipt of acceptance by e-mail from the NSA, it shall contact the NSA through the NSA registry.

A format according to X.509:

```
CertificateList  ::=  SEQUENCE  {
     tbsCertList          TBSCertList,
     signatureAlgorithm   AlgorithmIdentifier,
     signatureValue       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, version MUST be v2
                               }  OPTIONAL,
     crlExtensions           [0]  EXPLICIT Extensions OPTIONAL
                                   -- if present, version MUST be v2
}
```

8 A format and method of providing the information on the certificate status by the NSA to the public through OCSP

The NSA provides the information only about the validity of expired certificates which were issued by CA being accredited by the NSA, i.e. qualified certificates and maintenance certificates in the format X.509 [10]. The NSA provides the information in the format of indirect OCSP [8] containing the extension Common PKI CertHash (Positive Statement) and also indirect CRL is possible to be used [10].

The NSA stores the following information for provision of expired certificate validity:

- Certificates and CRL according to X.509 [10] protected by chained integrity signatures with the time stamp of the signature;
- Time and date of the certificate revocation in the format GeneralizedTime and CRL which contains the information about the certificate revocation;
- The reason of the certificate revocation in the type CRLReason.

```
CRLReason ::= ENUMERATED {
        unspecified             (0),
        keyCompromise           (1),
        cACompromise            (2),
        affiliationChanged      (3),
        superseded              (4),
        cessationOfOperation    (5),
        certificateHold         (6),   -- value 7 is not used
        removeFromCRL           (8),
        privilegeWithdrawn      (9),
        aACompromise           (10) }
```

According to this data the NSA provides to verifier the information whether expired qualified certificates or maintenance certificates were (were not) revoked during their validity period. The request for providing the information about the certificate validity can be sent to e-mail address or URL address being published on the web page of the NSA for that purpose.

The profile of the OCSP request and response is defined in part 6. The example of MIME coding of the OCSP request and response is defined in Annex C. The request format is defined in RFC 2560 in part 4, article 1 and the response is defined in part 4, article 2 where the response extension *SingleResponse* must contain Common PKI positive statement certHash. The positive statement *certHash* extends the OCSP response according to RFC 2560 on certificate status to the hash value from the certificate whose status is contained in the response, thus the verifier is sure that OCSP knows the certificate and its status too. The status of expired certificate validity can be verified by using the OCSP response whose signature is verified by certification path. The first certificate of the certification path is verified by currently valid trusted root certificate and all algorithms used are considered to be secure according to currently valid signature policy. Thus the OCSP response is verified to currently valid root certificate and the OCSP response provides the status of already expired certificates of the whole certification path. The extension *certHash* ensures the positive response that the system which issues the OCSP response knows the certificate whose status it returns (was issued and is in the database).

The OCSP request and the OCSP response are provided in the form of e-mail attachment and also in the form defined in the Annex A.1 OCSP over HTTP (OCSP through HTTP protocol) of RFC 2560. In the e-mail message the response is provided in the type *BasicOCSPResponse* according to part 4, article 2.1 of RFC 2560.

Annex A (normative) Final status of the certificate validity

ITU-T X.509 (08/2005) | ISO/IEC 9594-8:2005, IETF RFC 5280 and IETF RFC 2560 define formats of certificates, CRL and OCSP and describe the rules for the verification of the certificate validity in the time of the certificate usage what is usable in systems which do not allow waiting and thus obtaining the real status of the certificate validity. These systems accept a potential risk that in the time when its corresponding private key was used, the certificate was already revoked but the information about the revocation was not available in the time of the certificate and private key usage. In verification of the qualified electronic signature validity which is realized to the time in the past when the signature was created, the risk with doubtful status of the certificate validity is not acceptable. For that reason it is required to use the condition ensuring the unambiguous and permanent status of the certificate validity verification in verification of qualified certificates and maintenance certificates.

A.1 Determination of the status of the certificate validity by CRL

A condition for the status verification of the certificate validity by CRL returns the status to the *ControlTime*. In case the validity of expired certificates is verified, CRL must contain the extension *expiredCertsOnCRL* with the value which is smaller than the value in the certificate being verified certificate.*notAfter*.

Table 8 Status according to CRL

```
1.   if ( certificate.notBefore < CRL.thisUpdate ) and
       ( ((CRL.expiredCertsOnCRL <= certificate.notAfter ) and ( 0 < CRL.expiredCertsOnCRL )) or
        (( CRL.thisUpdate <= certificate.notAfter ) and ( 0 = CRL.expiredCertsOnCRL )))  then
2.           if certificate is not in CRL then
3.                   if (ControlTime + cautionPeriod )  <=  CRL.thisUpdate then
                             VALID
4.                   else
                             INCOMPLETE VERIFICATION: waiting for a new CRL
5.           else
                     if ControlTime <  CRL[certificate].revocationDate then
                             VALID
6.                   else
                             INVALID
7.   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 already 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.

Explanation of the conditions:

1) CRL is issued in the time when it can contain the information about the certificate status.

2) The serial number of the certificate is not in CRL, thus the certificate is not revoked.

3) CRL was issued after the key usage in the past, when CRL must already contain the information about the revocation if occurred.

4) CRL is not issued in the time after the key usage, when CRL already must contain the stable information about the certificate status.

5) The certificate was revoked after the time of the key usage and thus the certificate is valid.

6) The certificate was revoked before and including the time of the key usage, therefore the certificate is invalid.

7) It is necessary to use such CRL which is issued in the time when CRL can contain the information about the revocation: in the time from the period of the certificate usage or if CRL contains expired certificates too, then from the period in which CRL may contain the status of the certificate being verified.

A.2 Determination of the status of the certificate validity by OCSP

A condition for the status verification of the certificate validity by OCSP response returns the status to the time ControlTime. The OCSP response must be issued:
- in time interval within which the certificate is usable <certificate.*notBefore*, certificate.*notAfter>;*
- or in the time in which the OCSP response may contain the information about the expired certificate validity which is indicated by a value *ArchiveCutoff* with the value smaller than the expiration time of the certificate being verified certificate.*notAfter;*
- or the OCSP response for the expired certificate contains the positive statement *CertHash* in the extension of the OCSP response with the hash value of the certificate being verified what confirms not only the knowledge about the status but also the certificate integrity with the currently secure hash algorithm.

Table 9 Status according to OCSP response

```
1.  if ( certificate.notBefore < OCSP[certificate].thisUpdate ) and
       ((( OCSP.ArchiveCutoff <= certificate.notAfter ) and ( 0 < OCSP.ArchiveCutoff )) or
        (( OCSP[certificate].thisUpdate <= certificate.notAfter) and (0 = OCSP.ArchiveCutoff )) or
        (OCSP[certificate].CertHash = certificate.CertHash) ) then
2.          if OCSP[certificate].CertStatus = good then
3.                  if (ControlTime + cautionPeriod ) <= OCSP[certificate].thisUpdate then
                            VALID
4.                  else
                            INCOMPLETE VERIFICATION: It is necessary to obtain a newer OCSP response
5.          else
                    if OCSP[certificate].CertStatus = revoked then
                            if Control Time < OCSP[certificate].revocationTime then
                                    VALID
6.                          else
                                    INVALID
7.                  else
                            INCOMPLETE AUTOMATIC VERIFICATION: OCSP does not know
                            the current status of the certificate validity because
                            OCSP[certificate].CertStatus = unknown
                            Verification is possible by other OCSP or CRL.
8.  else
            INCOMPLETE AUTOMATIC VERIFICATION: a request to CA for OCSP response or CRL which can
            contain the status of the certificate being verified.
```

Where:

- OCSP.*ArchiveCutoff* - if *ArchiveCutoff* is not present in the OCSP response, then its value is 0, otherwise the value stored in *ArchiveCutoff* is defined according to RFC 2560.
- OCSP[certificate].*CertHash* is the hash value of the certificate whose status is returned by the OCSP response (Common PKI private extensions). If this extension is found in the OCSP response, then the certificate status is known for OCSP and the hash value ensures the integrity by currently secure hash algorithm.
- Certificate.*CertHash* is the hash value of the certificate whose status is verified.

- OCSP.*producedAt* is the time of the OCSP response issuance.
- *OCSP*[certificate].*thisUpdate* is the time before and including which the information about the certificate status is already stable and permanent. The value must be smaller or equal to OCSP.*producedAt.*
- *OCSP*[certificate].*nextUpdate* is the auxiliary time about the availability of the latest occurrence of the information about the status. The OCSP response must not contain the item *nextUpdate* if the certificate, whose status is returned, is expired.
- Certificate.*notBefore* is the time since when it is possible to use the certificate and its status can be indicated in OCSP (CRL).
- Certificate.*notAfter* is the time after which the certificate status in CRL (OCSP) cannot be changed but can be indicated.
- OCSP[certificate].*revocationTime* is the time of the certificate revocation.
- OCSP[certificate].*CertStatus* is the status of the certificate being verified with the values: *good*, *revoked* and *unknown.*

Explanation of the conditions:

1) OCSP is issued in the time when it can contain the information about the certificate status.
2) The serial number of the certificate is not in OCSP (CRL), thus the certificate is not revoked.
3) OCSP was issued after the key usage in the past, when OCSP must already contain the information about the revocation if occurred.
4) OCSP is not issued in the time after the key usage when OCSP must already contain the stable information about the certificate status.
5) The certificate was revoked after the time of the key usage and thus the certificate is valid.
6) The certificate was revoked before and including the time of the key usage, therefore the certificate is invalid.
7) OCSP is not able to determine the status of the certificate validity; therefore it is necessary to obtain other OCSP or CRL.
8) It is necessary to use such OCSP (CRL) which is issued in the time when OCSP (CRL) can contain the information about the revocation: in the time from the period of the certificate usage or if OCSP (CRL) contains expired certificates too, then from the period in which OCSP (CRL) may contain the status of the certificate being verified.

A.3 Determination of the certificate accreditation/supervision status by TSL being issued according to Decision of EU Commission (2009/767/EC)

According to Commission Decision of 16 October 2009 which is setting out measures facilitating the use of procedures by electronic means through the ‘points of single contact’ under Directive 2006/123/EC of the European Parliament and of the Council on services in the internal market [notified under document C(2009) 7806] (Text with EEA relevance) (2009/767/EC) and on Corrigendum to Commission Decision 2009/767/EC of 16 October 2009 which is setting out measures facilitating the use of procedures by electronic means through the ‘points of single contact’ under Directive 2006/123/EC of the European Parliament and of the Council on services in the internal market:
http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2009:299:0018:0054:EN:PDF
the Authority issues TSL in the form of signed PDF and optionally signed TSL in the XML format.

The NSA publishes the certificate for TSL signing in the XML and PDF format in the Trusted List (of root certificates and signature policies) on its web page http://www.nbusr.sk/en/electronic-signature/signature-policies/index.html.

In verification of Slovak qualified certificates and Slovak maintenance certificates the foreign citizen-verifier uses the data stored in the Trusted List on the web page of the NSA while the trust is determined by trust in the Root NSA certificate. The NSA certificate was published not only on the web page of the Authority but also in the press (Verejná správa, edition number 25-26/2009 and Hospodárske noviny of 19 November 2009) what is also published on the web page of the NSA in order to ensure the trusted way of the root certificate verification other than the electronic way. If the verifier cannot find out which root certificate is trusted, he follows the rules being defined in his/her country for TSL issued in his/her country which is issued according to EU Decision 2009/767/EC.

In verification of foreign qualified certificates or maintenance certificates the verifier in the Slovak Republic begins with the verification of the trust in the Root NSA certificate; subsequently the verifier obtains the certificate for the signature verification of the Slovak TSL from the NSA Trusted List. The Slovak TSL is published on the web page of the NSA and contains the URL reference to EU TSL. EU TSL contains the list of URL references to national TSL of Member States and information on procedures of publishing the trusted Root e.g. in the national official journal. The URL reference to TSL of other EU Member State may point at TSL in the PDF or XML format or TSL in signed form or the URL reference may point at TSL through the secure channel SSL/TLS and URL may also contain the certificate which is used by verifier to verify the signature of the national TSL or to verify the creation of the secure channel SSL/TLS in order to obtain the national TSL.

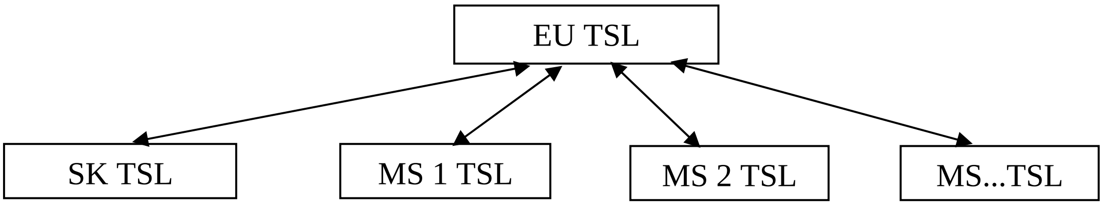
The relying party will use the national TSL to find out whether the certificate is qualified if it does not contain these data directly and whether the certificate was issued by an accredited certification service provider or supervised provider in the time of issuing the certificate being verified (to the time *thisUpdate*).

The steps which the relying party can do during the verification according to TSL are the following:

- According to data present in the electronic signature the verifier verifies the validity of the signer's certificate and maintenance certificates in compliance with the rules profiled by procedures of a particular EU Member State for the verification of X.509 certificate.
- The result of the verification is the intersection of information about the validity or revocation of the signer's certificate, of the certification path, of certificates for time stamp verification and of other maintenance certificates.
- According to the issuer certificate of the qualified certificate or the end user maintenance certificate the verifier will choose the Slovak TSL or a reference from the Slovak TSL to EU TSL through which it is possible to obtain the reference to TSL of other Member State where TSL contains the issuer certificate in the item *Service Digital Identity* or the issuer certificate is included in the sub tree of the certification service provider if the service provider has X.509 certificate in its internal hierarchy.
- According to the issuing time of the end user certificate (*thisUpdate*) the verifier verifies whether the issuer was accredited or supervised in issuing time and according to extensions present in TSL the verifier verifies whether it is the qualified certificate or the qualified certificate issued for SSCD if such information is not included directly in the end user certificate.
- If TSL in currently valid services does not contain the required certification service provider according to the issuing time of the signer's certificate (*thisUpdate*), the verifier

will look up through the TSL history and if even there he cannot find the required provider who was accredited or supervised to the time *thisUpdate*, he will proclaim the certificate as the certificate that does not meet the requirements for qualified certificates or maintenance certificates being issued pursuant to Directive on Electronic Signature, otherwise the certificate is considered to be the qualified certificate or the maintenance certificate.

- Then the following steps of the verification examine whether in the signature time, for example to the time from the signature time stamp or secure audit trail, the status of the certificate validity was provided by certification service providers who were accredited or supervised to the time *thisUpdate* of CRL or OCSP according to annex A.1 or A.2.
- The verifier proceeds in the same way in verification of time stamp certificates or certificates for signing of OCSP or CRL, thus in verification of maintenance certificates if they were issued by certification service providers who were accredited or supervised by supervisory body of a particular EU Member State in the time of the certificate issuance.



Annex B (informative) Examples of CRL

Note: Data used in the example of the list of revoked certificates are just informative and in fact they are not available.

```
SEQUENCE {
  SEQUENCE {
    INTEGER 1
    SEQUENCE {
      OBJECT IDENTIFIER sha256WithRSAEncryption (1 2 840 113549 1 1 11)
      NULL
      }
    SEQUENCE {
      SET {
        SEQUENCE {
          OBJECT IDENTIFIER countryName (2 5 4 6)
          PrintableString 'SK'
          }
        }
      SET {
        SEQUENCE {
          OBJECT IDENTIFIER localityName (2 5 4 7)
          UTF8String 'Bratislava'
          }
        }
      SET {
        SEQUENCE {
          OBJECT IDENTIFIER organizationName (2 5 4 10)
          UTF8String 'Narodny bezpecnostny urad'
          }
        }
      SET {
        SEQUENCE {
          OBJECT IDENTIFIER commonName (2 5 4 3)
          UTF8String 'CA NBU'
          }
        }
      }
    UTCTime 05/08/2010 12:58:57 GMT
    UTCTime 08/08/2010 20:14:39 GMT
    SEQUENCE {
      SEQUENCE {
        INTEGER 5061
        UTCTime 12/10/2009 20:01:17 GMT
        SEQUENCE {
          SEQUENCE {
            OBJECT IDENTIFIER cRLReason (2 5 29 21)
            OCTET STRING, encapsulates {
              ENUMERATED 5
              }
            }
          }
        }
      SEQUENCE {
        INTEGER 5101
        UTCTime 12/07/2005 13:02:00 GMT
        SEQUENCE {
          SEQUENCE {
            OBJECT IDENTIFIER cRLReason (2 5 29 21)
            OCTET STRING, encapsulates {
              ENUMERATED 1
            }
          }
        }
      }
    SEQUENCE {
      INTEGER 715
      UTCTime 13/07/2005 18:42:12 GMT
      }
    }
  [0] {
    SEQUENCE {
      SEQUENCE {
        OBJECT IDENTIFIER authorityKeyIdentifier (2 5 29 35)
        OCTET STRING, encapsulates {
          SEQUENCE {
            [0]
              DF AF AD 80 AA 83 A1 2A 1D BB DF 5C 33 4A 1D 8E
              11 82 5E 71
            }
          }
        }
      SEQUENCE {
        OBJECT IDENTIFIER cRLNumber (2 5 29 20)
        OCTET STRING, encapsulates {
          INTEGER 81
          }
        }
      SEQUENCE {
        OBJECT IDENTIFIER issuingDistributionPoint (2 5 29 28)
        OCTET STRING, encapsulates {
          SEQUENCE {
            [0] {
              [0] {
                [6]
                ' http://ep.nbusr.sk/kca/RootCaNBU3.crl'
                }
              }
            }
          }
        }
      }
    }
  }
SEQUENCE {
  OBJECT IDENTIFIER sha256WithRSAEncryption (1 2 840 113549 1 1 11)
  NULL
  }
BIT STRING
  93 FA 85 6E 57 60 B8 E2 08 D0 E1 1B A3 A9 ED 51
  D6 99 66 C3 CD ED B7 6B 95 25 41 F3 EC 7C BB ED
  7C 35 8D 16 01 13 3C 7A 32 66 E5 7D B6 9D D6 D0
  4A 9A 39 53 29 11 40 F9 59 F5 1C 0F D1 CA BD 04
  2C 4B 97 54 35 21 0E D7 71 FA 93 F3 52 3E 59 A0
  9C 27 E7 6A E1 EF B8 20 4E 92 E7 F8 5F AD A3 80
  9F 98 A7 FC D0 54 05 27 F8 8F 4F C4 14 DE 35 6F
  47 42 F5 6D F6 49 DF 44 64 9E 07 6F 9B 50 7F 0C
          [ Another 128 bytes skipped ]
}
```

Annex C (informative) Examples of data being sent in MIME

Data which are sent to the Authority must contain minimal MIME attributes and parameters of types defined in the following examples but may also contain MIME attributes and parameters of other types.

Table 10 Basic MIME types for e-mail and HTTP protocol

	Registered MIME Content-Type	**Short description**
1.	`message/rfc822`	General envelope marking of MIME message containing MIME types specified bellow.

2.	`multipart/mixed;`
	`boundary="divider of content parts"`
		It defines the sequence of signed documents whose MIME coding is separated by a divider indicated in the attribute *boundary*.

3.	`text/plain; charset=UTF-8`	ASCII textual document in UTF-8 coding.
4.	`application/pkix-cert`	DER coded certificate in base64 and the file extension is ".cer"
5.	`application/pkix-crl`	DER coded CRL in base64 and the file extension is ".crl"
6.	`application/ocsp-reques`	DER coded OCSPRequest in base64 and the file extension is ".ORQ"
7.	`application/ocsp-response`	DER coded BasicOCSPResponse in base64 and the file extension is ".ORS"
8.	`application/pkcs7-mime`	It contains DER coded CMS in base64 which includes objects of types EnvelopedData or SignedData. MIME heading must contain the parameter smime-type=enveloped-data and the file extension is ".p7m"
9.	`application/pkcs7-signature`	It contains one DER coded CMS object in base64 which includes objects of the type SignedData. The file extension application/pkcs7-signature is ".p7m" and ".p7s" in external signature.

Table 11 Basic MIME types of coding

	MIME Content-Transfer-Encoding	**Short description**
1.	`8bit`	Coding of the character up to 8 bits.
2.	`base64`	Coding of the document by Base64.

C.1 The example of encrypted message format

```
MIME-Version: 1.0
Content-Type: application/pkcs7-mime;
       smime-type=enveloped-data;
       name="smime.p7m";
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
       filename="smime.p7m"

MIKGHwYJKoZIhvcNAQcDoIKGEDCChgwCAQAxggFDMIIBPwIBADAnMBsxCzAJBgNVBAYTAlJVMQww
CgYDVQQDEwN3d3cCCF24n57g9O/YMA0GCSqGSIb3DQEBAQUABIIBAE5PWSZksLBHA7h2gS6xCLhq
n4ZwYP7WU9iKZIHJ846ZYmcy2gwPZY8geNatGFa+nQKJNGkPgmRfD90Nf0Sy4JPYNkuZDp+nMYLP
...
p4nP+cFreMIEAiFS9SwfwzUN/EX//cfQ6A==
```

C.2 The example of signed message format

```
MIME-Version: 1.0
Content-Type: application/pkcs7-mime;
       smime-type=signed-data;
       name=smime.p7m
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
       filename=smime.p7m

567GhIGfHfYT6ghyHhHUujpfyF4f8HHGTrfvhJhjH776tbB9HG4VQbnj7
...
HUujhJh4VQpfyF467GhIGfHfYGTrfvbnjT6jH7756tbB9H7n8HHGghyHh
6YT64V0GhIGfHfQbnj75
```

C.3 The example of the format of sent certificates in one multipart MIME coding

```
Content-Type: multipart/mixed;
       boundary="----=_NextPart_000_"

This is a multi-part message in MIME format.

------=_NextPart_000_
Content-type: text/plain; charset=UTF-8
Content-Transfer-Encoding: 8bit

The attachment of e-mail contains qualified certificates and maintenance
certificates for maintenance of QES.

------=_NextPart_000_
Content-Type: application/pkix-cert
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
       filename="cert1.cer"

e1xydGYxXGFuc2lcYW5zaWNwZzEyNTBcZGVmZjBcZGVmbGFuZzEwNTF7XGZvbnR0Ymx7XGYwXGZz
d2lzc1xmY2hhcnNldDIzOHtcKlxmbmFtZSBBcmlhbDt9QXJpYWwgQ0U7fXtcZjFcZm5pbFxmY2hh
...
YmVcJ2ZhXCdlOGEgXCc5ZVwnZWRcJzllbGlcJ2U4a3UgbVwnZTRzYSBuXCdmYVwnOWQgYSBtXCdm
ZGxcJ2U4aVwnZThrYSBrXCdmNFwnZjIgXGxhbmcxMDMzXGYxXHBhcg0KfQ0KAA==

------=_NextPart_000_
Content-Type: application/pkix-cert
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
       filename="cert2.cer"

JVBERi0xLjQKJcfsj6IKNSAwIG9iago8PC9MZW5ndGggNiAwIFIvRmlsdGVyIC9GbGF0ZURlY29k
ZT4+CnN0cmVhbQp4nIVSPU8DMQwVLZRyoEJL+doyJsOFON9ekRASG9VtlKmITkVq+f8STu+uOemQ
...
dCAxIDAgUiAvSW5mbyAyIDAgUgovSUQgWzw1QzIyNUI0RkIxQzU2RTVFMEUxOTAyQzgyNTdDOUI4
Nj48NUMyMjVCNEZCMUM1NkU1RTBFMTkwMkM4MjU3QzlCODY+XQo+PgpzdGFydHhyZWYKMTQ5MjIK
JSVFT0YK

------=_NextPart_000_--
```

C.4 The example of the format of sent CRL in one multipart MIME coding

```
Content-Type: multipart/mixed;
       boundary="----=_NextPart_000_"

This is a multi-part message in MIME format.

------=_NextPart_000_
Content-type: text/plain; charset=UTF-8
Content-Transfer-Encoding: 8bit

The attachment of e-mail contains CRL of revoked qualified certificates and
maintenance certificates for maintenance of QES.

------=_NextPart_000_
Content-Type: application/pkix-crl
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
       filename="crl1.crl"

e1xydGYxXGFuc2lcYW5zaWNwZzEyNTBcZGVmZjBcZGVmbGFuZzEwNTF7XGZvbnR0Ymx7XGYwXGZz
d2lzc1xmY2hhcnNldDIzOHtcKlxmbmFtZSBBcmlhbDt9QXJpYWwgQ0U7fXtcZjFcZm5pbFxmY2hh
...
NFx1YzFccGFyZFxmMFxmczIwXCdjOGlzdG8gdGVzdCBcJ2U4byBcJzlhXCdlOGlqIFwnOWRhIFwn
YmVcJ2ZhXCdlOGEgXCc5ZVwnZWRcJzllbGlcJ2U4a3UgbVwnZTRzYSBuXCdmYVwnOWQgYSBtXCdm
ZGxcJ2U4aVwnZThrYSBrXCdmNFwnZjIgXGxhbmcxMDMzXGYxXHBhcg0KfQ0KAA==

------=_NextPart_000_
Content-Type: application/pkix-crl
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
       filename="crl2.crl"

JVBERi0xLjQKJcfsj6IKNSAwIG9iago8PC9MZW5ndGggNiAwIFIvRmlsdGVyIC9GbGF0ZURlY29k
ZT4+CnN0cmVhbQp4nIVSPU8DMQwVLZRyoEJL+doyJsOFON9ekRASG9VtlKmITkVq+f8STu+uOemQ
...
dCAxIDAgUiAvSW5mbyAyIDAgUgovSUQgWzw1QzIyNUI0RkIxQzU2RTVFMEUxOTAyQzgyNTdDOUI4
Nj48NUMyMjVCNEZCMUM1NkU1RTBFMTkwMkM4MjU3QzlCODY+XQo+PgpzdGFydHhyZWYKMTQ5MjIK
JSVFT0YK

------=_NextPart_000_--
```

C.5 The example of confirmation in the format of integrity signature from the Authority

```
MIME-Version: 1.0
Content-Type: multipart/signed;
      protocol="application/pkcs7-signature";
      micalg=SHA256;
      boundary="--=_NextPart16x10x2008at11x57x17x4CB5"

This is a multi-part message in MIME format.

----=_NextPart16x10x2008at11x57x17x4CB5
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: 8bit

FILE=selesp.cer
HASH(SHA256:2 16 840 1 101 3 4 2 1)=
3ACF7A60B3F1219AE46E8E0F83D1B1C2C44249FD9578520EC19BDF6D39693B50
```

```
NOTICE=NO - Neoverený - nezaradený do databázy NBÚ
FILE=aca_disig.cer
HASH(SHA256:2 16 840 1 101 3 4 2 1)=
5872456739B61BFEB55D8715567A2E815FA09147AF0AC998685CA27B5B969547
NOTICE=OK - Overený - zaradený do databázy NBÚ
FILE=Korenova_CA_pre_kvalifikovane_certifikaty_1.cer
HASH(SHA256:2 16 840 1 101 3 4 2 1)=
6FBF021174831BE8B5889C9077F7BD6C385B5541B759E2F096D7D3BDBF774CDB
NOTICE=OK - Overený - zaradený do databázy NBÚ

----=_NextPart16x10x2008at11x57x17x4CB5
Content-Type: application/pkcs7-signature;
      name="smime.p7s"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
      filename="smime.p7s"
```

```
MIIFvgYJKoZIhvcNAQcCoIIFrzCCBasCAQExDzANBglghkgBZQMEAgEFADALBgkqhkiG9w0BBwGg
ggNsMIIDaDCCAlCgAwIBAgIIWzP7mMqijPwwDQYJKoZIhvcNAQEFBQAwGjELMAkGA1UEBhMCU0sx
CzAJBgNVBAMTAmphMB4XDTA4MDkxNTEwNDg0NloXDTA5MDkxNTEwNDg0NlowGjELMAkGA1UEBhMC
...
nWueKbh/pGtMXsotpc4OQrmdxtJEwB0ADY2pnWxgT/z3CoA4ZFO0nbrLpw3IzlWgEckXJAfG4VlX
V1XnLKS+c+itA2y8Z1gye2e1sMw0aSomyIsRUBM7TExP0jGb0rygCF3g6u3VpSlwSg==
```

```
----=_NextPart16x10x2008at11x57x17x4CB5--
```

C.6 The example of the request format for the certificate status

```
MIME-Version: 1.0
Content-Type: application/ocsp-reques;
       name="ocsp.orq";
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
       filename="ocsp.orq"
```

```
MIKGHwYJKoZIhvcNAQcDoIKGEDCChgwCAQAxggFDMIIBPwIBADAnMBsxCzAJBgNVBAYTAlJVMQww
CgYDVQQDEwN3d3cCCF24n57g9O/YMA0GCSqGSIb3DQEBAQUABIIBAE5PWSZksLBHA7h2gS6xCLhq
...
p4nP+cFreMIEAiFS9SwfwzUN/EX//cfQ6A==
```

C.7 The example of the response format to the request for the certificate status

```
MIME-Version: 1.0
Content-Type: application/ocsp-response;
       name="ocsp.ors";
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
       filename="ocsp.ors"
```

```
MIKGHwYJKoZIhvcNAQcDoIKGEDCChgwCAQAxggFDMIIBPwIBADAnMBsxCzAJBgNVBAYTAlJVMQww
CgYDVQQDEwN3d3cCCF24n57g9O/YMA0GCSqGSIb3DQEBAQUABIIBAE5PWSZksLBHA7h2gS6xCLhq
n4ZwYP7WU9iKZIHJ846ZYmcy2gwPZY8geNatGFa+nQKJNGkPgmRfD90Nf0Sy4JPYNkuZDp+nMYLP
...
p4nP+cFreMIEAiFS9SwfwzUN/EX//cfQ6A==
```

Annex D (informative) Bibliography

Basic documents of the Slovak legislation on electronic signature
http://www.nbusr.sk/en/electronic-signature/legislation/index.html
Qualified electronic signature formats
http://www.nbusr.sk/en/electronic-signature/approved-formats/index.html
Certification path building and validity verification of certificates
http://www.nbusr.sk/en/electronic-signature/verification/index.html

- IETF RFC 4158 "Internet X.509 Public Key Infrastructure: Certification Path Building"

NOTE: Available at http://www.rfc-archive.org/getrfc.php?rfc=4158

- IETF RFC 5217 "Multi-Domain PKI Interoperability" July 2008

NOTE: Available at http://www.rfc-archive.org/getrfc.php?rfc=5217

- IETF RFC 4853 (2007): "Cryptographic Message Syntax (CMS) Multiple Signer Clarification"
- IETF RFC 3447 (2003): "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1"
- ISO/IEC 8825-1:1998, Information technology — ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)
- ISO/IEC 19794-2:2005, Biometrics — Biometric Data Interchange Formats — Part 2: Finger Minutiae
- IETF RFC 3279 (2002): "Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile"
- IETF RFC 4055 (2005): "Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure - Certificate and Certificate Revocation List (CRL) Profile"
- IETF RFC 3281 (2002): "An Internet Attribute Certificate profile for Authorization"
- IETF RFC 3370 (2002): "Cryptographic Message Syntax (CMS) Algorithms"
- ETSI TR 102 038: "TC Security - Electronic Signatures and Infrastructures (ESI); XML format for signature policies"
- ETSI TS 101 861: "Time stamping profile"
- EN 14890-2:2008: "Application Interface for smart cards used as Secure Signature Creation Devices - Part 2: Additional Services"
- ETSI TS 101 456: "Electronic Signatures and Infrastructures (ESI); Policy requirements for certification authorities issuing qualified certificates"
- ETSI TS 102 042: "Electronic Signatures and Infrastructures (ESI); Policy requirements for certification authorities issuing public key certificates"
- CWA 14167-1: "Security Requirements for Trustworthy Systems Managing Certificates for Electronic Signatures - Part 1: System Security Requirements"
- CWA 14167-2: "Security Requirements for Trustworthy Systems Managing Certificates for Electronic Signatures - Part 2: Cryptographic module for CSP Signing Operations with Backup - Protection Profile"
- CWA 14167-3: "Security Requirements for Trustworthy Systems Managing Certificates for Electronic Signatures - Part 3: Cryptographic module for CSP key generation services - Protection profile (CMCKG-PP)"
- CWA 14167-4: "Security Requirements for Trustworthy Systems Managing Certificates for Electronic Signatures - Part 4: Cryptographic module for CSP signing operations - Protection profile - CMCSO PP"
- W3C Recommendation (10 June 2008): "XML Signature Syntax and Processing (Second Edition)"

NOTE: Available at http://www.w3.org/TR/xmldsig-core/

- W3C Recommendation (10 December 2002): "XML Encryption Syntax and Processing"

NOTE: Available at http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/

- CWA 14169: "Secure Signature-Creation Devices "EAL 4+"
- IETF RFC 4949 "Internet Security Glossary, Version 2" August 2007

NOTE: Available at http://www.rfc-archive.org/getrfc.php?rfc=4949

- NIST X.509 path validation test suite

NOTE: Available at http://csrc.nist.gov/pki/testing/x509paths.html http://csrc.nist.gov/pki/testing/pathdiscovery.html

- Object Identifier (OID) Repository: ITU-T X.660 & X.670 Recommendation series (or ISO/IEC 9834 series of International Standards)

NOTE: Available at http://www.oid-info.com/

- FESA – Forum of European Supervisory Authorities,

NOTE: Available at http://www.fesa.rtr.at

- OID tree structure,

NOTE: Available at http://www.darmstadt.gmd.de/secude/Doc/htm/oidgraph.htm

- Common PKI specification: "COMMON PKI SPECIFICATIONS FOR INTEROPERABLE APPLICATIONS FROM T7 & TELETRUST - SPECIFICATION PART 4:OPERATIONAL PROTOCOLS, http://www.common-pki.org/uploads/media/Common-PKI_v2.0.pdf"
- Internet Draft "X.509 Public Key Infrastructure: Additional Algorithms and Identifiers for DSA and ECDSA"

NOTE: Available at http://tools.ietf.org/html/draft-ietf-pkix-sha2-dsa-ecdsa-05

- Internet Draft "X.509 Public Key Infrastructure Time-Stamp Protocol (TSP) "

NOTE: Available at http://tools.ietf.org/html/draft-ietf-pkix-rfc3161bis-01

- TeleTrusT Deutschland e. V., "OID-Liste",

NOTE: Available at http://www.teletrust.de/index.php?id=171

European Commission http://ec.europa.eu/

- Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures

NOTE: Available at
http://eur-lex.europa.eu/smartapi/cgi/sga_doc?smartapi!celexapi!prod!CELEXnumdoc&lg=EN&numdoc=31999L0093&model=guichett

- IDABC stands for Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens. - eSignature Agenda & Presentations

NOTE: Available at http://ec.europa.eu/idabc/en/document/7312

- European Network and Information Security Agency (ENISA)

NOTE: Available at http://www.enisa.europa.eu/

- PKIX Status Pages http://tools.ietf.org/wg/pkix/

Annex E History

Version	Date of issuing	Note	Editor
V 1.0	30.9.2004	First edition (revoked)	Peter Rybár
V 1.1	14.8.2005	Second edition	Peter Rybár, NSA
V 1.2	6.11.2005	Unified NSA format	Peter Rybár, NSA
V 3.0 No.2188/2010/IBEP/OEP-001	17.1.2010	Addition of OCSP, delivering the information to the OCSP NSA and the TSL implementation under Decision of EU Commission 2009/767/ES.	Peter Rybár, NSA