NÁRODNÝ BEZPEČNOSTNÝ ÚRAD

Verzia 1.4

Kontrola certifikačnej cesty

19. 11. 2006

NÁRODNÝ BEZPEČNOSTNÝ ÚRAD

Sekcia informačnej bezpečnosti a elektronického podpisu

Budatínska č. 30, 850 07 Bratislava 57

http://www.nbusr.sk/

e-mail: sep@nbusr.sk

Obsah

1 Úvod 4
2 Predmet dokumentu 4
3 Odkazy 5
4 Skratky 6
5 Komponenty certifikačnej cesty 7
6 Pravidlá určujúce postup kontroly certifikačnej cesty 10
7 Posun času kontroly pri overovaní podpisov v certifikačnej ceste 11
8 Archivovanie a overovanie archivovanej certifikačnej cesty 12
9 Atribúty certifikátu X.509 pri kontrole certifikačnej cesty 13
9.1 Postupnosť kontroly položiek pri vytváraní certifikačnej cesty 13
10 Overovanie certifikačnej cesty 14
10.1 Algoritmus zostavenia certifikačnej cesty 15
10.2 Vyhľadávanie certifikačných ciest pomocou rekurzívnej procedúry 16
10.3 Overenie nájdených certifikačných ciest 17
Príloha A (informatívna) Overovanie pomocou OCSP s pozitívnou odpoveďou a s OCSP odpoveďou podľa RFC 2560 19
A.1 OCSP s pozitívnou odpoveďou na základe údajov z databázy 19
A.2 OCSP odpoveď podľa RFC 2560 20
Príloha B (normatívna) Overovanie platnosti certifikátu 21
B.1 Overenie pomocou OCSP 21
Tabuľka 1. Overenie s OCSP 21
B.2 Overenie pomocou CRL 23
Tabuľka 2. Overenie s CRL 23
Príloha C (informatívna) Revízie vykonané od predošlého vydania 24
C.1 Pridané požiadavky 24
C.2 Upravené požiadavky 24
C.3 Vysvetlenia 24
C.4 Publikačné zmeny 24
Príloha D (informatívna) Zoznam použitej literatúry 25
Príloha E História 26

1 Úvod

Používanie technológií založených na PKI a certifikátoch X.509 získalo po implementácii požiadaviek z Európskej smernice 1999/93/ES v členských krajinách EU nový právny rozmer. Jednotný postup aplikovania pravidiel na vytvorenie a overenie certifikačnej cesty je jedným zo základných predpokladov, aby bolo možné výsledky, získané na základe technických postupov, správne interpretovať a vyvodiť z nich správne rozhodnutia. Využitie PKI technológie a certifikátov X.509 dovoľuje veľkú variabilitu, ktorú je potrebné zjednotiť. Toto je možné dosiahnuť definovaním a publikovaním profilov na zabezpečenie jednotných postupov a požiadaviek, ktoré prinesú rovnaké výsledky tak v rámci jedného štátu, ako aj medzinárodne.

2 Predmet dokumentu

Účelom dokumentu „Kontrola certifikačnej cesty“ je popis procesu vytvárania a overovania certifikačnej cesty podľa platnej legislatívy SR, štandardov X.509, ETSI a RFC požiadaviek kladených na kvalifikované certifikáty a zaručené elektronické podpisy (Qualified Electronic Signatures), so snahou zabezpečiť zodpovedajúce overenie v podmienkach SR, zároveň s ohľadom na snahu dosiahnuť interoperabilitu s členskými krajinami EÚ.

Legislatívne i technologicky sa interoperabilita v krajinách EÚ buduje na základoch EU smernice 1999/93/ES a akceptovania dokumentov štandardizačných inštitúcií. Základom je požiadavka, podľa ktorej členské štáty musia zaistiť, aby zaručený elektronický podpis (Qualified Electronic Signature) na základe článku 5, Európskej smernice 1999/93/ES:

a) spĺňal právne požiadavky na podpis vo vzťahu k údajom v elektronickej forme rovnako, ako ich spĺňa vlastnoručný podpis vo vzťahu k papierovým dokumentom,
b) bol prípustný ako dôkaz v konaní pred súdom.

Podstatou tohto dokumentu nie je vytvorenie iba obecného štandardu pre uvedenú oblasť, ale vytvorenie jednoznačného minimálneho profilu a odporúčania pre poskytovateľov certifikačných služieb, tvorcov aplikácií a používateľov elektronického podpisu.

3 Odkazy

Odkazy na dokumenty, ktoré definujú použité typy a postupy.

[1] RFC 3280	X.509 PKI Certificate and Certificate Revocation List	04/2002
[2] RFC 2560	X.509 PKI Online Certificate Status Protocol	06/1999
[3] RFC 3852	Cryptographic Message Syntax	07/2004
[4] RFC 3161	Time-Stamp Protocol (TSP)	08/2001
[5] RFC 3739	Qualified Certificates Profile	03/2004
[6] ETSI	TS 101 862 V1.3.2 Qualified Certificate Profile	06/2004
[7] ETSI	TS 101 733 Electronic Signature Formats	
[8] ETSI	TR 102 272 ASN.1 format for signature policies	
[9] ETSI	TR 102 038 XML format for signature policies	
[10] ISO/IEC	ITU-T X.509 / ISO/IEC 9594	08/2005
[11] ETSI	TS 102 280 X.509 V.3 Cert. Profile for Cert. Issued to Natural Persons	
[12] ETSI	TS 102 231 Provision of harmonized Trust-service status information	

4 Skratky

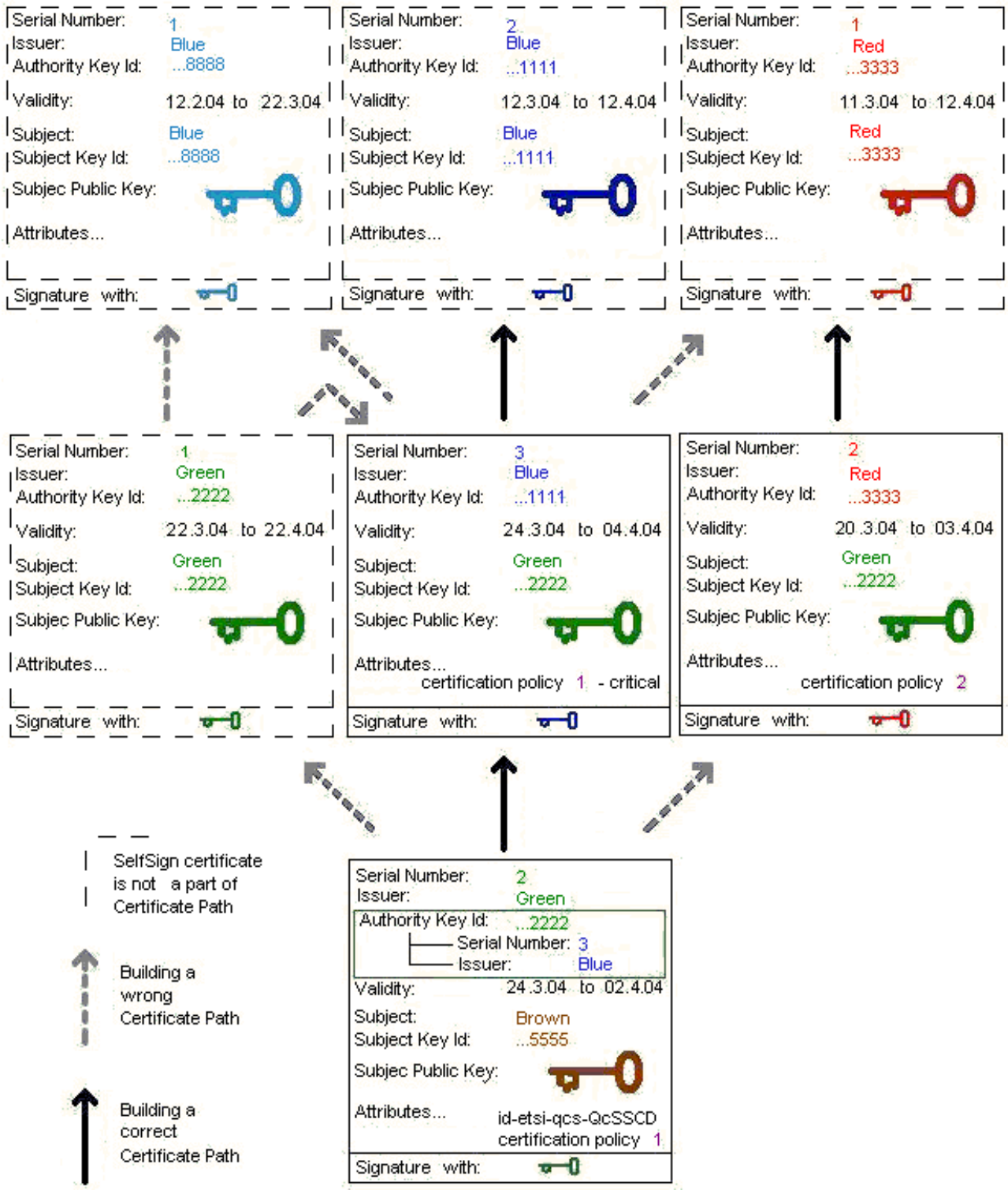
ASCII	American Standard Code for Information Interchange
ASN.1	Abstract Syntax Notation 1
CA	Certification Authority
CAdES	CMS Advanced Electronic Signature
CMS	Cryptographic Message Syntax
CRL	Certificate Revocation List
DER	Distinguished Encoding Rules (for ASN.1)
ESS	Enhanced Security Services (enhances CMS)
GMT	Greenwich Mean Time
HTTP	HyperText Transfer Protocol
ISIS-MTT	Industrial Signature Interoperability Standard - MailTrusT
ISO	International Organization for Standardization
MIME	Multipurpose Internet Mail Extensions
OCSP	Online Certificate Status Provider / Protocol
OID	Object IDentifier
PKCS	Public Key Cryptographic Standards, Standards published by RSA, Labs.
PKIX	internet X.509 Public Key Infrastructure
PVM	Path Validation Module
QC	Qualified Certificate
RSA	Rivest, Shamir and Adleman Algorithm
SHA	Secure Hash Algorithm
SSCD	Secure-Signature-Creation Device
TSA	Time-Stamping Authorities
TSP	Time Stamp Protocol
TST	Time-Stamp Token
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
XAdES	XML Advanced Electronic Signature
XML	extensible Markup Language
ZEP	Zaručený elektronický podpis (Qualified Electronic Signature)

5 Komponenty certifikačnej cesty

Certifikačná cesta sa zostavuje od **certifikátu koncovej entity** až po **dôveryhodnú koreňovú CA.** Teda certifikačné cesty pozostávajú z **certifikátu koncového užívateľa** [11] a **certifikátov certifikačných autorít CA** až po CA, ktorej certifikát je vydaný **dôveryhodným bodom** (***trust anchor*** - dvojica **DN mena** a **verejného kľúča** (Distinguished Name and Public Key)**,** zvyčajne reprezentovaná vo forme koreňového certifikátu). **Koreňový certifikát (*selfSigned*)** nie je súčasťou certifikačnej cesty a slúži len na uchovanie dôveryhodného bodu (DN meno a verejný kľúč) **koreňovej CA** v PKIX.

V nasledujúcom obrázku sú na príklade naznačené správne a chybne zostavené certifikačné cesty, kde certifikát je identifikovaný dvojicou (meno vydavateľa certifikátu, sériové číslo certifikátu):

1. ***selfSigned* certifikáty** certifikačných autorít: exspirovaný certifikát, nový certifikát, ktorý bol vydaný pred exspirovaním certifikátu (modrý vlastník - subject a modrý kľúčový pár) a certifikát nedôveryhodnej (pre overovateľa neznámej) CA (červený vlastník a kľúčový pár)
2. *selfSigned* certifikát, **certifikát akreditovanej CA** a CA certifikát vydaný nedôveryhodnou CA, pretože červený *selfSign* nemá dôveru overovateľa (zelený vlastník a kľúčový pár)
3. **certifikát koncovej entity** - podpisovateľa[11] (hnedý certifikát a kľúčový pár)



Chybne vytvorené certifikačné cesty:

1. (Zelený, 2) – (Zelený, 1):
   a. koreňové certifikáty sa nevkladajú do certifikačnej cesty
   b. nezhoduje sa meno v *AuthorityKeyIdentifier authorityCertIssuer* s menom *Issuer* v certifikáte vydavateľa, „Modrý“ ≠ „Zelený“
   c. nezhoduje sa sériové číslo certifikátu v *AuthorityKeyIdentifier authorityCertSerialNumber*, so *serialNumber* v certifikáte vydavateľa, 3 ≠ 1

2. (Zelený, 2) – (Červený, 2):
   a. nezhoduje sa meno v *AuthorityKeyIdentifier authorityCertIssuer* s menom *Issuer* v certifikáte vydavateľa, „Modrý“ ≠ „Červený“
   b. nezhoduje sa sériové číslo certifikátu v *AuthorityKeyIdentifier authorityCertSerialNumber*, so *serialNumber* v certifikáte vydavateľa, 3 ≠ 2
3. (Zelený, 1) – (Modrý, 1):
   a. koreňové certifikáty sa nevkladajú do certifikačnej cesty
   b. nezhoduje sa meno v *Issuer* s menom *Subject* v certifikáte vydavateľa, „Zelený“ ≠ „Modrý“
   c. nezhoduje sa *AuthorityKeyIdentifier keyIdentifier* so *SubjectKeyIdentifier* v certifikáte vydavateľa, ...2222 ≠ ...8888
   d. nepodarí sa overiť podpis certifikátu s verejným kľúčom nesprávneho vydavateľa
   e. vydaný certifikát nie je vydaný v čase platnosti certifikátu nesprávneho vydavateľa
4. (Modrý, 3) – (Modrý, 1):
   a. koreňové certifikáty sa nevkladajú do certifikačnej cesty
   b. nezhoduje sa *AuthorityKeyIdentifier keyIdentifier* so *SubjectKeyIdentifier* v certifikáte vydavateľa, ...1111 ≠ ...8888
   c. nepodarí sa overiť podpis certifikátu s verejným kľúčom nesprávneho vydavateľa
   d. vydaný certifikát nie je vydaný v čase platnosti certifikátu nesprávneho vydavateľa
5. (Modrý, 3) – (Červený, 1):
   a. koreňové certifikáty sa nevkladajú do certifikačnej cesty
   b. nezhoduje sa meno v *Issuer* s menom *Subject* v certifikáte vydavateľa, „Modrý“ ≠ „Červený“
   c. nezhoduje sa *AuthorityKeyIdentifier keyIdentifier* so *SubjectKeyIdentifier* v certifikáte vydavateľa, ...1111 ≠ ...3333
   d. nepodarí sa overiť podpis certifikátu s verejným kľúčom nesprávneho vydavateľa

6 Pravidlá určujúce postup kontroly certifikačnej cesty

Čas kontroly je čas, ku ktorému sa overuje platnosť elektronického podpisu dokumentu, certifikátu, CRL alebo OCSP v certifikačnej ceste:

- je to čas najstaršej platnej časovej pečiatky digitálneho podpisu v elektronickom podpise,
- pri certifikáte musí byť tento čas z intervalu certifikátu (*notBefore, notAfter*),
- pri CRL je to čas (*thisUpdate*),
- pri OCSP je to čas (*producedAt*),
- alebo čas overovania z bezpečného auditného záznamu obsahujúceho hash podpisu (pri podpise bez časovej pečiatky),
- alebo čas blízky aktuálnemu času overovania, v ktorom bolo vydané CRL (OCSP) na overenie certifikátu podpisovateľa. Je potrebné získať aj CRL (OCSP) na overenie celej certifikačnej cesty až po certifikát vydaný koreňovou CA, kde každé CRL (OCSP) v ceste, od CRL (OCSP) na overenie certifikátu podpisovateľa dokumentu po koreňovú CA, je vydané v rovnakom alebo neskoršom čase ako predchádzajúce CRL (OCSP).

CautionPeriod (grace period) je časový interval, ktorý umožní, aby sa informácie o zrušení certifikátu zneplatňovacím procesom zverejnili pre overovateľov, teda je to minimálny čas, počas ktorého musí overovateľ čakať od času kontroly, aby získal záväzné informácie o zrušení certifikátov. Tiež je to časový limit pre poskytovateľov certifikačných služieb, aby informácie o zrušení zverejnili.
Teda *cautionPeriod* je časový interval, ktorý je maximom z *grace period* jednotlivých certifikačných autorít, ktorých certifikáty tvoria certifikačnú cestu. Maximálny interval *grace period* je na Slovensku definovaný legislatívou na 24 hodín.

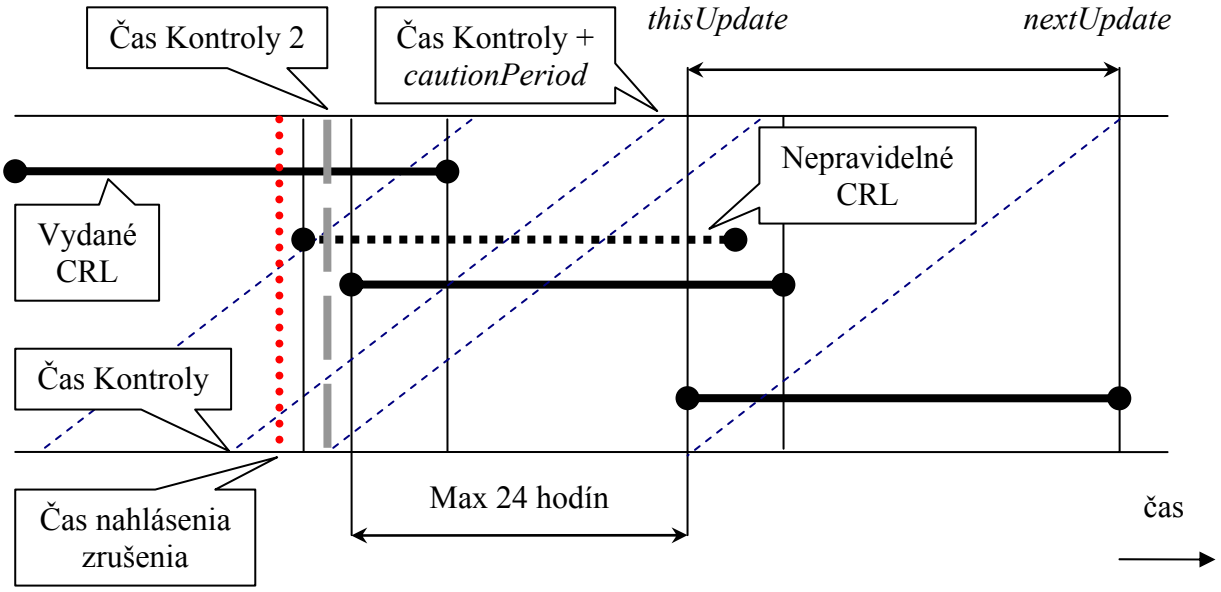
Položka *cautionPeriod* sa nachádza napríklad v podpisovej politike alebo v nastaveniach aplikácií overujúcich certifikačnú cestu.

Čas v CRL *nextUpdate* môže byť kratší než je *CRL.thisUpdate* + *gracePeriod* a nemusí byť pravidelný v jednotlivých CRL (napríklad o 8 hodín, o 16 hodín).

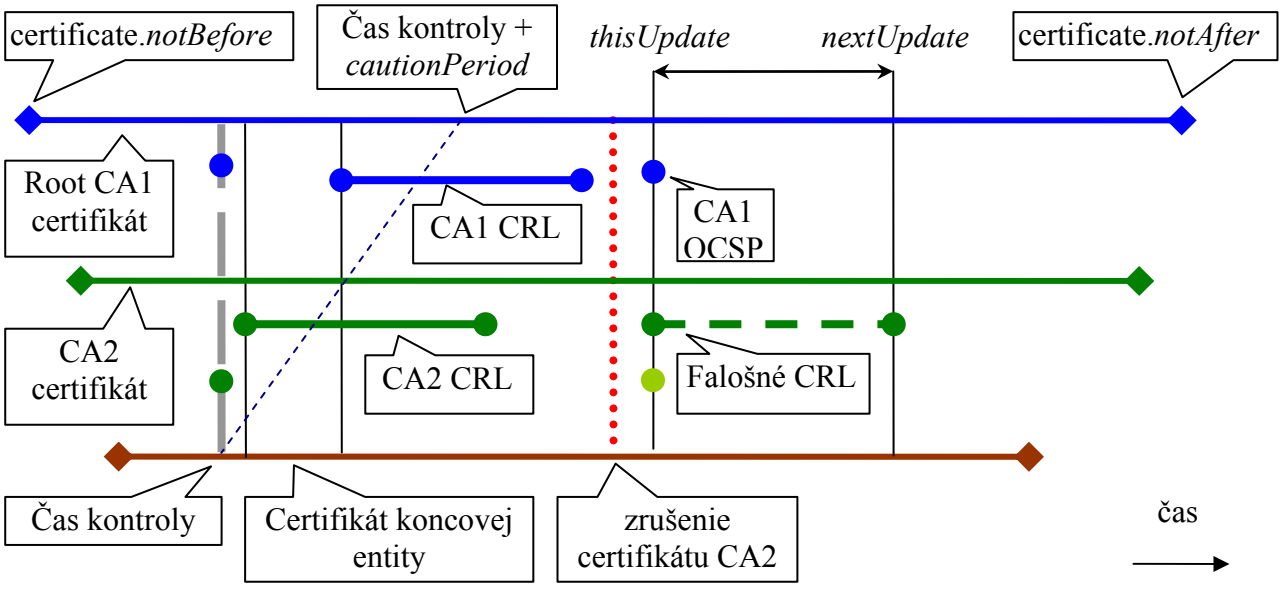
Pri overovaní platnosti certifikátov, ktoré boli vydané v súlade s pravidlami slovenskej legislatívy, (teda obsahujú OID certifikačnej politiky s hodnotou: 1.3.158.36061701.0.0.0.1.2.2), musí platiť:

- certifikát v CRL (OCSP) nesmie byť v stave *certificateHold* a *removeFromCRL,* teda jeho platnosť nemôže byť na určitý čas pozastavená;
- CRL nesmie byť vydané po čase *nextUpdate* posledne vydaného CRL;
- ak CA vydá nové CRL v čase platnosti iného (aktuálne platného) CRL, v položke *nextUpdate* nového CRL musí byť čas väčší alebo rovný času *nextUpdate* aktuálne platného CRL, vydané CRL musia tvoriť časovú reťaz;
- ak je vydané CRL_1 ($OCSP_1$), tak nemôže byť vydané CRL_2 ($OCSP_2$) také, ktoré by zrušilo certifikát pred časom vydania CRL_1($OCSP_1$), v ktorom certifikát ešte nebol zrušený.

7 Posun času kontroly pri overovaní podpisov v certifikačnej ceste



Pri kontrole podpisu CRL (OCSP) s certifikátom z certifikačnej cesty a následnej kontrole platnosti certifikátu, ktorým je CRL (OCSP) overené, je potrebné splniť pravidlo výberu CRL (OCSP) podľa času vydania. Pravidlo pre výber CRL (OCSP) znie: **Pre overenie certifikátov v certifikačnej ceste sa vyberá CRL (OCSP) s časom vydania CRL.*thisUpdate* (OCSP.*producedAt*) tak, že každé CRL (OCSP) nadriadenej CA je vydané po čase vydania CRL.*thisUpdate* (OCSP.*producedAt*) podriadenej CA.** Tento postup je potrebné dodržať kvôli overeniu platnosti podpisu CRL (OCSP), ktoré je vydané neskôr, než je čas kontroly, keďže mohlo dôjsť k zrušeniu certifikátu CA, ktorým bolo CRL(OCSP) vydané, hierarchicky nadriadenou CA.



8 Archivovanie a overovanie archivovanej certifikačnej cesty

Na ochranu platnosti všetkých podpisov, ktoré podpisovateľ a CA zrealizovali (certifikáty, CRL, atď...), a ktoré v budúcnosti môžu byť posudzované ako neplatné (kvôli zrušeniu certifikátu alebo „zostarnutiu“ algoritmu), je potrebné použiť časovú pečiatku na celú certifikačnú cestu spolu s CRL alebo OCSP. Vďaka použitiu časovej pečiatky sa celá certifikačná cesta spolu s CRL a OCSP môže použiť na dlhodobé overenie platných podpisov. Takáto pečiatka sa nazýva **archívna časová pečiatka**. Pred jej použitím je potrebné overiť a doplniť všetky podpisy a časové pečiatky, ktoré sa archívnou pečiatkou pečiatkujú, o CRL (OCSP) na overenie ich platnosti až po certifikát podpísaný koreňovým certifikátom.

Ak podpis neobsahuje novšiu archívnu časovú pečiatku, tak sa platnosť certifikátov jednotlivých časových pečiatok overuje v intervale od času, ktorý časová pečiatka obsahuje, až do aktuálneho času, pričom sa odporúča overovať s čo najaktuálnejšími údajmi. Zjednodušene sa dá povedať, že je potrebné získať aktuálne CRL (OCSP) overované koreňovým certifikátom a potom hľadať CRL (OCSP) vydané podriadenou CA, ktoré je staršie, ale zároveň nesmie byť staršie ako je čas v overovanej časovej pečiatke. Takto sa pokračuje, až kým sa nenájde CRL (OCSP) na overenie certifikátu koncovej entity (end user certificate). Ak podpis obsahuje viac archívnych časových pečiatok, tak interval na overenie platnosti archivovaných časových pečiatok je nasledovný: (čas z overovanej archivovanej pečiatky; čas z nasledujúcej archívnej pečiatky).

Archívna časová pečiatka sa musí vytvoriť ešte v čase, kedy sú všetky certifikáty v certifikačnej ceste neexspirované a na ich overenie existujú CRL (OCSP), aby ich bolo možné po archívnom opečiatkovaní spätne overiť.

Pri overovaní údajov, ktorých integrita je chránená archívnou časovou pečiatkou, sa certifikačná cesta overuje až po certifikát vydaný koreňovou CA. Pretože (pri dlhodobom overovaní) v čase, kedy sa overuje platnosť certifikačnej cesty, už pôvodný koreňový certifikát mohol exspirovať (a teda mohol byť odstránený z úložísk slúžiacich na uchovanie aktuálneho dôveryhodného koreňového certifikátu (trustAnchor – DN meno + verejný kľúč)), je potrebné získať dôveryhodnú informáciu o tom, ktorý koreňový certifikát bol v čase overovania archivovaných údajov označený ako dôveryhodný koreňový certifikát. Teda je potrebné mať evidovanú a publikovanú históriu dôveryhodných koreňových certifikátov.
Históriu o dôveryhodných koreňových certifikátoch by mala zverejňovať organizácia, ktorá vykonáva akreditáciu a dohľad nad činnosťou CA, a teda má všetky informácie, potrebné na evidovanie histórie. V rámci Európskej únie sa predpokladá vydávanie informácií o akreditovaných poskytovateľoch certifikačných služieb a informácií o histórii poskytovania akreditovaných certifikačných služieb formou TSL listu [12]. Doposiaľ však nie sú v rámci EÚ jasne stanovené pravidlá pre vydávanie TSL.
Ako riešenie tejto situácie (v rámci SR) je možné vydávať zoznam už exspirovaných dôveryhodných koreňových certifikátov vo forme archivačného podpisu z jednotlivých koreňových certifikátov. Pre publikovanie tohto zoznamu budú platiť pravidlá definované v dokumente NBÚ „**Dôveryhodné zverejňovanie zoznamov elektronických dokumentov**“.

Podpisy zoznamov akreditovaných poskytovateľov certifikačných služieb sa overujú k aktuálne dôveryhodnému koreňovému certifikátu. Záznamy o akreditovaných poskytovateľoch v zozname obsahujú aj informáciu o exspirovaných koreňových certifikátoch, ktoré boli dôveryhodné v minulosti, a ktoré je možné použiť na overenie údajov chránených archívnou časovou pečiatkou.

9 Atribúty certifikátu X.509 pri kontrole certifikačnej cesty

Pri zostavovaní certifikačnej cesty sa kontroluje zhoda v položkách, ktoré obsahuje vydaný certifikát a certifikát vydavateľa. Ak párovú položku obsahuje vydaný (alebo vydavateľov) certifikát, tak ju musí obsahovať aj vydavateľov (alebo vydaný) certifikát, inak sa certifikát zamietne.

Každé vytváranie certifikačnej cesty musí spĺňať minimálne nižšie uvedené kroky, aby sa zabránilo nejednoznačnému vytvoreniu certifikačnej cesty.

9.1 Postupnosť kontroly položiek pri vytváraní certifikačnej cesty

1. *Issuer* meno vydavateľa overovaného certifikátu sa musí zhodovať so *Subject* menom subjektu v certifikáte vydavateľa;
2. formát kvalifikovaného certifikátu musí zodpovedať certifikátu X.509 v3 v DER kódovaní;
3. podpis vydaného certifikátu sa musí overiť verejným kľúčom z certifikátu vydavateľa (podľa pravidiel definovaných v slovenskej legislatíve, v EÚ sa odporúča SHA-1 nahradiť funkciou SHA-256);
4. *v AuthorityKeyIdentifier* overovaného certifikátu sa *keyIdentifier* musí zhodovať so *SubjectKeyIdentifier* v certifikáte vydavateľa, každý certifikát musí obsahovať *SubjectKeyIdentifier;*
5. ak overovaný certifikát v *AuthorityKeyIdentifier* obsahuje *authorityCertIssuer*, tak meno sa musí zhodovať s *Issuer* menom v certifikáte vydavateľa;
6. ak overovaný certifikát v *AuthorityKeyIdentifier* obsahuje *authorityCertSerialNumber*, tak sa musí zhodovať so *serialNumber* v certifikáte vydavateľa.
7. Čas kontroly musí byť v intervale (*notBefore, notAfter)* - certifikát nesmie byť v čase kontroly exspirovaný.
8. Platnosť certifikátu sa overí na základe postupu, ktorý popisuje Príloha B.
9. Musí existovať prienik intervalov platnosti kontrolovaného certifikátu a certifikátu vydavateľa, kde čas kontroly je v spomenutom intervale prieniku.
10. Dĺžka cesty je kontrolovaná podľa zmenšujúcej sa premennej *maxPathLength*, pričom *maxPathLength* sa nastaví na *BasicConstraints.pathLengthConstraint* z certifikátu, ak je *BasicConstraints.pathLengthConstraint* menšia než *maxPathLength.* Ak je *maxPathLength* nulová, tak potom za certifikátom už nemôže nasledovať CA certifikát.
11. Certifikát vydavateľa musí obsahovať:
    - *BasicConstraints.Ca* nastavené na hodnotu *TRUE*,
    - *KeyUsage* obsahujúce *keyCertSign a* ak podpisuje CRL tak aj *cRLSign.*
12. Kvalifikovaný certifikát koncovej entity musí obsahovať:
    - *KeyUsage* s hodnotou *nonRepudiation* a prípadne aj *digitalSignature,*
    - ak je certifikát určený pre overenie časových pečiatok, tak *extendedKeyUsage* musí obsahovať *id-kp-timeStamping,*
    - ak je certifikát určený pre overenie OCSP, tak *extendedKeyUsage* musí obsahovať minimálne *id-kp-OCSPSigning.*
13. Kvalifikované certifikáty vydané v súlade so slovenskou legislatívou musia obsahovať v rozšírení *CertificatePolicies* minimálne OID *qualifiSK* (1 3 158 36061701 0 0 0 1 2 2). Podľa ETSI dokumentu [6] kvalifikované certifikáty fyzickej osoby vydané pre SSCD obsahujú OIDy *id-etsi-qcs-QcCompliance* a *id-etsi-qcs-QcSSCD* v rozšírení *QCStatements*.

Kvalifikovaný certifikát koncovej entity (podpisovateľa), vydaný akreditovanou CA musí vždy obsahovať položku *AuthorityKeyIdentifier.keyIdentifier* , za predpokladu, že by mohlo dôjsť k nejasnosti pri zostavovaní certifikačnej cesty, musia byť vyplnené všetky tri položky.

Rozšírenie certifikátu id-ce-authorityKeyIdentifier OBJECT IDENTIFIER ::= { id-ce 35 }

```
AuthorityKeyIdentifier ::= SEQUENCE {
    keyIdentifier             [0] KeyIdentifier           OPTIONAL,
    authorityCertIssuer       [1] GeneralNames            OPTIONAL,
    authorityCertSerialNumber [2] CertificateSerialNumber OPTIONAL  }
```

Medzi ďalšie položky, ktoré je potrebné vedieť spracovať a kontrolovať patria:

- *Policy Mappings* - mapovanie politiky
- *Name Constraints* - obmedzuje menný priestor podstromu
- *Policy Constraints* - obmedzuje politiky a mapovanie

Položky, ktoré pomáhajú pri podpisovaní a tiež pri zostavovaní a overovaní certifikačnej cesty:

- *CRL Distribution Points* - obsahuje cesty pre získanie CRL
- *Authority Information Access* - obsahuje cesty na CA certifikáty a OCSP
- *Subject Information Access* - obsahuje cesty na TimeStamp server

10 Overovanie certifikačnej cesty

Procedúra PVM (Path Validation Module) je jedným z kritických komponentov aplikácie, ktorá overuje certifikačnú cestu certifikátov X.509. Aplikácia overovanú cestu získa z podpisu alebo si ju zostrojí na základe podmienok pre vytvorenie certifikačnej cesty.

Výsledok procedúry PVM na overenie certifikačnej cesty musí byť zhodný s výsledkom overenia podľa dokumentu RFC 3280 v 6. kapitole. Aplikácia nemusí používať algoritmy presne podľa kapitoly 6. RFC 3280, ale výsledky musia byť zhodné.

PVM potvrdí aplikácii platnosť certifikátu koncovej entity pre vybraný účel, ku ktorému sú vytvorené certifikačné cesty končiace na certifikátoch overených dôveryhodným koreňovým certifikátom, napríklad certifikátom koreňovej CA NBÚ (KCA). V praxi je možné aj vytvorenie viacerých ciest, preto by aplikácia mala vedieť takýto stav spracovať a vybrať akceptovateľnú cestu.

Akceptovateľnú cestu môže aplikácia vybrať na základe vstupných podmienok do PVM. Vstupom môže byť dôveryhodný koreňový certifikát, ale aj množina certifikačných politík, ktorá určuje, aká cesta je akceptovateľná. Vstupy do algoritmu sú získané aj z podpisovej politiky, ak bola aplikáciou použitá. Výber cesty na základe certifikačnej politiky je určený hodnotami v rozšíreniach certifikátov a to hlavne: *Certificate Policies, Policy Mappings, Policy Constraints, Inhibit Any-Policy.*

Množina akceptovateľných politík môže vstúpiť do PVM buď:
1. explicitne, napríklad z položky podpisovej politiky
2. alebo môže byť vyžadovaná priamo v certifikáte, v ktorom je nastavené rozšírenie *PolicyConstraints requireExplicitPolicy* na kritické.

V prvom prípade sa vyžaduje, aby všetky certifikáty v certifikačnej ceste obsahovali neprázdny prienik s množinou explicitných politík.

V druhom prípade sa požaduje, aby certifikačné politiky z certifikátu, ktorý obsahuje kritické rozšírenie *PolicyConstraints,* boli buď priamo alebo prostredníctvom *policyMappings* premapovania, obsiahnuté v certifikátoch certifikačnej cesty, až po overovaný certifikát.

ETSI [6] identifikácia kvalifikovaného certifikátu v rozšírení qCStatements, ktorá identifikuje kvalifikované certifikáty fyzickej osoby vydané na verejný kľúč, ku ktorému patriaci podpisový súkromný kľúč je uložený na bezpečnom zariadení pre vytváranie elektronického podpisu SSCD:

```
id-etsi-qcs OBJECT IDENTIFIER ::= { itu-t(0) identified-organization(4) etsi(0)
id-qc-profile(1862) 1 }

id-etsi-qcs-QcSSCD OBJECT IDENTIFIER ::= { id-etsi-qcs 4 }

esi4-qcStatement-4 QC-STATEMENT ::= { IDENTIFIED BY id-etsi-qcs-QcSSCD }
```

```
An Identifier of the statement (represented by an OID), made by the
CA, stating that the private key associated with the public key in
the certificate is stored in a Secure Signature Creation Device
according to Annex III of the EU Directive 1999/93/EC [1], as
implemented in the law of the country where the CA is established.
```

10.1 Algoritmus zostavenia certifikačnej cesty

Vstupom do algoritmu zostavovania certifikačnej cesty sú hodnoty, z ktorých niektoré môžu byť súčasťou podpisovej politiky.

Zoznam údajov a pravidiel, požadovaných pri zostavovaní certifikačnej cesty:

- **Certifikát koncovej entity**, ktorej verejným kľúčom sa overuje elektronický podpis.
- Množina používateľom vyžadovaných politík ***userInitialPolicy***,
  - ktorých prienik s položkami certifikátu *certificatePolicies* je neprázdna množina,
  - ak je v *userInitialPolicy* uvedené *anyPolicy,* akceptovateľné sú všetky politiky.
- ***validPolicySet*** množina akceptovateľných politík, ktorá vzniká ako prienik s položkami certifikátu *certificatePolicies* a ak je povolené mapovanie politík, tak je upravená prostredníctvom *policyMapping*. Na začiatku *validPolicySet* nie je inicializovaná a preberie položky z prvého certifikátu certifikačnej cesty, ktorý je overovaný koreňovým certifikátom.
- **Zoznam certifikátov**, ktoré môžu tvoriť certifikačnú cestu.
- **Zoznam CRL (OCSP)**.
- **Čas kontroly**, v ktorom sa certifikačná cesta zostavuje a kontroluje (napríklad čas najstaršej platnej časovej pečiatky podpisu).
- **Zoznam implicitne dôveryhodných koreňových CA** certifikátov vo forme *selfSigned* certifikátov, ktoré netvoria certifikačnú cestu, ale obsahujú trustAnchor (DN meno a verejný kľúč). Dôveryhodný trustAnchor nemusí byť uložený len vo forme certifikátu, ale môže sa nachádzať aj v TSL [12] ako samostatná dvojica.
- ***permittedSubtrees*** je množina mien, do ktorej musia patriť všetky mená subjektov certifikátov nasledujúcej certifikačnej cesty smerom od koreňového certifikátu. Na začiatku nie je inicializovaná a akceptuje všetky mená. Kontrolujú sa len položky obsiahnuté v *permittedSubtrees,* teda certifikát môže obsahovať ľubovoľné iné. Aplikácia musí vedieť spracovať:
  - *subject - DistinguishedName*
  - *subjectAltName.directoryName - DistinguishedName*
  - *subjectAltName.rfc822Name*
  - *subjectAltName.dNSName*
  - *subjectAltName.uniformResourceIdentifier*
  - *subjectAltName.iPAddress*
- ***excludedSubtrees*** je množina mien, do ktorej nesmú patriť všetky mená subjektov certifikátov nasledujúcej certifikačnej cesty smerom od koreňového certifikátu. Na začiatku

algoritmu je množina prázdna. Stav overenia *permittedSubtrees* nemá vplyv na overovanie *excludedSubtrees*. Kontrolujú sa len položky obsiahnuté v *excludedSubtrees,* teda certifikát môže obsahovať ľubovolné iné. Aplikácia musí vedieť spracovať:

- *subject (DistinguishedName)*
- *subjectAltName.directoryName (DistinguishedName)*
- *subjectAltName.rfc822Name*
- *subjectAltName.dNSName*
- *subjectAltName.uniformResourceIdentifier*
- *subjectAltName.iPAddress*

- ***requireExplicitPolicySkipCerts*** je hodnota, ktorá určuje, koľko certifikátov bude preskočených, než sa začne vyžadovať kontrola s *userInitialPolicy* a *validPolicySet.* Počiatočná hodnota je nastavená na dĺžku cesty + 1. Nasledujúce certifikáty môžu hodnotu len zmenšiť. Ak je hodnota v *PolicyConstraints.requireExplicitPolicy* menšia, tak na túto hodnotu. Ak je jej hodnota nulová, vyžaduje sa, aby pre aktuálny a každý ďalší certifikát v certifikačnej ceste existovala neprázdna množina z prieniku:
  - *userInitialPolicy* a zjednotenia (*policyIdentifier* z *CertificatePolicies*)
  - a *validPolicySet* a zjednotenia (*policyIdentifier* z *CertificatePolicies*).
- ***inhibitPolicyMappingSkipCerts*** je hodnota ktorá určuje, koľko certifikátov bude preskočených, než sa zakáže mapovanie politík. Počiatočná hodnota je nastavená na dĺžku cesty + 1. Nasledujúce certifikáty môžu hodnotu len zmenšiť. A ak je hodnota v *PolicyConstraints.inhibitPolicyMapping* menšia, tak na túto hodnotu.
- ***InhibitAnyPolicySkipCerts*** je hodnota ktorá určuje, koľko certifikátov bude preskočených, než sa zakáže použitie *AnyPolicy.* Počiatočná hodnota je nastavená na dĺžku cesty + 1. Nasledujúce certifikáty môžu hodnotu len zmenšiť. Ak je hodnota v *InhibitAnyPolicy* menšia, tak na túto hodnotu. Hodnota a práca s *AnyPolicy,* ak nie je v ďalších častiach spomínaná, alebo cez *InhibitAnyPolicySkipCerts* zakázaná, tak sa uplatňuje pri množinových operáciách so zjednotenou množinou OIDov z rozšírenia *CertificatePolicies.*
- ***maxPathLength*** je inicializovaný na hodnotu dĺžky certifikačnej cesty a zmenšuje sa pri každom CA certifikáte, okrem *selfSigned* certifikátu. Nasledujúce certifikáty môžu hodnotu len zmenšiť. Ak je hodnota v *BasicConstraints.pathLenConstraint* menšia, tak na túto hodnotu. *MaxPathLength* určuje, koľko CA certifikátov môže v ceste ešte nasledovať.

10.2 Vyhľadávanie certifikačných ciest pomocou rekurzívnej procedúry

Algoritmus vyhľadávania certifikačnej cesty musí zabezpečiť nájdenie certifikačnej cesty po certifikát, ktorý sa overuje implicitne „dôveryhodným koreňovým certifikátom“. Pri prehľadávaní ciest musí byť zabezpečené, aby sa algoritmus nezacyklil vďaka krížovej certifikácií jednotlivých CA certifikátov. Ak lokálna databáza neobsahuje certifikát vydavateľa, je možné ho získať z adresy uloženej v *AuthorityInformationAccess.*

Certifikačnú cestu môžeme popísať nasledovne:

- Pre všetky certifikáty C_i, kde i je z { 1, ..., n - 1 } platí, že meno subjektu certifikátu C_{i+1} je meno vydavateľa certifikátu C_i.
- Certifikát C_n, je implicitne dôveryhodný koreňový certifikát *selfSigned*, pričom pri overovaní *pathLenConstraint* - dĺžky certifikačnej cesty a *NameConstraints* sa tento certifikát neberie do úvahy, lebo nie je súčasťou certifikačnej cesty.
- Certifikát C_1, je certifikát koncovej entity, ktorý overujeme a ktorej verejný kľuč je použitý na overovanie, napríklad podpisu dokumentu alebo časovej pečiatky.

Pri overovaní podpisu sa najprv vyhľadáva certifikát koncovej entity, na základe identifikátorov: *issuer*, *serialNumber, hash certifikátu* a *SubjectKeyIdentifier.* Vhodnosť certifikátu pre požadovaný

účel sa overí na základe rozšírení certifikátu, napríklad: *KeyUsage*, *ExtendedKeyUsage, CertificatePolicies, QcStatements* alebo ďalších rozšírení podľa typu použitia certifikátu.

Ak sa podarilo získať certifikát koncovej entity C_i, s i = 1, pokračuje sa rekurzívnym algoritmom, ktorý sa snaží nájsť požadovanú certifikačnú cestu.

1. Vyhľadáva sa k certifikátu C_i certifikát C_{i+1}, ktorý spĺňa podmienky špecifikované v kapitole 9.1 a ktorý nie je v množine zakázaných certifikátov, alebo už obsiahnutý v certifikačnej ceste (okrem *selfSigned*), alebo už nájdených cestách, aby sa zabránilo rekurzívnemu zacykleniu.
2. Ak sa nepodarilo nájsť certifikát C_{i+1}, uloží sa C_i do množiny zakázaných certifikátov, zmenší sa i o 1 a ak je i = 0 algoritmus sa ukončí, inak je potrebné vyhľadávať ďalšiu cestu, napríklad cez krížové certifikáty zopakovaním od kroku 1.
3. Ak je certifikát $C_{i+1} = C_i$, tak bol nájdený koreňový certifikát.
    a. Ak je C_i implicitne dôveryhodný, tak sa certifikačná cestu po C_i odloží do zoznamu nájdených ciest, zmenší sa i o 1 a ak je i = 0, algoritmus sa ukončí, inak je potrebné nájsť ďalšiu cestu, napríklad cez krížové certifikáty, zopakovaním od kroku 1.
    b. Ak nie je C_i implicitne dôveryhodný, uloží sa C_i do množiny zakázaných certifikátov, zmenšíme i o 1 a ak je i = 0 algoritmus sa ukončí, inak je potrebné nájsť ďalšiu cestu, napríklad cez krížové certifikáty, zopakovaním od kroku 1.
4. Certifikát C_{i+1} sa uloží do aktuálnej cesty, zväčší sa i o 1 a pokračuje sa krokom 1.

Ak je certifikačná cesta vytvorená alebo kontrolovaná na základe podpisovej politiky, ktorá obsahuje zoznam dôveryhodných koreňových certifikátov, tak potom implicitne dôveryhodné koreňové certifikáty sú len tie, ktoré sa nachádzajú aj v zozname dôveryhodných koreňových certifikátov v podpisovej politike. Inak povedané, pri použití podpisovej politiky je implicitne dôveryhodný koreňový certifikát len ten, ktorému overovateľ dôveruje a je súčasne aj v podpisovej politike.
Ak zoznam nájdených ciest nie je prázdny, potom algoritmus vyberie najvhodnejšie certifikačné cesty (na základe legislatívnych alebo technologických požiadaviek ...) a overí ich algoritmom, ktorý je uvedený v nasledujúcej kapitole 10.3., inak sa algoritmus ukončí s chybou o nezostrojení cesty k implicitne dôveryhodnému koreňovému certifikátu.

10.3 Overenie nájdených certifikačných ciest

Vstupom do procesu overovania môžu byť už pripravené certifikačné cesty z elektronických podpisov alebo z predchádzajúceho algoritmu na vyhľadanie certifikačných ciest. Certifikačné cesty sa budú overovať každá samostatne.

Nultým krokom pri overovaní je zmena poradia certifikátov v overovanej certifikačnej ceste:

- Pre všetky certifikáty C_i, kde i je z { 1, ..., n - 1 } platí, že meno subjektu certifikátu C_i je meno vydavateľa certifikátu C_{i+1}.
- Certifikát C_1 je implicitne dôveryhodný koreňový certifikát *selfSigned*. Tento certifikát netvorí certifikačnú cestu.
- Certifikát C_n je certifikát koncovej entity, ktorý overujeme a ktorej verejný kľuč je použitý na overovanie, napríklad podpisu dokumentu alebo časovej pečiatky.

Pre i = 1 až n prekontroluj certifikačnú cestu s certifikátmi C_i, a ak sa niektorá z nasledujúcich podmienok nesplní, algoritmus sa ukonči s výsledkom NEPLATNÝ, inak PLATNÝ.

1. Ak i = 1, over či je C_i *selfSigned* certifikát a či je implicitne dôveryhodný a over C_i s C_i podľa postupu v kapitole 9.1., pričom pri overovaní podľa podmienok z kapitoly 9.1.sa vynechajú body 4. až 9. a v 10. bode sa iba inicializuje premenná *maxPathLength.*
2. Ak i > 1, over certifikát C_i s C_{i-1} podľa postupu v kapitole 9.1.
3. (Ak i = 1, tento bod sa nemusí realizovať.) Na základe postupu, ktorý popisuje Príloha B sa vyhľadá CRL alebo OCSP v lokálnej databáze a ak očakávané CRL alebo OCSP nie je dostupné, tak sa získa z adresy uloženej v *CRLDistributionPoints* alebo *AuthorityInformationAccess*. Over podpis CRL alebo OCSP s certifikátom C_{i-1} (ak sa jedná o nepriame CRL alebo OCSP, tak sa overenie zrealizuje cez samostatnú certifikačnú cestu) a over platnosť C_i pomocou CRL alebo OCSP.
4. Ak i je rôzne od 1, prekontroluj C_i pomocou *permittedSubtrees* podľa kapitoly 10.1.
5. Ak i je rôzne od 1, prekontroluj C_i pomocou *excludedSubtrees* podľa kapitoly 10.1.
6. Ak certifikát C_i obsahuje *NameConstraints.permittedSubtrees,* tak ulož do *permittedSubtrees* prienik z *NameConstraints.permittedSubtrees* a *permittedSubtrees.*
7. Ak certifikát C_i obsahuje *NameConstraints.excludedSubtrees,* tak ulož do *excludedSubtrees* zjednotenie z *NameConstraints.excludedSubtrees* a *excludedSubtrees.*
8. Do množiny *explicitPolicis* ulož zjednotenie OIDov z certifikátu C_i z položiek *CertificatePolicies.*
9. Ak je *requireExplicitPolicySkipCerts* = 0, tak potom:
   a. musí byť neprázdny prienik medzi *userInitialPolicy* a *explicitPolicis,*
   b. musí byť neprázdny prienik medzi *validPolicySet* a *explicitPolicis.*
10. *ValidPolicySet* nastav na hodnotu prieniku *validPolicySet* a *explicitPolicis.*
11. Ak je *inhibitPolicyMappingSkipCerts* > 0, tak potom sa pokús vytvoriť množinu *mapedPolicies* tak, že pre každú dvojicu z certifikátu C_i *PolicyMappings*(*issuerDomainPolicy*, *subjectDomainPolicy*), ktorej *issuerDomainPolicy* je v množine *validPolicySet,* pridaj do *mapedPolicies* OID *subjectDomainPolicy* a z *validPolicySet* nakoniec vymaž politiku *issuerDomainPolicy.* Nakoniec do *validPolicySet* ulož zjednotenie *validPolicySet* a *mapedPolicies*.
12. Ak je *requireExplicitPolicySkipCerts* = 0, tak potom musí byť neprázdny prienik medzi *userInitialPolicy* a *validPolicySet.*
13. Ak certifikát C_i obsahuje *PolicyConstraints.requireExplicitPolicy* a hodnota je menšia než je v *requireExplicitPolicySkipCerts,* nastav *requireExplicitPolicySkipCerts na hodnotu PolicyConstraints.requireExplicitPolicy.* Inak zmenši *requireExplicitPolicySkipCerts* o jedna.
14. Ak certifikát C_i obsahuje *PolicyConstraints.inhibitPolicyMapping* a hodnota je menšia než je v *inhibitPolicyMappingSkipCerts,* nastav *inhibitPolicyMappingSkipCerts na hodnotu PolicyConstraints.inhibitPolicyMapping.* Inak zmenši *inhibitPolicyMappingSkipCerts* o jedna.
15. Ak certifikát C_i obsahuje *InhibitAnyPolicy* a hodnota je menšia než je v *InhibitAnyPolicySkipCerts,* nastav *InhibitAnyPolicySkipCerts na hodnotu InhibitAnyPolicy.* Inak zmenši *InhibitAnyPolicySkipCerts* o jedna.
16. Ak *maxPathLength* je nulové, môže už len nasledovať certifikát koncovej entity $C_{i=n.}$
17. Ak certifikát C_i obsahuje *BasicConstraints.pathLenConstraint* a *maxPathLength* je väčšia, tak sa *maxPathLength* nastaví na *BasicConstraints.pathLenConstraint*. Inak sa *maxPathLength* zmenší o jedna.
18. Certifikát C_i nesmie obsahovať kritické neznáme rozšírenia certifikátu.
19. Pokračuje sa spracovávaním ďalšieho certifikátu, bodom 1.

Príloha A (informatívna) Overovanie pomocou OCSP s pozitívnou odpoveďou a s OCSP odpoveďou podľa RFC 2560

A.1 OCSP s pozitívnou odpoveďou na základe údajov z databázy

Blok podmienok pre overenie platnosti certifikátu pomocou OCSP založenom na databáze je prehľadný a jednoduchý, lebo databáza obsahuje potrebné údaje:

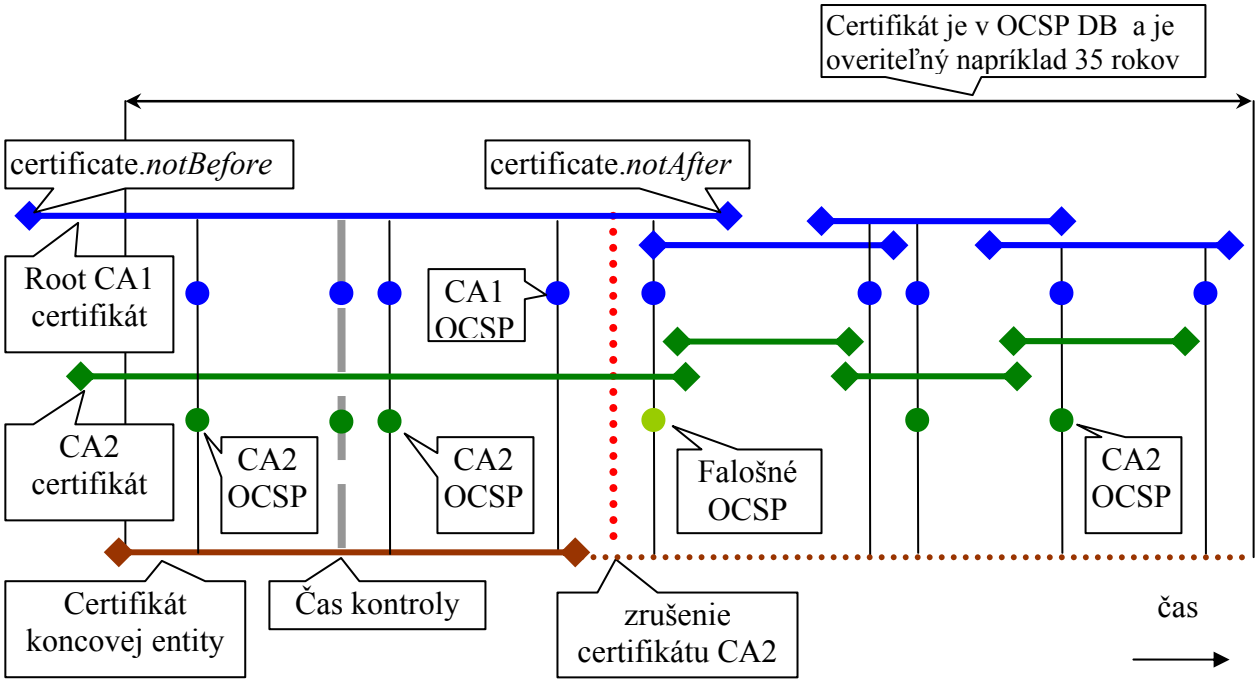
- certifikát,
- stav certifikátu,
- v prípade zrušenia certifikátu: čas a dôvod zrušenia certifikátu,

za dlhé obdobie, napríklad 35 rokov a v ľubovoľnom čase môže overovateľ požiadať o stav certifikátu bez nutnosti overenia, či overovaný certifikát exspiroval.

Hlavným rozdielom oproti OCSP podľa RFC 2560, ktoré preberá informácie o neplatnosti certifikátu zo záznamov z CRL, je pozitívna odpoveď o stave certifikátu. V rozšírení odpovede sa totiž nachádza hash hodnota z certifikátu, ktorého stav odpoveď obsahuje, teda overovateľ si je istý, že OCSP certifikát a aj jeho stav pozná.

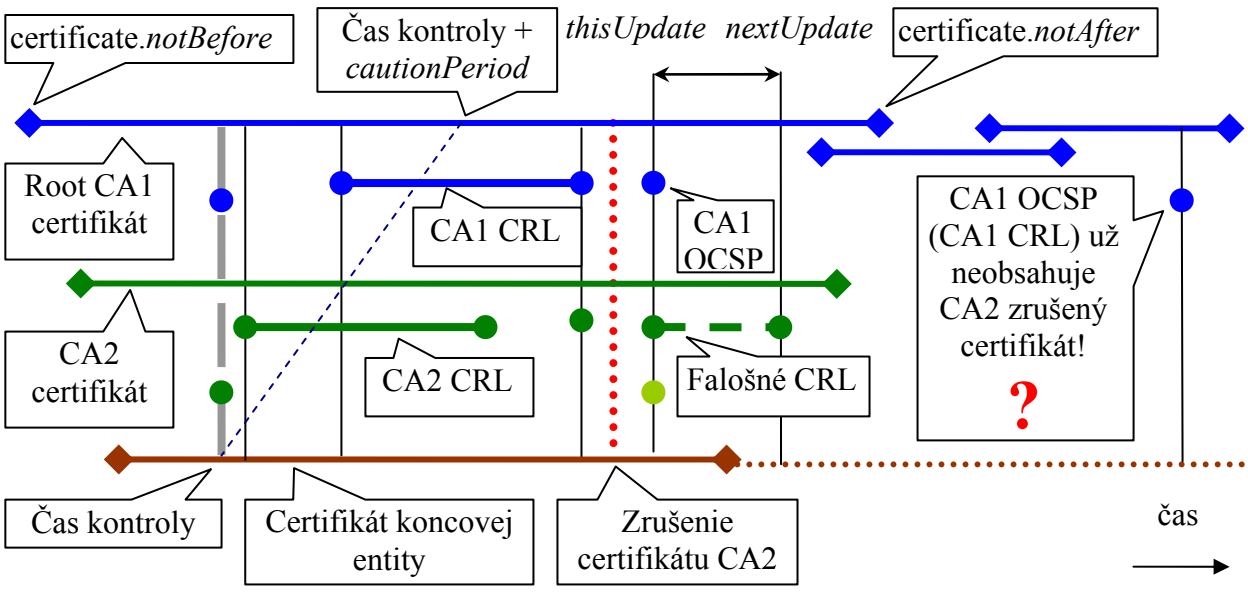
Ďalšou výhodou je, že stav platnosti exspirovaného certifikátu je možné overiť s OCSP, ktorého podpis je overovaný s certifikačnou cestou, ktorej prvý certifikát je overovaný aktuálnym platným dôveryhodným koreňovým certifikátom. Teda OCSP sa overuje k aktuálne platnému koreňovému certifikátu, aj keď stavy certifikátov, ktoré OCSP vracia, sú už dávno spolu s celou certifikačnou cestou exspirované.

OCSP, ktoré je založené na záznamoch v databáze, musí obsahovať rozšírenie *certHash* definované v ISIS-MTT Optional SigG-Profile. Rozšírenie *certHash* zabezpečuje pozitívnu odpoveď, že systém, ktorý vydáva OCSP pozná certifikát, ktorého stav vracia (bol vydaný a je v databáze).



A.2 OCSP odpoveď podľa RFC 2560

Ak je použité OCSP, ktoré číta stav platnosti certifikátu z CRL, potom je OCSP možné použiť len v čase, v ktorom sa údaje o stave certifikátu môžu nachádzať v CLR. Teda jedinou výhodou je, že ak CRL narastie do extrémnej veľkosti, tak OCSP odpoveď bude menšia, ale zasa naopak pri neustálom žiadaní stavu cez OCSP vzrastú nároky na výkon OCSP servera.



Príloha B (normatívna) Overovanie platnosti certifikátu

B.1 Overenie pomocou OCSP

Blok podmienok pre overenie platnosti certifikátu pomocou OCSP. OCSP odpoveď musí obsahovať stav, ktorý bol známy po čase certificate.*notBefore* overovaného certifikátu, alebo v čase, kedy aj po exspirovaní certifikátu sú pre OCSP dostupné informácie o stave certifikátu. Teda OCSP obsahuje *ArchiveCutoff* s hodnotou menšou ako je čas exspirovania certifikátu, alebo pozitívne prehlásenie vo forme *CertHash* v rozšírení OCSP odpovede o tom, že OCSP pozná certifikát a jeho stav.

Tabuľka 1. Overenie s OCSP

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** (ČasKontroly + *cautionPeriod*) <= *OCSP*[certificate].*thisUpdate* **then**
   PLATNÝ
4. **else**
   NEÚPLNÉ OVERENIE – získanie novej odpovede OCSP
5. **else**
   if *OCSP*[certificate].*CertStatus* = *revoked* **then**
   if ČasKontroly < *OCSP*[certificate].*revocationTime* **then**
   PLATNÝ
6. **else**
   NEPLATNÝ
7. **else**
   NEÚPLNÉ AUTOMATICKÉ OVERENIE – OCSP nepozná aktuálny stav certifikátu, lebo *OCSP*[certifikát].*CertStatus* = *unknown*
   Je potrebné získať OCSP z inej adresy alebo overovať na základe CRL.
8. **else**
   NEÚPLNÉ AUTOMATICKÉ OVERENIE – žiadosť na CA o CRL alebo OCSP vydané v čase platnosti certifikátu + časový interval, v ktorom je ešte prítomný záznam o zrušení certifikátu v CRL alebo OCSP.

Kde:

- OCSP.ArchiveCutoff ak nie je v OCSP odpovedi, tak má hodnotu 0, inak podľa RFC 2560.
- OCSP[certifikát].*CertHash* je hash certifikátu, ktorého stav OCSP vracia (ISIS-MTT private extensions), ak sa toto rozšírenie v OCSP nachádza, predstavuje pozitívnu informáciu, že OCSP pozná certifikát a stav overovaného certifikátu.
- Certifikát.*CertHash* je hash certifikátu, ktorého platnosť sa overuje.
- OCSP.*producedAt* je čas vydania OCSP.
- *OCSP*[certificate].*thisUpdate* je čas, ku ktorému boli známe korektné informácie o stave certifikátu.
- Certifikát.*notBefore* je čas začiatku platnosti overovaného certifikátu.
- Certifikát.*notAfter* je čas, po ktorom certifikát exspiruje.

- OCSP[certifikát].*revocationTime* je dátum zrušenia certifikátu.
- OCSP[certifikát].*CertStatus* je stav certifikátu v OCSP, ktorý môže mať len 3 hodnoty.

Vysvetlenia k blokom podmienky:

1. OCSP je vydané v čase platnosti certifikátu + časový interval, počas ktorého je záznam o zrušení certifikátu pre OCSP známy aj po exspirovaní certifikátu,
2. certifikát nebol zrušený, nie je v OCSP,
3. stav certifikátu v OCSP je známy po čase kontroly, pričom pri overovaní certifikátu vydaného podľa pravidiel slovenskej legislatívy (OID certifikačnej politiky 1.3.158.36061701.0.0.0.1.2.2) sa môže nastaviť *cautionPeriod* na hodnotu nula a výsledok overenia bude zhodný, ako keby sa ešte čakalo *cautionPeriod,*
4. stav certifikátu v OCSP nie je známy po čase kontroly a treba požiadať o nové OCSP,
5. certifikát bol zrušený po čase kontroly, teda je platný,
6. certifikát je zrušený v OCSP pred časom kontroly,
7. OCSP nevie určiť stav certifikátu, treba skúsiť iné OCSP alebo CRL,
8. je potrebné získať OCSP alebo CRL vydané v čase, kedy certifikát ešte neexspiruje + časový interval v ktorom je stav certifikátu v OCSP alebo CRL ešte známy a je vydané po čase kontroly.

B.2 Overenie pomocou CRL

Blok podmienok pre overenie platnosti certifikátu pomocou CRL. CRL odpoveď musí obsahovať stav, ktorý bol známy po čase certificate.*notBefore* overovaného certifikátu, alebo v čase, kedy aj po exspirovaní certifikátu sú v CRL dostupné informácie o stave certifikátu. Teda CRL obsahuje *expiredCertsOnCRL* s hodnotou menšou ako je čas exspirovania certifikátu.

Tabuľka 2. Overenie s CRL

1. **if** (certifikát.*notBefore* < *CRL.thisUpdate*) **and**
   ((CRL.*expiredCertsOnCRL* <= certifikát.*notAfter*) **and** (0 < CRL.*expiredCertsOnCRL*) **or**
   (*CRL.thisUpdate* <= certifikát.*notAfter*) **and** (0 = CRL.*expiredCertsOnCRL*)) **then**
2. **if** certifikát **is not in** *CRL* **then**
3. **If** (ČasKontroly + *cautionPeriod*) <= *CRL.thisUpdate* **then**
   PLATNÝ
4. **else**
   NEÚPLNÉ OVERENIE – čakanie na nové CRL
5. **else**
   if ČasKontroly < *CRL*[certifikát].*revocationDate* **then**
   PLATNÝ
6. **else**
   NEPLATNÝ
7. **else**
   NEÚPLNÉ AUTOMATICKÉ OVERENIE – žiadosť na CA o CRL vydané v čase platnosti certifikátu + časový interval, v ktorom je ešte prítomný záznam o zrušení certifikátu v CRL.

Kde:

- CRL.*expiredCertsOnCRL* ak nie je v CRL rozšírení, tak má hodnotu 0, inak podľa ITU-T Rec. X.509(08/2005) [10].
- *CRL.thisUpdate* je čas, ku ktorému boli známe korektné informácie o stave certifikátu.
- Certifikát.*notBefore* je čas začiatku platnosti overovaného certifikátu.
- Certifikát.*notAfter* je čas, po ktorom certifikát exspiruje.
- *CRL*[certifikát].*revocationDate* je dátum zrušenia certifikátu v CRL.

Vysvetlenia k blokom podmienky:

1. CRL je vydané v čase platnosti certifikátu + časový interval, počas ktorého je záznam o zrušení certifikátu v CRL známy aj po exspirovaní certifikátu.
2. Certifikát nebol zrušený, nie je v CRL.
3. Stav certifikátu v CRL je známy po čase kontroly, pričom pri overovaní certifikátu vydaného podľa pravidiel slovenskej legislatívy (OID certifikačnej politiky 1.3.158.36061701.0.0.0.1.2.2) sa môže nastaviť *cautionPeriod* na hodnotu nula a výsledok overenia bude zhodný, ako keby sa ešte čakalo *cautionPeriod.*
4. CRL nie je vydané po čase kontroly a treba čakať na nové CRL.
5. Certifikát bol zrušený po čase kontroly, teda je platný.
6. Certifikát je zrušený v CRL pred časom kontroly.
7. Je potrebné získať CRL vydané v čase, kedy certifikát ešte neexspiruje + časový interval v ktorom je stav certifikátu v CRL ešte známy a je vydané po čase kontroly.

Príloha C (informatívna) Revízie vykonané od predošlého vydania

C.1 Pridané požiadavky

Pridané položky, ktoré významne zmenili význam predchádzajúcich požiadaviek:

V prílohe B.2 bolo pridané overenie na základe CRL.*expiredCertsOnCRL*.

Záver kapitoly 7 bol presunutý a doplnený do kapitoly 8, ktorá definuje postupy pri archivovaní.

C.2 Upravené požiadavky

Položky, ktoré upravujú predchádzajúce požiadavky:

Kapitola 3 sa presunula do kapitoly 5 a jednoznačne sa zadefinovalo, že koreňový certifikát nie je súčasťou certifikačnej cesty.

V prílohe B sa zlúčili požiadavky z kapitoly 5.

C.3 Vysvetlenia

Položky, ktoré boli zmenené pre vysvetlenie predchádzajúcich požiadaviek:

Definícia *času kontroly* sa doplnila o ďalšie spresnenia.

Vzťah medzi *grace period* a *caution period* sa jednoznačne zadefinoval.

V kapitolách 9 a 10 boli doplnené podrobnejšie vysvetlenia.

C.4 Publikačné zmeny

Zmeny, ktoré neovplyvňujú technický význam dokumentu:

Úvod z kapitoly 5 bol zredukovaný a presunutý do prílohy A.

Kapitola 6 sa odstránila.

Číslovanie od kapitoly 8 sa zväčšilo o jedna.

Príloha D (informatívna) Zoznam použitej literatúry

Základné dokumenty legislatívy Slovenskej republiky pre elektronický podpis
http://www.nbusr.sk/sk/elektronicky-podpis/legislativa/index.html

Formáty zaručených elektronických podpisov
http://www.nbusr.sk/sk/elektronicky-podpis/schvalene-formaty/index.html

ETSI TS 102 176-1: "Electronic Signatures and Infrastructures (ESI); Algorithms and Parameters for Secure Electronic Signatures; Part 1: Hash functions and asymmetric algorithms".

ETSI TS 101 903 "XML Advanced Electronic Signatures (XAdES)."

ETSI TR 102 041 "Signature Policies Report"

ETSI TS 102 231 V2.1.1 (2006-03) Electronic Signatures and Infrastructures (ESI); Provision of harmonized Trust-service status information

US Secure Hash Algorithms (SHA and HMAC-SHA)
http://www.rfc-archive.org/getrfc.php?rfc=4634

Internet X.509 Public Key Infrastructure: Certification Path Building
http://www.rfc-archive.org/getrfc.php?rfc=4158

NIST testovacie sady na overenie certifikačnej cesty
http://csrc.nist.gov/pki/testing/x509paths.html
http://csrc.nist.gov/pki/testing/pathdiscovery.html

Príloha E História

Verzia:	Dátum vydania:	Poznámka:	Vypracoval:
Verzia 1.1	29.9.2005	Prvé vydanie, zrušené	Ing. Peter Rybár, NBÚ
Verzia 1.2	16.10.2005	Pridanie OCSP	Ing. Peter Rybár, NBÚ
Verzia 1.2.1	24.11.2005	Zjednodušenie podmienok	Ing. Peter Rybár, NBÚ
Verzia 1.3 Č.: 1891/2006/IBEP-001	19.3.2006	Spresnenie OCSP, overenie TSP a mapovanie politík	Ing. Peter Rybár, NBÚ
Verzia 1.4 Č.: 1891/2006/IBEP-008	19.11.2006	Doplnenie a sprehľadnenie podmienok pre zabezpečenie interoperability	Ing. Peter Rybár, NBÚ