

OCSP Responder s konsolidovanou databází CRL

Marek Šándor



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2012/2013

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Marek ŠÁNDOR**
Osobní číslo: **A10073**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**
Forma studia: **kombinovaná**

/

Téma práce: **OCSP responder s konsolidovanou databází CRL.**

Zásady pro vypracování:

- 1. Prostudujte protokol OCSP (Online Certificate Status Protocol) a jeho implementace v open source projektech OpenCA a EJBCA.**
- 2. Analyzujte možnost využití uvedených projektů při implementaci OCSP responderu s konsolidovanou databází CRL českých certifikačních autorit.**
- 3. Implementujte OCSP responder s konsolidovanou databází CRL.**
- 4. V jazyce Java implementujte vzorovou aplikaci, využívající OCSP responder k rychlému ověření revokovaných klientských certifikátů.**
- 5. Proveďte analýzu možných bezpečnostních rizik výsledného systému.**

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. ZAKHOUR, Sharon et al. Java 6: výukový kurz. Vyd. 1. Brno: Computer Press, 2007. 534 s. ISBN 978-80-251-1575-6.
2. MYERS, M. et al X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OSCP RFC 2560. [online]. June 1999 [cit. 2013-02-06]. Dostupné z: <http://datatracker.ietf.org/doc/rfc2560/>
3. DOSTÁLEK, Libor, VOHNOUTOVÁ, Marta a KNOTEK, Miroslav. Velký průvodce infrastrukturou PKI a technologií elektronického podpisu. 2., aktualiz. vyd. Brno: Computer Press, 2009. 542 s. ISBN 978-80-251-2619-6.
4. DAVIES, Joshua A. Implementing SSL/TLS using cryptography and PKI. Hoboken, N.J.: Wiley, 2011, xxxii, 663 p. ISBN 978-111-8038-772.
5. DEACON, A. a R. HURST. The Lightweight Online Certificate Status Protocol (OCSP) Profile for High-Volume Environments RFC 5019. [online]. September 2007. [cit. 2013-02-07]. Dostupné z: <http://http://datatracker.ietf.org/doc/rfc5019/>
6. OSCP Responder. [online]. 2011 [cit. 2013-02-06]. Dostupné z: <http://www.openca.org/projects/ocspd/>
7. EJBCA [online]. 2012 [cit. 2013-02-07]. Dostupné z: <http://www.ejbca.org/>
8. Vyhláška 212/2012: vyhláška o ověřování platnosti zaručeného elektronického podpisu. In: Sbírka zákonů, částka 75. 2012.

Vedoucí bakalářské práce:

Ing. Tomáš Dulík, Ph.D.

Ústav informatiky a umělé inteligence

Datum zadání bakalářské práce:

24. února 2013


Termín odevzdání bakalářské práce:

14. června 2013

Ve Zlíně dne 24. února 2013


prof. Ing. Vladimír Vašek, CSc.
děkan




prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona; beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo –bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

ABSTRAKT

V předložené práci popisujeme problematiku vytvoření služby pro „online“ ověřování platnosti certifikátů „OCSP Responderu“ ve vztahu k českým akreditovaným certifikačním autoritám. Navrhovaný OCSP Responder by měl být vhodný pro nasazení v interní infrastruktuře naší organizace a centralizovaně zajišťovat službu OCSP pro ověření stavu platnosti přihlašovacích certifikátů (vydaných těmito autoritami) využívaných v našich aplikacích. Zaměříme se na využití open source projektů nabízejících modul OCSP Responderu a navrhujeme ukázkovou aplikaci využívající tuto službu.

Klíčová slova: OCSP, CRL, X.509, Digitální certifikát, Revokace

ABSTRACT

In the present thesis we describe the problem of creating services for online validation of certificates "OCSP Responder" in relation to the Czech accredited Certification Authorities. The proposed OCSP Responder should be suitable for use in internal infrastructure of our organization and provide a centralized service for OCSP status verification of certificates (issued by the authorities) used in our applications for authentication. We will focus on the use of Open Source projects offering OCSP responder as a module and propose a sample application using the service.

Keywords: OCSP, CRL, X.509, Digital Certificate, Revocation

Chtěl bych poděkovat vedoucímu bakalářské práce, panu Ing. Tomáši Dulíkovi, Ph.D. za veškerou pomoc a podporu při zpracovávání zadaného tématu stejně tak jako za vědomosti, které nám trpělivě předával během celého studia formou poutavou a obsahem postihujícím současné trendy a dění v IT. Dále bych chtěl poděkovat mé ženě Lucii a dětem Lucince a Elišce za nesmírnou trpělivost a podporu během psaní této práce.

Obsah

ÚVOD	9
I TEORETICKÁ ČÁST	9
1 POTŘEBA VLASTNÍHO ŘEŠENÍ	11
1.1 AKREDITOVANÉ CERTIFIKAČNÍ AUTORITY V ČR	12
2 PROTOKOL OCSP	14
2.1 PLATNOST CERTIFIKÁTU A JEHO ODVOLÁNÍ	14
2.2 ASN.1 - JAZYK PRO POPIS DATOVÝCH STRUKTUR	14
2.3 PŘÍKLAD DEFINICE CRL Z RFC-5280 [5]	15
2.4 KÓDOVÁNÍ BER, DER, BASE64	15
2.5 CRL	16
2.6 OCSP JAKO SLUŽBA PRO ONLINE OVĚŘENÍ STAVU CERTIFIKÁTU .	17
2.7 POŽADAVEK (REQUEST) A ODPOVĚĎ (RESPONSE)	19
3 OPEN SOURCE PROJEKTY IMPLEMENTUJÍCÍ OCSP RESPON-	
DERY	21
3.1 OPENCA	21
3.2 EJBCA	22
II PRAKTICKÁ ČÁST	23
4 IMPLEMENTACE OCSP RESPONDERU S KONSOLIDOVANOU	
DATABÁZÍ CRL	24
4.1 SPUŠTĚNÍ OCSP RESPONDERU	28
4.2 TEST OCSP RESPONDERU	29
4.2.1 Nalezení revokovaného certifikátu	29
4.2.2 Kontrola platnosti certifikátu přes webové rozhraní	30
4.2.3 Test volání s openssl	31
4.2.4 Test s OCSP Responderem PostSignum	31
4.3 OCSP POŽADAVEK PODROBNĚ	34
4.3.1 OCSP požadavek metodou HTTP GET	36
4.4 DALŠÍ MOŽNÁ VYLEPŠENÍ	37
5 VZOROVÁ APLIKACE S VYUŽITÍM SLUŽEB OCSP RESPONDERU	39
6 BEZPEČNOSTÍ ASPEKTY ŘEŠENÍ OCSP RESPONDERU	41
ZÁVĚR	42
SEZNAM POUŽITÉ LITERATURY	44
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	45

SEZNAM OBRÁZKŮ	46
SEZNAM TABULEK	47

ÚVOD

V průběhu celého života se dostáváme do situací, kdy si klademe otázky a hledáme na ně odpovědi. Je jedno v jakém je to věku a čeho se týkají. Co však mají společného, je úvaha předcházející náš další krok při hledání odpovědi. Tou úvahou je, kudy se vydáme, kdo je ten, komu důvěřujeme natolik, že se na něj obrátíme abychom u něj našli odpověď. Podle povahy kladené otázky a očekávané odpovědi to může být rodič, učitel, přítel, kniha...

V hmatatelném světě kolem nás umíme jasně rozlišit, kdo se ptá, na co a jak se ptá a koho, nebo čeho se ptá. Zloděj chystající se vyloupit banku se nepůjde zeptat před její vchod ochranky, jaký je kód k zabezpečení vstupu a už vůbec tuto otázku nevykřičí z tramvajové zastávky na druhé straně ulice. Předchozí věta se jeví, jako „tak trochu“ přitažená za vlasy, není-liž pravda ... ale teď, zkusme se zamyslet sami nad sebou, kolikrát jsme křičeli do „drátů“ našeho počítače všelijaké otázky, bez uvědomění si toho, koho se ptáme, kdo nás u toho vidí, kdo nás u toho slyší.

Člověk se prý od jiných živočichů odlišuje rozumem, zachovejme si tedy alespoň zdravý selský rozum nejen ve světě, kterého se můžeme dotknout, ale i v tom pouze zdánlivě anonymním a nehmatatelném...

Cílem předkládaného textu je podat praktický a ucelený návod k vybudování služby OCSP Responderu integrovatelného do vlastní infrastruktury a podporujícího akreditované certifikační autority v ČR . V teoretické části se nejprve zamyslíme nad důvody, které nás mohou vést k rozhodnutí vybudovat vlastní řešení a dále se seznámíme s nezbytným minimem informací tak, abychom v praktické části mohli implementovat funkční OCSP Responder poskytující konsolidovanou informaci o revokovaných certifikátech.

I. TEORETICKÁ ČÁST

1 POTŘEBA VLASTNÍHO ŘEŠENÍ

Nutnost ověření platnosti certifikátu vyplývá z povahy jeho použití. V případě certifikátu použitého pro přihlašovací účely (autentizaci) bychom dozajista nechtěli, aby se po uplynutí doby jeho platnosti byl někdo schopen tímto certifikátem dále přihlašovat. Ověření časové platnosti je jen jednou z povinností (tou jednodušší), kterou máme jako spoléhající se strana při zpracování certifikátu. Tou další je zjištění, zda nebyl certifikát z nějakého důvodu odvolán. Tuto povinnost nám ukládají vydávající certifikační autority (CA) ve svých „Certifikačních politikách“, jak si ukážeme v kapitole 2.1 na straně 14.

„S odvoláním certifikátu je to podobné, jako s odvoláním platební karty. Nejprve nám platební kartu někdo odcizí, aniž bychom si toho všimli. Jakmile to zjistíme, nahlásíme ztrátu karty jejímu vydavateli. Vydavatel tuto informaci zpracuje a výsledkem je, že se karta dostane na stop list.

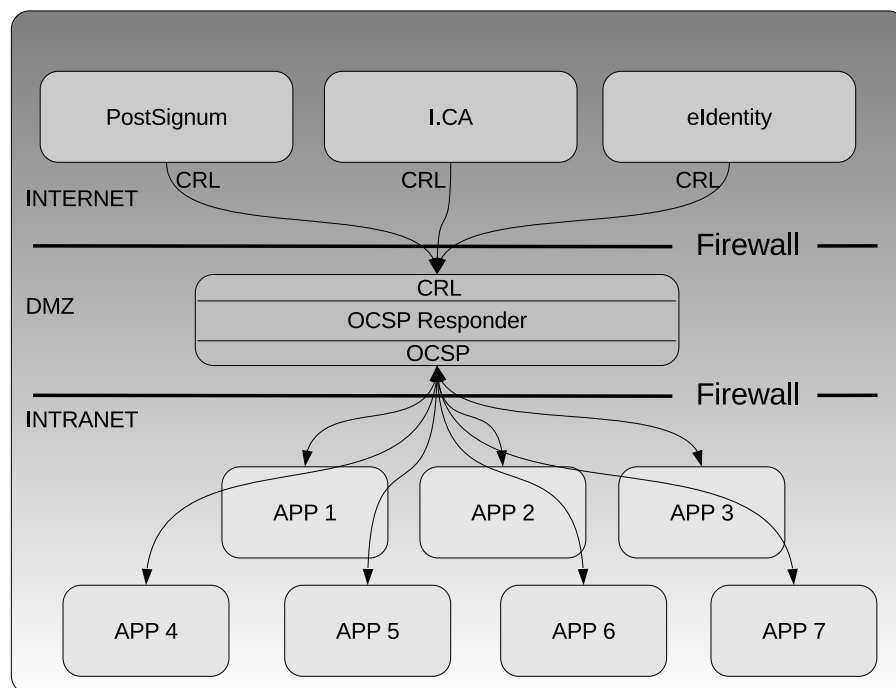
Každý obchodník je povinen si ověřit (před tím, než přijme platbu prostřednictvím karty), zdali karta není na stoplistu (automaticky to provádí platební terminál). Pokud si to neověří a karta na stop listu je, půjdou náklady na jeho vrub. Veškeré diskuze jsou jen o tom, kdo ručí za kartu mezi jejím odcizením a okamžikem, kdy se dostane na stop list. Toto mezidobí může vzít na sebe vydavatel karty jako službu držiteli karty, karta se může pojistit apod.“ [1, kap. 5]

Obdobně, jako je to v předchozím příkladu i naše aplikace musí platnost certifikátu ověřit (tak jako u příkladu platební karty). U platební karty její držitel kontaktuje banku a ta kartu umístí na stop list. Certifikační autorita, po přijetí požadavku na odvolání certifikátu od jeho držitele, jej umístí na seznam revokovaných certifikátů (dále jen CRL) ¹⁾.

V rámci organizace máme zpravidla více aplikací, které akceptují certifikáty a je tedy nezbytné, aby prováděli ověřování platnosti. Přenesením problematiky udržování konzistentní konsolidované informace o platnosti (odvolání) certifikátu z jednotlivých aplikací na centralizovanou interní službu sjednotíme postupy, jakým tuto platnost zajišťujeme. Takovéto řešení máme navíc plně pod kontrolou, což je z bezpečnostního hlediska žádoucí. V dalších kapitolách rozebereme podrobněji postupy a podmínky zveřejňování CRL na CA a způsob jakým informaci o stavu platnosti certifikátu zpřístupníme „online“ s využitím OCSP ²⁾. Typický příklad je na obrázku 1 na následující straně.

¹⁾Z anglického Certificate Revocation List.

²⁾Z anglického Online Certificate Status Protocol.



Obrázek 1: OSCP Responder

1.1 Akreditované certifikační autority v ČR

V ČR je používání elektronického podpisu a s tím i souvisejících certifikátů upraveno zákonem č. 227/2000 Sb.³⁾ vycházejícího ze směrnice EU. Zákon v zásadě „pouze“ volnou formou popisuje, jaká jsou práva a povinnosti, ale nedozvím se z něj informace potřebné k vybudování samotného technického řešení. Zde se budeme muset obrátit na dokumenty X.509, RFC-2560 (OCSP), RFC-5280 (CRL) a další s nimi související.

„... Dne 15. dubna 2010 nabyla účinnosti novela zákona o elektronickém podpisu (č. 101/2010 Sb.). Tento předpis v reakci na komitologické rozhodnutí 2009/767/ES přidává Ministerstvu vnitra povinnost vést a zveřejňovat seznam důvěryhodných certifikačních služeb a stanoví orgánům veřejné moci povinnost uznávat kvalifikované certifikáty vydané v ostatních členských státech EU. ...“ [10].

České prostředí nám nabízí tři akreditované certifikační autority (dále jen CA)

- I.CA - První certifikační autorita, a. s. - <http://www.ica.cz/>,
- Certifikační autorita PostSignum - <http://www.postsignum.cz/>,
- eIdentity a.s. - <http://www.eidentity.cz/>.

Pokud budujeme aplikaci, která má být přístupná široké veřejnosti, nebo pro státní

³⁾Zákona č. 227/2000 Sb. o elektronickém podpisu s posledními změnami provedenými zákonem č. 167/2012 Sb.

správu a zároveň nechceme nebo nemůžeme budovat vlastní Certifikační Autoritu ⁴⁾ pravděpodobně dostaneme v zadání požadavek na akceptaci certifikátů právě od těchto českých akreditovaných CA.

Uvedené certifikační autority nabízí certifikáty pro účely zaručeného elektronického podpisu (pro osoby) nebo elektronické značky (pro systémy), označované jako „kvalifikované“ a jejich použití je vymezeno zákonem. ⁵⁾ Jedno z omezení je, že nesmějí být použity pro autentizační účely a proto certifikační autority nabízí i tzv. „komerční“ certifikáty, které již pro autentizaci použít lze. Technicky jde však v obou případech o standardní digitální certifikát.

Nadále bude v této práci (pokud nebude uvedeno jinak) uvažováno o „komerčním“ certifikátu použitého pro přihlašovací účely (autentizaci) vydaného jednou z těchto tří akreditovaných CA.

⁴⁾Tak jako to dělají například banky

⁵⁾Související RFC je 3739 „Internet X.509 Public Key Infrastructure Qualified Certificates Profile“

2 PROTOKOL OCSP

Dříve, než se dostaneme k samotnému protokolu OCSP, seznámíme se ve stručnosti s některými související pojmy, jako je odvolání certifikátu nebo jazyk pro popis datových struktur ASN.1 a kódování BER/DER a Base64.

2.1 Platnost certifikátu a jeho odvolání

Platnost certifikátu z pohledu času je uvedena přímo v jeho těle a specifikuje ji pole `notAfter`. U našich akreditovaných CA je zpravidla nastavena na jeden rok [7] a vychází z podmínek uvedených v dokumentu „Certifikační politika“ [2] dané CA. Další možností, jak může dojít k zneplatnění je podáním žádosti o odvolání jeho platnosti. Technicky je možné aby byla platnost certifikátu i pozastavena, ale tuto službu naše CA neposkytují [2, str. 33, kap. 4.9.13].

Zneplatněné certifikáty jsou zveřejňovány na seznamu zneplatněných certifikátů CRL v pravidelných časových intervalech. V certifikační politice je uvedeno, jakým způsobem jsou tyto seznamy poskytovány [2, (str. 33, kap. 4.10.1)] i v jakých časových intervalech.

Příklad z certifikační politiky I.CA [2]

4.10 Služby související s ověřováním statutu certifikátu

4.10.1 Funkční charakteristiky

Seznamy veřejných certifikátů jsou poskytovány formou zveřejňování informací, seznamy zneplatněných certifikátů jsou poskytovány jak formou zveřejňování informací, tak uvedením distribučních míst CRL ve vydaném certifikátu.

4.10.2 Dostupnost služeb

Služba poskytování veřejných certifikátů formou zveřejňování informací je dostupná 7 dní v týdnu 24 hodin denně. I.CA garantuje zajištění nepřetržité dostupnosti (7dní v týdnu 24 hodin denně) a integrity seznamu zneplatněných certifikátů (platné CRL).

2.2 ASN.1 - jazyk pro popis datových struktur

Jazyk ASN.1¹⁾ se používá pro přesný popis datových struktur v podobě srozumitelné pro člověka. Definuje **typ** jako pojmenovanou množinu hodnot a **identifikátor**, který můžeme přirovnat k pojmu proměnná známou z programovacích jazyků. Identifikátor začíná vždy malým písmenem a typ velkým. RFC týkající se CRL a OCSP popisují datové struktury právě v jazyce ASN.1. Uveďme si příklad definice struktury CRL.

¹⁾Z anglického Abstract Syntax Notation One

2.3 Příklad definice CRL z RFC-5280 [5]

```
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
}
```

2.4 Kódování BER, DER, Base64

ASN.1 je sice čitelné pro člověka, ale nehodí se pro samotnou počítačovou komunikaci. Zde je požadavkem, aby byla struktura nezávislá na architektuře procesoru. Proto se provádí kódování do BER ²⁾, specifikujiícího jednotlivé bity přenášených bajtů. DER ³⁾ kódování je zjednodušující a upřesňující podmnožinou kódování BER, která zajišťuje, že zdrojová data budou zakódována vždy stejně. Oproti tomu v BER je možné provést kódování jedné a té samé informace vícero způsoby a všechny budou platné. Jak u certifikátů, tak i u CRL a OCSP, záleží na každém bitu stejně jako v kryptografii a proto je i v samotném RFC-5280 požadováno, aby data pro podpis byla v ASN.1 a kódována DER. [5, (kap. 5.1)]

Údaj kódovaný v BER/DER se skládá ze tří polí:[1, s. 176]

- typ dat,
- délka dat,
- data (toto pole je nepovinné).

Názorně si kódování typu INTEGER ukážeme v kap. 4.3.

²⁾Basic Encoding Rules

³⁾Distinguished Encoding Rules

Tabulka 1: Přehled univerzálních tágů

Typ	Tág desítkově	Tág šest- náctkově	Význam
END OF CONTENTS (EOC)	0	0	Konce pole dat v kódování BER pro případ, že data jsou nedefinované délky. Kódování DER EOC nepoužívá, protože DER nepoužívá nedefinovanou délku.
BOOLEAN	1	1	Nabývá hodnot TRUE a FALSE.
INTEGER	2	2	Celá kladná i záporná čísla a nula.
BIT STRING	3	3	Řetězec bitů - může být zadán binárně nebo hexadecimálně.
OCTET STRING	4	4	Řetězec bajtů (blíže nespecifikovaný).
NULL	5	5	Prázdný typ.
OBJECT IDENTIFIER	6	6	Řetězec bajtů (blíže nespecifikovaný).

Tabulka znázorňuje pouze prvních 7 univerzálních tágů, ale celkově je jich 24. [1, s. 178]

Data kódovaná v DER jsou ale binární, nezobrazitelná na terminálu a při komunikaci mezi jednotlivými systémy může docházet k problémům. Proto se zavádí ještě další typ kódování - Base64, které je sedmibitové ⁴⁾

2.5 CRL

Po odvolání certifikátu uvede CA jeho sériové číslo na seznamu revokovaných certifikátů - CRL.

Certificate Revocation List je specifikován v RFC-5280. ⁵⁾Jednotlivé typy CRL mohou být: [1, (kap. 16)]

- Úplné CRL - obsahuje seznam všech odvolaných certifikátů, jejichž původní platnost dosud nevypršela.

⁴⁾ČSN ISO/IEC 646 (36 9104) Informační technika. 7-bitový kódovaný soubor znaků ISO pro výměnu informací Těž ISO 7.

⁵⁾Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile - <http://www.ietf.org/rfc/rfc5280.txt>

- Rozdílové CRL - obsahuje seznam všech certifikátů, jejichž původní platnost dosud nevypršela a které byly odvolány od vydání posledního úplného CRL.
- Omezené CRL - obsahuje specifickou podmnožinu CRL, přičemž tato podmnožina je omezena důvodem odvolání nebo typem certifikátu.

CRL obsahuje sériová čísla všech odvolaných certifikátů (může být i prázdný), kterým ještě neskončila původní platnost (`notAfter`). Při konfiguraci našeho OCSP Responderu budeme používat CRL úplné, zveřejňované na URI uvedené v samotném certifikátu (CRL - X509v3 CRL Distribution Points).

```
[... zacatek certifikatu vynechan ...]
X509v3 extensions:
  509v3 Subject Key Identifier:
    E9:EF:BA:FE:4F:68:DC:47:EF:77:03:43:D4:0A:63:EB:3C:2E:C7:C1
  X509v3 Certificate Policies:
    Policy: 1.2.203.27112489.1.100.1.2.2
           CPS: http://www.ccaeid.cz/cca2/cp-cc.pdf

  X509v3 Basic Constraints:
    CA:FALSE
  X509v3 Authority Key Identifier:
    keyid:E5:40:2B:62:63:25:E5:90:E3:00:03:28:2A:50:17:71:F9:61:DE:19

  X509v3 CRL Distribution Points:

    Full Name:
      URI:http://www.ccaeid.cz/cca2/crl/actual.crl

    Full Name:
      URI:http://pub1.ccaeid.cz/cca2/crl/actual.crl

    Full Name:
      URI:http://pub2.ccaeid.cz/cca2/crl/actual.crl
  X509v3 Subject Alternative Name:
    email:marek.sandor@gmail.com
  X509v3 Key Usage: critical
    Digital Signature, Non Repudiation, Key Encipherment, Data Encipherment
[... pokračování certifikatu ...]
```

2.6 OCSP jako služba pro online ověření stavu certifikátu

Ověřit, zda nebyl certifikát odvolán (revokován) můžeme v zásadě dvojím způsobem. Porovnáním sériového čísla certifikátu s offline seznamem odvolaných certifikátů (CRL) a nebo s využitím online služby OCSP. Jak jsme již uvedli v kapitole 2.1 záleží na CA, kdy zveřejní nový CRL a tuto informaci popisuje ve své certifikační politice. V současnosti je u všech tří akreditovaných CA zveřejněn nový CRL prakticky do několika málo minut od zneplatnění certifikátu, ale CA uvádí povinnost zveřejnit nové CRL do 24 hodin. Jako příklad uvedeme opět část z certifikační politiky I.CA.

Příklad z certifikační politiky I.CA [2]**4.9.5 Maximální doba, za kterou musí poskytovatel realizovat požadavek na zneplatnění certifikátu**

Reakcí I.CA na přijetí platné žádosti o zneplatnění certifikátu je jeho neprodlené zneplatnění. Do doby zveřejnění seznamu zneplatněných certifikátů je dotýčný certifikát zablokován. Po dobu zablokování je certifikát platný a případná odpovědnost za škodu vzniklou použitím takového certifikátu v době jeho zablokování nelze nárokovat na I.CA.

Maximální prodlení mezi zneplatněním certifikátu a zveřejněním seznamu zneplatněných certifikátů, na kterém je tento certifikát poprvé uveden, je nejvýše 24 hodin. Odblokování certifikátu, který byl zablokován na základě platné žádosti o jeho zneplatnění, I.CA nepovoluje.

4.9.6 Povinnosti spoléhajících se stran při ověřování, zda nebyl certifikát zneplatněn

Spoléhající se strany jsou povinny provádět veškeré úkony potřebné k tomu, aby si ověřily, že elektronické podpisy jsou platné a jim odpovídající certifikáty nebyly zneplatněny. Pro tyto účely jsou spoléhající se strany povinny používat CRL, vydaná a elektronicky podepsaná I.CA. Déle platí ustanovení kapitoly 4.5.2.

4.9.7 Periodicita vydávání seznamu zneplatněných certifikátů

Seznam zneplatněných certifikátů je společností První certifikační autorita, a.s. vydáván neprodleně po úspěšném přijetí požadavku na zneplatnění certifikátu, nejvýše do 24 hodin od vydání předchozího CRL.

4.9.8 Maximální zpoždění při vydávání seznamu zneplatněných certifikátů

V procesu vydávání CRL je vždy dodrženo ustanovení kapitoly 4.9.5.

4.9.9 Možnost ověřování statutu certifikátu on-line („dále OCSP“)

Služba může být poskytována smluvním partnerům za specifických podmínek.

4.9.10 Požadavky při ověřování statutu certifikátu na on-line

Viz kapitola 4.9.9.

Jak je vidět službu OCSP I.CA poskytuje smluvním partnerům za specifických podmínek, což v překladu znamená, že tuto službu poskytuje za úplatu a po uzavření smlouvy je každý dotaz řádně zpoplatněn. Možnost nastavit si „online“ ověření certifikátu od I.CA v našem internetovém prohlížeči je hudbnou daleké budoucnosti (stejně tak jako ale i pro ostatní CA v ČR).

PostSignum na svých stránkách zveřejnilo dne 17. 12. 2012 informaci o tom, že uvádí službu OCSP do pilotního provozu. [http://www.postsignum.cz/novinky_postsignum.html#N97] Po kontaktování uvedené adresy jsem dostal odpověď, že služba není určena pro fyzické osoby a tedy lze očekávat obdobný přístup jako u společnosti I.CA. Platná

certifikační politika však stále službu uvádí jako neposkytovanou. [3, kap. 7.3] ⁶⁾

Publikovaná informace na internetových stránkách PostSignum

Nová služba pro ověření stavu certifikátu - OCSP

17.12.2012 10:20

Certifikační autorita PostSignum dosud zveřejňuje informace o zneplatněných certifikátech na seznamu zneplatněných certifikátů (CRL). Tento seznam je obvykle na distribučních bodech aktualizován neprodleně po zneplatnění některého certifikátu.

V rámci rozšiřování služeb jsme uvedli do pilotního provozu službu On-line ověřování stavu certifikátu (OCSP). Při využívání této služby je informace o stavu certifikátů zjišťována v reálném čase a není nutné stahovat objemné seznamy CRL.

V případě zájmu o zřízení služby, která je po dobu pilotního provozu zdarma, kontaktujte certifikační autoritu PostSignum na e-mailovou adresu ocsp.postsignum@cpost.cz.

CA eIdentity službu OCSP neposkytuje pro jistotu vůbec. [4, kap. 4.9.9]

Pokud tedy chceme využít v rámci naší organizace v různých aplikacích možnosti přihlašování certifikátem od české externí CA nezbyvá nám, než pracovat se zveřejňovanými CRL v každé aplikaci zvlášť. Tento přístup přináší mnohá rizika a zejména v situacích, kdy využíváme služeb třetích stran pro správu našeho IT (outsourcing), nemusíme mít možnost přímé kontroly nad procesem zpracovávání a aktualizací CRL.

Částečně můžeme tuto situaci řešit vybudováním vlastní služby OCSP Responderu. Zde je potřeba podotknout, že stále budeme závislí na zpracování CRL jednotlivých CA (tedy i na časových prodlevách při aktualizaci CRL), ale provedeme toto zpracování centralizovaně a nabídneme standardní jednotné rozhraní pro ověřování platnosti certifikátů pro všechny aplikace podporující OCSP dotazování v naší organizaci. Z pohledu bezpečnosti je možné takovýto systém vyčlenit ze standardní podpory a nebo pro něj zavést jiná bezpečnostní pravidla.

2.7 Požadavek (request) a odpověď (response)

Protokol OCSP je založen na principu dotaz/odpověď. Požadavek (dotaz) na ověření vyšle aplikace požadující ověření platnosti certifikátu (OCSP Klient) a server (OCSP Responder) zašle zpět podepsanou odpověď se statusem ověřovaného certifikátu. Klient s Responderem komunikuje pomocí http protokolu. Na výběr jsou 2 možnosti GET/POST. [1, s. 274]

V praktické části v příkladu 4.3.1 na straně 36 si ukážeme, jak vypadá požadavek zasílaný HTTP POST i GET.

⁶⁾Nicméně pro studijní účely jsem nakonec dostal balíček „100 dotazů“ zdarma.

OCSP požadavek obsahuje následující data [6, kap. 2.1]

- verze protokolu
- servisní požadavek
- identifikátor cílového certifikátu
- volitelná rozšíření, které mohou být zpracována OCSP Responderem

OCSP odpověď (`OCSPResponse`) se skládá z obálky a těla odpovědi (`responseBytes`). OCSP Obálka je jakousi minimální odpovědí OCSP Serveru, protože zejména v případě chyby může být tělo odpovědi prázdné. [1, s. 269]

Příklad definice OCSP odpovědi z RFC-2560 [6]

```
OCSPResponse ::= SEQUENCE {  
    responseStatus      OCSPResponseStatus,  
    responseBytes       [0] EXPLICIT ResponseBytes OPTIONAL  
}  
  
OCSPResponseStatus ::= ENUMERATED {  
    successful           (0), --Response has valid confirmations  
    malformedRequest     (1), --Illegal confirmation request  
    internalError        (2), --Internal error in issuer  
    tryLater             (3), --Try again later  
                        --(4) is not used  
    sigRequired          (5), --Must sign the request  
    unauthorized         (6) --Request unauthorized  
}
```

3 OPEN SOURCE PROJEKTY IMPLEMENTUJÍCÍ OCSP RESPONDERY

Jak vyplývá ze zadání práce, rozebereme dále možnosti implementace OCSP Responderu s využitím open source projektů:

- OpenCA OCSP Responder - <http://www.openca.org/>,
- EJBCA - <http://www.ejbca.org/>.

Pro úplnost uvedu ještě další dva Open Source projekty, které implementují OCSP Responder:

- openssl ocsdp - <http://www.openssl.org/docs/apps/ocsp.html> - jednoduchý OCPS responder vhodný zejména pro testování a ladění,
- dirmngr - <http://www.gnupg.org/documentation/manuals/dirmngr/> - další z možných OCSP responderů, využívá jej např. projekt GnuPG 2. ¹⁾

Další nástroje

Pro účely testování, analýzy a názorných ukázek budou v praktické části použity další nástroje (utility):

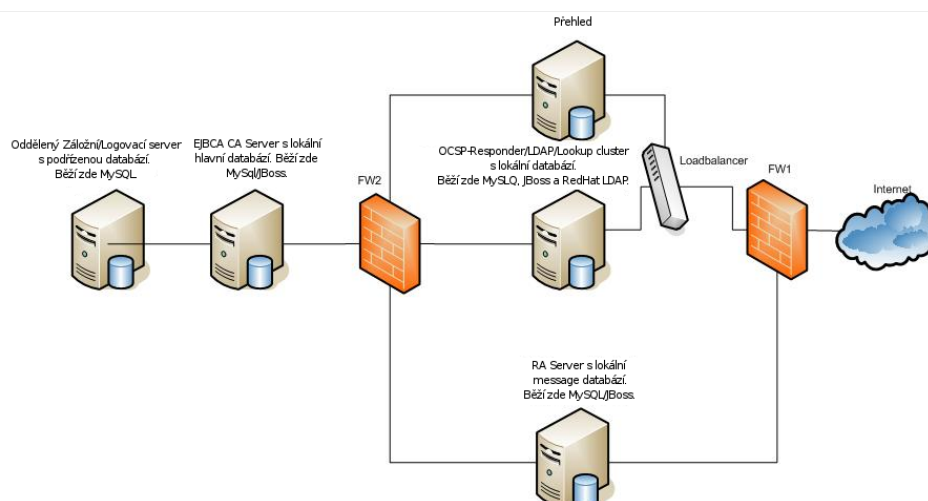
- openssl - <http://www.openssl.org/>,
- WIRESHARK - <http://www.wireshark.org/> pro sledování síťové komunikace,
- DumpASN1 - ASN.1 object dump/syntax check program - <http://www.cs.auckland.ac.nz/~pgut001/>,
- od - dump files in octal and other formats - <http://www.gnu.org/software/coreutils/manual/coreutils.html#od-invocation> ,
- Bless Hex Editor - <http://home.gna.org/bless/index.html>.

3.1 OpenCA

Jednou z částí projektu OpenCA je modul „OCSP Responder“ (ocspd) implementující funkcionalitu dle RFC ²⁾ 2560 - *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*.

¹⁾GnuPG je GNU projekt - kompletní a „free“ implementace OpenPGP standardu jak je definováno v RFC-4880.

²⁾RFC - Request for Comments - The Internet Engineering Task Force (IETF) <http://www.ietf.org/>



Obrázek 3: Architektura certifikační autority od EJBCA

Jeho výhodou je nenáročnost na instalaci a poměrně přímočará konfigurace. Napsaný je v jazyce C se závislostí na knihovně LibPKI, která je také součástí projektu OpenCA. Vhodný bude pro menší řešení a my si jej představíme více včetně funkční konfigurace v praktické části. [8]



Obrázek 1: Logo OpenCA

3.2 EJBCA

Pokud je naším cílem vybudovat opravdu robustní řešení s vysokou dostupností a škálovatelností, doporučuji zvolit OCSP Responder z projektu EJBCA. Jedná se o komplexní řešení celé PKI infrastruktury a stejně tak jako u OpenCA i zde je možné vybudovat oddělený OCSP responder.

Podmínkou nasazení je na J2EE serveru (JBoss, Tomcat, ...) „deployovat“ samostatnou validační službu a nakonfigurovat v ní OCSP Responder nebo jej nakonfigurovat společně s komponentami CA. Řešení je kompletně napsané v jazyce Java, hodí se tedy pro integraci do vlastních Java řešení.



Obrázek 2: Logo EJBCA

Z mnoha důvodů je vhodné oddělit OCSP Responder od Certifikační autority (CA) [9, architecture-ocsp]. Pokud chceme zajistit, že služba bude dostatečně výkonná, vybudujeme několik synchronizovaných OCSP responderů a vložíme před ně „load“ balancer. Takovým může být třeba řešení od společnosti F5³⁾, která již však není Open Source.

Architektura na obrázku 3 ukazuje příklad zabezpečené certifikační autority s odděleným OCSP Responderem, tak jak ji představuje projekt EJBCA.⁴⁾ [9]

³⁾www.f5.com/products/big-ip/

⁴⁾Texty u obrázků byly přeloženy.

II. PRAKTICKÁ ČÁST

4 IMPLEMENTACE OCSP RESPONDERU S KONSOLIDOVANOU DATABÁZÍ CRL

Náš ukázkový OCSP Responder postavíme s využitím „ocspd“ z projektu OpenCA. Nekladu si za cíl podat kompletní popis jednotlivých použitých komponent, ale na jednom místě soustředit údaje a svoje zkušenosti s vytvořením funkčního responderu pro CA v ČR. Osobně bych takovouto „příručku“ ocenil v době, kdy jsem stál před úkolem připravit podobné řešení a nikde jsem ji nenašel. Na obrázku 1 je jednoduše znázorněn princip našeho responderu. Z externích CA v internetu stahuje CRL, ukládá je ve vlastní databázi a vyřizuje požadavky našich aplikací v intranetu. Protože je responder na rozhraní vnitřní a vnější sítě je vhodné jej umístit do demilitarizované zóny (DMZ), která je oddělena firewally s pravidly definujícími povolenou komunikaci na portech.

Při instalaci je možné zvolit variantu předkompilovaného binárního balíčku nebo zdrojového kódu a následné kompilace. V našem případě zvolíme zdrojové soubory, které s výhodou využijeme i při odhalování příčin případných problémů ať již při instalaci tak i při samotném běhu aplikace. Všechny ukázky v následujících kapitolách budu provádět na operačním systému Linux, v mém případě na distribuci Ubuntu 12.04.2 LTS a zmíním se o některých problémech a jejich řešeních.

Stránky projektu OpenCA nabízí již předkompilované balíčky pro různé platformy, pro Linux jsou to varianty textového RPM installeru i grafického instalačního programu. Vyzkoušel jsem všechny možnosti, ale my si vybereme kompilaci ze zdrojových souborů protože se samotnou dokumentací na stránkách projektu jsem si nakonec při konfiguraci a problémech nevystačil. Navíc máme plnou kontrolu nad konečnou podobou výsledných binárních souborů a závislostí.

K dispozici je jako přímý odkaz archiv verze 2.1.0 na www.openca.org/projects/ocspd/sources.shtml a dále je možné přejít odkazem **releases** do repozitáře verzí. My použijeme poslední dostupnou verzi 2.1.1.¹⁾ Dobrou praxí je po stažení ověřit, že soubor odpovídá publikovanému otisku.

```
$ wget http://ftp.openca.org/openca-ocspd/releases/v2.1.1/sources/openca-ocspd-2.1.1.tar.gz
$ wget http://ftp.openca.org/openca-ocspd/releases/v2.1.1/sources/openca-ocspd-2.1.1.tar.gz.sha1
$ sha1sum openca-ocspd-2.1.1.tar.gz && cat openca-ocspd-2.1.1.tar.gz.sha1
c9d88c319f41955a9c3a9b3f4ad501cedcba2703  openca-ocspd-2.1.1.tar.gz
c9d88c319f41955a9c3a9b3f4ad501cedcba2703
```

Otisky si odpovídají, můžeme tedy přejít k samotné instalaci.

¹⁾<http://ftp.openca.org/openca-ocspd/releases/v2.1.1/sources/openca-ocspd-2.1.1.tar.gz>

Postupným spuštěním

```
./configure
make
make install
```

dojde ke kompilaci a instalaci do prostředí včetně vzorových nastavení, které následně upravíme. Pokud nechceme výchozí nastavení (umístění binárních souborů, vlastní závislosti, nebo jinou konfiguraci), nastavíme volby při spuštění `./configure`. Seznam voleb dostaneme přes `./configure --help`.

Velkou výhodou otevřeného kódu je možnost jeho úpravy, třeba jen drobnou modifikací, která vyřeší problém a může například vést k lepší integraci do cílového prostředí. V našem případě jsem provedl úpravu ve zdroji `ocspd.c` tak, aby bylo možné volit přepínačem `-x` mezi chybovým výstupem na `stderr` a nebo výchozí `syslog`. (Po instalaci a spuštění jsem totiž nedostával i přes volby `-v` a `-debug` žádná hlášení do `syslogu`. Jedním z možných řešení bylo přidat tuto volbu, protože čas věnovaný odhalování příčiny, proč se do `syslogu` nepíše, začínal být nad rámec původního plánu.)

Po spuštění `make install` se zkopíruje samotný `ocspd` program do `/usr/local/sbin/` a založí se adresářová struktura v `/usr/local/etc/`.²⁾ Vzorové konfigurační soubory nahradíme naší vlastní konfigurací pro CA eIdentity, PostSignum, I.CA a přidáme i vlastní CA vytvořenou pro testovací účely (UTB-BP). Tou jsme vystavili podepisovací certifikát OCSP odpovědí našeho responderu, kterému budeme důvěřovat.

Program `ocspd` umí pracovat s formáty `conf` i `xml`, my použijeme XML formát a strukturu „sub d-konfigurací“ v adresáři `ca.d` a `pki/token.d`

V konfiguraci `ocspd.xml` nás bude především zajímat hodnota pole `<pki:crlAutoReload>` kterou nastavíme na 3600 sekund. Určujeme tím, jak často má `ocspd` provádět stažení CRL z distribučního bodu CA. Dokumentace k jednotlivým položkám XML konfigurace se nachází přímo ve vzorovém konfiguračním souboru jako XML komentáře a v našich příkladech budeme tyto řádky z důvodu přehlednosti zpravidla vynechávat. Rozhodnete-li se pro formát `conf`, naleznete detailní popis v manuálových stránkách (`man ocspd`).

```
<?xml version="1.0" ?>
<!-- OCSP Daemon configuration --> <pki:serverConfig xmlns:pki="http://www.openca.org/
openca/pki/1/0/0">
  <pki:general>
    <pki:pkiConfigDir>/usr/local/etc/ocspd/pki</pki:pkiConfigDir>
    <pki:token>ocspServerToken</pki:token>
    <pki:caConfigDir>/usr/local/etc/ocspd/ca.d</pki:caConfigDir>
    <pki:pidFile>/usr/local/var/run/ocspd.pid</pki:pidFile>
    <pki:spawnThreads>10</pki:spawnThreads>
    <pki:crlAutoReload>3600</pki:crlAutoReload>
    <pki:crlReloadExpired>yes</pki:crlReloadExpired>
    <pki:crlCheckValidity>600</pki:crlCheckValidity>
  </pki:general>
```

²⁾Pokud jsme ponechali při kompilaci výchozí nastavení

```

<!-- Security Related Configurations -->
<pki:security>
  <pki:user>nobody</pki:user>
  <pki:group>daemon</pki:group>
</pki:security>
<!-- Service Network Configuration -->
<pki:network>
  <pki:bindAddress>http://0.0.0.0:6060</pki:bindAddress>
  <pki:httpProtocol>1.0</pki:httpProtocol>
  <pki:httpBaseUrl></pki:httpBaseUrl>
  <pki:timeOut>5</pki:timeOut>
</pki:network>
<!-- OCSP response configuration -->
<pki:response>
  <pki:maxReqSize>8192</pki:maxReqSize>
  <pki:digestAlgorithm>sha1</pki:digestAlgorithm>
  <pki:addResponseKeyID>yes</pki:addResponseKeyID>
  <pki:validity>
    <pki:days>0</pki:days>
    <pki:mins>3</pki:mins>
  </pki:validity>
</pki:response>
</pki:serverConfig>

```

Výpis 1: ocspd.xml

Atribut `<pki:token>` v `ocspd.xml` ponecháme na výchozí hodnotě `ocspServerToken` a musíme upravit konfigurační soubor `pki/token/software.xml`. Nastavím atributy `<pki:keypair>`, `<pki:cert>` a `<pki:cacert>`. Pro testovací účely nemáme klíč podepisující odpovědi chráněn heslem, ale pro produkční nasazení by nastaveno být mělo. OCSP podporuje i hardwarové úložiště pro klíče HSM ³⁾.

```

<?xml version="1.0" ?>
<!-- PKI service configurations -->
<pki:tokenConfig xmlns:pki="http://www.openca.org/openca/pki/1/0/0">
  <pki:name>ocspServerToken</pki:name>
  <pki:type>software</pki:type>
  <pki:keypair>file:///usr/local/etc/ocspd/private/ocspd-001.key.pem</pki:keypair>
  <pki:cert>file:///usr/local/etc/ocspd/certs/ocspd-001.cert.pem</pki:cert>
  <pki:cacert>file:///usr/local/etc/ocspd/certs/UTB-BP-cacert.pem</pki:cacert>
  <pki:passin>none</pki:passin>
</pki:tokenConfig>

```

Výpis 2: software.xml

Konfigurační soubory jednotlivých CA jsou v adresáři `ca.d` a pro každou CA je jeden XML. V případě PostSignum a eIdentity budeme mít konfigurace pro root a sub CA, I.CA sub CA nepoužívá, proto si vystačíme jen s jedním konfiguračním souborem.

```

<?xml version="1.0" ?>
<!-- OCSP Daemon configuration -->
<pki:caConfig xmlns:pki="http://www.openca.org/openca/pki/1/0/0">
  <pki:name>Eidentity-cca</pki:name>
  <pki:caCertUrl>http://www.acaeid.cz/cca/cca.pem</pki:caCertUrl>
  <pki:crlUrl>http://www.acaeid.cz/cca/crl/actual.crl</pki:crlUrl>
  <pki:caCompromised>no</pki:caCompromised>
</pki:caConfig>

```

Výpis 3: eidentity-cca2.xml

³⁾Hardware Security Module

Ostatní konfigurace provedeme obdobným způsobem, podle následujících údajů:

```
##### I.CA #####$
# Korenovy certifikat certifikacni authority provydavane komercni certifikaty SHA2
caCertUrl http://ica.cz/userfiles/files/certifikaty/SHA2/sica_root_key_20090901.pem$
crlUrl http://s.ica.cz/sica09.crl$
# řekoenov ácertifikty čicertifikan authority pro áevydan čikomern ácertifikty SHA1$
caCertUrl http://ica.cz/Userfiles/files/certifikaty/SHA1/sica_root_20080311.pem$
crlUrl http://s.ica.cz/sica08.crl$
##### PostSignum #####$
# Korenova certifikacni autorita PostSignum Root QCA$
caCertUrl http://www.postsignum.cz/files/ca/postsignum_qca_root.pem$
crlUrl http://www.postsignum.cz/crl/psrootqca.crl$
# Komerční certifikacni autorita PostSignum Public CA$
caCertUrl http://www.postsignum.cz/files/ca/postsignum_vca_sub.pem$
crlUrl http://www.postsignum.cz/crl/pspublicca.crl$
# Korenova certifikacni autorita PostSignum Root QCA 2$
caCertUrl http://www.postsignum.cz/files/ca/postsignum_qca2_root.pem$
crlUrl http://www.postsignum.cz/crl/psrootqca2.crl$
# Komerční certifikacni autorita PostSignum Public CA 2$
caCertUrl http://www.postsignum.cz/files/ca/postsignum_vca2_sub.pem$
crlUrl http://www.postsignum.cz/crl/pspublicca2.crl$
##### eIdentity #####$
# Korenovy kvalifikovany systemovy certifikat$
caCertUrl http://www.acaeid.cz/root2/root2.pem$
crlUrl http://www.acaeid.cz/root2/crl/actual.crl$
# Kvalifikovany systemovy certifikat komercni CA$
caCertUrl http://www.acaeid.cz/cca2/cca2.pem$
crlUrl http://www.acaeid.cz/cca2/crl/actual.crl$
# Korenova certifikacni autorita RCA / Certifikaty platne pro provoz do 31.12.2010$
# Korenovy kvalifikovany systemovy certifikat$
caCertUrl http://www.acaeid.cz/root/rca.pem$
crlUrl http://www.acaeid.cz/root/crl/actual.crl$
# Kvalifikovany systemovy certifikat komercni CA$
caCertUrl http://www.acaeid.cz/cca/cca.pem$
crlUrl http://www.acaeid.cz/cca/crl/actual.crl$
```

Výpis 4: parametry pro caCertUrl a crlUrl

Po úpravách bychom měli mít v adresáři `/usr/local/etc/ocspd` obdobnou strukturu.

```
/usr/local/etc/ocspd: tree -F
.
|-- ca.d/
|   |-- eidentity-cca2.xml
|   |-- eidentity-cca.xml
|   |-- eidentity-rca.xml
|   |-- eidentity-root2.xml
|   |-- postsignum_qca2_root.xml
|   |-- postsignum_qca_root.xml
|   |-- postsignum_vca2_sub.xml
|   |-- postsignum_vca_sub.xml
|   |-- sica_root_20080311.xml
|   |-- sica_root_key_20090901.xml
|   +-- UTB-BP.xml
|-- certs/
|   |-- 70234ebd.r0 -> ocspd-001.cert.pem
|   |-- ca-ext-certs/
|   |   |-- bundle.pem
|   |   |-- http://ica.cz_userfiles_files_certifikaty_SHA1_sica_root_20080311.pem
|   |   |-- http://ica.cz_userfiles_files_certifikaty_SHA2_sica_root_key_20090901.pem
|   |   |-- http://www.acaeid.cz_cca2_cca2.pem
|   |   |-- http://www.acaeid.cz_cca_cca.pem
|   |   |-- http://www.acaeid.cz_root2_root2.pem
|   |   |-- http://www.acaeid.cz_root_rca.pem
|   |   |-- http://www.postsignum.cz_files_ca_postsignum_qca2_root.pem
|   |   |-- http://www.postsignum.cz_files_ca_postsignum_qca_root.pem
|   |   |-- http://www.postsignum.cz_files_ca_postsignum_vca2_sub.pem
```

```
| | |-- http://www.postsignum.cz_files_ca_postsignum_vca_sub.pem
| | |-- postsignum-bundle-1.pem
| | +-- postsignum-bundle.pem
| |-- ocsdpd-001.cert.pem
| +-- UTB-BP-cacert.pem
|-- crls/
|-- ocsdpd.conf
|-- ocsdpd.xml
|-- ocsdpd.xml.2013-06-01
|-- pki/
| |-- hsm.d/
| |-- profile.d/
| +-- token.d/
| |-- eracom.xml
| |-- etoken.xml
| |-- software.xml
| +-- software.xml.2013-06-03
+-- private/
+-- ocsdpd-001.key.pem
```

Výpis 5: Struktura konfiguračních adresářů a souborů OCSPD

V konfiguracích můžeme použít jako zdroj pro certifikát nebo CRL i lokální soubor nebo uložení v LDAPu. Uvedená konfigurace má tu výhodu, že po spuštění a inicializaci `ocsdpd` máme okamžitě k dispozici funkční službu, která si pravidelně sama aktualizuje CRL z CA.

4.1 Spuštění OCSP Responderu

Spuštěním `ocsdpd` s konfiguračním souborem máme připravenou službu OCSP Responderu k použití.

```
$ sudo ocsdpd -debug -x -v -c /usr/local/etc/ocsdpd/ocsdpd.xml
```

```
OpenCA's OCSP Responder - v2.1.1
(c) 2002-2009 by Massimiliano Pala and OpenCA Project
OpenCA licensed software

Jun 12 12:13:43 2013 GMT [23863] GENERAL: OpenCA OCSPD v2.1.1 - starting.
Jun 12 12:13:43 2013 GMT [23863] INFO: [token.c:702]::DEBUG::Can not load profiles (/
home/sandorm/.libpki/profile.d)

Jun 12 12:13:43 2013 GMT [23863] INFO: [pki_config.c:981]::DEBUG::Loading file /usr/
local/etc/ocsdpd/ca.d/sica_root_20080311.xml
Jun 12 12:13:43 2013 GMT [23863] INFO: [pki_config.c:997]::DEBUG::Loaded /usr/local/
etc/ocsdpd/ca.d/sica_root_20080311.xml file
Jun 12 12:13:43 2013 GMT [23863] INFO: [pki_config.c:981]::DEBUG::Loading file /usr/
local/etc/ocsdpd/ca.d/postsignum_vca2_sub.xml
[....]
Jun 12 12:13:43 2013 GMT [23863] INFO: Expired CRLs Reload Disabled
Jun 12 12:13:43 2013 GMT [23863] INFO: [config.c:317]::DEBUG::Building CA List
Jun 12 12:13:43 2013 GMT [23863] INFO: [net/pki_socket.c:123]::DEBUG::Creating a~
simple connection
Jun 12 12:13:43 2013 GMT [23863] INFO: [net/sock.c:270]::DEBUG::Connection Successful
to ica.cz:80
Jun 12 12:13:43 2013 GMT [23863] INFO: [config.c:743]::DEBUG::Building CA_ENTRY_CERTID
Jun 12 12:13:43 2013 GMT [23863] INFO: [config.c:436]::DEBUG::Got CRL Url -> http://s.
ica.cz/sica08.crl
Jun 12 12:13:43 2013 GMT [23863] INFO: [net/pki_socket.c:123]::DEBUG::Creating a~
simple connection
Jun 12 12:13:43 2013 GMT [23863] INFO: [net/sock.c:270]::DEBUG::Connection Successful
to s.ica.cz:80
```

```
Jun 12 12:13:43 2013 GMT [23863] INFO: [crl.c:193]::DEBUG::CRL signature is verified!
Jun 12 12:13:43 2013 GMT [23863] INFO: [crl.c:216]::DEBUG::CRL and CA cert [1] check
ok
Jun 12 12:13:43 2013 GMT [23863] INFO: CRL matching CA cert ok [ 1 ]
Jun 12 12:13:43 2013 GMT [23863] INFO: [crl.c:285]::DEBUG::CRL::Verify 1 [OK=1]
Jun 12 12:13:43 2013 GMT [23863] INFO: INFO::CRL::14 Entries [ sica_root_20080311-SHA1
]
[....]
```

Rychlou kontrolou zjistíme, zda se server usadil na portu dle naší konfigurace a zda na portu odpovídá.

```
$ netstat -an | grep 6060
tcp        0      0 0.0.0.0:6060          0.0.0.0:*            LISTEN
```

```
$ echo "" | telnet localhost 6060
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Connection closed by foreign host.
```

Vše se zdá být v pořádku.

4.2 Test OCSP Responderu

Test provedeme v následujících krocích:

1. nalezení vhodného certifikátu výběrem ze souboru CRL,
2. zobrazení a kontrola jeho platnosti přes webové rozhraní,
3. kontrola přes náš lokální OCSP Responder,
4. kontrola přes OCSP Responder PostSignum.

4.2.1 Nalezení revokovaného certifikátu

Pro následující testy budeme hledat jakýkoliv odvolaný certifikát. Nejjednodušší cestou je stažení souboru CRL z CA a následný výběr sériového čísla certifikátu, který je odvolaný.

Vybrali jsme si například CA PostSignum. Z adresy <http://www.postsignum.cz/crl/pspublicca2.crl> jsme si stáhli jejich CRL soubor.

```
$ wget http://www.postsignum.cz/crl/pspublicca2.crl
```

Dle RFC-5280 [5, s. 69] může nabývat pole „reasonCode“ následujících hodnot.

```
-- reasonCode ::= { 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) }

```

Pomocí utility `openssl crl` jsme zobrazili CRL v čitelné podobě a našli v našem případě certifikát se sériovým číslem 0x08BF1B, který má v poli `reasonCode` uvedenu hodnotu `Superseded`, což je jeden z možných důvodů odvolání.

```

$ openssl crl -inform DER -in pspublicca2.crl -noout -text | tail -21 | head -8
Serial Number: 08BF1B
Revocation Date: Jun 12 10:32:04 2013 GMT
CRL entry extensions:
  X509v3 CRL Reason Code:
    Superseded
Signature Algorithm: sha256WithRSAEncryption
52:72:99:7b:cc:dc:c6:e6:7f:2b:f7:ce:07:24:c4:8d:7e:dc:
b6:b5:cd:d7:65:3a:4d:1b:8e:90:00:ae:c0:5b:68:f1:67:a8:

```

4.2.2 Kontrola platnosti certifikátu přes webové rozhraní

V předcházejícím odstavci jsme našli certifikát se sériovým číslem 0x08BF1B. Pokud známe sériové číslo certifikátu, na adrese www.postsignum.cz/certifikaty_uzivatelu.html si k němu můžeme dohledat podrobnosti viz. Obrázek 1.

Certifikáty uživatelů



Vyhledání podle sériového čísla certifikátu

Subjekt	T=Pracovník přepážky,serialNumber=P235299,CN=H...,OU=68022,OU=Czech POINT,O=Česká pošta, s.p. [IČ 47114983],C=CZ	
E-mailová adresa:	posta.p306054@cpost.cz	
Sériové číslo	573211	
Vydán dne	14.6.2012	
Platný do	14.6.2013	
Vystavitel	PostSignum VCA	
Revokováno	12.6.2013 12:32:04	
Stav	Revokovaný	DER / PEM / TXT(UTF-8)

Počet nalezených certifikátů: 1	Z toho určených ke zveřejnění: 1
1	

Obrázek 1: Odvolaný certifikát

4.2.3 Test volání s openssl

```
$ openssl ocsp -issuer ./http://www.postsignum.cz_files_ca_postsignum_vca2_sub.pem -
serial 0x08BF1B -req_text -host localhost:6060 -VAfile /usr/local/etc/ocspd/certs/
ocspd-001.cert.pem
```

Povšimněme si v příkladu parametru `-VAfile`, který u `openssl` určuje soubor s certifikátem, kterému důvěřujeme při ověření podpisu odpovědi. Vždy bychom měli ověřit podpis odpovědi, zajistíme si tím, že není podvržená (po cestě ji nikdo nezměnil). Více podrobností se dozvíme v kapitole týkající se bezpečnosti.

```
OCSP Request Data:
  Version: 1 (0x0)
  Requestor List:
    Certificate ID:
      Hash Algorithm: sha1
      Issuer Name Hash: 2DEDBBB2C0CEC1B4EDC7979EE3A59D89B4F2BC2C
      Issuer Key Hash: 48EF3ED4EA8989A3E9E23FDBEF8C42B10AF8C8D1
      Serial Number: 08BF1B
  Request Extensions:
    OCSP Nonce:
      0410FD2E32999797289784DFDB56A50DA8CF
Response verify OK
0x08BF1B: revoked
  This Update: Jun 12 12:15:25 2013 GMT
  Next Update: Jun 12 12:18:25 2013 GMT
  Reason: superseded
  Revocation Time: Jun 12 10:32:04 2013 GMT
```

Následující LOG našeho OCSP Responderu ukazuje, že obdržel požadavek, našel správnou CA podle `nameHash`, vrací status for 573211 is REVOKED a Response signed ok!.

```
Jun 12 12:15:18 2013 GMT [23863] INFO: [threads.c:70]:DEBUG::Thread [6] - got fd 5
Jun 12 12:15:18 2013 GMT [23863] INFO: [core.c:164]:DEBUG::CORE::Waiting on connect
Jun 12 12:15:18 2013 GMT [23863] INFO: [threads.c:83]:DEBUG::[Thread::6] Got resp
from socket
Jun 12 12:15:18 2013 GMT [23863] INFO: request for certificate serial 573211
Jun 12 12:15:18 2013 GMT [23863] INFO: [response.c:440]:DEBUG::CRL::CA [
sica_root_20080311-SHA1] nameHash mismatch (1)
Jun 12 12:15:18 2013 GMT [23863] INFO: [response.c:448]:DEBUG::CRL::CA [
postsignum_vca2_sub] nameHash OK
Jun 12 12:15:18 2013 GMT [23863] INFO: Status for 573211 is REVOKED
Jun 12 12:15:18 2013 GMT [23863] INFO: [response.c:243]:DEBUG::Token Algorithm:
sha256WithRSAEncryption
Jun 12 12:15:18 2013 GMT [23863] INFO: [hsm_main.c:511]:DEBUG::Using HSM for Key
Operations
Jun 12 12:15:18 2013 GMT [23863] INFO: [hsm_main.c:519]:DEBUG::HSM sign() callback
called
Jun 12 12:15:18 2013 GMT [23863] INFO: [response.c:320]:DEBUG::Response signed ok!
Jun 12 12:15:18 2013 GMT [23863] INFO: [threads.c:90]:DEBUG::[Thread::6] Built resp
from socket
```

4.2.4 Test s OCSP Responderem PostSignum

V kapitole 2.6 na straně 18 jsme se dozvěděli, že PostSignum nabízí OCSP službu zatím jen v pilotním provozu a není veřejně dostupná. Pro účely této práce jsem kontaktoval

oddělení vývoje v PostSignum a dohodl se s nimi na možnosti zřízení přístupu ke službě pro studijní účely. Jednou z podmínek bylo vyplnění formuláře s kontaktními údaji a se způsobem, jakým se ke službě budu přihlašovat. Na výběr byly dány tři možnosti autentizace⁴⁾:

- certifikátem PostSignum VCA, nebo
- jménem/heslem (basic), nebo
- lze požadavky zasílat bez autentizace z vyhrazené IP adresy.

Já jsem zvolil variantu BASIC autentizace, tedy jménem a heslem zasílaným současně s požadavkem v jeho hlavičce. Nenašel jsem vhodnou volbu v openssl ocspl klientu, která by umožňovala autentizovat se proti serveru, proto zde provedeme postupně založení dotazu, odeslání POST požadavku s BASIC autentizací na URL OCSP služby PostSignum a opět následné zpracování odpovědi pomocí openssl ocspl klienta.⁵⁾

Vygenerujeme OCSP požadavek a uložíme ho do souboru `req_out_08BF1B.out`. Výstupní formát je DER.

```
$ openssl ocspl -issuer ./http://www.postsignum.cz_files_ca_postsignum_vca2_sub.pem -
serial 0x08BF1B -req_text -reqout ~/Documents/SKOLA/BAKALARSKA-PRACE/PODKLADY/
req_out_08BF1B.out
```

DER enkódovaná binární data zašleme POST požadavkem a odpověď uložíme do souboru `resp_out_08BF1B.out.5`. Program `curl` umožňuje autentizaci v parametru `--user` a stejně tak nastavení položek v hlavičce požadavku, jak vyžaduje RFC-2560. [6, příloha A. kap. 1.1 Request].

```
$ curl -k --verbose --data-binary @./req_out_08BF1B.out -H "Content-Type:application/
ocsp-request" -o resp_out_08BF1B.out.5 --user jmeno:heslo https://www3.postsignum.
cz/OCSP/VCA2/OCSP_user/
```

Odpověď prohlédneme opět pomocí `openssl ocspl`.

```
$ openssl ocspl -respin ./resp_out_08BF1B.out.5 -text
```

```
OCSP Response Data:
OCSP Response Status: successful (0x0)
Response Type: Basic OCSP Response
Version: 1 (0x0)
Responder Id: C = CZ, 0~ = "\C4\8Ces\3\A1 po\C5\A1ta, s.p. [I\C4\8C 47114983]",
OU = PostSignum Services, CN = PostSignum VCA - OCSP Responder 2
Produced At: Jun 12 14:59:07 2013 GMT
Responses:
Certificate ID:
```

⁴⁾Dle dokumentu *Certifikační autorita PostSignum, „Služba on-line ověření stavu certifikátu – OCS“*, *Technické řešení, PILOTNÍ PROVOZ*, který však není volně dostupný.

⁵⁾Čerpáno z příspěvku unmitigatedrisk.com/?p=42 doplněného o autentizaci.

Certifikáty uživatelů



Vyhledání podle sériového čísla certifikátu

Subjekt	serialNumber=P245271,CN= [redacted] á,OU=163334,OU=Czech POINT,O=Česká pošta, s.p. [IČ 47114983],C=CZ	
E-mailová adresa:	posta.p306054@cpost.cz	
Sériové číslo	585253	
Vydán dne	19.9.2012	
Platný do	19.9.2013	
Vystavitel	PostSignum VCA	
Stav	Platný	DER / PEM / TXT(UTF-8)

Počet nalezených certifikátů: 1	Z toho určených ke zveřejnění: 1
1	

Obrázek 2: Platný certifikát

```

Hash Algorithm: sha1
Issuer Name Hash: 2DEDBBB2C0CEC1B4EDC7979EE3A59D89B4F2BC2C
Issuer Key Hash: 48EF3ED4EA8989A3E9E23FDBEF8C42B10AF8C8D1
Serial Number: 08BF1B
Cert Status: revoked
Revocation Time: Jun 12 10:32:04 2013 GMT
Revocation Reason: superseded (0x4)
This Update: Jun 12 14:59:07 2013 GMT

Response Extensions:
  OCSP Nonce:
    0410F8B111704491CE4B4F7BF36DF7D11ACB
Signature Algorithm: sha1WithRSAEncryption
  69:7a:e9:bf:21:67:8c:72:68:cf:23:48:41:d7:97:08:5d:78:
  ad:94:14:3f:06:65:7a:85:7a:78:a8:64:ff:2f:08:4e:38:50:
[... zkraceno ...]
```

Odpověď od našeho OCSP Responderu i OCSP Responderu PostSignum vrací shodný výsledek, certifikát se sériovým číslem je „revokovaný“.

Pro úplnost uvedeme ještě test s certifikátem, který je platný. Na obrázku 2 máme platný certifikát se SN 585253.

Zavoláme náš OCSP Responder a vidíme, že odpověď je Response verify OK = 0x08EE25: good.

```
$ openssl ocsp -issuer ./http://www.postsignum.cz_files_ca_postsignum_vca2_sub.pem -
serial 0x08EE25 -req_text -host localhost:6060 -VAfile /usr/local/etc/ocspd/certs/
ocspd-001.cert.pem
```

```

0OCSP Request Data:
  Version: 1 (0x0)
  Requestor List:
    Certificate ID:
      Hash Algorithm: sha1
      Issuer Name Hash: 2DEDBBB2C0CEC1B4EDC7979EE3A59D89B4F2BC2C
      Issuer Key Hash: 48EF3ED4EA8989A3E9E23FDBEF8C42B10AF8C8D1
      Serial Number: 08EE25
```

```

Request Extensions:
  OCSP Nonce:
    04108D1E64264E90D95CC7D41808F1EB0725
Response verify OK
0x08EE25: good
  This Update: Jun 12 20:59:54 2013 GMT
  Next Update: Jun 12 21:02:54 2013 GMT

```

Log našeho OCSP Responderu ukazuje, že obdržel požadavek, našel správnou CA podle nameHash, vrací status VALID for 585253 a Response signed ok!.

```

Jun 12 20:59:54 2013 GMT [23863] INFO: [threads.c:70]::DEBUG::Thread [4] - got fd 5
Jun 12 20:59:54 2013 GMT [23863] INFO: [core.c:164]::DEBUG::CORE::Waiting on connect
Jun 12 20:59:54 2013 GMT [23863] INFO: [threads.c:83]::DEBUG::[Thread::4] Got resp
from socket
Jun 12 20:59:54 2013 GMT [23863] INFO: request for certificate serial 585253
Jun 12 20:59:54 2013 GMT [23863] INFO: [response.c:440]::DEBUG::CRL::CA [
sica_root_20080311-SHA1] nameHash mismatch (1)
Jun 12 20:59:54 2013 GMT [23863] INFO: [response.c:448]::DEBUG::CRL::CA [
postsignum_vca2_sub] nameHash OK
Jun 12 20:59:54 2013 GMT [23863] INFO: status VALID for 585253
Jun 12 20:59:54 2013 GMT [23863] INFO: [response.c:243]::DEBUG::Token Algorithm:
sha256WithRSAEncryption
Jun 12 20:59:54 2013 GMT [23863] INFO: [hsm_main.c:511]::DEBUG::Using HSM for Key
Operations
Jun 12 20:59:54 2013 GMT [23863] INFO: [hsm_main.c:519]::DEBUG::HSM sign() callback
called
Jun 12 20:59:54 2013 GMT [23863] INFO: [response.c:320]::DEBUG::Response signed ok!
Jun 12 20:59:54 2013 GMT [23863] INFO: [threads.c:90]::DEBUG::[Thread::4] Built resp
from socket

```

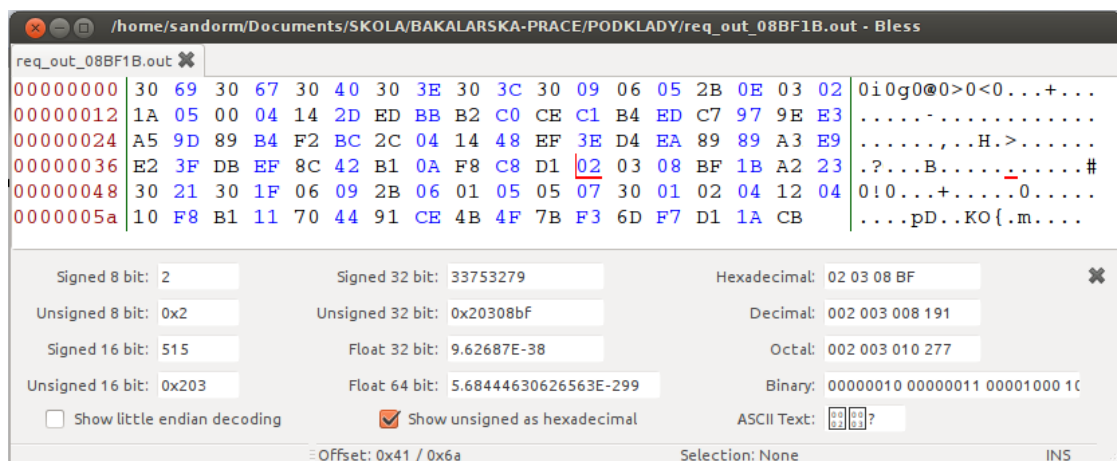
4.3 OCSP požadavek podrobně

V kapitole 2.4 jsme krátce nahlédli pod kapotu ASN.1 a kódování DER. Podíváme se nyní prakticky na vygenerovaný požadavek. Najdeme a dekodujeme jednoduchý tág pro INTEGER. Na data nabídnou pohled třemi různými programy. Grafický Bless Hex Editor, textový od a dumpasn1. Bless Hex Editor je typickým zástupcem programů s jednoduchým grafickým rozhraním a jeho ovládání je shodné s většinou obdobných programů. Za zmínku stojí utilita od, která není příliš známá, ale najdeme ji v naprosté většině operačních systémů Unixového typu a velmi dobře nám poslouží pro náhled na binární data.

Požadavek req_out_08BF1B.out máme v DER kódování a z popisu struktury víme, že musí obsahovat pole identifikující certifikát - sériové číslo.

V našem případě budeme hledat sériové číslo, které má v hexadecimálním tvaru hodnotu <02 03 08 BF 1B>, což dle tabulky 1 na straně 16 reprezentuje:

- údaj typu INTEGER - tág má hodnotu 02,
- délka je 03, tzn. že v datech budou následovat 3 bajty dat,
- data 08 BF 1B odpovídají hexadecimálnímu tvaru sériového čísla z našeho certifikátu.



Obrázek 3: Bless Hex Editor

Na obrázku 3 vidíme v poli „hexadecimal“ hledanou hodnotu.

Obdobný pohled dostaneme i s využitím programu od.

```
$ od -t x1 req_out_08BF1B.out
00000000 30 69 30 67 30 40 30 3e 30 3c 30 09 06 05 2b 0e
00000020 03 02 1a 05 00 04 14 2d ed bb b2 c0 ce c1 b4 ed
00000040 c7 97 9e e3 a5 9d 89 b4 f2 bc 2c 04 14 48 ef 3e
00000060 d4 ea 89 89 a3 e9 e2 3f db ef 8c 42 b1 0a f8 c8
00000100 d1 02 03 08 bf 1b a2 23 30 21 30 1f 06 09 2b 06
00000120 01 05 05 07 30 01 02 04 12 04 10 f8 b1 11 70 44
00000140 91 ce 4b 4f 7b f3 6d f7 d1 1a cb
```

Program dumpasn1 je vytvořen právě pro účely analýzy ASN.1 a můžeme si jím ověřit, že jsme data skutečně správně přečetli.

```
$ dumpasn1 -dhh req_out_08BF1B.out
<30 69 30 67 30 40 30 3E 30 3C 30 09 06 05 2B 0E 03 02 1A 05 00 04 14 2D>
0 105: SEQUENCE {
  <30 67 30 40 30 3E 30 3C 30 09 06 05 2B 0E 03 02 1A 05 00 04 14 2D ED BB>
2 103: . SEQUENCE {
  <30 40 30 3E 30 3C 30 09 06 05 2B 0E 03 02 1A 05 00 04 14 2D ED BB B2 C0>
4 64: . . SEQUENCE {
  <30 3E 30 3C 30 09 06 05 2B 0E 03 02 1A 05 00 04 14 2D ED BB B2 C0 CE C1>
6 62: . . . SEQUENCE {
  <30 3C 30 09 06 05 2B 0E 03 02 1A 05 00 04 14 2D ED BB B2 C0 CE C1 B4 ED>
8 60: . . . . SEQUENCE {
  <30 09 06 05 2B 0E 03 02 1A 05 00>
10 9: . . . . . SEQUENCE {
  <06 05 2B 0E 03 02 1A>
12 5: . . . . . OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
  <05 00>
19 0: . . . . . NULL
  : . . . . . }
  <04 14 2D ED BB B2 C0 CE C1 B4 ED C7 97 9E E3 A5 9D 89 B4 F2 BC 2C>
21 20: . . . . . OCTET STRING
  : . . . . . 2D ED BB B2 C0 CE C1 B4 ED C7 97 9E E3 A5 9D 89
  : . . . . . B4 F2 BC 2C
  <04 14 48 EF 3E D4 EA 89 89 A3 E9 E2 3F DB EF 8C 42 B1 0A F8 C8 D1>
43 20: . . . . . OCTET STRING
  : . . . . . 48 EF 3E D4 EA 89 89 A3 E9 E2 3F DB EF 8C 42 B1
  : . . . . . 0A F8 C8 D1
  <02 03 08 BF 1B>
65 3: . . . . . INTEGER 573211
  : . . . . . }
  : . . . . . }
```

```

    : . . . . }
    <A2 23 30 21 30 1F 06 09 2B 06 01 05 05 07 30 01 02 04 12 04 10 F8 B1 11>
70    35: . . [2] {
    <30 21 30 1F 06 09 2B 06 01 05 05 07 30 01 02 04 12 04 10 F8 B1 11 70 44>
72    33: . . . SEQUENCE {
    <30 1F 06 09 2B 06 01 05 05 07 30 01 02 04 12 04 10 F8 B1 11 70 44 91 CE>
74    31: . . . . SEQUENCE {
    <06 09 2B 06 01 05 05 07 30 01 02>
76    9: . . . . . OBJECT IDENTIFIER ocspNonce (1 3 6 1 5 5 7 48 1 2)
    <04 12 04 10 F8 B1 11 70 44 91 CE 4B 4F 7B F3 6D F7 D1 1A CB>
87    18: . . . . . OCTET STRING, encapsulates {
    <04 10 F8 B1 11 70 44 91 CE 4B 4F 7B F3 6D F7 D1 1A CB>
89    16: . . . . . . OCTET STRING
        : . . . . . . F8 B1 11 70 44 91 CE 4B 4F 7B F3 6D F7 D1 1A CB
        : . . . . . . }
        : . . . . . }
        : . . . . }
        : . . . }
        : . . }
        : . }

0 warnings, 0 errors.

```

4.3.1 OSCP požadavek metodou HTTP GET

Ve všech předchozích příkladech jsme požadavek zasílali metodou HTTP POST.

Pokud je požadavek kratší než 255 bajtů je možné použít dle RFC-2560 metodu GET.[6, s. 16] Dotaz bude mít tvar:

```
GET URI/Base64_kodovany_dotaz
```

Dříve připravený požadavek `req_out_08BF1B.out` encodujeme do Base64, tak jak to vyžaduje RFC-2560 [6, s. 16].

```
openssl enc -in req_out_08BF1B.out -out req_out_08BF1B.out.base64 -a
```

Následně provede `urlencode`, tzn., že převedeme znaky, které není možné použít v HTTP dotazu přímo a můžeme provést testovací volání.

```
urlencode MGkwZzBAMD4wPDAJBGrDgMCGgUABBBQt7buywM7Bt03H157jpZ2JtPK8LAQUS08+10qJiaPp4j/  
b74xcsQr4yNECAwi/G6IjMCEwHwYJKwYBBQUHMAECBBIEEPixEXBEkc5LT3vzbffrGss=
```

MGkwZzBAMd4wPDAJBgUrDgMCGGuABbQ7t7buywM7bT03H157jpZ2tJPK8LAQUS08%2B10qJiaPp4j%2Fb74xCsQr4vNECAwi%2FG6IjMCEwHwYJKwYBBQUHMAECBBIIEEPixEXBEk5LT3vzbffRGss%3D

```
curl --verbose --url http://localhost:6060/  
  MGkwZzBAMD4wPDAJBgUrdgMCGgUABBBQt7buywM7BtO3H157jpZ2JtPK8LAQUS08%2B10qJiaPp4j%2  
  Fb74xCsQr4yNECAwi%2FG6IjMCEwHwYJKwYBBQUHMAECBBIEEPixEXBEkc5LT3vzbffRGss%3D
```

Na požadavky HTTP GET při testech s OSCP Responderem OpenCA jsem nedostával korektní odpovědi, stejně tak jako u PostSignum. V popisu technického řešení od PostSignum není ani metoda GET podporována.

4.4 Další možná vylepšení

Pokud se rozhodneme pro vlastní mechanismus aktualizace lokálně uložených CRL v souboru, (vzhledem k velikosti již dnes publikovaných CRL, případně frekvenci požadavků na kontrolu, zda nebylo vydáno nové), mohlo by se vám hodit optimalizovat metodu kontroly a stažení. Pokud budete provádět například příliš časté stahování, např. v intervalech pod jednu minutu, dozajista vás časem bude kontaktovat CA, protože takovéto chování vyhodnotí s velkou pravděpodobností na pokus o zahlcení služby.

V následujícím fragmentu `bash` skriptu je ukázka, jak je možné toto vylepšení realizovat.

Nejdříve se ujistíme, že CRL neexpirovalo (`check_all_crl_validity`), tzn. nejsme za datem `nextUpdate`, což by znamenalo okamžitou nutnost stažení celého CRL souboru z CA. Pokud tomu tak není, můžeme s výhodou využít hodnotu pole `ETag`, která nám vlastně poskytuje informaci o „otisku“ souboru dostupného na požadovaném odkaze.^{6) 7)}

Pokud se liší hodnoty `ETagu` již staženého CRL a aktuálního na CA, provede se opět plné stažení souboru. Touto metodou se ošetří nutnost stahovat velké objemy dat, pokud se nezměnily. `ETag` není povinnou položkou v odpovědi http serveru, ale všechny tři české CA jej v odpovědi t. č. uvádějí.

Příklad kontroly s využitím `ETagu`

```
### ... cast skriptu vypustena ....
function check_all_crl_validity {
    ls -1 ${CRL_PEM}/*.pem | while read line; do
        echo "###_check_all_crl_validity_:_${CRL_BIN}/crl-get-nextupdate.sh_${line}"
        ${CRL_BIN}/crl-get-nextupdate.sh ${line}
    done
}

function main {
    #### kontrola validity CRL ####
    check_all_crl_validity # zkontroluj casovou platnost CRL (netxtUpdate)

    CURRENT_DATE=$(date +"%Y-%m-%d_%H:%M:%S")
    cat $CRL_LIST | grep -v ^"# " | while read line; do
        new_line=$(echo "${line}" | sed "s/\\/_/g")
        touch "${CRL_ROOT}/head/${new_line}"
        ${CRL_BIN_OS}/curl -I "${line}" | grep -i etag > "${CRL_TMP}/${new_line}.tmp"
        ### obsluha chyby nemoznosti stahnout z~URL ....
        res=$?
        if [ ${res} -gt 0 ]; then
            log_write "E:201:${CURRENT_DATE}:_RESULT:_${res}:_Nepodarilo_
                se_provest_curl/HEAD_CRL_z~CA_-_${line}" >> ${CRL_LOG}/crl
                -downloader.log
            upload_crl_to_arp #pro jistotu, vzdy doplnim do profilu
        fi
    done
}
```

⁶⁾Tohoto mechanismu se běžně používá např. v proxy serverech s „cache“.

⁷⁾Více o `ETag` v RFC-2616 [11, s. 125, kap. 14.19]

```
diff "${CRL_ROOT}/head/${new_line}" "${CRL_TMP}/${new_line}.tmp"
res=$?
if [ ${res} -eq 1 ]; then
    echo "RESULT:␣${res}"
    ${CRL_BIN_OS}/curl -I "${line}" | grep -i etag > "${CRL_ROOT}/
    head/${new_line}"
    ${CRL_BIN}/curl-get-one.sh "${line}"
    upload_crl_to_arp
fi
done
```

Příklad odpovědi ze serveru PostSignum

Na volání HTTP HEAD vrátí server hlavičku, ve které je ETag vidět.

```
$ curl -I http://www.postsignum.cz/crl/psrootqca2.crl
HTTP/1.1 200 OK
Date: Wed, 12 Jun 2013 11:05:26 GMT
Server: Apache/2.2.22
Last-Modified: Thu, 03 Jan 2013 07:51:32 GMT
ETag: "57600000000217a-241-4d25da19c2900"
Accept-Ranges: bytes
Content-Length: 577
Connection: close
Content-Type: application/x-pkcs7-crl
```

5 VZOROVÁ APLIKACE S VYUŽITÍM SLUŽEB OCSP RESPONDERU

Na závěr si ukážeme krátkou aplikaci v jazyce Java. Nepsaným standardem v „javovém světě“ se stal balíček kryptografického API „The Legion of the Bouncy Castle“ a my jej použijeme při tvorbě vzorové aplikace (testováno bylo s knihovnami 1.46 pro JDK 1.6). Testovací aplikace nastiňuje, jak provést kroky nutné k plnému ověření, zda nebyl certifikát odvolán, s využitím OCSP protokolu. Třída uvádí příklad, jak by mohly podnikové aplikace integrovat ověření do vlastního řešení.



Obrázek 1: Logo Legion of the Bouncy Castle

Demo aplikaci tvoří jedna třída OCSPTestApp. (Úplný výpis zdrojového kódu je v příloze 7 na straně 48) Její metody vidíme na obrázku 2.

OCSPTestApp	
validate	(X509Certificate, BigInteger, String, X509Certificate) : int
loadCertificate	(String) : X509Certificate
generateOCSPRequest	(X509Certificate, BigInteger) : OCSPReq
sendOCSPRequest	(String, OCSPReq) : OCSPResp
main	(String[]) : void

Obrázek 2: Třída OCSPTestApp

Metoda **main**, požaduje na vstupu čtyři parametry:

1. certifikát vystavující CA pro kontrolované sériové číslo,
2. certifikát, kterým byla podepsána OCSP odpověď (ekvivalent `-VAfile` u `openssl ocspl`),
3. sériové číslo testovaného certifikátu (dekadicky),
4. URL OCSPD Responderu.

Číselné hodnoty statusu v ověření odpovídají specifikaci v [6, s. 18].

Příklad volání demo aplikace pro kontrolu odvolaného certifikátu se sériovým číslem 573211

```
$ /home/sandorm/src/workspace/OCSPTest/bin: java \  
> cz.sandor.bp.ocsp.OCSPTestApp \  
> /usr/local/etc/ocspd/certs/ca-ext-certs/http:__www.postsignum.  
  cz_files_ca_postsignum_vca2_sub.pem \  
> /usr/local/etc/ocspd/certs/ocspd-001.cert.pem \  
> 573211 http://localhost:6060  
URL = http://localhost:6060  
OCSP Singnature verification: OK!  
resp[0] certStatus: org.bouncycastle.ocsp.RevokedStatus@37123712  
resp[0] certSN: 573211  
Status overeni v OCSP odpovedi = 1
```

Příklad platného certifikátu se sériovým číslem 585253

```
$ /home/sandorm/src/workspace/OCSPTest/bin: java \  
> cz.sandor.bp.ocsp.OCSPTestApp \  
> /usr/local/etc/ocspd/certs/ca-ext-certs/http:__www.postsignum.  
  cz_files_ca_postsignum_vca2_sub.pem \  
> /usr/local/etc/ocspd/certs/ocspd-001.cert.pem \  
> 585253 http://localhost:6060  
URL = http://localhost:6060  
OCSP Singnature verification: OK!  
resp[0] certStatus: null  
resp[0] certSN: 585253  
Status overeni v OCSP odpovedi = 0
```


6 BEZPEČNOSTÍ ASPEKTY ŘEŠENÍ OCSP RESPONDERU

V průběhu předchozích kapitol jsme se zaměřili na vytvoření funkčního řešení s minimem vynaloženého úsilí. Předpokládáme, že OCSP Responder, když už se rozhodneme pro jeho vytvoření, bude začleněn do komplexního prostředí naší organizace. Responder se bude nacházet na rozhraní dvou (nebo více) sítí, externí a interní, jak je vidět například na obrázcích 3 a 1. Samotné RFC 2560 [6] v kapitole 5. rozebírá bezpečnostní aspekty OCSP protokolu. Při vytváření služby bychom měli brát v úvahu některé aspekty:

- pokusy o zahlcení služby (DOS Attack),
- problematiku předpřipravených odpovědí,
- nutnosti kontrolovat přijímané odpovědi,
- možnosti podvrhnutí opakované odpovědi klientovi.

Náš vlastní OCSP Responder podepisuje odpověď certifikátem, který jsme nastavili při konfiguraci a spoléhající se aplikace by měla vždy ověřit, že je tato odpověď podepsána právě tímto certifikátem. V konfiguraci dále nastavujeme tyto parametry dotýkající se bezpečnosti:

- časová platnost odpovědi,
- počet procesů, které jsou připraveny zpracovávat požadavky,
- podepisující klíč.

Systém (server), ve kterém je provozován OCSP Responder, by měl mít speciální bezpečnostní politiku zajišťující, že správci nebudou mít přístup např. k podepisovacímu klíči, nebo, že nebudou znát heslo, kterým je chráněn. Nabízí se zde možnost využití HSM modulů. Jak ochrana klíče heslem, tak i uložení klíče v HSM jsou modulem `ocspd` podporovány.

OCSP Responder jsme vybudovali pro účely využití v interních sítích, nestojíme tedy před problémem možného útoku z internetu, protože komunikace je navazována pouze „zevnitř“ a „z venku“ vidět není. Mnohem větší hrozbou pro nás mohou být interní zaměstnanci, případně správci síťové infrastruktury. Některé problémy můžeme řešit šifrovanou komunikací (např. HTTPS), v ostatních případech nám nezbude, než identifikovat rizika a přijmout opatření, která budou v mnoha případech pouze organizačního charakteru.

Zjednodušeně řečeno, vybudování OCSP Responderu a jeho provozování musí být plně v souladu s bezpečnostní politikou organizace. Bližší informace k bezpečnostnímu projektu je možné nalézt například na stránkách <http://www.27000.org/iso-27001.htm>.

ZÁVĚR

Cílem této práce bylo vybudovat funkční OCSP Responder s konsolidovanou databází CRL českých akreditovaných certifikačních autorit.

V teoretické části jsme se seznámili s normami souvisejícími s PKI a legislativními dokumenty. Vysvětlili jsme rozdíl mezi CRL a OCSP. Jejich datové struktury jsme popsali v jazyce ASN.1 a provedli kódování do binární podoby. Odkázali jsme se na oficiální zdroje z českých CA a zjistili, že není jednoduché ani levné naše případné aplikace napojit na jejich služby, které jsou i tak stále v plenkách (pokud vůbec). Pro vytvoření ukázkového řešení jsme zvolili OCSP Responder z balíčku projektu OpenCA a seznámili čtenáře s dalšími Open Source projekty poskytující moduly pro OCSP.

V praktické části jsme nainstalovali OCSP Responder z balíku OpenCA. Krok za krokem prošli nastavením, tak aby nakonec byl schopen odpovídat na dotazy ověřující platnost komerčních certifikátů od společností I.CA, PostSignum a eIdentity.

Po čas celé práce jsme používali výhradně volně dostupné programové vybavení, počínaje operačním systémem přes balíky pro vybudování OCSP Responderu a utility pro ladění, konče textovým procesorem \LaTeX . Nakonec jsme vytvořili ukázkový kód v jazyce Java za použití crypto knihoven Bouncy Castle a náš OCSP Responder touto testovací aplikací vyzkoušeli.

 \LaTeX

ZÁVĚR V ANGLIČTINĚ

The aim of this thesis was to build a functional OCSP Responder with consolidated CRL database of Czech accredited certification authorities.

In the theoretical part we learned about the standards associated with PKI and legislative documents. We explained the difference between CRL and OCSP. Their data structures are described in ASN.1 language and we made encoding of them into binary code. We bequeathed to the official sources of the Czech CA and found that it is not easy nor cheap to connect our eventual application to their services, which are still in its infancy (if at all). To create a sample solution we chose OCSP Responder package from Project OpenCA and familiarize readers with other Open Source projects providing modules for OCSP.

In the practical part we installed the OCSP Responder from OpenCA package. We went Step-by-step through setting and we build service to be able to answer to requests verifying the validity of commercial certificate from I.CA, PostSignum and eIdentity.

For the practical work of this theses we used only freely available software, from the operating system via packages for building the OCSP responder and utility for debugging, ending with a word processor \LaTeX . Finally, we created a sample code in Java

 \LaTeX

language using the Bouncy Castle crypto libraries and OCSP Responder and tried it by our application.

Reference

- [1] DOSTÁLEK, Libor, VOHNOUTOVÁ, Marta a KNOTEK, Miroslav. *Velký průvodce infrastrukturou PKI a technologií elektronického podpisu*. 2., aktualiz. vyd. Brno: Computer Press, 2009. 542 s. ISBN 978-80-251-2619-6.
- [2] *Certifikační politika pro komerční certifikáty - verze 3.1. První certifikační autorita, a. s.* [online]. 2013 [cit. 2013-07-10]. Dostupné z: http://www.ica.cz/Userfiles/files/politika/CP_KCv31.pdf
- [3] *Příloha č. 5 Certifikační politika PostSignum Public CA pro komerční osobní certifikáty*. [online]. verze 2.1. Červenec 2012 [cit. 2013-07-10]. Dostupné z: http://www.postsignum.cz/files/politiky/VCA_osobni.crt_v2-1.pdf
- [4] *ACAeID100.1 Certifikační politika - CC*. [online]. verze 2.2. Únor 2010 [cit. 2013-07-10]. Dostupné z: <http://www.acaeid.cz/cca2/cp-cc.pdf>
- [5] COOPER, et al. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile (RFC 5280)*. [online]. May 2008 [cit. 2013-07-10]. Dostupné z: <http://www.ietf.org/rfc/rfc5280.txt>
- [6] MYERS, M. et al *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP RFC 2560*. [online]. June 1999 [cit. 2013-07-06]. Dostupné z: <http://www.ietf.org/rfc/rfc2560.txt>
- [7] I.CA, a.s. (I.CA). *I.CA - Komerční certifikát*. [online]. 2013 [cit. 2013-06-06]. Dostupné z: <http://www.ica.cz/Komercni-certifikat>
- [8] *OCSP Responder*. [online]. 2011 [cit. 2013-07-06]. Dostupné z: <http://www.openca.org/projects/ocspd/>
- [9] *EJBCA*. [online]. 2012 [cit. 2013-02-07]. Dostupné z: <http://www.ejbca.org/>
- [10] Odbor Hlavního architekta eGovernment *Zákon č. 227/2000 Sb., o elektronickém podpisu*. [online]. 2013 [cit. 2013-02-07]. Dostupné z: <http://www.mvcr.cz/clanek/zakon-c-227-2000-sb-o-elektronickem-podpisu.aspx>
- [11] FIELDING, et al. *Hypertext Transfer Protocol – HTTP/1.1*. [online]. June 1999 [cit. 2013-07-11]. Dostupné z: <http://www.ietf.org/rfc/rfc2616.txt>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ASN.1	Abstract Syntax Notation One
BER	Basic Encoding Rules
CA	Certifikační autorita
CRL	Certificate Revocation List - seznam odvolaných certifikátů
DMZ	Demilitarizovaná zóna
DER	Distinguished Encoding Rules
DOS	Denial of Service
HTTP	Hypertext Transfer Protocol
HSM	Hardware Security Module
JDK	Java Development Kit
J2EE	Java Platform, Enterprise Edition
LDAP	Lightweight Directory Access Protocol
OCSP	Online Certificate Status Protocol
RFC	Request for Comments
XML	Extensible Markup Language

Seznam obrázků

Obr. 1. OCSP Responder	12
Obr. 3. Architektura certifikační autority od EJBCA	22
Obr. 1. Logo OpenCA.....	22
Obr. 2. Logo EJBCA	22
Obr. 1. Odvolaný certifikát	30
Obr. 2. Platný certifikát	33
Obr. 3. Bless Hex Editor	35
Obr. 1. Logo Legion of the Bouncy Castle	39
Obr. 2. Třída OCSPTestApp	39

Seznam tabulek

Tab. 1. Přehled univerzálních tagů	16
--	----

PŘÍLOHA P VII. OCSPTTESTAPP

Výpis 6: Zdrojový kód OCSPTestApp.java

```
package cz.sandor.bp.ocsp;

import java.io.BufferedOutputStream;
import java.io.DataOutputStream;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.math.BigInteger;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.security.Provider;
import java.security.Security;
import java.security.cert.X509Certificate;
import java.util.Vector;
import java.security.NoSuchProviderException;
import java.security.cert.CertificateFactory;

import org.bouncycastle.asn1.ASN1ObjectIdentifier;
import org.bouncycastle.asn1.DEROctetString;
import org.bouncycastle.asn1.ocsp.OCSPObjectIdentifiers;
import org.bouncycastle.asn1.x509.X509Extension;
import org.bouncycastle.asn1.x509.X509Extensions;
import org.bouncycastle.jce.provider.BouncyCastleProvider;
import org.bouncycastle.ocsp.BasicOCSPResp;
import org.bouncycastle.ocsp.CertificateID;
import org.bouncycastle.ocsp.OCSPException;
import org.bouncycastle.ocsp.OCSPReq;
import org.bouncycastle.ocsp.OCSPReqGenerator;
import org.bouncycastle.ocsp.OCSPResp;
import org.bouncycastle.ocsp.OCSPRespStatus;
import org.bouncycastle.ocsp.RevokedStatus;
import org.bouncycastle.ocsp.SingleResp;
import org.bouncycastle.ocsp.UnknownStatus;

public class OCSPTestApp {

    public static final int OCSP_NO_URL = -1;
    public static final int OCSP_VALID = 0;
    public static final int OCSP_REVOKED = 1;
    public static final int OCSP_UNKNOWN = 2;
    public static final int OCSP_ERROR = 3; // ostatni chyby

    private OCSPReq generateOCSPRequest(X509Certificate issuerCert,
        BigInteger serialNumber) {
        try {
            OCSPReqGenerator gen = new OCSPReqGenerator();

            CertificateID id = new CertificateID(CertificateID.HASH_SHA1,
                issuerCert, serialNumber);
            gen.addRequest(id);

            BigInteger nonce = BigInteger.valueOf(System.currentTimeMillis());

            Vector<ASN1ObjectIdentifier> oids = new Vector<ASN1ObjectIdentifier>();
            Vector<X509Extension> values = new Vector<X509Extension>();

            oids.add(OCSPObjectIdentifiers.id_pkix_ocsp_nonce);
            values.add(new X509Extension(false, new DEROctetString(nonce
                .toByteArray())));

            gen.setRequestExtensions(new X509Extensions(oids, values));

            return gen.generate();
        } catch (OCSPException e) {
            return null;
        }
    }
}
```



```

}

private OCSPResp sendOCSPRequest(String url, OCSPReq request) {
    try {
        URL u = new URL(url);
        HttpURLConnection con = (HttpURLConnection) u.openConnection();

        con.setRequestProperty("Content-Type", "application/ocsp-request");
        con.setRequestProperty("Accept", "application/ocsp-response");

        con.setConnectTimeout(5000);
        con.setDoOutput(true);

        OutputStream out = con.getOutputStream();
        DataOutputStream dataOut = new DataOutputStream(
            new BufferedOutputStream(out));

        dataOut.write(request.getEncoded());
        dataOut.flush();
        dataOut.close();

        if (con.getResponseCode() / 100 != 2) {
            System.out.println("Chyba pri spojeni s OCSP na URL '" + url
                + "': " + con.getResponseCode());
            return null;
        }

        InputStream in = (InputStream) con.getContent();
        return new OCSPResp(in);
    } catch (MalformedURLException e1) {
        System.out.println("Chybna OCSP URL: '" + url + "'.");
        return null;
    } catch (IOException e2) {
        System.out.println("Chyba IO - " + e2);
        return null;
    }
}

// public int validate(X509Certificate issuerCert, X509Certificate cert,
// String url) throws IOException
public int validate(X509Certificate issuerCert, BigInteger certSn,
    String url, X509Certificate ocspCert) throws IOException {
    // set BouncyCastle provider
    Provider provider = new BouncyCastleProvider();
    if (Security.getProvider("BC") == null) {
        Security.addProvider(provider);
    }

    // Vygenerovani pozadavku
    // OCSPReq request = generateOCSPRequest(issuerCert,
    // cert.getSerialNumber());
    OCSPReq request = generateOCSPRequest(issuerCert, certSn);

    // send request
    OCSPResp response = sendOCSPRequest(url, request);
    if (response == null) {
        return OCSP_ERROR;
    }

    // read response
    if (response.getStatus() != OCSPRespStatus.SUCCESSFUL) {
        System.out.println("OCSP response status " + response.getStatus()
            + ", URL '" + url + "'.");
        return OCSP_ERROR;
    }

    BasicOCSPResp basicResponse = null;
    try {
        basicResponse = (BasicOCSPResp) response.getResponseObject();
    } catch (OCSPException e) {
        System.out
            .println("Nemohu precis basic reponse object ze serveur '"
                + url + "' response: " + e);
        return OCSP_ERROR;
    }
}

```

```

try {
    if (!basicResponse.verify(ocspdCert.getPublicKey(),
        provider.getName())) {
        System.out.println("Chyba OCSP Signature verification na url '"
            + url + "'.");
        // return OCSP_ERROR;
        System.out.println(OCSP_ERROR);
    }
    else {
        System.out.println("OCSP Singnature verification: OK!");
    }
} catch (NoSuchProviderException e1) {
    System.out
        .println("Chyba overeni OCSP odpovedi. Provider neni dostupny.");
    return OCSP_ERROR;
} catch (OCSPException e2) {
    System.out.println("Nemohu overit OCSP response url '" + url
        + "': " + e2);
    return OCSP_ERROR;
}

SingleResp[] sresp = basicResponse.getResponses();
for (int i = 0; i < sresp.length; i++) {
    System.out.println("resp[" + i + "] certStatus: "
        + sresp[i].getCertStatus());
    System.out.println("resp[" + i + "] certSN: "
        + sresp[i].getCertID().getSerialNumber());
    CertificateID id = sresp[i].getCertID();
    if (id != null) {
        Object certStatus = sresp[i].getCertStatus();
        if (certStatus == null) {
            return OCSP_INVALID;
        }
        else if (certStatus instanceof RevokedStatus) {
            return OCSP_REVOKED;
        }
        else if (certStatus instanceof UnknownStatus) {
            return OCSP_UNKNOWN;
        }
    }
}
System.out.println("Chybny OCSP response status "
    + (sresp.length > 0 ? sresp[0].getCertStatus().getClass()
        .getCanonicalName() : ""));
return OCSP_ERROR;
}

public static X509Certificate loadCertificate(String filename) {
    try {
        FileInputStream is = new FileInputStream(filename);

        if (Security.getProvider("BC") == null)
            Security.insertProviderAt(new BouncyCastleProvider(), 1);

        CertificateFactory cf;
        try {
            cf = CertificateFactory.getInstance("X.509", "BC");
        } catch (NoSuchProviderException e) {
            System.out.println("NO provider BouncyCastle");
            cf = CertificateFactory.getInstance("X.509");
        }
        return (X509Certificate) cf.generateCertificate(is);
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    }
}

public static void main(String[] args) {

    String issuerCertPath = "/home/sandorm/Documents/SKOLA/BAKALARSKA-PRACE/CA/
        eidentity.cacert.pem";
    String ocspdCertPath = issuerCertPath;
    String url = "http://127.0.0.1:6060/";
    BigInteger certSn = null;

```

```
        if (args.length > 2) {
            issuerCertPath = args[0];
            ocspdCertPath = args[1];
            certSn = new BigInteger(args[2]);
            url = args[3];
        } else {
            System.out
                .println("Parametry jsou [OCSPDCertPath] [IssuerCertPath] [http://host:port]
                        [CertSerialNumber_Decimalne] !!!");
            return;
        }

        System.out.println("URL = " + url);

        X509Certificate issuerCert = loadCertificate(issuerCertPath);
        X509Certificate ocspdCert = loadCertificate(ocspdCertPath);

        OCSPTestApp testApp = new OCSPTestApp();
        int result;
        try {
            result = testApp.validate(issuerCert, certSn, url, ocspdCert);
            System.out.println("Status overeni v OCSP odpovedi = " + result);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

// platny PostSignum: 585253
// revokovany PostSignum:573211
// revokovany eIdentity: 2140297227 (Hexadecimalne: 0x7f92580b)
```

PŘÍLOHA P VIII. VÝPIS SÍŤOVÉ KOMUNIKACE OCSP POŽADAVKU- ODPOVĚDI

```

00000000 50 4f 53 54 20 2f 20 48 54 54 50 2f 31 2e 31 0d POST / HTTP/1.1.
00000010 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65 3a 20 61 .Content -Type: a
00000020 70 70 6c 69 63 61 74 69 6f 6e 2f 6f 63 73 70 2d pplicati on/ocsp-
00000030 72 65 71 75 65 73 74 0d 0a 41 63 63 65 70 74 3a request. .Accept:
00000040 20 61 70 70 6c 69 63 61 74 69 6f 6e 2f 6f 63 73 applica tion/ocs
00000050 70 2d 72 65 73 70 6f 6e 73 65 0d 0a 55 73 65 72 p-respon se..User
00000060 2d 41 67 65 6e 74 3a 20 4a 61 76 61 2f 31 2e 37 -Agent: Java/1.7
00000070 2e 30 0d 0a 48 6f 73 74 3a 20 31 32 37 2e 30 2e .0..Host : 127.0.
00000080 30 2e 31 3a 36 30 36 30 0d 0a 43 6f 6e 6e 65 63 0.1:6060 ..Connec
00000090 74 69 6f 6e 3a 20 6b 65 65 70 2d 61 6c 69 76 65 tion: ke ep-alive
000000A0 0d 0a 43 6f 6e 74 65 6e 74 2d 4c 65 6e 67 74 68 ..Conten t-Length
000000B0 3a 20 39 36 0d 0a 0d 0a : 96....
000000B8 30 5e 30 5c 30 41 30 3f 30 3d 30 09 06 05 2b 0e 0^0\0A0? 0=0...+.
000000C8 03 02 1a 05 00 04 14 19 27 81 90 df ed 57 3e 51 ..... '....W>Q
000000D8 8a c4 06 f1 b8 5a 5d 84 6f b4 3e 04 14 e5 40 2b .....Z]. o.>...@+
000000E8 62 63 25 e5 90 e3 00 03 28 2a 50 17 71 f9 61 de bc%.... (*P.q.a.
000000F8 19 02 04 6d a4 fd 63 a2 17 30 15 30 13 06 09 2b ...m..c. .0.0...+
00000108 06 01 05 05 07 30 01 02 04 06 01 3f 1f 44 0e a1 .....0.. ...?.D..

00000000 48 54 54 50 2f 31 2e 30 20 32 30 30 20 4f 4b 0d HTTP/1.0 200 OK.
00000010 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65 3a 20 61 .Content -Type: a
00000020 70 70 6c 69 63 61 74 69 6f 6e 2f 6f 63 73 70 2d pplicati on/ocsp-
00000030 72 65 73 70 6f 6e 73 65 0d 0a 43 6f 6e 74 65 6e response ..Conten
00000040 74 2d 54 72 61 6e 73 66 65 72 2d 45 6e 63 6f 64 t-Transf er-Encod
00000050 69 6e 67 3a 20 42 69 6e 61 72 79 0d 0a ing: Bin ary..
0000005D 43 6f 6e 74 65 6e 74 2d 4c 65 6e 67 74 68 3a 20 Content- Length:
0000006D 32 36 36 36 0d 0a 2666..
00000073 44 61 74 65 3a 20 4a 75 6e 20 20 37 20 31 35 3a Date: Jun 7 15:
00000083 33 32 3a 30 31 20 32 30 31 33 20 47 4d 54 0d 0a 32:01 20 13 GMT..
00000093 45 78 70 69 72 65 73 3a 20 4a 75 6e 20 20 37 20 Expires: Jun 7
000000A3 31 35 3a 33 35 3a 30 31 20 32 30 31 33 20 47 4d 15:35:01 2013 GM
000000B3 54 0d 0a T..
000000B6 0d 0a ..
000000B8 30 82 0a 66 0a 00 a0 82 0a 60 30 82 0a 5c 06 09 0..f.... '0..\..
000000C8 2b 06 01 05 05 07 30 01 01 04 82 0a 4d 30 82 0a +.....0. ....M0..
000000D8 49 30 82 01 22 a1 81 8c 30 81 89 31 0b 30 09 06 IO..."... 0..1.0..
000000E8 03 55 04 06 13 02 43 5a 31 17 30 15 06 03 55 04 .U....CZ 1.0...U.
000000F8 08 13 0e 43 7a 65 63 68 20 52 65 70 75 62 6c 69 ...Czech Republi
00000108 63 31 0d 30 0b 06 03 55 04 07 13 04 5a 6c 69 6e c1.0...U ...Zlin
00000118 31 0c 30 0a 06 03 55 04 0a 13 03 55 54 42 31 0c 1.0...U. ...UTB1.
00000128 30 0a 06 03 55 04 0b 13 03 46 41 49 31 12 30 10 0...U... .FAI1.0.
00000138 06 03 55 04 03 13 09 4f 43 53 50 44 2d 30 30 31 ..U....0 CSPD-001
00000148 31 22 30 20 06 09 2a 86 48 86 f7 0d 01 09 01 16 1"0 ...*. H.....
00000158 13 6f 63 73 70 64 2d 30 30 31 40 73 61 6e 64 6f .ocspd-0 01@sando
00000168 72 2e 63 7a 18 0f 32 30 31 33 30 36 30 37 31 35 r.cz..20 13060715
00000178 33 32 30 31 5a 30 67 30 65 30 3d 30 09 06 05 2b 3201Z0g0 e0=0...+
00000188 0e 03 02 1a 05 00 04 14 19 27 81 90 df ed 57 3e ..... '....W>
00000198 51 8a c4 06 f1 b8 5a 5d 84 6f b4 3e 04 14 e5 40 q.....Z] .o.>...@
000001A8 2b 62 63 25 e5 90 e3 00 03 28 2a 50 17 71 f9 61 +bc%.... (*P.q.a
000001B8 de 19 02 04 6d a4 fd 63 80 00 18 0f 32 30 31 33 ....m..c ....2013
000001C8 30 36 30 37 31 35 33 32 30 31 5a a0 11 18 0f 32 06071532 01Z....2
000001D8 30 31 33 30 36 30 37 31 35 33 35 30 31 5a a1 17 01306071 53501Z...
000001E8 30 15 30 13 06 09 2b 06 01 05 05 07 30 01 02 04 0.0...+. ....0...
000001F8 06 01 3f 1f 44 0e a1 30 0d 06 09 2a 86 48 86 f7 ...?.D..0 ...*.H..
00000208 0d 01 01 0b 05 00 03 82 02 01 00 50 18 cd ec d2 ..... .P....
00000218 f8 c2 52 c8 c7 05 b4 7e 49 60 e1 6e e8 88 35 ca ..R....~ I'.n..5.
00000228 e6 55 b3 e2 3e 8c b2 37 17 d8 59 15 a4 5b ee 90 U...>..7 ...Y...[.
00000238 51 15 83 f7 57 93 5d a1 53 88 88 a5 21 1a 63 8f q.....W.]. S...!.c.
00000248 28 2e 07 73 98 4b 93 6e 35 61 96 2e f9 d0 eb e7 (.s.K.n 5a.....
00000258 0b 67 24 72 2f ce e2 38 de 78 10 8f be 28 85 88 .g$r/../8 .x...(.
00000268 d7 74 9e 8a 90 ca ff 3c e7 00 54 c1 3a 0c a1 28 .t.....< ..T!:(
00000278 eb 71 de c9 1f 17 f8 c8 ea 03 c6 21 9f 7b 00 8f .q..... .!..{..
00000288 f4 23 b2 d4 bb 4e a0 68 5d 3a 4b 5a 9c 90 5e fd .#...N.h ]:KZ..^
00000298 5f 4b e4 b8 25 7a e4 8d 9b 95 7c 65 ce 5f 54 b6 _K..%z.. ..|e._T.
000002A8 5e 39 ba c2 39 17 82 b7 2c d8 77 21 24 33 fa 94 ^9..9... ,w!$3..
000002B8 18 e3 8b 4d bd fd e3 2e d2 e7 c8 ee ae 68 16 fe ...M.... .h..
000002C8 c9 50 d6 a2 96 3b e1 4c e0 79 e8 34 01 2d 59 f6 .P...;.L .y.4.-Y.
000002D8 e5 eb 0f 1e 99 c4 d8 3c ee 51 ac a3 fe 28 a8 cf .....< .Q...(.
000002E8 28 3c 7c 4e 38 49 2c fd 99 b9 33 a8 82 26 0a f7 (<|N8I,. ..3..&..
000002F8 aa b0 f5 ea 22 de 9a cd 05 ad c4 99 8f b9 f4 99 ...."....

```

```
00000308 bf 45 a9 a1 38 0e 35 93 86 7b 63 d5 a1 e7 ad 99 .E..8.5. .{c.....
00000318 84 01 59 92 5d aa e3 e1 23 70 88 8a 50 ac cd d9 ..Y.]... #p..P...
00000328 0c 57 11 45 36 c2 32 62 5a aa aa ea d8 e0 8e 14 .W.E6.2b Z.....
00000338 fb d7 b2 01 ea b9 e8 d6 0a 91 89 5b 3e 5d 17 79 .....[>].y
00000348 89 2e c8 54 87 70 b2 5c a2 ca d0 08 3b 7d 05 d1 ...T.p.\ ....;}.
00000358 21 24 dc 2d a9 97 e2 0e 7f 62 b8 c0 f6 d1 fc 21 !$.-.... .b.....!
00000368 93 bc 2e 2e 04 95 06 5b 1e fc b7 0c 31 97 50 5a .....[ ....1.PZ
00000378 1b 35 4b 10 78 80 89 39 cb 14 f2 5c a0 59 4e eb .5K.x..9 ...\.YN.
00000388 77 16 e6 3c 92 15 9c 04 3e 80 6c 97 5b f5 6a c8 w...<.... >.l.[.j.
00000398 02 52 7b 95 84 cc 6f 31 8f 2b fb 8e 7b 58 98 84 .R{...o1 .+...{X..
000003A8 55 91 05 03 18 9a 8c bb 4d cb 49 d6 4f 0b 21 87 U..... M.I.O.!.
000003B8 62 08 57 a0 b8 8f f5 fc 2b 68 6a 6b 11 75 8b 96 b.W..... +hjk.u..
000003C8 c9 67 62 44 65 01 7d 06 57 bd 5a af 79 e2 f5 a5 .gbDe.}. W.Z.y...
000003D8 64 43 62 c5 3a 5a 93 7c 69 1c 17 d3 07 52 35 e7 dCb.:Z.| i....R5.
000003E8 42 ba 2e 33 87 83 be 45 6a 08 1c 88 1f 32 5b ed B..3...E j....2[.
000003F8 36 1c af fa 43 a9 0d 98 96 88 50 a6 1a a8 ab 97 6...C... .P.....
00000408 ea 33 fe b4 02 d9 5d c9 cc c8 43 a0 82 07 0b 30 .3....]. .C....0
00000418 82 07 07 30 82 07 03 30 82 04 eb a0 03 02 01 02 ...0....0 .....
00000428 02 01 01 30 0d 06 09 2a 86 48 86 f7 0d 01 01 05 ...0....* .H.....
00000438 05 00 30 81 89 31 0b 30 09 06 03 55 04 06 13 02 ..0..1.0 ...U....
00000448 43 5a 31 17 30 15 06 03 55 04 08 13 0e 43 7a 65 CZ1.0... U....Cze
00000458 63 68 20 52 65 70 75 62 6c 69 63 31 0d 30 0b 06 ch Repub lic1.0..
00000468 03 55 04 07 13 04 5a 6c 69 6e 31 0c 30 0a 06 03 .U....Zl in1.0..
00000478 55 04 0a 13 03 55 54 42 31 0c 30 0a 06 03 55 04 U....UTB 1.0...U.
00000488 0b 13 03 46 41 49 31 0f 30 0d 06 03 55 04 03 13 ...FAI1. 0...U...
00000498 06 55 54 42 2d 42 50 31 25 30 23 06 09 2a 86 48 .UTB-BP1 %0#...*.H
000004A8 86 f7 0d 01 09 01 16 16 6d 61 72 65 6b 2e 73 61 ..... marek.sa
000004B8 6e 64 6f 72 40 67 6d 61 69 6c 2e 63 6f 6d 30 1e ndor@gma il.com0.
000004C8 17 0d 31 33 30 36 30 31 31 38 33 35 33 31 5a 17 ..130601 183531Z.
000004D8 0d 31 34 30 36 30 31 31 38 33 35 33 31 5a 30 81 .1406011 83531Z0.
000004E8 89 31 0b 30 09 06 03 55 04 06 13 02 43 5a 31 17 .1.0...U ...CZ1.
000004F8 30 15 06 03 55 04 08 13 0e 43 7a 65 63 68 20 52 0...U... .Czech R
00000508 65 70 75 62 6c 69 63 31 0d 30 0b 06 03 55 04 07 epublic1 .0...U...
00000518 13 04 5a 6c 69 6e 31 0c 30 0a 06 03 55 04 0a 13 ..Zlin1. 0...U...
00000528 03 55 54 42 31 0c 30 0a 06 03 55 04 0b 13 03 46 .UTB1.0. .U....F
00000538 41 49 31 12 30 10 06 03 55 04 03 13 09 4f 43 53 AI1.0... U....OCS
00000548 50 44 2d 30 30 31 31 22 30 20 06 09 2a 86 48 86 PD-0011" 0 ...*.H.
00000558 f7 0d 01 09 01 16 13 6f 63 73 70 64 2d 30 30 31 ..... cspd-001
00000568 40 73 61 6e 64 6f 72 2e 63 7a 30 82 02 22 30 0d @sador. cz0..."0.
00000578 06 09 2a 86 48 86 f7 0d 01 01 01 05 00 03 82 02 ...*.H... .....
00000588 0f 00 30 82 02 0a 02 82 02 01 00 e7 4f 37 b7 b3 ..0..... ....07..
00000598 49 e1 e8 00 9e f1 9f 83 7c 3a ec e5 92 c5 1b 9b I..... |:.....
000005A8 0c 96 4e 3f a5 22 42 07 47 e0 2c 05 48 eb 25 d0 ..N?."B. G.,.H.%.
000005B8 be 37 b6 e6 3b 34 c6 f1 b3 cf a6 e1 d3 34 b7 37 .7...;4... ....4.7
000005C8 3b 14 1c f0 67 fe f7 6d b2 90 bf 89 2e 29 e9 40 ;...g..m .....).@
000005D8 90 3c 29 bb f4 e7 5a 19 d3 c0 a5 67 a7 bc 7e 38 .<)...Z. ...g...~8
000005E8 ae 9d 79 ba da a4 fe c9 5f 3f f2 31 67 c0 a3 26 .y..... _?.1g...&
000005F8 bd 3c 49 21 af 4a a2 90 90 28 e1 92 ec 6c ee e0 .<I!.J... (.1...
00000608 75 18 e1 cc cb 7f 2d 94 7e 4c 91 b8 9c 19 e0 2e u.....- .~L.....
00000618 3b bc 65 ee 94 4c cb 8b 80 ba 91 f3 65 f6 c1 d9 ;.e..L... ....e...
00000628 0e 99 2b 4f 93 66 d2 14 d9 4e 76 26 5d 06 f3 30 .+0.f... .Nv&]..0
00000638 78 86 b6 90 03 c9 5b ae 6d 93 db 44 fd 91 78 d1 x.....[. m..D...
00000648 26 c8 74 cf 7a 74 a7 b5 e6 1d 9d 13 a3 35 81 0d &.t.zt... ....5..
00000658 a0 6e 49 be 06 c1 2b de 72 ed fe 06 5d 7e da 31 .nI...+. r...~.1
00000668 6e f5 00 76 66 06 86 83 60 20 e7 e0 c3 04 87 7b n...vf... ' .....{
00000678 26 c4 4f 41 ad 33 e2 f8 aa 3e 4f 70 d4 83 6c 85 &.0A.3... .>Op..1.
00000688 a5 13 af 9e 4d 1d 0d 01 b5 b4 f9 ad 0d d7 d8 a5 ....M... .....
00000698 cc ad 6d 9e b5 37 ef 60 3d ad af 69 db fc d7 27 ..m..7.' =..i...'
000006A8 b2 38 08 03 bc 5c 75 cf 95 a0 a0 cb 04 8c ad ea .8...\u. ....
000006B8 9c 68 e8 fc 66 e8 4a e8 f9 b8 2d 28 db fe 3d 42 .h..f.J. ...-(.=B
000006C8 60 9e b0 9c 28 0d 18 54 ab 0a bc 0e d6 bb 86 bd '...(.T .....
000006D8 39 80 c0 eb 52 91 ac b8 db 28 05 8b 03 cd 70 93 9...R... (.....p.
000006E8 cc 57 76 9e 08 f8 9f bd 6e 38 9a 2e ec 59 ca 27 .Wv..... n8...Y.'
000006F8 86 48 aa 37 f7 e1 a4 bd 87 c7 20 ad d1 10 87 2c .H.7.... . ....,
00000708 7e 44 ee eb c9 09 0a f2 b0 b3 33 c0 6e 32 1b 77 ~D..... ..3.n2.w
00000718 dc f2 eb 31 59 9a 3e 54 78 b6 31 f5 ca 72 5d f0 ...1Y.>T x.1..r].
00000728 11 1e c6 ad 89 f1 96 aa f7 c5 ea 44 b6 21 0c fd ..... .D!..
00000738 a1 fb 7e 52 c6 5d 35 69 a9 4e 18 dc 91 fc c1 bb ..~R.]5i .N.....
00000748 89 3f f4 39 04 76 fb f2 5e e1 37 7e 7c 2c ff 78 .?.9.v... ^.7~|,x
00000758 4e 02 2a fe 69 81 d9 3c 60 86 4c 88 ae 7c b4 85 N*.i..< 'L...|..
00000768 1a 68 af b6 84 1e 3e 76 ef a3 dc 5a 75 df 9c 54 .h...>v ...Zu..T
00000778 90 e2 c7 36 ef db 90 94 36 68 51 ae df fc c1 95 ...6.... 6hQ....
00000788 ac a5 c4 4b 39 db 74 d0 2d 42 5f 02 03 01 00 01 ...K9.t. -B_....
00000798 a3 82 01 72 30 82 01 6e 30 09 06 03 55 1d 13 04 ...r0..n 0...U...
```

```
000007A8 02 30 00 30 11 06 09 60 86 48 01 86 f8 42 01 01 .0.0... ' .H...B..
000007B8 04 04 03 02 06 40 30 2b 06 09 60 86 48 01 86 f8 .....@0+ ..'.H...
000007C8 42 01 0d 04 1e 16 1c 54 69 6e 79 43 41 20 47 65 B.....T inyCA Ge
000007D8 6e 65 72 61 74 65 64 20 43 65 72 74 69 66 69 63 nerated Certific
000007E8 61 74 65 30 1d 06 03 55 1d 0e 04 16 04 14 5a 78 ate0...U .....Zx
000007F8 8c 04 2b f7 9e 27 c1 90 15 db b5 9a d7 46 31 77 ..+...'. ....F1w
00000808 ae 5e 30 81 be 06 03 55 1d 23 04 81 b6 30 81 b3 .^0...U .#...0..
00000818 80 14 0e 38 0a 00 44 2c 02 c1 f7 31 91 19 20 32 ...8...D, ...1.. 2
00000828 c1 40 b2 1f 90 cf a1 81 8f a4 81 8c 30 81 89 31 .@..... ....0..1
00000838 0b 30 09 06 03 55 04 06 13 02 43 5a 31 17 30 15 .0...U... ..CZ1.0.
00000848 06 03 55 04 08 13 0e 43 7a 65 63 68 20 52 65 70 ..U....C zech Rep
00000858 75 62 6c 69 63 31 0d 30 0b 06 03 55 04 07 13 04 ublic1.0 ...U....
00000868 5a 6c 69 6e 31 0c 30 0a 06 03 55 04 0a 13 03 55 Zlin1.0. ..U....U
00000878 54 42 31 0c 30 0a 06 03 55 04 0b 13 03 46 41 49 TB1.0... U....FAI
00000888 31 0f 30 0d 06 03 55 04 03 13 06 55 54 42 2d 42 1.0...U. ...UTB-B
00000898 50 31 25 30 23 06 09 2a 86 48 86 f7 0d 01 09 01 P1%0#...* .H.....
000008A8 16 16 6d 61 72 65 6b 2e 73 61 6e 64 6f 72 40 67 ..marek. sandor@g
000008B8 6d 61 69 6c 2e 63 6f 6d 82 09 00 a4 60 2a 2d b3 mail.com ....'*-
000008C8 f1 07 6d 30 21 06 03 55 1d 12 04 1a 30 18 81 16 ..m0!...U .....0...
000008D8 6d 61 72 65 6b 2e 73 61 6e 64 6f 72 40 67 6d 61 marek.sa ndor@gma
000008E8 69 6c 2e 63 6f 6d 30 1e 06 03 55 1d 11 04 17 30 il.com0. ..U....0
000008F8 15 81 13 6f 63 73 70 64 2d 30 30 31 40 73 61 6e ...ocspd -001@san
00000908 64 6f 72 2e 63 7a 30 0d 06 09 2a 86 48 86 f7 0d dor.cz0. ...*H...
00000918 01 01 05 05 00 03 82 02 01 00 42 29 ae b4 09 ce .....B)....
00000928 43 a3 91 6c a5 14 7a 32 09 ca e8 f1 4b a0 f2 0b C...l..z2 ....K...
00000938 f1 2f 07 e6 0f a6 6b 19 30 e8 0d 6b f0 1c b6 58 ./....k. 0...k...X
00000948 96 7c 8f 55 c0 21 55 40 70 26 f7 5b 03 ad 29 a8 .|.U.!U@ p&.[.]).
00000958 2f f8 0f 35 61 65 a7 a4 0e 4b 67 bc 50 30 68 c7 /.5ae... .Kg.P0h.
00000968 62 d9 0f e0 49 5b ce 78 f9 2e 1d b6 f2 cc a9 3e b...I[.x .....>
00000978 fc 2a 59 fe 26 7b 61 0f 37 a9 1a 1f b2 37 2f 7b .*Y.&{a. 7....7/{
00000988 01 81 a1 eb 0b 78 81 f9 17 33 3d 68 aa 1d 61 72 .....x... .3=h...ar
00000998 68 c2 4a 97 ca 85 ef c7 05 af 84 ca 90 c3 3a 8a h.J..... .....:
000009A8 06 30 e8 85 68 16 e6 ba c4 13 94 f8 b1 d4 9a 71 .0...h... .....q
000009B8 f9 57 14 c3 b4 2f cc ee 27 99 0e 78 7a 66 58 0b .W.../. '...zzfX.
000009C8 a7 53 b8 05 b3 73 d3 43 2c 7b cf 80 9b d0 f8 8e .S....s.C ,{.....
000009D8 d9 fc b5 a1 24 4b 9e bd 03 80 3d f7 23 12 eb e2 ....$K... ..=#...
000009E8 9e 8a f1 3a ba 67 93 57 5e ca ff 6a 4b 06 70 96 ....g.W ^..jK.p.
000009F8 39 90 11 1c 77 71 3f 2a 32 90 1c c4 2b 02 d6 fc 9...wq?* 2...+...
00000A08 b7 40 21 5e d0 e2 b7 c1 d8 f8 f9 16 ae ab 1e b4 .@!~.... .....
00000A18 c5 a3 45 8e 60 7d db 4d 21 88 64 eb 5b b5 3e d5 ..E.'}.M !.d.[.>.
00000A28 d2 c0 34 d7 41 62 38 32 29 ba f6 28 3d 76 da 29 ..4.Ab82 )..(<=v.)
00000A38 8e da 4e 60 b9 f7 61 3b 68 c2 f9 81 70 3a 0c cf ..N'.a; h...p:...
00000A48 a3 fc 1c b3 4c 6d 92 b2 8a 42 68 71 7f 6c 0b 30 ....Lm... .Bhq.1.0
00000A58 bc c3 f9 d9 16 ad 1c 90 fd df f2 46 42 7d b8 a1 ..... ..FB}...
00000A68 e9 5e a3 f0 ed 22 60 0a bc cd 54 89 92 52 7e e7 .^...."'. ..T...R~.
00000A78 7a 26 4b 86 87 4c df 48 70 e6 91 b6 1b da 1e b3 z&K...L.H p.....
00000A88 cb 61 d4 52 15 34 11 a7 8a 3d 1f 31 d2 31 21 c7 .a.R.4... .=.1.1!.
00000A98 cd 13 24 26 ef 6a 50 26 05 2f a7 ac 00 21 57 5f ..$&.jP& ./...!W_
00000AA8 d9 87 c1 ab 79 22 78 5c 88 08 e1 c2 e3 36 98 de ....y"x\ .....6..
00000AB8 30 b6 fa 50 52 97 78 dc e1 32 dc 81 9c 59 cd 6e 0..PR.x. .2...Y.n
00000AC8 36 c3 0f 64 cb 25 1b b5 08 ae ad 6a 5e 52 89 5c 6..d.%... ..j~R.\
00000AD8 ea 8c 00 63 ac d7 25 5b e4 a4 92 4f 2c 71 70 8d ...c...%[ ...0,qp.
00000AE8 76 1d 71 a0 0d 66 61 93 d0 79 0b b5 24 8b 02 ee v.q...fa. .y...$.
00000AF8 e6 1b 40 1c 47 2a 40 8e ce d4 5c a0 09 cf 9a b2 ..@.G*@. ..\.....
00000B08 bf ba 74 86 e7 fb 60 a9 9e 92 4b a2 df 27 a3 57 ..t...'. ..K...'.W
00000B18 65 0e 8d 78 90 ca 19 23 d3 eb e...x...# ..
```