Slovenská technická univerzita v Bratislave
Fakulta Elektrotechniky a Informatiky
Študijný odbor INFORMATIKA

Radovan Semančík

Certifikačná autorita pre malé organizácie

Diplomová práca

Vedenie diplomovej práce: doc. Ing. Margaréta Kotočová, CSc.
december 1999

Anotácia

Slovenská technická univerzita v Bratislave
Fakulta Elektrotechniky a Informatiky
Študijný odbor: INFORMATIKA
Autor: Radovan Semančík
Diplomová práca: Certifikačná autorita pre malé organizácie
Vedenie diplomovej práce: doc. Ing. Margaréta Kotočová, CSc.
december 1999

Práca sa zaoberá problémami bezpečnosti počítačových sietí z hľadiska kryptografie a kryptografickej infraštruktúry. Ponúka úvod do problematiky kryptografickej infraštruktúry a systémov certifikačných autorít. V dokumente je uvedená analýza a návrh jednoduchého systému certifikačnej autority pre malé organizácie. Tento systém je založený na báze technológií WWW a poskytuje základnú funkčnosť certifikačnej autority.

Systém je implementovaný v prostredí operačného systému UNIX s použitím WWW servera Apache a aplikačnej kostry Jewels ako základných stavebných prvkov. Na realizáciu kryptografických transformácií bola použitá kryptografická knižnica OpenSSL. Implementovaný systém je schopný evidovať žiadosti o certifikát, vydávať a zneplatňovať certifikáty. Správca systému môže definovať operátorov systému a ich práva. Systém bol testovaný pre spoluprácu s prehliadačmi Netscape Communicator 4.x a Miscrosoft Internet Explorer 4.x.

Anotation

Slovak University of Technology, Bratislava
Faculty of Electrical Engeneering and Information Technology
Degree Course: INFORMATICS
Author: Radovan Semančík
Thesis: Certification authority for small organisations
Supervisor: doc. Ing. Margaréta Kotočová, CSc.
1999 december

This document is concerning security problems of computer networking seen from the point of cryptography and cryptoraphic infrastructure. It includes an introduction to public key infrastructure and certificate authority systems. An analysis and desing of simple certificate authority system for small organisations is main part of this document. This certificate authority system is based on WWW technology.

System is designed and implemented for UNIX operating systems, using Apache WWW server and Jewels application framework as key components. OpenSSL cryptographic library is used for cryptographic transformations. Implemented systems is able to accept certificate signing requests, issue and revoke certificates. System administrator can define certificate authority operators and their access rights. System was tested with Nescape Communicator 4.x and Microsoft Internet Explorer 4.x WWW browsers.

ZADANIE .....

Vyhlásenie

Prehlasujem, že diplomovú prácu som vypracoval sám a inú ako uvedenú literatúru som nepoužil.

V Bratislave dňa 13.12.1999

Radovan Semančík

Obsah

1 Úvod

Rozmach dátových sietí a distribuovaných systémov spôsobil prudký skok v oblasti informačných technológií. Geograficky vzdialené aplikácie dokážu spolu komunikovať, systémy rôznych výrobcov dokážu spolupracovať v otvorenenj komunikačnej sieti a fyzické umiestenenie koncového používateľa vzhľadom na systém poskytujúci služby má čoraz menší význam. Takéto verejné, otvorené a geograficky rozľahlé systémy však priniesli aj niekoľko vážnych problémov. Jedným z nich je problém bezpečnosti prenášaných a uložených informácií. Dôležitým aspektom bezpečnosti otvorených komunikačných sietí bolo dokazovanie identity používateľov - autentifikácia.

V roku 1976 sa začala písať nová éra kryptografie a informačnej bezpečnosti. V tomto roku Whitfield Diffie a Martin Hellman predstavili verejnosti koncept asymetrickej kryptografie. O rok neskôr, v roku 1977, Ron Rivest, Adi Shamir a Leonar Adleman priviedli na svet jeden z doteraz najpoužívanejších asymetrických kryptosystémov - RSA [2]. Od tých čias sa asymetrické kryptosystémy uplatňujú najmä v oblasti informačnej bezpečnosti ako základný kameň pre silnú distribuovanú autentifikáciu a výmenu kľúčov. Napriek tomu však veľkej väčšine prenášaných informácií chýba akákoľvek ochrana. Väčšina informácií prúdiacich po komunikačných linkách Internetu, ako aj iných globálnych a podnikových sietí, je prenášaná v otvorenom tvare, bez kryptografickej ochrany. Množstvo autenfikačných systémov sa spolieha len na statické heslá. Existuje množstvo systémov, ktoré umožňujú zvýšiť úroveň bezpečnosti klasických systémov, často sa však jedná len o nesystémové zachraňovanie nepriaznivej situácie.

Infraštruktúra certifikačných autorít má ambície stať sa jednou z kľúčových častí riešenia bezpečnostných problémov globálnych sietí. Je to riešenie prijaté veľkými softvérovými firmami a tak dodávané v softvérových produktoch miliónom používateľov. Je to riešenie flexibilné, prispôsobiteľné lokálnym podmienkam a požiadavkám. Hierarchia certifikačných autorít v hrubých rysoch zodpovedá hierarchií prirodzených autorít, preto sa ľahko implementuje v organizovaných spoločnostiach.

2 Teoretický úvod

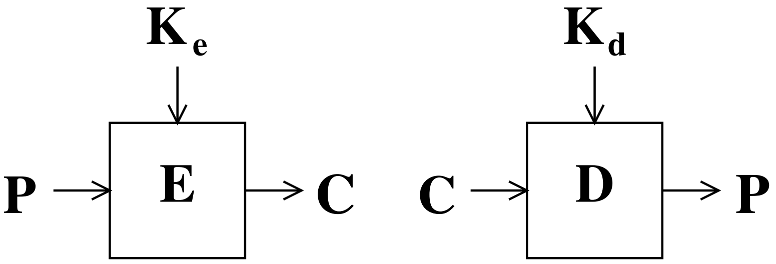
Za účelom uvedenia čitateľa do problému kryptografickej infraštruktúry a certifikačných autorít budú v nasledujúcich kapitolách uvedené základy kryptografie a príbuzných oblastí. Taktiež budú uvedené jednoduché príklady použitia týchto techník.

2.1 Základy kryptografie

Kryptografia už od čias staroveku slúžila na ochranu citlivých informácií. Najväčší rozvoj však dosiahla až s nástupom výpočtovej techniky. Výkonné počítačové systémy umožnili realizovať nesmierne zložité algoritmy a kryptografia sa stáva samostatným vedným odborom.

2.1.1 Šifra

Šifra, ako ju poznáme z klasickej kryptografie, pozostáva zo šifrovacieho a dešifrovacieho algoritmu. Šifrovací algoritmus transformuje správu vo forme otvoreného textu (P) na zašifrovanú správu (C). Na túto transformáciu je použitý šifrovací kľúč (K_e) [2]. Dešifrovací algoritmus naopak transformuje šifrovanú správu (C) na správu v otvorenom tvare (P) pomocou dešifrovacieho kľúča (K_d). Schematicky je šifra zobrazené na obrázku 1.



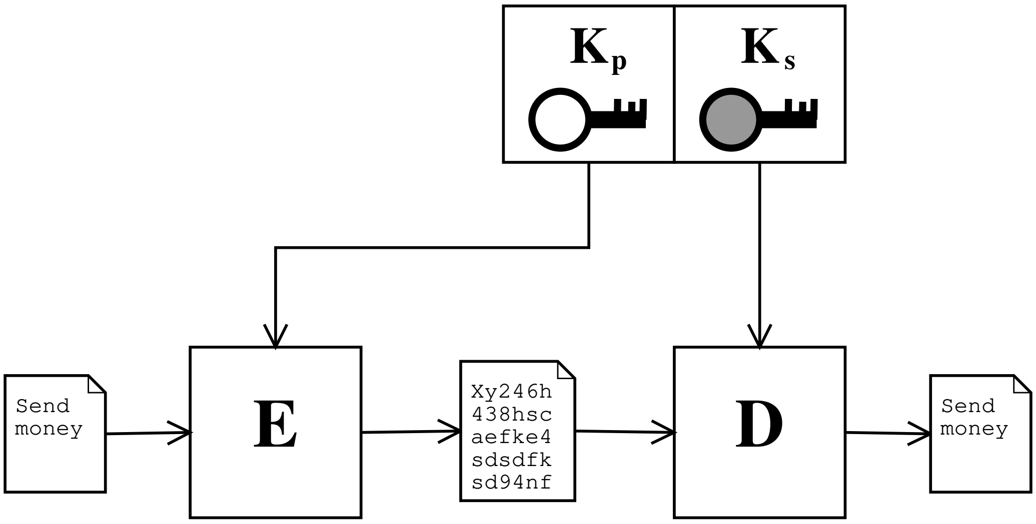
Obrázok 1: Šifra

Správa v šifrovanom tvare (C) je čitateľná len pre toho, kto vlastní dešifrovací kľúč (K_d). Šifrovací aj dešifrovací algoritmus sú vo väčšine prípadov verejne k dispozícií a jediná tajná informácia sú použité kľúče.

Symetrické šifry Klasické historické šifry a aj mnoho súčasných šifier používa rovnaký kľúč na šifrovanie aj dešifrovanie ($K_e = K_d$). Kľúč v tomto prípade predstavuje zdieľané tajomstvo medzi komunikujúcimi stranami. Ktokoľvek, kto vlastní kľúč, je schopný správu prečítať. Tieto šifry sa nazývajú symetrické alebo šifry s tajným kľúčom, keďže kľúč predstavuje zdieľané tajomstvo a na jeho utajení je založená bezpečnosť celej šifry. Väčšinou je nutné pred začatím komunikácie dohodnúť spoločný kľúč pomocou bezpečného kanálu. Medzi súčasných zástupcov symetrických šifier patria dobre známe šifry DES, RC4, IDEA, BLOWFISH a iné [2].

Asymetrické šifry Na druhej strane pri asymetrických šifrách je šifrovací a dešifrovací kľúč rozdielny ($K_p \neq K_s$)[1]. Šifrovací kľúč sa nazýva verejný kľúč a dešifrovací kľúč sa nazýva súkromný kľúč. Pri asymetrických šifrách je zaujímavý fakt, že ten kto správu zašifruje, už ju ani sám nevie prečítať. Šifrovanie je jednosmerná operácia. Na prečítanie zašifrovanej správy je potrebný súkromný kľúč. Verejný kľúč je teda možné voľne šíriť a ktokoľvek bude potrebovať poslať správu, ním môže správu zašifrovať. Nikto iný okrem majiteľa súkromného kľúča však správu neprečíta.

Súkromný a verejný kľúč sú na sebe matematicky závislé ($K_p = f(K_s)$) . Pri tom však platí, že pomocou verejného kľúča nie je možné zistiť súkromný kľúč. Verejný a súkromný kľúč, ktoré patria k sebe sa nazývajú kľúčový pár. Základné použitie asymetrickej šifry ilustruje obrázok 2.



Obrázok 2: Typické použitie asymetrickej šifry

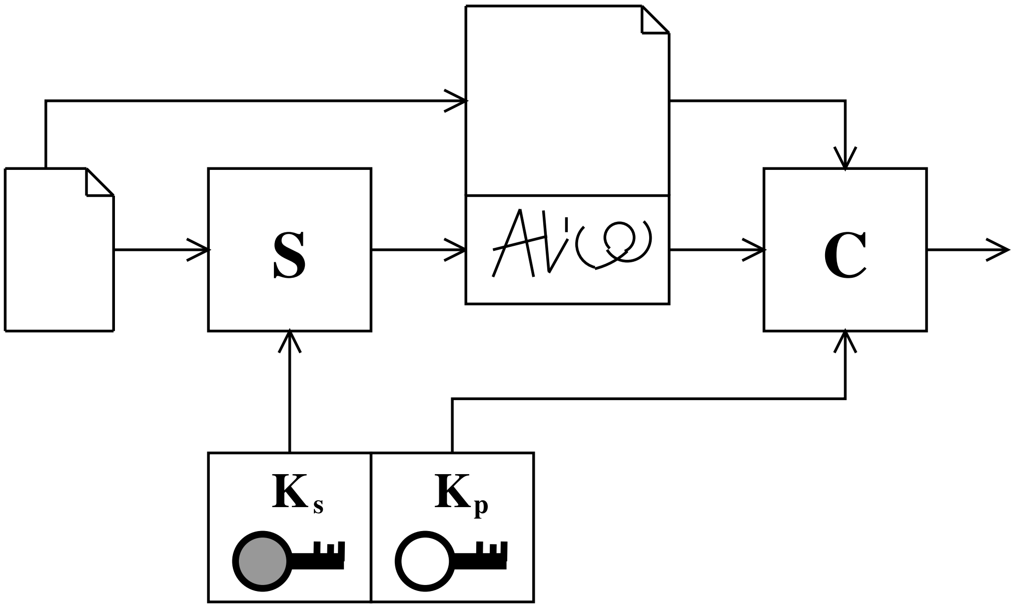
Asymetrické šifry odbúravajú nutnosť udržovať spoločné tajomstvo, keďže verejný kľúč nemusí byť držaný v tajnosti a nie je potrebný bezpečný kanál na výmenu kľúčov. Medzi najznámejšie súčasné šifry s verejným kľúčom patria RSA, Diffie-Hellman, ElGamal a iné.

2.1.2 Elektronický podpis

Elektronický podpis je obdoba klasického podpisu. Zaručuje autenticitu podpísanej správy. Po matematickej stránke je elektronický podpis funkciou obsahu správy (M) a súkromného kľúča (Obrázok 3). Výstupom tejto funkcie je hodnota (S), ktorá sa väčšinou pripojí na koniec správy a je spolu so správou odoslaná. Príjemca správy môže na základe vlastníctva verejného kľúča odosielateľa overiť platnosť podpisu. Keďže elektronický podpis je funkciou správy, modifikácia správy po jej podpísaní spôsobí neplatnosť podpisu.

---

[1] Pri asymetrických šifrách sa šifrovací kľúč nazýva *verehný kľúč* a oznčuje sa ako K_p. Dešifrovací kľúč sa nazýva súkromný kľúč a označuje sa K_s.



Obrázok 3: Elektronický podpis

2.2 Certifikáty

Sila každej šifry je v hlavnej miere založená na správnom zaobchádzaní s kľúčovým materiálom. Pri symetrických šifrách musí byť kľúč držaný v tajnosti, pretože kto vlastní kľúč, môže dešifrovať správu. Pri asymetrických šifrách je privátny (tajný) kľúč najcitlivejšou časťou systému. Ale aj manažment verejných kľúčov asymetrických kryptosystémov predstavuje netriviálny problém. V praxi môže nastať situácia, keď vzniknú pochybnosti o vlastníctve istého verejného kľúča a identite majiteľa príslušného privátneho kľúča. Vhodnou manipuláciou kľúčového materiálu je prakticky možné narušiť bezpečnosť aplikácie asymetrického kryptosystému. V takomto prípade nastáva nutnosť kryptograficky viazať identitu vlastníka kľúčového páru a jeho príslušného verejného kľúča. Ideálnym nástrojom na realizáciu tejto väzby je elektronický podpis dôveryhodnej tretej strany. Subjekt, ktorý vykonáva túto väzbu sa nazýva *certifikačná autorita.*

Dátová štruktúra realizujúca väzbu medzi verejným kľúčom a subjektom, ktorému tento kľúč patrí sa nazýva digitálny certifikát[2]. Formát digitálneho certifikátu je definovaný štandardom ITU-*T* X.509. Podľa tohto štandardu certifikát môže obsahovať:

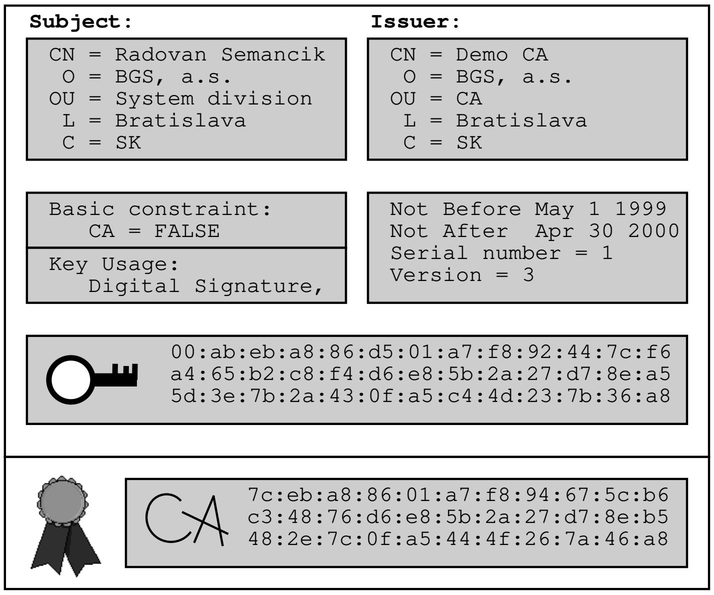
- **Identifikáciu subjektu:** Štrukturované meno subjektu, ktorému certifikát patrí. Meno je uložené vo formáte definovanom štandardom ITU-*T* X.500 "distinguished name".
- **Identifikácia vydávajúcej autority:** Štrukturované meno autority, ktorá certifikát vydala. Meno je uložené v rovnakom formáte ako meno subjektu.
- **Určenie platnosti certifikátu:** Dátum začiatku a konca obdobia platnosti certifikátu.

---

[2] V prípade, že tento subjekt je skutočná osoba, certifikát sa zvykne označovať ako "Digital ID".

- **Verejný kľúč subjektu:** Binárne dáta tvoriace verejný kľúč subjektu. Subjekt musí vlastniť prislúchajúci privátny kľúč. Spolu s binárnymi údajmi kľúča sú uvedené aj parametre kryptosystému, pre ktorý je kľúč určený (použitý šifrovací, sumačný a podpisovací algoritmus, dĺžky príslušných kľúčov, atď.)
- **Sériové číslo certifikátu:** Sériové číslo unikátne identifikuje certifikát v rámci certifikačnej autority.
- **Verzia certifikátu:** Verziové číslo formátu certifikátu upresňuje samotnú štruktúru certifikátu. Podľa verzie certifikátu sa napríklad rozlišuje, či certifikát môže obsahovať rozširujúce polia.
- **Upresňujúce rozšírenia:** Presnejšie špecifikujú obmedzenia a predpokladané použitie certifikátu. Rozšírenia môžu špecifikovať obmedzenia certifikátu, napríklad či sa môže použiť ako certifikát CA alebo nie. Iné rozšírenia ponúkajú používateľovi certifikátu dodatočné informácie o certifikáte alebo jeho vlastníkovi.
- **Podpis certifikačnej autority:** Celá dátová štruktúra certifikátu je podpísaná privátnym kľúčom certifikačnej autority. Tento podpis zaručuje integritu certifikátu a vylučuje akúkoľvek možnosť manipulácie s ním.

Schematický príklad štruktúry certifikátu je ilustrovaný na obrázku 4, výpis skutočného certifikátu je uvedený v tabuľke 1.



Obrázok 4: Štruktúra certifikátu

Certifikáty je možné vydať takmer pre čokoľvek. Certifikát môže potvrdzovať totožnosť fyzickej osoby (napr. "John Doe"), funkcie v spoločnosti ("AnyThing Help Desk"), Organizácie alebo časti organizácie ("IT division, AnyThing inc."), môže potvrdzovať identitu servera na Internete ("www.anything.com"), alebo dokonca autenticitu a neporušenosť softvérového produktu.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 6 (0x6)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=SK, L=Bratislava, O=Business Global Systems, CN=Test CA/Email=semancik@charon.bgs.sk
        Validity
            Not Before: Nov 23 09:04:13 1999 GMT
            Not After : Nov 22 09:04:13 2000 GMT
        Subject: C=SK, ST=, L=Valhala, O=Valhala, OU=, CN=Odin/Email=odin@valhala.gov
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (768 bit)
                Modulus (768 bit):
                    00:ab:eb:a8:86:d5:01:a7:a1:f8:92:44:7c:f6:93:
                    f1:e5:b7:a8:71:c1:d8:b5:92:0a:bd:4e:bd:54:c2:
                    2d:a7:f8:84:88:43:0e:f4:96:0e:d0:d9:55:48:cb:
                    90:4b:ad:68:91:4e:34:9c:b5:55:c7:c2:d8:b5:96:
                    b0:6b:e6:5b:ce:48:06:9e:05:67:bb:98:fd:63:86:
                    60:de:c4:2c:a9:11:3b:30:6b:d8:4f:6d:b4:ed:7a:
                    77:96:90:f1:0b:b1:79
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Cert Type:
                SSL Client, S/MIME
            X509v3 Key Usage:
                Digital Signature, Non Repudiation, Key Encipherment
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                C8:82:D0:CE:6D:26:56:DD:E6:77:69:8A:BA:2A:E6:27:F8:C7:13:22
            X509v3 Authority Key Identifier:
                keyid:E3:16:E2:DB:4D:B1:9A:9C:3F:64:C0:14:51:4C:63:E9:76:D1:DB:8D
                DirName:/C=SK/L=Bratislava/O=Business Global Systems/CN=Test CA/Email=semancik@charon.bgs.sk
                serial:00
            X509v3 Subject Alternative Name:
                email:odin@valhala.gov
            X509v3 Issuer Alternative Name:
                <EMPTY>
            X509v3 CRL Distribution Points:
                URI:http://identity.bgs.sk/
    Signature Algorithm: md5WithRSAEncryption
        a2:c4:15:fa:67:11:dd:5f:5d:50:49:10:13:48:ee:6f:ca:f1:
        92:98:62:2d:a4:b9:34:26:1e:ef:20:25:b9:2d:81:9f:9e:fc:
        40:52:14:ef:3a:84:fc:55:62:92:55:fc:86:84:75:ca:ed:df:
        28:b3:b6:48:36:63:e5:35:a0:e6:82:1e:f9:2a:46:42:d2:af:
        4c:46:29:83:66:ab:2c:f5:58:51:db:ef:b2:b2:af:50:3c:a5:
        6e:3c:46:c3:7c:4c:5c:4f:19:05:fb:2d:00:72:4d:5f:9e:66:
        74:7b:3f:ac:b0:77:56:52:a9:49:8d:82:43:66:7c:33:17:a5:
        08:6c
```

Tabuľka 1: Príklad výpisu obsahu certifikátu

2.3 Certifikačná autorita

Certifikačná autorita je dôveryhodná tretia strana, ktorá pomocou vydávania elektronických certifikátov potvrdzuje identitu subjektov v istej územnej alebo organizačnej oblasti. Hlavnou úlohou certifikačnej autority je vydávať, spravovať a rušiť certifikáty svojich klientov.

2.3.1 Vydávanie certifikátov

Spôsob, akým certifikačná autorita vydáva certifikáty, sa nazýva proces certifikácie (certificate enrollment). Tento proces pozostáva z niekoľkých krokov (Tabuľka 2) .

Krok 1	Generovanie páru kľúčov žiadateľom
Krok 2	Odoslanie žiadosti o certifikát (certification request) žiadateľom
Krok 3	Overenie totožnosti žiadateľa certifikačnou autoritou a podpísanie certifikátu
Krok 4	Odoslanie certifikátu žiadateľovi alebo prebratie certifikátu žiadateľom

Tabuľka 2: Proces certifikácie

Prvým krokom je generovanie kľúčového páru na strane žiadateľa o certifikát. Vygenerovaný súkromný kľúč sa bezpečne uloží na strane žiadateľa a neprenáša sa na stranu certifikačnej autority, takže certifikačná autorita nemusí, a ani by nikdy nemala, poznať súkromný kľúč žiadateľa o certifikát. Druhým krokom je generovanie žiadosti o certifikát. Žiadosť o certifikát pozostáva z verejného kľúča žiadateľa o certifikát a informácií o jeho identite, celé podpísané súkromným kľúčom žiadateľa. Táto žiadosť o certifikát je odoslaná certifikačnej autorite. V treťom kroku si certifikačná autorita overí identitu žiadateľa o certifikát. Toto overenie môže siahať od jednoduchých metód ako napríklad poslanie správy s heslom na e-mailovú adresu žiadateľa až po nutnosť osobného kontaktu žiadateľa a zástupcu CA. Po úspešnej autentifikácií žiadateľa certifikačná autorita podpíše verejný kľúč žiadateľa spolu s informáciami o jeho identite uloženými vo formáte X.509, čím vznikne výsledný certifikát. Posledným krokom je odoslanie certifikátu žiadateľovi, alebo inak prevedené prevzatie certifikátu žiadateľom.

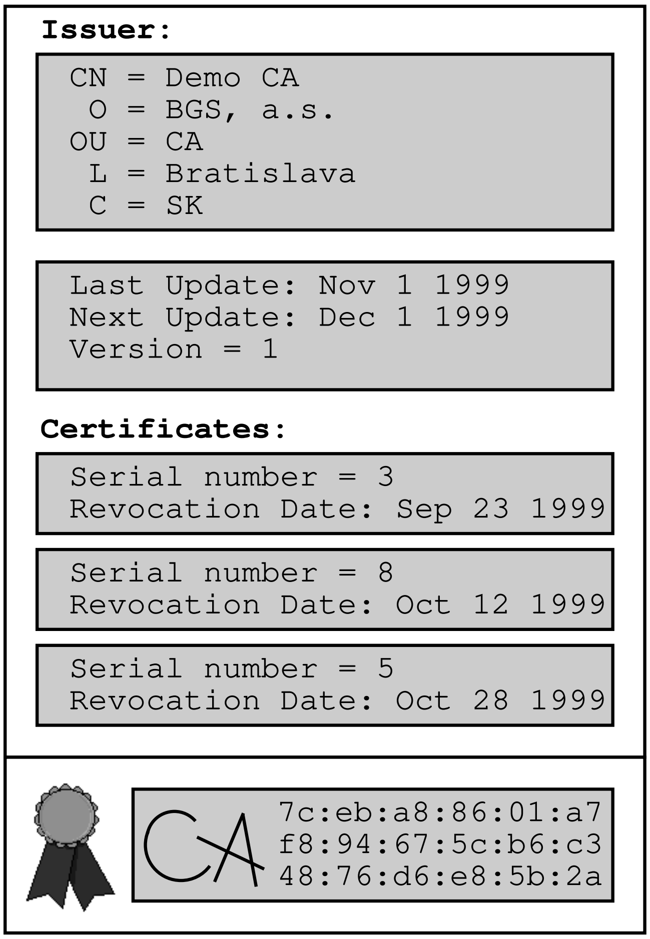
2.3.2 Používanie certifikátu

Keď je digitálny certifikát podpísaný certifikačnou autoritou, nevyžaduje si jeho použitie žiadnu akciu zo strany certifikačnej autority. Certifikovaný subjekt predkladá digitálny certifikát svojim komunikačným partnerom na overenie a jediná aktivita certifikačnej autority spojená s overovaním certifikátov, je udržovať aktuálny zoznam zneplatnených certifikátov alebo iným spôsobom publikovať informácie o platnosti certifikátu. Doba platnosti je v certifikáte explicitne uvedená a po jej vypršaní sa certifikát stáva neplatným bez ďalšieho zásahu certifikačnej autority.

2.3.3 Odobratie certifikátu

V praxi sa môže stať, že vydaný certifikát je nutné zneplatniť ešte pred vypršaním doby jeho platnosti. Keďže certifikát je informácia, ktorá sa dá voľne kopírovať, nedá sa držiteľovi priamo odobrať. Za účelom zneplatňovania certifikátov, ktorých doba platnosti ešte neskončila, si certifikačná autorita udržuje zoznamy zneplatnených certifikátov, (CRL, Certificate Revocation List). Ak má byť certifikát zneplatnený, je umiestnený na tento zoznam. Ak niekto robí dôležité operácie, pri ktorých sa spolieha na certifikát, mal by si skontrolovať jeho platnosť prehľadaním zoznamu zneplatnených certifikátov.

Štruktúra zoznamu zneplatnených certifikátov je pevne určená a tak je možné overovanie platnosti certifikítov automatizovať. Celý zoznam je zabezpečený podpisom certifikačnej autority, takže ho nie je možné sfalšovať. Zoznam zneplatnených certifikátov obsahuje dátum svojho vydania a dátum vydania ďalšieho zoznamu, takže je možné udržovať jeho lokálnu kópiu a pravideľne ju aktualizovať. Príklad štruktúry zoznamu zneplatnených certifikátov je na obrázku 5.



Issuer:

CN = Demo CA
O = BGS, a.s.
OU = CA
L = Bratislava
C = SK

Last Update: Nov 1 1999
Next Update: Dec 1 1999
Version = 1

Certificates:

Serial number = 3
Revocation Date: Sep 23 1999

Serial number = 8
Revocation Date: Oct 12 1999

Serial number = 5
Revocation Date: Oct 28 1999

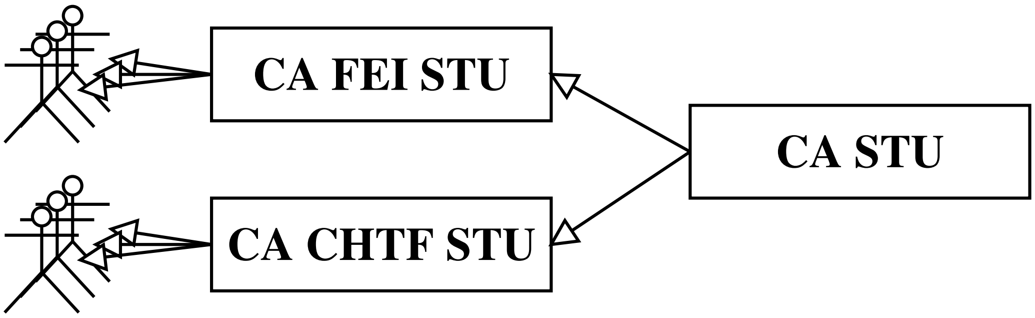
7c:eb:a8:86:01:a7
f8:94:67:5c:b6:c3
48:76:d6:e8:5b:2a

Obrázok 5: Zoznam zneplatnených certifikátov

2.4 PKI - Infraštruktúra

Certifikačná autorita svojim podpisom podvrdzuje certifikáty. Verejný kľúč certifikačnej autority však tiež môže byť potvrdený certifikátom nadradenej certifikačnej autority[3]. Táto nadradená certifikačná autorita môže byť zase certifikovaná certifikačnou autoritou ešte vyšej úrovne. Takto vzniká stromová štruktúra certifikačných autorít, založená na reťazení certifikátov.

Infraštruktúra vybudovaná pomocou kryptografického viazania certifikátov sa nazýva PKI - Infraštruktúra verejných kľúčov (*Publick Key Infrastructure*). Najčastejšie používaný typ PKI je strom previazaných certifikačných autorít, ktorá najlepšie zodpovedá organizácií autorít skutočného sveta. Certifikačné autority vznikajú v miestach, ktoré prirodzene môžu overovať svojich klientov. Firmy certifikujú svojich zamestnancov, školy svojich študentov, obce svojich obyvateľov. Tieto lokálne certifikačné autority sú certifikované nadradenými certifikačnými autoritami s väčšou územnou pôsobnosťou. Napríklad certifikačná autorita univerzity vydá certifikát pre podradenú certifikačnú autoritu fakulty. Certifikačná autorita regiónu bude certifikovať certifikačné autority obcí. Príklad takto vytvorenej hierarchie je ilustrovaný na obrázku 6. Certifikačné autority FEI STU a CHTF STU vydávjú certifikáty svojim študentom a zamestnancom. Certifikačná autorita STU má pôsobnosť pre celú univerzitu a vydáva certifikáty pre certifikačné autority jednotlivých fakúlt. Takýmto spôsobom môže vzniknúť tranzitívny vzťah dôvery medzi jednotlivými fakultami.



Obrázok 6: Príklad stromu certifikačných autorít

V praxi existuje väčšinou niekoľko nezávislých hierarchií a jedna fyzická osoba môže mať niekoľko nezávislých identít certifikovaných v nezávislých stromoch. Napríklad, obchodný riaditeľ firmy bude mať certifikát vydaný na "JOHN DOE, SALES DIRECTOR, ANYTHING, INC", ktorý bude používať na služobnú komunikáciu a druhý certifikát vydaný na "JOHN DOE, BAKER STREET 222, LONDON, UK" používaný na súkromnú komunikáciu. Taktiež je možné takéto nezávislé stromy spojiť na úrovni ich koreňov pomocou mechanizmu krížovej certifikácie (*cross-certification*) certifikačných autorít.

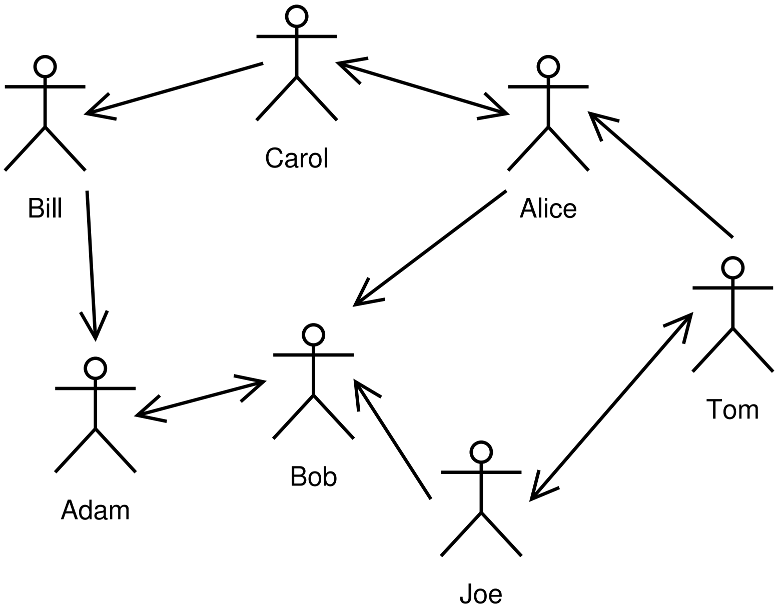
---

[3] Alebo certifikačná autorita môže sama podpisovať svoj verejný kľúč, čo je typický prípad certifikačných autorít najvyšej úrovne. Takýto certifikát sa označuje ako *"self signed"*.

Hierarchická kryptografická infraštruktúra je typická pre väčšinu moderných systémov, poskytujúcich štrukturované kryptografické skužby. Medzi inými sú to systémy S/MIME, PEM, IPsec, SKIP, SSL a TLS.

2.4.1 Pavučina dôvery

V prípade, že certifikát nerozlišuje, či je jeho vlastník oprávnený vydávať ďalšie certifikáty, vzniká sieť dôvery, ktorá sa neriadi žiadnymi pevne určenými pravidlami. Takáto sieť sa vo všeobecnosti označuje ako *"Pavučiva dôvery"* (Web of trust) [1]. Príklad takejto štruktúry je na obrázku 7 .Takáto infraštruktútra je vhodná pre systémy, v ktorých nie je určená žiadna pevná štruktúra. Infraštruktúra sa ľahko buduje, pretože nie je nutné dopredu určovať úlohy jednotlivých subjektov. Je úlohou každého zúčastneného posúdiť, ktorým subjektom dôveruje a teda im vydá príslušný certifikát. Na druhej strane je nesmierne zložité sledovať niť dôvery vo všeobecnom orientovanom grafe, aký táto štruktúra poskytuje. V mnohých prípadoch dokonca neexistuje prijateľná cesta medzi dvoma subjektami v štruktúre. Navyše, aj keď existuje tranzitívny vzťah medzi subjektami, je náročné vyhodnotiť stupeň dôveryhodnosti tohto vzťahu.



Obrázok 7: Pavučina dôvery

Štruktúra zodpovedajúca takejto pavučine je typická napríklad pre systém zabezpečenia elektronickej pošty OpenPGP.

2.5 Zabezpečenie transportnej vrstvy - SSL

Protokol SSL (*Secure Socket Layer*) je protokol relačnej vrstvy, poskytujúci bezpečnostné služby medzi transportnou (TCP) a aplikačnou vrstvou protokolovej rodiny TCP/IP. Protokol SSL zabezpečuje komunikáciu medzi klientom a serverom ich vzájomnou kryptografickou autentifikáciou, použitím digitálnych certifikátov a šifrovaním prenášaných údajov [3, 4]. Protokol umožňuje pri nadväzovaní spojenia dohodnúť kryptografické algoritmy a ich parametre v závislosti od exportných a legálnych obmedzení. Protokol SSL slúži ako základ pre protokol TLS (*Transport Layer Security*), ktorý prebieha štandardizáciou v IETF.

V priebehu nadväzovania spojenia so serverom pomocou protokolu SSL si server a klient vytvoria unikátny identifikátor relácie (*"session ID"*), ktorý je možné pri opätovnom nadväzovaní spojenia použiť a tak vynechať úvodnú komunikáciu. Server prezentuje strane klienta svoj certifikát a prípadne žiada certifikát aj od klienta. Po výmene certifikátov sa obe strany dohodnú na spoločných kryptografických algoritmoch,ktoré budú použité na ochranu prenášaných údajov. Dostupné kryptografické algoritmy pre SSL verzie 3 sumarizuje tabuľka 3 . Aplikačné údaje a aj samotné protokolové jednotky SSL, sú chránené pomocou dohodnutých kryptografických algoritmov.

Šifrovací algoritmus			Kryptografická suma	
Názov	Typ	Dĺžka kľúča	Názov	dĺžka
Žiadny			Žiadna	
RC4	prúdový	40 bitov	MD5	128 bitov
RC4	prúdový	128 bitov	SHA-1	160 bitov
RC2	blokový	40 bitov		
DES	blokový	40 bitov		
DES	blokový	54 bitov		
3DES	blokový	168 bitov		
IDEA	blokový	128 bitov		
Fortezza	blokový	96 bitov		

Tabuľka 3: Kryptografické algoritmy na ochranu prenášaných údajov v SSL

3 Analýza problému

Bezpečnosť komunikácie v prostredí verejných dátových sietí a najmä siete Internet je všeobecne uznávaný problém [1]. Drvivá väčšina informácií je prenášaná v otvorenom tvare a najpoužívanejšia autentifikačná technika sú statické heslá. Najmä populárnym službám, ako sú elektronická pošta a WWW chýbajú široko používané prvky zabezpečenia. Elektronická pošta sa stala jednou z najpoužívanejších služieb Internetu, a predsa je stále neuveriteľne triviálne sfalšovať zdrojovú adresu správy elektronickej pošty. V poslednom období vzniká v prostredí Internetu čoraz viac aplikácií založených na technológií WWW, ich ochrana je však minimálna. Heslá používané na autorizáciu aplikácií WWW sú takmer nechránené a vhodne situovaný útočník môže ľahko realizovať útok opakovaním[1]. Tieto podmienky spôsobujú, že útočník vhodne umiestnený na komunikačných kanáloch môže získať takmer neobmedzený prístup a dokonale impersonifikovať svoju obeť. Riziko je hlavne sústredené u poskytovateľov Internetových služieb, ktorí v našich podmienkach často zamestnávajú málo kvalifikovaný personál, ich celková úroveň bezpečnosti je nízka a často sa stávajú cieľom Internetových útočníkov.

V čase rozmachu elektronického obchodu sa Internet stáva médiom aj pre finančné transakcie a medzifiremnú komunikáciu. Údaje prenášané v takýchto systémoch majú často vysoký stupeň senzitivity, preto sa v poslednom čase čoraz častejšie hovorí o ochrane údajov v spojitosti s elektronickým obchodom na Internete. Na ochranu údajov v globálnej verejnej sieti, ako je Internet, je nutné používať silné kryptografické metódy. Základom kryptografie v otvorených sieťach sú prostriedky asymetrickej kryptografie a infraštruktúra, ktorá je pomocou nich vytvorená. Certifikačné autority tvoria základný stavebný prvok tejto infraštruktúry.

3.1 Funkcia a štruktúra

Certifikačná autorita by mala poskytovať svoje služby verejne alebo na vnútornej úrovni organizácií. Z pohľadu globálnych informačných sietí môžeme teda definovať pôsobnosť certifikačnej autority pre oblasť Internetu alebo oblasti Intranetov organizácií. Keďže certifikačná autorita musí operovať v silne heterogénnom prostredí, jej funkčnosť v oblasti komunikácie s klientami by mala byť založená na stabilných, dobre preskúmaných princípoch. Ako ideálne aplikácie takéhoto druhu sa osvedčili aplikácie založené na princípoch služby WWW. Aplikačný protokol HTTP [5], ako aj formát prenášanej hypertextovej informácie HTML [6], sú dobre etablované a široko používané štandardy. Prehliadače WWW existujú takmer na akejkoľvek klientskej platforme a sú vyvíjané rapídnym tempom.

Prostredie WWW poskytuje všetky nástroje na realizáciu základnej aplikácie. Niektoré WWW prehliadače dokonca priamo obsahujú podporu pre prácu s certifikátmi a základné kryptografické funkcie, ktoré môže aplikácia využiť. Na strane servera existuje množstvo komponentov a podporných produktov na uľahčenie realizácie aplikácie. Aplikácie je tiež odľahčená od priamej prdpory zobrazovania grafického používateľského rozhrania a nie je viazaná na jednu platoformu.

3.2 Použitie certifikačnej autority

V nasledujúcich odstavcoch budú popísané možnosti použitia certifikačnej autority a vydaných certifikátov. Reálne použitie certifikátov závisí od kontrétnej aplikačnej oblasti a spôsobu nasadenia certifikačnej autority.

3.2.1 S/MIME- Bezpečná pošta

Štandard pre zabezpečenie elektronickej pošty S/MIME bol navrhnutý organizáciou RSA Labs. a prijatý množstvom výrobcov. Dnes je S/MIME považovaný za priemyselný štandard a je podporovaný veľkou väčšinou komerčných softvérových produktov. Systém S/MIME počíta s hierarchickou infraštruktúrou certifikačných autorít a na svoju správnu činnosť potrebuje osobný certifikát vydaný certifikačnou autoritou pre koncového používateľa.

3.2.2 SSL - Bezpečné servery

Pomocou SSL je štandardne riešené zabezpečenie bežných služieb Internetu, ako napríklad HTTP, POP3, IMAP, atď. [1]. Najčastejšie sa SSL používa na zabezpečenie WWW serverov. Rozšírenie štandardného protokolu HTTP o bezpečnostné služby SSL sa nazýva HTTPS. Tento protokol umožňuje dramaticky zvýšiť bezpečnosť servera s minimálnou námahou. Pre WWW aplikácie je tento protokol absolútne transparentný, nevyžaduje si žiadnu zmenu aplikácie. Ochrana údajov (šifrovanie, MAC) prebieha medzi prehliadačom koncového používateľa a procesmi WWW servera. WWW server predkladá používateľovi svoj certifikát, vylučujúc tak možnosť zámeny IP adresy servera [1]. Samotný koncový používateľ nepotrebuje osobný certifikát, ak ho však má, môže ho použiť na autentifikáciu. Takáto silná autentifikácia je oveľa bezpečnejšia ako autentifikácia heslami.

3.2.3 S/WAN- Bezpečné siete

Iniciatíva S/WAN (Secure WAN) vznikla na podnet RSA Laboratories a smeruje k vytvoreniu bezpečného prostredia pre rozľahlé a globálne počítačové siete. Hlavnou aktivitou iniciatívy S/WAN je podpora bezpečnostných rozšírení protokolovej rodiny TCP/IP – IPsec. IPsec je štandardizované rozšírenie TCP/IP prinášajúce bezpečnostné služby na sieťovú úroveň. Zabezpečenie je realizované už na úrovni IP paketov tak, že transportné a aplikačné dáta zostávajú nezmenené. Ochrana pomocou IPsec je teda pre aplikácie absolútne transparentná. Sieťové zariadenia, a v prípade mobilných alebo vzdialených používateľov aj sieťové operačné systémy, však musia dokázať medzi sebou komunikovať pomocou IPsec. Automatická výmena kľúčov v IPsec je založená na certifikátoch. V globálnych sieťach je totiž takmer nemožné vytvoriť funkčný a efektívny bezpečnostný systém bez použitia kryptografickej infraštruktúry (PKI).

3.3 Podobné riešenia problému

V nasledujúcich kapitolách budú zhrnuté riešenia problému svojou povahou blízke riešeniu, ktoré poskytuje tento dokument. Pri každom riešení sú uvedené jeho základné vlastnosti, použité technológie (ak sú známe) a dostupnosť riešenia.

3.3.1 OpenCA

Projekt OpenCA je dobrovoľná iniciatíva vyvinúť plne funkčnú softvérovú certifikačnú autoritu. Projekt je založený na použití štandardných prístupov a protokolov a využíva plnú kryptografiu, bez aplikácie exportných obmedzení niektorých štátov. V čase písania tejto práce bola aktuálna vývojová verzia OpenCA 0.2.0, ktorá umožňovala prijímať žiadosti o certifikát, potvrdzovať ich a podpisovať a publikovať CRL. Projekt je založený na WWW technológií, využíva možnosti kryptografickej knižnice OpenSSL a dominantná časť je realizovaná v jazyku Perl. Produkt je k dispozícií bez poplatku pod podmienkami voľnej "umeleckej" licencie [15].

3.3.2 Verisign OnSite

Produkt OnSite firmy Verisign ponúka kompletné riešenie poskytnutím prenajatých certifikačných služieb na hardvérovo zabezpečenej platforme. Kryptografické kľúče sú držané na fyzicky neprístupných miestach, servery aj telekomunikačné spojenia sú redundantné a zálohované. Zaručené služby 7x24h. Firma Verisign má nesporne dlhú tradíciu a hlboké skúsenosti v oblasti PKI a certifikačných služieb. Certifikačné služby zahŕňajú vydávanie, spravovanie, odoberanie a obnovovanie certifikátov. Služby sú poskytované outsourcingovým systémom na hardvérových zariadeniach firmy Verisign. Vydané certifikáty sú rozoznávané bežnými prehliadačmi WWW, softvérovými balíkmi pre elektronickú poštu ako aj niektorými zariadeniami IPsec [7].

Služby sú poskytované na komerčnej báze, používateľské certifikáty sú v cenových reláciách 200Sk – 500Sk na jeden používateľský certifikát (min. 500 certifikátov) a 13 000Sk – 15 000Sk za jeden certifikát servera (min. 10 certifikátov) [8].

3.3.3 Entrust PKI

Riešenie PKI od firmy Entrust je softvérový balík, realizujúci služby certifikačnej autority. Produkt v základnej verzií používa softvérové kryptografické prostriedky s možnosťou doobjednania hardvérového riešenia. Kľúče sú uchovávané v on-line databáze. Na klientský prístup k systému je vyžadovaný špeciálny softvérový balík. Systém podporuje rekonštrukciu kľúčov. Softvérové riešenia sú poskytované na komerčnej báze.

3.3.4 Baltimore UniCert

Systém certifikačnej autority UniCert od firmy Baltimore je riešenie smerujúce hlavne do oblasti elektronického obchodu a podnikových bezpečnostných systémov. Systém je modulárne postavený, podporujúci moduly registračných autorít spojených na centrálny modul certifikačnej autority. Podporuje vydávanie certifikátov pre SSL, S/MIME aj VPN, podporuje LDAP a X.500. Systém je vhodný ako Internetová, Intranetová a Extranetová certifikačná autorita. Je poskytovaný ako softvérové riešenie pre platformy Windows NT a Solaris, v cenovej relácií cca 840 Sk za certifikát (granularita 1000 certifikátov), cena produktu je cca 1 600 000 Sk.

3.4 Navrhované riešenie problému

Riešenia popísané v predchádzajúcej kapitole nenaplňujú požiadavky na certifkačné služby pre prostredie malej oragnizácie. Všetky komerčne poskytované riešenia majú vysokú počiatočnú cenu (cca 500 000 - 2 milióny Sk) a zväčša aj vysoké prevádzkové náklady. Nasadenie takýchto riešení v našich podmienkach je reálne len vo veľkých organizáciách, ktoré si okrem nákladov na samotný produkt môžu dovoliť zaplatiť aj potrebný personál a jeho odbornú prípravu. Na druhej strane nekomerčné riešenia sú len v začiatkoch a neposkytujú reálne možnosti nasadenia v produkčnom prostredí bez nutnosti zúčastňovať sa na ich vývoji alebo testovaní.

Certifikačná autorita vhodná do prostredia malých organizácií by sa mala vyznačovať jednoduchým ovládaním a hlavne prijateľnou cenou. Systém by nemal byť náročný na hardvérové prostredie a umožňovať rozrastanie systému a škálovateľnosť. Réžia vydávania certifikátov by nemala presiahnuť cenu komerčných poskytovateľov certifikačných služieb. Prístup klientov k certifikačnej autorite by mal byť možný bez ohľadu na platformu klientských pracovných staníc.

4 Špecifikácia požiadaviek

Pre plnohodnotné využitie vlastností certifikačnej autority je nutné na jej prevádzku uplatniť silné požiadavky, hlavne v oblasti globálnej dostupnosti služieb certifikačnej autority. Požiadavky na bezpečnosť systému certifikačnej autority sú tiež nesmierne dôležité, je však neefektívne budovať certifikačnú autoritu s neporovnateľne vyšším stupňom bezpečnosti ako je stupeň bezpečnosti samotnej cieľovej aplikácie. Hlavné oblasti požiadaviek na systém certifikačnej autority sú zhrnuté v nasledujúcom odstavci, plná špecifikácia požiadaviek je uvedená v prílohe A.

4.1 Hlavné oblasti požiadaviek

- **Dostupnosť služieb:** Služby certifikačnej autority musia byť priamo dostupné z väčšiny bežných klienstských platforiem. Musí byť použitý štandardizovaný spôsob prístupu k službám, umožňujúci poskytovať certifikačné služby bez silného územného obmedzenia.
- **Používateľské rozhranie:** Služby bežne dostupné verejnosti a rutinnej obsluhe certifikačnej autority musia byť prístupné pomocou používateľsky príjemného rozhrania. Rozhranie musí používateľovi poskytovať len služby, na ktoré má právo a dať mu možnosť autentifikovať sa.
- **Bezpečnosť:** Služby poskytované certifikačnou autoritou musia vykazovať primeraný stupeň bezpečnosti a odolnosti proti zneužitiu. Každý používateľ oprávnený vykonávať zmeny na certifikačnej autorite musí byť jednoznačne autentifikovaný. Certifikačná autorita musí mať k dispozícií nielen autentifikáciu statickými heslami, ale aj silný autentifikačný mechanizmus. Autorizácia používateľských akcií musí byť vykonávaná na základe princípu minimálnych privilégií.
- **Vydávanie certifikátov:** Klient certifikačnej autority musí mať možnosť podať žiadosť o certifikát elektronickým spôsobom, bez nutnosti osobného kontaktu s pracovníkmi certifikačnej autority. Pracovník certifikačnej autority musí mať možnosť overiť identitu žiadateľa pred podpísaním certifikátu. Žiadateľ musí mať možnosť elektronicky prebrať svoj certifikát po jeho podpísaní certifikačnou autoritou.
- **Zneplatňovanie certifikátov:** Operátor certifikačnej autority musí mať možnosť zneplatniť certifikát v platnom stave jeho automatickým umiestnením na zoznam zneplatnených certifikátov.
- **Poskytovanie informácií:** Klienti certifikačnej autority musia mať možnosť získať informácie o stave ich žiadosti a informácie o certifikátoch vydaných certifikačnou autoritou. Tieto možnosti však musia byť obmedzené tak, aby minimalizovali možnosť získané informácie zneužiť. Certifikačná autorita musí poskytovať aj bežné verejné informácie v správnych formátoch. Medzi tieto informácie patria:

  - Certifikát certifikačnej autority a prípadný ďalší reťazec certifikátov až ku koreňu certifikačnej hierarchie.
  - Zoznam zneplatnených certifikátov, vydávaný pravidelne.
  - Samotné vydané certifikáty.

- **Finančná dostupnosť:** Systém musí pozostávať z takých komponentov, aby jeho prevádzka nebola finančnej náročná.

5 Návrh

V nasledujúcich kapitolách bude popísaný systém, ktorý zopovedá požiadavkám načrtnutým v predchádzajúcej kapitole a špecifikovaným v prílohe A. Na úvod budú definované konvencie použité v ďalšom texte, neskôr bude uvedený návrh architektúry celého systému a jeho jednotlivých komponentov.

5.1 Postup riešenia problému

Niektoré implementačné rozhodnutia boli uskutočnené už vo fáze návrhu. Ide najmä o konkrétnu voľbu kľúčových komponentov systému. Toto rozhodnutie vo fáze návrhu prispelo k zefektívneniu procesu návrhu a implementácie. Už samotné objektové štruktúry vo fáze návrhu boli prispôsobené konvenciám a možnostiam implelemntačnej platformy (kapitola 5.3).

Voľba kryptografickej knižnice OpenSSL bola takmer jediným možným riešením. OpenSSL je jediná voľne dostupná krytografická knižnica, ktorá obsahuje potrebné funkcie na spracovanie certifikátov. Keďže v čase písania tohto dokumentu bola k dispozícií len vývojová verzia 0.9.4 knižnice OpenSSL, mnohé prvky systému sa už v návrhu museli prispôsobiť niektorým neustáleným črtám tejto knižnice (pozri tiež kapitolu 6.7.1).

Ako hlavný implementačný jazyk bol zvolený jazyk Perl. Tento platformovo nezávislý interpretačný jazyk bol zvolený najmä vďaka svojej flexibilite. Ďalšie prednosti jazyka Perl, ktoré prispeli k jeho výberu sú uvedené v tabuľke 4 . Voľba implementačného jazyka istým spôsobom ovplyvnila aj konvencie použité pri návrhu systému, najmä spôsob zápisu dátových typov v objektových diagramoch. Tento spôsob však značne uľahčil implementáciu systému a jeho transparentnosť a zrozumiteľnosť. Uvedený spôsob bol už predtým výhodne použitý pri vývoji aplikačnej kostry Jewels a preto sa javilo výhodné ho použiť aj v tomto projekte.

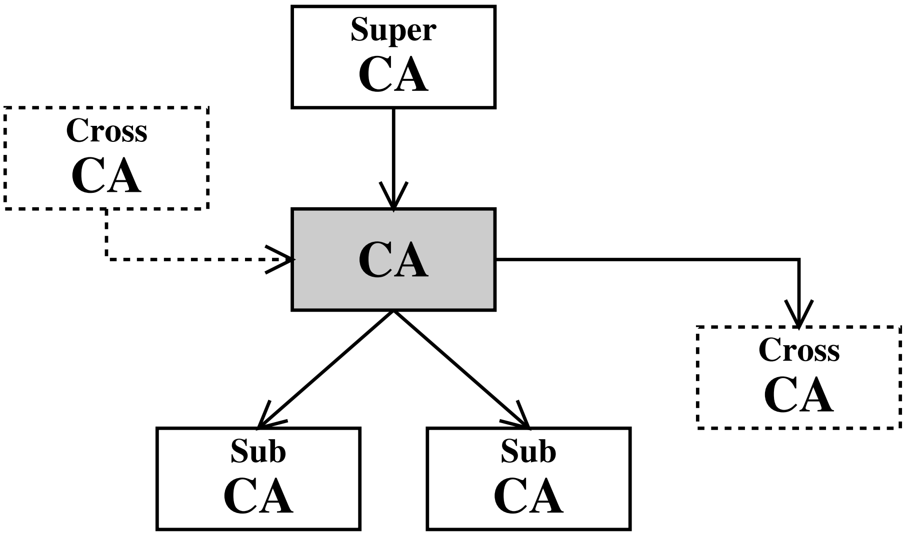
Štrukturované programovanie	Podporuje moduárne delenie aplikácie, nezávislosť modulov
Objektovo orientované programovanie	Podporuje.
Implementačné platformy	UNIX, VMS, Win32
Použitie komponentov	Možstvo externých komponentov "CPAN"
Spôsob vykonávania	Interpretačný, predkompilovanie programu

Tabuľka 4: Vlastnosti programovacieho jazyka Perl

5.2 Návrh kryptografickej infraštruktúry

Na základe faktov uvedených v kapitole 2.4 je nesporne najvýhodnejšia a aj najpoužívanejšia hierachická infraštruktúra. Prísne stromová štruktúra sa však veľmi ťažko vytvára, pretože musí byť určený jeden uzol ako koreň stromu. V praxi je však takmer nemožné ustanoviť jednu certifikačnú autoritu ako globálne centrum a tak vznikajú oddelené stromové štruktúry, každá so svojim vlastným koreňom. Je pravdepodobné, že stromy na ktorých vrchole je certifikačná autorita s malou územnou alebo organizačnou pôsobnosťou sa v priebehu času budú pripájať do stromov s širšou pôsobnosťou, vždy však budú existovať aspoň dva oddelené stromy s konkurenčnými certifikačnými autoritami na vrcholoch.

Spájať oddelené podstromy bez explicitného vyjadrenia hierarchie možno pomocou mechanizmu krížovej certifikácie partnerských certifikačných autorít. Takto možno preklenúť neochotu certifikačných autorít vyšších úrovní spájať oddelené stormy. Typický element takato vzniknutej štruktúry je zobrazený na obrázku 8. Zobrazená certifikačná autorita vlastní certifikát od svojej nadradenej certifikačnej autority (Super CA) a vydáva certifikáty svojim podradeným certifikačným autoritám (Sub CA). Má však možnosť vydať certifikát partnerskej certifikačnej autorite podobným spôsobom ako vydáva certifikáty svojim podradeným certifikačným autoritám. Podobne, ak partnerská certifikačná autorita vydala certifikát pre našu certifikačnú autoritu, môžeme tento certifikát použiť podobne ako certifikát nadradenenej certifikačnej autority.



Obrázok 8: Situácia certifikačnej autority v infraštruktúre.

5.3 Použité konvencie

Použité objektové diagramy vychádzajú z metodiky UML (*Unified Modeling Language*), sú však mierne modifikované v prospech lepšej zrozumiteľnosti návrhu systému. Keďže hlavným implelmentačným jazykom systému bude jazyk Perl, aj konvencie použité v diagramoch vychádzajú z tohto jazyka. Modifikácie sa dotkli najmä definície použitých údajových typov. Vysvetlenie značiek údajových typov je uvedené v tabuľke 5, vysvetlenie významu použitých šípok je v tabuľke 6.

značnka	význam	príklad, poznámka
$	Skalárna hodnota	textový reťazec, číslo, ...
$format	Skalárna hodnota v symbolickom formáte "format"	$xml = text vo formáte XML, $filename = meno súboru
@	Usporiadaný zoznam	("abc",123,"ab43cd")
%	Kľúčovaný zoznam (*Hash*)	(k1 => "val1", k2 => "val2")
\	Generická referencia	Referencia na skalárnu hodnotu, referencia na zoznam, ...
\$	Referencia na skalárnu hodnotu	
\@	Referencia na zoznam	
\%	Referencia na kľúčovaný zoznam.	
\Class	Referencia na objekt triedy "Class"	
\self	Referencia na objekt tej triedy v ktorej sa funkcia nachádza	Používané najmä v konštruktoroch objektov
?	Neurčený typ	

Tabuľka 5: Údajové typy použité v objektových diagramoch

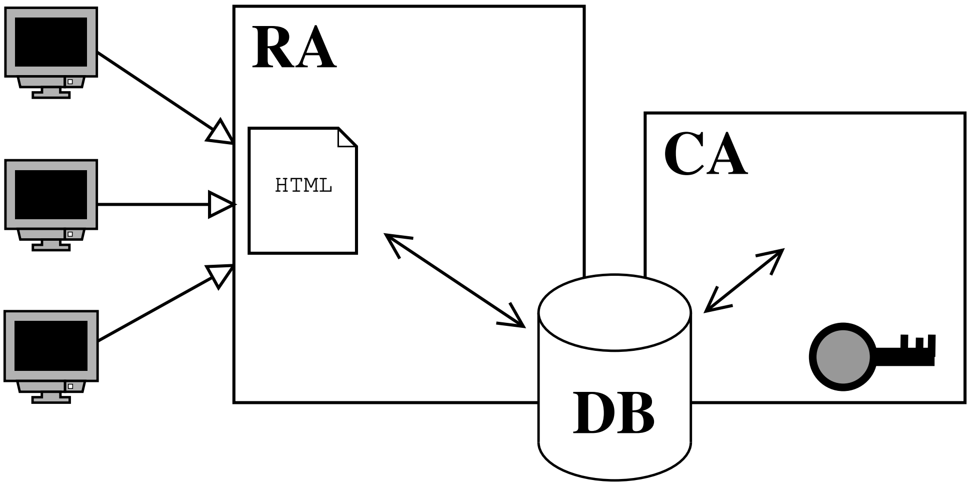
5.4 Kontext systému

Systém sa skladá zo štyroch hlavných častí (Obr. 9):

- **Klientské stanice** predstavujú pracovné stanice používateľov systémov certifikačnej autority. Každá klientská stanica je vybavená WWW prehliadačom a pomocou neho používateľ pristupuje k službám certifikačnej autority. Moderné WWW prehliadače majú v sebe zabudovaný aj kryptografický podsystém a dokážu v spolupráci s aplikáciou certifikačnej autority podať žiadosť o certifikát a prevzať následne podpísaný certifikát. Taktiež umožňujú použiť kryptografické metódy na realizáciu silnej autentifikácie a ochrany prenášaných údajov šifrovaním. Operačný systém klientskej stanice nie je v tomto prípade dôležitý.

typ šípky	názov	význam
A - - - -> B	závislosť	Funkcia A je závislá na funkcií B. Premenná A ukladá objekt typu B.
A - - - -▷ B	realizácia	Funkcia B realizuje činnosť funkcie A.
A ——▷ B	generalizácia	Dedičnosť. Trieda A je podtrieda triedy B.
A -create-> B	vytváranie	Funkcia A vytvára objekt triedy B.

Tabuľka 6: Typy šípok použité v objektových diagramoch



Obrázok 9: Kontextový diagram systému

- **Registračná autorita (RA)** prijíma a spracováva žiadosti a akcie z klientských staníc, eviduje žiadosti o certifikát a certifikáty, umožňuje ich vyhľadávanie, sledovanie a manipuláciu s ich stavom. Taktiež realizuje autentifikáciu používateľov a kontrolu prístupu k službám. Používateľské rozhranie je realizované pomocou dynamicky vytváraných HTML dokumentov, ktoré sú prezentované klientským staniciam.
- **Databáza (DB)** udržuje informácie o certifikátoch, žiadostiach o certifikát, používateľoch a ich rozpracovaných reláciach.
- **Certifikačná autorita (CA)** podpisuje certifikáty a vydáva zoznamy zneplatnených certifikátov. Privátny kľúč certifikačnej autority je prístupný len tomuto komponentu.

V ďalšom popise budeme považovať klientské stanice za externú časť systému, keďže ich konfigurácia a funkcia sú mimo kontextu systému a ich použitie na účely využívania služieb certifikačnej autority sa nevymyká z rámca ich bežného používania.

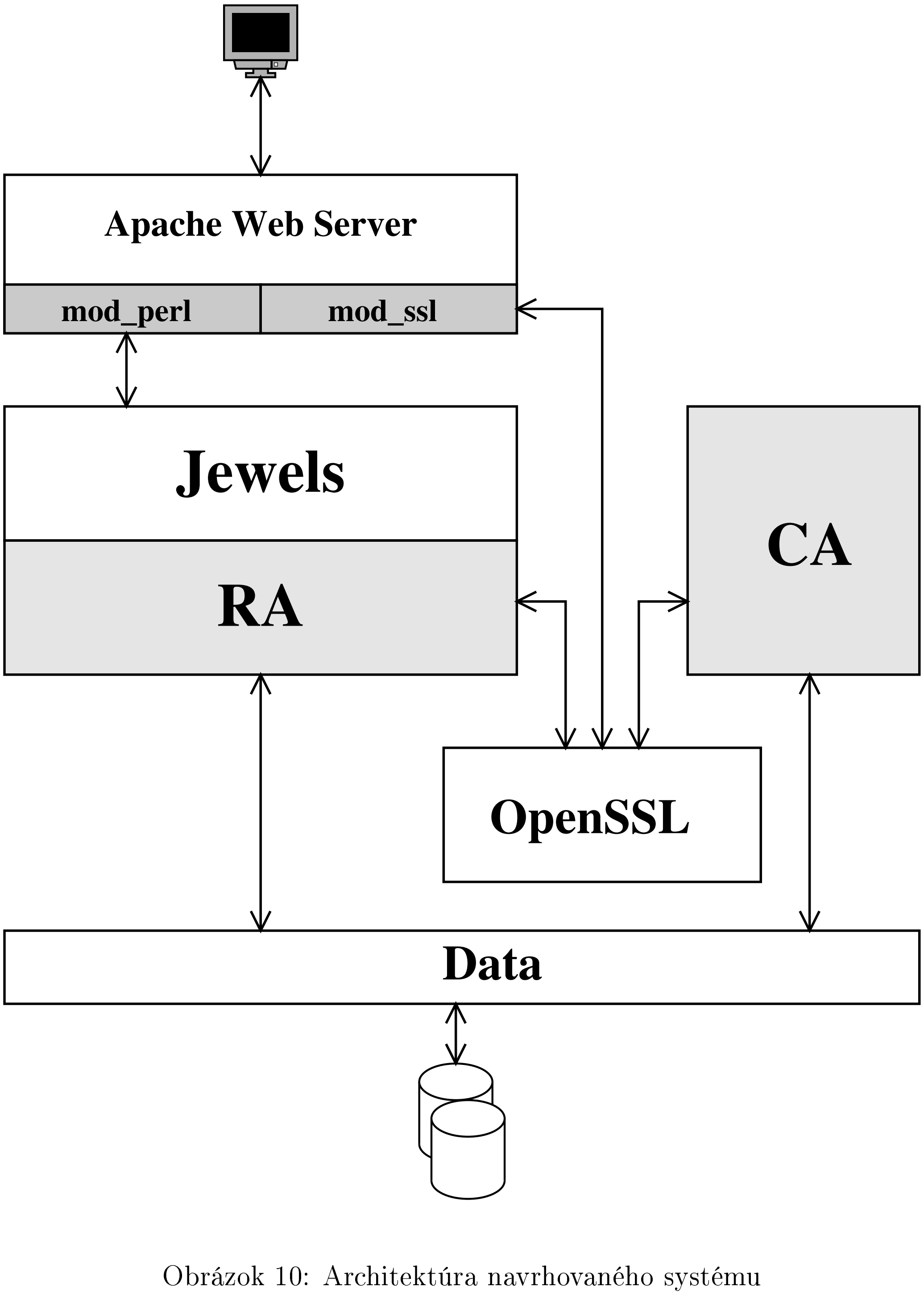
Verejné, operátorské a administračné rozhranie - Registračná Autorita (RA) - je od samotného jadra certifikačnej autority oddelené. Takáto architektúra zaručuje vyšiu bezpečnosť privátneho kľúča certifikačnej autority. Samotná cetifikačná autorita môže dokonca byť fyzicky umiestnená na inom zariadení ako je registračná autorita. Pre systémy s vyšším stupňom bezpečnosti sa doporučuje umiestniť certifikačnú autoritu na zariadenie, ktoré nie je pripojené k žiadnej komunikačnej sieti a údaje prenášať manuálne na vymeniteľných médiách.

5.5 Architektúra systému

Hlavnou časťou celého systému je serverová časť, obsahujúca WWW server Apache, aplikačnú kostru Jewels, samotnú aplikáciu RA, aplikáciu CA a databázu (obrázok 10).

WWW server prijíma protokolové jednotky protokolu HTTP z klientskej stanice. Pomocou svojho modulu mod_ssl spracuje spojenie chránené protokolom SSL (HTTPS). Web server vykoná základnú kontrolu týchto príkazov a pomocou modulu mod_perl vyvolá ovládač aplikačnej kostry Jewels. Aplikačná kostra spracuje zatiaľ nespracovaný vstup od klienta a pripraví prostredie pre vykonanie aplikácie registračnej autority. Ďalej zabezpečuje autentifikáciu používateľov aplikácie, autorizačné prvky, kompletizáciu výstupu a aplikovanie výsledného vzhľadu na výstup aplikácie. Aplikačná kostra je úzko viazaná so samotnou aplikáciou registračnej autority a používa rovnaký spôsob prístupu k databáze ako samotná aplikácia. Kryptografické služby pre aplikáciu a WWW server zabezpečuje kryptografická knižnica OpenSSL [11].

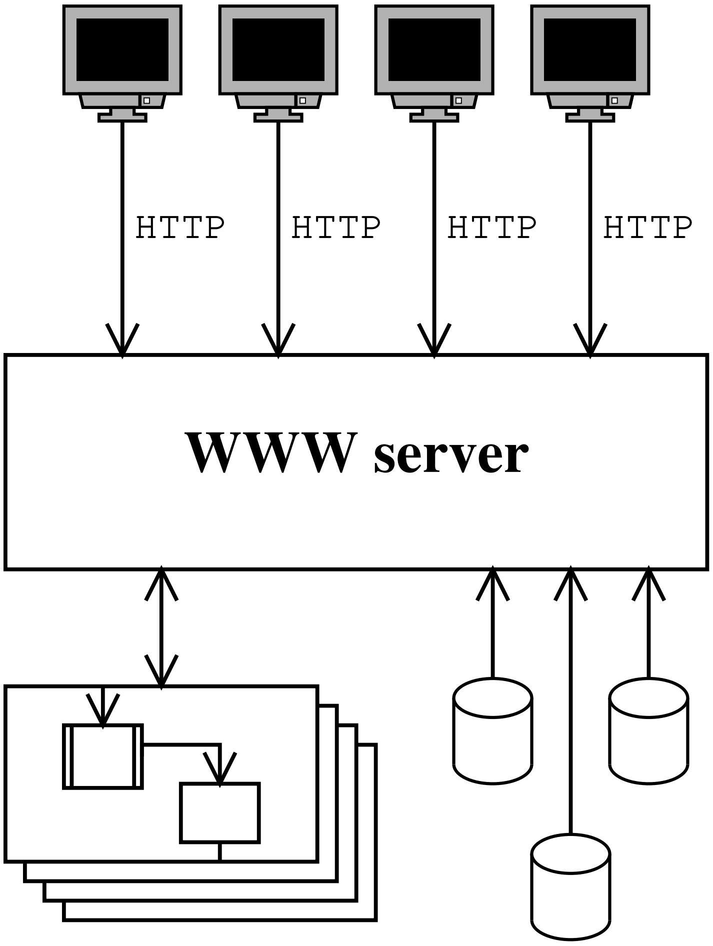
Aplikácia certifikačnej autority je od celého systému WWW servera a aplikačnej kostry čiastočne oddelená. Jediná zdieľaná dátová časť je databáza a jej obslužné moduly (modul Data na obrázku 10). Certifikačná autorita používa rovnakú kryptografickú knižnicu ako WWW server a aplikácia registračnej autority, v prípade nutnosti je možné použiť inú inštanciu tejto knižnice. Celý systém je navrhovaný tak, aby v prípade vyších požiadaviek na bezpečnosť dovolil fyzicky oddeliť registračnú autoritu od samotnej certifikačnej autority.



Obrázok 10: Architektúra navrhovaného systému

5.6 WWW server

WWW server je proces serverovej časti systému, ktorý prijíma protokolové jednotky protokolu HTTP a interpretuje ich. V prípade žiadostí o jednoduché stránky ich WWW server uspokojí vo vlastnej réžií. V prípade žiadostí o dynamicky generovaný obsah, www server odovzdá žiadosť obslužnému programu (Obrázok 11) . Riešenia www serverov bývajú väčšinou silne konfigurovateľné a prispôsobiteľné požiadavkám systému. Rozdiel býva hlavne v generovaní dynamického obsahu v škále podporovaných programovacích jazykov, rozhraní a možností optimalizácie.



Obrázok 11: Funkcia www servera

5.6.1 Apache WWW server

Ako hlavná platforma pre www server bol zvolený www server Apache. Apache je dielom neziskovej korporácie "The Apacha Software Foundation" [12], ktorá vznikla roku 1999 zo združenia jednotlivcov "The Apache Group". Toto združenia pracovalo na vývoji www servera Apacha od roku 1995.

WWW server Apache je modulárna platforma HTTP servera s množstvom vlastností zodpovedajúcich serverom komerčnej triedy. Server Apache bol pre túto aplikáciu zvolený vďaka svojej robustnosti, širokému spektru poskytovaných väzieb na rôzne programovacie jazyky a silným možnostiam jeho integrácie so samotnou aplikáciou. Apache už vo svojom základe podporuje paralelné spracovanie a je stavaný na vysokú záťaž. Taktiež je k dispozícií vo verziách pre väčšinu bežných operačných systémov a harvérových platforiem.

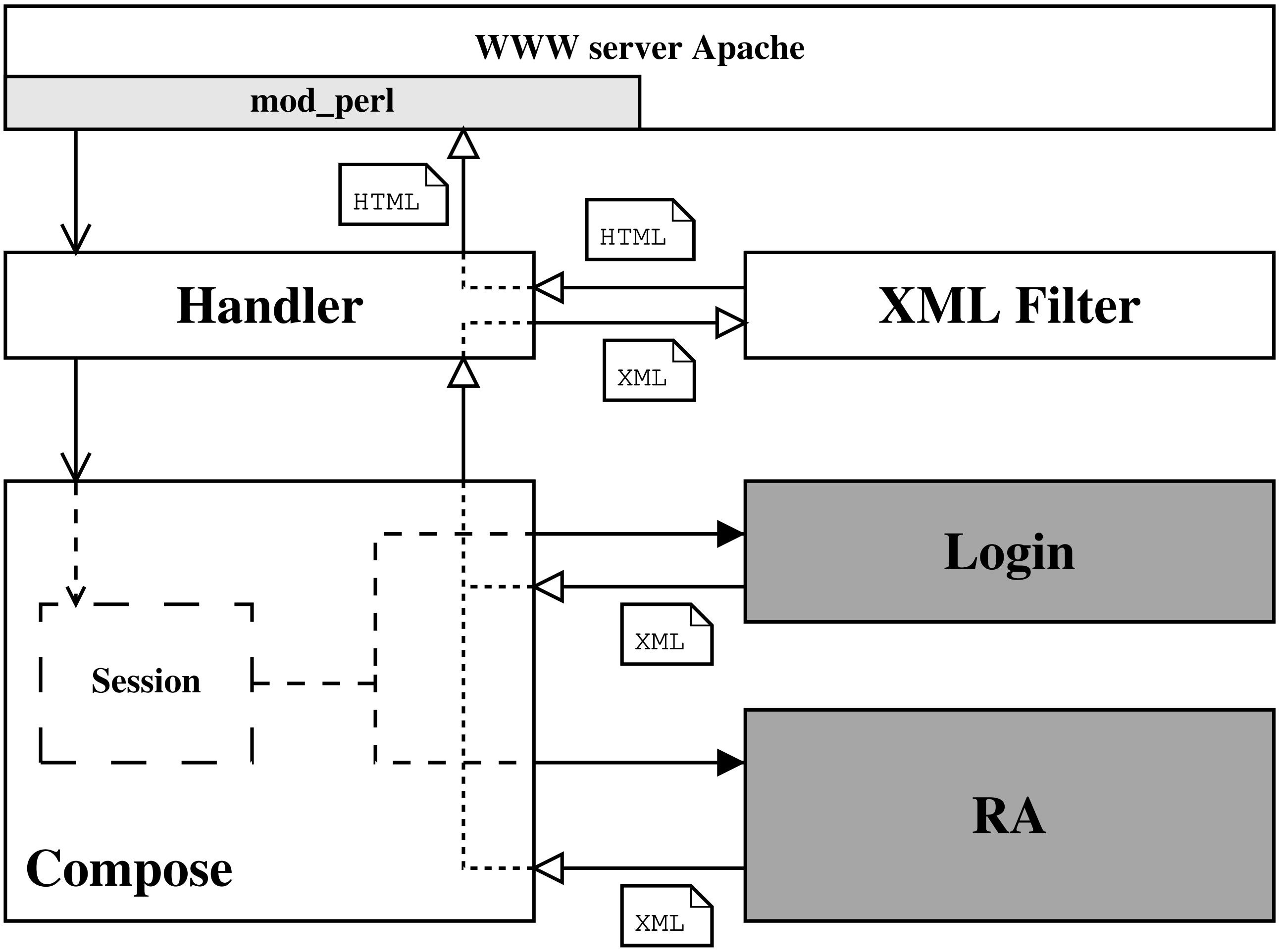
Kľúčovú funkčnosť pre potreby aplikácie registračnej autority poskytujú dva moduly servera Apache. Prvý z nich, modul mod_ssl, pridáva jadru servera možnosť komunikácie zabezpečeným protokolom HTTPS a tiež možnosť overovať osobné certifikáty poskytnuté klientskou stranou. Druhý kľúčový modul je modul *mod_perl*, ktorý zabezpečuje integráciu interných štruktúr www severa s modulmi vykonávanými v jazyku *perl* a tiež podporu vstaveného interpretera tohto jazyka.

5.7 Aplikačná kostra Jewels

Aplikačná kostra predstavuje základné prostredie, v ktorom aplikácia operuje. Je to súbor funkcií a údajových štruktúr, ktorý uľahčuje interakciu aplikácie s okolím a vsádza aplikáciu do celkového kontextu systému. Aplikačná kostra *Jewels* bola vyvinutá špeciálne pre kombináciu WWW servera Apache a programovacieho jazyka Perl. Je integrovaná so štruktúrami WWW servera pomocou jeho modulu *mod_perl,* a preto je vykonávanie celej aplikácie efektívne aj pri použití jazyka vysokej úrovne.

Hlavným formátom definície výstupných údajov je XML [13]. Tento univerzálny, rozšíriteľný formát definície údajov je ideálny na generovanie výstupných zostáv nezávisle na vzhľade ich výslednej prezentácie. Pretože súčasné prehliadače nie sú schopné správne interpretovať tento formát, je v konečnej fáze spracovania požiadavky formát XML konvertovaný na konvenčný formát WWW stránok HTML a takto odoslaný prehliadaču.

Postup spracovania požiadavky je naznačený na obrázku 12. V prvej fáze spracovania je pomocou WWW servera prijatá protokolová jednotka protokolu HTTP/HTTPS. Tá je WWW serverom transparentne spacovaná a pomocou modulu *mod_perl* odovzdaná na ďalšie spracovanie aplikačnej kostre Jewels. Prvým modulom aplikačnej kostry, ktorý prijme požiadavku je modul *Handler.* Tento modul ďalej odovzdá požiadavku v správnom formáte modulu *Compose*, ktorý predstavuje hlavnú riadiacu časť systému. Modul *Compose* vytvorí objekt *Session* z údajov poskytnutých modulom *Handler.* Objekt *Session* reprezentuje používateľskú reláciu a obsahuje všetky parametre a nastavenia nutné na vykonanie požiadavky. V ďalšom kroku je objekt *Session* poskytnutý jednotlivým aplikačným modulom. V našom prípade ide o moduly *Login* a *RA.* Tieto moduly na základe stavu objektu *Session* produkujú výstup vo formáte XML, ktorý obsahuje údaje, ktoré majú byť zobrazené používateľovi. Modul *Compose* tieto údaje vhodne spojí tak, aby tvorili platný XML dokument a tento dokument odošle modulu *Handler.* Modul Handler môže na základe informácií o pristupujúcom prehliadači modifikovať formát výstupného dokumentu tak, aby sa používateľovi správne zobrazil. V štandardnom prípade sa na túto modifikáciu použije modul XML Filter, ktorý na základe dokumentu vo



Obrázok 12: Architektúra aplikačnej kostry Jewels použitá ako základ pre aplikáciu registračnej autority

formáte XML a prislúchajúcich šablón vytvorí dokument vo formáte HTML. Tento dokument sa nakoniec pomocou WWW servera odošle žiadajúcemu klientovi.

5.7.1 Autentifikácia, autorizácia

Okrem spracovania vstupu a produkovania výstupu je hlavnou úlohou aplikačnej kostry autentifikovať používateľov a autorizovať ich akcie v systéme registračnej autority. Autentifikácia sa vykonáva transparentne pri vytváraní objektu *Session*. Ak klient pristupuje pomocou zabezpečeného protokolu HTTPS, je WWW serverom vyzvaný k prezentovaniu osobného certifikátu. V prípade, že klient tento certifikát prezentuje a server je ho schopný overiť, použijú sa údaje z certifikátu na overenie totožnosti používateľa. V tomto prípade sa použije silná autentifikácia na základe generovania zdieľaného tajomstva a dokazovania vzájomnej identity servera a klienta pomocou certifikátov.

V prípade, že nemohlo dôjsť k autentifikácií používateľa pomocou certifikátov, je používateľovi poskytnutá možnosť autentifikovať sa heslom. Zobrazenie samotného formulára umožňujúceho autentifikáciu heslom je realizované pomocou modulu *Login*, samotné overenie autentifikačných detailov je však realizované pri vytváraní objektu relácie *Session*.

Autorizácia akcií v systéme registračnej autority sa uskutočňuje na základe privilégií. Každá akcia v systéme má pridelenú krátku textovú reprezentáciu – privilégium. Pomocou tohto textového reťazca je pri každom vyvolaní akcie overované, či aktuálny používateľ má právo akciu vykonať alebo nie. Privilégia definované v systéme registračnej autority sú sumarizované v tabuľke 7 . Okrem týchto preddefinovaných privilégií existuje špeciálne privilégium označované "*". Toto privilégium dáva svojmu vlastníkovi oprávnenie na akúkoľvek akciu. Pre jednoduchšiu manipuláciu s privilégiami sú tieto zoskupené podľa logických celkov – rolí. Rola je typické zoskupenie privilégií, ktoré sa prideľuje používateľovi aby mohol vykonávať svoju typickú sadu úloh. Definované role a k nim prislúchajúce privilégiá sú zhrnuté v tabuľke 8 [4].

privilégium	poskytuje oprávnenie na:
approve	Potvrdiť alebo zamietnuť žiadosť o certifikát
revoke	Zneplatniť žiadosť o certifikát
listCerts	Umožniť zobraziť zoznam žiadostí a certifikátov
useradmin	Spravovať používateľov a ich práva
config	Konfigurovať parametre RA

Tabuľka 7: Privilégia

rola	privilégia	použitie
raOper	approve, revoke	Operátor registračnej autority
raAdmin	useradmin,config	Správca registračnej autority
superuser	*	Superpoužívateľ

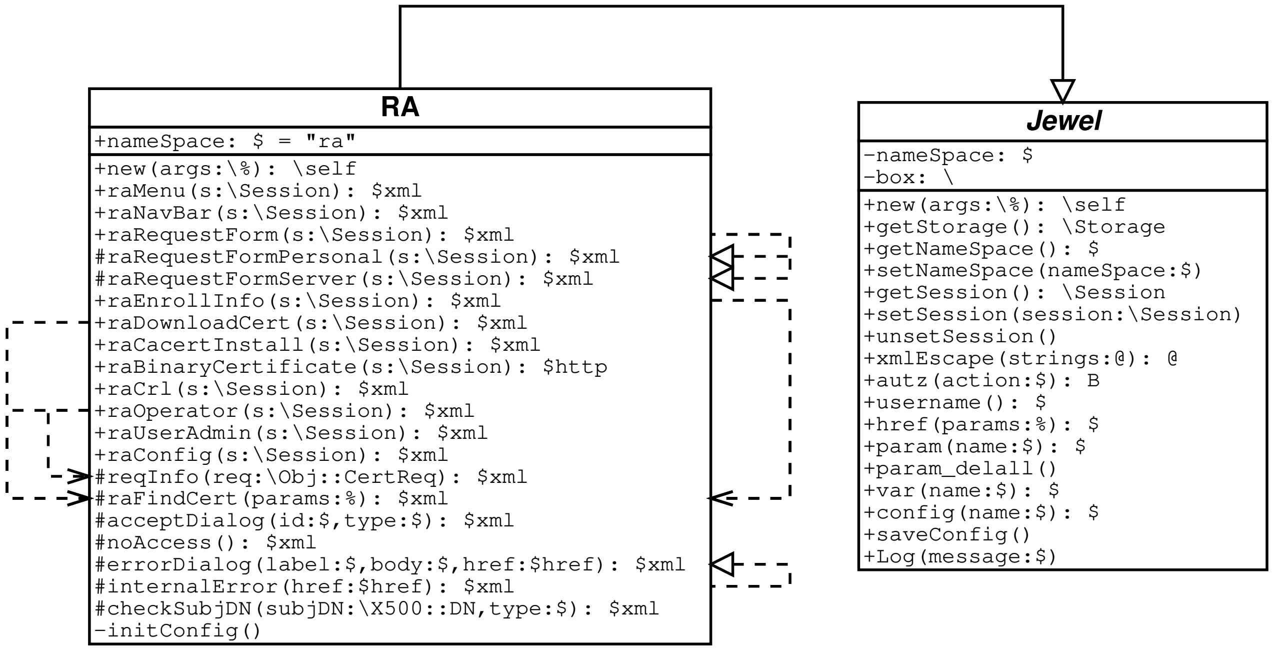
Tabuľka 8: Preddefinované používateľské role

Objektový diagram opisujúci presnú štruktúru autentizačného a autorizačného podsystému je uvedený v prílohe B.

5.8 Registračná autorita

Funkčným jadrom celého systému je aplikácia registračnej autority. Registračná autorita slúži na prímanie nových žiadostí o certifikát, potvrdzovanie, zamietanie a zneplatňovanie certifikátov, poskytovanie informácií o stave certifikátu alebo žiadosti o certifikát, poskytovania samotných certifikátov a ďalšie podporné funkcie. Aplikácia registračnej autority je integrovaná s aplikačnou kostrou *Jewels* a intenzívne využíva jej funkcie.

Funkcie aplikácie sú realizované jedným aplikačným objektom (objekt triedy *RA*), ktorý je odvodený od abstraktnej triedy všeobecnej aplikácie (trieda *Jewel*). Metódy oboch tried sú zobrazené v objektovom diagrame na obrázku 13. Význam jednotlivých metód produkujúcich výstup vo formáte XML je uvedený v tabuľke 9 . Tieto metódy sú volané z modulu aplikačnej kostry *Compose*.



Obrázok 13: Objekt aplikácie registračnej autority

Na vykonanie samotných krytpografických operácií na certifikátoch, žiadostiach o certifikát a zozname zneplatnených certifikátov je použitý modul krytpografickej knižnice *OpenSSL*. Reprezentáciu a ukladanie údajov zabezpečuje modul Data. Modul registračnej autority zabezpečuje len samotnú aplikačnú logiku a spolu s modulom *Compose* aj vysoko-úrovňové riadenie chodu aplikácie.

---

[4] Rolu "superuser", ktorá obsahuje špeciálne privilégium "*", sa z bezpečnostných dôvodov doporučuje ihneď po inštalácií a oživení produktu odstrániť.

metóda	prístup	výstupný formát	význam
raMenu	bez obmedzenia	XML	Zobrazenie úvodnej ponuky
raNavBar	bez obmedzenia	XML	Zobrazenie navigačnej lišty, kontextovo senzitívne
raRequestForm	bez obmedzenia	XML	Zobrazenie formulára na podanie žiadosti o certifikát
raEnrollInfo	obmedzené vyhľadávaním	XML	Zobrazenie informácií o stave certifikátu alebo žiadosti o certifikát
raDownloadCert	obmedzené vyhľadávaním	XML	Zobrazenie formulára pre prevzatie a inštaláciu certifikátu
raCacertInstall	bez obmezenia	XML	Zobrazenie ponuky pre inštaláciu certifikátu CA
raCrl	bez obmedzenia	XML	Zobrazenie zoznamu zneplatnených certifikátov
raOperator	approve,revoke	XML	Rozhranie pre operátora registračnej autority
raUserAdmin	useradmin	XML	Rozhranie pre správcu používateľov
raConfig	config	XML	Formulár pre nastavenie parametrov RA
raBinaryCertificate	bez obmedzenia	HTTP	Prenos binárnych dát certifikátov

Tabuľka 9: Výstupné metódy objektu *RA*

5.9 Reprezentácia a ukladanie údajov

V systéme registračnej autority sú údaje reprezentované vo forme objektov. Tieto objekty sú udržiavané v systéme len počas spravovania HTTP požiadavky, po jej vykonaní sú uložené, ak to ich charakter vyžaduje. Pri ďalšej požiadavke na ten istý objekt je objekt obnovený do pôvodného stavu.

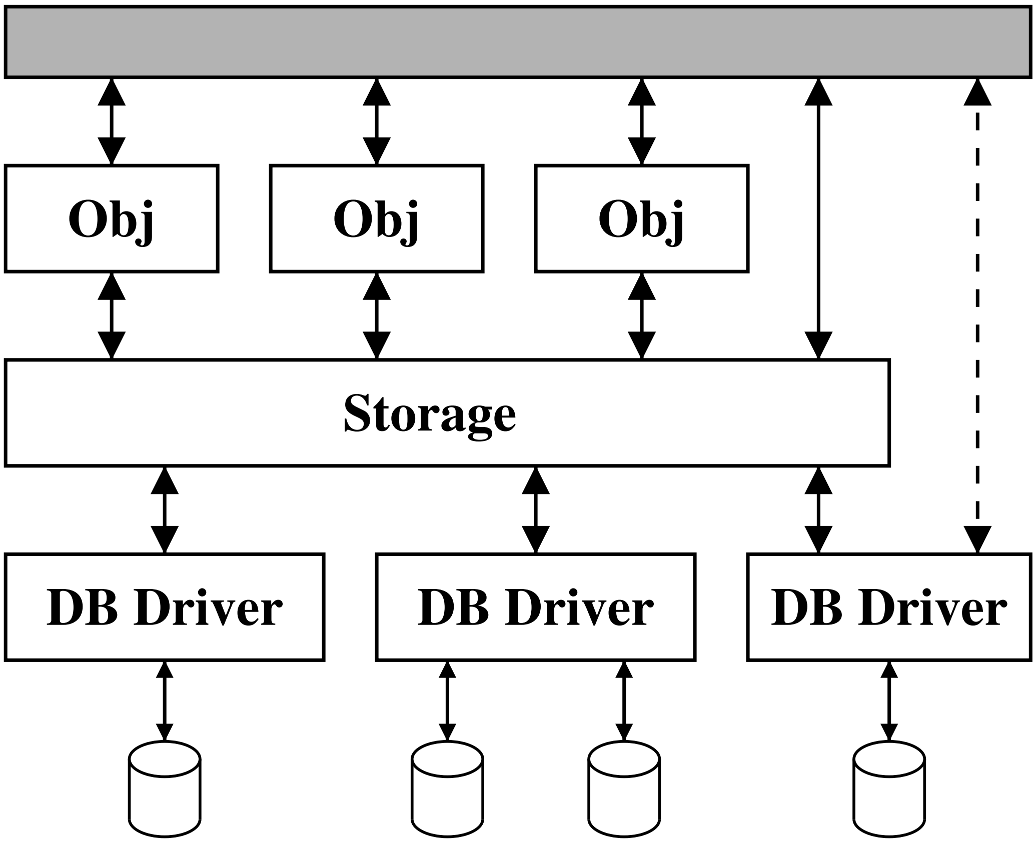
Architektúra reprezentácie a ukladania údajov je vrstvová (obr. 14), pripomínajúca architektúru abstraktného stroja [10]. Najnižšiu úroveň tvorí ovládač databázy. Jeho úlohou je poskytnúť programátorské rozhranie (API) pre prístup k databázovým štruktúram v aplikačnom programovacom jazyku. Na tejto úrovni sa komunikácia odohráva pomocou SQL príkazov alebo nízkoúrovňových volaní funkcií ovládača.

Druhú úroveň tvorí modul Storage. Jeho úlohou je poskytnúť jednotné rozhranie pre rôzne ovládače databázy a rôzne druhy reprezentácie údajov a metód prístupu k nim. Modul pracuje s abstrakciou databázových tabuliek, pričom však tieto tabuľky môžu byť fyzicky implementované napríklad pomocou rôznych súborov. Predpokladá sa, že každá tabuľka má unikátny stĺpec nazvaný *ID*, ktorý je jedinečným identifikátorom riadku v tabuľke (primárny kľúč). Na úrovni modulu *Storage* prebieha komunikácia pomocou volania jednoduchých metód objektu s textovými parametrami.

Tretia úroveň je tvorená rôznymi modulmi reprezentácie údajov[5]. Tieto moduly interpretujú

---

[5] Je to napríklad modul žiadosti o certifikát Obj::CertReq, modul definície používateľa Obj::User, atď.



Obrázok 14: Architektúra reprezentácie a ukladania údajov

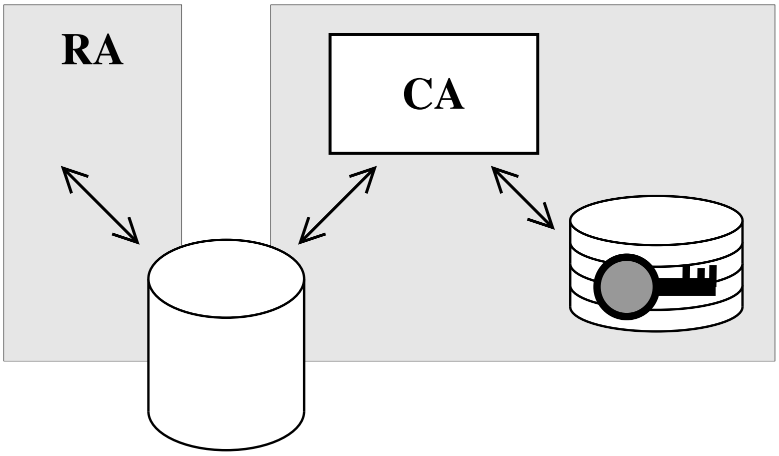
obsah tabuliek poskytnutých modulom *Storage* a realizujú serializáciu a deserializáciu objektov. V tejto vrstve je možné realizovať aj prvky vynucovania konzistencie a vzájomné väzby medzi objektami. Údaje sú zapisované a získavané pomocou volania metód konkrétnych objektov, pričom jeden objekt zodpovedá jednému riadku abstraktnej tabuľky.

Objektové diagramy ilustrujúce podrobnejšiu štruktúru podsystému reprezentácie a ukladania údajov sa nachádzajú v prílohe B.

5.10 Certifikačná autorita

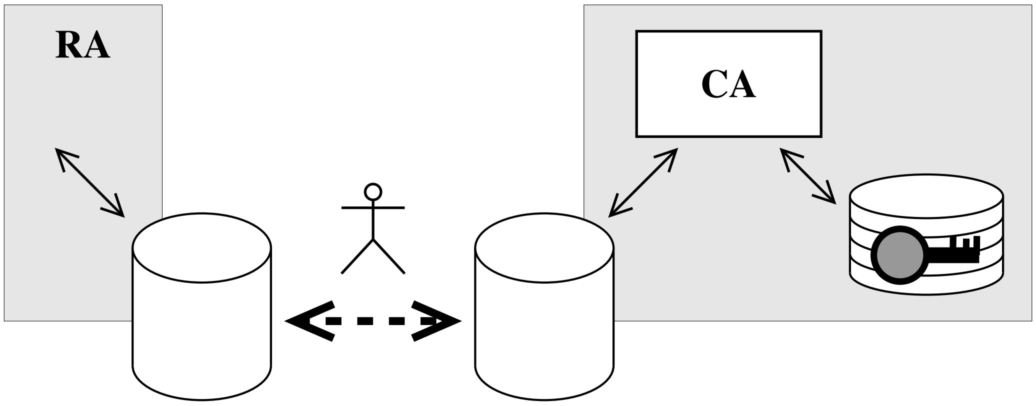
Systém samotnej certifikačnej autority sa v prípade navrhnutej architektúry redukoval len na jednoduchú kryptografickú tranfsormáciu žiadosti o certifikát na samotný certifikát. Táto transformácia je realizovaná pomocou vhodne konfigurovanej aplikácie *ca*, obsiahnutej v kryptografickej knižnici *OpenSSL*. Konfigurácia tejto aplikácie je kľúčovým prvkom certifikačnej autority, pretože definuje formát výsledného certifikátu a tiež politiku pridelovania certifikátov (pozri prílohu E). Okrem podpisovania certifikátov je úlohou certifikačnej autority tiež generovať zoznam zneplatnených certifikátov.

Komunikácia modulu certifikačnej autority so zvyškom aplikácie je realizovaná len pomocou záznamov v zdieľanej databáze (obrázok 15). Takéto usporiadanie minimalizuje možnosti neautorizovaného prístupu k privátnemu kľúču certifikačnej autority ak operačný systém umožňuje



Obrázok 15: Architektúra modulu certifikačnej autority

oddelenie samotnej aplikácie a modulu certifikačnej autority. Taktiež je možnosť certifikačnú autoritu úplne fyzicky oddeliť od zvyšku aplikácie a potrebné údaje replikovať buď manuálne alebo poloautomaticky (obr. 16) . Takáto konfigurácia systému zaručuje podstatne vyšší stupeň bezpečnosti.



Obrázok 16: Oddelenie certifikačnej autority a zvyšku aplikácie

5.11 Kryptografická knižnica OpenSSL

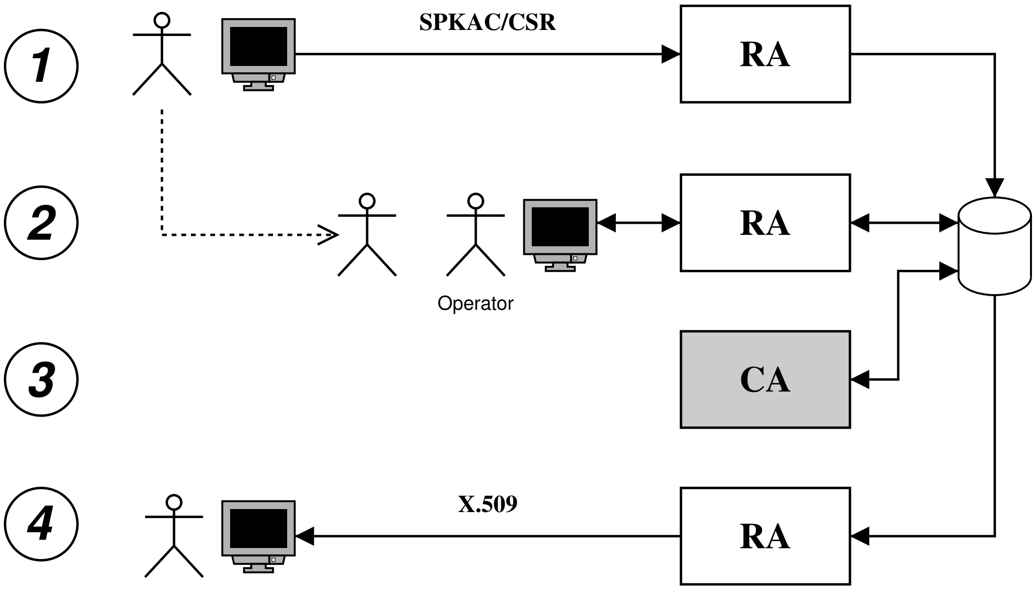
Pre účely registračnej autority a certifikačnej autority bol navrhnutý modul *OpenSSL*, ktorý zabaľuje funkcie knižnice OpenSSL pomocou jednoduchého programátorského rozhrania v implementačnom jazyku. Tento modul sa skladá z výkonnej časti (modul *OpenSSL*) a časti zabaľujúcej kryptografické údaje ako napr. žiadosti o certifikát a certifikáty (modul *OpenSSL::Data*). Funkcie modulu *OpenSSL* zodpovedajú funkciám aplikácie openssl a majú podobné vstupy aj výstupy. Presnejšia špecifikácia modulu *OpenSSL* a *OpenSSL::Data* sú v prílohe B.

Samotné kryptografické transformácie sú realizované pomocou funkcií kryptografickej knižnice OpenSSL. Projekt OpenSSL je kolaboratívna snaha skupiny dobrovoľníkov smerujúca k vytvoreniu kvalitnej implementácie kryptografických funkcií [11]. Projekt je nasmerovaný hlavne na podporu protokolov SSL/TLS, obsahuje však aj iné kryptografické funkcie. Jednou z čŕt OpenSSL je aj intenzívna podpora pre zaobchádzanie s certifikátmi, podpora rôznych formátov a kódovaní certifikačného materiálu, ako aj implementácia algoritmou pre kontrolu a podpisovanie certifikátov. Knižnica OpenSSL je založená na projekte SSLeay iniciovanom Ericom Youngom a Timom Hudsonom.

5.12 Žiadosť o certifikát

Všeobecný popis procesu žiadosti o certifikát bol popísaný v kapitole 2.3.1. Tento proces sa však líši v závislosti od kontrétneho certifikačného systému. V tejto kapitole bude popísaný proces získavania certifikátov tak, ako je realizovaný popisovaným certifikačným systémom.

Proces vydávania certifikátov pozostáva zo štyroch základných krokov (obrázok 17) :



Obrázok 17: Proces vydávania digitálnych certifikátov

1. **Podanie žiadosti o certifikát.** Klient si pomocou svojho programového vybavenia[6] vygeneruje kľúčový pár a v spolupráci s vhodnou HTML stránkou registračnej autority vytvorí a odošle žiadosť o certifikát vo formáte *CSR* alebo *SPKAC*. Vygenerovaná žiadosť je na strane registračnej autority skontrolovaná na formálnu správnosť a platnosť podpisu a je zaradená do databázy.

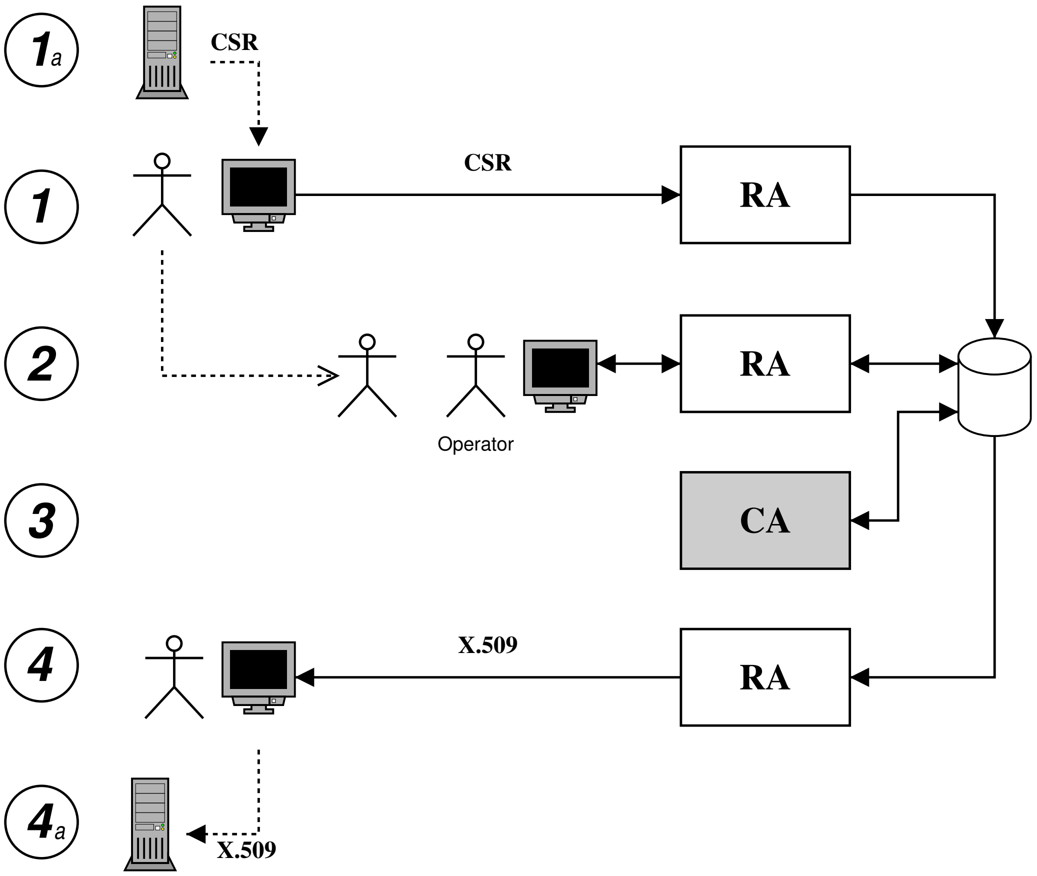
2. **Overenie totožnosti.** Žiadateľ o certitikát potvrdí operátorovi registračnej autority svoju totožnoť. Kontrétne vykonanie tohto kroku je závislé od politiky danej certifikačnej autority. V niektorých prípadoch sa môže vyžadovať osobná prítomnosť žiadateľa v priestoroch registračnej autority, v iných môže stačiť notársky overená žiadosť, atď. Na

základe overenia totožnosti žiadateľa operátor registračnej autority žiadosť potvrdí alebo zamietne.

3. **Podpis certifikátu.** Certifikačná autorita na podnet registračnej autority podpíše potvrdené žiadosti o certifikát. Podpísané certifikáty sa uložia do databázy.

4. **Prebratie certifikátu.** Po podpise certifikátu si žiadateľ môže prevziať certifikát. Certifikát je v spolupráci s klientským softvérom žiadateľa nainštalovaný a správne asociovaný so žiadosťou o certifikát. Od tejto chvíle je možné certifikát používať.

Pri vydávaní certifikátov pre bezpečné servery sú na strane žiadateľa potrebné okrem vyššie uvedených krokov ešte dva dodatočné úkony (obrázok 18). Keďže väčšina serverových systémov nedokáže priamo spolupracovať s certifikačnou autoritou, je nutné aby žiadateľ manuálne vygeneroval kľúčový pár a žiadosť o certifikát pre systém bezpečného servera a takto vygenerovanú žiadosť o certifikát odoslal certifikačnej autorite. Podobne pri získaní certifikátu je nutné tento manuálne nainštalovať do systému bezpečného servera. Spôsoby generovania kľúčových párov a žiadostí o certifikát a inštalovania certifikátov v systémoch bezpečných serverov sú závislé od kontrétneho výrobcu serverového softvéru.



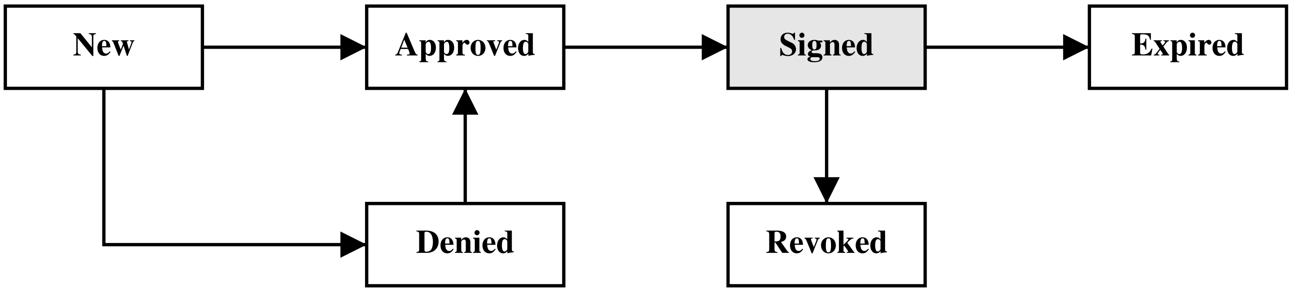
Obrázok 18: Proces vydávania certifikátov pre bezpečné servery

---

[6] Vo väčšine prípadov ide o WWW prehliadač alebo klient elektronickej pošty.

5.13 Stav žiadosti o certifikát

Žiadosť o certifikát sa v systéme môže nachádzať v šiestich rôznych stavoch. Tieto stavy pomocou ich systémových mien zobrazuje obr. 19 a presnejšie sú popísané v nasledujúcom zozname.



Obrázok 19: Stavový diagram žiadosti o certifikát

- **Nová žiadosť (New):** Žiadosť prijatá do systému. Pri svojom vstupe prešla základnými kontrolami konzistencie, žiadna ďalšia akcia však nebola vykonaná.
- **Potvrdená žiadosť (Approved):** Údaje obsiahnuté v žiadosti a autenticita verejného kľúča boli skontrolované operátorom. Žiadosť je pripravená na podpísanie certifikačnou autoritou.
- **Zamietnutá žiadosť (Denied):** Pri kontrole údajov obsiahnutých v žiadosti operátor zistil nekonzistenciu alebo nebola uspokojivo dokázaná autenticita verejného kľúča.
- **Podpísaný certifikát (Signed):** Certifikát bol úspešne vygenerovaný a je platný.
- **Zneplatnený certifikát (Revoked):** Certifikát bol zneplatnený operátorom a umiestnený na zoznam zneplatnených certifiátov certifikačnej autority.
- **Vypršaný certifikát (Expired):** Čas platnosti certifikátu vypršal, certifikát nie je ďalej platný.

6 Implementácia

V tejto časti bude opísaná implementácia systému navrhnutého v predchádzajúcej kapitole. Bude opísaný výber konkrétnych implementačných komponentov, použitie implementačných techník a detaily cieľovej platformy a použitého implementačného jazyka.

6.1 Operačný systém

Systém bol implementovaný paralelne na dvoch operačných systémoch typu UNIX. Vývojová verzia bola prevádzkovaná na platforme SUN Solaris 2.6, systém bol však testovaný hlavne pre prostredie operačného systému Linux. Tieto dve platformy boli zámerne zvolené ako reprezentanti serverových operačných systémov strednej a nízkej triedy, pretože v týchto triedach sa predpokladá uplatnenie certifikačnej autority.

Platforma typu UNIX bola zvolená vďaka svojej vysokej spoľahlivosti. Hlavným dôvodom však bola transparencia štruktúr operačných systémov typu UNIX, čo veľkou mierou prispeje k možnosti kvalitného bezpečnostného auditu celého systému certifikačnej autority. Taktiež implementácie ostatných komponentov systému dosahujú najvyššie kvality práve v prostredí týchto operačných systémov.

Implementácia celého systému na iné varianty UNIXu ako je Solaris a Linux je možná s minimálnymi očakávanými problémami. Implementácia v iných operačných systémoch (napr. Windows NT) je taktiež možná, vyžadovala by si však drobné prispôsobenie aplikácie na lokálne odlišnosti operačných systémov.

6.2 Programovací jazyk Perl

Ako hlavný implementačný jazyk bol zvolený programovací jazyk Perl. Počas implementácia bola s výhodou využité objektové vlastnosti jazyka Perl a celý systém bol implementovaný podľa objektového návrhu. K rýchlemu vývoju tiež prispelo množstvo hotových komponentov dostupných v distribučnej sieti Perlu "CPAN".

6.2.1 Použité moduly

Moduly programovacieho jazyka Perl, použité pri vývoji aplikácie sú sumarizované v tabuľke 10 . Modul CGI::Minimal bol modifikovaný pre korektnú činnosť v perzistentnom prostredí integrovaného interpretera WWW servera Apache. Delta súbor realizujúci túto modifikáciu je v prílohe F.

modul	verzia	autor	popis
DBI	1.13	Tim Bunce	Rozhranie k databázovým ovládačom.
DBD::Oracle	1.03	Tim Bunce	Ovládač databázy Oracle
CGI::Minimal	1.02	Benjamin Franz	Spracovanie CGI formulárov
Apache::Session	1.03	Jeffrey Baker	Perzistencia relácií WWW servera
Data::Dumper	2.101	Gurusamy Sarathy	Serializácia dátových štruktúr
Time-modules	99.062301	David Muir Sharnoff	Práca s časovými údajmi

Tabuľka 10: Použité moduly jazyka Perl.

6.3 Apache web server

V implementačnom prostredí bol použitý Apache web server vo verzií 1.3.9/Unix. Základná konfigurácia servera je sumarizovaná v tabuľke , plný výpis konfigurácie je možné nájsť v prílohe D. Základný adresár, v ktorom je server nainštalovaný je /opt/apache, adresárová štruktúra pod úrovňou tohto adresára je popísaná v tabuľke 12. Samotná aplikácia spolu s aplikačnou kostrou Jewels je uložená v adresári /opt/apache/lib/perl/Jewels.

Verzia	1.3.9
Použité moduly	env, log_config, mime, negotiation, status, include, autoindex, dir, cgi, actions, alias, access, auth, setenvif, ssl, perl
Verzia mod_perl	1.21
Konfigurácia mod_perl	EVERYTHING=1
Verzia mod_ssl	2.4.8-1.3.9
Konfigurácia mod_ssl	–enable-rule=SSL_SDBM
Základný adresár	/opt/apache

Tabuľka 11: Základná konfigurácia WWW servera Apache.

Aplikačná kostra Jewels je spúšťaná v prostredí integrovaného interpretera jazyka Perl. Jednotlivé moduly aplikácie sú do systému zavedené pri jeho štarte a pri obsluhe prichádzajúcich požiadaviek znovupoužité. Riadenie je odovzdané modulu *Handler* aplikačnej kostry v prípade, že prichádzajúca požiadavka je adresovaná na podstrom “/ca” stromu objektov WWW servera.

6.3.1 Obslužné procesy

WWW server Apache pracuje na princípe paralelného spracovania prichádzajúcicho požiadaviek. Jeden z procesov servera je určený na prijímanie požiadaviek z externého prostredia, ostatné procesy požiadavky spracovávajú bez toho, aby na ne hlavný proces musel čakať. Množstvo obsluhujúcich procesov sa prispôsobuje množstvu požiadaviek za čas. Začiatočné množstvo obslužných procesov, limity a dynamické hranice je možné ovládať pomocou parametrov konfiguračného súboru (príloha D).

/opt/apache/		
	conf	konfiguračné súbory pre Apache a jeho moduly
	bin	spustiteľné súrobory servera
	admin	rozhranie pre správu a ladenie aplikácie
	icons	ikony pre prehliadač obsahu adresárov
	img	obrázkové súbory pre aplikáciu
	include	hlavičkové súbory pre aplikácie v jazyku C
	lib	všeobecné knižničné súbory aplikačná kostra Jewels
	logs	záznamové súbory
	man	manuálové stránky servera
	sessions	uložené používateľské relácie
	static	statické HTML stránky
	xml	šablóny pre XML údaje

Tabuľka 12: Adresárová štruktúra inštalácie WWW servera Apache

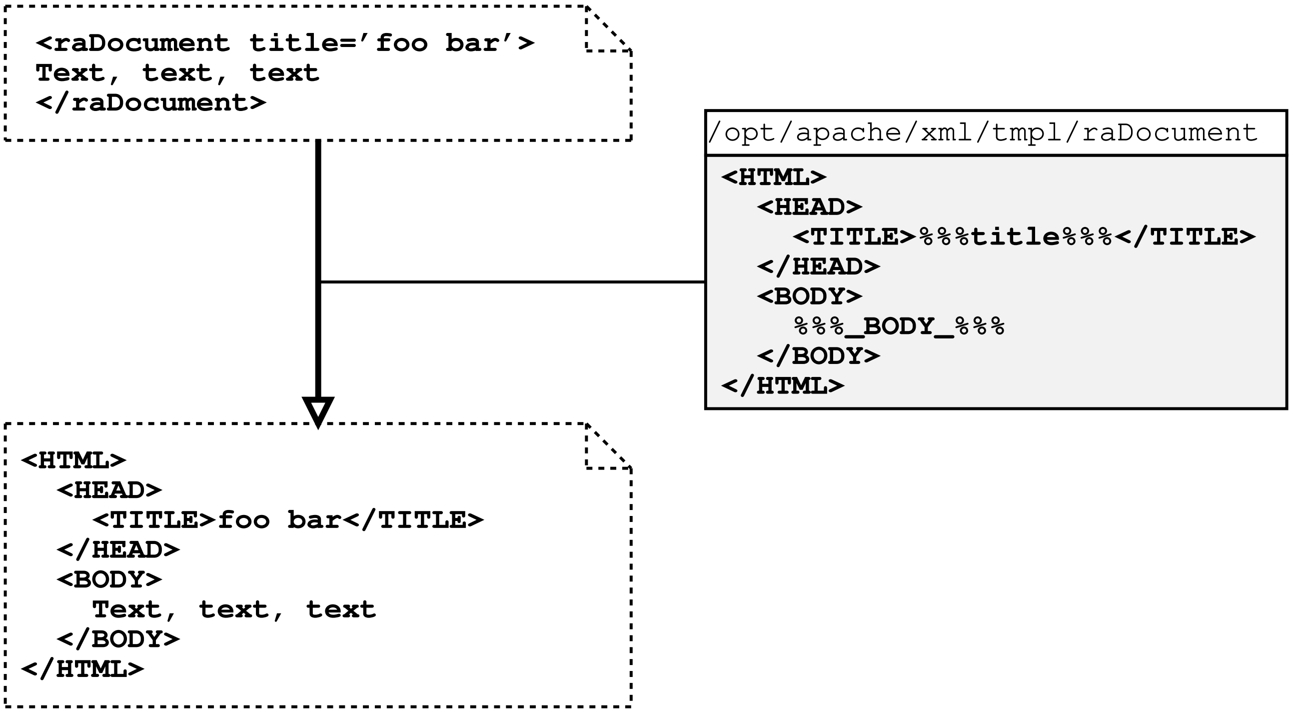
Všetky obslužné procesy WWW servera sa štartujú ako kópie hlavného procesu. To umožňuje moderným operačným systémom šetriť pamäťové prostriedky transparentým zdieľaním rovnakých pamäťových stránok obslužných procesov (mechanizmus *copy-on-write*). Hlavný proces vykoná pred svojim rozvetvením inicializáciu všetkých potrebných modulov a dátových štruktúr. Štart obslužného procesu je potom efektívny, pretože z hlavného procesu zdedí všetky moduly a dátové štruktúry už pripravené a inicializované. Výnimkou z tohto mechanizmu sú databázové spojenia, ktré nie je možné takto kopírovať. Mechanizmus na optimalizáciu zaobchádzania s databázovými spojeniami je popísaný v kapitole 6.4.2.

6.4 Aplikačná kostra Jewels

Pri vývoji systému certifikačnej autority bola použitá aplikačná kostra Jewels vo vývojovej verzií 0.2. Samotná aplikačná kostra sa skladá z množstva modulov jazyka Perl, usporiadaných do hierarchickej štruktúry. Aplikácia je s aplikačnou kostrou tesne previazaná a nachádza sa v tých istých adresárových štruktúrach ako aplikačná kostra.

6.4.1 Vzhľad výstupu

Vzhľad výstupu aplikačnej kostry a tým aj vzhľad celej aplikácie je možné modifikovať pomocou zmeny šablón na preklad XML výstupu na HTML stránky. Tieto šablóny sú štandardne uložené v adresári /opt/apache/xml/tmpl. Každý tu uložený súbor a podadresár má meno zhodné s XML elementom (*"XML tag"*), ktorému zodpovedá. V šablóne je možné vykonávať substitúciu atribútov XML elementu, ako je to ilustrované na obrázku 20. Jednoduchou textovou zámenou je možné realizovať aj komplexné transformácie XML štruktúr do HTML dokumentov. V prípade zložitejších šablón je možné definovať niekoľko šablón pre jeden XML tak, že výber kontrétnej šablóny bude realizovaný v závislosti od hodnoty atribútov XML elementu. Takúto šablónu je možné vytvoriť vytvorením podadresára s menom rovnakým ako je meno XML



```
<raDocument title='foo bar'>
Text, text, text
</raDocument>
```

```
<HTML>
  <HEAD>
    <TITLE>%%%title%%%</TITLE>
  </HEAD>
  <BODY>
    %%%_BODY_%%%
  </BODY>
</HTML>
```

```
<HTML>
  <HEAD>
    <TITLE>foo bar</TITLE>
  </HEAD>
  <BODY>
    Text, text, text
  </BODY>
</HTML>
```

Obrázok 20: Príklad použitia šablóny na preklad XML dokumentu na HTML dokument.

elementu a do tohto podadresára umiestniť jednotlivé šablóny. Meno súboru takto umiestnenej šablóny predstavuje logický výraz (tabuľka 13) pomocou ktorého sa vyberie konkrétna šablóna.

výraz	význam
attr=val	hodnota atribútu *attr* je rovná hodnote *val*
attr=	atribút *attr* je prázdny alebo neexistuje
attr=*	atribút *attr* má ľubovoľnú neprázdnu hodnotu
a1=v1+a2=v2	atribút *a1* má hodnotu *v1* a súčasne atribút *a2* má hodnotu *v2*

Tabuľka 13: Výrazy pre výber komplexných šablón.

Usporiadanie XML elementov typického výstupu aplikácie registračnej autority a význam dôležitých XML elementov sú uvedené v prílohe C.

6.4.2 Prezistencia databázových spojení

Spôsob spracovania požiadaviek klientov je v prostredí WWW založený na transakčnom modeli, bez explicitného určenia kontextu vykonávania aplikácie. Štandardný spôsob obsluhy v tomto prostredí je izolácia jednotlivých transakcií s nastavovaním kontextu vykonávania na začiatku spracovania transakcie a rušenie použitého kontextu na konci spracovania. Takýto prístup však môže priniesť značnú neefektivitu využívania prostriedkov, najmä pri prostriedkoch s drahou inicializáciou. Jedným z takýchto prostriedkou sú aj spojenia aplikácie s databázovým systémom.

Aplikačná kostra Jewels sa snaží otvoriť databázové spojenia hneď po štarte obslužného procesu, pri prvej požiadavke na prístup k databáze. Toto spojenie zostáva aktívne aj po obslúžení transakcie ktorá ho otvorila a čaká na ďaľšiu pridelenú transakciu. Každá ďalšia transakcia toto spojenie využije, bez nutnosti čakania na otvorenie nového spojenia. Spojenie sa uzatvára až keď obslužný proces zaniká v dôsledku poklesu množstva požiadaviek alebo uplynutia jeho trvanlivosti.

6.5 Registračná autorita

Aplikácia registračnej autority je realizovaná modulom jazyka Perl, integrovaným v prostredí aplikačnej kostry Jewels. Tento modul je nazvaný *Jewels::RA* a je uložený v systéme súborov ako súbor */opt/apache/lib/perl/Jewels/RA.pm* a implementuje triedu *Jewels::RA*, ktorej jediná inštancia je aplikácia registračnej autority. Táto inštancia je vytvorená pri štarte servera v hlavnom procese servera a je kopírovaná do jednotlivých obslužných procesov (pozri kapitolu 6.3.1). Aplikácia v prípade potreby komunikuje s kryptografickou knižnicou *OpenSSL* pomocou modulu *Jewels::OpenSSL* (pozri kapitolu 6.7) a s databázou pomocou modulu reprezentáciu údajov *Jewels::Obj::CertReq.* Prístup k binárnym údajom certifikátov a iným kryptografickým údajom realizuje modul *Jewels::OpenSSL::Data.* Na reprezentáciu mien systému *X.500* bol implementovaný modul *X500::DN.* Tento modul má objektový charakter a dokáže rozoznávať, reprezentovať a porovnávať mená vo formáte systému *X.500.* Objektový diagram modulu *X500::DN* je v prílohe B.

6.6 Certifikačná autorita

Funkcie certifikačnej autority sú realizované pomocou programu *ca* v jazyku perl. Tento program môže byť umiestenený kdekoľvek v systéme súborov, štandardne je však uložený pod menom *ca.pl* v domácom adresári vlastníka certifikačnej autority. Tento program realizuje podpisovanie certifikátov a vydávanie zoznamov zneplatnených certifikátov. Prístup do databázy je realizovaný modulom *Jewels::Obj::CertReq* a ku kryptografickým údajom sa pristupuje pomocou modulu *Jewels::OpenSSL::Data.* Činnosť certifikačnej autority je ovládaná manuálne a z veľkej miery je určená obsahom konfiguračného súboru knižnice *OpenSSL* (pozri prílohu E).

6.7 Kryptografická knižnica OpenSSL

Na realizáciu kryptografických funkcií v systéme je použitá kryptografická knižnica OpenSSL. Táto knižnica realizuje všetky potrebné krytpografické transformácie, najmä kontrolu a podpisovanie certifikátov a interpretovanie ich obsahu. Použité funkcie aplikácie openssl, ktorá je súčaťou knižnice OpenSSL sú uvedené v tabuľke 14 . Knižnica OpenSSL spolupracuje s aplikáciami certifikačnej a registračnej autority pomocou modulu Jewels::OpenSSL. Objektový diagram tohto modulu možno nájsť v prílohe B.

funkcia	vstupný formát	popis
req	CSR	generovanie a interpretovanie žiadosti o certifikát
spkac	SPKAC	interpretovania a overonavie žiadosti vo formáte SPKAC
x509	X.509	interpretovanie obsahu a konverzia kódovaní certifikátov
ca	CSR, SPKAC	podpisovanie certifikátov, generovanie CRL

Tabuľka 14: Funkcie aplikácie openssl

6.7.1 Použité verzie knižnice

Kryptografická knižnica OpenSSL bola na realizáciu kryptografických funkcií aplikácií použitá vo vývojovej verzii z 18.11.1999. Posledná vydaná verzia v čase písania tohto dokumentu neobsahovala požadované funkcie potrebné na overenie konzistencie žiadostí o certifikát vo formáte SPKAC generovaných prehliadačom Netscape Navigator. Knižnica v tejto verzií však nebola použiteľná na spoluprácu s WWW serverom Apache, kde musela byť použitá stabilná verzia 0.9.4 knižnice.

6.8 Databáza Oracle

Na ukladanie údajov v systéme bol použitý relačný databázový systém Oracle. Tento databázový systém bol zvolený pre svoju vysokú robustnosť a dostupnosť na implementačných platformách. Na platforme Solaris bol použitý systém Oracle 8.0.4 Enterprise Edition, na platforme Linux to bol Oracle 8.0.5. V databázovom systéme bol vytvorený jediný používateľ, ktorý slúžil na prístup k databáze. Samotný prístup bol realizovaný pomocu databázového ovládača DBD::Oracle a rozhrania relačnej databázy DBI. Oba tieto komponenty sú modulmi jazyka perl a boli použité ako najnižšia vsrtva modelu prístupu k údajom aplikačnej kostry Jewels (pozri kapitola 5.9).

Databáza Oracle bola použitá z dôvodu uľahčenia vývoja a testovania. Výkono tohto databázového systému poskytuje dostatočnú rezervu pre rast systému a jeho škálovateľnosť. V prípade nevhodnosti databázy Oracle z licenčných a cenových dôvodov je možné ju s minimálnou námahou nahradiť za inú databázu (pozri kapitola 5.9).

6.9 Konfiguračný podsystém

Konfiguračné parametre systému sa rozdelujú do dvoch hlavných skupín:

- **Inštalačné parametre.** Tieto parametre určujú rozloženie aplikácií a ich komponentov v súborovom systéme, konfiguráciu WWW servera, konfiguráciu databázového prístupu, atď. Inštalačné parametre sa nastavujú pri inštalácií produktu a počas chodu aplikácie zostávajú nezmenené.

- **Parametre vykonávania aplikácie.** Tieto parametre určujú implicitné nastavenia niektorých hodnôt, konfiguráciu používateľov, atď. Parametre tohto typu sa menia počas chodu aplikácie.

Inštalačné parametre sa v systéme špecifikujú v konfiguračnom súbore WWW servera apache, spolu s vlastnou konfiguráciou WWW servera. Takáto konfigurácia je realizovaná buď ovlyvňovaním premenných prostredia alebo nastavovaním globálnych premenných modulov jazyka Perl (tabuľka 15) . Nastavované globálne premenné patria modulu *Jewel::InitConfig*, ktorý je vytvorený len na zhromažďovanie inicializačnej konfigurácie. Tieto paramentre sú načítané systémom iba raz - pri štarte WWW servera, počas vykonávania aplikácie ich už týmto spôsobom nie je možné meniť.

premenné prostredia	PerlSetEnv ORACLE_SID "ec"
	PerlSetEnv NLS_LANG "american_america.ee8iso8859p2"
globálne premenné modulov	$Jewels::InitConfig::configFilename = "/opt/apache/xml/caMsieInstCert.tmpl";
	$Jewels::InitConfig::raBaseDir = ’/var/opt/Jewels/ca’;
	$Jewels::InitConfig::raOpenSSLBase = "/opt/openssl";

Tabuľka 15: Príklad inštalačných parametrov systému

Parametre vykonávania systému sú meniteľné aj počas vykonávania aplikácie. Ich reprezentácia je uložená v súbore, ktorý predstavuje vykonateľný úsek programu v jazyku Perl a výsledkom tohto vykonania je dátová štruktúra obsahujúca parametre. Ich nastavenie je možné pomocou modulu *Jewels::Config*. Čítanie a zápis takýchto parametrov je možné pomocou metódy *config* triedy *Jewels::Jewel*, z ktorej sú odvodené všetky triedy aplikácií, teda aj aplikácia registračnej autority.

7 Overenie riešenia

Produkt bol testovaný s dvoma bežnými WWW prehliadačmi a dvoma WWW servermi. Testované funkcie a sumarizované výsledky testov sú uvedené v tabuľke 16 . Uplatnené boli nasledujúce testovacie postupy:

produkt	druh	verzia	testovací postup					
			Req	Dl	Sign	Enc	SSL	Auth
Netscape Communicator	klient	4.61,4.7	OK	OK	OK	OK	N/A	OK
MS Internet Explorer	klient	4	OK	OK	N/A	N/A	N/A	OK
MS Outlook Express	klient		N/A	N/A	OK	OK	N/A	N/A
Apache WWW server	server	1.3.9	OK	OK	N/A	N/A	OK	OK
MS Internet Information Server	server		OK	OK	N/A	N/A	OK	N/A

Tabuľka 16: Výsledky testov certifikačnej autority

- **Žiadosť o certifikát (Req).** Klientský softvér sa spojil so stránkou certifikačnej autority a po zadaní príslušných identifikačných informácií používateľom vygeneroval kľúčový pár. Zo zadaných informácií klientský softvér vygeneroval žiadosť o certifikát a odoslal ju certifikačnej autorite. Pri serverových produktoch bolo nutné vygenerovať kľúčový pár a žiadosť o certifikát osobitným postupom a potom žiadosť o certifikát odoslať certifikačnej autorite pomocou WWW formulára.

- **Prevzatie certifikátu (Dl).** Klientský softvér sa spojil s certifikačnou autoritou, našiel hľadaný certifikát a dokázal ho prevziať a pripraviť na používanie. Pri serverových systémoch sa jednalo o manuálne prevzatie certifikátu a jeho inštaláciu.

- **Elektronický podpis (Sign).** Klientský softvér dokázal vytvoriť elektronicky podpísanú správu. Po prijatí podpísanej správy dokázal klientský softvér podpis overiť pomocou certifikátu odosielateľa.

- **Šifrovanie správ (Enc).** Klientský softvér dokázal zašifrovať odosielanú správu pomocou certifikátu príjemcu. Prijatú šifrovanú správu dokázal rozšifrovať.

- **SSL server (SSL).** Server dokázal obsluhovať požiadavky bezpečne pomocou protokolu SSL. Server prezentoval klientom svoj certifikát a klientský softvér bol tento certifikát schopný overiť. Testované s Netscape Navigatorom 4.x a MS Internet Explorerom 4.

- **Autentifikácia (Auth).** Server alebo klient boli schopní participovať vo vzájomnej autentifikácií pomocou pridelených certifikátov.

Všetky aplikovateľné testy prebehli úspešne, môžeme teda konštatovať, že systém naplnil určené požiadavky (príloha A).

8 Zhodnotenie

Implementovaný produkt spĺňa základné požiadavky na funkciu certifikačnej autority v prodtredí malých organizácií. Systém je schopný evidovať prijaté žiadosti o certifikát, overovať ich správnosť, vydávať certifikáty a zneplatňovať certifikáty. Je umožnené aj vyhľadávanie certifikátov a uverejňovanie zoznamu zneplatnených certifikátov. Pre používateľov, operátorov a správcov systému je implementované grafické používateľské rozhranie na báze WWW. Vzhľad rozhrania je modifikovateľný a tak ho je možné prispôsobiť vzhľadu firemných stránok. Autorizácia je možná pomocou statických hesiel, je však podporovaná aj silná autentifikácia založená na protokole SSL. Autorizácia je realizovaná na princípe rolí a privilégií, čo umožňuje dostatočne presne definovať právomoc používateľov.

Systém je implementovaný v prostredí operačného systému UNIX. Hlavné komponenty systému sú WWW servera Apache, aplikačná kostra Jewels a kryptografická knižnica OpenSSL. Všetky použité komponenty sú uvedené v tabuľke17. Komponenty, u ktorých je uvedený integrátor (uvedený kurzívou), sú skupinovým autorstvom niekoľkých subjektov. Väčšina komponentov systému je dostupná bez obmedzení ako voľný softvér. Všetky komponenty je možné použiť bez úhrady na vývoj alebo štúdium.

názov	autor / *integrátor*	verzia	popis
RA	Radovan Semančík		Registračná autorita
CA	Radovan Semančík		Certifikačná autorita
Apache	*Apache Software Foundation*	1.3.9	WWW server
Perl	*Larry Wall*	5.005	Programovací jazyk
Jewels	*Radovan Semančík*	0.2	Aplikačná kostra
OpenSSL	*OpenSSL project team*	snapshot	Kryptografická knižnica
DBI	*Tim Bunce*	1.13	Rozhranie k databázovým ovládačom.
DBD::Oracle	*Tim Bunce*	1.03	Ovládač databázy Oracle
CGI::Minimal	*Benjamin Franz*	1.02	Spracovanie CGI formulárov
Apache::Session	*Jeffrey Baker*	1.03	Perzistencia relácií WWW servera
Data::Dumper	*Gurusamy Sarathy*	2.101	Serializácia dátových štruktúr
Oracle RDBMS	Oracle, inc.	8.0.x	Relačná databáza

Tabuľka 17: Použité komponenty

Práca nadväzuje na predchádzajúci diplomový projekt [1], kde bola certifikačná autorita doporučená ako jedno z kľúčových riešení na vytvorenie kryptografickej infraštruktúry v prostredí Internetu.

9 Záver

Cieľom projektu bolo navrhnúť a implementovať systém certifikačnej autority pre prostredie malých organizácií. Ako hlavné požiadavky boli identifikované požiadavky na dostupnosť služieb, jednoduché ovládanie, bezpečnosť a finančnú nenáročnosť. Systém bol navrhnutý ako aplikácia na báze WWW a interne rozdelený na časť registračnej autority a certifikačnej autority. Systém bol implementovaný v operačnom systéme UNIX.

Hlavným komponentom registračnej autority je WWW server Apache spolu s jeho modulmi. Samotná aplikácia je implementovaná v programovacom jazyku perl využívajúc služby aplikačnej kostry Jewels. Registračná autorita poskytuje intuitívne grafické používateľské rozhranie pre všetky triedy používateľov systému. Podporovaný je aj systém silnej autifikácie používateľov pomocou prezentovania certifikátov a autorizačný systém založený na definícií rolí a prideľovaní privilégií. Systém certifikačnej autority je navrhnutý tak, aby dokázal vykonávať svoje úlohy oddelene od registračnej autority. Jeho hlavným stavebným prvkom je kryptografická knižnica OpenSSL, ktorá realizuje kryptografické transformácie. Systém podporuje reťazenie certifikátov a tak ho je možné zaradiť do kryptografickej infraštruktúry.

Rozvoj modernej kryptografie prispel k širokému rozšíreniu kryptografických bezpečnostných služieb, a tak spôsobil aj vznik infraštruktúry certifikačných autorít. Certifikačná autorita sa stáva základným stavebným prvkom siete dôvery, akousi elektronickou baštou moderného sveta. Dúfajme, že certifikačné autority nesklamú na ne vloženú dôveru a že ich služby budú vždy bezpečné a spoľahlivé.

Referencia

[1] Semančík, R - Ochrana proti prienikom v dátových sieťach, diplomový projekt, Máj 1999

[2] RSA Laboratories: Answers to Frequently Asked Questions About Today's Cryptography, 1996.

[3] Engelschall, R. S.: Mod_ssl user manual, version 2.3, 1999.

[4] Freier, A., Karlton, P., Kocher, P.: The SSL Protocol Version 3.0, Internet draft, November 1996. http://home.netscape.com/eng/ssl3/draft302.txt

[5] Fielding, R., Gettys, J., Mogul, J., Frystyk H. Berners-Lee, T.: RFC2068 Hypertext Transfer Protocol – HTTP/1.1, Január 1997

[6] Ragett, D. HTML 3.2 Reference Specification, Január 1997. http://www.w3.org/TR/REC-html32.html

[7] Verisign Enterprise & Security Solutions, http://www.verising.com/

[8] TrustWise Solutions, http://www.trustwise.com/

[9] Entrust Technologies, http://www.entrust.com/

[10] Sommerville, I.: Software Engeneering, Addison-Weasley, 1996

[11] OpenSSL: Open Source Toolkit for SSL/TLS, http://www.openssl.org/

[12] The Apache Software Foundation, http://www.apache.org/

[13] Microsoft Corporation: XML Tutorial, http://msdn.microsoft.com/xml/tutorial/default.asp

[14] Wall, L., Christiansen, T., Schwartz, R. L.: Programming Perl, O'Reilly, 1996

[15] The Apache Software Foundation: The Apache License, http://www.apache.org/LICENSE.txt

[16] Gereck, E.: Overview of Certification Systems: X.509, CA, PGP and SKIP. August 1998, http://mcg.org.br/cert.htm

[17] Gutmann, P: X.509 Style Guide, September 1999, http://www.cs.auckland.ac.nz/˜pgut001/pubs/x509guide.txt

[18] Young, A, Cicovic, N. K., Chadwick, D.: Trust Models in ICE-TEL, 1996, http://www.darmstadt.gmd.de/ice-tel/

Prílohy

A Katalóg požiadaviek

A.1 Dostupnosť služieb

A.1.1 Služby certifikačnej autority musia byť globálne dostupné.

Služby certifikačnej autority musia byť dostupné na globálnom princípe pomocou informačných sietí. Prístup zo všetkých miest a všetkých platforiem musí byť jednotný, ak to platforma technicky umožňuje (napríklad grafickým rozlíšením zobrazovacej jednotky).

A.1.2 Služby certifikačnej autority nesmú byť silno územne viazané.

Certifikačné služby musí byť možné poskytovať aj bez umiestnenia systémov certifikačnej autority do územnej blízkosti klientov. Certifikačná autorita musí mať možnosť bežného operovania aj z územne vzdialenej lokácie.

A.1.3 Služby certifikačnej autority musia byť poskytované tak, aby sa zaručila kontinuita ich poskytovania v čase.

Princípy, na ktorých je založené poskytovanie certifikačných služieb musia byť natoľko stabilné, štandardizované a zaužívané, aby bola zaručená kontinuita poskytovania týchto služieb aj v prostredí rapídne sa meniacich heterogénnych sietí. Služby certifikačnej autority nesmú byť vo veľkej miere ovplyvnené zmenou transportného mechanizmu, operarčnej platformy ani klientskej časti, pokiaľ je technicky možné takúto odolnosť zabezpečiť.

A.2 Používateľské rozhranie

A.2.1 Interaktívne používateľské rozhranie

Používateľské rozhranie musí dávať používateľovi možnosti jednoducho vykonávať jeho akcie. Rozhranie musí ošetrovať chybové stavy a informovať používateľa v prípade, ak niektorý z chybových stavov nastal. Komunikácia musí byť založená na grafickom princípe, ak to podporuje klientský systém používateľa.

A.2.2 Používateľské rozhranie sa prispôsobuje právam používateľa

Používateľské rozhranie ponúkne používateľovi len tie možnosti, ktoré sú vykonateľné pri jeho súčastných prístupových právach. Ak používateľ nie je autentifikovaný (je *anonymný*), používateľské rozhranie mu dá možnosť autentifikovať sa. Ak je používateľ autentifikovaný, používateľské rozhranie mu musí dať možnosť ukončiť svoju reláciu (“odhlásiť sa”) ak je to technicky možné a účelné.

A.2.3 Modifikovateľný vzhľad používateľského rozhrania

Vzhľad používateľského rozhrania musí byť možné s primeranou námahou prispôsobiť požiadavkám prevádzkovateľa certifikačných služieb.

A.3 Bezpečnosť

A.3.1 Stupeň bezpečnosti primeraný použitiu

Stypeň bezpečnosti, odolnosti voči chybám a cieleným útokom by mal byť primeraný použitiu certifikačnej autority. Systém by mal poskytovať možnosti, ako stupeň bezpečnosti v prípade potreby zvýšiť.

A.3.2 Jednoznačná identifikácia používateľov

Všetci používatelia operujúci v systéme certifikačnej autority musia byť jednoznačne identifikovaní, alebo prehlásení za *anonymných*. Anonymní používatelia majú minimálne práva len na vykonávanie "verejných" činností (napr. podanie žiadosti o certifikát, vyhľadanie certifikátu, atď.). Pri každej operácií v systéme certifikačnej autority, ktorá ma chrakter operácie správcu alebo operátora, musí byť identifikovaný konkrétny fyzický používateľ, ktorý akciu realizuje.

A.3.3 Silný autentifikačný mechanizmus

Autentikácia používateľov musí byť realizovaná uplatnením silnej krytpografickej autentifikácie alebo systémom statických hesiel. Statické heslá majú slúžiť na začiatočnú konfiguráciu prístupových parametrov správcov certifikačnej autority a v produkčnej prevádzke by sa nemali používať. V reálnej prevádzke sa bude používať silný kryptografický autentifikačný mechanizmus, založený na overovaní identít komunikujúcich strán pomocou digitálnych certifikátov.

A.3.4 Prispôsobivý autorizačný mechanizmus

Autorizácia akcií používateľov sa musí vykonávať na základe minimálnych privilégií. Každému operátorovi a správcovi certifikačnej autority musí byť možné prideliť takú sadu prístupových práv, aby mohol vykonávať iba akcie, na ktoré má oraganizačné oprávnenie, ostatné akcie mu budú odopreté. Tento systém musí mať aspoň takú granularitu, aby dokázal rozlíšiť prístupové privilégiá pre operátora certifikačnej autority a správcu používateľov certifikačnej autority.

A.4 Vydávanie certifikátov

A.4.1 Žiadosť o certifikát

Klient certifikačnej autority musí mať možnosť podať žiadosť o certifikát priamo do systému certifikačnej autority pomocou svojho klientského systému. Podanie žiadosti o certifikát sa uskutoční výhradne elektronickou formou, bez nutnosti fyzickej participácie žiadateľa o certifikát v mieste prevádzkovania certifikačnej autority. Ak je softvér klienta schopný spolupracovať s certifikačnou autoritou pri generovaní žiadosti o certifikát, certifikačná autorita musí participovať na tejto spolupráci a prijať takto vygenerovanú žiadosť o certifikát. Podávanie žiadosti o certifikát nie je podmiené predchádzajúcou autentifikáciou používateľa.

A.4.2 Spracovanie žiadosti o certifikát

Operátor certifikačnej autority musí mať možnosť prezerať prijaté žiadosti o certifikát a ich obsah. Operátor musí mať možnosť potvrdiť alebo zamietnuť prijatú žiadosť o certifikát. Potvrdené žiadosti o certifikát sú pripravené na podpísanie, zamietnuté žiadosti zostávajú v tomto stave až do nasledujúceho potvrdenia alebo zrušenia zo systému.

A.4.3 Podpísanie certifikátu

Potvrdené žiadosti o certifikát budú podpísané kľúčom certifikačnej autority. Podpísanie žiadosti sa nemusí nevyhnutne uskutočniť ihneď po potvrdení žiadosti, uskutoční sa až vtedy, keď na to nastanú vhodné technické alebo organizačné podmienky (napr. pravidelne každú hodinu, ale ak bude potvrdených aspon 5 žiadostí, atď.). Podpísané certifikáty prechádzajú do platného stavu, uložia sa v systéme certifikačnej autority a sú k dispozícií na prevzatie.

A.4.4 Prevzatie certifikátu

Klient certifikačnej autority musí byť schopný jednoducho lokalizovať svoj certifikát v systéme a prevzať ho. Certifikát musí byť poskytnutý na prevzatie rovnakým spôsobom, akým bola podaná žiadosť. Ak klientský softvér žiadateľa podporuje certifikačný proces a žiadosť bola vykonaná pomocou tejto vlastnosti, prevzatie certifikátu musí tento proces dokončiť.

A.5 Zneplatňovanie certifikátov

Operátor certifikačnej autority musí mať právo previesť certifikát z platného do zneplatneného stavu. Zneplatnené certifikáty sa pripoja k zoznamu zneplatnených certifikátov certifikačnej autority. Toto pripojenie sa nemusí realizovať ihneď po prevedení certifikátu do zneplatneného stavu, ale až keď nastanú vhodné podmienky.

A.6 Poskytovanie informácií

A.6.1 Informácie o postupe procesu certifikácie

Klient certifikačnej autority musí mať možnosť zistiť, v akom stave sa nachádza jeho certifikát alebo žiadosť o certifikát. Poskytovanie týchto informácií musí byť realizované tak, aby sa minimalizovala možnosť zneužiť žiadateľove osobné údaje. Taktiež musí byť zabránené možnosti získať výpis všetkých, alebo aspoň väčšieho množstva klientov certifikačnej autority.

A.6.2 Certifikáty

Certifikačná autorita musí poskytovať podpísané certifikáty ako verejnú informáciu. Certifikáty musia byť poskytované vo všetkých bežných formátoch a v správnych typoch pre jednotlivé rozšírené klientské systémy. Vyhľadávanie certifikátov musí byť realizované tak, aby sa minimalizovala možnosť získať zoznam osobných údajov väčšieho množstva klientov certifikačnej autority.

A.6.3 Certifikáty a certifikačné reťaze certifikačnej autority

Certifikačná autorita musí verejne poskytovať svoj vlastný certifikát a v prípade, že je tento certifikát časťou reťazca certifikátov, aj celý reťazec certifikátov v správnom formáte. Certifikát certifikačnej autority, prípadne jej reťazec certifikátov, musia byť publikované tak, aby umožňovali pohodlnú inštaláciu do klientskych systémov.

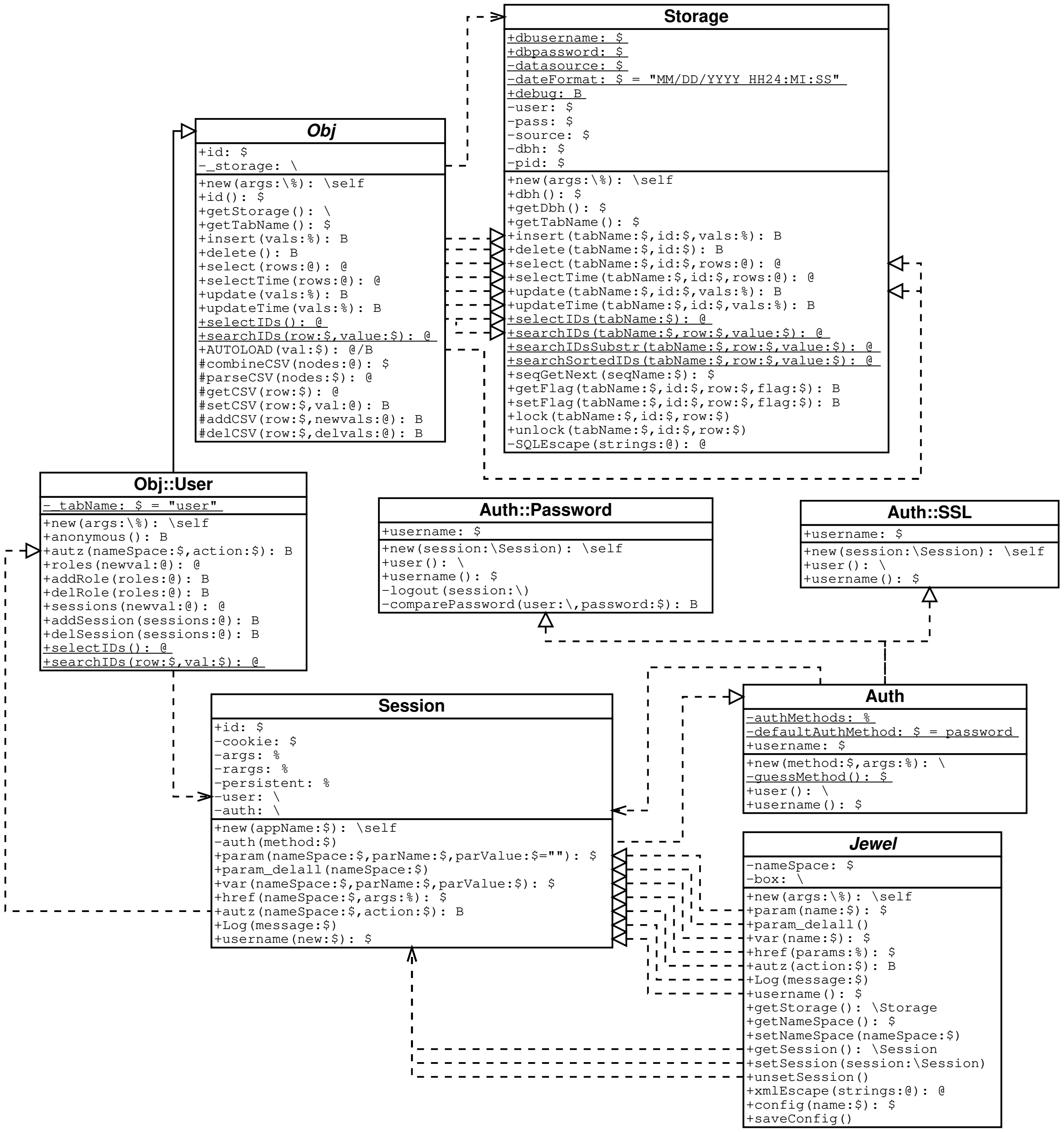
A.6.4 Zoznam zneplatnených certifikátov

Zoznam zneplatnených certifikátov certifikačnej autority musí byť publikovaný pravidelne a správne formátovaný. Aktuálny zoznam musí byť stále k dispozícií klientom certifikačnej autority na overenie platnosti certifikátov.

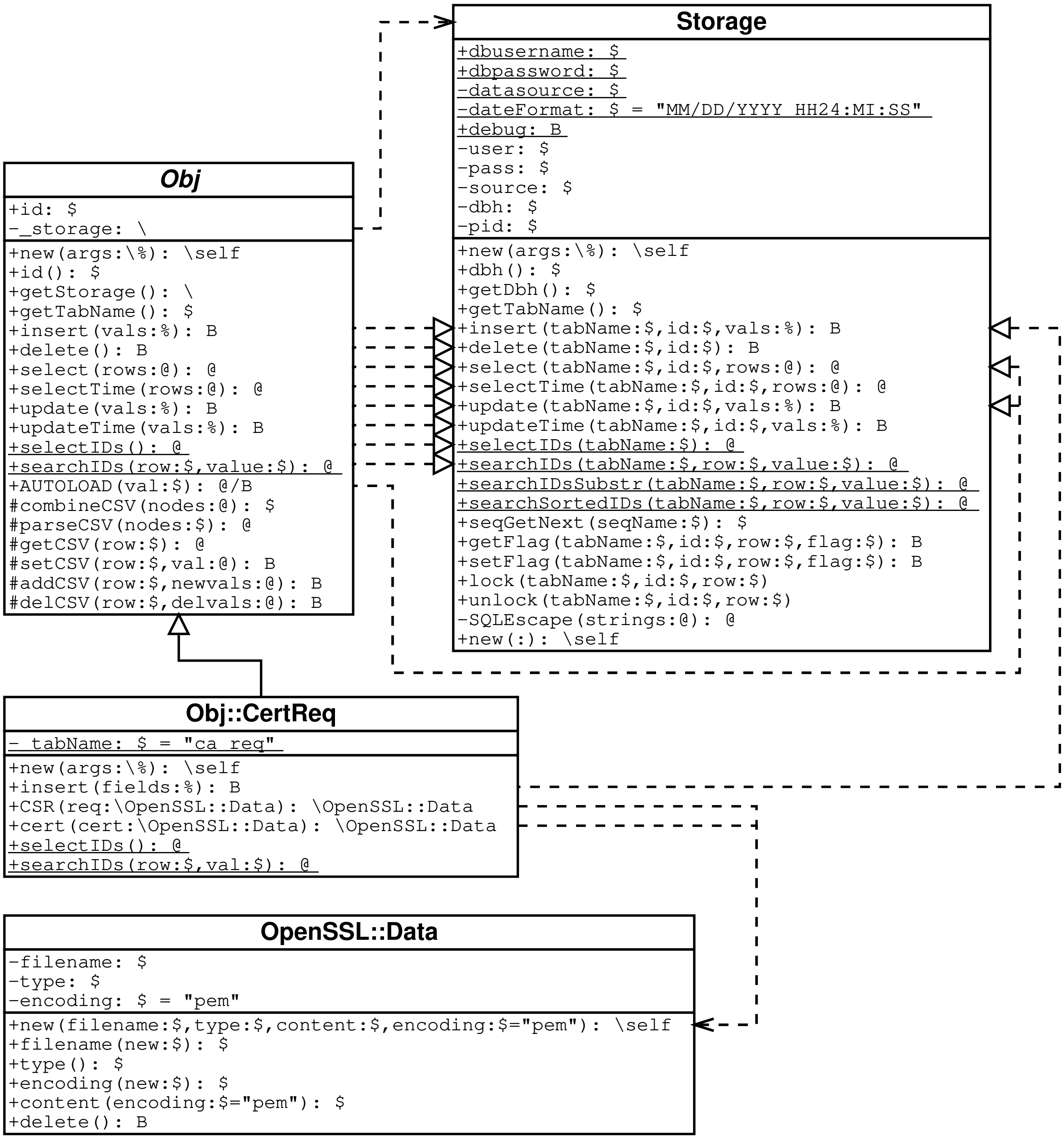
A.7 Finančná dostupnosť

Systém musí pozostávať z takých komponentov, aby náklady na jeho zaobstaranie a prevádzku neprevýšili cenu služieb komerčných poskytovateľov certifikátov.

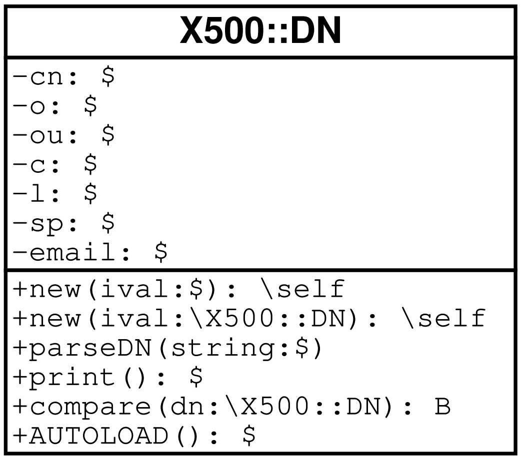
B Objektové diagramy



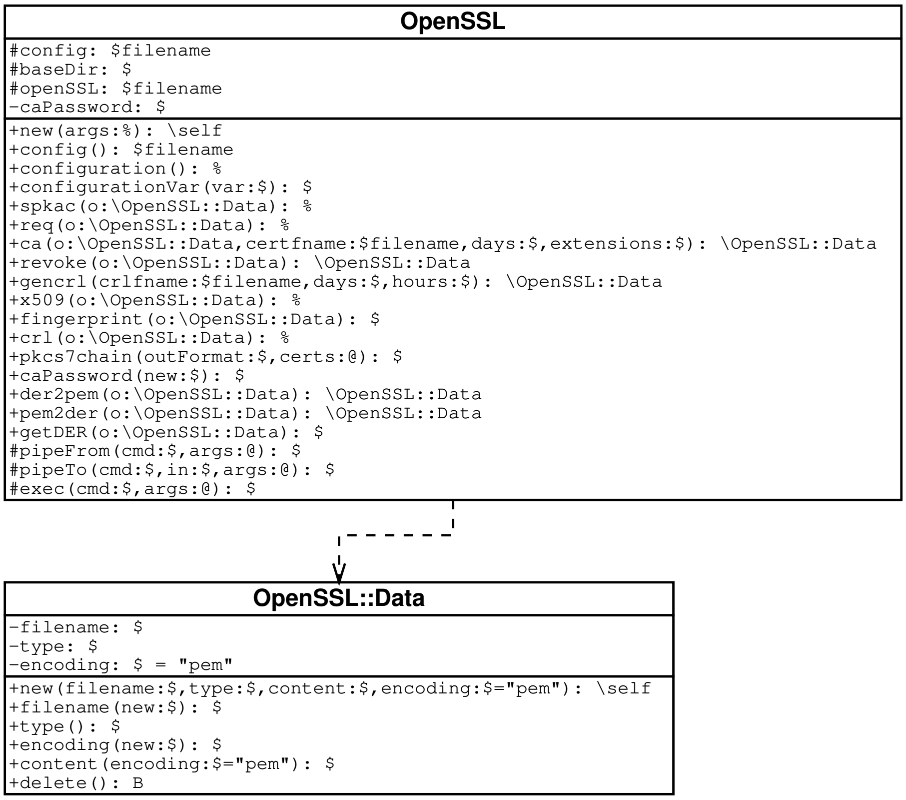
Jewels: Používateľ, relácia, autentifikačný a autorizačný subsystém



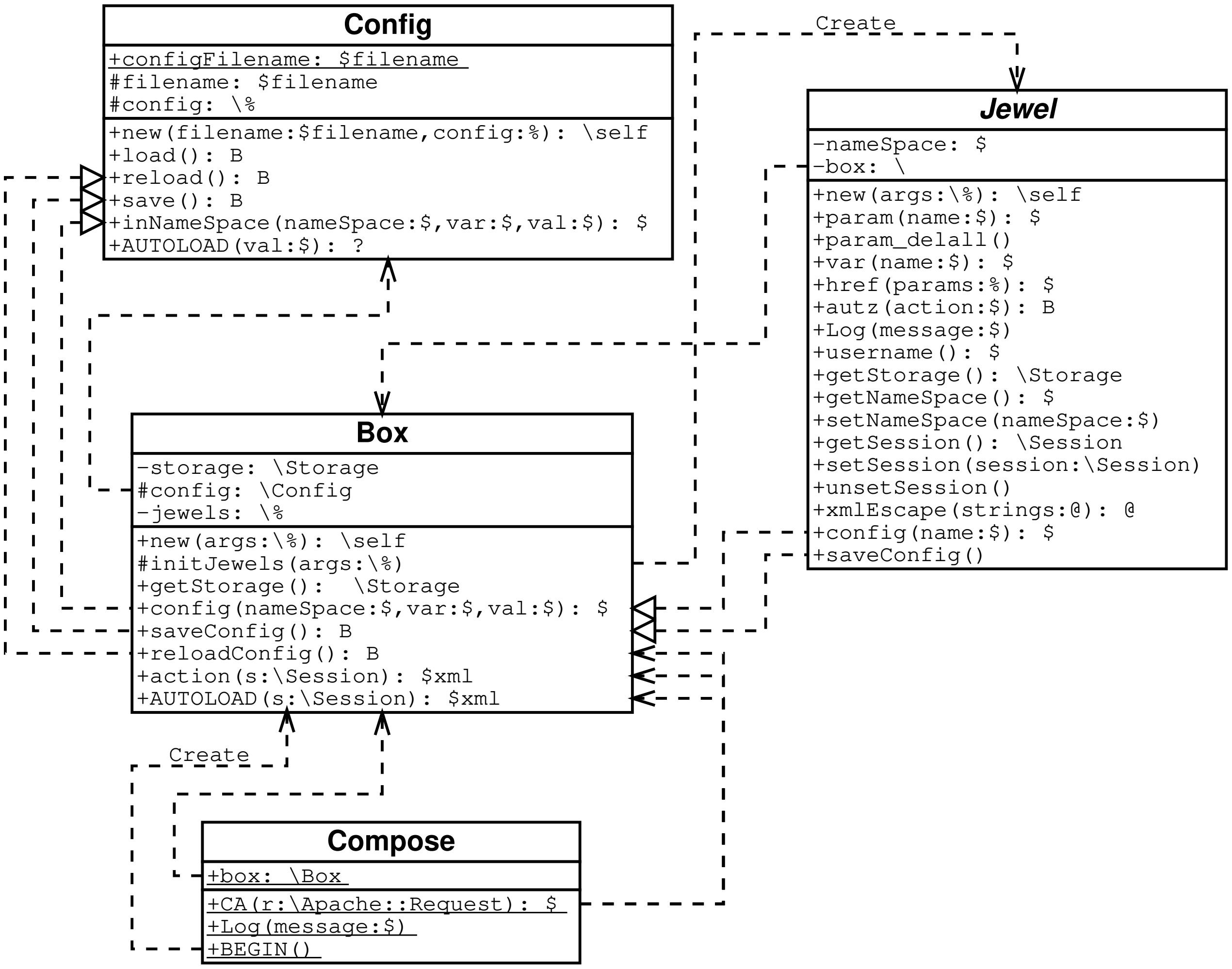
Objektový diagram reprezentácie a ukladania údajov



Objekt reprezentujúci meno štandardu X.500

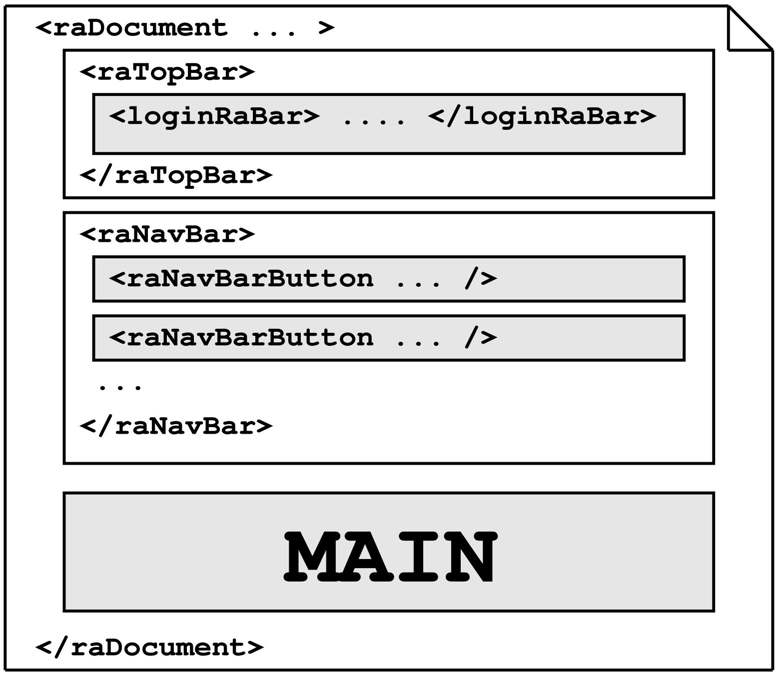


Objektový diagram modulu OpenSSL



Konfiguračný podsystém

C Používateľské rozhranie



```
<raDocument ... >
  <raTopBar>
    <loginRaBar> .... </loginRaBar>
  </raTopBar>
  <raNavBar>
    <raNavBarButton ... />
    <raNavBarButton ... />
   ...
  </raNavBar>
  MAIN
</raDocument>
```

Štruktúra XML dokumentu pred vstupom do výstupného filtra

XML element	vložené elementy	význam
raDocument	všetky ostatné	Kontext dokumentu
raTopBar	loginRaBar	Titulok dokumentu, ...
raNavBar	raNavBarButton	Navigačná lišta
raDialog	raDialogInput raDialogText raButtonPannel	Dialóg používateľského rozhrania
raForm	raFormInput	Formulár na vstup údajov
raRequestForm	raFormInput	Formulár pre žiadosť o osobný certifikát
raListTable	raListTableHeadings raListTableRow	Zoznam položiek vo forme tabuľky
raDownloadDialog	raDownloadDefaultText raDownloadDefaultIcon raDownloadDialog raDownloadFormatTable	Dialóg inštalácie certifikátu
raMenu	raMenuItem	Hlavné menu systému

Význam dôležitých XML elementov.

D Konfigurácia WWW servera Apache

```
##
## httpd.conf -- Apache HTTP server configuration file
##

###############################################################################
### Section 1: Global Environment
###############################################################################
ServerType standalone
ServerRoot "/opt/apache"
#LockFile /var/run/httpd.lock
PidFile /var/run/httpd.pid
ScoreBoardFile /var/run/httpd.scoreboard
#ResourceConfig conf/srm.conf
#AccessConfig conf/access.conf

Timeout 300
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15

# Server-pool size regulation.
###################################
MinSpareServers 5
MaxSpareServers 10
StartServers 5
MaxClients 20
MaxRequestsPerChild 200
ExtendedStatus On

###############################################################################
### Section 2: ’Main’ server configuration
################################################################################

Port 80
Listen 80
<IfDefine SSL>
Listen 443
</IfDefine>
```

```
User apache
Group www

ServerAdmin webmaster@alpha.metropola.sk
#ServerName alpha.metropola.sk
UseCanonicalName On
DocumentRoot "/opt/apache/static"

<Directory />
    Options FollowSymLinks
    AllowOverride None
</Directory>

<Directory "/opt/apache/static">
    Options Indexes FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

DirectoryIndex index.html index.htm index.pl

TypesConfig /opt/apache/conf/mime.types
DefaultType text/plain

<IfModule mod_mime_magic.c>
    MIMEMagicFile /opt/apache/conf/magic
</IfModule>

###########
# Logging #
###########

HostnameLookups Off

LogLevel warn
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
```

```
ErrorLog /opt/apache/logs/error.log
CustomLog /opt/apache/logs/access.log common

#CustomLog /opt/apache/logs/referer_log referer
#CustomLog /opt/apache/logs/agent_log agent
#CustomLog /opt/apache/logs/access_log combined

# Set to one of:  On | Off | EMail
ServerSignature On

# Directory Indexes
#######################

IndexOptions FancyIndexing

AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip
AddIconByType (TXT,/icons/text.gif) text/*
AddIconByType (IMG,/icons/image2.gif) image/*
AddIconByType (SND,/icons/sound2.gif) audio/*
AddIconByType (VID,/icons/movie.gif) video/*
AddIcon /icons/binary.gif .bin .exe
AddIcon /icons/binhex.gif .hqx
AddIcon /icons/tar.gif .tar
AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
AddIcon /icons/a.gif .ps .ai .eps
AddIcon /icons/layout.gif .html .shtml .htm .pdf
AddIcon /icons/text.gif .txt
AddIcon /icons/c.gif .c
AddIcon /icons/p.gif .pl .py
AddIcon /icons/f.gif .for
AddIcon /icons/dvi.gif .dvi
AddIcon /icons/uuencoded.gif .uu
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
AddIcon /icons/tex.gif .tex
AddIcon /icons/bomb.gif core
AddIcon /icons/back.gif ..
AddIcon /icons/hand.right.gif README
AddIcon /icons/folder.gif ^^DIRECTORY^^
AddIcon /icons/blank.gif ^^BLANKICON^^
DefaultIcon /icons/unknown.gif
```

```
#AddDescription "GZIP compressed document" .gz
#AddDescription "tar archive" .tar
#AddDescription "GZIP compressed tar archive" .tgz
ReadmeName README
HeaderName HEADER
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t
AddEncoding x-compress Z
AddEncoding x-gzip gz tgz

Alias /icons/ "/opt/apache/icons/"
<Directory "/opt/apache/icons">
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>



AddLanguage en .en
AddLanguage fr .fr
AddLanguage de .de
AddLanguage da .da
AddLanguage el .el
AddLanguage it .it
AddLanguage sk .sk
AddLanguage cz .cz
LanguagePriority sk cz en de fr

BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0
BrowserMatch "RealPlayer 4\.0" force-response-1.0
BrowserMatch "Java/1\.0" force-response-1.0
BrowserMatch "JDK/1\.0" force-response-1.0



# Server status, server info locations
##########################################

<Location /server-status>
    SetHandler server-status
```

```
    Order deny,allow
    Deny from all
    Allow from .bgs.sk localhost
</Location>

<Location /server-info>
    SetHandler server-info
    Order deny,allow
    Deny from all
    Allow from .bgs.sk localhost
</Location>

################################################################################
### CA #########################################################################

<Perl>
$Jewels::InitConfig::configFilename = "/var/opt/Jewels/config.pl";
$Jewels::InitConfig::raBaseDir = '/var/opt/Jewels/ca';
$Jewels::InitConfig::raOpenSSLBase = "/opt/openssl";
</Perl>

alias /img /opt/apache/img
alias /static /opt/apache/static

# Oracle ...

PerlSetEnv ORACLE_HOME  "/opt/oracle/app/oracle/product/8.0.5"
PerlSetEnv ORAHOME      "/opt/oracle/app/oracle/product/8.0.5"
PerlSetEnv ORACLE_BASE  "/opt/oracle"
PerlSetEnv ORACLE_SID   "alpha"
PerlSetEnv NLS_LANG     "american_america.ee8iso8859p2"
PerlSetEnv ORA_NLS33    "/opt/oracle/app/oracle/product/8.0.5/ocommon/nls/admin/data"

<Perl>
use Jewels::Storage;

$Jewels::Storage::dbusername = "jewels";
$Jewels::Storage::dbpassword = "sesame";
</Perl>

<Location /static>
```

```
    Options Indexes FollowSymLinks
    AllowOverride None
    SetHandler default-handler
    Order allow,deny
    Allow from all
</Location>

<Location /img>
    Options Indexes FollowSymLinks
    AllowOverride None
    SetHandler default-handler
    Order allow,deny
    Allow from all
</Location>

<Location /ca>
        #PerlFixupHandler +Apache::DB
        SetHandler perl-script
        PerlHandler Jewels::Handler::CA
        allow from all
        PerlSendHeader On
</Location>

Perlrequire /opt/apache/conf/perl-startup.pl

<Perl>

$Jewels::Cat::topPageContent="/opt/apache/xml/topPage.xml.tmpl";
$Jewels::Weather::imgPrefix = "/static/apps/pocasie/";

</Perl>

<Location /perl-status>
        SetHandler perl-script
        PerlHandler Apache::Status
        deny from all
        allow from .bgs.sk localhost
</Location>

PerlTaintCheck Off
```

```
###############################################################################
### SSL
###############################################################################
#
#   Some MIME-types for downloading Certificates and CRLs
#
<IfDefine SSL>
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl    .crl
</IfDefine>

<IfModule mod_ssl.c>

SSLPassPhraseDialog  builtin
#SSLSessionCache        none
#SSLSessionCache        shm:/opt/apache/logs/ssl_scache(512000)
SSLSessionCache         dbm:/opt/apache/logs/ssl_scache
SSLSessionCacheTimeout  300
SSLMutex  file:/opt/apache/logs/ssl_mutex
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin

SSLLog      /opt/apache/logs/ssl_engine.log
SSLLogLevel info
</IfModule>

<IfDefine SSL>

<VirtualHost alpha.metropola.sk:443>

  SSLVerifyClient optional
  SSLEngine on
  SSLOptions +OptRenegotiate
  RedirectMatch ^/$ https://alpha.metropola.sk/ca/

  <Location /static>
    Options Indexes FollowSymLinks
    AllowOverride None
    SetHandler default-handler
    Order allow,deny
    Allow from all
```

```
  </Location>

  <Location /ca>
        #PerlFixupHandler +Apache::DB
        SetHandler perl-script
        PerlHandler Jewels::Handler::CA
        allow from all
        PerlSendHeader On
  </Location>

  <Location /img>
    Options Indexes FollowSymLinks
    AllowOverride None
    SetHandler default-handler
    Order allow,deny
    Allow from all
  </Location>

  DocumentRoot "/opt/apache/static"
  ServerName aplha.metropola.sk
  ServerAdmin webmaster@alpha.metropola.sk
  ErrorLog /opt/apache/logs/error.log
  TransferLog /opt/apache/logs/access.log

SSLCertificateFile /opt/apache/conf/ssl/alpha.metropola.sk-cert.pem
SSLCertificateKeyFile /opt/apache/conf/ssl/alpha.metropola.sk-privkey.pem
SSLCertificateChainFile /opt/apache/conf/ssl/alpha.metropola.sk-cert.pem
SSLCACertificateFile /opt/apache/conf/ssl/identity-ca-cert.pem
SSLVerifyDepth  10

SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown

CustomLog /opt/apache/logs/ssl_request.log \
          "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"

</VirtualHost>

</IfDefine>
```

E Konfigurácia OpenSSL

```
#
# openssl.cnf (/var/opt/Jewels/ca/openssl.cnf)
# OpenSSL configuration file for Certification Authority.
#

# Definícia hlavného adresára pre certifikačnú autoritu

CA_BASE                 = /var/opt/Jewels/ca



RANDFILE                = $CA_BASE/.rnd
oid_file                = $CA_BASE/.oid
oid_section             = new_oids

[ new_oids ]

####################################################################
# Definícia certifikačnej autority
# definície adresárov pre uloženie jednotlivých súborov

[ ca ]
default_ca      = CA_default            # The default ca section

[ CA_default ]

dir             = $CA_BASE              # Where everything is kept
certs           = $dir/certs            # Where the issued certs are kept
crl_dir         = $dir/crl              # Where the issued crl are kept
database        = $dir/index.txt        # database index file.
new_certs_dir   = $dir/newcerts         # default place for new certs.

certificate     = $dir/cacert.pem       # The CA certificate
serial          = $dir/serial           # The current serial number
crl             = $dir/crl.pem          # The current CRL
private_key     = $dir/private/cakey.pem# The private key
RANDFILE        = $dir/private/.rand    # private random number file


# Tieto konfiguračné direktívy sú nepoužité - prednosť má definícia v
# systéme registračnej autority

x509_extensions = usr_cert              # The extentions to add to the cert

# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
# so this is commented out by default to leave a V1 CRL.
```

```
# crl_extensions        = crl_ext

default_days    = 365                   # how long to certify for
default_crl_days= 30                    # how long before next CRL
default_md      = md5                   # which md to use.
preserve        = no                    # keep passed DN ordering

# Definícia politiky certifikačnej autority.
# Štandardne nastavené politika policy_anything dovolí podpísať čokoľvek, čo
# registračná autorita potvrdí

policy          = policy_anything

# For the CA policy
[ policy_match ]
countryName             = match
stateOrProvinceName     = match
organizationName        = match
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional

# For the 'anything' policy
# At this point in time, you must list all acceptable 'object'
# types.
[ policy_anything ]
countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional

####################################################################
# CSR generation definitions
#
# Definície generovania CSR, nie su v systéme použité
#
[ req ]
default_bits            = 1024
default_keyfile         = privkey.pem
distinguished_name      = req_distinguished_name
attributes              = req_attributes
#x509_extensions        = v3_ca # The extentions to add to the self signed cert

dirstring_type = nobmp
```

```
# req_extensions = v3_req # The extensions to add to a certificate request

[ req_distinguished_name ]
countryName                     = Country Name (2 letter code)
countryName_default             = SK
countryName_min                 = 2
countryName_max                 = 2

stateOrProvinceName             = State or Province Name (full name)
stateOrProvinceName_default     =

localityName                    = Locality Name (eg, city)
localityName_default            = Bratislava

0.organizationName              = Organization Name (eg, company)
0.organizationName_default      =

# we can do this but it is not needed normally :-)
#1.organizationName             = Second Organization Name (eg, company)
#1.organizationName_default     =

organizationalUnitName          = Organizational Unit Name (eg, section)
organizationalUnitName_default  =

commonName                      = Common Name (eg, YOUR name)
commonName_max                  = 64

emailAddress                    = Email Address
emailAddress_max                = 40

# SET-ex3                       = SET extension number 3

[ req_attributes ]
challengePassword               = A challenge password
challengePassword_min           = 4
challengePassword_max           = 20

unstructuredName                = An optional company name

[ v3_req ]

# Extensions to add to a certificate request

basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment

################################################################################
# Certificate Extensions Section
```

```
################################################################################
#
# Extensions defined here will be applied to approppriate certificate types.
# Mapping of certificate types and extensions section is done in CA batch
# processing script.
#
# Definície rozšírení certifikátov pre jednotlivé typy certifikátov.
#
################################################################################

#########################
# Personal Certificate  #
#                       #
#  Osobný certifikát    #
#########################

[ personal_cert ]

# This is end-user (e.g. no CA) certificate
basicConstraints=CA:FALSE

# Set client certificate type for Netrscape Browsers
nsCertType = client, email

# ... alternative containg object signing
# nsCertType = client, email, objsign

# keyUsage set for a client certificate
keyUsage = nonRepudiation, digitalSignature, keyEncipherment

# This will be displayed in Netscape's comment listbox.
# CONFIGURABLE
nsComment                       = "OpenSSL Generated Certificate"

# PKIX recommendations harmless if included in all certificates.
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer:always

# This stuff is for subjectAltName and issuerAltname.
# Import the email address. Copy subject details.
subjectAltName=email:copy
issuerAltName=issuer:copy

# URI of CRL distribution ...

# Definícia distribučného bodu CRL, *NUTNÉ MODIFIKOVAŤ*
```

```
crlDistributionPoints=URI:http://sun.bgs.sk/ca/get?reqID=crl

nsCaRevocationUrl              = http://sun.bgs.sk/ca/get?reqID=crl
#nsBaseUrl
#nsRevocationUrl
#nsRenewalUrl
#nsCaPolicyUrl
#nsSslServerName


##################################
# Server Certificate             #
#                                #
# Certifikát bezpečného servera #
##################################

[ server_cert ]

# This is end-server (e.g. no CA) certificate
basicConstraints=CA:FALSE

# Set server certificate type for Netrscape Browsers
nsCertType = server

# keyUsage set for a server certificate
keyUsage = nonRepudiation, digitalSignature, keyEncipherment

# This will be displayed in Netscape's comment listbox.
# CONFIGURABLE
nsComment                      = "OpenSSL Generated Certificate"

# PKIX recommendations harmless if included in all certificates.
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer:always

# This stuff is for subjectAltName and issuerAltname.
# Import the email address. Copy subject details.
subjectAltName=email:copy
issuerAltName=issuer:copy

# URI of CRL distribution ...

# Definícia distribučného bodu CRL, *NUTNÉ MODIFIKOVAŤ*

crlDistributionPoints=URI:http://sun.bgs.sk/ca/get?reqID=crl

nsCaRevocationUrl              = http://sun.bgs.sk/ca/get?reqID=crl
#nsBaseUrl
```

```
#nsRevocationUrl
#nsRenewalUrl
#nsCaPolicyUrl
#nsSslServerName


#######################
# Sub-CA Certificate  #
#######################

# Not used - yet
# Nepoužité

[ ca_cert ]

# This is a CA certificate (not critical extension, some software will fail)
basicConstraints=CA:true

# Set CA certificate type for Netrscape Browsers
nsCertType = sslCA, emailCA, objCA

# keyUsage set for a CA certificate
keyUsage = cRLSign, keyCertSign

# This will be displayed in Netscape's comment listbox.
# CONFIGURABLE
nsComment                      = "OpenSSL Generated Certificate"

# PKIX recommendations harmless if included in all certificates.
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer:always

# This stuff is for subjectAltName and issuerAltname.
# Import the email address. Copy subject details.
subjectAltName=email:copy
issuerAltName=issuer:copy

# URI of CRL distribution ...

# Definícia distribučného bodu CRL, *NUTNÉ MODIFIKOVAŤ*

crlDistributionPoints=URI:http://sun.bgs.sk/ca/get?reqID=crl

nsCaRevocationUrl              = http://sun.bgs.sk/ca/get?reqID=crl
#nsBaseUrl
#nsRevocationUrl
#nsRenewalUrl
#nsCaPolicyUrl
```

```
#nsSslServerName

###############################################################################
# CRL Extensions Section
###############################################################################
#
# Extensions defined here will be applied to generated CRLs.
#
# Not used due to Netscape Navigator bug ...
#
# Rizšírenie CRL, nepoužité
#
###############################################################################


[ crl_ext ]

# CRL extensions.
# Only issuerAltName and authorityKeyIdentifier make any sense in a CRL.

# issuerAltName=issuer:copy
authorityKeyIdentifier=keyid:always,issuer:always
```

F Delta súbory

F.1 CGI::Minimal

Modifikovaný súbor sa nazýva *Minimal.pm* a nachádza sa v distribúcií balíka *CGI-Minimal* a adresári *lib/CGI/*.

```
31a32,33
> # Modified to work in mod_perl by Radovan Semancik (semancik@alert.sk)
>
108c110
<       if ($form_initial_read) {
—
> #     if ($form_initial_read) {
111c113
<       }
—
> #     }
```

G Používateľská príručka

G.1 Používateľské prostredie

Keď sa pomocou svojho prehliadača používateľ spojí so serverom registračnej autority, objaví sa mu používateľské prostredie podobné tomu, čo ukazuje obrázok 21. V hornej časti prostredia sa nachádza titulok certifikačnej autority a v pravom hornom rohu prostredia je umiestnený autentifikačný dialóg[1]. Pod titulkom a autentifikačným dialógom je umiestnená navigačná lišta. Tu sa budú objavovať tlačidlá, ktoré majú používateľovi uľahčiť navigáciu prostredím. Pod navigačnou lištou je hlavná časť prostredia, kde práve môžno vidieť výber možných činností. Obsah tejto časti sa bude líšiť v závislosti od vykonávanej akcie.

http://sun.bgs.sk/ca/?login%3aaction=logout

Certifikačná Autorita

Žiadosť o osobný certifikát

Žiadosť o certifikát bezpečného servera

Stav certifikačného procesu

Prebrať certifikát

Nainštalovať certifikát CA

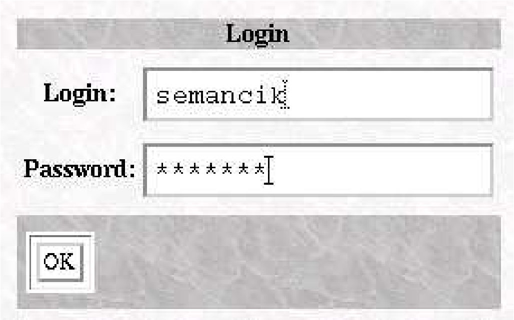
Zoznam zneplatnených certifikátov

Obrázok 21: Celkový vzhľad použávateľského prostredia registračnej autority

V ponuke sa budú zobrazovať len akcie, na ktoré má používateľ momentálne právo. Pri prvom vstupe do systému má práva používateľa *anonymous*, čiže má právo len na verejne prístupné služby. Na iného používateľa sa možno kedykoľvek autentifikovať použitím autenfitikačného dialógu v pravom hornom rohu prostredia (obrázok 22). Po úspešnom prihlásení sa tento dialóg zmení na informáciu o práve prihlásenom používateľovi spolu s možnosťou odhlásenia sa (obr. 23).

---

[1] Ak pristupujete pomocou protokolu SSL a máte osobný certifikát, ktorý systém rozonáva, tento dialóg bude v stave "autentifikovaný".



Obrázok 22: Autentifikačný dialóg

Obrázok 23: Autentifikovaný používateľ

G.2 Hlavné menu systému

Pomocou hlavného menu je možné sa dostať ku všetkým funkciám systému prástupným z grafického rozhrania (obrázok 24) . Jednotlivé položky menu sú popísané v nasledujúcom zozname:

- **Žiadosť o osobný certifikát:** formulár, ktorý umožňuje podať žiadosť o osobný certifikát.
- **Žiadosť o certifikát bezpečného servera:** formulár umožňujúci odoslať žiadosť o certifikát servera.
- **Stav certifikačného procesu:** dialóg umožňujúci zisťovať stav procesu certifikácie, žiadosti o certifikát a stav samotného certifikátu.
- **Prebrať certifikát:** dialóg umožňujúci používateľovi prebrať vygenerovaný certifikát v rôznych formátoch.
- **Nainštalovať certifikát CA:** dialóg umožňujúci používateľovi nainštalovať certifikát CA v rôznych formátoch.
- **Zoznam zneplatnených certifikátov:** výpis zoznamu zneplatnených certifikítov.
- **Operátor:** prostredie pre operátora registračnej autority.
- **Správa používateľov:** prostredie pre správu používateľov a ich oprávnení.
- **Konfigurácia RA:** prostredie pre zmenu parametrov registračnej autority.

Obrázok 24: Hlavné menu systému pri plných prístupových právach

G.3 Žiadosť o osobný certifikát

Dialóg žiadosti o osobný certifikát spolupracuje s klientským softvérom, preto je rôzny pre rôzne prehliadače. Príklad dialógu pre prehliadač Netscape Communicator je na obrázku 25. Tento dialóg obsahuje vo verziách pre všetky prehliadače tieto položky:

Žiadosť o osobný certifikát

Meno: Radovan Semancik

E-mail: semancik@bgs.sk

Organizácia: Business Global Syst

Organizáčná jednotka: System division

Mesto: Bratislava

Región:

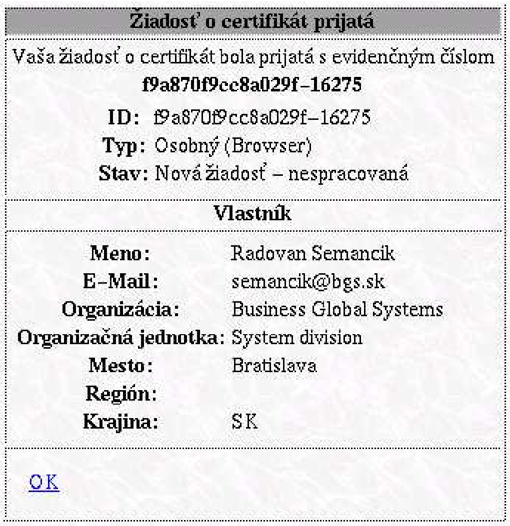
Krajina: SK

Dĺžka kľúča 1024 (High Grade)

Obrázok 25: Žiadosť o osobný certifikát

- **Meno:** Meno osoby, ktorá je vlastníkom certifikátu.
- **E-Mail:** Elektronická adresa majiteľa certifikátu.
- **Organizácia:** Organizácia, pod ktorú majiteľ certifikátu patrí.
- **Organizačná jednotka:** Organizačná jednotka pod ktorú majiteľ certifikátu prislúcha (napr. oddelenie alebo divízia).
- **Mesto:** Identifikácia lokality, v ktorej sa vlastník certifikátu nachádza.
- **Región:** Špecifikácia regiónu, v ktorom sa nachádza vlastník certifikátu.
- **Krajina:** Dvojpísmenové (ISO) označenie krajiny.

Všetky položky môžu byť zadávané len v základnej znakovej sade ASCII, bez použitia diakritiky. Niektoré verzie tohto dialógu môžu obsahovať aj ďaľšie položky, ako napríklad požadovanú dĺžku vygenerovaného kľúča (obr. 25). Po vyplnení príslušných polí je žiadosť odoslaná pomocou tlačidla "OK". Pri odosielaní žiadosti dochádza ku generovaniu kľúčového páru a samotnej žiadosti a počas tohto procesu môžu byť zobrazené dodatočné dialógy. Vzhľad týchto dialógov závisí od konkrétnej verzie použitého prehliadača.



Žiadosť o certifikát prijatá

Vaša žiadosť o certifikát bola prijatá s evidenčným číslom
f9a870f9cc8a029f-16275

ID: f9a870f9cc8a029f-16275
Typ: Osobný (Browser)
Stav: Nová žiadosť – nespracovaná

Vlastník

Meno:	Radovan Semancik
E-Mail:	semancik@bgs.sk
Organizácia:	Business Global Systems
Organizačná jednotka:	System division
Mesto:	Bratislava
Región:	
Krajina:	SK

Obrázok 26: Prijatie žiadosti o certifikát

Po úspešnom odoslaní žiadosti o certifikát, server certifikačnej autority zobrazí prijaté informácie spolu s identifikačným číslom žiadosti (obr. 26) . Toto identifikačné číslo môže byť vyžadované ako dôkaz podania žiadosti pracovníkom obsluhy registračnej autority. Dodatočné inštrukcie uvedené v tomto dialógu je možné konfigurovať pomocou konfiguračného rozhrania (pozri kapitolu G.9).

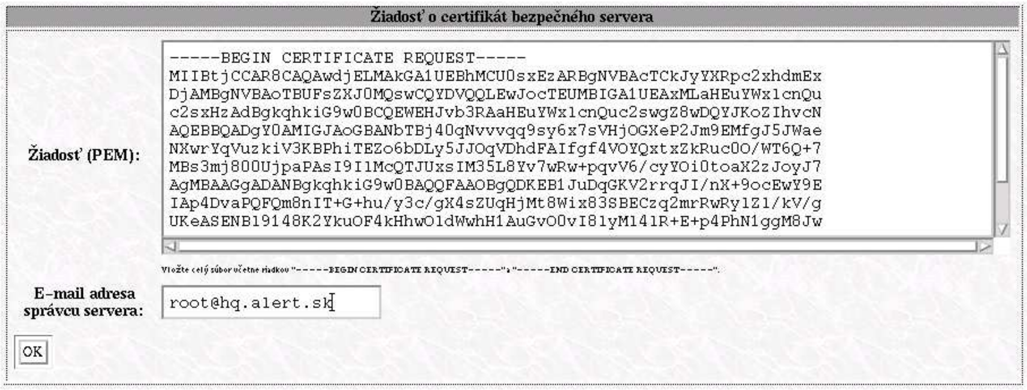
G.4 Žiadosť o certifikát bezpečného servera

Žiadosť o certifikát bezpečného servera obsahuje všetky položky ako žiadosť o osobný certifikát (kapitola 25) s malými rozdielmi:

- Žiadosť o certifikát nie je generovaná automaticky, je nutné ju vygenerovať manuálne v spolupráci s cieľovým serverom. Príklad generovania žiadosti o certifikát pomocou kryptografickej knižnice OpenSSL (Apache Web server) je uvedený v kapitole G.4.1.
- **Meno** predstavuje meno servera. V prípade servera na Internete alebo Intranete je to jeho plné doménové meno (FQDN), pod akým ho vidia pristupujúci klienti.
- **E-Mail** je adresa správcu alebo prevádzkovateľa servera. Môže byť špecifikovaná buď priamo v žiadosti o certifikát alebo v špeciálnom poli formulára žiadosti o certifikát.

Žiadosť o certifikát sa odošle do systému registračnej autority pomocou WWW formulára (obr. 27). Do veľkého textového poľa je nutné vložiť (*"copy&paste"*) žiadosť o certifikát v kódovaní PEM a vyplniť elektronickú adresu správcu servera, ak nie je uvedená priamo v žiadosti. Po stlačení tlačidla OK sa žiadosť o certifikát odošle.

Dialóg prijatia žiadosti o certifikát je rovanký ako pri osobných certifikátoch, taktiež platia rovnaké obmedzenia na obsah jednotlivých položiek (pozri kapitola G.3).



Žiadosť (PEM):

```
-----BEGIN CERTIFICATE REQUEST-----
MIIBtjCCAR8CAQAwdjELMAkGA1UEBhMCU0sxEzARBgNVBAcTCkJyYXRpc2xhdmEx
DjAMBgNVBAoTBUFsZXJ0MQswCQYDVQQLEwJocTEUMBIGA1UEAxMLaHEuYWx1cnQu
c2sxHzAdBgkqhkiG9w0BCQEWEHJvb3RAaHEuYWx1cnQuc2swgZ8wDQYJKoZIhvcN
AQEBBQADgY0AMIGJAoGBANbTBj40qNvvvqq9sy6x7sVHjOGXeP2Jm9EMfgJ5JWae
NXwrYqVuzkiV3KBPhiTEZo6bDLy5JJOqVDhdFAIfgf4VOYQxtxZkRuc0O/WT6Q+7
MBs3mj800UjpaPAsI9I1McQTJUxsIM35L8Yv7wRw+pqvV6/cyYOi0toaX2zJoyJ7
AgMBAAGgADANBgkqhkiG9w0BAQQFAAOBgQDKEB1JuDqGKV2rrqJI/nX+9ocEwY9E
IAp4DvaPQFQm8nIT+G+hu/y3c/gX4sZUqHjMt8Wix83SBECzq2mrRwRy1Z1/kV/g
UKeASENB19148K2YkuOF4kHhwO1dWwhH1AuGvO0vI81yM141R+E+p4PhN1ggM8Jw
```

Vložte celý súbor včetne riadkov "-----BEGIN CERTIFICATE REQUEST-----" a "-----END CERTIFICATE REQUEST-----".

E-mail adresa správcu servera: root@hq.alert.sk

Obrázok 27: Žiadosť o certifikát bezpečného servera

G.4.1 Príklad generovania žiadosti o certifikát pomocou OpenSSL

Kryptografická knižnica OpenSSL (*http://www.openssl.org/*) obsahuje aplikáciu *openssl*, ktorá môže slúžiť aj na generovanie kľúčových párov a žiadostí o certifikát. Takýto spôsob správy certifikátov pomocou externej knižnice využíva napríklad server Apache. Pomocu funkcie req aplikácie openssl možno jednoducho vygenerovať naraz kľúčový pár aj žiadosť o certifikát. Typický postup je uvedený na obrázku 28. Po spustení príkazu "openssl req" prebehlo generovanie kľúčového páru, zašifrovanie privátneho kľúča zvoleným heslom a vygenerovanie žiadosti o certifikát na základe zadaných údajov. Privátny kľúč je uložený ako súbor s menom "privkey.pem" v aktuálnom adresári a verejný kľúč je zapísaný do žiadosti o certifikát. Samotná žiadosť o certifikát je uložená do súboru špecifikovanom parametrom "-out", čiže v tomto prípade ide o súbor "server-req.pem" v aktuálnom adresári. Obsah tohto súboru je potom nutné vložiť do formulára žiadosti o certifikát registračnej autority.

Súbor s privátnym kľúčom (privkey.pem) je nutné precízne chrániť, pretože každý, kto sa dostane k jeho obsahu[2] môže server impersonifikovať. Najlepšie je ihneď po vygenerovaní súboru zakázať prístup ostatným používateľom k jeho obsahu. V prípade, že generovanie je vykonávané na multipoužívateľskom systéme (nie na pracovnej stanici) doporučuje sa vykonávať generovanie kľúčov v adresári, ku ktorému má prístup len jeho vlastník. V prípade, že požadujete uložiť privátny kľúč v otvorenom tvare[3], môžete pri vytváraní žiadosti o certifikát pridať prepínač "-nodes". V tomto prípade je však nutné oveľa úzkostlivejšie dbať na utajenie obsahu súboru s privátnym kľúčom.

G.5 Vyhľadávanie certifikátu

Na niektoré akcie v systéme registračnej autority je nutné špecifikovať certifikát, na ktorom sa má akcia vykonať. Túto špecifikácia certifikátu alebo žiadosti o certifikát je možné uskutočniť pomocou dialógu vyhľadávania certifikátu (obr. 29). Vyhľadávať je možné podľa čísla žiadosti,

---

[2] Útočník musí dokázať tento obsah rozšifrovať, ak je kľúč šifrovaný. Šifra, ktorá kľúč chráni je v aktuálnej verzii DES s dĺžkou kľúča 56 bitov.

[3] Napríklad pre potreby automatického štartu servera.

```
radovan:[~/work/ca]>openssl req -new -out server-req.pem
Using configuration from /opt/openssl/ssl/openssl.cnf
Generating a 1024 bit RSA private key
.........+++++
.....................................+++++
writing new private key to 'privkey.pem'
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:SK
State or Province Name (full name) [Some-State]:.
Locality Name (eg, city) []:Horna Dolna
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Ja a syn, s.r.o.
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:www.jaasyn.sk
Email Address []:webmaster@jaasyn.sk

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
radovan:[~/work/ca]>
```

Obrázok 28: Príklad vytvorenia kľúčového páru a žiadosti o certifikát pomocou aplikácie *openssl*

Vyhľadať žiadosť/certifikát

Meno majiteľa / servera: Odin

Stav: Nedefinovaný

ID	Stav	Meno	Email	Sériové číslo
e74a5287f2b229e5-29666	Platný	Odin	odin@valhala.gov	6

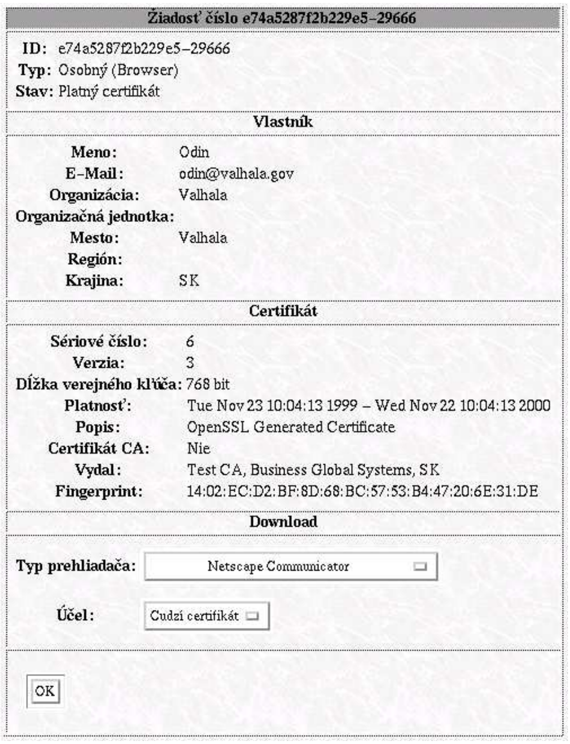
Obrázok 29: Vyhľadávanie žiadosti o certifikát alebo certifikátu

sériového čísla certifikátu (ak bol vydaný), mena majiteľa, elektronickej adresy majiteľa alebo stavu certifikátu (ak to dovolia prístupové práva). Na vyhľadávanie je nutné zadať presné údaje, tak ako boli zadané v žiadosti. Toto obmedzenie je realizované za účelom čo možno najviac obmedziť možnosť zisťovania stavu cudzích certifikátov bez znalosti aspoň elektronickej adresy majiteľa certifikátu.

Nájdené certifikáty sú zobrazené pod vyhľadávacím dialógom. Želanú akciu je možné vykonať kliknutím na číslo žiadosti. Do vyhľadávacieho dialógu je možné kedykoľvek sa dostať pomocou navigačného tlačidla "Hľadaj".

G.6 Prebrať certifikát a nainštalovať certifikát CA

Dialógy "prebrať certifikát" a "nainštalovať certifikát CA" majú podobný vzhľad, preto budú popísané spolu. Dialóg "prebrať certifikát" slúži na prebratie podpísaného certifikátu jeho žiadateľom. Po úspešnom vybraní certifikátu pomocou vyhľadávacieho dialógu (kapitola G.5) je zobrazený dialóg v ktorom môžete nájsť údaje o certifikáte a jeho žiadateľovi (obrázok 30).



Žiadosť číslo e74a5287f2b229e5-29666

ID: e74a5287f2b229e5-29666
Typ: Osobný (Browser)
Stav: Platný certifikát

Vlastník

Meno: Odin
E-Mail: odin@valhala.gov
Organizácia: Valhala
Organizačná jednotka:
Mesto: Valhala
Región:
Krajina: SK

Certifikát

Sériové číslo: 6
Verzia: 3
Dĺžka verejného kľúča: 768 bit
Platnosť: Tue Nov 23 10:04:13 1999 – Wed Nov 22 10:04:13 2000
Popis: OpenSSL Generated Certificate
Certifikát CA: Nie
Vydal: Test CA, Business Global Systems, SK
Fingerprint: 14:02:EC:D2:BF:8D:68:BC:57:53:B4:47:20:6E:31:DE

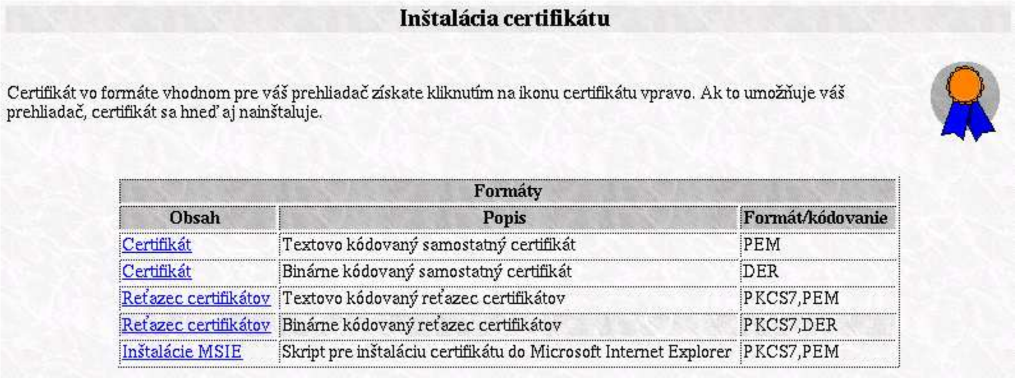
Download

Typ prehliadača: Netscape Communicator
Účel: Cudzí certifikát

Obrázok 30: Prebratie certifikátu, prvý krok

Okrem týchto údajov sa tu nachádza aj ovládací prvok na špecifikáciu typu WWW prehliadača alebo WWW servera, pre ktorý je certifikát určený. Druhý ovládací prvok špecifikuje, či ide o vlastný certifikát (preberá žiadateľ) alebo o cudzí certifikát (preberá iná osoba ako žiadateľ).

Po potvrdení prvého dialógu sa zobrazí dialóg výberu formátu (obrázok 31) . Tento dialóge je už takmer zhodný aj s dialógom "nainštalovať certifikát CA". V hornej časti je informácia o ďalšom postupe inštalácie certifikátu. Ikona certifikátu v pravo hore predstavuje linku na



Inštalácia certifikátu

Certifikát vo formáte vhodnom pre váš prehliadač získate kliknutím na ikonu certifikátu vpravo. Ak to umožňuje váš prehliadač, certifikát sa hneď aj nainštaluje.

Formáty

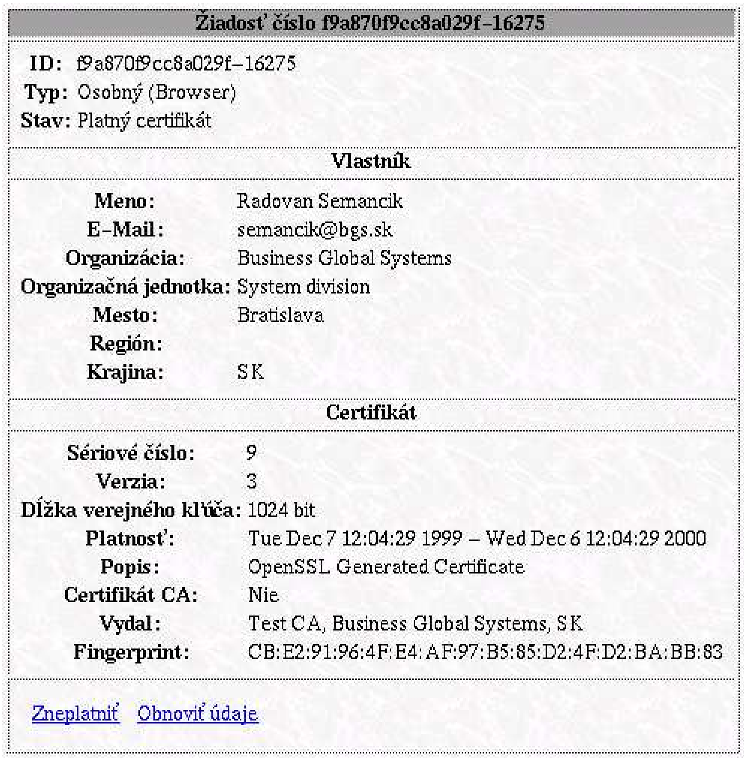
Obsah	Popis	Formát/kódovanie
Certifikát	Textovo kódovaný samostatný certifikát	PEM
Certifikát	Binárne kódovaný samostatný certifikát	DER
Reťazec certifikátov	Textovo kódovaný reťazec certifikátov	PKCS7,PEM
Reťazec certifikátov	Binárne kódovaný reťazec certifikátov	PKCS7,DER
Inštalácie MSIE	Skript pre inštaláciu certifikátu do Microsoft Internet Explorer	PKCS7,PEM

Obrázok 31: Prebratie certifikátu, druhý krok

binárne dáta certifikátu vo formáte vhodnom pre vami zvolený systém. V prípade, že by ste potrebovali certifikát v inom formáte, môžete si vhodný formát zvoliť z tabuľky v dolnej časti obrazovky.

G.7 Rozhranie operátora RA

Používateľské rozhranie pre operátora slúži na povrdzovanie, zamietanie a zneplatňovanie žiadostí o certifikát a certifikátov. Prvým krokom je výber žiadosti o certifikát pomocou vyhľadávacieho dialógu (kapitola G.5). Po zvolení relevantnej žiadosti o certifikát je táto žiadosť zobrazená aj s príslušným certifikátom. V spodnej časti dialógu sú zobrazené akcie aplikovateľné na aktuálnu žiadosť (obrázok 32). Kliknutím na príslušnú akciu sa táto akcia vykoná s



Žiadosť číslo f9a870f9cc8a029f-16275

ID: f9a870f9cc8a029f-16275
Typ: Osobný (Browser)
Stav: Platný certifikát

Vlastník

Meno: Radovan Semancik
E-Mail: semancik@bgs.sk
Organizácia: Business Global Systems
Organizačná jednotka: System division
Mesto: Bratislava
Región:
Krajina: SK

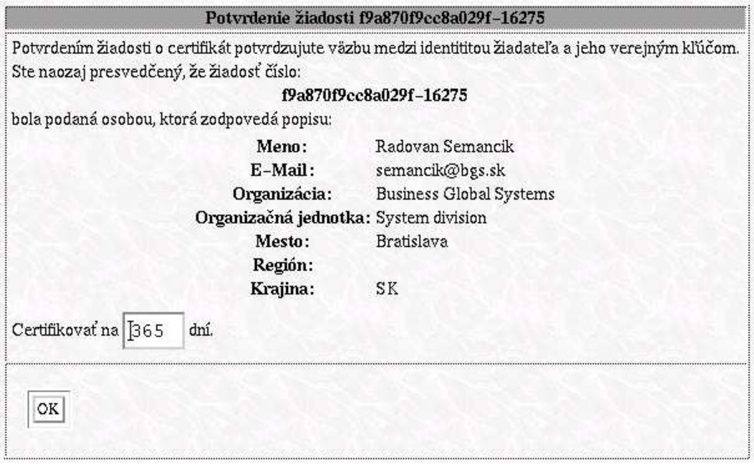
Certifikát

Sériové číslo: 9
Verzia: 3
Dĺžka verejného kľúča: 1024 bit
Platnosť: Tue Dec 7 12:04:29 1999 – Wed Dec 6 12:04:29 2000
Popis: OpenSSL Generated Certificate
Certifikát CA: Nie
Vydal: Test CA, Business Global Systems, SK
Fingerprint: CB:E2:91:96:4F:E4:AF:97:B5:85:D2:4F:D2:BA:BB:83

Obrázok 32: Rozhranie operátora registračnej autority prípadným potvrdzovacím dialógom. Niektoré akcie sú v nasledujúcich podkapitolách popísané podrobnejšie.

G.7.1 Potvrdenie žiadosti o certifikát

Potvrdením žiadosti o certifikát sa žiadosť uvedie do takého stavu, že pri najbližšom cykle certifikačnej autority bude takto označená žiadosť podpísaná a vygeneruje sa príslušný certifikát. Potvrdiť žiadosť o certifikát možno kliknutím na akciu “Potvrdiť” v dialógu operátora registračnej autority. Po výbere tejto akcie sa zobrazí dialóg s upozorňujúcim textom a ovládacími prvkami (obr. 33) . V momentálnej implementácií je jediným ovládacím prvkom vstup špecifikujúci dĺžku platnosti vytvoreného certifikátu. Pri otvorení dialógu sa v tomto vstupe objaví hodnota, ktorá bola zvolená ako štandardná v konfiguračnom dialógu (kapitola G.9). Po potrdení dialógu sa žiadosť prevedie do stavu *“potvrdená”*.


Potvrdenie žiadosti f9a870f9cc8a029f-16275

Potvrdením žiadosti o certifikát potvrdzujute väzbu medzi identititou žiadateľa a jeho verejným kľúčom.
Ste naozaj presvedčený, že žiadosť číslo:

f9a870f9cc8a029f-16275

bola podaná osobou, ktorá zodpovedá popisu:

Meno: Radovan Semancik
E-Mail: semancik@bgs.sk
Organizácia: Business Global Systems
Organizačná jednotka: System division
Mesto: Bratislava
Región:
Krajina: SK

Certifikovať na 365 dní.

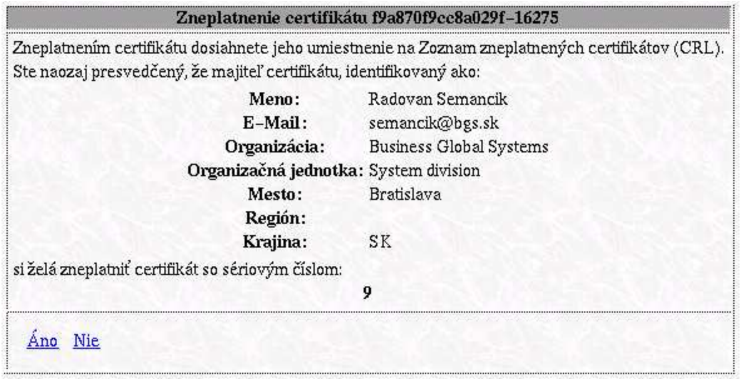
Obrázok 33: Potvrdenie žiadosti o certifikát

G.7.2 Zneplatnenie certifikátu

Zneplatnením certifikátu sa dosiahne jeho umiestnenie do zoznamu zneplatnených certifikátov certifikačnej autority. Zneplatnenie je možné realizovať pomocou akcie “Zneplatniť” v dialógu operátora registračnej autority. Po výbere tejto akcie sa zobrazí potvrdzujúci dialóg (obrázok 34), poskytujúci dodatočné informácie. Po potvrdení zneplatnenia sa certifikát dostane do takého stavu, že pri najbližšom cykle certifikačnej autority bude umiestenený do zoznamu zneplatnených certifikátov.

G.8 Rozhranie správy používateľov RA

Pomocou rozhrania správy používateľov je možné pridávať a mazať používateľov, modifikovať ich parametre a prístupové práva. Základná stránka rozhrania ponúka zoznam všetkých definovaných používateľov registračnej autority a všetky definované role (obr. 35). Kliknutím na



Zneplatnenie certifikátu f9a870f9cc8a029f-16275

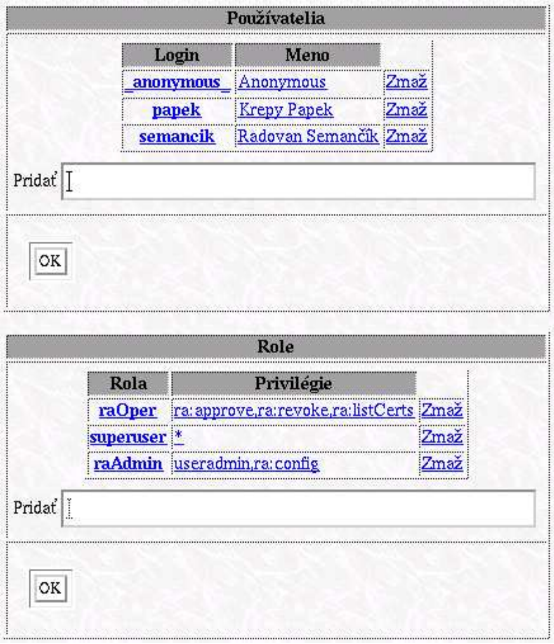
Zneplatnením certifikátu dosiahnete jeho umiestnenie na Zoznam zneplatnených certifikátov (CRL).
Ste naozaj presvedčený, že majiteľ certifikátu, identifikovaný ako:

Meno:	Radovan Semancik
E-Mail:	semancik@bgs.sk
Organizácia:	Business Global Systems
Organizačná jednotka:	System division
Mesto:	Bratislava
Región:	
Krajina:	SK

si želá zneplatniť certifikát so sériovým číslom:

9

Obrázok 34: Zneplatnenie certifikátu



Používatelia

Login	Meno	
anonymous	Anonymous	Zmaž
papek	Krepy Papek	Zmaž
semancik	Radovan Semančík	Zmaž

Role

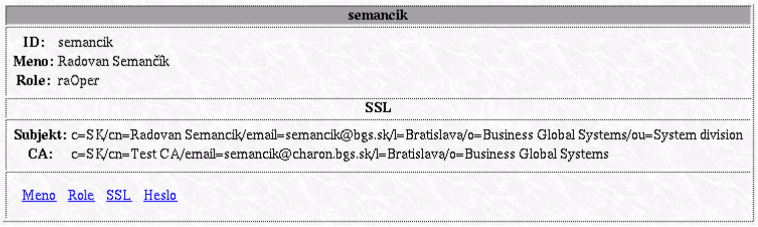
Rola	Privilégie	
raOper	ra:approve,ra:revoke,ra:listCerts	Zmaž
superuser	*	Zmaž
raAdmin	useradmin,ra:config	Zmaž

Obrázok 35: Správa používateľov

používateľa alebo rolu sa rozvinie dialóg modifikácie používateľa alebo role. Zapísaním mena používateľa alebo role do príslušného textového poľa "Pridaj" je možné vytvárať nových používateľov alebo role. Klikutím na tlačidlo "Zmaž" je možné príslušného používateľa alebo rolu zrušiť. Dialógy modifikácie používateľov a rolí budú popísané v nasledujúcich podkapitolách.

G.8.1 Modifikácia používateľov

Dialóg modifikácie používateľov je možné aktivovať výberom používateľa z hlavnej stránky alebo vytvorením nového používateľa. Dialóg zobrazuje všeobecné parametre používateľa (obrázok 36) a ponúka možnosti tieto parametre modifikovať pomocou ovládacích prvkov v spodnej časti dialógu. Funkcia týchto ovládacích prvkov je popísaná v nasledujúcom texte.


semancik

ID: semancik
Meno: Radovan Semančík
Role: raOper

SSL

Subjekt: c=SK/cn=Radovan Semancik/email=semancik@bgs.sk/l=Bratislava/o=Business Global Systems/ou=System division
CA: c=SK/cn=Test CA/email=semancik@charon.bgs.sk/l=Bratislava/o=Business Global Systems

Obrázok 36: Informácie o používateľovi

- **Meno.** Modifikácia skutočného mena používateľa.
- **Rola.** Modifikácia zoznamu rolí, ktoré používateľ má.
- **SSL.** Modifikácia osobného certifikátu používaného na prístup pomocou SSL autentifikácie. Výber certifikátu je realizovaný pomocou dialógu vyhľadávania certifikátu (kapitola G.5).
- **Heslo.** Modifikácia prístupového hesla.

G.8.2 Modifikácia rolí

Dialóg modifikácie rolí je aktivovaný výberom role z hlavného dialógu správcu používateľov alebo pridaním novej role. Dialóg modifikácie role umožňuje pridávať a uberať privilégiá. Pridanie privilégií je realizované vypísaním mena privilégie do textového poľa a stlačením tlačidla "OK", zrušenie je možné pomocou tlačidla "Odobrať" pri príslušnom privilégiu (obr. 37). Mená privilégii použitých v systéme registračnej autority sú uvedené v tabuľke 18 . Ich mená je možné špecifikovať s voliteľným prefixom *"ra:"*. V prípade špecifikácie tohto prefixu sa pridelené privilégiá budú vzťahovať len na aplikáciu registračnej autority. V opačnom prípade budú privilégiá platné aj pre iné aplikácie použité v aplikačnej kostre *Jewels*.

Rolu *"superuser"* obsahujúcu špeciálne privilégium *"*"* sa doporučuje ihneď po inštalácií a konfigurácií certifikačnej autority zrušiť, aby sa zabránilo jej zneužitiu.

raOper

Privilégium	
ra:approve	Odobrať
ra:revoke	Odobrať
ra:listCerts	Odobrať

Pridať

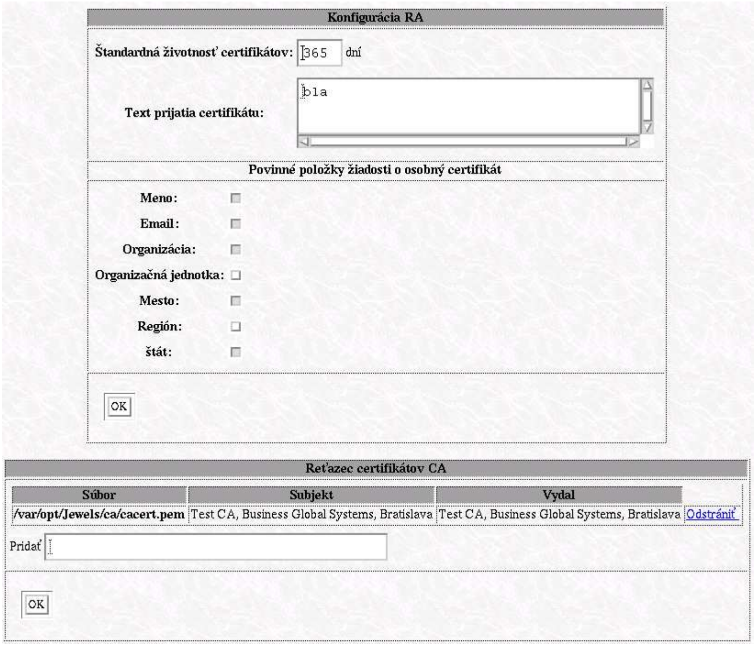
Obrázok 37: Modifikácia rolí

privilégium	poskytuje oprávnenie na:
approve	Potvrdiť alebo zamietnuť žiadosť o certifikát
revoke	Zneplatniť žiadosť o certifikát
listCerts	Umožniť zobraziť zoznam žiadostí a certifikátov
useradmin	Spravovať používateľov a ich práva

Tabuľka 18: Privilégiá použité v systéme registračnej autority

G.9 Konfigurácia parametrov RA

Pomocou konfiguračného dialógu možno meniť nastavenie všeobecných parametrov registračnej autority. Dialóg umožňuje konfigurovať tieto parametre (obr. 38):



Konfigurácia RA

Štandardná životnosť certifikátov: 365 dní

Text prijatia certifikátu: bla

Povinné položky žiadosti o osobný certifikát

Meno: ☐
Email: ☐
Organizácia: ☐
Organizačná jednotka: ☐
Mesto: ☐
Región: ☐
štát: ☐

Reťazec certifikátov CA

Súbor	Subjekt	Vydal	
/var/opt/Jewels/ca/cacert.pem	Test CA, Business Global Systems, Bratislava	Test CA, Business Global Systems, Bratislava	Odstrániť

Pridať

Obrázok 38: Konfigurácia parametrov registračnej autority

- Štandardná dĺžka platnosti certifikátu. Tento parameter definuje štandardnú životnosť certifikátov, ktorá bude ponúknutá operátorovi registračnej autority pri povrdzovaní certifikátov (kapitola G.7.1).

- Text zobrazovaný pri prijatí žiadosti o certifikát. Tento text bude zobrazovaný pri prijatí žiadosti o certifikát (obr. 26) a mal by obsahovať inštrukcie pre žiadateľa ako ďalej postupovať v certifikačnom procese.
- Povinné položky žiadosti o certifikát. Tu je možné určiť, ktoré položky žiadosti o certifikát musia byť povinne vyplnené, aby registračná autorita žiadosť prijala.
- Reťazec certifikátov certifikačnej autority. Reťazec certifikátov certifikačnej autority sa používa v prípade, že certifikačná autorita má nadradenú certifikačnú autoritu alebo je krížovo certifikovaná (*cross-certified*). Súbory, špecifikované v tomto dialógu musia byť umiestené v systéme súbrov servera registračnej autority.

H Inštalačná príručka

H.1 Predpoklady

Pred inštaláciou systému certifikačnej autority musí byť nainštalovaný nasledovný softvér:

- Oracle Relational Database Management System, verzia 8.0.x.
- programovací jazyk Perl.
- Moduly jazyka Perl: DBI, DBD::Oracle, CGI::Minimal, Apache::Session, Data::Dumper, Time-modules
- Kryptografická knižnica OpenSSL, vývojová verzia.

H.1.1 Oracle RDBMS

Nainštalujte databázový systém Oracle podľa návodu dodávanom so systémom Oracle. Predpokladaný adresár inštalácie je /opt/oracle. Predpokladaná verzia systému Oracle je 8.0.5. Po inštalácií databázového systému vytvorte databázový objekt a používateľa. Tento používateľ musí mať právo vytvoriť databázovú reláciu a pristupovať k údajom. Štandardné meno tohto používateľa je “jewels”. Pomocou priloženého SQL scriptu (`setup/ca-create-tables.sql`) vytvorte potrebné databázové tabuľky.

H.1.2 Perl a moduly

Pomocou lokálneho systému softvérových balíkov operačného systému nainštalujte programovací jazyk Perl. Podrobnejší popis inštalácie nájdete v dokumentácií k operačnému systému. V prípade, že váš operačný systém túto možnosť neposkytuje, musíte Perl nainštalovať zo zdrojovej formy. Viac detailov o tejto inštalácií sa dozviete z dokumentácie balíka Perl na inštalačnom médiu.

Nainštalujte moduly jazyka perl podľa tabuľky 19 . Všetky moduly jazyka perl sa inštalujú nasledujúcim postupom:

Modul	verzia	meno súboru
DBI	1.13	DBI-1.13.tar.gz
DBD::Oracle	1.03	CGI-Minimal-1.02.tar.gz
CGI::Minimal	1.02	CGI-Minimal-1.02.tar.gz
Apache::Session	1.03	Apache-Session-1.03.tar.gz
Data::Dumper	2.101	DBI-1.13.tar.gz
Time-modules	99.062301	Time-modules-99.062301.tar.gz
modperl	1.21	mod_perl-1.21.tar.gz

Tabuľka 19:

1. Skopírujte balík z inštalačného média do dočasného adresára.
2. Rozbaliť dodávaný balík pomocou príkazu “`tar xvf` ***`meno-súboru`***`.tar.gz`”.
3. Zmeniť aktuálny adresár do práve vytvroreného adresára balíku pomocou príkazu “`cd` ***`meno-adresára`***”.
4. Pre modul CGI::Minimal je nutné aplikovať delta súbor uvedený v prílohe F.
5. Nakonfigurovať balík pomocou príkazu “`perl Makefile.PL`”.
6. Zostaviť balík pomocou príkazu “`make`”.
7. Overiť funkčnosť balíka pomocou príkazu “`make test`”.
8. Nainštalovať balík pomocou príkazu “`make install`”. Tento krok si vyžaduje práva superpoužívateľa.
9. Teraz môžete pracovný adresár zrušiť. Súbory už nie sú naďalej potrebné.

H.1.3 Kryptografická knižnica OpenSSL

Nainštalujte kryptografickú knižnicu OpenSSL vo vývojovej verzií (*snapshot*) do adresára /opt/openssl. Môžete to dosiahnuť nasledujúcim postupom:

1. Skopírujte balík z inštalačného média do dočasného adresára.
2. Rozbaliť dodávaný balík pomocou príkazu “`tar xvf openssl-SNAP-`***`dátum`***`.tar.gz`”.
3. Zmeniť aktuálny adresár do práve vytvroreného adresára balíku pomocou príkazu “`cd openssl-SNAP-`***`dátum`***”.
4. Nakonfigurovať balík pomocou príkazu “***`./config --prefix=/opt/openssl`***”.
5. Zostaviť balík pomocou príkazu “`make`”.
6. Overiť funkčnosť balíka pomocou príkazu “`make test`”.
7. Nainštalovať balík pomocou príkazu “make install”. Tento krok si vyžaduje práva superpoužívateľa.
8. Teraz môžete pracovný adresár zrušiť. Súbory už nie sú naďalej potrebné.
9. Pridajte adresár `/opt/openssl/bin` do vyhľadávacej cesty (PATH).

H.2 Inštalácia aplikačnej kostry Jewels a samotnej aplikácie

Pred inštaláciou samotnej aplikácie je nutné vytvoriť používateľa, ktorý bude aplikáciu vlastniť. Štandardne sa tento používateľ volá “`apache`”. Ak zmeníte meno tohto používateľa, bude nutné zmeniť vlastníka inštalovaných súborov a tiež zmeniť definíciu vlastníka WWW servera v súbore `httpd.conf`.

Aplikačná kostra Jewels je uložená na inštalačnom médiu v jednom balíku spolu s aplikáciou a WWW serverom Apache. Server Apache je dodávaný v binárnej forme, pripravený pre platformy Debian GNU/Linux 2.1 a Solaris 2.6. Aplikáciu je možné nainštalovať rozbalením balíku z inštalačného média do adresára “/opt/apache” pomocou príkazu “`tar xvfp apache-jewels-ca-`*`architektúra`*`-` V prípade, že nie je možné použiť binárnu verziu, je možné použiť vlastnú verziu WWW servera Apache, musí však obsahovať podporu pre moduly SSL a Perl.

Po inštalácií musí nasledovať konfigurácia systému editovaním súboru `/opt/apache/conf/httpd.conf`.

H.3 Vytvorenie certifikačnej autority

1. Zmeňte aktuálny adresár na “`/var/opt/Jewels/ca`”.
2. Vytvorte kľúčový pár a žiadosť o certifikát pre svoju certifikačnú autortu pomocou príkazu “`openssl req -new -out ca-req.pem` -keyout ca-privkey.pem”.
3. Získajte certifikát pre svoju certifikačnú autoritu.
    (a) Odošlite žiadosť o certifikát nadradenej certifikačnej autorite a výsledný certifikát uložte do súboru “`ca-cert.pem`”, alebo
    (b) Vytvorte vlastný certifikát (“self-signed”) pomocou príkazu “`openssl x509 -req -in ca-req.pem -out ca-cert.pem -signkey ca-privkey.pem` -days *`počet-dní`*”.
4. Uložte certifikát do súboru `/var/opt/Jewels/ca/cacert.pem` a súkromný kľúč do súboru `/var/opt/Jewels/ca/private/cakey.pem`.
5. Presvedčte sa, že súbor “`index.txt`” je prázdny, a že súbor “`serial`” obsahuje číslo 1.
6. Vaša certifikačná autorita je pripravená.

I CD-ROM Médium

adresár	súbor	obsah
document/		Tento dokument
source/		Zdrojový tvar dokumentu
	diplPracaSemancik.lyx	LyX
	diplPracaSemancik.tex	LaTeX
	diplPracaSemancik.txt	ASCII, ISO8859-2
print/		Tlačiteľný tvar dokumentu
	diplPracaSemancik.ps	PostScript
product/		Softvérový produkt a komponenty
Jewels/		Aplikačná kostra Jewels, zdrojové súbory
apache/	apache_1.3.9.tar.gz	WWW server Apache
	mod_ssl-2.4.2-1.3.9.tar.gz	Modul SSL pre Apache
bin/	apache-jewels-ca-Linux-glibc-2.0-19991209.tar.gz	Aplikácia kompilovaná pre Linux
	apache-jewels-ca-Solaris-2.6-sun4u-19991209.tar.gz	Aplikácia kompilovaná pre Solaris
ca/	ca.pl	Certifikačná autorita, zdrojový súbor
openssl/		Kryptografická knižnica OpenSSL
perl/		Programovací jazyk perl a jeho moduly
ra/	RA.pm	Registračná autorita, zdrojový súbor
setup/	ca-create-tables.sql	Skript na vytvorenie databázových tabuliek
	ca_cert.ext	Definícia X.509 rozšírení pre bežný certifikát CA