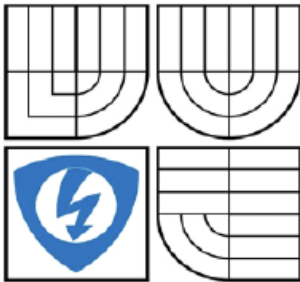


VYSOKÉ UČENÍ TECHNICKÉ  
V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A  
KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND  
COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

## REALIZACE CERTIFIKAČNÍ AUTORITY A DIGITÁLNÍHO PODPISU

IMPLEMENTATION OF CERTIFICATION AUTHORITY AND DIGITAL SIGNATURE

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

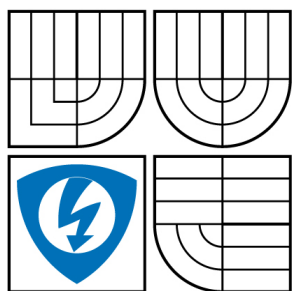
AUTOR PRÁCE  
AUTHOR

Bc. MARTIN TROJÁK

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. PETRA LAMBERTOVÁ

BRNO2008



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav telekomunikací

## Diplomová práce

magisterský navazující studijní obor  
Telekomunikační a informační technika

**Student:** Troják Martin Bc.

**ID:** 89383

**Ročník:** 2

**Akademický rok:** 2007/2008

### NÁZEV TÉMATU:

**Realizace certifikační autority a digitálního podpisu**

### POKYNY PRO VYPRACOVÁNÍ:

Nastudujte principy infrastruktury veřejných klíčů. Zaměřte se především na možnosti implementace certifikační autority a digitálního podpisu. Vytvořte aplikaci, která bude bezpečně komunikovat pomocí SSL.

### DOPORUČENÁ LITERATURA:

- [1] Elektronický podpis : Přehled právní úpravy, komentář k prováděcí vyhlášce k zákonu o elektronickém podpisu a výklad základních pojmů. Olomouc : ANAG, 2002. 141 s. ISBN 80-7263-125-X.
- [2] DOSTÁLEK, Libor, VOHNOUTOVÁ, Marta. Velký průvodce infrastrukturou PKI a technologií elektronického podpisu. 1. vyd. Brno : Computer Press, 2006. 534 s. ISBN 80-251-0828-7.
- [3] DOSTÁLEK, Libor. Velký průvodce protokoly TCP/IP : Bezpečnost. 2. aktualiz. vyd. Praha : Computer Press, 2003. xvi, 571 s. ISBN 80-7226-849-X.

**Termín zadání:** 11.2.2008

**Termín odevzdání:** 28.5.2008

**Vedoucí práce:** Ing. Petra Lambertová

**prof. Ing. Kamil Vrba, CSc.**  
*předseda oborové rady*

### UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

# LICENČNÍ SMLOUVA

## POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

### 1. Pan/paní

Jméno a příjmení: Bc. Martin Troják  
Bytem: Mezi Trhy 1/108, 74601, Opava - Město  
Narozen/a (datum a místo): 21.12.1983, Opava

(dále jen "autor")

a

### 2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií  
se sídlem Údolní 244/53, 60200 Brno 2  
jejímž jménem jedná na základě písemného pověření děkanem fakulty:  
prof. Ing. Kamil Vrba, CSc.

(dále jen "nabyvatel")

## Článek 1

### Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- disertační práce
- diplomová práce
- bakalářská práce

jiná práce, jejíž druh je specifikován jako .....

(dále jen VŠKP nebo dílo)

Název VŠKP: Realizace certifikační autority a digitálního podpisu

Vedoucí/školicitel VŠKP: Ing. Petra Lambertová

Ústav: Ústav telekomunikací

Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:

- tištěné formě - počet exemplářů 1
- elektronické formě - počet exemplářů 1

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

## Článek 2

### Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
  - ihned po uzavření této smlouvy
  - 1 rok po uzavření této smlouvy
  - 3 roky po uzavření této smlouvy
  - 5 let po uzavření této smlouvy
  - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

## Článek 3

### Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

.....

Nabyvatel

.....

Autor

## ABSTRAKT

Diplomová práce se zabývá problematikou certifikačních autorit a digitálního podpisu. Je zde rozebrána podstata digitálních certifikátů a certifikační autority. Popisuje nejpoužívanější šifrovací a hashovací metody, jež se využívají při komunikaci pomocí certifikátů a digitálního podpisu. Rozbor je zaměřen na Infrastrukturu veřejných klíčů jako normy, jejímiž pravidly se certifikační autority řídí. Je zde uvedeno, jakým způsobem se vytváří certifikační autorita a digitální certifikát. Dále je rozebrán podrobný princip digitálního podpisu. Další kapitoly se zabývají rozбором protokolu SSL, principem jeho funkce a následně jeho využitím. Dále je zde uvedena praktická část týkající se realizace certifikační autority a informačního systému. Je uveden použitý software, jeho konfigurace a následně jsou uvedeny postupy při použití aplikace a její realizace.

## KLÍČOVÁ SLOVA

PKI, infrastruktura veřejných klíčů, certifikační autorita, digitální podpis, SSL

## ABSTRACT

This master's thesis deals with problems of certification authorities and digital signature. There are analyzed principles of digital certificates and certification authorities. It describes the the most widely used cryptosystems and hash functions, which are used in communications with certificates and digital signature. Analysis is focused on Public key infrastructure standard, which describes rules of creating of certification authority and digital signature. There is also described detailed principle of digital signature. Next chapters deals with studying of protocol SSL, principles of functions and usage of SSL. Practical part of this thesis realizes certification authority and information system. There is shown used software and configuration of it. Last part describes procedures during using application and her realization.

## KEYWORDS

PKI, public key infrastructure, certification authority, digital signature, SSL, secure socket layer

### PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma Realizace certifikační autority a digitálního podpisu jsem vypracoval samostatně pod vedením vedoucí diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení §152 trestního zákona č. 140/1961 Sb.

V Brně dne .....

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucí diplomové práce Ing. Petře Lambertové, za velmi užitečnou metodickou pomoc a cenné rady při zpracování diplomové práce.

V Brně dne .....

.....

(podpis autora)

## Seznam zkratek

- 3DES (Triple DES) Triple Data Encryption Standard,
- BSD (Berkeley software distribution license) typ licence, umožňuje volné šíření licencovaného obsahu
- CA Certificate Authority, certifikační autorita
- CLR (Seznam odvolaných certifikátů)
- CÚ Certifikační úřad, jedná se o synonymum s certifikační autoritou, tento termín je využit v českém překladu operačního systému MS Windows Server 2003
- DES (Data Encryption Standard) typ symetrického blokového šifrovacího algoritmu
- DNS (Domain Name System) je hierarchický systém doménových jmen, který je realizován servery DNS a protokolem stejného jména, kterým si vyměňují informace.
- DSA (Digital Signature Algorithm) algoritmus digitálního podpisu
- DVCS (Data Validation and Certification Server) server, informující o platnosti certifikátu
- FIPS (Federal Information Processing Standards) veřejné standardy vydané vládou Spojených států amerických pro nemilitární oblast využití
- IDEA (International Data Encryption Algorithm) typ symetrického blokového šifrovacího algoritmu
- ITU-T (ITU Telecommunication Standardization Sector) sektor sdružující standardy pro telekomunikace
- IVR (Interactive voice response) interaktivní hlasová odezva, využití jako telefonní záznamník pro odvolávání certifikátů
- LDAP (Lightweight Directory Access Protocol) protokol pro ukládání a přístup k datům na serveru
- MAC (Message Authentication Code) autentifikační algoritmus založený na hashovací funkci s použitím klíče
- MD-5 (Message-Digest algorithm 5) typ hashovacího algoritmu
- MIT (Massachusetts Institute of Technology) Massachusettský technologický institut
- NAT (Network Address Translation) překlad síťových adres



- NIST (National Institute for Standards and Technology) je americká vládní instituce pro vydávání standardů a osvědčení ohledně materiálů
- NSA (The National Security Agency/Central Security Service) Národní bezpečnostní agentura/Centrální bezpečnostní služba je vládní kryptologická organizace Spojených států amerických, spadající pod ministerstvo obrany.
- OCSP (Online Certificate Status Protocol) protokol pro zjišťování stavu odvolaného digitálního certifikátu X.509
- PKI (Public Key Infrastructure) Infrastruktura veřejných klíčů
- PHP (Personal Home Page, později Hypertext preprocessor) je skriptovací programovací jazyk, určený především pro programování dynamických internetových stránek
- RA (Registrační autorita)
- RC4 (Rivest Cipher 4) typ symetrického proudového šifrovacího algoritmu
- RSA (Rivest Shamir Adleman) šifrovací algoritmus s veřejným klíčem vhodný mimo jiné pro digitální podpis
- SET (Secure Electronic Transaction) protokol pro zabezpečení transakcí přes nezabezpečené internetové sítě
- SHA-x (Secure Hash Algorithm) skupina typů hashovacích funkcí
- SSL (Secure socket layer) kryptografický protokol pro bezpečnou komunikaci na Internetu
- TLS (Transport Layer Security) nový kryptografický protokol pro bezpečnou komunikaci na Internetu, alternativa k SSL
- X.509 standard, který definuje formát a syntaxi certifikátů. Zabývá se také specifikací autentizačních služeb

# Obsah

Úvod .....	13
<b>1 Infrastruktura veřejných klíčů.....</b>	<b>14</b>
1.1 Kryptografické systémy s veřejným klíčem .....	14
1.1.1 Diffie-Hellmanova výměna klíče .....	15
1.1.2 RSA .....	16
1.1.3 El Gamal .....	17
1.1.4 DSA .....	18
1.2 Hash .....	18
1.2.1 MD2, MD4, MD5 .....	19
1.2.2 SHA (SHA-1), SHA-2 .....	19
1.2.3 Haval.....	19
<b>2 Digitální podpis .....</b>	<b>20</b>
2.1 DSA – použití při digitálním podepisování .....	21
<b>3 Digitální certifikát.....</b>	<b>23</b>
3.1 Základní struktura certifikátu.....	24
3.2 Existenční cyklus certifikátu.....	26
3.3 Certifikační autorita .....	26
3.4 Postup práce certifikační autority a příklad použití certifikátu.....	28
<b>4 Vytvoření CA a certifikátu v operačním systému Microsoft Windows Server 2003 .....</b>	<b>30</b>
4.1 Certifikační autorita .....	30
4.2 Žádost o certifikát .....	33
4.3 Vyhotovení certifikátu .....	34
4.4 Instalace certifikátu.....	34
<b>5 SSL .....</b>	<b>35</b>
5.1 Princip SSL .....	35
5.2 Handshake SSL.....	37
5.3 Change Cipher Spec Protocol a Alert Protocol .....	39

5.4 Record Protocol .....	39
<b>6 Použité nástroje pro realizaci certifikační autority a digitálního podpisu .....</b>	<b>41</b>
6.1 Konfigurace VMware Player .....	41
6.2 Instalace EasyPHP a jeho konfigurace .....	42
6.2.1 Implementace OpenSSL .....	42
6.2.2 Tvorba certifikátu serveru.....	42
6.2.3 Nastavení Apache pro podporu ModSSL .....	43
<b>7 Webové rozhraní aplikace.....</b>	<b>44</b>
7.1 Prezentace webových stránek a postup použití.....	44
7.2 Struktura webového rozhraní.....	47
7.3 Použité funkce.....	49
7.4 Zachytávání komunikace pomocí síťového analyzátoru WireShark .....	51
<b>8 Závěr .....</b>	<b>53</b>
<b>Literatura .....</b>	<b>54</b>
<b>Přílohy.....</b>	<b>56</b>
Generování certifikátu serveru.....	56

## Seznam obrázků

Obr. 1: Základní princip systémů v veřejném klíčem .....	15
Obr. 2: Základní princip hashovací funkce.....	18
Obr. 3: Princip vytvoření digitálního podpisu .....	20
Obr. 4: Princip ověření digitálního podpisu.....	21
Obr. 5: Schéma certifikační autority - instituce .....	26
Obr. 6: Činnost certifikační autority a šifrovaná komunikace.....	29
Obr. 7: Instalace certifikační autority, 1. část .....	30
Obr. 8: Instalace certifikační autority, 2. část .....	31
Obr. 9: Instalace certifikační autority, 3. část .....	32
Obr.10: Instalace certifikační autority, 4. část .....	32
Obr.11: Žádost o certifikát .....	33
Obr.12: Digitální certifikát .....	34
Obr.13: Umístění SSL v TCP/IP modelu.....	35
Obr.14: Sestava protokolu SSL .....	36
Obr.15: Struktura protokolu SSL Handshake Protocol .....	37
Obr.16: Komunikace mezi klientem a serverem pomocí SSL Handshake Protocol .....	38
Obr.17: Vznik datové části rekordu .....	40
Obr.18: Formulář generátoru certifikátů .....	44
Obr.19: Vyhotovení certifikátu certifikační autoritou .....	45
Obr.20: Přihlašovací formulář .....	46
Obr.21: Osobní data po přihlášení .....	46
Obr.22: Struktura webového rozhraní .....	47
Obr.23: Tabulka „certifikáty“ .....	48
Obr.24: Tabulka „připojení“ .....	49
Obr.25: Zachytávání testovacích šifrovaných aplikačních dat .....	51
Obr.26: Zachycení šifrovaných dat při přihlašování do informačního systému.....	52

## Seznam tabulek

Tab. 1: Porovnání položek certifikátu a občanského průkazu .....	23
Tab. 2: Přehled relativních jedinečných jmen .....	25
Tab. 3: Přehled a význam jednotlivých souborů ve webovém rozhraní .....	48

## Úvod

V současnosti je Internet využíván pro všestrannou komunikaci. Uživatelé přes Internet přenášejí informace menší či větší důležitosti. Právě v případě vysoké důležitosti je potřeba, aby mezi nimi existovala důvěra. Jedním ze způsobů, jak důvěru zaručit, je jednoznačná vzájemná identifikace a autentizace obou uživatelů pomocí certifikátu. Certifikát obsahuje mnoho důvěrných informací a jedinečný klíč, s jehož pomocí je možno datovou komunikaci šifrovat. Vydáváním digitálních certifikátů se zabývá certifikační autorita.

Dalším problémem je zabezpečení nedotknutelnosti přenášených zpráv, tedy ujištění příjemce zprávy o tom, že zpráva opravdu pochází od odesílatele a nebyla při přenosu sítí modifikována. Tímto se zabývá digitální podpis.

Pro aplikaci certifikátů a digitálního podpisu vznikl souhrn norem nazvaný Infrastruktura veřejných klíčů.

Diplomová práce se zabývá problematikou certifikačních autorit a digitálního podpisu. Je zde rozebrána podstata digitálních certifikátů a certifikační autority. Popisuje využívané šifrovací a hashovací metody, jež se využívají při komunikaci pomocí certifikátů a digitálního podpisu. Rozbor je zaměřen na Infrastrukturu veřejných klíčů jako normy, jejímiž pravidly se certifikační autority řídí. Je zde uvedeno, jakým způsobem se vytváří certifikační autorita a digitální certifikát. Dále je zde rozebrán podrobný princip digitálního podpisu. Další kapitoly se zabývají rozбором protokolu SSL, principem jeho funkce a následně jeho využitím. Dále je zde uvedena praktická část týkající se realizace certifikační autority a informačního systému. Je zde uveden použitý software a jeho konfigurace a následně jsou uvedeny postupy při použití aplikace a její realizace. Cílem diplomové práce je zrealizování certifikační autority a vytvoření přístupu do informačního systému s využitím zabezpečovacího protokolu SSL.

# 1 Infrastruktura veřejných klíčů

Pojem infrastruktura veřejných klíčů (PKI, Public Key Infrastructure) vyjadřuje komplexní systém, který svým uživatelům poskytuje služby v oblasti šifrování pomocí systémů s veřejným klíčem a služby spojené s digitálními podpisy. Účelem infrastruktury veřejných klíčů je pak hlavně správa jednotlivých soukromých klíčů a certifikátů.

PKI byla založena na základě toho, že v minulosti vzniklo několik norem, které využívaly kryptografii s veřejnými klíči. Zmíněné normy neměly přílišnou vzájemnou návaznost, ale také neřešily veškerou problematiku, u které bylo třeba mít vše zabezpečeno. Vývojářské společnosti vytvářely software, jež se začal zabývat vzájemnou spoluprací s ostatním softwarem, ale komunikace neprobíhala vždy podle představ. Vznikla proto pracovní skupina, jež se pokusila vytvořit soustavu norem, které by překonaly výše uvedené problémy. Díky tomu vzniká nová generace norem, která není příliš ideální, ale je velkým krokem dopředu. Systém těchto protokolů je vyjádřen pojmem PKI.

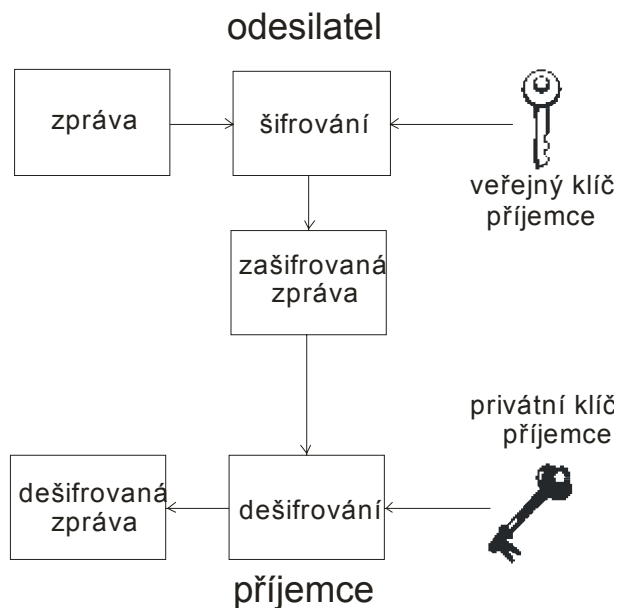
Normy PKI vycházejí z norem ITU-T řady X.500 RFC-3280, popis certifikátu přesněji z normy X.509 (viz Seznam zkratk). Ne všechny parametry jsou ovšem přímo převzaty z těchto norem, většina praktik je upravena pro určení na Internet. Tímto dochází k tomu, že ne všechna rozšíření certifikátů popsána v normě ITU-T X.509 jsou normami PKI podporována.

Alternativou k PKI pro využití na Internetu je např. systém SET, jež je určen pro platby platebními kartami přes Internet. SET používá rovněž certifikáty dle normy X.509, ovšem nikoli podle RFC-3280. Vzájemná kompatibilita tudíž mezi technologiemi není. [1]

## 1.1 Kryptografické systémy s veřejným klíčem

Kryptografické systémy s veřejným klíčem využívají, na rozdíl od systémů s privátním klíčem, pro zašifrování zprávy tzv. *veřejný klíč* a pro dešifrování *klíč privátní (tajný)*. Veřejný klíč je založen na principu toho, že je možné jej klidně zveřejnit, a přesto nedojde k porušení bezpečnosti zprávy nebo dešifrovacího klíče. Velice často jsou tyto systémy označovány také jako systémy s asymetrickým klíčem. Podstatný základ asymetrického šifrování tvoří vlastnost, že jestliže je něco zašifrováno jedním z výše uvedených klíčů, pak rozšifrování je možno pouze druhým klíčem z dané dvojice klíčů. Jestliže tedy zpráva zašifruji „mým“ privátním klíčem, pak dešifrovat tuto zprávu může jen někdo, kdo má „můj“ veřejný klíč. Analogicky, jestliže pro zašifrování zprávy použiji „Váš“ veřejný klíč, pak pro dešifrování je nyní potřeba právě „Váš“ privátní klíč. Zde je zřejmé, že privátní a veřejný klíč tvoří vždy pevně spjatou dvojici. Z toho vyplývá, že je možno jeden jednoznačně vypočítat z hodnoty druhého. Zde ale platí pravidlo: získat hodnotu veřejného klíče z klíče privátního je možno poměrně snadno a rychle, ale naopak je to prakticky nemožné. Problém ovšem nespočívá v tom, že by nebylo známo, jak hodnotu klíče spočítat, ale důvodem je velká náročnost výpočetního výkonu počítače. Výpočet hodnoty by takto mohl trvat i několik desítek či stovek let.

Z výše uvedeného vyplývá, že privátní klíč je nutno velice hlídat a udržovat v tajnosti, naopak klíč veřejný je určen k tomu, aby byl zveřejněn. Zveřejnění tohoto klíče je nutné, aby každý mohl rozluštit zprávy zašifrované privátním klíčem a zjistit tak, kdo je skutečným autorem. Tento postup zašifrování dokumentu privátním klíčem a poté dešifrování veřejným klíčem pro zjištění toho, kdo dokument svým privátním klíčem zašifroval, se označuje jako digitální podpis. Princip digitálního podpisu bude popsán dalších kapitolách.



Obr. 1: Základní princip systémů v veřejném klíčem

### 1.1.1 Diffie-Hellmanova výměna klíče

Systém pro výměnu kryptografických klíčů mezi dvěma stranami. V podstatě nejde o šifrovací algoritmus, ale jedná se o metodiku pro vytvoření a výměnu sdíleného privátního klíče přes veřejné komunikační kanály.

Dva uživatelé Petr a Pavel se dohodnou na 2 číslech,  $p$  a  $g$ ; nejedná se o tajná čísla, proto je možno je vyměnit bez potřeby zabezpečení. Uživatel Petr si zvolí tajné číslo  $a$ , uživatel Pavel číslo  $b$ . Uživatel Petr nyní provede výpočet  $A$  dle rovnice

$$A = g^a \bmod p, \quad (1.1.1.1)$$

uživatel Pavel poté

$$B = g^b \bmod p. \quad (1.1.1.2)$$

Obě vypočtená čísla  $A$  a  $B$  si nyní vymění opět nezabezpečeným kanálem. Petr nyní vypočte  $K$  z rovnice

$$K = B^a \bmod p, \quad (1.1.1.3)$$

Pavel vypočte své  $K$

$$K = A^b \bmod p. \quad (1.1.1.4)$$

Oba uživatelé tímto získají společný výsledek  $K$ , který se označí jako tajný klíč. Tento tajný klíč se nyní použije pro zašifrování a dešifrování zprávy symetrickou šifrou. Na náhodně zvolená čísla jsou ovšem kladeny požadavky a to takové, že  $p$  musí být prvočíslo určující modul systému a  $g$  ležící ve zvoleném modulu. Platí tedy:  $0 < g \leq p-1$ . Tato podmínka vychází z rovnice

$$g^{ab} = g^{ba}. \quad (1.1.1.5)$$

Jestliže nyní dojde k tomu, že budou odposlechnuty informace přenášené pro vytvoření privátního klíče, tento klíč nebude možno odvodit. [3]

### 1.1.2 RSA

Kryptografický systém s veřejným klíčem, vyvinutý profesory MIT Ronaldem Rivestem, Adi Shamirem a Leopoldem Adlemanem. RSA je možno použít jednak jako šifrovací algoritmus a také jako základ pro systém digitálních podpisů. Dle použité implementace může klíč mít libovolnou délku. Využívá tzv. faktorizace, která je založena na rozkladu čísla na součin mocnin prvočísel.

*Základ:*

Zvolí se 2 velká prvočísla  $p$  a  $q$ . Vypočtou se hodnoty čísel

$$n = (p \cdot q), \quad (1.1.2.1)$$

$$r = (p - 1) \cdot (q - 1). \quad (1.1.2.2)$$

Zvolí se hodnota veřejného klíče  $VK$ , pokud možno co největší, tato hodnota musí být nesoudělná s hodnotou čísla  $r$ . Vypočte se tajný šifrovací klíč  $TK$  dle rovnice

$$TK = VK^{-1} \text{ mod } r. \quad (1.1.2.3)$$

*Šifrování:*

Zpráva, která bude šifrována ( $Z$ ), se rozdělí na bloky symbolů, které mají stejnou délku, přičemž každý  $i$ -tý blok zprávy bude brán jako číslo  $Z_i$ . Zde platí podmínka, že  $Z_i$  musí být menší než  $n$ . Nyní se každý  $i$ -tý blok zprávy postupně zašifruje dle výpočtu

$$K_i = Z_i^{VK} \text{ mod } n. \quad (1.1.2.4)$$

Bloky  $K_i$  se zřetězí a vytvoří se kryptogram  $K$ , který se odesílá příjemci

*Dešifrování:*

Přijatý kryptogram  $K$  se rozdělí opět na bloky  $K_i$ , které byly vytvořeny při šifrování. Jednotlivé  $i$ -té bloky kryptogramu  $K$  se dešifrují dle rovnice

$$Z_i = K_i^{TK} \text{ mod } n. \quad (1.1.2.5)$$

Zřetěžením všech získaných bloků  $Z_i$  se sestaví dešifrovaná zpráva.

Při použití algoritmu RSA ve skutečných aplikacích se pracuje s čísly, která obsahují stovky řádů. Početní operace s takto velkými čísly jsou pochopitelně velice náročné na výpočetní výkon, proto se v aplikacích minimalizuje počet operací RSA, které se budou provádět. Provádí se to tak, že se pomocí RSA nešifruje celá zpráva, ale pouze privátní (symetrický) klíč a tímto klíčem je pak zašifrována zpráva pomocí symetrické šifry DES (Data Encryption Standard) nebo IDEA (International Data Encryption Algorithm). [3]



### 1.1.3 El Gamal

Algoritmus založený roku 1985, jehož podstatou je, podobně jako u RSA, faktorizace. Podobně jako RSA se používá k šifrování a digitálnímu podpisu.

*Základ:*

Zvolení velkého prvočísla  $p$ , k němuž se nalezne primitivní kořen  $g$  (není podmínkou, že dvojice  $p$  a  $g$  bude jedinečná). Zvolení privátního klíče  $PK$  pro nějž platí, že musí být menší než  $(p-1)$ . Získání hodnoty veřejného klíče  $VK$  podle rovnice

$$VK = g^{PK} \bmod p \quad (1.1.3.1)$$

Zveřejnění námi zvolených prvočísel  $p$  a  $g$  a vypočteného veřejného klíče  $VK$ , privátní klíč se nesmí zveřejnit, bude jej znát pouze uživatel, kterému je šifrovaná zpráva adresována.

*Šifrování:*

Pro šifrování každé zprávy  $Z$  musí být vybráno náhodné číslo  $m$ , jež musí být menší než  $(p-1)$ . Získání čísla  $A$  pomocí rovnice

$$A = g^m \bmod p \quad (1.1.3.2)$$

Podobně jako v RSA se zpráva  $Z$  rozdělí na  $j$  bloků, které mají stejnou délku. Získají se tím čísla  $Z_j$ , přičemž každé toto číslo odpovídá  $j$ -tému bloku zprávy. Jednotlivé bloky  $Z_j$  se zašifrují pomocí rovnice

$$K_j = (VK^m \cdot Z_j) \bmod p. \quad (1.1.3.3)$$

Všechny bloky  $K_j$  se zřetězí, čímž se získá kryptogram  $K$ . Tento kryptogram se odesílá adresátovi spolu s číslem  $A$

*Dešifrování:*

Přijatý kryptogram  $K$  si adresát rozdělí opět na bloky  $K_j$  a pomocí přijatého čísla  $A$  si vypočte hodnotu čísla  $B$  pomocí rovnice

$$B = A^{PK} \bmod p. \quad (1.1.3.4)$$

Následně k vypočtenému číslu si vypočte inverzní hodnotu, tedy  $D^{-1}$ . Všechny  $j$ -té části kryptogramu se nyní dešifrují dle rovnice

$$Z_j = (K_j \cdot D^{-1}) \bmod p. \quad (1.1.3.5)$$

Všechny dešifrované bloky se pak sloučí a získá se tím dešifrovaná zpráva. [3]

### 1.1.4 DSA

Asymetrický kryptosystém DSA (Digital Signature Algorithm) byl vyvinut agenturou NSA od níž tento kryptosystém převzal institut NIST a použil jej jako federální standard pro zpracování informací (FIPS). V algoritmu DSA je možno použít klíče různých délek, ale dle standardu FIPS se používá délka klíče pouze 512 a 1024 bitů. Tento kryptosystém je používán pouze pro digitální podpis, jeho algoritmus bude popsán v části Digitální podpis.

## 1.2 Hash

Význam slova hash se dá vyjádřit více českými výrazy. Může být brán jako kontrolní součet, miniatura, ale nejlépe vypovídající termín je slovo výtah.

Hashovací funkce je jednocestná funkce, která vytváří z libovolně dlouhé zprávy krátký řetězec o konstantní délce. Hash se dá vyjádřit jako algoritmus, který je schopen výpočetně snadno získat výsledek. Ovšem zpětně dospět z hodnoty hash k původní zprávě je technicky velmi náročné či v určitých případech téměř nemožné. Řetězec získaný ve výsledku výpočtu hash musí v co největší míře ukázat charakter původní zprávy.

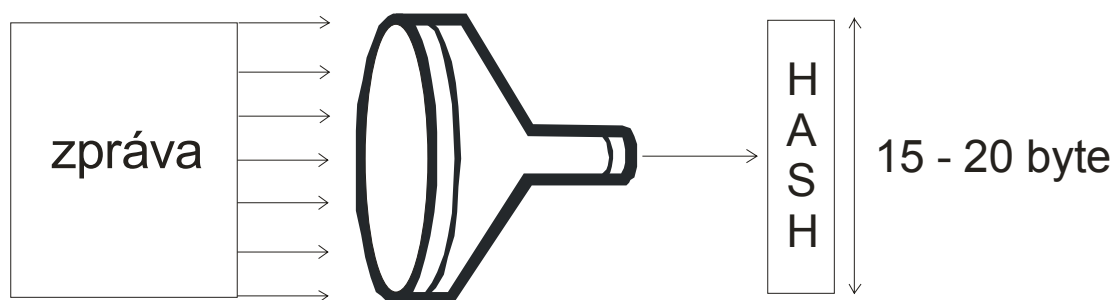
Výsledný hash mívá nejčastěji velikosti

- a) 16 bajtů v případě algoritmu MD-5
- b) 20 bajtů v případě algoritmu SHA-1

Významnou vlastností hashovací funkce je, že je deterministická, což má za následek, že jestliže opakovaně vytváříme hash pro určitou zprávu, tak nám pokaždé vznikne stejný hash.

Další vlastností algoritmu by měla být nesnadná nebo nemožná odvoditelnost či invertovatelnost. To znamená, že jestliže je známá výstupní hodnota hashovací funkce, pak nesmí být možné sestavit původní text, ať už reverzací hashovací funkce nebo zjišťováním závislosti mezi vstupem a výstupem. Máme-li hodnotu hash o velikosti 128bitů, pak rozhodně není možné získat původní zprávu pomocí hrubé síly, protože celkový počet možností je asi  $1,7 \cdot 10^{38}$  zpráv, které je nutno vyzkoušet, abychom získali požadovaný hash. Počet těchto možností je velký tudíž je velmi nepravděpodobné, že by během několika tisíců let mohly vzniknout dvě zprávy, které by měly stejný výstupní hash.

Poslední významnou vlastností hashovací funkce je ta, že změníme-li původní zprávu, pak ve výsledném hash se nám tato změna projeví ve velké míře. Změníme-li přesněji pouze jeden bit, pak změna se projeví v polovině bitů hash.



Obr. 2: Základní princip hashovací funkce

Pro vytvoření hash existuje velké množství algoritmů. Většina z nich funguje na podobném principu, jediné rozdíly jsou tvořeny rychlostí jejich provádění a drobnými odlišnostmi ve vlastnostech. Zde jsou uvedeny ty nejpoužívanější algoritmy:

### 1.2.1 MD2, MD4, MD5

Hashovací funkce MD2 byla navržena roku 1989. Autorem je Ronald Rivest. V květnu roku 1992 byla uznána jako standard RFC 1319. MD2 pracuje na principu toho, že zpracovává datové bloky o délce 128 bitů a na konci vytvoří výsledný hash dlouhý 128 bitů. Tato hashovací funkce neměla mnoho výrazných slabín, ale byla velice pomalá. Ronald Rivest proto vyvinul novou hashovací funkci MD4, jež vycházela z MD2. Algoritmus MD4 byl certifikován jako RFC 1186 a 1320. Tento algoritmus byl velice rychlý a kompaktní. Po zveřejnění několika možností útoku kryptografickými útočníky, vyvinul Ronald Rivest v současnosti poslední verzi MD5. Tato verze algoritmu je certifikována jako RFC1321 a obsahuje nové množství přidávaných operací. Je tedy mírně pomalejší než MD4. Všechny tři verze pracují na obdobném postupu při zpracování dat, mají tedy podobné slabiny. Algoritmus MD5 je distribuován společností *RSA Data Security* jako volně šiřitelný algoritmus pro digitální podpis a mnoho certifikačních autorit nabízí možnost jeho použití, není ovšem příliš doporučován. [5]

### 1.2.2 SHA (SHA-1), SHA-2

SHA je zkratka názvu hashovacího algoritmu *Secure Hash Algorithm*. Vyvinula jej instituce NIST ve spolupráci s NSA. SHA využívá velice podobný algoritmus jako MD4, ovšem výstupem je řetězec o délce 160 bitů. V únoru 2005 byl zveřejněn objev algoritmu, který umožňuje nalézt kolizi podstatně rychleji než hrubou silou. Výpočetní náročnost je ale stále mimo současnou techniku.

Standard SHA-2 sdružuje hashovací funkce SHA-256, SHA-512, SHA-384 a SHA-224. Jednotlivé funkce produkují hash různé délky. Číslo v názvu označuje výslednou bitovou délku hash. Jedná se o dlouhé řetězce, proto žádný z výše uvedených algoritmů nebyl prolomen. Útok hrubou silou v případě SHA-2 je nepoužitelný. [6]

### 1.2.3 Haval

Základem pro vznik algoritmu Haval byl algoritmus MD5. Jeho modifikaci provedli Yuliang Zheng, Josef Pieprzyk a Jennifer Seberry. Významnou vlastností algoritmu Haval je ta, že je možno jej modifikovat tak, aby vytvářel výstupní hash určitého souboru v délce od 92 do 256 bitů. Je možno také ovlivnit počet průchodů algoritmem, neboli rund. Tato vlastnost má pak za následek zvýšení rychlosti oproti MD5, ovšem za podmínky určitého snížení bezpečnosti řetězce hash. Jestliže se ovšem zvýší délka výsledného hashe, pak Haval může ve výsledku vytvářet mnohem bezpečnější kód než MD5. V praxi se tento hashovací algoritmus příliš nepoužívá. [5]

## 2 Digitální podpis

Digitální podpis je jednou z nejznámějších technik autentizace. Ověřuje, zda zpráva, která byla odeslána od odesílatele k příjemci, byla opravdu vytvořena odesílatelem, jež je vlastník použitého digitálního podpisu a že při přenosu Internetem nebyla určitým způsobem upravena či změněna útočníkem. Pro digitální podpis se využívá systémů s veřejným klíčem, tedy asymetrického šifrování. Přestože se používá šifrování, tak úkolem podpisu není žádné zabezpečení zprávy, pouze zajištění autentičnosti původu.

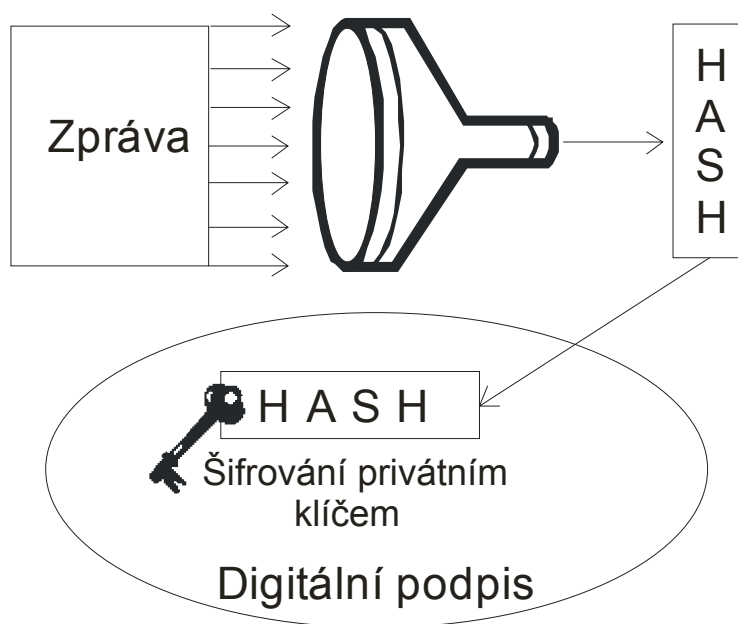
*Vytvoření digitálního podpisu:*

- 1.krok: výpočet kontrolního součtu (hash) ze zprávy
- 2.krok: šifrování výsledného kontrolního součtu pomocí privátního klíče uživatele, který podpis vytváří (viz Obr.3).

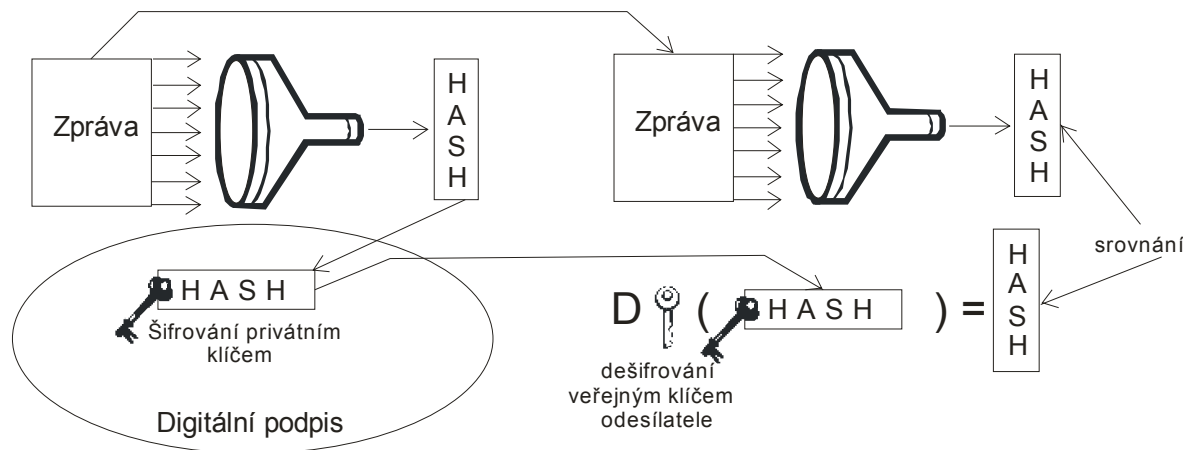
Nyní odesílatel zprávu spolu s digitálním podpisem odešle příjemci a ten provede následující postup.

*Ověření pravosti digitálního podpisu:*

- 1.krok: příjemce si vypočte hash z přijaté zprávy
- 2.krok: příjemce pomocí veřejného klíče odesílatele dešifruje přijatý digitální podpis
- 3.krok: příjemce srovnává dešifrovaný digitální podpis, tedy hash, se svým vlastnoručně spočteným hash z přijaté zprávy. Jestliže hodnoty obou hash jsou shodné, pak je potvrzena autentizace a je jasné, že elektronický podpis mohl vytvořit pouze odesílatel, jež jediný vlastní svůj privátní klíč (viz Obr.4). [5]



Obr. 3: Princip vytvoření digitálního podpisu



Obr. 4: Princip ověření digitálního podpisu

Privátní klíč pro použití v digitálním podpisu je nutné bezpečně uložit. Mezi šifrováním a digitálním podpisem existuje podstatný rozdíl. Pro vytvoření digitálního podpisu použije odesílatel svůj privátní klíč. Pro jeho ověření se využije klíč veřejný. V případě šifrování je zpráva zašifrována pomocí veřejného klíče a následně dešifrována klíčem privátním. Operace podepisování zprávy odpovídá operaci dešifrování zašifrované zprávy a naopak.

Podepisování digitálním podpisem je možno použít přímo na zprávě, tedy tím, že se zašifruje privátním klíčem. To se prakticky neprovádí, podepisuje se pouze na hash. Pokud by se zpráva celá musela zašifrovat, pak by tato operace probíhala dlouhou dobu, protože každý algoritmus s veřejným klíčem probíhá velice dlouho. Pro podepsání zprávy o velikosti řádově megabajtů by šifrování probíhalo několik hodin nebo dokonce dní.

Pro vytváření digitálního podpisu se v současnosti využívají kombinace hashovací funkce MD5 a systému s veřejným klíčem RSA. Jinou možností je využití algoritmu pro získání hash SHA-1 a ElGamalova systému s veřejným klíčem. Složením těchto algoritmů vznikl algoritmus pro digitální podpis DSA. [3]

## 2.1 DSA – použití při digitálním podepisování

Při vytváření digitálního podpisu pomocí algoritmu DSA je podobně jako u ostatních systémů s veřejným klíčem potřeba definovat určité parametry. Prvním parametrem, který se zvolí je  $p$ . Musí se jednat o prvočíslo o délce od 512 do 1024 bitů. Dále se zvolí 160 bitů dlouhý parametr  $q$ , jež je prvočíselným faktorem z  $p-1$ . Třetím parametrem je  $g$ , jež má hodnotu dle vzorce

$$g = \tilde{t} \bmod p. \quad (2.1.1)$$

Hodnotu  $z$  lze určit ze vzorce  $z = (p-1)/q$ , přičemž hodnota  $t$  musí být menší než  $q$ . Dále musí platit, že  $\tilde{t} \bmod p$  musí mít vyšší hodnotu než 1. Zvolí se tajný klíč  $x$ , jež musí mít vyšší hodnotu než parametr  $q$ . Veřejný klíč se určí pomocí vzorce

$$y = g^x \bmod p. \quad (2.1.2)$$

Podpis probíhá dle následujícího postupu. Program vygeneruje pro každou podepisovanou zprávu jiný náhodně zvolený parametr  $k$ , jež je menší než  $q$ . Vypočte  $r$  dle vzorce

$$r = (g^k \bmod p) \bmod q. \quad (2.1.3)$$

Určí se hash  $h$  podepisované zprávy. Vypočítá se číslo pomocí vzorce

$$s = [k^{-1} \cdot (h + x \cdot r)] \bmod q. \quad (2.1.4)$$

Vzniká dvojice čísel  $(r,s)$ , jež tvoří digitální podpis zprávy.

Po přenosu podepsané zprávy zkontroluje příjemce digitální podpis následujícím postupem. Vypočte hodnotu

$$w = s^{-1} \bmod q. \quad (2.1.5)$$

Pomocí vypočteného  $w$  se pak zjistí hodnoty čísel  $i$  a  $j$  podle následujících vzorců:

$$i = (h \cdot w) \bmod q, \quad (2.1.6)$$

$$j = (r \cdot w) \bmod q. \quad (2.1.7)$$

Posledním výpočtem je zjištění hodnoty  $v$  dle vzorce

$$v = [(g^i \cdot y^j) \bmod p] \bmod q. \quad (2.1.8)$$

Pokud jsou hodnoty  $v$  a  $r$  shodné, pak je podpis autentický. [3]

### 3 Digitální certifikát

Digitální certifikát je soubor dat ve stanoveném formátu, který identifikuje osobu nebo server (elektronický obchod, server s internetovým bankovníctvím, poštovní server, VPN server...). Může během elektronické komunikace mezi dvěma subjekty (osoba/osoba, osoba/server, server/server) zajistit šifrování přenášených dat, ověření jedné a/nebo druhé strany, rozpoznání neoprávněné modifikace dat a s tím související digitální podpis.

Celý systém digitálních certifikátů je postaven na matematické podstatě, ze které vycházejí algoritmy používané k šifrování. K tomu je nutné, aby v počítači uživatele (nebo na serveru) existoval tzv. privátní klíč. Tento soukromý klíč vygeneruje uživatel ve svém počítači (resp. administrátor na serveru). Jelikož se z hlediska bezpečnosti jedná o velice kritickou součást, má uživatel za povinnost chránit soukromý klíč na bezpečném úložišti, zabránit jeho ztrátě a zároveň jeho zneužití jinými subjekty. Pokud to neomezují organizační nebo technické požadavky, měl by být chráněn heslem, které je známé pouze uživateli. K privátnímu klíči má vygenerován také klíč veřejný, který ovšem není tajný. [3]

Certifikát je možno přirovnat k občanskému průkazu či pasu. Zatímco občanský průkaz se vydává v tištěné podobě, certifikát je digitálně podepsanou datovou strukturou, jejíž základní součástí je veřejný klíč držitele certifikátu. V Tab.1 je uvedeno srovnání položek občanského průkazu a certifikátu. [2]

Tab. 1: Porovnání položek certifikátu a občanského průkazu

Certifikát	Občanský průkaz
Verze 0 ... X.509 verze 1 (1988) 1 ... X.509 verze 2 2 ... X.509 verze 3	Verze (federální, červený český, karta,..)
Sériové číslo	Číslo a série občanského průkazu
Algoritmus použitý pro podpis	Razítko či samolepka přes fotografii
Vydal (identifikace certifikační autority dle X.509)	Vydal
Platnost od-do	Platnost
Jméno a adresa (identifikace vlastníka)	Jméno a adresa
Veřejný klíč	-
Rozšíření certifikátu	Další údaje
Digitální podpis certifikátu	Vlastní razítko, samolepka přes fotografii

Teoreticky by pro autorizaci člověka, tedy pro ověření práv daného klienta, a autentizaci, tedy pro jednoznačné ověření subjektu (člověka) mohl existovat jediný typ certifikátu. To ovšem v praxi není možné, výsledkem jsou tři typy certifikátů:

- a) Pro elektronický podpis
- b) Pro autentizaci
- c) Pro šifrování

### 3.1 Základní struktura certifikátu

Nyní zde budou rozebrány jednotlivé položky certifikátu a jeho struktura.

#### *Verze certifikátu (Version)*

Verze certifikátu označuje, z jaké verze normy X.509 je certifikát odvozen, zda verze 1, 2 či 3. Pokud je zde využita verze 1, pak má položka hodnotu 0, pokud verze 2, pak hodnotu 1. V případě použití verze 3, která je také v současné době nejčastější, je nastavena hodnota 2.

#### *Pořadové číslo certifikátu*

Pořadové číslo certifikátu (Serial number) má formát celého kladného čísla. Podmínkou je, aby bylo jednoznačné a nezaměnitelné v rámci dané certifikační autority. Nesmí tedy dojít k tomu, že by certifikační autorita vytvořila dva certifikáty, jež by měly stejné pořadové číslo.

#### *Algoritmus podpisu*

Tato položka certifikátu (Signature Algorithm) uvádí algoritmy, jež použila certifikační autorita při vytváření digitálního podpisu certifikátu. Je v ní uvedena přesná dvojice použitých algoritmů: první pro výpočet hash, druhý je systém veřejných klíčů, jímž je hash zašifrován.

#### *Platnost*

Položka platnost popisuje časové omezení platnosti certifikátu „od“ (Not Before) – „do“ (Not After).

Důvody proč má certifikát omezenou dobu platnosti:

- a) Organizační: aplikace má omezenou dobu platnosti
- b) Bezpečnostní: ke zvýšení bezpečnosti

Doba platnosti musí být mnohem kratší než je doba potřebná k prolomení šifry certifikovaného veřejného klíče. Tato krátká životnost bývá velkým problémem u certifikátů certifikačních autorit, které by měly být vydávány na alespoň pětkrát delší dobu než je životnost uživatelských certifikátů. Pokud se totiž zkrátí životnost, pak je třeba častěji obnovovat uživatelské certifikáty.

Důležitá informace je, že přestože dojde k vypršení platnosti certifikátu, pak je stále tento certifikát potřebný, nesmí být zahozen. Po vypršení platnosti se pouze zamezí podepisování nové zprávy pomocí privátního klíče, jež přísluší do dvojice s veřejným klíčem obsaženým v certifikátu. K ověřování digitálního podpisu zpráv vytvořených ještě během platnosti daného certifikátu je také potřebný nyní již prošlý certifikát.

#### *Jedinečné jméno*

Položka, jež zní anglicky Distinguished Name byla původně zavedena v normách ITU řady X.500, přesně v X.501. Pomocí jedinečného jména je takto možno vytvořit obdobu celosvětového telefonního seznamu. Jedinečné jméno obsahuje velké množství podrobnějších informací o vlastníkově certifikátu. Tyto podrobnější informace lze nazvat jako relativní jedinečné jména (Relative Distinguished Name) viz Tab.2.



Tab. 2: Přehled relativních jedinečných jmen

Identifikátor / zkratka	Atribut	Význam
Common Name /CN	commonName	Název objektu, může být jméno a příjmení či u serveru DNS jméno
Serial number	serialNumber	Rozlišuje certifikáty v případě stejného jedinečného jména. Používá se u kvalifikovaných certifikátů.
Country /C	countryName	Zkratka státu dle normy ISO3166 (např. CZ= Česká Republika, SK = Slovensko)
Locality /L	localityName	Umístění (např. město)
Organization /O	organizationName	Název firmy
DNQualifier	dnQualifier	Slouží k rozlišení různých certifikovaných objektů, kterým by jinak vycházel stejný předmět.
E-mail Address /E	emailAddress či pkcs9mail	Adresa elektronické pošty (dle RFC-822).
Domain Komponent /DC	domainComponent	Řetězce z DNS jména. (pro feec.vutbr.cz je DC=feec, DC=vutbr, DC=cz )

#### *Vydavatel certifikátu*

Vydavatel (Issuer) označuje jedinečné jméno certifikační autority, která certifikát vydala. Certifikační autorita musí mít jedinečné jméno v rámci všech existujících certifikačních aktivit. Spolu s pořadovým číslem certifikátu jednoznačně identifikuje certifikát.

#### *Předmět certifikátu*

Předmět obsahuje identifikační údaje, které musí být jedinečné v rámci v všech objektů certifikovaných danou certifikační autoritou. Podstatou předmětu je, že certifikační autorita nesmí vydat dvěma různým osobám certifikát se stejným předmětem, ale smí vydat více certifikátů se stejným předmětem jedné osobě.

#### *Veřejný klíč*

Tato položka (Subjekt Public Key) obsahuje veřejný klíč a identifikátor algoritmu, pro nějž tento klíč je určen. Narozdíl od položky Algoritmus podpisu je zde specifikován algoritmus, jemuž je určen certifikovaný veřejný klíč. [3]

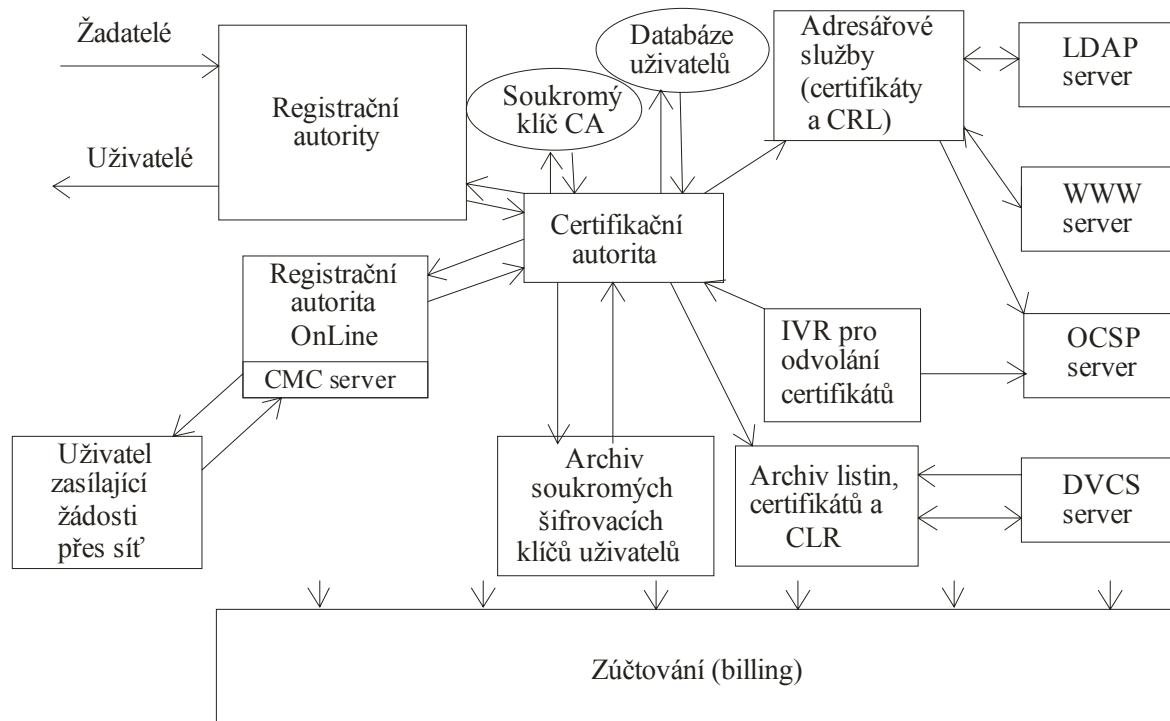
### 3.2 Existenční cyklus certifikátu

Certifikát během svého životního cyklu prochází několika etapami. Zde je následující výpis „života“ certifikátu:

1. Vytvoření žádosti o certifikát: uživatel musí podat žádost o vydání certifikátu certifikační autoritě
2. Vydání certifikátu: certifikační autorita vydá certifikát na základě údajů uvedených uživatelem, jež o certifikát požádal
3. Platnost certifikátu: po vydání certifikátu, nemusí být certifikát okamžitě platný. Jeho platnost začne až od doby, jež je uvedena v položce certifikátu „od“ (Not Before). Skončí buď vypršením platnosti nebo odvoláním certifikátu.
4. Vypršení platnosti certifikátu: nastává po vypršení doby „do“ (Not After) uvedené v certifikátu
5. Odvolání certifikátu: může nastat pouze před vypršením původně stanovené doby platnosti. Jedná se pouze o zveřejnění identifikace certifikátu do seznamu odvolaných certifikátů (CLR). [4]

### 3.3 Certifikační autorita

Certifikační autorita je pojem jež vyjadřuje dva významy. Jednak se jedná o aplikaci, jež vydává digitální certifikáty, nebo o instituci, která zajišťuje proces vydávání certifikátů. V dalším textu bude instituce certifikační autority označena jako CA. Rozšířená struktura instituce certifikační autority je uvedena na Obr.5.



Obr. 5: Schéma certifikační autority – instituce

Nejdůležitější části CA jsou:

#### *Privátní klíč CA*

Nejdůležitější aktivum CA. Pokud by došlo k úniku této informace, pak musí dojít ke zrušení činnosti CA, protože při odcizení privátního klíče útočníkem má útočník přístup ke všem certifikátům. Dále CA musí ochraňovat sekvenci vydaných čísel certifikátů. Pokud by došlo k vydání dvou na sobě nezávislých certifikátů se stejným sériovým číslem, pak by to pro CA znamenalo ukončení činnosti. Dále CA vlastní také veřejný klíč CA, jež ovšem není tajný.

#### *Databáze uživatelů*

Obsahuje osobní údaje o uživateli, jež jsou zaregistrovaní u CA. Patří zde identifikace průkazu totožnosti, rodné číslo apod. CA si musí kontrolovat, zda nevydává určité osobě certifikát se stejným předmětem jež má jiná osoba.

#### *Archiv privátních šifrovacích klíčů uživatelů*

Jsou zde uloženy privátní klíče uživatelů. CA musí zabezpečit ochranu těchto klíčů, v případě jejich ztráty nebo zcizení dojde k nemožnosti šifrování či dešifrování dat uživatele respektive ke zneužití jeho certifikátu.

#### *Archiv listin uložených na CA a archiv vydaných certifikátů a CRL*

Vydané certifikáty a CRL jsou veřejné informace, ale CA je potřebuje pro kontrolu pravosti dokumentů, jež jsou podepsány certifikáty dané CA. Při jejich ztrátě by nebylo možné kontrolovat pravost podepsaných dokumentů.

Další části z nichž je CA sestavena závisí na službách, které CA poskytuje. Patří zde:

#### *Registrační autorita RA*

Zde se osobně vyřizují žádosti o certifikáty. Uživatel přinese požadavek na certifikát, registrační autorita ověří totožnost žadatele, verifikuje požadavek na certifikát a předá jej (zpravidla podepsanou RA) certifikační autoritě. Certifikační autorita ověří údaje z žádosti žadatele a údaje doplněné RA a vydá (či nevydá) příslušný certifikát. Vydaný certifikát může být vrácen na RA, kde je předán žadateli. Jiná strategie spočívá v tom, že RA vydá pouze jednorázové heslo pro vydání certifikátu a uživatel žádost o certifikát pošle elektronicky přes OnLine RA.

#### *OnLine registrační autorita RA*

OnLine RA přijímá žádosti elektronickou cestou. Tímto způsobem má uživatel možnost požádat o obnovení certifikátu v době platnosti starého certifikátu, a dále může žádat o vydání nového certifikátu na základě jednorázového hesla pro vydání certifikátu. Pokud již vlastní svůj platný podpisový certifikát, pak má možnost žádat o další certifikáty.

#### *IVR nebo telefonní záznamník*

IVR (interactive voice response) slouží k odvolávání certifikátu jiným způsobem než elektronicky. Nejčastějším způsobem je telefonické odvolání. Uživatel se musí autentizovat jednorázovým heslem pro odvolání certifikátu. Takto odvolané certifikáty jsou poté zařazeny do fronty certifikátů, jež čekají na odvolání. Informace o odvolání certifikátu jsou poté OnLine zveřejněny na OCSP serveru.

### *Adresářové služby*

V adresářových službách jsou uvedeny nejen informace o uživateli, u nichž sami uživatelé uznají za vhodné aby byly veřejně publikovány. Jsou zde především uvedeny vydané certifikáty a CLR, tedy odvolané certifikáty. Adresáře se zpravidla vytvářejí v několika zálohách, což je výhodně při výpadku příslušného serveru CA.

### *DVCS server*

DVCS server využívá svůj speciální protokol DVCSP. Informuje o platnosti certifikátu pomocí komunikace s CA, která mu tyto informace dá k dispozici.

### *Zúčtovací systém*

Úkolem zúčtovacího systému je vytvořit fakturu za využívané služby. [2]

Certifikační autorita CA bez připojení na Internet, nemá žádný význam. Připojení k internetu je ovšem velice složitá problematika. Certifikační autoritu je nutné od Internetu bezpečně oddělit, protože je v ní uloženo velké množství tajných informací, jejichž zneužití znamená ukončení existence CA. Uživatelé budou na CA přistupovat přes Internet. Pro zabezpečení komunikace je proto nutno použít Internetový FrontEnd Virtual Vault. Na něm poběží všechny potřebné servery, jež poté komunikují přímo s aplikací - certifikační autoritou. [2]

## **3.4 Postup práce certifikační autority a příklad použití certifikátu**

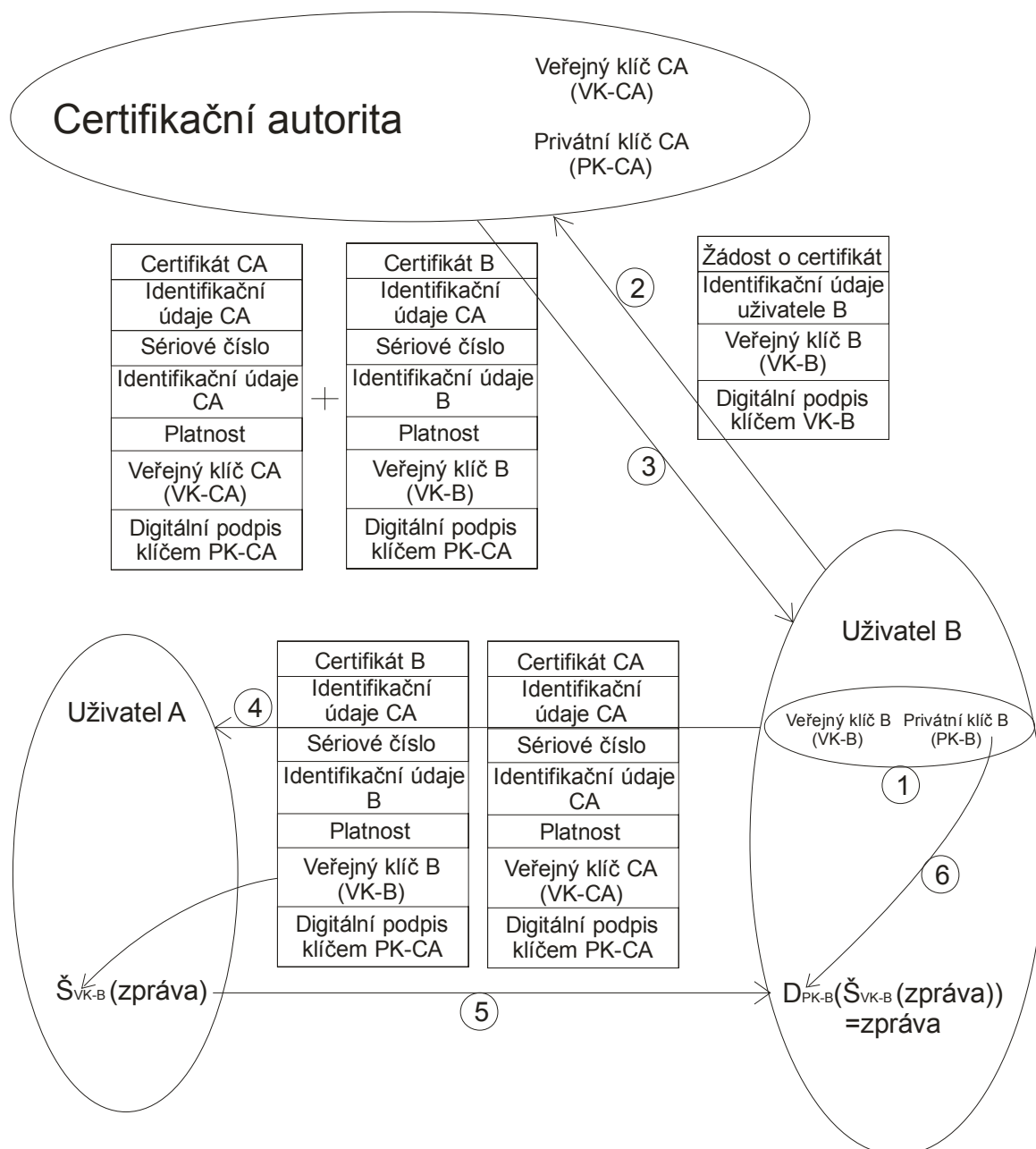
Aby bylo možno provádět operace spojené s použitím certifikátu, je nejdříve nutné jej vlastnit. Zde je uvedena procedura úkonů, jež se provádí průběžně pro získání certifikátu a pak během jeho využití. Tato procedura je objasněna pomocí obrázku a uvedených jednotlivých kroků (viz Obr.6)

Uživatel B si nejdříve musí vygenerovat jednoznačnou dvojici veřejného a privátního klíče (krok 1). Privátní klíč je velice důležitý, proto jej musí bezpečně uchovat.

Uživatel B vytvoří pomocí příkazového řádku nebo webových stránek žádost o certifikát. Takto vytvořenou žádost digitálně podepíše pomocí svého vygenerovaného privátního klíče a odešle certifikační autoritě od níž žádá certifikát (krok 2). V žádosti je mezi identifikačními informacemi také veřejný klíč, pomocí něhož může certifikační autorita ověřit digitální podpis žádosti o certifikát. V případě, že totožnost a digitální podpis uživatele je v pořádku, pak certifikační autorita může vystavit certifikát.

Certifikační autorita vytvoří certifikát se všemi náležitostmi a údaji, které k tomu patří včetně veřejného klíče CA. Celý certifikát digitálně podepíše pomocí svého privátního klíče CA. Nyní je třeba odeslat vystavený certifikát uživateli B (krok 3). Uživatel B obdrží také certifikát certifikační autority. Pomocí něj si ověří digitální podpis vystaveného certifikátu o něj žádal.

V této fázi je již možno uživatelům nabídnout využití šifrování zpráv. Nejdříve ovšem musí uživatel B odeslat uživateli A svůj certifikát (krok 4). Ten ověří digitální podpis a posoudí, zda certifikát je vydán certifikační autoritou, jež je pro něj důvěryhodná. V případě, že je vše v pořádku, může uživatel A použít veřejný klíč, jež je obsažen v certifikátu, k zašifrování zprávy určené pro uživatele B. Takto zašifrovaná zpráva je odeslána uživateli B (krok 5). Ten vlastní svůj privátní klíč a pomocí něj dešifruje přijatou zprávu od uživatele A (krok 6), čímž získá původní zprávu. [2]



Obr. 6: Činnost certifikační autority a šifrovaná komunikace

## 4 Vytvoření CA a certifikátu v operačním systému Microsoft Windows Server 2003

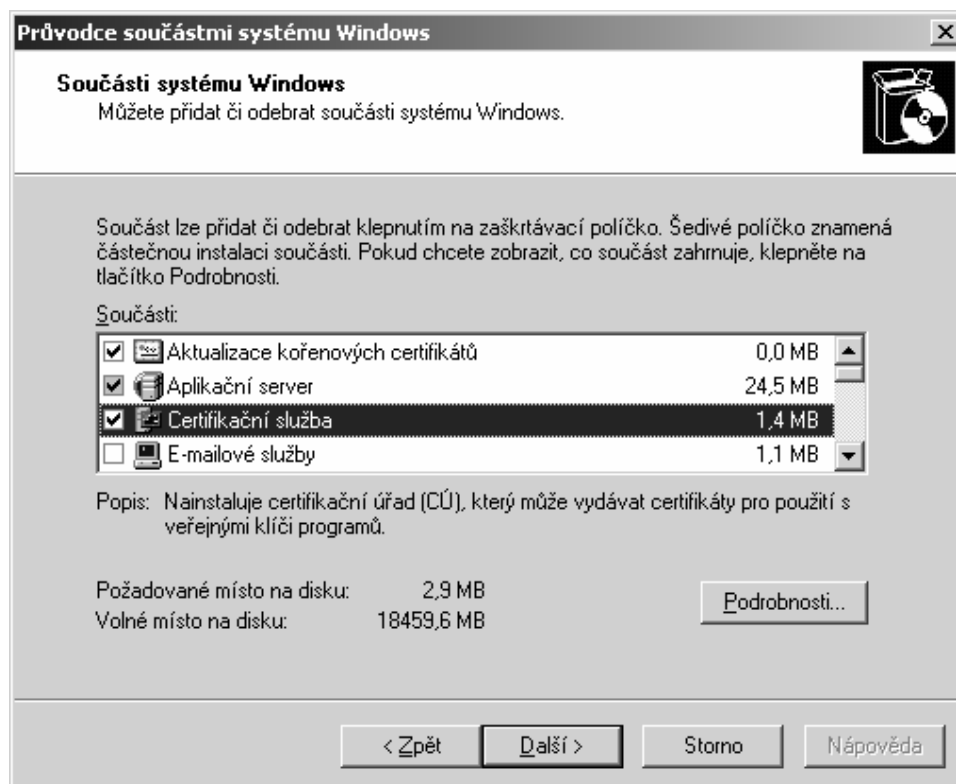
V první části kapitoly je uvedeno jakým způsobem se vytváří CA v operačním systému Windows Server 2003 Enterprise Edition. Další části kapitoly pojednávají o podávání žádosti o certifikát u vytvořené CA a jeho vyhotovení s následnou instalací do webového prohlížeče.

### 4.1 Certifikační autorita

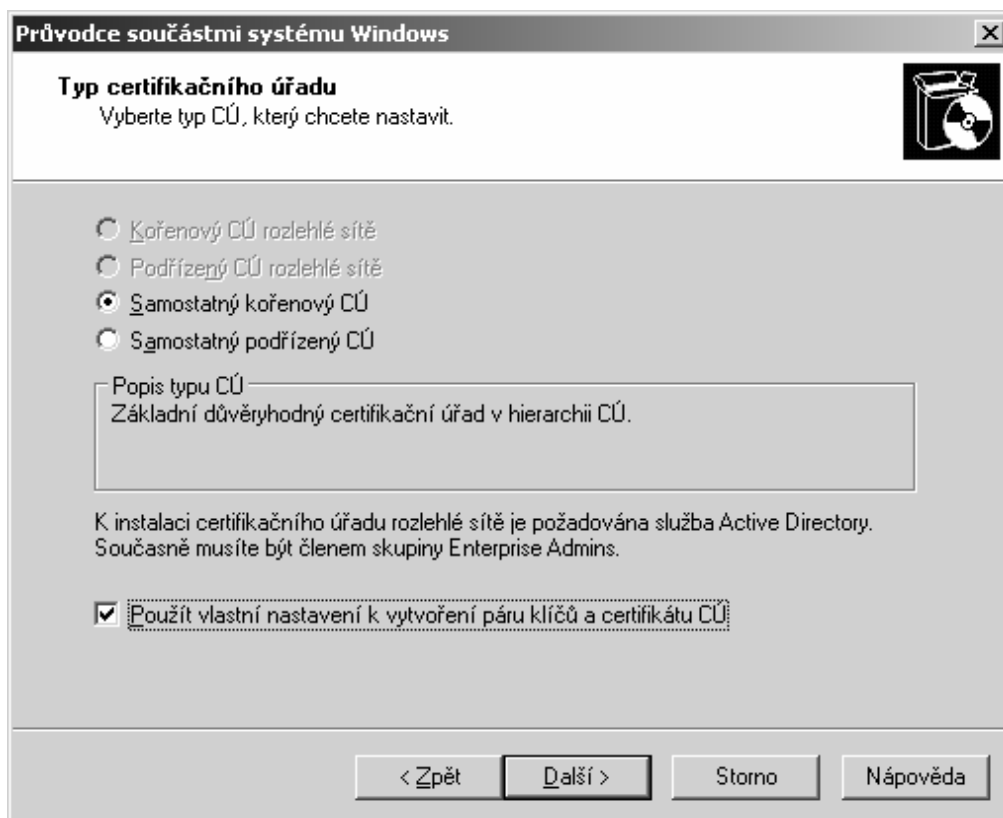
Na počítači, kde je nainstalován operační systém Windows Server 2003 je nejdříve potřeba nainstalovat nutný přídatný software, jež je součástí operačního systému. Jedná se o Certifikační službu a Aplikační server. To se provádí pomocí nabídky Start → Ovládací panely → Přidat nebo odebrat programy. Zde je třeba na levé straně okna vybrat položku Přidat nebo odebrat součásti systému.

Zobrazí se Průvodce součástmi systému Windows (viz Obr. 7). Zde musí uživatel vybrat a zaškrtnout položky Certifikační služba a Aplikační server. Položku Aplikační server je třeba ještě otevřít a zaškrtnout všechny podpoložky, jež obsahuje.

Následující okno zobrazuje první okno s konfigurací certifikační autority. Zde se nabízí, zda uživatel chce vytvořit Samostatný kořenový CÚ nebo Podřízený kořenový CÚ. Je vhodné vybrat Samostatný kořenový CÚ. Ve spodní části okna je nutné zatrhnout volbu „Použít vlastní nastavení pro vytvoření páru klíčů a certifikátu CÚ“ (viz Obr. 8), čímž se zpřístupní možnosti pro podrobnější nastavení vlastností klíčů.



Obr. 7: Instalace certifikační autority, 1. část

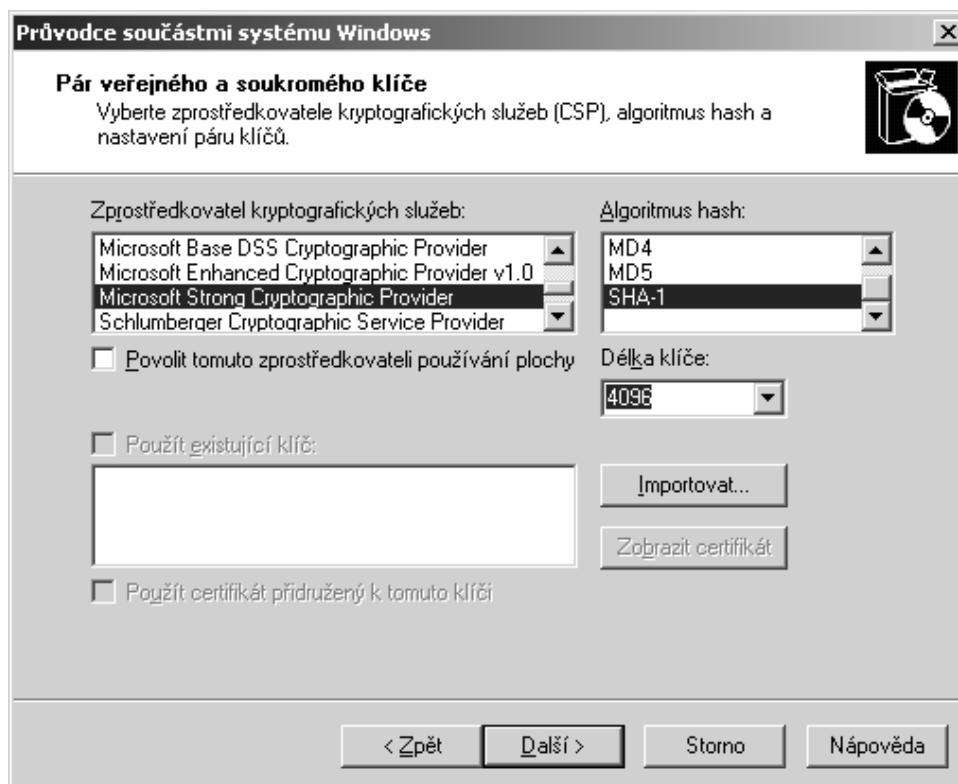


Obr. 8: Instalace certifikační autority, 2. část

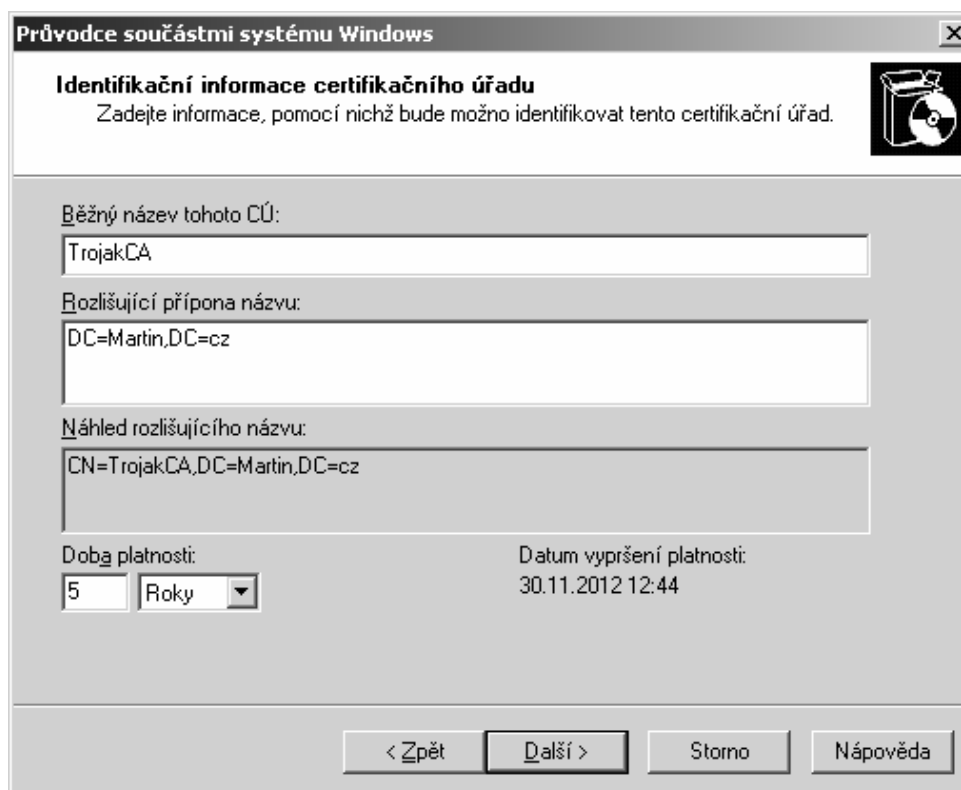
Další okno se zabývá definicemi kryptografických služeb (viz Obr. 9). V levé tabulce se vybírá do jakého úložiště budou uloženy dvojice veřejný/privátní klíč vytvářené CA. Pokud chce uživatel ukládat uvedené informace na pevný disk, pak vybere položku Microsoft Strong Cryptographic Provider. V pravé tabulce se určuje formát hash. V současné době je nejvhodnější a nejbezpečnější SHA-1. Poslední výběr se provádí pro volbu délky klíče. Jestliže uživatel vytváří „Samostatnou kořenovou CA“, pak je nutné pro nejvyšší bezpečnost použít hodnotu 4096 bitů. V případě Podřízené kořenové CA se standardně využívá hodnota 2048 bitů.

Následující okno nabízí možnost určit si jméno CA, rozlišující příponu a dobu platnosti. Standardní doba platnosti se udává na 5 let, ostatní položky jsou uvedeny v Obr. 10. V posledním okně se určuje cesta k jednotlivým souborům, tedy souborům log a záznamům o vydaných certifikátech.

Proběhne instalace přídatného softwaru a po provedení potvrdíme tlačítkem Dokončit.



Obr. 9: Instalace certifikační autority, 3. část



Obr.10: Instalace certifikační autority, 4. část



## 4.2 Žádost o certifikát

Pro vystavení certifikátu uživatele a certifikátu CA je potřeba podat žádost o certifikát. Pro přístup k podání žádosti o certifikát od CA v Microsoft Windows Server 2003 je možno se připojit ze vzdáleného počítače přes webové rozhraní. Tento postup bude nyní popsán.

Nejdříve je potřeba připojit se k webu certifikační autority pomocí adresy: `http://<IP adresa CA>/certsrv`. Prakticky to může být např. `http://192.168.30.130/certsrv`. Úvodní stránka nabídne možnost „Vytvořit certifikát“. Posléze je možno zvolit, o jaký typ certifikátu uživatel žádá. Svým výběrem se dostane na další stránku, kde se již zapisují identifikační informace o žadateli a další údaje, jež musí výsledný certifikát obsahovat (viz Obr. 11). Po odeslání žádosti musí uživatel čekat na vyřízení CA.

Certifikační služba společnosti **Microsoft** -- TrojakCA
[Domů](#)

### Upřesnit žádost o certifikát

---

**Identifikační informace:**

Jméno:

E-mailová adresa:

Společnost:

Oddělení:

Město:

Okres:

Země:

**Zamýšlený účel:**

**Možnosti klíče:**

Vytvořit novou sadu klíčů   
  Použít existující sadu klíčů

Zprostředkovatel CSP:

Použití klíče:  Výměna     Podpis     Obě

Délka klíče:    
 Min: 384    Max: 16384   
 (běžné délky klíčů: 512 1024 2048 4096 8192 16384)

Automatický název kontejneru klíče   
  Uživatelem určený název kontejneru klíče

Označit klíče jako exportovatelné  
 Povolit ochranu silným soukromým klíčem  
 Uložit certifikát v úložišti certifikátů místního počítače  
*Stores the certificate in the local computer store namísto uložení v úložišti certifikátů uživatele. Neinstaluje certifikát kořenového CÚ. Aby jste mohli generovat nebo použít klíč z úložiště místního počítače, musíte mít oprávnění správce.*

**Další možnosti:**

Formát požadavku:  CMC     PKCS10

Algoritmus Hash:

*Používá se pouze k podepsání požadavku.*

Uložit žádost do souboru

Atributy:

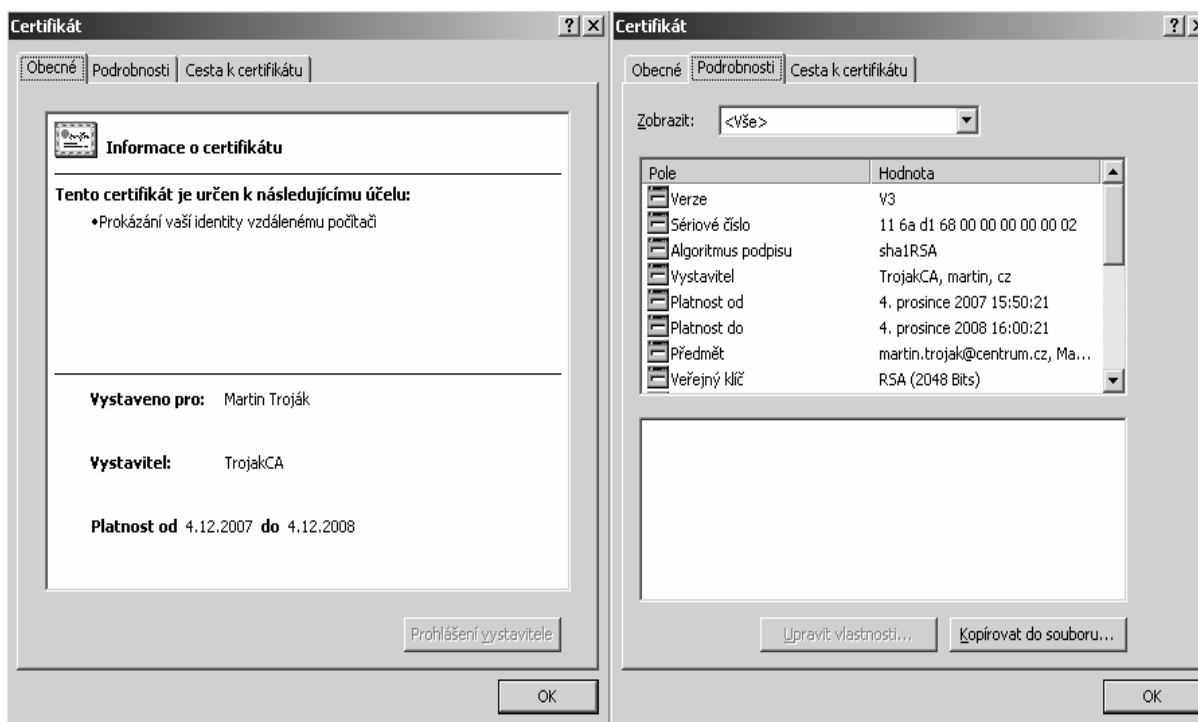
Popisný název:

Obr. 11: Žádost o certifikát

### 4.3 Vyhotovení certifikátu

Správa certifikátů v operačním systému Windows Server 2003 probíhá přes konzolu *mmc*. Spouští se přes nabídku Start → Spustit → mmc. Zde se konfigurují služby operačního systému. Pro správu certifikátů jsou nejdůležitější položky Certifikační úřad, Certifikáty a Šablony certifikátů. Po přijetí žádosti o certifikát buď CA automaticky sama vyhotoví uživateli certifikát (Obr. 12), nebo tento úkon nechá na operátorovi či registrační autoritě.

V tomto případě řeší vyhotovení certifikátu správce serveru. Správce zkontroluje údaje žadatele a rozhodne, zda se vyhotovení provede či nikoli.



Obr.12: Digitální certifikát

### 4.4 Instalace certifikátu

Jakmile CA vystaví žadateli certifikát, má uživatel možnost si tento vydaný certifikát implementovat do svého internetového prohlížeče. Slouží k tomu opět webové stránky CA, na kterých uživatel žádal o vyhotovení certifikátu. Certifikát se nainstaluje do webového prohlížeče a jakmile uživatel bude chtít přistupovat na webové stránky, jež vyžadují šifrovanou komunikaci pomocí tohoto certifikátu, provede se procedura uvedená v kapitole 3.4.

## 5 SSL

SSL (Secure socket layer) byl vyvinut v roce 1996 firmou Netscape jako nekomerční otevřený protokol. Jeho využití je tedy umožněno pro soukromé i komerční účely.

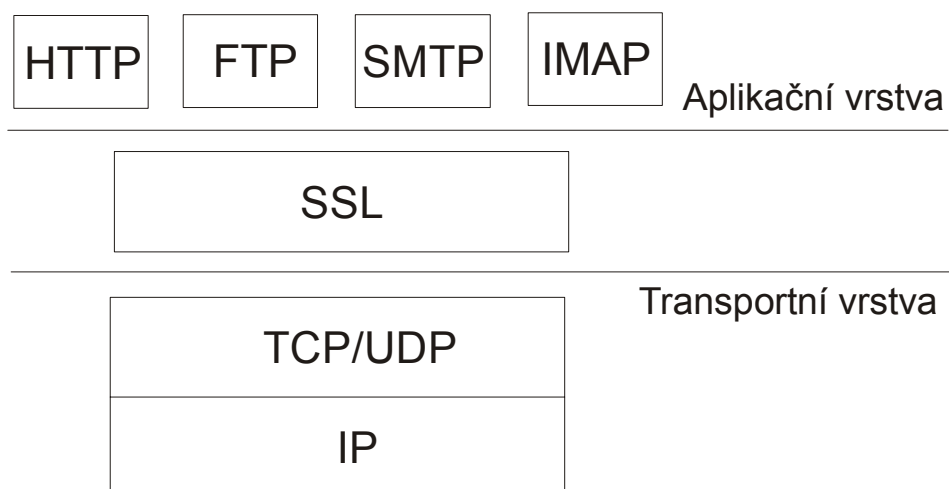
SSL je vrstva/protokol zabezpečující data na přechodu mezi aplikační a transportní vrstvou (protokolem TCP/IP). Zajišťuje se takto šifrování přenášených dat a autentizace serveru pomocí digitálních certifikátů. SSL není nijak omezeno pouze na protokol HTTP. SSL je možno použít i pro bezpečné připojení prostřednictvím FTP, NNTP ale i k poštovním službám přes SMTP, POP3, IMAP4 a řadu dalších protokolů. Označení těchto zabezpečených komunikačních protokolů je na konci rozšířeno o písmeno „s“. Je tedy možno využívat protokoly HTTPS, FTPS, NNTPS, SMTPS, POP3S a IMAP4S. [7]

### *Hlavní přínosy SSL*

- Bezpečnost šifrování: primárním přínosem protokolu SSL je ustavení bezpečného spojení mezi dvěma komunikujícími uzly. Poté co jsou iniciačním algoritmem vyměněny bezpečné klíče, je používáno symetrické šifrování.
- Spolehlivost: při přenosu zprávy je zajištěna integrita dat entitou MAC (Message Authentication Code).
- Interoperabilita: nezávisle vyvíjené aplikace jsou schopny si mezi sebou vyměňovat parametry bez vzájemné znalosti kódu.
- Rozšiřitelnost: možnost implementace jiných metod šifrování a výměny veřejných klíčů.
- Relativní efektivita: podpora komprimace dat nebo cacheování spojení. [7]

### 5.1 Princip SSL

Protokol SSL zajišťuje soukromí a spolehlivost pro komunikující aplikace, chrání data před odposloucháváním, zfalšováním a paděláním. V TCP/IP modelu je umístěn mezi transportní a aplikační vrstvou. Jedná se tedy o přidanou podvrstvu. Přesné umístění je uvedeno v Obr. 12. [8]



Obr.13: Umístění SSL v TCP/IP modelu



**b) Connection:** spojení mezi procesy probíhající v rámci jedné session

*Využité parametry:*

- Náhodné číslo generované klientem (ClientRandom).
- Náhodné číslo vygenerované serverem (ServerRandom). Obě náhodná čísla jsou na počátku přenášena nezabezpečeně.
- Tajemství pro výpočet kontrolního součtu používané serverem (server write MAC secret).
- Tajemství pro výpočet kontrolního součtu používané klientem (klient write MAC secret).
- Symetrický šifrovací klíč, kterým šifruje server (server write key).
- Symetrický šifrovací klíč, kterým šifruje klient (klient write key).
- Inicializační vektory (IV) používané pro blokové šifry.
- Číslo přijaté a číslo odeslané zprávy.

Každý proces connection existuje pouze v jednom session, ovšem v jedné session může existovat více connection. Struktura connection je definována parametry ověřování MAC. Struktura session je definována parametry šifrování. [9]

## 5.2 Handshake SSL

Protokol Handshake SSL bývá označován také jako key-exchange protocol, tedy protokol pro výměnu klíčů. Jeho význam spočívá v ustavení bezpečné cesty (session) mezi dvěma účastníky. Při vytváření session prochází Handshake SSL několika stavy. Nejdříve klient ověří server, dále si spolu klient a server domluví společné šifrovací algoritmy a šifry. V další fázi ověří server klienta. Poté pomocí asymetrické šifry vymění server s klientem šifrovací parametry, tedy sdílená hesla. Nakonec ustaví zabezpečené SSL spojení (connection). V Obr. 15 je uvedena struktura SSL Handshake Protocol, kterým se tyto operace provádějí. [10]

typ zprávy	délka zprávy (v bytech)	parametry přidané ke zprávě
8 bitů	24 bitů	

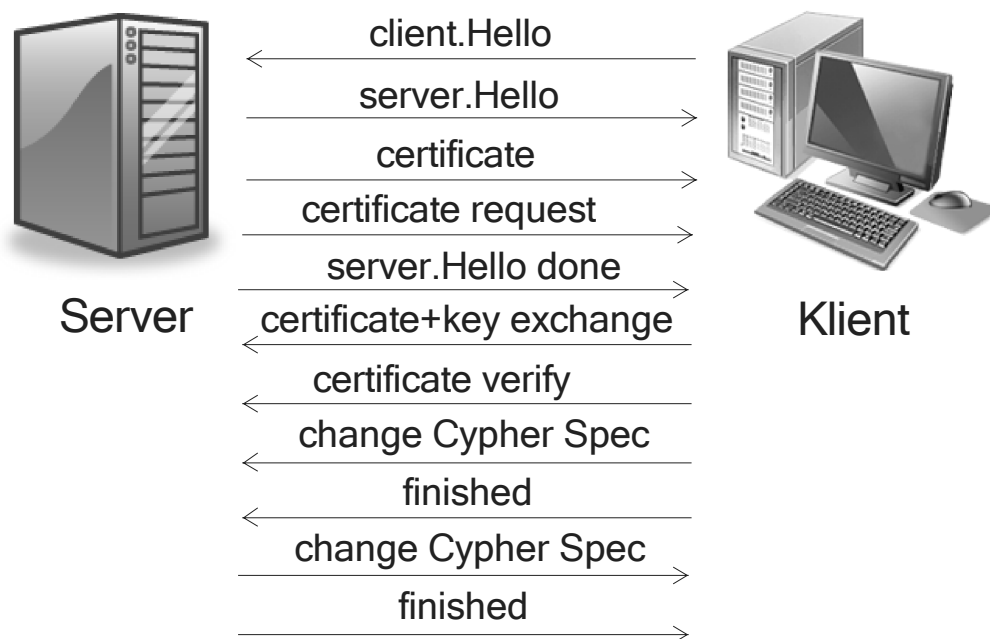
Obr.15: Struktura protokolu SSL Handshake Protocol

*Komunikace mezi klientem a serverem*

1. Klient odešle serveru zprávu client.Hello, která se skládá z informace o nejvyšší verzi SSL/TLS, dále o šifrách, jež klient podporuje. Obsahuje také klientem podporované kompresní metody a session ID. Při zakládání nové SSL session je hodnota session ID rovna 0. Posledními informacemi zprávy jsou náhodně vygenerovaná data pro testování šifrování.
2. Server odešle klientu zprávu server.Hello, jež obsahuje informace o verzi SSL/TLS, která bude použita pro SSL session, a šifru, která bude pro SSL session použita. Dalšími informacemi jsou ID session pro danou SSL session a náhodná data pro testování šifrování.

3. Server odesílá klientovi zprávu Certificate. Skládá se z certifikátu serveru a v případě existence dalších zřetězených certifikátů a také certifikátu CA, jímž je certifikát serveru podepsán.
4. Server odesílá klientovi požadavek na certifikát.
5. Server odesílá klientovi zprávu server.Hello done. Znamená to, že server dokončil fázi Handshake.
6. Klient odesílá serveru svůj certifikát a v něm obsažený privátní a veřejný klíč.
7. Klient informuje server o tom, že ověřil jeho certifikát.
8. Klient odesílá serveru zprávu CHANGE\_CIPHER\_SPEC, což značí, že následující odeslaná data v rámci dané SSL session budou zašifrována. Nešifruje se pouze záhlaví dat.
9. Klient odesílá zprávu FINISHED. Je složena z přehledu všech SSL handshake zpráv, které byly doposud vzájemně předány mezi klientem a serverem. Jedná se o ověření, zda nedošlo ke ztrátě informací, jež nebyly zašifrovány při handshake.
10. Server odesílá klientovi zprávu CHANGE\_CIPHER\_SPEC, což značí, že následující odeslaná data v rámci dané SSL session budou zašifrována.
11. Server odesílá klientovi zprávu FINISHED. Je složena z přehledu všech SSL handshake zpráv, které byly doposud vzájemně předány mezi klientem a serverem. [11]

Popis komunikace mezi klientem a serverem je zobrazen v Obr. 16.



Obr.16: Komunikace mezi klientem a serverem pomocí SSL Handshake Protocol

V případě, že se o komunikaci se serverem snaží proces stejného klienta, pak tento proces může využít spojení (session) se serverem s nímž již komunikuje jiný proces. Není tedy třeba provádět úplný handshake proces, ale stačí zkrácená verze využívající existující session ID. [10]

### 5.3 Change Cipher Spec Protocol a Alert Protocol

SSL Change Cipher Spec Protocol je využit v jedné z fází SSL Handshake protokolu. Umožňuje účastníkům přesun z vyčkávacího do provozního stavu. Dochází zde k ukončení nešifrované komunikace při výměně certifikátů a je nově využito šifrované komunikace pomocí šifrovacích a ověřovacích (MAC) algoritmů, jež byly definovány v předchozích fázích Handshake protokolu.

Zpráva tohoto protokolu má délku 1 bajt a je šifrována a komprimována pomocí dohodnutých algoritmů.

SSL Alert Protocol předává informace o chybách, jež se objevují v průběhu celého spojení (connection). Existují dvě úrovně výstrah: fatální a varovná. V případě existence fatální výstrahy dojde okamžitě k ukončení spojení. Všechna ostatní spojení komunikující ve stejné session pokračují v komunikaci bez přerušení, ovšem session ID dané session bude označeno jako neplatné a nebude možno vytvořit nové connection v rámci zneplatněné session. Zpráva je tvořena dvěma částmi po 8 bitech, první část nese označení Level, slouží k indikaci fatální nebo varovné zprávy. Druhá část, Alert, indikuje specifickou výstrahu. [12]

Příklady fatálních výstrah:

- `Bad_record_mac` - špatná hodnota MAC
- `Decompression_failure` - délka zprávy po dekompresi překročila maximum
- `Handshake_failure` - chyba při vyjednávání parametrů

Příklady varovných výstrah:

- `Certificate_expired` - certifikát vypršel
- `Certificate_revoked` - certifikát byl zrušen tím kdo jej vystavil
- `Unsupported_certificate` - nepodporovaný typ certifikátu

### 5.4 Record Protocol

Record protokol se stará o balení dat, jež budou přenášena do objektu record. Objekt rekord je tvořen hlavičkou a daty. Délka hlavičky recordu je 5byte.

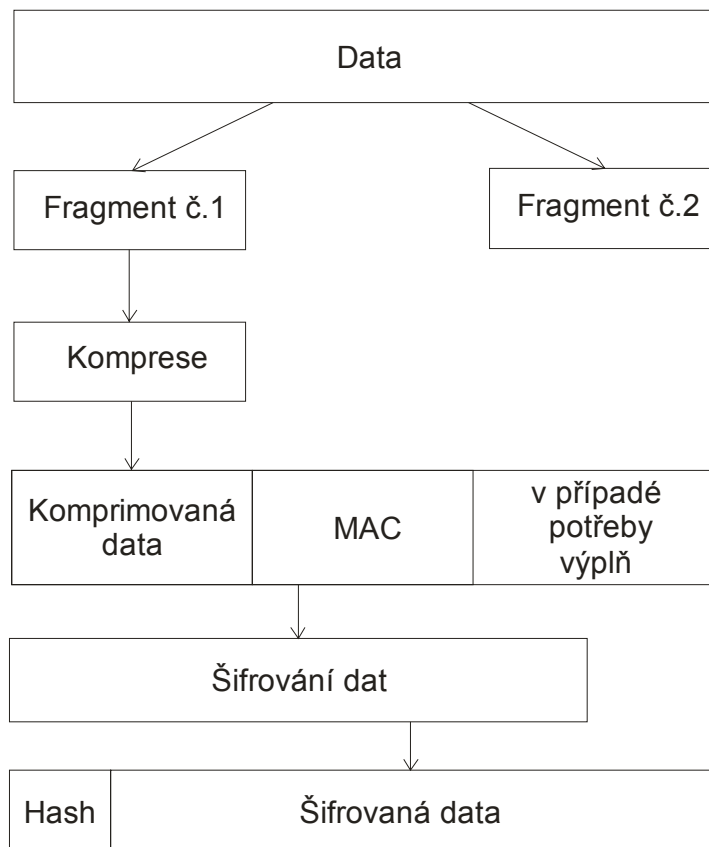
Hlavička se skládá z:

- *Type* ( 8 bit ) – slouží k indikaci datového typu a protokolu vyšší vrstvy, jež následně data zpracovává. Typy jsou: `change_cipher_spec` (změna specifikace šifrování), dále `alert` (výstraha), `handshake` a `application_data` (aplikační data).
- *Version* (16 bitů) – informace o verzi SSL protokolu (major i minor).
- *Length* (16 bitů) – informace o délce datového pole.

Za hlavičkou následují data, jež jsou podrobena úpravami ve 4 fázích (Obr. 17).

*Fragmentace*

Aplikační data jsou rozfragmentována do textových SSL recordů o maximální délce  $2^{14}$  bytů. Výstupem z fragmentačního bloku je `SSLPlaintext`.



Obr.17: Vznik datové části záznamu

### Komprese

Ve fázi Handshake došlo k dohodě ohledně použitého komprimačního protokolu. Nyní je tento protokol využit pro komprimaci textových záznamů tzv. SSLPlaintext. Během komprimace nesmí dojít ke ztrátě dat. Výsledkem komprimace je SSLCompressed. Jeho délka nesmí překročit velikost výše uvedené maximální délky.

### Přidání autentizačního řetězce:

K vytvořenému komprimovanému záznamu SSLCompressed je nyní přidán autentizační řetězec. Je tvořen:  $h = H(\text{AKO} \parallel \text{ipad} \parallel H(\text{AKO} \parallel \text{opad} \parallel \text{SN} \parallel \text{PT} \parallel \text{SL} \parallel S))$ , kde  $H$  = hashovací funkce, AKO je autentizační klíč odesílatele, *ipad* tvoří řetězec bajtů 0x36, *opad* je řetězec bajtů 0x5C, SN je číslo zprávy, PT je číslo protokolu v nadřazené vrstvě OSI, SL je délka segmentu SSLCompressed. Podporovány jsou dva hashovací algoritmy, jednak MD5 a dále SHA-1.

### Šifrování

Pro šifrování jednotlivých bloků jsou použity blokové a proudové šifry symetrického šifrování. V případě proudových šifer není potřeba doplňovat velikost bloku, tedy record je využit přímo ve formátu, v jakém jej proces šifrování získal od předchozího procesu. Naopak u blokové šifry je nutné doplnit délku recordu na délku bloku dané šifry. Celková délka zašifrovaných dat nesmí překročit  $2^{14}$  byte.

SSL v3 podporuje blokové šifry IDEA, DES, 3DES, Fortezza a proudovou šifru RC4. [13]



## 6 Použité nástroje pro realizaci certifikační autority a digitálního podpisu

V této kapitole jsou uvedeny nástroje s jejichž pomocí bylo vytvořena praktická část diplomové práce. Byl využit VMware Workstation pro tvorbu virtuálního počítače a VMware Player pro provoz. Dále webový server Apache pro Win32, programování prostředí proběhlo pomocí programovacího jazyka PHP a MySQL. Pro tvorbu certifikátů bylo využito OpenSSL. Rovněž pro funkčnost a podporu SSL bylo potřeba implementovat do webového serveru Apache Mod\_SSL.

### *VMware Workstation a VMware Player*

Praktická část práce je tvořena v operačním systému Microsoft Windows Server 2003 R2, jež byl nainstalován do virtuálního počítače VMware. Operační systém, kterým je provozován VMware Player verze 2.0.3, je Microsoft Windows XP SP2.

### *EasyPHP*

Pro aplikaci PHP skriptů byl využit instalační balíček EasyPHP verze 1.8. Součástí instalace je webový server Apache 1.3.33, dále MySQL 4.1.9, PHP 4.3.10 a PHPMyAdmin 2.6.1. Pro správu celého balíku aplikací se využívá konfigurační nástroj EasyPHP, který v sobě soustředí možnosti konfigurace a řízení všech nástrojů dohromady.

Velikou výhodou tohoto balíku je možnost spouštět EasyPHP z flash disku, kdy se při startu přegenerují konfigurační soubory, je zde tedy přínosná možnost přenosu aplikace na jiný počítač. Dalšími výhodami jsou snadná konfigurace prostředí, dostupná správa extenzí, starty a restarty serverů apod.

### *Apache Mod\_SSL*

Pro implementaci SSL funkcí do webového serveru je použit Apache 1.3.39 Mod\_SSL\_2.8.30. Jedná se o funkční konfiguraci Apache web serveru s importovanými knihovnami ModSSL. Balík ModSSL vyvinul v dubnu roku 1998 Ralf S. Engelschall na základě volně šiřitelného souboru nástrojů OpenSSL. OpenSSL byla vyvinuta Benem Laure z knihoven SSLeay od tvůrců Erica A. Young a Tima J. Hudson.

Balík ModSSL má licenci BSD, tedy je volně šiřitelný v nekomerční i komerční sféře vývoje.

### *Konfigurace aplikací*

Pro správnou funkčnost webového serveru Apache je potřeba provést konfiguraci všech použitých programů včetně podpory MySQL. Nejdříve se instaluje programový balíček EasyPHP, poté se implementuje OpenSSL. Následuje vytvoření serverového certifikátu pomocí OpenSSL a nastavení Apache pro podporu ModSSL.

## 6.1 Konfigurace VMware Player

Při instalaci VMware Playeru se do operačního systému Microsoft Windows XP SP2 přidají virtuální síťové karty, které je potřeba nakonfigurovat pro správnou funkčnost. Jelikož je serverový certifikát tvořen pro pevně danou IP adresu, je nutné mít povolenu pouze jednu virtuální síťovou kartu a pro protokol TCP/IP mít nastavenou IP adresu 192.168.30.1 s maskou podsítě 255.255.255.0. Tato virtuální síťová karta má funkci výchozí brány pro síťovou kartu ve virtuálním operačním systému Microsoft Windows Server 2003.

Ve virtuálním operačním systému MS Windows Server 2003 R2, provozovaném pomocí VMware Playeru, je následně nutné mít nastavenou konfiguraci síťové karty pro protokol TCP/IP na *192.168.30.145*. Současně je potřeba mít ve spuštěném VMware Playeru provozováno připojení k síti Ethernet pomocí NAT. Pokud je systém a WMware Player nakonfigurován jinak, než je uvedeno, nebude umožněn přístup k webovému rozhraní aplikace, které je popisováno v kapitole 7.

## 6.2 Instalace EasyPHP a jeho konfigurace

Instalace balíčku EasyPHP v1.8 proběhla pomocí instalačního programu. Soubor aplikací EasyPHP byl nainstalován na jednotku pevného disku C: do složky easyphp. Umístění bylo zvoleno pro zajištění přístupu ke konfiguračním souborům a optimalizaci délky cest k souborům. V této fázi se testuje přístup k webovému serveru Apache pomocí webového prohlížeče ve virtuálním operačním systému zadáním adresy `http://localhost`.

Další krok se týká konfigurace portu pro podporu SSL. Protokol http využívá pro komunikaci standardně port 80. Komunikace pomocí protokolu https, jež podporuje SSL, probíhá na portu 443. Pro správnou funkci obou uvedených protokolů je potřeba provést konfiguraci webového serveru Apache. Konfigurační soubor `httpd.conf` se nachází v `c:\easyphp\apache\conf\`. V tomto souboru se změní následující. Zakomentuje se řádek *Port 80*. Přidá se řádek *Listen 443* pro naslouchání serveru na portu 443 a *Listen 127.0.0.1:80* pro přístup administrátora do MySQL databáze. V položce `ServerName` se nastaví DNS jméno nebo IP adresa serveru, použil jsem `localhost`. Nyní se server otestuje zadáním adresy `http://192.168.30.145:443/` do webového prohlížeče v operačním systému Microsoft Windows XP SP2. Otestuje se tím funkčnost, zatím stále bez běhu SSL [14].

### 6.2.1 Implementace OpenSSL

První fáze se týká stažení balíčků `Apache_1.3.39-Mod_SSL_2.8.30-Win32` a `OpenSSL verze 0.9.8g-win32` a konfiguračního souboru `OpenSSL openssl.cnf` z odkazu `http://tud.at/programm/openssl.cnf`. Druhá fáze spočívá v rozbalení obou aplikací do složek. Soubor `openssl.cnf` se uloží do adresáře, kde jsou rozbaleny knihovny `OpenSSL`. Z téže složky `SSL` se zkopírují soubory `ssleay32.dll` a `libeay32.dll` do adresáře `Windows\System32`.

### 6.2.2 Tvorba certifikátu serveru

Pro správnou funkci `SSL` v `Apache` se vygeneruje serverový certifikát. Jedná se o `self-signed` certifikát, tedy certifikát podepsaný vlastním privátním klíčem.

Certifikáty se generují pomocí knihoven `OpenSSL`. Po spuštění příkazového řádku se ve složce `OpenSSL` zadávají následující příkazy. V první fázi je vytvořena žádost, jež se uloží do souboru `server.req`. Následně se uloží vygenerovaný privátní klíč do souboru `privkey.pem`.

```
openssl req -config openssl.cnf -new -out server.req
```

Po zadání výše uvedeného příkazu je uživatel dotázán na `passphrase`, heslo privátního klíče. Následně se zadávají identifikační údaje serveru a nejdůležitější položka

Common Name. Zde se zadává DNS jméno nebo IP adresa serveru. V mém případě je zadáno localhost.

Pro přístup webového serveru Apache k privátnímu klíči se pomocí následujícího příkazu odstraní z privátního klíče passphrase. Musí být zajištěno, aby přístup k souboru privátního klíče měl pouze Apache server a administrátor.

```
openssl rsa -in privkey.pem -out server.key
```

Další fází je vytvoření certifikátu. Ze žádosti uložené v souboru server.req je vygenerován certifikát a uložen do souboru server.cer. Certifikát je podepsán svým privátním klíčem ze souboru server.key. Položka `-days` definuje dobu platnosti certifikátu.

```
openssl x509 -in server.req -out server.cer -req -signkey  
server.key -days 365
```

Vytvořený certifikát server.cer a privátní klíč server.key se zkopírují do nové složky `c:\easyphp\apache\conf\ssl`.

### 6.2.3 Nastavení Apache pro podporu ModSSL

Pozastaví se webový server Apache. Z adresáře, kde je rozbalen balíček `Apache_1.3.39-Mod_SSL_2.8.30-Win32` se zkopírují všechny soubory `*.exe`, `*.dll` a `*.so` do adresáře `c:\easyphp\apache` dle původní adresářové struktury. Jedná se o soubory, jež jsou zkompileovány s podporou ModSSL.

Další konfigurace se týká nastavení webového serveru pro podporu SSL. Do konfiguračního souboru Apache `httpd.conf` se v části `LoadModule` přidá příkaz `LoadModule ssl_module modules/mod_ssl.so`. Do sekce `AddModule` se přidá příkaz `AddModule mod_ssl.c`.

V poslední fázi se na konec konfiguračního souboru přidají informace o certifikátu a aktivaci SSL. V položce `VirtualHost` se uvede DNS jméno nebo IP adresa webového serveru. V našem případě se jedná o označení localhost.

```
SSLMutex sem  
SSLRandomSeed startup builtin  
SSLSessionCache none  
SSLLog logs/SSL.log  
SSLLogLevel info  
<VirtualHost 192.168.30.145:443>  
    SSLEngine On  
    SSLCertificateFile conf/ssl/server.cer  
    SSLCertificateKeyFile conf/ssl/server.key  
</VirtualHost>
```

Apache se znovu spustí. Takto nakonfigurovaný webový server s komunikací pomocí SSL se otestuje zadáním adresy `https://192.168.30.145/` ve webovém prohlížeči.  
[14]

## 7 Webové rozhraní aplikace

Tato kapitola popisuje vytvořené webové rozhraní, jež je součástí diplomová práce. Dále se zabývá způsobem použití a popisem zabezpečeného webového rozhraní. První podkapitola se zabývá postupem při vstupu uživatele na webové stránky. Ve druhé kapitole je uvedena struktura webových stránek. Třetí podkapitola vysvětluje princip použitých funkcí a postupů při komunikaci mezi stránkami a databází. Současně jsou uvedeny nejdůležitější použité zdrojové kódy. Poslední kapitola se zabývá testováním komunikace a zachytáváním šifrovaných dat pomocí analyzátoru síťového provozu Wireshark. Stránky pracují na následujícím principu: Po zadání IP adresy webových stránek do prohlížeče uživatel přijme certifikát serveru. Poté mu certifikační autorita vygeneruje uživatelský certifikát. Pomocí tohoto certifikátu se poté uživatel přihlásí na svůj účet, kde jsou zobrazeny jemu určené informace.

### 7.1 Prezentace webových stránek a postup použití

Přístup na webové stránky probíhá pomocí zadání následujícího řetězce: `https://<IP adresa serveru Apache>/`. Defaultně je IP adresa nastavena na 192.168.30.145. Uživatel přijme certifikát, načež proběhne handshake a dojde k zahájení šifrovaného spojení. Uvítá jej úvodní stránka, která nabídne přístup k certifikační autoritě či k informačnímu systému. Pro přístup do informačního systému je nejdříve potřeba požádat o osobní certifikát. Je tedy potřeba kliknout na odkaz „Certifikační autorita“ či obrázek „CA Troják“. Zobrazí se formulář, který musí je třeba vyplnit viz Obr. 18. V případě, že uživatel zadá do položky „Jedinečné jméno“ údaj, který již byl v minulosti zadán a uložen do databáze, pak je požádán o zadání jiného jména. Pro kontrolu zadaného hesla je potřeba stejný údaj vyplnit také do položky „Ověření hesla“. Pro pokračování je níže umístěno tlačítko „Vygenerovat certifikát“.

Vítá Vás Certifikační autorita CA Troják

Vyplňte prosím údaje o osobě, pro kterou bude certifikát vygenerován.

Jedinečné jméno:	<input type="text" value="Martin Troják"/>
E-mailová adresa:	<input type="text" value="martin.trojak@centrum.cz"/>
Název organizace:	<input type="text" value="FEKT"/>
Název oddělení:	<input type="text" value="Student"/>
Město:	<input type="text" value="Brno"/>
Stát:	<input type="text" value="Česka Republika"/>
Zkratka státu:	<input type="text" value="cz"/>

Nyní Vás poprosíme o zadání hesla pro přihlašování:

Heslo:	<input type="password" value="•••••"/>
Ověření hesla:	<input type="password" value="•••••"/>

Obr.18: Formulář generátoru certifikátů

Jakmile certifikační autorita zjistí, že některý údaj byl zadán chybně, uživatele o tom informuje a požádá o nápravu chyby. V případě pozitivního výsledku kontroly údajů proběhne generování certifikátu. V průběhu tvorby certifikátu jsou uživateli zobrazeny jednotlivé fáze generování viz Obr. 19. Po úspěšném vytvoření certifikátu si uživatel kliknutím na odkaz „Uložit certifikát na pevný disk“ musí certifikát uložit na bezpečné místo. V případě ztráty certifikátu či prozrazení bude nutno vygenerovat certifikát nový s novými údaji včetně nového „Jedinečného jména“.

**Probíhá vytvoření Vašeho certifikátu:**

**Kontrola zadaných údajů...**  
Zadané údaje = array (  
    'commonName' => 'Martin Troják',  
    'emailAddress' => 'martin.trojak@centrum.cz',  
    'organizationName' => 'FEKT',  
    'organizationalUnitName' => 'Student',  
    'localityName' => 'Brno',  
    'stateOrProvinceName' => 'Ceska Republika',  
    'countryName' => 'cz',  
  )  
Proběhlo v pořádku.

**Generování privátního klíče...**  
Proběhlo v pořádku.

**Vytvoření žádosti o certifikát...**  
Proběhlo v pořádku.

**Podepisování žádosti...**  
Proběhlo v pořádku.

**Exportování certifikátu X.509...**  
Resource id #11 Proběhlo v pořádku.

**Exportování zašifrovaného privátního klíče...**  
Proběhlo v pořádku.

**Ukládání certifikátu...**  
Proběhlo v pořádku.

**Ukládání zašifrovaného privátního klíče...**  
Proběhlo v pořádku.

**Certifikát byl vytvořen**

Nyní máte možnost si certifikát uložit na pevný disk a bezpečně jej uchovat.  
--> [Uložit certifikát na pevný disk](#)

Obr.19: Vyhotovení certifikátu certifikační autoritou

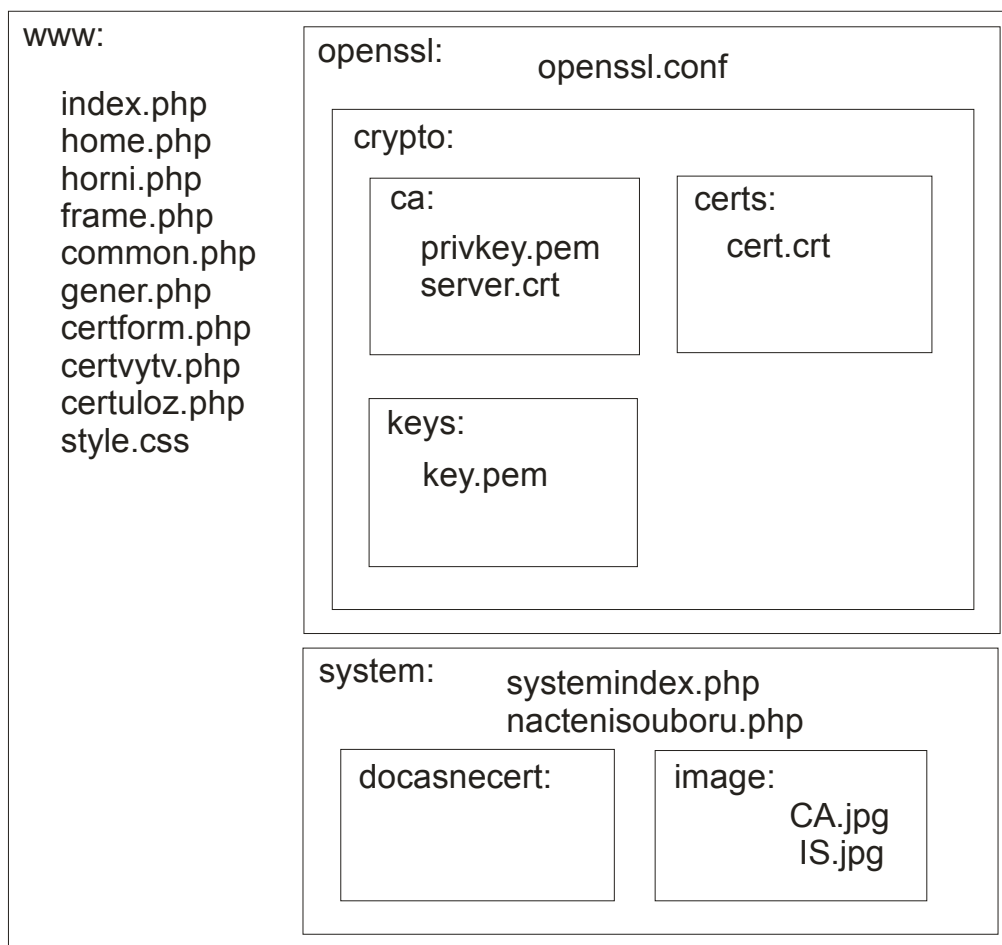
Po úspěšném uložení certifikátu do bezpečného úložiště je uživatel připraven pro vstup do informačního systému. Kliknutím na odkaz „Informační systém“ v levé části webových stránek se zobrazí přístupový formulář. Pomocí tlačítka se vybere certifikát z tajného úložiště. Následně je nutno zadat do položky „Jméno“ údaj, jež byl uveden před generováním certifikátu v položce „Jedinečné jméno“. Poslední kontrolní informace je



## 7.2 Struktura webového rozhraní

Struktura certifikační autority a informačního systému je tvořena soubory a složkami, jež jsou uvedeny na Obr. 22. Význam a popis jednotlivých souborů je následně stručně uveden v Tab. 3.

Uživatel po zadání IP adresy serveru do prohlížeče v operačním systému Microsoft Windows XP SP2 přijme certifikát, jež je definován ve složce konfigurace webového serveru Apache v souboru server.crt. Základní struktura webového rozhraní je dána v souboru index.php, následně hlavní uvítací stránka je tvořena home.php. Odkazy k certifikační autoritě a informačnímu systému jsou definovány v souboru frame.php. Pomocí souboru certform.php uživatel vyplňuje formulář pro vytvoření svého certifikátu pro přístup do informačního systému. Data vyplněná ve formuláři jsou předána souboru certvytv.php. Pravidla pro generování certifikátu jsou uvedena v souboru openssl.conf ve složce openssl. Při generování certifikátu je žádost o podání certifikátu podepsána certifikátem serveru server.crt a privátním klíčem privkey.pem ve složce ./openssl/crypto/ca. Následuje vyexportování certifikátu do souboru cert.crt ve složce ./openssl/crypto/certs. Současně je uložen privátní klíč uživatele do souboru key.pem ve složce ./openssl/crypto/keys. Veškerá uživatelská data jsou současně uložena o databáze, jež bude popsána později. Po vygenerování certifikátu je na konci stránky uveden odkaz na soubor certuloz.php, jež uloží certifikát na uživatelem uvedené místo. Po uložení certifikátu je soubor cert.crt smazán. Při tvorbě dalšího certifikátu je soubor key.pem přepsán novým certifikátem.



Obr.22: Struktura webového rozhraní

Tab. 3: Přehled a význam jednotlivých souborů ve webovém rozhraní

<b>www</b>	
index.php	Základní struktura webového rozhraní
home.php	Uvítací stránka
frame.php	Odkazy k certifikační autoritě a informačnímu systému
common.php	Základní globální funkce
gener.php	Přechod mezi formulářem CA a generátorem certifikátů CA
certform.php	Formulář pro zadání uživatelských údajů potřebných pro vytvoření uživatelského certifikátu
certvytv.php	Generátor uživatelských certifikátů
certuloz.php	Funkce pro uložení uživatelského certifikátu na určené místo
www/openssl	
openssl.conf	Konfigurační soubor OpenSSL
<b>www/openssl/crypto/ca</b>	
privkey.pem	Privátní klíč serveru a certifikační autority
server.crt	Kořenový certifikát serveru a certifikační autority
<b>www/openssl/crypto/certs</b>	
cert.crt	Dočasný uživatelský certifikát
<b>www/openssl/crypto/keys</b>	
key.pem	Dočasný soubor s privátním klíčem uživatele
<b>www/system</b>	
systemindex.php	Formulář pro přihlášení do informačního systému
nactenisouboru.php	Přístup k datům určeným uživateli

Jakmile získal uživatel od certifikační autority svůj osobní certifikát, přihlašuje se do informačního systému. Zadávání uživatelských údajů a výběr uživatele certifikátu definuje soubor systemindex.php ve složce /system. Následné načtení certifikátu, kontrolu údajů a certifikátu řeší soubor nactenisouboru.php.

Osobní údaje, které jsou uživatelem vyplňovány ve formuláři certifikační autority, jsou po vygenerování certifikátu uloženy do databáze „cert“. Současně je do databáze uložen vygenerovaný privátní klíč a certifikát. Pro ukládání výše zmíněných údajů je zavedena tabulka „certifikáty“ viz Obr. 23. Následně při přihlašování do informačního systému jsou přihlašovací údaje srovnávány s informacemi v databázi.

	Sloupec	Typ	Porovnávání	Vlastnosti	Nulový	Výchozí	Extra
<input type="checkbox"/>	jedinecnejmeno	mediumtext	latin2_czech_cs		Ne		
<input type="checkbox"/>	heslo	mediumtext	latin2_czech_cs		Ne		
<input type="checkbox"/>	certifikat	longtext	latin2_czech_cs		Ne		
<input type="checkbox"/>	hashcertifikatu	longtext	latin2_czech_cs		Ne		
<input type="checkbox"/>	privatekey	longtext	latin2_czech_cs		Ne		

Obr.23: Tabulka „certifikáty“

Po přihlášení uživatele do informačního systému dojde k uložení osobních údajů, IP adresy uživatele, přesného času a hash certifikátu do tabulky „pripojeni“ viz Obr. 24.



	Sloupec	Typ	Porovnávání	Vlastnosti	Nulový	Výchozí	Extra
<input type="checkbox"/>	jedinecnejmeno	text	latin2_czech_cs		Ne		
<input type="checkbox"/>	heslo	text	latin2_czech_cs		Ne		
<input type="checkbox"/>	cas	text	latin2_czech_cs		Ne		
<input type="checkbox"/>	ipadresa	text	latin2_czech_cs		Ne		
<input type="checkbox"/>	hashnactcert	text	latin2_czech_cs		Ne		

Obr.24: Tabulka „pripojení“

### 7.3 Použité funkce

V této kapitole bude uveden a podrobně rozepsán postup při generování certifikátu certifikační autoritou pomocí údajů zadaných uživatelem ve formuláři. Kompletní zdrojový kód je obsažen v souboru certvytv.php.

První fáze je tvořena generováním privátního klíče `$privkey = openssl_pkey_new()`. Proměnná `$privkey` je následně použita pro vytvoření žádosti o certifikát. Příkazem `$csr = openssl_csr_new($_REQUEST['dn'], $privkey)` je do proměnné `$csr` žádost uložena. Proměnná `$_REQUEST['dn']`, která je současně potřeba pro generování žádosti je získána jako pole array z formuláře, kde jdou vyplňovány uživatelské údaje. Další fáze spočívá v podpisu žádosti certifikační autoritou. Jsou pro to použity následující příkazy. První trojice příkazů se zabývá otevřením a načtením potřebných informací z certifikátu certifikační autority. Další trojice slouží k načtení privátního klíče certifikační autority. Poslední níže uvedený příkaz podepisuje žádost `$csr` pomocí `$cacert` a `$privkey`. Doba platnosti certifikátu je nastavena na 365 dní.

```
$fp=fopen("./openssl/crypto/ca/server.crt","r");
$cacert=fread($fp,8192);
fclose($fp);
$fp1=fopen("./openssl/crypto/ca/privkey.pem","r");

$privk=fread($fp1,8192);
fclose($fp1);
$privkey = openssl_get_privatekey($privk,$passphraseserver);

$sscert = openssl_csr_sign($csr, $cacert, $privkey, 365);
```

Následující fáze exportuje certifikát dle normy X.509. Slouží k tomu funkce `openssl_x509_export($sscert,$myCert)`. Certifikát je vyexportován do proměnné `$myCert`.

Podobně jako certifikát je pomocí příkazu `openssl_pkey_export($privkey,$myKey,$passPhrase)` vyexportován privátní klíč zašifrovaný pomocí `$privkey`, tedy privátního klíče certifikační autority. Dále proběhne pomocí níže uvedených příkazů uložení certifikátu a privátního klíče uživatele do dočasných souborů `cert.crt` resp. `key.pem` na serveru.

```
$certFile = "./openssl/crypto/certs/cert.crt";
$keyFile = "./openssl/crypto/keys/key.pem";

$fp = fopen($certFile, 'w');
fputs($fp, $myCert);
fclose($fp);

$fp = fopen($keyFile, 'w');
fputs($fp, $myKey);
fclose($fp);
```

Poslední fáze spočívá v uložení všech důležitých údajů do databáze „cert“. Proběhne uložení jedinečného jména, hesla, certifikátu, hashe certifikátu a privátního klíče uživatele.

```
$hashcertifikatu=sha1($myCert);
$passPhrase=sha1($passPhrase);
$db="cert";
$tb="certifikaty";
$spojeni=mysql_connect("localhost","root","");
mysql_select_db($db, $spojeni);
mysql_query("INSERT INTO $tb values ('$jedinecnejmeno','$pass
Phrase','$myCert','$hashcertifikatu','$myKey')", $spojeni);
```

## 7.4 Zachytávání komunikace pomocí síťového analyzátoru WireShark

Pro otestování bezpečnosti zabezpečených stránek je použit program WireShark 1.0. Pro názornou ukázkou detekce šifrovaných dat jsou uvedeny Obr. 25 a Obr. 26.

Na prvním obrázku je v horní části zobrazen průběh handshake SLL. Uživatel a server se pozdraví, a následně mezi uživatelem a serverem dojde k autentizaci a předání certifikátu. Zvýrazněný řádek v Obr. 25 zobrazuje vybraný paket. Prostřední část zobrazuje strukturu přenášeného paketu. Dolní část zobrazuje data celého paketu. Ze zobrazených dat je patrný přenos testovacích šifrovaných dat.

```

15 1.919895 192.168.30.1 192.168.30.145 SSL Client Hello
16 1.924442 192.168.30.145 192.168.30.1 SSLV3 Server Hello, Certificate, Server Hello Done
17 1.925249 192.168.30.1 192.168.30.145 SSLV3 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
18 1.928281 192.168.30.145 192.168.30.1 SSLV3 Change Cipher Spec, Encrypted Handshake Message
19 1.933053 192.168.30.1 192.168.30.145 SSLV3 Application Data

```

---

```

# Frame 19 (675 bytes on wire, 675 bytes captured)
# Ethernet II, Src: vmware_c0:00:08 (00:50:56:c0:00:08), Dst: vmware_1b:0f:bb (00:0c:29:1b:0f:bb)
# Internet Protocol, Src: 192.168.30.1 (192.168.30.1), Dst: 192.168.30.145 (192.168.30.145)
# Transmission Control Protocol, Src Port: event_listener (3017), Dst Port: https (443), Seq: 275, Ack: 828, Len: 621
# Secure Socket Layer
  # SSLv3 Record Layer: Application Data Protocol: http
    Content Type: Application Data (23)
    Version: SSL 3.0 (0x0300)
    Length: 616
    Encrypted Application Data: 1E7715D222BDEF8C7ACD8F1CC8135DED97BC50C312F7C854...

```

---

```

0000 00 0c 29 1b 0f bb 00 50 56 c0 00 08 08 00 45 00 ..)....P V.....E.
0010 02 95 f2 f3 40 00 80 06 47 8c c0 a8 1e 01 c0 a8 ....@... G.....
0020 1e 91 0b c9 01 bb 96 65 a6 8e 95 13 8c d0 50 18 .....e.....P.
0030 fc c4 12 bd 00 00 17 03 00 02 68 1e 77 15 d2 22 .....h.w...
0040 bd ef 8c 7a cd 8f 1c c8 13 5d ed 97 bc 50 c3 12 ..z....]...P.
0050 f7 c8 54 69 61 8c 0b 9d 35 76 3a 7b b9 ee ac 5c ..Tia...5VZ[...
0060 8a 63 e5 61 4d b7 9c 43 b2 49 e0 55 29 de e7 63 ..c.aM.C.I.U)...c
0070 2f ce f2 fa a9 81 c7 1f 09 73 e7 0c 55 3f 18 47 /......s.U?..G
0080 42 57 03 78 8d 54 5b 5b da 6d f3 9e b6 1d 1d de BW.x.T[.m.....
0090 f2 69 dc 19 53 54 57 d0 23 d3 15 06 b7 dd c2 72 ..i..STW.#.....r
00a0 88 03 77 74 51 ca de 72 72 7c ca 56 b0 3f d1 a8 |...F...P.AL...:
00b0 7c ef b1 f4 fb 46 94 ec 50 11 41 4c 3a 80 88 3a |%.....7.....
00c0 9c 3a 25 9d aa e7 cd f1 37 1e 0b f8 05 ae 60 ac ..*...s.i..9.
00d0 a7 f2 2a ce cf 3a d6 b9 0d 73 18 69 db 1e 39 91 ..z.i..5....0.B.
00e0 ac 7a da 69 fd de 35 e1 01 a4 f0 05 30 11 42 03 ...7..7....Q...
00f0 a5 1d 2e 37 9b ce 37 b5 fd ec 18 a3 0b 51 c3 d1 ...A.y...w..13X
0100 86 e0 1a 41 b2 19 79 a2 c9 a3 77 c9 c9 31 33 58 ...Xw.\.Z...A.
0110 f7 2e 07 58 57 c1 00 5c 08 5a aa 60 bc 86 5e e1 ...H.p...MY...;i...
0120 09 8e 1d 48 cd 70 11 07 4d 59 8e f8 3b 69 c3 d5 ...{....06....
0130 d7 fb 03 7b fc 0d 02 c2 82 0d 30 36 0e 93 97 f5 ...}....P91...
0140 84 b1 0c dd 29 d0 ec d4 b2 c7 50 39 6c 9c 06 92 .....-i.i>.<
0150 9e ca c0 1d e3 ba 2d dd 19 69 ef be 9d 3e fc 3c ...../.~4.06.
0160 96 df be f4 f3 2f 2c 15 fb 91 7e 34 f9 6f 36 1a ...F..I *n....
0170 81 b4 46 82 b2 7d 49 20 b4 2a 6e 15 81 b9 09 b4 .....&..5...$7.
0180 e1 8e 8d 0a a3 b4 ef 26 c9 35 ad fa 86 24 37 e8 ...#..28...h....
0190 8a 23 13 32 38 ad 81 d7 2e 68 cf c7 1e e2 ad f1 ..U.G.*.%.....
01a0 bc 55 82 ee 47 f5 2a f9 16 25 f3 f5 7f ef 9e e6 S...<2Mr 5.qt/.0
01b0 53 18 06 3c f5 32 4d 72 35 a6 71 74 2f 89 fb 30 @.e.\.F...gVP
01c0 e2 40 97 65 06 a8 5c f2 46 ad 80 b9 ad 67 5c 50 Nqw....0>.k...
01d0 4e 71 77 15 e4 1f d1 dc 30 3e 8b a1 6b 85 cc e0 ..8....V.../...w
01e0 bf a8 38 5f ae e5 cc 0b 56 1c f4 2f c3 c2 c6 77 V.....3...
01f0 56 17 93 e6 b2 0a 88 84 b1 db d5 c1 d5 d6 33 2e ..->2I..9R.I.<..
0200 b9 93 78 2d 32 49 f2 d9 39 52 05 21 08 3c d4 0a ..yr"3B".*T.C$.{.
0210 79 72 32 33 42 27 06 2a 54 bf 43 24 7f 19 7b a0 ..#.e.ks =.....
0220 94 09 23 fd 65 c0 6b 53 e8 3d 18 ff 82 ee da c9 ZG.z....?..4]
0230 7a 47 c3 7a 0c cb fb c9 c0 9e e0 3f 99 0e 34 6c 1...;...).E....
0240 31 02 17 ce 90 3b ff f1 a7 29 9f 45 fc 0e b9 94 .>@.k.k.Ti...[r.
0250 ad 3e 75 40 e8 d5 6b a9 54 69 85 a0 b7 5b 72 b8 .C.RM.On ...[.&
0260 ea 43 c2 72 4d 96 4f 6e 0d ca 3a fb 5b c2 8c 26 p.Cq....\...g.
0270 70 b1 43 71 fb c4 b2 f6 ca c3 d9 da 5c 81 67 00 ..).....N.I.X.I.
0280 6a c1 1c 0a a1 d9 bc 4e c7 fa 49 03 58 a5 7c c5 ..8...o...2...o..+
0290 fa 86 38 9c 14 6f ba 09 f6 32 c9 a6 4f b0 02 2b ...
02a0 f5 99 85

```

Obr.25: Zachytávání testovacích šifrovaných aplikačních dat

Na následujícím obrázku (Obr. 26) je uvedeno šifrované přihlášení do informačního systému. Při přihlašování dochází k přenosu identifikačních údajů, přesněji uživatelského jména a hesla, a certifikátu, pomocí něhož je přístup do systému mnohem bezpečnější.

8	0.014491	192.168.30.1	192.168.30.145	SSLv3	Application Data
9	0.023531	192.168.30.1	192.168.30.145	TCP	[TCP segment of a reassembled PDU]
10	0.023550	192.168.30.1	192.168.30.145	SSLv3	Application Data
11	0.023733	192.168.30.145	192.168.30.1	TCP	https > dvt-data [ACK] Seq=828 Ack=2773 win=64240 Len=0
12	0.214821	192.168.30.145	192.168.30.1	TCP	[TCP segment of a reassembled PDU]
13	0.214936	192.168.30.145	192.168.30.1	SSLv3	Application Data
14	0.214974	192.168.30.1	192.168.30.145	TCP	dvt-data > https [ACK] Seq=2773 Ack=3680 win=65535 Len=0
[+] Frame 13 (1446 bytes on wire, 1446 bytes captured)					
[+] Ethernet II, Src: Vmware_lb:0f:bb (00:0c:29:1b:0f:bb), Dst: vmware_c0:00:08 (00:50:56:c0:00:08)					
[+] Internet Protocol, Src: 192.168.30.145 (192.168.30.145), Dst: 192.168.30.1 (192.168.30.1)					
[+] Transmission Control Protocol, Src Port: https (443), Dst Port: dvt-data (3247), Seq: 2288, Ack: 2773, Len: 1392					
[+] [Reassembled TCP Segments (2852 bytes): #12(1460), #13(1392)]					
[+] Secure Socket Layer					
[+] SSLv3 Record Layer: Application Data Protocol: http					
Content Type: Application Data (23)					
Version: SSL 3.0 (0x0300)					
Length: 2847					
Encrypted Application Data: 9AE929E0720F7217D429B02EAADC3B9CF2FBF45AD33C849D...					
0000	00 50 56 c0 00 08 00 0c	29 1b 0f bb 08 00 45 00	.PV.....).....E.		
0010	05 98 01 61 40 00 80 06	36 1c c0 a8 1e 91 c0 a8	...a@... 6.....		
0020	1e 01 01 bb 0c af 32 9d	8d 19 3f cd 25 9f 50 18	.....2. .?%.P.		
0030	fa f0 64 71 00 00 d3 39	0f 4f d3 f4 c5 d7 24 24	..dq...9 .0.....\$		
0040	66 ae ba 59 06 42 ca ee	68 66 bd 64 3e 64 36 49	f..Y.B... hf.d>d6I		
0050	15 e4 e1 ba a0 f0 2a 66	97 00 7e ee f8 5e e3 9c	.....*f ..~..A..		
0060	bb 06 66 8a a0 b1 02 22	1f cf b6 26 0d e9 38 56	..f...." ...&..8V		
0070	b8 be ea 0e 09 2b ae c9	57 f8 de db 53 46 cb 93	.....+. w...5F..		
0080	e7 f7 3e 00 36 95 d2 59	f8 86 70 aa da b7 a4 e2	..>.6..Y ..p....		
0090	b7 41 cd f6 f7 e6 41 e0	72 4f 93 d9 9a 69 ff c6	.A...A. ro...i..		
00a0	47 1e 0d 47 19 1a a6 80	f1 62 40 56 c6 28 d9 c0	G..G... .b@v.(.		
00b0	b4 34 91 97 4a 5a e3 86	de f5 98 28 0a 73 06 47	.4..JZ.. ..(.s.G		
00c0	10 54 6e 03 03 4c 58 64	6e 0b db 31 07 93 b2 42	.Tn..LXd n...1...B		
00d0	c6 9e 39 e5 69 e7 39 dd	75 f1 bc c6 13 00 a2 cd	..9.i.9. u.....		
00e0	cf 5c c4 81 c1 15 6f a9	01 aa 69 8a 1a 41 78 1c	.\....o. .i..Ax.		
00f0	a7 09 87 0d 86 6d 97 99	ad 4f 44 c0 dd 17 92 39	....m... .OD...9		
0100	7c 93 c3 60 0b 78 e0 e9	59 07 ca 60 60 a7 a3 4a	. .x... Y... .J		
0110	5a 72 09 77 16 65 94 e4	6d f4 d6 51 70 0a 17 59	Zr.w.e. m. Qp..Y		
0120	5b c8 0b 18 47 e3 57 37	2a 61 07 cd b4 7d 7b 55	[...G.w7 *a...}U		
0130	49 14 97 7c 15 7a 5f d3	77 f1 ab eb 7d 62 0f c2	I.. .z. w...}b..		
0140	02 59 82 57 88 da 2f d6	b7 22 c3 4b 6a a9 4b 1b	.Y.w.../...Kj.k.		
0150	62 16 0f 60 58 7b f1 65	af 32 fa 8e 92 21 2c 0e	b..X{.e .2...!..		
0160	c1 8d e4 27 da 14 3a bd	cd 48 7e b1 d1 1c 3b 90	.....: .H...;.		
0170	cc fa 2c 1c 53 92 8f df	18 1d 30 8f 5a 5c b2 cd	....S... ..0.Z\..		
0180	65 71 38 5d a0 ad a2 86	75 70 5c 69 1f 67 47 71	eq8].... up\i.ggq		
0190	73 77 77 4f 6a fc c0 0f	61 1d 56 cb 0c 75 2c b6	swwoj... a.v..u..		
01a0	8c 20 7c 53 20 e8 be 3d	11 a3 ae 77 37 74 6c d3	.  S ... = ...w7t1.		
01b0	44 0c 81 b1 8d 4b f2 41	db 86 f4 b0 04 64 20 be	D...K.A .....d .		
01c0	1d 2e 84 fb b2 d2 15 16	ad 3a 78 02 86 36 e9 a7	.....:x..6..		
01d0	13 b6 a2 83 a6 ae 82 c2	45 a8 88 31 5c f5 0e 54	..... E..1\..T		
01e0	02 2c be ce 8f be 9b 17	2d f2 5a 63 aa 28 17 62	..... -.Zc.(.b		
01f0	40 29 f8 c9 86 f9 51 ba	4f 2a 34 f2 ec c3 5c f7	@)....Q. o*4... \.		
0200	40 eb 2e 78 d3 2b fc 47	78 b1 2e 70 c0 4a c3 9a	@..x.+g x..p.j..		
0210	73 10 cb fc 08 61 0d 45	65 0f 9d 8b a3 5d 57 39	s...a.E e...jw9		
0220	cd 47 dc da 6f 29 02 d6	36 8d c6 da 60 2f ae 4d	.G..o).. 6.../M		
0230	2a c3 4e 7a a1 9b 7e bf	3e 7d 0d c9 91 a4 ab fd	*.NZ...~ >}.....		
0240	02 01 1d 00 bc d6 87 ea	cd 47 5e 2d 09 a2 37 0d	..... .GA...7.		
0250	1f e2 fd 68 2d 19 77 d8	3a 8e 93 48 55 5d 32 9d	...h-.w. ...HU]2.		

Obr.26: Zachycení šifrovaných dat při přihlašování do informačního systému

Pomocí síťového analyzátoru Wireshark bylo otestováno zabezpečení dat při komunikaci mezi uživatelem a serverem. Veškerá data jsou bezpečně šifrována pomocí protokolu SSL v3, není tedy možno je dešifrovat bez znalosti potřebného dešifrovacího klíče.

## 8 Závěr

Cílem diplomové práce bylo zrealizování certifikační autority a vytvoření přístupu do informačního systému při využití bezpečného protokolu SSL. Součástí práce je vytvoření uceleného přehledu postupů Infrastruktury veřejných klíčů, a dále podrobný rozbor problematiky protokolu SSL, respektive jeho možnosti aplikace.

První kapitola se zabývala historií a termínem Infrastruktura veřejných klíčů. Byly uvedeny principy šifrovacích algoritmů a hashovacích funkcí, jež jsou využívány v certifikátech respektive v digitálním podpisu.

Ve druhé kapitole byl objasněn podrobný princip digitálního podpisu, nejčastěji používaná šifrování a dále způsob a smysl využití.

Třetí kapitola se věnovala podrobnému rozboru digitálních certifikátů. Byla zde rozebrána struktura certifikátu a jeho životní cyklus, kterým při své existenci prochází. Podstatná část kapitoly byla tvořena popisem práce certifikační autority, její strukturou a funkcemi jednotlivých bloků, a příkladem využití digitálního certifikátu.

Následující kapitola se zabývala tvorbou kořenové certifikační autority v operačním systému Microsoft Windows 2003 Server, jejím nastavením a konfigurací pro vyhotovení digitálních certifikátů. Dále byl digitální certifikát vyhotoven a nainstalován do webového prohlížeče pro okamžitou možnost využití.

Pátá kapitola se zabývala rozбором problematiky protokolu SSL. Je zde podrobně rozebrána jeho struktura a procesy, které jsou při komunikaci pomocí uvedeného protokolu vykonány.

Další kapitola seznamuje čtenáře s nástroji a programy, které byly využity pro vytvoření certifikační autority a informačního systému. Jsou zde uvedeny postupy, jak nakonfigurovat využitý software pro správnou funkci vytvořené aplikace.

Poslední kapitola rozebírá vyvinutou webovou aplikaci, postup při jejím použití a uvádí základní využití funkce. Následně je aplikace z hlediska bezpečnosti otestována.

Zabezpečení webových aplikací pomocí protokolu SSL je v současnosti nahrazeno bezpečnostním protokolem TLS verze 1.0. Tento protokol se od SSL verze 3.0 liší jen v několika rozdílech, především v podpoře nejnovějších technologií. Protokol TLS má velkou perspektivu do budoucna. Dle předpokladů bude nadále vyvíjen a jeho využití pro bezpečnou komunikaci nadále poroste.

## Literatura

- [1] NÁDENÍČEK, Petr. Pravdy o elektronickém podpisu a šifrování (10) - systémy PKI aneb netřeba kanónu na vrabce [online]. 2003 [cit. 2007-11-15]. Dostupný z WWW: <<http://www.svetsiti.cz/view.asp?rubrika=Tutorialy&temaID=244&clanekID=255>>.
- [2] DOSTÁLEK, Libor, VOHNOUTOVÁ, Marta. Velký průvodce infrastrukturou PKI a technologií elektronického podpisu. 1. autoriz. vyd. Brno : Computer Press, 2006. 534 s. ISBN 80-251-0828-7.
- [3] BURDA, Karel. Bezpečnost informačních systémů. Brno : [s.n.], 2005. 104 s.
- [4] DOSTÁLEK, Libor. Tutoriál PKI [online]. 2002 [cit. 2007-11-20]. Dostupný z WWW: <<http://www.cpress.cz/knihy/tcp-ip-bezp/Tutorial/Tutorial.htm>>.
- [5] KMENT, Vojtěch. Hašovací funkce: Jak se odolává hackerům [online]. 2005 [cit. 2007-12-05]. Dostupný z WWW: <<http://www.lupa.cz/clanky/hasovaci-funkce-jak-se-odolava-hackerum>>.
- [6] KAČMAŘÍK, Vojtěch. Výuková podpora předmětu Internetové technologie [online]. 2006 [cit. 2007-12-07]. Dostupný z WWW: <<http://homel.vsb.cz/~kac061/#kapitola16>>.
- [7] RÚŽIČKA, Pavel. Bezpečnost především : použití SSL [online]. 2002 , 6. června. 2002 [cit. 2008-03-13]. Dostupný z WWW: <<http://interval.cz/clanky/bezpecnost-predevsim-pouziti-ssl/>>.
- [8] ODVÁRKA, Petr. SSL protokol : Princip a přínosy [online]. 2002 , 25. dubna 2002 [cit. 2008-03-20]. Dostupný z WWW: <<http://www.svetsiti.cz/view.asp?rubrika=Tutorialy&temaID=171&clanekID=187>>.
- [9] ODVÁRKA, Petr. SSL protokol : Používané šifry, spojení a jeho struktura [online]. 2002 , 30. dubna 2002 [cit. 2008-03-20]. Dostupný z WWW: <<http://www.svetsiti.cz/view.asp?rubrika=Tutorialy&temaID=171&clanekID=185>>.
- [10] ODVÁRKA, Petr. SSL protokol : SSL Handshake Protocol [online]. 2002 , 9. května 2002 [cit. 2008-03-20]. Dostupný z WWW: <<http://www.svetsiti.cz/view.asp?rubrika=Tutorialy&temaID=171&clanekID=183>>.
- [11] SSL tutorial [online]. 2004 , 14. 7. 2004 [cit. 2008-03-15]. Dostupný z WWW: <<http://www.eventhelix.com/RealtimeMantra/Networking/SSL.pdf>>.
- [12] ODVÁRKA, Petr. SSL protokol : Change Cipher Spec Protocol a Alert Protocol [online]. 2002 , 16. května 2002 [cit. 2008-03-20]. Dostupný z WWW: <<http://www.svetsiti.cz/view.asp?rubrika=Tutorialy&temaID=171&clanekID=182>>.
- [13] ODVÁRKA, Petr. SSL protokol : SSL Record Protocol [online]. 2002 , 21. května 2002 [cit. 2008-03-20]. Dostupný z WWW: <<http://www.svetsiti.cz/view.asp?rubrika=Tutorialy&temaID=171&clanekID=180>>.

- [14] BÁRÁNY, Balázs. The Apache + SSL on Win32 HOWTO [online]. 2007 , 2007-12-22 [cit. 2008-03-22]. Dostupný z WWW: <<http://tud.at/programm/apache-ssl-win32-howto.php3>>.

## Přílohy

### Generování certifikátu serveru

```
C:\openssl>openssl req -config openssl.cnf -new -out server.req
Loading 'screen' into random state - done
Generating a 1024 bit RSA private key
.....+++++
.....+++++
writing new private key to 'privkey.pem'
Enter PEM pass phrase:
Verifying - 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) []:cz
State or Province Name (full name) []:Czech Republic
Locality Name (eg, city) []:Brno
Organization Name (eg, company) []:VUT
Organizational Unit Name (eg, section) []:FEKT
Common Name (eg, your websites domain name) []:localhost
Email Address []:martin.trojak@centrum.cz

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:

C:\openssl>openssl rsa -in privkey.pem -out server.key
Enter pass phrase for privkey.pem:
writing RSA key

C:\openssl>openssl x509 -in server.req -out server.cer -req
-signkey server.key -days 365
Loading 'screen' into random state - done
Signature ok
subject=/C=cz/ST=CzechRepublic/L=Brno/O=VUT/OU=FEKT/CN=localhost/e
mailAddress=martin.trojak@centrum.cz
Getting Private key
```