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

DIGITÁLNÍ CERTIFIKÁTY A CERTIFIKAČNÍ AUTORITY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE Bc. ONDŘEJ LEPA
AUTHOR

BRNO 2014

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

DIGITÁLNÍ CERTIFIKÁTY A CERTIFIKAČNÍ AUTORITA

DIGITAL CERTIFICATES AND CERTIFICATION AUTHORITY

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE **Bc. ONDŘEJ LEPA**
AUTHOR
VEDOUCÍ PRÁCE **Ing. VLASTIMIL ČLUPEK**
SUPERVISOR

BRNO 2014

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Ondřej Lepa ***ID:*** 115218
Ročník: 2 ***Akademický rok:*** 2013/2014

NÁZEV TÉMATU:

Digitální certifikáty a certifikační autority

POKYNY PRO VYPRACOVÁNÍ:

Proveďte souhrnnou bezpečnostní analýzu problematiky digitálních certifikátů a certifikačních autorit. Zaměřte se na jejich princip činnosti, matematický základ a kryptografickou sílu. Popište exitující útoky na certifikáty a certifikační autority a způsoby ochran proti nim. Navrhněte, vytvořte a analyzujte zabezpečené spojení komunikujících stran využívající ke své činnosti certifikáty a certifikační autority.

DOPORUČENÁ LITERATURA:

[1] MENEZES, Alfred J.; OORSCHOT, Paul C. van; VANSTONE, Scott A. Handbook of AppliedCryptography. USA: CRC Press, 1996. 816 s. ISBN 0-8493-8523-7.

[2] PIPER, F.; MURPHY, S. Kryptografie. 1. vyd. v českém jazyce. Překlad Pavel Mondschein. Praha: Dokořán, 2006, 157 s. ISBN 80-736-3074-5.

[3] BUDIŠ, P. Elektronický podpis a jeho aplikace v praxi. 1. vyd. Olomouc: ANAG, 2008, 157 s. ISBN978-80-7263-465-1.

Termín zadání: 10.2.2014 ***Termín odevzdání:*** 30.5.2014

Vedoucí práce: Ing. Vlastimil Člupek

Konzultanti diplomové práce:

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 části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato diplomová práce se zabývá problematikou certifikátů a certifikačních autorit, certifikační cestou PKI a způsobem zabezpečení. Dále se práce zabývá strukturou digitálního certifikátu samotného a možností zneužití informací v něm uvedených, stejně tak možností zneužití certifikátu třetí stranou a proklamací nedůvěryhodného certifikátu do klientova systému.

KLÍČOVÁ SLOVA

certifikát, certifikační autority, bezpečnost, PKI, asymetrická kryptografie, šifrování, ověřovaní identity, digitální podpis, openSSL

ABSTRACT

This diploma thesis deals with certification and certification authorities, certification path PKI and principles of its validation and security. Also deals with structure of certificate itself and possible misuse of included information. Moreover, possibility of misues ot third party certificates and proclamation of untrusted certificate into clients systém.

KEYWORDS

certificates, certification authorities, security, PKI, asymmetric cryptography, cyphers, authentisation, digital signature, openSSL

LEPA, O. *Digitální certifikáty a certifikační autority.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2014. 91 s. Vedoucí diplomové práce Ing. Vlastimil Člupek.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovo práci na téma Certifikáty a certifikační autorita jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury umístěné na konci práce. Jako autor uvedené semestrální 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/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne ....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce Ing. Vlastimilu Člupkovi za pedagogickou a odbornou pomoc a celkovou podporu při tvorbě semestrální práce.

V Brně dne .....

.....
(podpis autora)

SIX research centre
sensor, information and communication systems

Faculty of Electrical Engineering
and Communication

Brno University of Technology
Technicka 12, CZ-61600 Brno, Czechia

http://www.six.feec.vutbr.cz

Výzkum popsaný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

2007–13
OP Výzkum a vývoj pro inovace

Obsah

Úvod

V dnešní době je třeba neustále brát na zřetel možnosti internetu. Nejen pomocí sociálních sítí, a jejich služeb, již není těžké sledovat pohyb a chování lidí na internetu. Stačí se správně „zeptat“ vyhledávače. Ve všech těchto případech je ale na vině především člověk sám. Většina služeb umožňuje uchovávat informace v soukromí a tím určit míru soukromí informací. Je to ovšem poměrně paradoxní nastavení u sociálních sítí, které jsou postavené na otevřeném sdílení informací mezi uživateli. Bohužel jsou data zpracovávána i informační databází, která hledá klíčová data.

Problém nastává, pokud se jedná o důležité informace a ne jen o sdílení informací na zdi sociální sítě, kterou by svět vidět neměl. Například při předávání soukromých informací mezi uživatelem a bankou, pojišťovnou, při ověřování identity například na serveru pracovního úřadu, při přístupu k datům na firemním serveru. Aplikací, u kterých je potřeba dodržet značnou míru zabezpečení, je spousta.

V takovém případě je potřeba myslet na lepší ochranu než „pouze“ heslo a uživatelské jméno. Potenciální útočník by mohl zfalšovat vstupní stránku, získat tak potřebné informace (právě uživatelské jméno a heslo) a dále je používat podle vlastního uvážení. Proto bylo v informatice zavedeno několik druhů ochrany dat a komunikace. V této práci se budu zabývat problematikou Certifikátů a Certifikačních autorit, které se využívají při autentizaci jak uživatele žádajícího o přístup, tak tázané stránky/serveru. Eliminuje se tím tak možnost podvrhnutí falešné vstupní stránky i zneužití přihlašovacích údajů. Díky nim můžeme zavádět pojmy jako elektronický či digitální podpis, autentizace uživatele a v některých zemích Evropské unie je tak dokonce používat namísto rukou psaného podpisu při podpisu smlouvy či dokumentů.

V této práci bude rozebrána struktura uživatelského certifikátu, jeho ověřování ze strany certifikační autority a slabé stránky tohoto systému, který by mohli vést ke zfalšování, podvržení, nebo jinému způsobu kompromitace systému certifikátů.

Cíle práce, metody a postupy zpracování

Primárním cílem práce této práce je komplexní popis a rozbor problematiky digitálních certifikátů a certifikačních autorit.

Pro dosažení hlavního cíle jsem práci rozdělil do tří celků. První celek je věnován teoretické stránce problematiky s důrazem na způsoby šifrování a možné principy zabezpečování dat. Dále také informativní strukturu certifikátů a způsob jeho získávání a ověřování v rámci certifikační cesty *PKI*.

Druhý celek práce se věnuje praktickému rozboru problematiky s realizací uzavřeného certifikačního systému a prací s certifikátem samotným pod platformou Microsoft Windows. V této části je rozebrán i způsob uložení certifikátů v systému Windows. Obsahem posledního celku práce je především sumarizace bezpečnosti a poukázání na slabé stránky systému digitálních certifikátů a certifikačních autorit.

1. Základní způsoby zabezpečení komunikace

V internetové bezpečnosti je několik základních způsobů, jak chránit sdílená data proti napadnutí, ukradení nebo nežádoucí změně třetí stranou. Například nejjednodušší způsob, jak bezpečně ověřit integritu dat, resp. zjistit, jestli zpráva nebyla oproti originálu pozměněna, je tzv. hash. Ten ověřuje, jestli nebyla obsah zprávy nějak pozměněn; například při přenosu souboru obsahující bankovní informace je potřeba zaručit, aby nikdo nezměnil číslo účtu ve svůj prospěch. Hashovací funkce je jednosměrná matematická funkce, na jejíž vstup je přivedena zpráva libovolné délky a na jejímž výstupu vznikne otisk zprávy (hash) definované délky. Jelikož je funkce jednosměrná, tak by z hashe neměla jít získat původní zpráva, tedy zajišťuje soukromí hashované zprávy. Hash hesla se používá při ověřování z důvodu ochrany hesla samotného. Nepřenáší se tedy otevřený řetězec znaků, ale pouze otisk hesla a ten se porovnává s otiskem v databázi. Pro zabezpečení proti odposlechu otevřeného hesla, nebo jeho hashe, je tedy potřeba zprávy šifrovat. Věda, která se zabývající šifrováním, a obecně šiframi, se nazývá kryptografie. Podle způsobu šifrování a dešifrování zpráv, resp. způsobu zabezpečení, se kryptografie rozlišuje na symetrickou a asymetrickou kryptografii.

Kryptografie je nauka o metodách utajování významu zprávy, pomocí převodu do podoby, která je čitelná jen se speciální znalostí. První zašifrování zprávy pochází už z doby před naším letopočtem. Do historie kryptografie se zapsal i Julius Caesar vynalezením šifry, která je po něm pojmenována jako Caesarova šifra. Období kryptografie lze dělit do dvou částí:

- Klasická kryptografie

  Tato etapa trvala do poloviny 20. stol. Vyznačovala se především jednoduchým způsobem šifrování bez potřeby složitých pomůcek. Během 1. poloviny 20. stol. začaly vznikat sofistikované přístroje, umožňující složité postupy při šifrování.

- Moderní kryptografie

  K šifrování se zpravidla nepoužívají žádné zvlášť vytvářené přístroje, ale klasické počítače vybavené specializovaným programem. Moderní kryptografie využívá matematické funkce zabývající se především složitou vyčíslitelností některých matematických problémů.

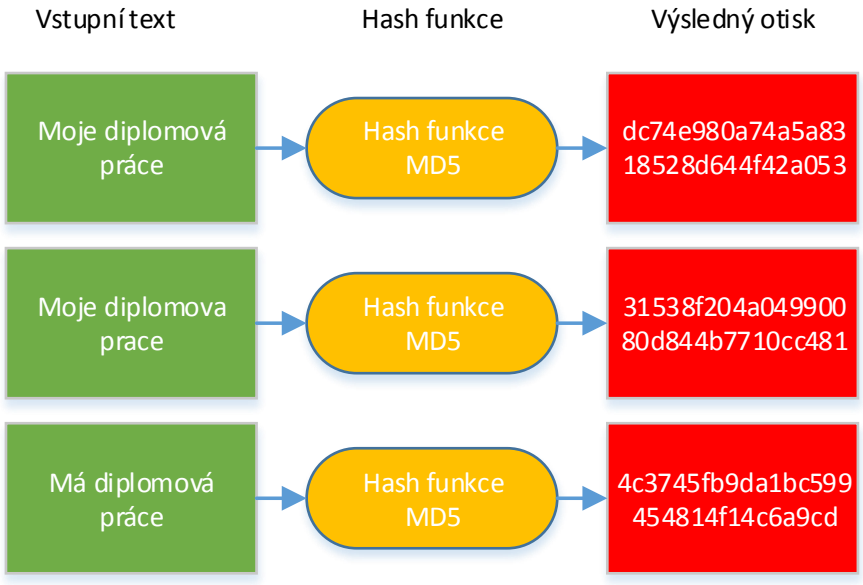
Další důležitým pojmem je autentizace. Autentizace slouží k jednoznačnému určení uživatele či entity. Cílem je zajistit, že systém přesně ví, s jakým uživatelem, či entitou, komunikuje. Například při bankovních operacích je kritické, aby byla ověřena identita původce zprávy nebo uživatele služeb. Autentizace se dá většinou zaručit pomocí něčeho,

co uživatel zná (například heslo, PIN); co má (čipová karta) nebo čím je (otisk prstu, snímek oční sítnice).

1.1. Hash

Jak již bylo řečeno, hash je nejspíš nejzákladnější a nejjednodušší, zároveň však asi nejdůležitější funkcí v počítačové bezpečnosti. Jedná se o digitální otisk zprávy vytvořený pomocí jednosměrných matematických funkcí. Jednosměrné funkce jsou algoritmy, které nejsou výpočetně příliš náročné, je však velice výpočetně náročné k výsledku najít původní text. Hashovací funkce z libovolně dlouhého textu, nebo řetězce, vytvoří krátký řetězec konstantní délky. Výsledný řetězec maximálně charakterizuje původní text a i minimální změna v původním řetězci způsobí velké změny v hash otisku zprávy. Nejedná se tedy pouze o zabezpečení proti změně, ale zároveň je použitelné pro ověření integrity zprávy. Je známo, že přenos informací není vždy 100% ani při kabelovém spojení a přenos hashe na konci přenosu souboru je asi nejspolehlivější kontrola integrity.

Typická velikost hashe pro nejpoužívanější, a zároveň nejstarší, algoritmy MD-5 a SHA-1 je 16B, resp. 20B, avšak velikost otisku se zvyšuje v závislosti na použitém algoritmu. Obecně platí, že čím delší je otisk, tím složitější funkce se za ním skrývá. Algoritmy MD-5 a SHA-1 se dnes již nedoporučuje používat v případě opravdu soukromých dat, jelikož jsou považovány za slabé a existují postupy, pomocí kterých lze hash prolomit. [1] Nutno podotknout, že prolomení je časově velmi náročné a neznamená zjištění původní zprávy, nýbrž sestavení takové náhradní zprávy, která má stejný otisk.



Obrázek 1-1 Viditelná změna otisku při malé změně vstupu

K tomu je možné použít tak zvané „rainbow tables“, což je v podstatě databáze předem vypočítaných hash otisků v různých kombinacích písmen a číslic. Například tabulka vytvořená pro funkci MD-5, která je sestavena z kombinací malý i velká písmen a číslic, při hesle o délce 1-8 znaků, má velikost 160 GB. Udávaná úspešnost je pak 99.9% [2]
Z výše zmíněného popisu tedy lze definovat nejdůležitější vlastnosti kvalitní hash funkce:

- **Jednosměrnost:**
  Každá hash funkce musí být jednosměrná tak, že k ní neexistuje inverzní algoritmus. K otisku nelze v časově omezeném úseku, jednoznačně najít text, z kterého byl tento otisk vypočítán. Například certifikáty ke kvalifikovaným elektronickým podpisům jsou zpravidla vydávány s platností omezenou na rok či dva. To už je časově omezený úsek.
- **Bezkoliznost (*slabá)*:**
  V reálném časovém úseku nesmí být možné k jednomu textu, u kterého známe i otisk, nalézt druhý text, který bude mít stejný otisk. To znamená, že pro dané *x* nelze nalézt druhý argument *x ‘* ve smyslu
  $$x' \neq x$$
  přičemž by platilo
  $$\mathrm{h}(x) = \mathrm{h}(x') = \mathrm{y}$$
- **Bezkoliznost (*silná*):**
  Také by nemělo být možno v reálně krátkém čase nalézt jakékoliv dva různé texty odpovídající stejnému hash otisku. Od slabé bezkoliznosti se tedy liší tím, že se zde hodnota *x* volí libovolně. To také znamená, že v tomto případě si útočník může volit i x i x' s jediným cílem, aby se obě *x* pomocí hashovací funkce otiskly na jednu hodnotu. To předpokládá, že útočník je v takové pozici, že může ovlivnit vstupní hodnotu hash funkce.

Ideální funkce musí disponovat všemi třemi vlastnostmi. V posledních letech se ovšem objevují nové postupy, pomocí nichž lze pro některé tradiční funkce spočítat kolize. Neboli systematicky najít dva různé vstupní řetězce vedoucí ke stejnému otisku (např. výše zmíněné Rainbow tables). V důsledku toho jsou tyto funkce nedostatečně zabezpečené proti falšování podpisů nebo padělání digitálních certifikátů. Rodina hash algoritmů se tímto rozrůstá o další variace a upravené verze. Jde vidět, že trend se drží hlavně zvyšování výstupního řetězce znaků, který způsobuje mnohem větší výpočetní nároky na nalezení kolizí.

Je ale vidět, že pokud je znám postup (např. u SHA algoritmu) není pouhé zvyšování výstupního slova dostačující. Teoreticky je možné pomocí známého inverzního algoritmu, při snížení počtu průchodů SHA algoritmu, získat odpovídající řetězec. To se ale v praxi a při plném využití doposud nepodařilo, a tak funkce z rodiny SHA-2 (SHA-256/224 a SHA-512/384) stále zůstávají bezpečné. Dá se tedy předpokládat, že rodina SHA-3 bude, jako nejmladší algoritmus, sloužit jako potenciální nástupce, nebo alternativa v případě prolomení SHA-2.[3]

Hash funkce	Velikost výstupu (bit)	Nalezená kolize
MD2	128	ANO
MD4	128	ANO
MD5	128	ANO
SHA-0	160	ANO
SHA-1	160	ANO
SHA-2	224	TEORETICKY
SHA-2	256	TEORETICKY
SHA-2	384	TEORETICKY
SHA-2	512	TEORETICKY
SHA-3	224	NE
SHA-3	256	NE
SHA-3	384	NE
SHA-3	512	NE
Tiger	128	ANO
Tiger	160	ANO
Tiger	192	ANO
Whirlpool	512	ANO

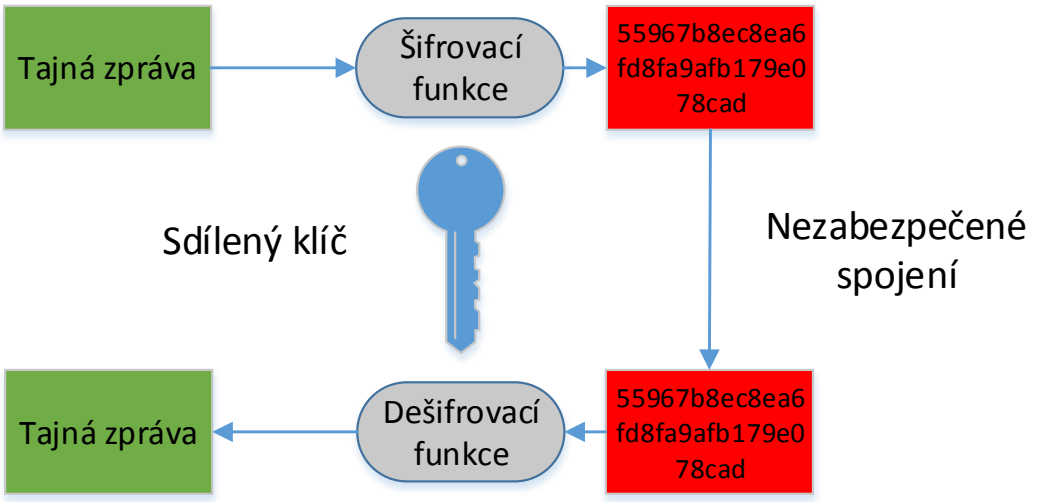
Obrázek 1-2 Používané hash funkce [4]

1.2. Šifrování

Ochrana dat pomocí šifrování je dnes běžný způsob zabezpečení elektronických dat. Šifrování umožňuje chránit soukromí vlastníka dat. Principem šifrování je logická operace provedená podle pravidel šifrovací funkce. Většinou se jedná o logické operace součtu, XOR a nebo substituce. Tímto je zpráva převedena do nečitelné nebo nesmyslné formy, jež zabraňuje svévolnému rozluštění. Šifrování také zabezpečuje nedotknutelnost zprávy během přenosu. Kdokoliv se pokusí pozměnit přenášenou zprávu, způsobí znatelné změny struktury a integrity, tudíž je změna lehce odhalitelná.

1.2.1. Symetrická kryptografie

Princip symetrické kryptografie je ve své podstatě velmi jednoduchý. Celý systém si lze představit tak, že uživatel chce odeslat data, například textový soubor s číslem účtu, proto tyto data zabezpečí pomocí jednoduchého hesla. Každý kdo tyto data při přenosu zachytí, zjistí, že data jsou zabezpečena a tudíž nepoužitelná. Jediný, kdo se k obsahu dat dokáže dostat je uživatel, který zná heslo. Tím vyvstává nejpalčivější otázka symetrické kryptografie, a to zabezpečení přenosu hesla u symetrické kryptografie. Všechny komunikující strany, resp. strany zahrnuté do kýžené chráněné komunikace si musí předem zvolit takové heslo, které bude všem známé. Tento klíč, tvořený například kombinací písmen a čísel (pro větší bezpečnost), se tak přenáší velmi obtížně. Heslo je tedy vhodné nesdílet, ale vytvořit jednorázově podle daných pravidel. Například kombinací prvních 4 písmen názvu měsíce a 4 číslic aktuálního roku. Takto vytvořený klíč je bezpečný z matematického hlediska, ale nebezpečný z hlediska logiky. I naprostý amatér při prolamování hesla nejprve vyzkouší známé údaje, např. jméno, příjmení atd. Někoho tak může napadnout i tato kombinace.



Obrázek 1-3 Princip symetrické kryptografie

Lze tedy logicky říci, že symetrická kryptografie je velmi náchylná na prozrazené, nebo lehce odhadnutelné, heslo. Dalším faktorem, který hraje velkou roli, je v podstatě nepřítomnost jakékoliv autentizace. Pokud komunikace probíhá pouze mezi dvěma účastníky, dá se předpokládat, že ten, kdo zná heslo je oprávněn. Tato premisa však při prozrazení hesla padá a protější osobě se potom již nedá věřit.

Symetrická kryptografie je, pro svou výpočetní nenáročnost a rychlost, velmi vhodná pro časově náročné spojení, které upřednostňuje kvalitu nad špičkovou bezpečností. Pokud bude komunikace sestavena na kratší časový úsek, nestihne potenciální útočník ani analyzovat

dostatečné množství dat, aby se mohl pokusit heslo prolomit a narušit tak bezpečnost komunikace mezi všemi účastníky komunikace.

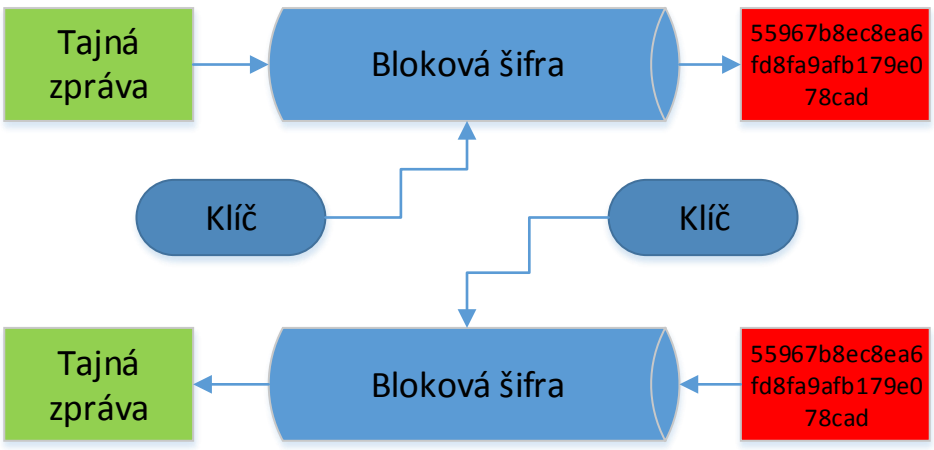
Bloková šifra

Je to typ symetrické šifry, která pracuje s jednotlivými bloky pevně dané délky. Pokud bude původní zpráva větší, rozdělí se do *x* bloků stejné délky, ta je dána šifrou, a na zbylé místo v bloku (pokud délka zprávy není přímo dělitelná délkou bloku) je použita výplň. Každý blok zvlášť je potom šifrován pomocí tajného klíče. Dešifrování potom probíhá obdobným způsobem, pouze v opačném směru.

Šifrování pomocí blokových šifer má ale jednu zásadní nevýhodu a tou je neustálé používání stejného klíče k zašifrování jednotlivých bloků. Tím vzrůstá možnost „odchycení" klíčovacího algoritmu a získání klíče. Proto se u blokových šifer používá mód, který zahrnutím obsahu předchozího bloku do bloku následujícího zajišťuje, že nelze přehazovat jednotlivé šifrované bloky.

DES

Zkratka znamená Data Encryption Standard a označuje šifru zveřejněnou roku 1977, standardizovanou v roce 1979. V současnosti je tato šifra považována za nespolehlivou, protože používá klíč pouze o délce 64 bitů, z toho 8b je kontrolních a 56b efektivních. Znamená to tedy 56 bitů = 2^{56} = 72 057 594 037 927 936 možností. Šifra DES je relativně jednoduše prolomitelná za využití metody hrubé síly a specializovaného hardware (např. DES Cracker [5]). Navíc algoritmus obsahuje slabiny, které dále snižují bezpečnost šifry, takže s pomocí distribuovaných výpočtů (např. botnet [6] nebo legální cestou pomocí např. www.*distributed.net*), může být prolomen v čase kratším než 24 hodin. Přesto je považována za relativně bezpečný ve formě Triple DES, na který jsou vyvinuty pouze teoretické útoky.



Obrázek 1-4 Princip šifrování a dešifrování blokovou šifrou

3DES

3DES neboli Triple Data Encryption Standard je oproti předchůdci DES založen na trojí aplikaci šifrování DES. Tím se zvyšuje odolnost vůči útokům hrubou silou, aniž by bylo potřeba vytvořit úplně nový algoritmus, za cenu pomalejšího šifrování. Místo jednoho klíče je použit balík klíčů skládající se ze 3DES klíčů K_1, K_2 a K_3. Každý z nich má délku 56-bit + 8 paritních bitů. 3DES šifruje zprávu po 64-bit blocích. Nejprve šifruje pomocí K_1, poté dešifruje pomocí K_2 a posléze znovu šifruje pomocí K_3. Postup dešifrování je potom inverzní: dešifrovat pomocí K_3, šifrovat pomocí K_2 a naposledy dešifrovat pomocí K_1. 3DES dále umožňuje tři způsoby použití, kdy:

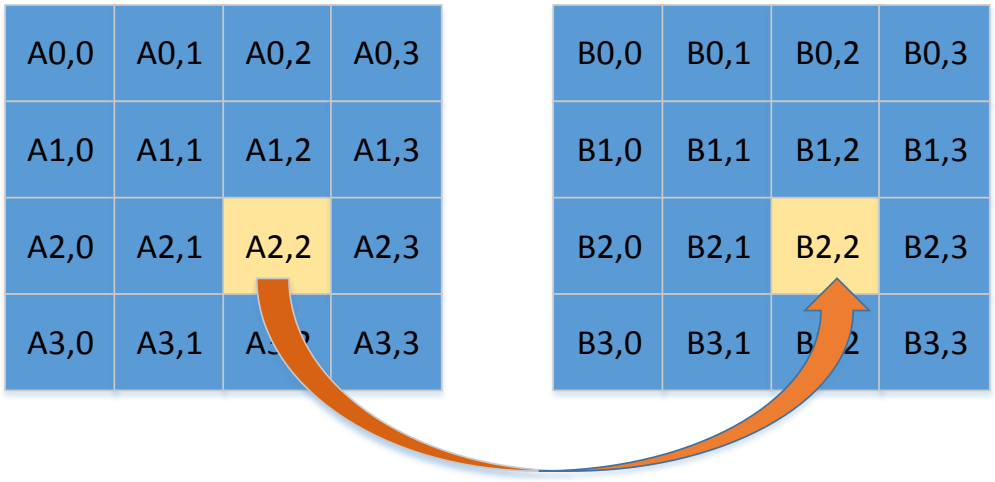
- Všechny klíče jsou na sobě nezávislé (nejsilnější možnost).
- K_1 a K_2 jsou na sobě nezávislé, ale K_1 a K_3 jsou identické (slabší, ale silnější než samotné DES).
- K_1, K_2 a K_3 jsou totožné. (Odpovídá DES, kvůli zpětné kompatibilitě).

AES

V roce 1997 byla vyhlášena soutěž o nový šifrovací standard. Úkolem bylo nalézt blokovou šifru s klíčem o délce 128b, 192b a 256b. Vítězem se stala šifra Rijndael [7], která byla standardizována v roce 2001 institutem NIST [8]. Velikost zpracovávaných bloků je 128 bitů a šifrování probíhá pomocí aplikace transformací, kdy se každá skládá z několika kroků. Některé závisí na volbě klíče. Dešifrování je potom inverzní proces.
Zatímco AES má pevně danou velikost bloku na 128 bitů a velikost klíče na 128, 192 nebo 256 bitů, původní algoritmus Rijndeal podporuje navíc bloky o délce 160, 192, 224 a 256 bitů, dále pak délky klíče 160 a 224 bitů. AES pracuje s maticí bytů 4×4 označovanou jako stav.

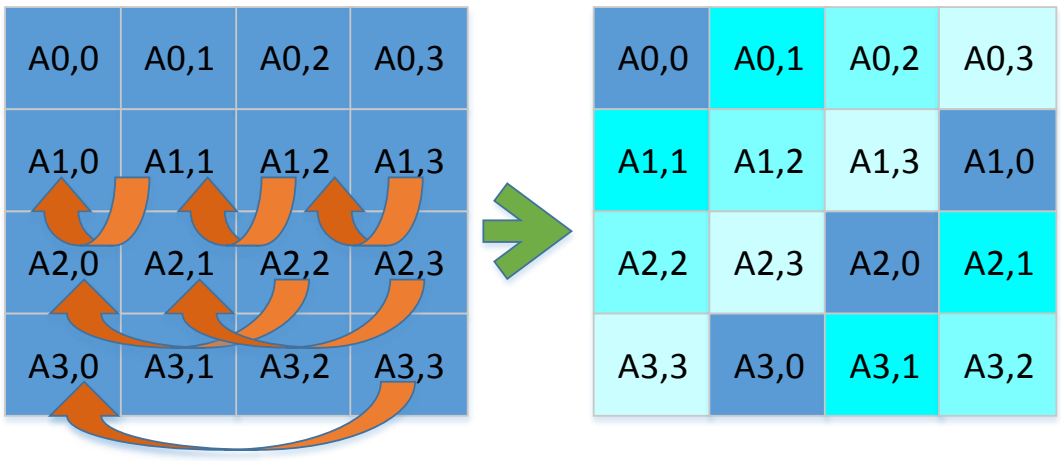
Šifrování probíhá ve čtyřech krocích:

1. **SubBytes** - jednoduchá substituce, kde je každý byte nahrazen jiným podle předem daného klíče, Rijndael-S-Box (8 bitů). Tato operace zajišťuje nelineárnost šifry a má zabránit útokům založeným na jednoduchých algebraických vlastnostech.



Obrázek 1-5 Substituce bitů v matici při použití AES [21]

2. **ShiftRows** — v tomto kroku se jednotlivé byty matice postupně posunou po řádcích o daný počet pozic tak, aby na se diagonále objevily byty ze stejného základního sloupce matice. Výsledný posun potom vypadá následně:

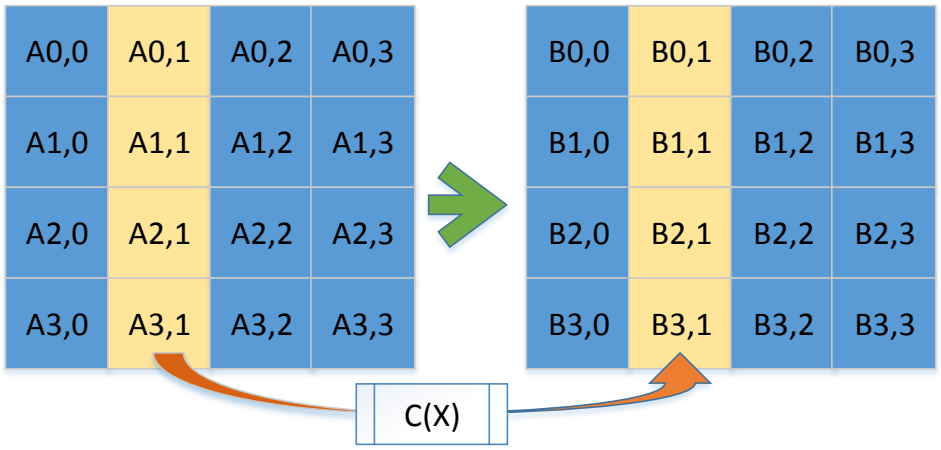


Obrázek 1-6 Posunutí bitů v řádcích matice při použití AES [21]

3. **MixColumns** — při této operaci dochází k přesunutí sloupců v rámci lineární transformace a zároveň je každý sloupec vynásoben stejným polynomem C*(x)*.

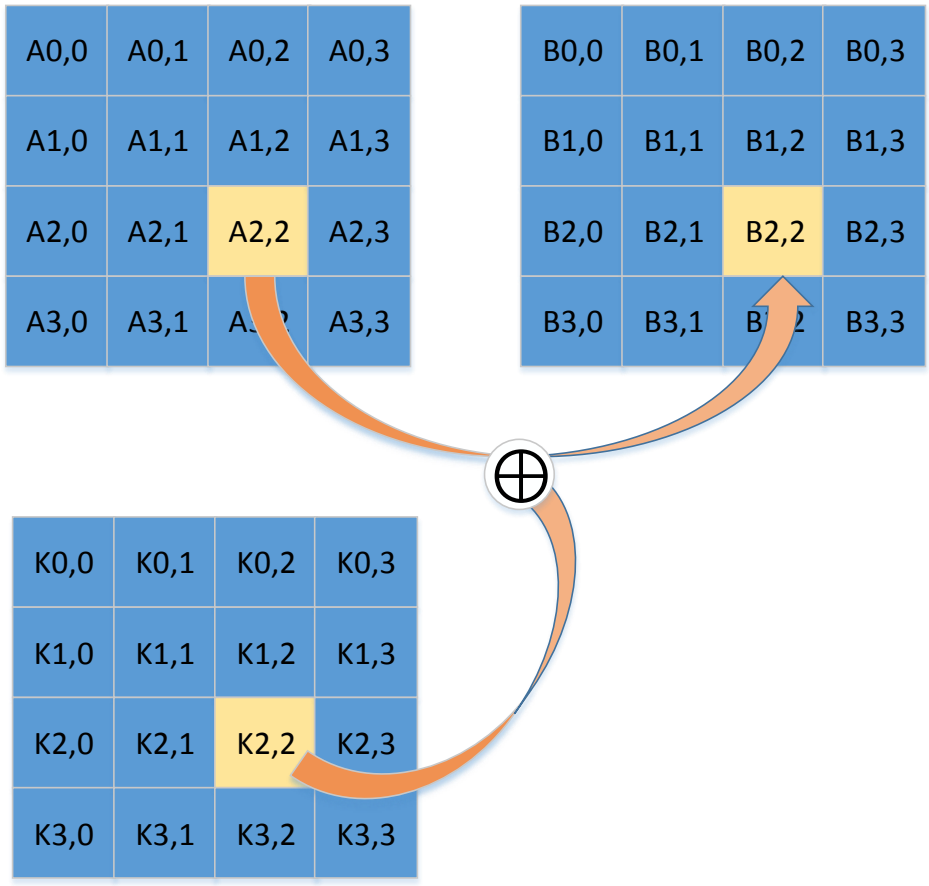
$$C(x) = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

Společně s předchozím krokem ShiftRows je krokem MixColumns zaručena dostatečná difuze šifrování a tím je přidáno na bezpečnosti použité funkce.



Obrázek 1-7 Prohození sloupců matice při použití AES [21]

4. **AddRoundKey** — každý byte je zkombinován se subklíčem (subklíč je získán z původního klíče pomocí Rijndaelovy tabulky [22]). Každý jednotlivý byte subklíče se zkombinuje s příslušným bytem zprávy a dostaneme výslednou šifru. Ke kombinování se používá logická funce XOR.



Obrázek 1-8 Kombinace bitu a sub-klíče při použití AES [21]

Standardu AES nehrozí, díky substitučně - permutivnímu postupu, útok hrubou silou, jako třeba u DES pomocí DES Crackeru. Výběr nového šifrovacího standardu trval více než čtyři roky, podílelo se na něm množství odborníku, tudíž se dá předpokládat, že šifra AES zůstane v blízké době neprolomitelná. Několik pokusů o narušení však proběhlo, veskrze neúspěšně, nebo se později ukázalo, že pokus by nemohl fungovat. Podle Nicolase Courtoise a Josefa Pieprzyka je znám i teoretický postup nalezení klíče [9], který ale pro 128 bitový klíč sestává z 8000 kvadratických rovnic o 1600 neznámých. Tato metoda byla, roku 2004 na konferenci AES4, okomentována jedním z tvůrců, V. Rijmenem, jako vysněná a nepoužitelná.

Proudová šifra

Proudová šifra je typ symetrické šifry, kde je vstupní datový tok kombinován (většinou pomocí funkce XOR) s pseudonáhodným proudem bitů vytvořeným z klíče pomocí šifrovacího algoritmu. Výsledkem je šifrovaný datový tok, ve kterém je klíč proudové šifry, na rozdíl od šifry blokové, nejprve vygenerován jako posloupnost hesla *h(1), h(2)…h(x)*, při čemž každý znak otevřeného textu je šifrován jinou transformací. Tuto transformaci určuje jeho pozice a odpovídající hodnota hesla na této pozici. To poukazuje na zásadní rozdíl oproti blokové šifře, kde jsou data šifrována pořád stejným řetězcem. Proudové šifry jsou typicky rychlejší než blokové šifry a i pro implementaci potřebují jednodušší hardware. Naopak jsou na rozdíl od blokových šifer náchylnější ke kryptoanalytickým útokům, pokud jsou nevhodně implementovány (počáteční stav nesmí být použit dvakrát). Navíc jsou šifry citlivé na ztrátu dat, kdy dochází ke ztrátě šifrovacího řetězce.

Proudové šifry jsou rychlejší než blokové. Vhodnější jsou tam, kde není možné využívat množství paměti. Typickým příkladem je telekomunikace a jiné real-time spojení, kde není vhodné čekat na naplnění paměti blokem dat, ale je třeba data šifrovat ihned. Mezi zástupce proudových šifer patří algoritmy RC4, FISH, Helix, SEAL, WAKE a další. Proudové šifry se dělí do dvou skupin, podle způsobu synchronizace.

Synchronní proudové šifry

Synchronní proudová šifra je taková, která nevyužívá k šifrování text (původní ani šifrovaný). Data jsou šifrována pomocí klíče a stavu, ve kterém se funkce nachází. Při dešifrování je proto potřeba mít stejný klíč a postupně procházet ekvivalentními stavy. Pokud se při přenosu ztratí jediný bit (nebo přibude), pak se synchronizace ztratí. Pro zachování synchronizace se používají další mechanizmy.

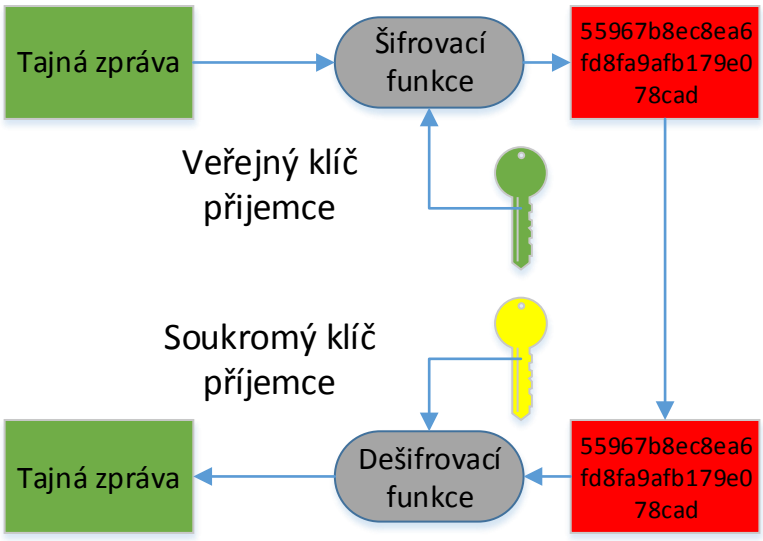
Proudové šifry s vlastní synchronizací (asynchronní)

Asynchronní proudové šifry využívají k šifrování i dešifrování mimo klíče i několik předchozích *n*-bitů šifrovaného textu. Tím se zvyšuje bezpečnost, ale při ztrátě jednoho bitu se následujících n bitů zprávy dekóduje špatně. Po těchto n bitech se další data již dekódují opět správně.

1.2.2. Asymetrická kryptografie

Jak už může být z názvu patrné, hlavním rozdílem mezi symetrickou a asymetrickou kryptografií, je to, že asymetrická nebude na obou stranách rovnocenná. Místo jednoho sdíleného hesla totiž asymetrický systém používá dvojici klíčů. Jeden klíč slouží vždy k šifrování a druhý k dešifrování, nazývají se veřejný klíč V_K a soukromý klíč S_k. Veřejný klíč, jak je z názvu patrné, není třeba držet v bezpečí, na rozdíl od soukromého, a je volně k dispozici. Naopak je v zájmu vlastníka soukromého klíče S_k, aby svůj veřejný klíč volně rozšířil mezi potenciální komunikační partnery, jelikož pomocí veřejného klíče *Vk* protější strana může ověřovat identitu.

Základní vlastností šifrování na bázi asymetrických algoritmů je skutečnost, že je relativně jednoduché s jeho použitím šifrovat text, ale s pouhou znalostí veřejného klíče a veřejným klíčem šifrované zprávy je velice obtížné získat původní zprávu.



Obrázek 1-9 Princip asymetrické kryptografie

Algoritmus RSA

Jedním z nejznámějších asymetrických kryptosystémů je dnes RSA. Název je složen ze jmen tvůrců Rivest, Shamir a Adleman, který byl jako jeden z prvních zaveden do komerční praxe.

Celý systém je založený na předpokladu, že rozložit číslo na součin prvočísel (faktorizace) je velmi obtížná úloha. Z čísla n=pq je tedy v reálním čase prakticky nemožné zjistit činitele p a q. Pro vysvětlení principu tvorby klíče jsou komunikující strany pojmenovány *A* a *B*. Nejprve si strana A bude muset vyrobit pár veřejného a soukromého klíče [10]:

- Zvolí dvě různá velká náhodná prvočísla p a q.
- Spočítá jejich součin $n = pq$.
- Spočítá hodnotu Eulerovy funkce $\varphi(n) = (p - 1) \times (q - 1)$.
- Zvolí celé číslo e menší než $\varphi(n)$, které je s $\varphi(n)$ nesoudělné.
- Nalezne číslo d tak, aby platilo $de \equiv 1 \pmod{\varphi(n)}$.
- Pokud e je prvočíslo tak $d = (1 + r \times \varphi(n)) / e$, kde $r = [(e - 1) \times \varphi(n)^{(e-2)}]$

Veřejný klíč je potom dvojice *(n, e)*, kde je *n* označováno jako modul a *e* jako šifrovací veřejný klíč. Soukromým klíčem se stává dvojice *(n, d)*, kde se *d* označuje jako dešifrovací exponent. Po vypočtení základní dvojice klíčů, může strana A **veřejný** klíč *(n, e)* s klidným svědomím pustit do světa, nebo na vyžádání poslat komukoli, kdo chce zahájit komunikaci, čili straně B. Samotná dvojice je ale k ničemu, pokud se s její pomocí nešifruje. Proto strana B rozdělí původní zprávu *M* na bloky *m* odpovídající nerovnici *(m < n)* a dále vytvoří šifrovanou zprávu *c* pomocí rovnice

$$c = m^e \bmod n$$

Takto šifrovanou zprávu, resp. *c*, odešle bez starostí protistraně *A*, která jí pomocí obrácené rovnice

$$m = c^d \bmod n$$

dešifruje a získá z ní bloky zprávy *m*, kterou bez problémů sloučí do původní zprávy *M*. Tento proces je umožněn faktem, vyplívajícím z rovnice

$$c^d \equiv (m^e)^d \equiv m^{ed} \pmod{n}$$

Současně platí $ed \equiv 1 \pmod{p - 1}$ a $ed \equiv 1 \pmod{q - 1}$. Lze tedy použít malou Fermatovu větu, která říká, že pro každé prvočíslo *p* a každé celé číslo *a* takové, že nejmenší společný dělitel LCD(*a*,*p*) je roven jedné, platí rovnice [11]

$$a^{p-1} \equiv 1 \pmod{p}$$

$$a^p \equiv a \pmod{p}$$

To v podstatě znamená, že číslo $(a^p - a)$ je dělitelné prvočíslem *p*. V případě RSA je tedy možno použít úpravu

$$m^{ed} \equiv m \pmod{p}$$

$$m^{ed} \equiv m \pmod{q}$$

Protože p i q jsou různá prvočísla, pomocí čínské zbytkové věty [12], můžeme říci, že:

$$m^{ed} \equiv m \pmod{pq}$$

a zároveň tak platí, že

$$c^{d} \equiv m \pmod{n}$$

Strana *A* tak dostala bezpečně šifrovaný záznam, který je pouze strana *A* schopna rozšifrovat. Ani strana *B* není schopna rozluštit šifrovanou zprávu. Pro vzájemnou komunikaci je tedy nezbytně nutné, aby si obě straně ještě před započetím komunikace vzájemně vyměnili svoje veřejné klíče V_k.

Délka klíče u RSA algoritmu, která se považuje za ještě bezpečnou, je 1024 bitů. V dnešní době ale již nastupují klíče o délce 2048 nebo 4096 bitů. Záleží na požadované míře bezpečnosti. Existují ale i jiné algoritmy asymetrické kryptografie, například algoritmus Diffie-Hellman nebo metoda využití eliptických křivek - Eliptic Curve Cryptography.

Algoritmus Diffie - Hellman

Diffie – Hellman je protokol sloužící k ustanovení tajného symetrického šifrovacího klíč. Byl popsán v roce 1976 Whitfieldem Diffiem a Martinem Hellmanem. Je to algoritmus umožňující výměnu klíče i přes nezabezpečený kanál. Velikou výhodou je, že potenciální útočník, který odposlouchává komunikaci, nemá šanci tento klíč zachytit. Klíč je totiž sestaven všemi účastníky komunikace a nikdy není přenášen v čitelné podobě. S tím se ale váže nemožnost autentizace a nepříjemná otázka napadení komunikace a tím pádem zapojení útočníka do procesu tvorby klíče. Z toho důvodu je potřeba buď proces provádět v kombinaci s jiným zabezpečením, anebo pouze v případě, že útočník nemá možnost aktivně zasahovat do komunikace.

Princip ustanovení je v podstatě jednoduchý. Prvním krokem je dohodnutí dvou čísel p a g tak, aby p bylo prvočíslo určující modul systému, a g ležící ve zvoleném modulu. Platí tedy rovnice [13]

$$1 \leq g \leq p - 1$$

Nejedná se o čísla tajná, lze je tedy vyměnit bez potřeby zabezpečení. Strana *A* si zvolí vlastní tajné číslo a a strana *B* si zvolí číslo b. Strana *A* poté provede první výpočet obsahující soukromé číslo a dle rovnice [13]

$$A = g^{a} \bmod p$$

Strana *B* poté provede obdobný výpočet, ovšem za použití svého soukromého b

$$B = g^{b} \bmod p$$

Vypočtená čísla *A* i *B* si obě strany bez obav vymění, výpočty tajného klíče *K* jsou závislé na soukromých číslech *a* a *b* [13]

$$K = B^a \bmod p$$

Strana *B* opět současně za použití svého *b* vypočte stejné *K*

$$K = A^b \bmod p$$

Oba těmito výpočty dojdou ke společnému výsledku *K*, který se označí jako tajný klíč. Tímto klíčem je nyní možné šifrovat komunikaci tak, jak k tomu dochází při symetrickém způsobu šifrování komunikace.

Algoritmus eliptických křivek - ECC

Asymetrická kryptografie je postavena především na problém faktorizace velkého čísla složeného z prvočísel, řešení diskrétního logaritmu nebo eliptického diskrétního logaritmu. U systémů, založených na bázi eliptických křivek, se předpokládá, že nalezení hodnoty diskrétního logaritmu náhodného elementu eliptické křivky s ohledem na jediný známý základní bod je nemožné. Velikost eliptické křivky určuje složitost problému a menší délka klíčů u kryptosystémů postavených na bázi eliptických křivek je umožněna větší složitostí matematického problému, na kterém jsou eliptické křivky postaveny. Jedná se o řešení problému eliptického diskrétního logaritmu, přičemž rychlost provádění operací s eliptickými křivkami je výrazně vyšší než u systémů založených na faktorizaci.

Pro kryptografické účely je nutno pracovat pouze s celými čísly, které mají navíc v počítači omezenou velikost. Od geometrické představy křivky v rovině se souřadnicemi v reálných číslech je proto nutné přejít k celočíselným souřadnicím a operacím s modulární aritmetikou. Pokud souřadnice *x* a *y* omezíme na množinu { 0 , 1 , 2, 3 , ... $p - 1$ }, kde *p* je velké prvočíslo, a všechny výpočty provádíme modulo *p*, jedná se o tzv. Galoisovo těleso GF(*p*), které nahradí původní těleso reálných čísel. Rovnice samotné eliptické křivky *E* nad tělesem GF(*p*) je definována jako množina bodů (*x* , *y*) odpovídajících rovnici:

$$y^2 = x^3 + ax + b$$

kde koeficienty *a, b* jsou též prvky GF (*p*).[14] Klíčovou a relativně snadno proveditelnou operací je potom výpočet *n*-násobku bodu ležícího na dané eliptické křivce. Násobitel *n* zde zastupuje roli soukromého klíče. Reverzní operace, tedy zjištění násobitele *n* z výsledného bodu, je výpočetně neschůdné.

1.3. Digitální podpis

Dlouho dobu byla otázka autentizace v digitální komunikaci poměrně problematická. Díky principům asymetrické kryptografie se ale začal popularizovat i pojem digitální podpis. Ten se vyvinul až do dnešní podoby, kdy je možné použít digitální podpis i pro důvěrný styk mezi uživateli, státními orgány, bankami a tak dále. Digitální podpis lze také použít např. místo razítka, tudíž jeho použití není striktně osobní, ale může být i firemní, resp. může identifikovat právnickou osobu. Oproti procesu šifrování dat digitální podpis přímo identifikuje podepisující entitu. Může navíc využívat ještě certifikát, který slouží k ověření podpisu třetí nezávislou osobou. Tímto je zaručena maximální bezpečnost v komunikaci mezi uživateli. Jedná se však pouze o ověření integrity zprávy, resp. ověření, že zpráva nebyla od podepsání nijak pozměněna. Jelikož digitální podpis si může vytvořit v podstatě každý, jsou rozlišovány různé míry důvěryhodnosti [18]:

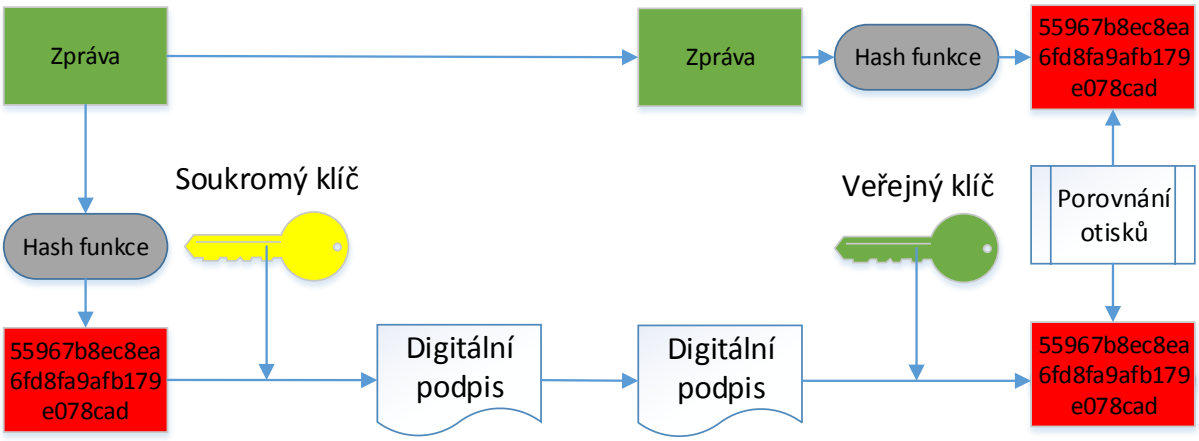
- elektronický podpis,
- zaručený elektronický podpis,
- zaručený elektronický podpis založený na certifikátu,
- zaručený elektronický podpis založený na kvalifikovaném certifikátu,
- zaručený elektronický podpis založený na kvalifikovaném certifikátu vydaném akreditovaným poskytovatelem certifikačních služeb.

Pokud je elektronický podpis založený na certifikátu vydaném akreditovaným poskytovatelem, je logicky jasné, že se jedná o nejdůvěryhodnější podobu používanou státní správou nebo velkými firmami. Jako asymetrický algoritmus se hojně používá standardizovaný systém RSA.

Aplikace digitálního podpisu probíhá ve dvou krocích využívajících dříve zmíněné funkce:

- Nejprve se vytvoří otisk dat pomocí hash funkce.
- Otisk se šifruje soukromým klíčem podepisujícího uživatele.

Digitální podpis je v tedy v principu pouze otisk podepisovaných dat zašifrovaný pomocí soukromého klíče. Odesilatel zasílá volně přístupnou kopii dat společně s digitálním podpisem. Nejedná se tedy o zabezpečení proti přečtení obsahu ve smyslu šifrování. Uživatel svým podpisem souhlasí s obsahem dat při odesílání a dává tím najevo, že jsou pro něj v pořádku a takto je chce odeslat.



Obrázek 1-10 Princip funkce digitálního podpisu

Ověřování digitálního podpisu potom probíhá ve třech krocích:

- Příjemce přijme data a vypočte otisk pomocí hash funkce
- Pomocí veřejného klíče příjemce dešifruje digitální podpis
- Následuje porovnání obou hash otisků.

Pokud se hash digitálního podpisu i nově vypočteného hashe přijatých dat shoduje, je jasné, že data nebyla po cestě nijak pozměněna a příjemce tak může důvěřovat tomu, co data reprezentují.

Potvrzením shodných hash otisků příjemce navíc ověřuje:

- Autentičnost – přijatá zpráva byla šifrována soukromým klíčem S_k, tudíž známého odesilatele
- Integritu zprávy – přijatá zpráva nebyla nijak pozměněna oproti odesílané verzi
- Nepopiratelnost – nelze popřít autorství zprávy
- Časové razítko – volitelně může podpis obsahovat i informace času o podepsání

1.3.1 Problematika vyčíslení matematický problémů

Bezpečnost asymetrické kryptografie spoléhá na problém vyčíslit určité matematické operace. Například při RSA je to problém faktorizace velkých čísel, při DH problematika diskrétního logaritmu a při eliptických křivkách je to eliptický diskrétní logaritmus. Tyto výpočty znamenají značnou výpočetní náročnost a při pokusu o prolomení zmanemá lineární prodlužování hesla geometrický nárůst času a pokusů potřebných k výpočtu kolizního řetězce. Se současnou popularizací kvantové fyziky a značnému úsilí ve vývoji kvantových počítačů se rychle zvyšuje nebezpečí použití jejich výpočtů, které využívají princip superpozice a kvantového provázání částic. Výpočty takových počítačů staví na principu reprezentace dat a struktur pomocí vlastností kvantových částic. Pro operace s takovými

operátory se využívají kvantové jevy mezi nimi. Obrovskou výhodou těchto strojů jsou tedy základní jednotky, se kterými se pracuje, tzv. qubity. Ty na rozdíl od klasických bitů mohou nabývat i hodnot mezi 1 a 0, kdy je pozice qubitu vyjádřena pomocí pravděpodobnosti výskytu v bodech *A(0)* a *B(1)* najednou.

V praxi to znamená, že pomocí qubitu výpočet probíhá v několika úrovních najednou a qubit reprezentuje všechny použitelné hodnoty. Potom tedy neplatí geometrický nárůst časové náročnosti výpočtu a v současnosti těžko prolomitelné šifrovací metody by mohly padnout v rámci minut, ne-li sekund.

Shorův algoritmus

Asi nejznámějším kvantovým algoritmem je právě Shorův algoritmus. Důležitý je hlavně proto, že dokáže efektivně obejít výpočetně náročnou faktorizaci velkého čísla. K tomu využívá posloupnost matematické vlny čísel, které vznikají při modulové aritmetice.

Ta využívá především faktu, že při použití určitého modula se u testování konkrétním čísle postupně vytvoří řada čísel, která má, stejně jako například vlna akustická, svou periodu opakujících se čísel. To se dá promítnout například při představě hodinového ciferníku. Při součtu hodin

$$10 + 8 = 18$$

zůstane na dvanáctihodinovém ciferníku ukazatel na pozici *6*. To znamená, že tento součet může být zapsán jako

$$10 + 8 = 6\ (mod\ 12)$$

Pokud je v podobné soustavě vybráno náhodné číslo, menší než je hodnota samotné soustavy, je možné s jeho pomocí získat původní čísla. Princip je v podstatě jednoduchý, stačí analyzovat vzniklou matematickou vlnu zvětšující se postupným umocňováním zvoleného čísla a pozorovat jeho periodu. V rámci příkladu lze použít například číslo 7.

$$7^1 = 7(mod\ 15);\ 7^2 = 4(mod\ 15);\ 7^3 = 13(mod\ 15);\ 7^4 = 1(mod\ 15);$$
$$7^5 = 7(mod\ 15);\ 7^6 = 4(mod\ 15);\ 7^8 = 13(mod\ 15);\ 7^9 = 1(mod\ 15);$$

Z toho plyne posloupnost zbytků, neboli matematická vlna, s periodou *4*

$$< 7\ ,4\ ,13\ ,1\ ,7\ ,4\ ,13\ ,1\ ,7\ \ldots >.$$

Další operace je už jednoduší a stačí periodu $p = 4$ rozdělit na polovinu a nově vzniklým parametrem exponovat původní použité číslo $y = 7$. Získáme tedy

$$p = 4;\ \ x = y^{\frac{p}{2}} = 49$$

Pro získání původních kořenů čísla 15 je potřeba v posledním kroku vzít nejbližší čísla okolo *x* a najít jejich společné dělitele s hodnotou soustavy.

$$LCD_a < 48\,, 15 >= 3;\ LCD_b < 50\,, 15 >= 5$$

Výpočet pro takto malé číslo je poměrně jednoduchý, ale pro větší čísla již ani tato cesta není lehce proveditelná. Nicméně s použitím kvantových výpočtů je tuto metodu velmi snadné provést současně pro několik čísel a získat tak matematickou vlnu velmi rychle pro mnoho náhodných čísel. Jelikož se navíc stále jedná o vlnu s určitou periodou, lze pro její analýzu použít Fourierovu transformaci a tak dále urychlit získání informací.

Zásadní problém reálného kvantovém počítače je tedy to, že standardní kryptografie je proti schopnostem kvantových výpočtů v podstatě bezbranná a zvyšování objemu hesla dále nezpůsobí kýžený nárůst bezpečnosti, jako u standardního výpočetního stroje. [26]

1.3.2 Post-Quantum Cryptography

V souvislosti s kvantovou aritmetikou se začala vyvíjet také adekvátní kryptografické metody. Nejedná se ale o kvantovou kryptografii. Ta využívá k přenosu klíče kvatové stavy částic a tím úspěšně eliminuje možnost odposlechu. Každé měření fotonů totiž ovlivní jeho stav a protější strana je tak okamžitě informována, že se je kanál odposloucháván. Mezi protokoly využívající tento fenomén patří například BB84 [29].

Naproti tomu se post-quantum kryptografie zabývá kryptografií odolnou i kvantovým počítačům.

Kryptografie s mřížovým základem

Tato skupina kryptosystémů zahrnuje šifrovací schémata NTRU, GGH nebo Ring-LWE. Jedná se o schémata využívající pro šifrování *n*-dimenzionální mřížky *L* v euklidovském prostoru R^n, která obsahuje množství bodů. Základem mřížky je taková skupina vektorů, pro kterou platí, že každý prvek *L* je jednoznačně reprezentován jejich lineární kombinací s celočíselnými koeficienty. Pokud je *n* alespoň 2, každá přížká má nekonečně mnoho různých základů. Každá mřížka v prostoru R^n má tedy nekonečně mnoho prvků ale jednotlivé šifrované položky, jako je text, soukromý nebo veřejný klíč atd., musí být vybrány

z konečného prostoru. Vzniká tím matematický problém, který je těžce řešitelný, pokud není znám další, unikátní identifikátor.

V současnosti se jako nejvíce zajímavý jeví systém Ring-LWE, který kombinuje výhody prokázané bezpečnosti a zároveň používají klíč použitelný pro všeobecné použití. [31]

Kryptografie s více proměnnými

Metoda využívá poznatku, že je složité vyřešit systémy s více proměnnými. Nicméně, i přes velké snahy zatím nevznikla bezpečné schéma založené na rovnicích s více proměnnými. Nicméně pro použití s kvantovým počítačem se jeví slibně metoda tzv. Nevyváženého oleje a octa - UOV. [32] Ta využívá toho, že oproti například RSA je základ klíčů mnohem komplexnější, založený na problému vyřešní *m* rovnic s *n* proměnnými, kdy veřejný klíč je celá soustavá rovnic. Jelikož by vyřešení soustavy *m, n* o velkých číslech zabralo obrovské množství výpočetních zdrojů, obsahuje soustava tzv. trapdoor systém, což je v podstatě soukromý klíč, který umožní výpočet soustavy.

Odolnost proti kvantovým výpočtům je potom jednoduchá. Neexistuje výpočetní algoritmus, který by kvantovému počítači dal výhodu nad standardním dnešním výpočetním strojem. Další velkou výhodou je, že operace potřebné k řešení rovnic jsou poměrně jednoduché a digitální podpis je tedy vytvořen nebo ověřen sčítáním a násobením relativně malých čísel, což umožňuje použití u méně výkonných zařízení.

Kryptografie izogenní supersingulární eliptické křivky

Systém je sestaven na vlastnostech supersingulárních eliptických křivek. S jejich pomocí lze vytvořir náhradu pro systém DH. Společně s faktem, že nový veřejný klíč je generován pro každou novou relaci a je v podstatě zcela náhodný znemožňuje použití jakéhokoliv deterministického algoritmu. [33] Jelikož jsou systém používán v izogenním systému, nehraje roli teoretická vyčíslitelnost diskrétního algoritmu, a proto nelze nasadit např. *Shorův algoritmus.*

Kvantová odoslnost symetrické kryptografie

Za předpokladu, že je pro každou novou relaci použit dostatečně velký klíč, jsou symetrické systémy jako například AES proti kvantovým výpočtům odolné již dnes. Situaci může navíc pomoct například ověřovací systém Kerberos. [34] Ten je navržen tak, aby zajišťoval silné zabezpečení zároveň s jednoduchým uživatelským rozhraním. Pro prokázání totožnosti v systému používá tzv. lístky, které vydává autentizační server obsluhující databázi uživatelů. Tyto lístky jsou velmi podobné např. certifikátům veřejného klíče, nicméně celý systém je postaven striktně symetricky.

2. Certifikát a struktura systému

Jelikož je matematicky velice složité prolomit klíče používané u asymetrických šifer, naskýtá se jednodušší cesta ke kompromitaci cizí komunikace a to pomocí podvrhnutí jiného veřejného klíče. Tato v praxi poměrně jednoduše proveditelná akce je velmi účinná, protože komunikující strany vůbec netuší, že je mezi nimi někdo třetí, tzv. „člověk uprostřed“ [25], kdo má dokonce možnost měnit zprávy.

Strany *A* a *B* chtějí pro zabezpečení komunikace využít výhody asymetrické kryptografie, proto si obě strany vygenerují svůj pár klíčů a chtějí si vyměnit své veřejné klíče. V tu chvíli ale přichází třetí strana *C*, která odchytává oba klíče a místo veřejného klíče Vk_A strany *A posílá* straně *B* svůj vlastní veřejný klíč Vk_C. Stejně tak straně *B* zašle vlastní veřejný klíč Vk_C a uloží si oba klíče Vk_A i Vk_B pro sebe. Strana *A* si tak myslí, že veřejný klíč, který obdržela, je od strany *B*, a naopak. Pro komunikující strany *A* i *B* je tedy komunikace bude zabezpečená. Strana *C* je tedy v postavení prostředníka přijímajícího zprávy od obou účastníků jak *A,* tak i *B*. Ty může je libovolně měnit, protože jsou šifrovány pomocí jejího veřejného klíče Vk_C. Je tedy potřeba zavádět systém ověřování vlastníka veřejného klíče. Tento systém se skládá z certifikátu a certifikační autority.

Certifikát lze považovat za elektronický ekvivalent občanského průkazu jednoznačně spojující fyzickou, resp. právnickou, totožnost s elektronickou. Tím zabezpečuje, že digitální identita je vázána na úředně ověřenou entitu. Technicky digitální certifikát *C* spojuje veřejný klíč V_k s osobou, aplikací anebo službou. Na rozdíl od jednoduchého digitálního podpisu není získání certifikátu soukromou věcí, ale aby byl celosvětově použitelný, musí být vydán důvěryhodnou certifikační autoritou *CA*. Kvůli nezpochybnitelnosti je certifikát *C* při vydání podepsán soukromým klíčem vydávající certifikační autority *CA*. Veřejný klíč *CA* je volně k dispozici a tak kdokoliv může ověřit pravost certifikátu *C*.

2.1. Vydání certifikátu

Žadatel si nejprve vygeneruje pár klíčů a poté vytvoří elektronickou žádost o certifikát. Jedná se o datový soubor ve standardním formátu (zpravidla podle normy PKCS#10), který obsahuje požadované identifikační údaje (v závislosti na typu certifikátu jsou požadovány odlišné údaje, například u osobního certifikátu bude požadováno jméno a trvalé bydliště žadatele), veřejný klíč žadatele a dále některé doplňkové informace. Celá žádost je poté podepsána vygenerovaným soukromým klíčem. Tím žadatel spolehlivě a nepopiratelně prokazuje, že je vlastníkem párového soukromého klíče. Nyní je třeba žádost doručit

certifikační autoritě, která musí ověřit, zda údaje uvedené v žádosti jsou pravdivé (v případě osobního certifikátu třeba kontrolou dokladu totožnosti žadatele) a také zda podpis žádosti je platný. V případě úspěšného ověření údajů certifikační autorita vystaví na základě žádosti platný certifikát.

Přestože se tento postup složitý, ve skutečnosti je téměř automatický, většinou stačí navštívit webové stránky certifikační autority, zvolit žádost o certifikát a do zobrazeného formuláře vyplnit požadované údaje. Vygenerování žádosti a její odeslání je potom zcela transparentní, někdy je pouze vyžadováno nahrání žádosti na disk a osobní doručení do sídla certifikační autority. Vystavený certifikát je poté možné jednoduše stáhnout z webu, může být zaslán emailem, případně opět předán osobně na paměťovém médiu.

Existuje několik typů certifikátu, rozlišených podle účelu a podle toho, jakému subjektu jsou vydány. Certifikát tedy nemusí být vydán pouze fyzické osobě, ale třeba i firmě, která může například elektronicky podepisovat software, který produkuje. Velmi často se používá tzv. serverový certifikát, který slouží pro zabezpečenou komunikaci klientova internetového prohlížeče se serverem pomocí protokolu SSL. Přirozeně pro různé typy certifikátu musí certifikační autorita ověřit jiné údaje, v případě firemního certifikátu to bude například výpis z obchodního rejstříku, u serverového certifikátu je třeba ověřit, že žadatel je vlastníkem domény, pro kterou je certifikát vydán.

2.2. Důvěryhodnost certifikační autority *CA*

Jelikož je certifikát vydán certifikační autoritou, měla by tato autorita být pro žadatele důvěryhodná. Nemusí tomu tak ale být v případě ověřujícího. Ten tedy nejprve musí ověřit podpis certifikátu *C* s certifikátem samotné certifikační autority *CA*. Pokud tento certifikát již není mezi, pro ověřujícího, důvěryhodnými *CA*, musí si jej najít na stránkách certifikační autority *CA* a rozhodnout se, jestli je pro něj tato *CA* důvěryhodná, či nikoli.

Pokud se rozhodne, že je, a přijme certifikát nové certifikační autority *CA* za důvěryhodný, vyextrahuje si z něj veřejný klíč *CA* a verifikuje s ním digitální podpis na ověřovaném certifikátu *C*. Pokud vše proběhne bez problémů, je certifikační autorita *CA* zařazena mezi důvěryhodné a ověřovaný certifikát *C* je přijat k šifrování komunikace.

2.2.1. Certifikáty certifikační autority

Jak bylo výše zmíněno, i certifikační autorita *CA* potřebuje svůj vlastní certifikát, sloužící k ověřování jejích podpisů. Z toho plyne logická otázka, kdo vydá certifikát *C* certifikační

autoritě, resp. kdo se zaručí za důvěryhodnost certifikační autority *CA*. Existují dvě možnosti, které se v praxi používají

- Certifikát *C* si vydá a podepíše sama certifikační autorita *CA*. Toto se označuje jako SELF SIGNED CERTIFICATE
- Certifikační autorita *CA* si vyžádá certifikát od jiné, hierarchicky vyšší, certifikační autority CA_H. Toto je označováno jako CROSS CERTIFICATE

Hlavním úkolem certifikátu *C* vydaném certifikační autoritě *CA* je maximálně se nechat rošířit a prohlásit tak veřejně danou *CA* za důvěryhodnou.

2.3. Struktura certifikátu

Jak již bylo řečeno, je digitální certifikát přirovnatelný k reálnému občanskému průkazu. Technicky je to ale digitálně podepsaná datová struktura, jejíž základní součástí je veřejný klíč držitele certifikátu. Touto ověřenou strukturou je možné verifikovat totožnost entity, osoby nebo služby a v některých zemích jsou dostupné i občanské průkazy vybavené čipem s tímto certifikátem. Celá datová struktura tvořící certifikát *C* musí být vytvořena podle standardu X.509 v aktuální verzi 3. Tento standard vydává společenství ITU. Z hlediska významového rozdělení struktury certifikátu se uvádí následující položky.

2.3.1. Verze certifikátu - *VERSION*

Tato položka vychází z použitého standardu certifikátu. Norma X.509 udává aktuálně tři verze, kdy připravovaná 4. verze již fyzicky strukturu nemění. Položka VERSION tedy může nabývat hodnoty 0 pro verzi 1, hodnotu 1 pro verzi 2 a hodnotu 2 pro verzi 3 a 4. Dnes jsou ale v zásadě používány pouze certifikáty ve verzi 3.

2.3.2. Pořadové číslo certifikátu – *SERIAL NUMBER*

Hodnota položky pořadového čísla SERIAL NUMBER je udávána jako celé kladné číslo. Toto číslo musí být jednoznačné v rámci konkrétní certifikační autority *CA*. Certifikační autorita tedy nesmí vydat dva certifikáty C_1 a C_X se stejnou hodnotou v položce pořadového čísla. Tato položka, společně s položkou vydavatele ISSUER jednoznačně identifikují certifikát.

2.3.3. Algoritmus podpisu – *SIGNATURE ALGORITHM*

Pole algoritmu podpisu specifikuje algoritmy, které certifikační autorita *CA* používá pro vytvoření elektronického podpisu certifikátu.

Tato položka je vždy specifikována dvojicí algoritmů:

1. Algoritmus použitý pro výpočet hash otisku
2. Asymetrický algoritmus použitý pro šifrování otisku

2.3.4. Platnost - *VALIDITY*

Tímto polem vydavatel vyznačuje použitelnou dobu certifikátu. Platnost je určena intervalem „od“ (NOT BEFORE), „do“ (NOT AFTER). Potřeba omezovat platnost vychází ze dvou hlavních důvodů – organizace a bezpečnost.

Organizace a správa

Důvodem může být například určitá životnost aplikace nebo předplatné služby. Především pak potřeba vydávat další certifikáty, což je finančně vhodné pro certifikační autoritu.

Bezpečnost

Životnost certifikátu nesmí být delší, než je doba potřebná k prolomení certifikovaného veřejného klíče. Tedy při použití delšího klíče je možné vydávat déle platné certifikáty. Další bezpečnostní důvod pro omezení platnosti je ten, že certifikační autorita *CA* musí mít certifikát vystaven na dobu alespoň 5× delší, než je platnost soukromích uživatelských certifikátů. Při kratší době se výrazně zvyšuje potřeba správy a obnovování uživatelských certifikátů. Po vypršení doby platnosti certifikátu jej není možné smazat nebo zahodit. Jím podepsané dokumenty stále vyžadují tento certifikát k ověření pravosti elektronického podpisu. S takto „prošlým“ certifikátem není nadále možné podepisovat nové dokumenty, ale stále obsahuje veřejný klíč V_K, který je párový k soukromému klíči S_k, jímž byly dokumenty v dané době podepisovány. Je tedy nadále platným důkazem vlastnictví obou klíčů. Pro archivaci podepsaných dokumentů se tedy doporučuje nejprve je dešifrovat pomocí certifikovaného klíče a poté je dodatečně zašifrovat klíčem archivu.

2.3.5. Položky Vydavatel – *ISSUER* a Předmět - *SUBJECT*

Vydavatel, ISSUER, jednoznačně identifikuje vydavatele certifikátu, tedy certifikační autoritu *CA*. Naopak položka předmětu SUBJECT jednoznačně identifikuje držitele certifikátu. Obě položky mají podle normy stanoven stejný datový formát označovaný jako DISTINGUISHED NAME – Jedinečné jméno.

Jedinečné (rozlišovací) jméno – DISTINGUISHED NAME

Jedinečné jméno slouží k jednoznačnému určení entity v rámci normy X.501 vydané ITU. Cílem norem řady X.500 je vytvoření celosvětové adresářové struktury. Takovou strukturou se rozumí seznam, kde každý záznam odpovídá unikátnímu jménu entity. Jedinečné jméno DISTINGUISHED NAME je tvořeno dílčími informacemi [23] o subjektu, které se nazývají

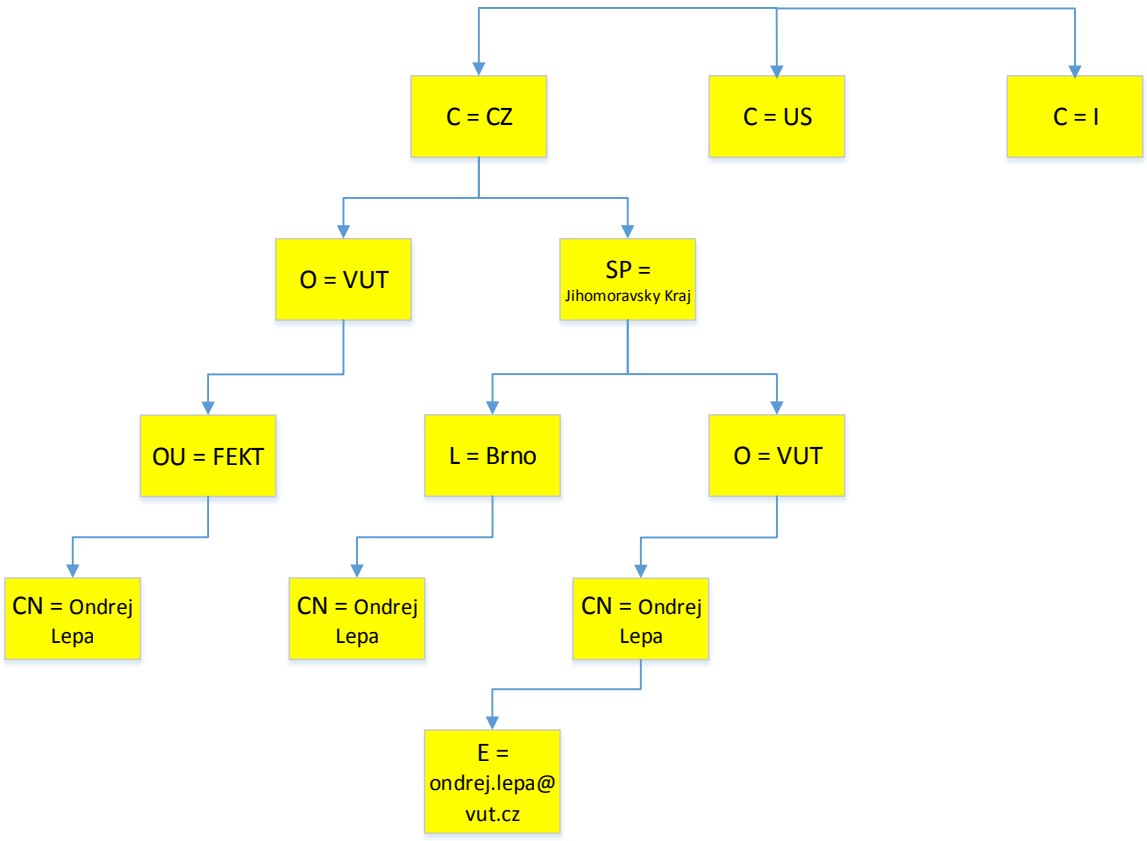
RELATIVE DISTINGUISHED NAME. To může být tvořeno množinou identifikátorů, v praxi se používá pouze jednoprvková množina, tvořená párem *identifikátor* + *hodnota*.

$$CommonName = Ondrej\ Lepa$$

Jedinečné jméno DN je pak tvořeno sekvencí RDN, kde se zápis může zkrátit pomocí zkratek, například takto:

$$CN = Ondrej\ Lepa, T = Student, OU = FEKT, O = VUT, C = CZ$$

V tomto zápisu je entita identifikovaná jako „Ondřej Lepa“ na pozici „student“ (organizační jednotky) „FEKT“, organizace „VUT“ v České republice („CZ“). Jak lze vidět na *obr 2-1*, je možné jednu konkrétní entitu identifikovat vícero možnými způsoby, přesto se stále jedná o stejnou entitu. V praxi se používají jednotlivé větvě stromu relativních jmen *RDN*. Výběr větve záleží na konkrétním použití výsledného certifikátu, kde pro firemní certifikaci rozhodně poslouží lépe identifikace pomocí větve týkající se ORGANISATION. Jednotlivé použití zkratek a objektů ve stromu jmen záleží na konkrétní certifikační politice. Certifikační autorita by neměla modifikovat jedinečná jména DISTINGUISHED NAME při kopírování z certifikační žádosti do finálního certifikátu.



Obrázek 2-1 Princip adresářové struktury jedinečných jmen

Vydavatel certifikátu – ISSUER

Vydavatel certifikátu je nezaměnitelná položka, určující jedinečné jméno certifikační autority *CA*. Je nutné, aby byla certifikační autorita *CA* jednoznačně identifikovatelná v rámci všechno certifikačních autorit. Potenciální útočník by mohl mít snahu vytvořit vlastní certifikační autoritu CA_X se stejným jménem, ale za použití vlastní dvojice soukromého a veřejného klíče S_K, resp. V_K. Pokud tedy certifikační autorita *CA* používá kořenový certifikát, SELF SIGNED CERTIFICATE, je třeba dávat pozor.

Předmět certifikátu - SUBJECT

Norma X.509 verze 3 vyžaduje, aby byl předmět certifikátu jedinečný v rámci všech objektů certifikovaných danou certifikační autoritou *CA*. Zajišťuje se tím, že *CA* nevydá různým osobám certifikát *C* se stejnou položkou SUBJECT. Pokud bude ale stejná konkrétní osoba žádat certifikační autoritu o další certifikát, může použít stejnou hodnotu v položce SUBJECT, jedná se přeci o stejnou osobu, identifikace tedy není problém. Je ale potřeba kontrolovat položku DISTINGUISHED NAME, aby omylem nedošlo k záměně jmenovců. Pokud by se náhodou shodovala i hodnota položky DISTINGUISHED NAME, k rozlišení se poté používají dodatkové položky dnQUALIFIER a serialNumber [1], které označují jednotlivé certifikované entity.

2.3.6. Veřejný klíč – *SUBJECT PUBLIC KEY*

Tato položka je tvořená sekvencí dvou informací.

- Identifikátorem algoritmu, pro který je veřejný klíč V_K určen
- Samotným veřejným klíčem

Informace o algoritmu na rozdíl od položky SIGNATURE ALGORITHM specifikuje algoritmus pro který je určen certifikovaný veřejný klíč. Například tedy při použití algoritmu Diffie-Hellman si žadatel certifikuje svoje veřejné DH číslo a položka SUBJECT PUBLIC KEY potom obsahuje identifikátor DH algoritmu a žadatelovo DH veřejné číslo.

2.3.7. Možná rozšíření certifikátu

Do rozšiřujících položek certifikátu jsou udány informace, které se do základních identifikačních položek nevešly. Ty dále rozšiřují identifikaci držitele certifikátu. Dle zásady se do certifikátu vkládají pouze informace týkající se přímé identifikace držitele certifikátu *C*. Pro definici přístupových práv, nebo určení pozice držitele certifikátu slouží Atributové certifikáty. O důležitosti rozšíření rozhoduje položka CRITICAL, která definuje závažnost.

[1] nezaměňovat s položkou SERIAL NUMBER označující číslo certifikátu – to musí být jedinečné

Pokud je hodnota nastavena jako TRUE, ověřující aplikace potom rozšíření věnuje pozornost. Pokud ale ověřující aplikace nerozumí některému ze závažných rozšíření, musí certifikát odmítnout, protože neví, které informace jsou důležité pro identifikaci a mohla by některou opomenout. [18]

Identifikátor klíče předmětu a Identifikátor klíče úřadu

Certifikační autorita může vlastnit více párů klíčů, každý s vlastním certifikátem a s jiným způsobem použití. To může být matoucí a není pak jednoznačné, který certifikát *C* certifikační autority *CA* má být použit k ověření soukromého certifikátu C_S. Certifikační autorita *CA* si tak označí jednotlivé veřejné klíče, většinou za použití hash otisku veřejného klíče V_{KX}, a tuto hodnotu poté doplní do položky IDENTIFIKÁTOR KLÍČE PŘEDMĚTU. Do každého vydaného soukromého certifikátu C_S potom certifikační autorita *CA* vkládá jako položku IDENTIFIKÁTOR KLÍČE ÚŘADU právě tento hash otisk použitého veřejného klíče V_{KX}.

Platnost soukromého klíče (DUE DATE)

Položka pomáhající definovat kratší dobu platnosti soukromého klíče, než je uvedeno v položce platnosti certifikátu. Díky tomu je umožněno ověřování podpisů po mnohem delší dobu, než podepisování. To podstatně usnadňuje ověřování postarších dokumentů. Bohužel se k tomu váže problematika certifikátů certifikačních autorit *CA*. Ta musí mít vždy platný certifikát C_{CA}, kterým je možné ověřit i starší soukromé certifikáty C_S. Pokud tedy certifikační autorita *CA* vydává certifikáty C_S s roční platností, musí *CA* vydat minimálně rok před vypršením vlastního certifikátu C_{CA} nový certifikát C_{CAN} tak, že C_{CA} bude ještě rok sloužit k ověřování starších certifikátů C_S, ale nové certifikáty C_S budou už podepisovány pomocí C_{CAN}. Certifikační autorita *CA* má tedy současně dva certifikáty, kdy mají oba stejné položky SUBJECT a ISSUER. Budou se však lišit v hodnotách položky SERIAL NUMBER a obsaženým veřejným klíčem.

Použití klíče (KEY USABILITY)

Toto pole slouží k omezení možných použití veřejného klíče obsaženého v certifikátu. Označení je uvedeno jako bitový řetězec, kde hodnota TRUE znamená možnost konkrétního použití a pokud se daný bit v řetězci nevyskytuje, předpokládaná hodnota je právě TRUE. Jednotlivé bity v řetězci jsou:

Digitální podpis (DIGITAL SIGNATURE)

Certifikát je určen k digitálnímu podpisu, ale zároveň není oprávněn k ověřování pravosti. Může ale sloužit k ověřování integrity dat i k autentizaci uživatelů.

Nepopiratelnost (NON-REPUDIATION)

Certifikát je určen k ověřování pravosti, resp. nepopiratelnosti, neodvolatelnosti. Pokud je dokument podepsaný takto označeným certifikátem nemůže být pochyb o jeho původu. Zároveň ale neslouží jako digitální podpis, za tímto účelem je potřeba vyplnit navíc bit digitálního podpisu jako TRUE.

Šifrování klíčů (KEY ENCIPHERMENT)

Takto označený certifikát je určen k šifrování klíčů. Data šifrovaná symetrickým klíčem jsou, společně s tímto klíčem, šifrována veřejným klíčem obsaženým v certifikátu s povolením KEY ENCIPHERMENT.

Šifrování dat (DATA ENCIPHERMENT)

Certifikát s tímto označením je možné použít přímo k šifrování dat. Klíč by měl být natolik silný, že není potřeba předem šifrovat pomocí symetrického klíče.

Sjednání klíče (KEY AGREEMENT)

S takto označeným certifikátem je možné sestavovat páry klíčů, např. pomocí algoritmu Diffie-Hellman

Podepisování certifikátu (KEY CERTIFICATION SIGN)

Využívá se u certifikátů certifikačních autorit. Pokud je hodnota tohoto bitu vyplněna jako TRUE, je možné s tímto certifikátem verifikovat další certifikáty

Podepisování seznamu odvolaných certifikátu (CRL SIGN)

Veřejný klíč takového certifikátu je možné využít k podepisování seznamu odvolaných certifikátů *CRL*.

Pouze k šifrování (ENCIPHER ONLY)

Váže se k položce KEY AGREEMENT společně s kterým je možné certifikát použít k dohodnutí symetrického klíče. S tímto certifikátem lze ale dohodnout klíč použitelný pouze k šifrování

Pouze k dešifrování (DENCIPHER ONLY)

Váže se k položce KEY AGREEMENT společně s kterou je možné certifikát použít k dohodnutí symetrického klíče. S tímto certifikátem lze ale dohodnout klíč použitelný pouze k dešifrování

Alternativní jméno předmětu

Pomocí alternativních jmen předmětu lze pro držitele definovat více jedinečných jmen DISTINGUISHED NAME. Certifikační autorita *CA* však nesmí opomenout kontrolu těchto

údajů stejně, jako u povinných údajů. Poskytnutá alternativní jména slouží k identifikaci dalších držitelových služeb a pseudonymů. Některá možná alternativní jména jsou:

- Jiný název (OTHER NAME) – například pseudonym; jiný identifikační údaj.
- Název v RFC822 (RFC822 NAME) – adresa elektronické pošty dle RFC822.
- DNS jméno (DNS NAME) – udává jméno serveru kde entita působí.
- Název v X.400 (X.400 ADDRESS) – adresy elektronické pošty dle X.400.
- DIRECTORY NAME – adresářové jméno dle X.500; stejný formát jako SUBJECT nebo ISSUER.
- IP ADDRESS – IP adresa serveru, pro IPv4 vymezuje 4 bajty; pro IPv6 pak 16 bajtů.

Certifikační politiky (CERTIFICATION POLICY)

Jedná se o zásady používání certifikátů vydaných konkrétní certifikační autoritou *CA*. Pole může obsahovat hypertextový odkaz přímo na dokument specifikující zásady používání certifikátu. Pokud ale obsahuje odkaz, není zaručeno, že dokument nebude pozměněn během doby platnosti certifikátu. Certifikační politika nicméně specifikuje postupy, praktiky a cíle sloužící k ověřování certifikátu a dále specifikuje pravidla, za kterých daná certifikační autorita *CA* certifikáty *C* vydává a jak za vydané certifikáty ručí. Tento dokument je veřejný a měl by být dostupný přímo na stránkách certifikační autority *CA*.

Mapování zásad

Tato položka se používá v certifikátech certifikačních autorit v případě CROSS CERTIFICATE, tedy když certifikační autorita *CA* vlastní certifikát C_X podepsaný jinou certifikační autoritou CA_X. Je pravděpodobné, že obě certifikační autority budou mít podobné certifikační politiky, ale ty se mohou v určitých skutečnostech lišit. Jelikož podpisem certifikátu C_X certifikační autorita CA_X přebírá jistou zodpovědnost za certifikační autoritu *CA*, je nezbytně nutné aby si při ověřování řetězce certifikátů certifikační politiky odpovídaly. Úkolem položky mapování zásad je tedy porovnávání jednotlivých certifikačních politik a oznámení jejich srovnatelnosti.

Omezení využívání certifikátu – CONSTRAINS

Certifikační autorita *CA* svým podpisem ručí za pravost a pravdivost vydaného certifikátu. Stejně tak ručí za informace v něm obsažené. Tím nepřímo přebírá zodpovědnost i certifikáty vydané držitelem daného certifikátu. Pokud se tedy držitel certifikátu *C* rozhodne prohlásit se za certifikační autoritu CA_D, stává se za něj původní certifikační autorita *CA* zodpovědnou. Toto lze omezit pomocí rozšíření POUŽITÍ KLÍČE, nebo přímo pomocí položky CONSTRAINS. To zamezuje svévolnému prohlášení se držitele certifikátu

C za certifikační autoritu CA_D, a pokud i přesto vydá certifikát, bude platný pouze v určité oblasti.

Distribuční místa seznamu odvolaných certifikátů

Pomocí informace v této položce je ověřovatel schopen jednoznačně identifikovat a nalézt seznam distribučních míst, kde nalezne seznam odvolaných certifikátu *CRL*. To buď odpovědnost samotné vydávající certifikační autority *CA*, nebo jí pověřené jiné certifikační autority CA_X. Pole může obsahovat buď seznam míst, nebo přímo URL odkaz na seznam.

2.3.8. Kvalifikovaný certifikát

Jedná se o speciální typ certifikátu, který používá ve své legislativě Evropské Unie. Tento typ rozšiřuje právní použití obecného certifikátu a snaží se tak umožnit nahrazení ručně psaného podpisu jeho digitální formou za pomoci právě kvalifikovaného certifikátu. Norma definující pravidla kvalifikovaného certifikátu je RFC-3739. Takto označený certifikát obsahuje identifikaci držitele na základě oficiální identity člověka nebo jeho pseudonymu. Certifikační autorita *CA* pak zná konkrétní osobu, která certifikát vydala. Položka předmětu SUBJECT toho certifikátu musí být jednoznačná pro danou osobu po celou dobu existence konkrétní *CA*. Certifikační autorita také musí kontrolovat, aby po celou dobu své existence necertifikovala dva stejné veřejné klíče, proto je musí uchovávat v databázi a kontrolovat proti duplicitě.

2.4. Životní cyklus certifikátu

Existují různé logické stavy, které certifikát může nabývat. S těmito stavy je vázána funkčnost samotného certifikátu. Navíc platný certifikát lze využít i pro ověřování totožnosti pro další certifikáty, čili pokud držitel vlastní například Podpisový certifikát, pro vydání dalšího certifikátu již nemusí dokazovat svou totožnost certifikační autoritě *CA* osobně, ale postačí mu prokázání se platným certifikátem. Tím spíš je potřeba kontrolovat stav certifikátu.

2.4.1. Vytvoření žádosti o certifikát

Samotné vytvoření žádosti probíhá pod záštitou certifikační autority *CA*, která má vlastní formulář vyžadovaných informací. Ten může být buď přímo na internetových stránkách autority, nebo i ke stažení kvůli případnému notářskému ověření v případě kvalifikovaných certifikátů. Vytvoření žádosti může předcházet také generace páru klíčů. Některé *CA* nabízí službu generace klíčů až po obdržení žádosti, ale toto je málo běžné.

2.4.2. Vydání certifikátu

Vystavení certifikátu držiteli a jeho případná publikace, pokud se jedná o certifikát *CA*.

2.4.3. Platnost certifikátu

Certifikát nemusí být platný ihned po vydání. O jeho platnosti rozhoduje položka NOT BEFORE a NOT AFTER. Tento interval může být změnět pouze uvedením certifikátu na seznam odvolaných certifikátů *CRL* a tím i jeho veřejnému znehodnocení.

2.4.4. Vypršení platnosti certifikátu

K expiraci certifikátu dochází při překročení data uvedeného v položce NOT AFTER. Tato doba je specifikována certifikační autoritou *CA*, jak je zmíněno již v popisu struktury certifikátu.

2.4.5. Odvolání certifikátu

K odvolání certifikátu dochází před uplynutím jeho stanovené doby platnosti zveřejněním jeho identifikace do seznamu *CRL*. To má na starosti certifikační autorita *CA* a ta jej zveřejňuje do všech *CRL* seznamů po celou deklarovanou dobu platnosti odvolaného certifikátu. K odvolání dochází z pravidla z těchto důvodů:

- Certifikační autorita registrovala požadavek certifikace od jiného uživatele se stejným veřejným klíčem. Musí proto zrušit i již vydaný certifikát, neboť došlo ke kolizi.
- Údaje uvedené v certifikátu již nejsou déle platné. Například při změně jména, bydliště atd.
- Držitel již nechce certifikát používat a sám *CA* zažádá o odvolání certifikátu.
- Soukromý klíč držitele certifikátu byl vyzrazen nebo zničen.

Při odvolávání certifikátu *C* z platnosti nezáleží až tak na pravidlech, jako spíše na rychlosti zveřejnění a odstavení certifikátu z provozu a platnosti. Pokud je certifikát *C* používán pouze v jedné aplikaci, například internetovém bankovnictví, je logické informovat nejprve provozovatele aplikace, tedy banku *B* a až poté certifikační autoritu *CA*, která tento certifikát vydala. Banka *B* okamžitě znepřístupní služby pro tento certifikát *C* a certifikační autorita *CA* vydá nový seznam odvolaných certifikátů *CRL*. Certifikační autority vydávají seznam odvolaných certifikátů *CRL* v pravidelných intervalech, je ale logické, že s odvoláním certifikátu *C* není možné čekat až do doby vydání nového seznamu *CRL* už jen kvůli možnému použití ve více aplikacích. V tomto případě pak nestačí informovat o zneplatnění certifikátu *C* pouze banku *B*. Je nutné co nejrychleji zablokovat platnost certifikátu jako

takového a to pokud možno online. K tomu slouží pomocný server ověření aktuální stavv certifikátu – *OCSP server*, který bude popsán později.

2.5. Certifikační a registrační autority

Samotnou certifikační autoritu lze chápat například jako třetí stranu, která je do maximální možné míry nezávislá a nekompromitovatelná. Ta vydává a ověřuje certifikáty vydané prověřeným držitelům. Certifikační autorita může být buď přímo aplikace nebo celá samostatná firma, nebo autonomní útvar v jejím rámci. Instituce *CA* lze rozdělit na několik menších základních částí.

2.5.1. Registrační autorita (*RA*)

Tuto část lze chápat jako přepážku, která osobně vyřizuje žádosti. Žadatel přinese požadavek obsahující podklady pro vytvoření certifikátu, *RA* ověří žadatelovu totožnost, verifikuje požadavek a předá jej samotné *CA*, čili jádru aplikace. Registrační autorita zpravidla tuto žádost podepisuje pomocí vlastního certifikátu. Stvrzuje tím, že verifikovala poskytnuté údaje a ručí tak za jejich správnost, současně ručí za žadatelovu identitu.

2.5.2. Jádro CA – Vydávající aplikace

Samotná aplikace vydávající certifikáty, která podepisuje schválené certifikáty svým soukromým klíčem. Tento soukromý klíč je tak nejdůležitějším vlastnictvím celé certifikační struktury. Pokud by se někdy stalo, že je soukromý klíč certifikační autority kompromitován, znamenalo by to nutnost odvolání všech platných certifikátů a nutnost tvorby nových. Pokud by ještě vůbec někdo kompromitované autoritě důvěřoval. Soukromé klíče *CA* jsou proto chráněny pomocí hardwarových tokenů nebo znepřístupněny přístupu zvenčí. Tím by měla být zajištěna maximální bezpečnost soukromého klíče. [15]

2.5.3. Databáze uživatelů

Obsahuje data o uživatelích a vydaných certifikátech. U soukromých dat může uživatel sám rozhodnout, jestli tyto informace chce zveřejnit. Adresáře jsou z pravidla několikrát zálohované, aby se minimalizovala možnost ztráty registračních a certifikačních dat. Navíc je tím usnadněna práce celé certifikační autority *CA*, která při výpadku serveru nemusí čekat a jednoduše ověří data v jedné ze záloh.

Databáze uživatelů navíc může obsahovat informace o poskytovaných službách, rozšířeních certifikátu a doplňkových funkcích, které jsou kritické pro zaúčtování služeb. Slouží tedy také jako zdroj dat pro audit a fakturaci služeb.

2.5.4. Archiv CRL

Tato část je kritická pro správné fungování celého certifikačního systému. Archiv odvolaných certifikátů musí být vždy dostupný, je proto uveřejňován několika kanály. Například na webu certifikační autority nebo pomocí protokolu *LDAP*. Při nedostupnosti *CRL* ztrácí ověřovatel možnost porovnání aktuální platnosti certifikátu a tím pádem ani pravost podepsaných dokumentů.

2.5.5. OCSP server

Jedná se o aktivní službu ověřování stavu certifikátů, která není, na rozdíl od listu odvolaných certifikátů *CRL,* zveřejňována periodicky jako seznam, ale je to okamžitě aktualizovaná databáze momentálně odvolaných certifikátů. *OCSP* server se prokazuje certifikátem C_{OCSP}, vydaným certifikační autoritou *CA,* který obsahuje unikátní rozšířené použití klíče, ve kterém je explicitně vyznačeno, že se jedná o *OCSP* službu a ta je oprávněna udávat aktuální stav certifikátu *C*. Tento status ověřuje ve spolupráci s certifikační autoritou *CA,* respektive jejím orgánem definovaným v certifikační politice.

3. Teoretické způsoby zneužití certifikačních systémů

Jelikož prolomení asymetrické šifry je velmi složité a prolamování klíčů se ztěžuje ruku v ruce se zesilováním ochrany, je lehce domyslitelné, že zrovna na tomto poli se zásadní bitva bezpečnosti neodehraje. Jednodušší možnost, a ve své podstatě i mnohem efektivnější, je podvržení samotných certifikátů. Zásadně ale záleží na „směru“, kterým se má podvod ubírat. Pokud se jedná o přístup do aplikace, kdy se uživatel má prokazovat platným certifikátem *C* dané certifikační autority *CA*, je situace poněkud složitější, než v případě získávání důvěry uživatele, držitele certifikátu *C,* pomocí zfalšovaného kořenového certifikátu *CA* (self-signed). Jedná se tedy o možnost kompromitace samostatné certifikační autority *CA*, čímž se může útočník stát dříve zmiňovaným útočníkem Man-In-the-Middle.

3.1. Ověřování kořenového certifikátu

Jelikož je *self-signed* kořenový certifikát C_{CA} certifikační autority *CA* vybaven veřejným klíčem VK_{CA}, kterému přísluší podepisující soukromý klíč SK_{CA}, nepřinese ověřování certifikátů valnou bezpečnost. V podstatě se touto kontrolou pouze ověří integrita datové struktury. Veřejný certifikát C_{CA} je navíc volně ke stažení na internetu, resp. redistribuován po lokální síti, za účelem co největší proklamace certifikační autority *CA*. Podvrhnout takový kořenový certifikát tedy není teoreticky nemožné, stačí získat originální kořenový certifikát C_{CA}, vygenerovat vlastní dvojici klíčů SK_{CA} a VK_{CA}, vytvořit nový, falešný, certifikát C_{CAX} s použitím informací z C_{CA}. Tyto informace nejsou nijak zabezpečené. Jsou pouze podepsané pomocí SK_{CA}. Pomocí takto získaných informací se tedy teoreticky lze vydávat za certifikační autoritu CA_X. V této fázi může potenciální útočník zneužít nově vytvořenou certifikační autoritu CA_{CAX} ke dvěma možným útokům.

3.1.1. Zfalšování uživatelského certifikátu C_X

K tomu stačí vykopírovat všechny informace z dostupného certifikátu *C,* dále vytvořit novou dvojici klíčů SK_X a VK_X a jelikož je útočník majitelem certifikační autority CA_X, může vytvořit nový certifikát C_X a bez potíží ji podepsat pomocí soukromého klíče SK_{CA}. Tím získá falešný certifikát uživatele, kterým může podepisovat dokumenty v jeho jménu.

3.1.2. Zneužití certifikační cesty a vnucení nového certifikátu uživateli

Pokud se útočníkovi, s vytvořenou certifikační autoritou CA_X podaří převzít kontrolu nad síťovou komunikací [2], může vnutit uživateli informaci, že jeho certifikát musí být obnoven

[2] V podstatě stačí filtrovat komunikaci uživatele *X* tak, aby přebíral útočníka *Y* jako bránu. Např. pomocí metody ARP spoofing [24]

z důvodu prošlé doby platnosti, nebo možné kompromitace. Jak již bylo řečeno v kapitole Vydání certifikátu, pár asymetrických klíčů *SK* a *VK* by si měl uživatel generovat sám. Nicméně, ne mnoho uživatelů se v certifikačním systému vyzná a tak je možné mu vnutit již vytvořený certifikát a eliminovat tak jeho povinnost vytvoření vlastního páru *SK* a *VK*. Většina nezkušených uživatelů nezapochybuje a rádi se vyhnout nutnosti generovat asymetrické klíče sami.

Obě tyto metody ale vyžadují, aby se jednalo o místní certifikační autoritu *CA*, která nevyžaduje křížové ověřování důvěryhodnosti cross-signed a, v případě vnucení nového certifikátu, aby uživatel přijal certifikační autoritu CA_X mezi důvěryhodné. Pokud certifikát C_{CA} byl podepisován křížově jinou certifikační autoritou CA_B doje ke kompromitaci falešného certifikátu C_{CAX} a ten by byl okamžitě vystaven na listinu *CRL* původní certifikační autority *CA*, čili znehodnocen. Jiná situace by ale mohla nastat v případě vystavení takové certifikační autority CA_X v rámci firemní sítě. Pokud se taková věc podaří a uživatelé vlastnící certifikáty *C* nejsou dostatečně opatrní a informovaní, je možné těmito metodami způsobit vážné problémy.

3.2. Vnucení vlastního certifikátu

Jelikož se dnes na internetu dá vydělat v podstatě na všem, není divu, že se rozmohly webové stránky s všelijakým pochybným obsahem a především spamová, nevyžádaná, pošta. Lze tedy předpokládat, že mezi různými podvody je různá spojitost. Pokud odesilatel pošty nepoužívá důvěryhodný MTA server, resp. jeho MTA server nevlastní certifikát *C* podepsaný důvěryhodnou certifikační autoritou *CA*, je tato pošta ve většině případů přesunuta mezi spam nebo podrobena jiným způsobům verifikace, např. statistické analýze podle Thomase Bayese. [16] Tento způsob identifikace serveru se nazývá whitelisting. Pokud se spamové poště, resp. podvodné internetové stránce, podaří získat důvěryhodný certifikát, ve většině případů ji nebude žádná bezpečnostní služba blokovat a může tak plně nabídnout a rozvinout svůj „potenciálně nebezpečný“ obsah.

3.2.1. Zneužití důvěryhodného certifikátu

Jednodušší ze dvou možných technik je „počestné“ získání důvěryhodného certifikátu *C* vydaného důvěryhodnou certifikační autoritou *CA*. Server MTA vlastnící tento certifikát bude potom rozesílat poštu, jež bude na základě whitelisting-u označována jako „neškodná“. To stejné platí pro webový server, který se nebude prohlížeči jevit jako „potenciálně“

nebezpečný [3]. Pokud tedy uživatel vidí certifikát *C* jako důvěryhodný a rozhodne se tomuto stavu věřit, může dát všanc svoje soukromé údaje.

3.2.2. Proklamace vlastního certifikátu

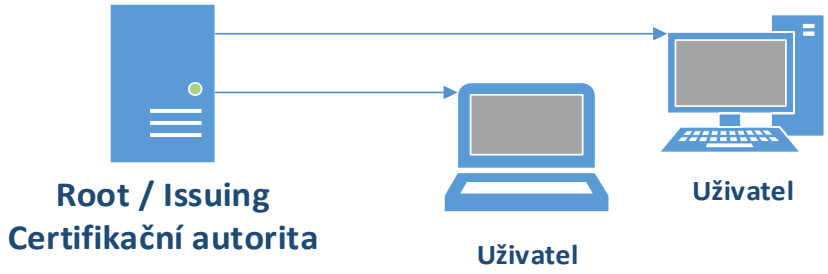
Většina prostých „nebezpečných“ stránek se neobtěžuje získáváním důvěryhodného certifikátu serveru *C* od důvěryhodné certifikační autority *CA*, ale vsází na neznalost uživatelů a automatizaci instalace certifikátů do systému. Při hledání na internetu potom stačí, aby stránka obsahovala zajímavý odkaz, a najde se velmi mnoho uživatelů, kteří certifikát bezmyšlenkovitě přidají mezi důvěryhodné nebo budou ignorovat varování prohlížeče. Opět potom stačí, aby stránka vyžadovala nějaké soukromé informace, ty potom může distribuovat dál nebo je sama použít nebo v kombinaci s předchozí metodou v rámci tzv. phishingu [17].

Obě metody vyžadují především neznalost uživatele, čehož je na internetu hojně využíváno. Nic to ovšem nemění na tom, že jde o velmi nebezpečné díry v bezpečnostním systému. Stačí jedno špatné kliknutí a počítač, co hůř i uživatel, je vystaven úniku informací, které můžou vést až k odcizení dat nebo neoprávněné bankovní transakci.

[3] Pokud neobsahuje vyhledávačem blokované odkazy, slova atp.

4. Reálný design systémů certifikačních autorit

Rozložení certifikační odpovědnosti v reálné topologii je poměrně zajímavou věcí. Nejbezpečnější je vždy to, co není připojeno, nebo alespoň není vidět. I v rámci zabezpečení WLAN sítí je běžnou praxí vypínat SSID broadcast. Nejedná se sice o prvoplánový zabezpečovací mechanismus, který síť nějak zabezpečí, ale co není vidět, nesvádí k prolomení. Zneviditelnění by tedy logicky mělo přinést svůj přínost na zabezpečení. Nicméně odpojením ROOT certifikační autority od přístupu k síti (internetu) znemožní její napadení a všechny jí vydané certifikáty jsou tedy nepopiratelné. Nový certifikát tedy může být vydán pouze lokálně a s nejvyšším možným dozorem. Nejcitlivější data tedy zůstávají „ukryta“ v nedostupném úložišti.



Obrázek 4-1 Single-Tier hierarchická struktura

4.1. Rozdělení typů designu

Při tvorbě certifikační topologie je vždy třeba se rozhodnout poměr mezi zvýšeným bezpečím a zvýšeným pohodlím. Možnosti topologie certifikačních autorit jsou poměrně rozmanité a je možné si sestavit topologii přímo na míru. Nicméně ověřování pomocí certifikátu může způsobit značné zpomalení komunikace. Například v rámci enterprise sítě, kde se počítá s jednou ROOT certifikační autoritou pro celou firmu, nebo alespoň pobočku, bude akumulačně nabírat zpoždění při ověřování. Nejedná se samozřejmě jen o zpoždění výpočetní, ale především přenosové. Nejzákladnější dělení je tedy možné z hlediska hierarchické stavby na následující tři typy.

4.1.1. Single Tier – jednoúrovňová struktura

Tento systém se skládá pouze z jednoho serveru – certifikační autority. Ta se chová zároveň jako ROOT, kořenová, a ISSUING, vydávající certifikační autorita. ROOT certifikační autorita je zde v podstatě virtuální, kdy se pomocí self-signed certifikátu se server prohlásí za kořenovou certifikační autoritu a dále se chová jako vydávající autorita, která na přání služeb nebo uživatelů může vydávat další certifikáty. Tato struktura se používá především tam, kde není třeba vyššího zabezpečení a důležitá je především rychlost a flexibilita.

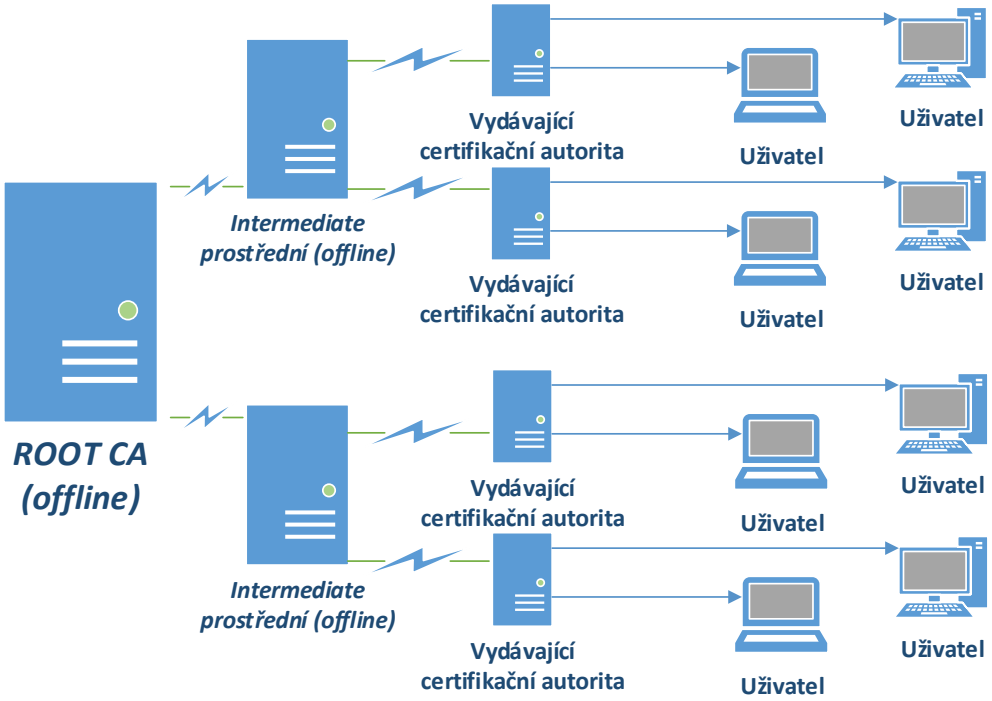
Zvýšení bezpečí lze dosáhnout pomocí HSM – hardwarovým zabezpečovacím modulem, který bude chránit primární klíč ROOT autority.

Obrázek 4-2 Two-tier hierarchická struktura

4.1.2. Two Tier – dvouúrovňová struktura

Při potřebě vyššího zabezpečení se ve většině firem používá dvou úrovňové zabezpečení, které naprosto postačuje standardním potřebám komerčního zabezpečení. Kořenová certifikační autorita ROOT se použije pouze pro vydávání podřízených certifikátů dalším serverům a dále se na ověřovacím procesu neúčastní. Tím se přidává značná porce bezpečnosti. Nejdůležitější část hierarchie, primární klíč kořenové autority, je totiž bezpečně uložen na offline serveru kořenové certifikační autority a není dostupný jinak, než lokálně. Tím je zaručena bezpečnost a identita podřízených serverů, které se starají o vydávání koncových certifikátů. Toto řešení zvyšuje flexibilitu hierarchie a může rozdělit úlohu vydávajících, ISSUING, autorit na více různých serverů. Tím ulehčí situaci vydávajícím autoritám, což při větším počtu přistupujících uživatelů může hrát důležitou roli. Především při představě nasazení v podnikové sféře, kde by na každou geografickou lokaci připadl jeden vydávající server, toto může výrazně snížit dobu potřebnou pro ověření identity koncového uživatele a podobně. Stejně tak je možné každému vydávajícímu serveru přiřadit jinou úroveň zabezpečení. Jedná se o zabezpečení konkrétní skupiny, ne celé společnosti. Bezpečnost je tedy možné upravit, podřídit použitému prostředí. Například kratší klíč pro ověřování uživatelských mobilních zařízení výrazně ušetří výpočetní náročnost při velkém množství zařízení. S rozdělením odpovědnosti na další servery se ale váže jedna nepříjemná věc a to, podpis seznamu odmítnutých certifikátů *CRL*. Ten musí být vydáván pravidelně, tedy i kořenová ROOT autorita musí pravidelně podepisovat jeho důvěryhodnost. Z hlediska praktické topologie je tento model poměrně zajímavý. Nemusí se jednat a další, plně vybavený server. Pro vydávání podřízených certifikátů, které není třeba vydávat pravidelně, postačí, aby kořenová ROOT certifikační autorita byla, společně s operačním systémem,

nainstalována na pevný disk a ten potom jednoduše odpojen od zdroje napájení, nebo například uložen bezpečně v sejfu.



Obrázek 4-3 Three-tier hierarchická struktura

4.1.3. Three Tier – tříúrovňová struktura

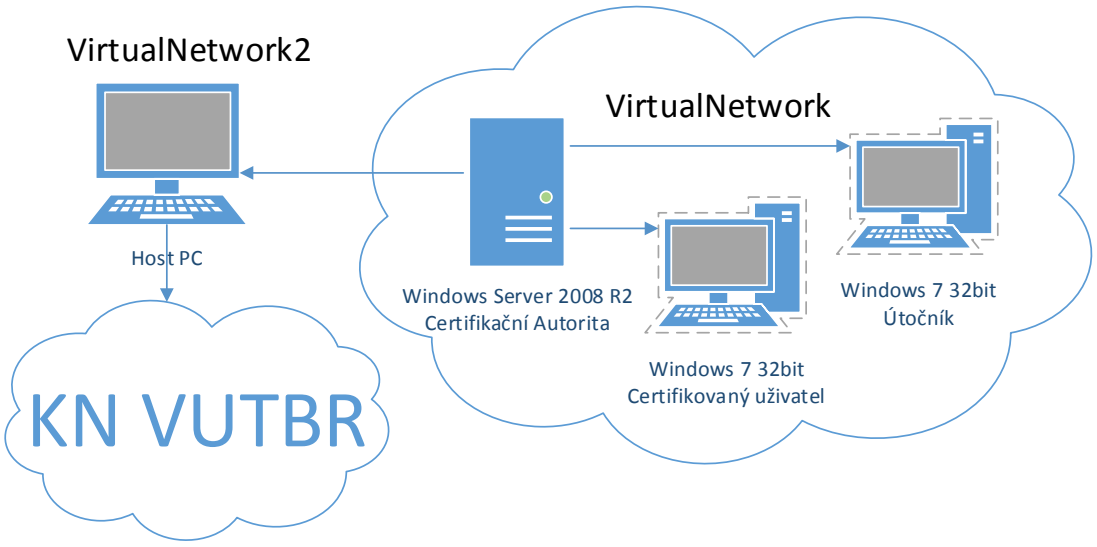
Rozdíl, mezi dvouúrovňovou a tříúrovňovou strukturou je vnoření další vrstvy CA_I mezi vydávající CA_V a kořenovou autoritu *CA*. Tento krok může mít několik důvodů, kdy jedním z nich je jednodušší správa konečných vydávajících *C*. Při kompromitaci několika soukromých klíčů vydávajících CA_V je mnohem jednodušší znehodnotit je na úrovni střední CA_I, přičemž další větve nejsou postiženy a mohou dále plnit svou funkci. Dalším důvodem je použití střední vrstvy jako autoritu certifikační politiky. Ta se stará o ověřování dat předtím, než vydá certifikát. Ověřování ale probíhá pouze na administrativní, ne technické úrovni. Tím dokáže kontrolovat, jaké certifikáty jsou vydávány.

Teoreticky se s každou další přidanou úrovní zvyšuje bezpečnost celkové struktury certifikační cesty. Zároveň se zvyšuje flexibilita a možnost pružně měnit design. Zároveň se bohužel zvyšují nároky na ovladatelnost a celkovou správu systému. Nemluvě o nákladech na tvorbu.

4.2. Zamýšlená testovací topologie

V rámci praktické části bylo původně zamýšleno sestavit prakticky fungující hierarchii při použití technologie poskytované společností Microsoft. Z dostupných zdrojů (fakultní

přístup do MS portálu MSDNA) byly získány instalační soubory pro systémy Windows Server 2008 R2, Windows Server 2012 R2, Windows 7 32bit. Dále byl použit open source software GNS3 pro emulaci síťových aktivních prvků. Virtuální prostředí, zprostředkující uživatelské stanice, bylo realizováno pomocí software společnosti Oracle (VirtualBox). Pro účely směrování komunikace virtuálních stanic bylo zamýšleno použit software společnosti Cisco. Systém IOS pro směrovače a přepínače byl nahrán do virtualizačního prostředí GNS3. Toto logické propojení jednotlivých virtuálních strojů potom mělo simulovat reálné prostředí a umožňovalo by sledování síťového provozu v co nejreálnější míře. V rámci snahy o kompromis mezi reálnou strukturou a přiměřenou obtížností návrhu certifikační cesty, byla zvolena jedno úrovňová certifikační struktura. Její topologie je vyobrazena na *obr. 4-4*

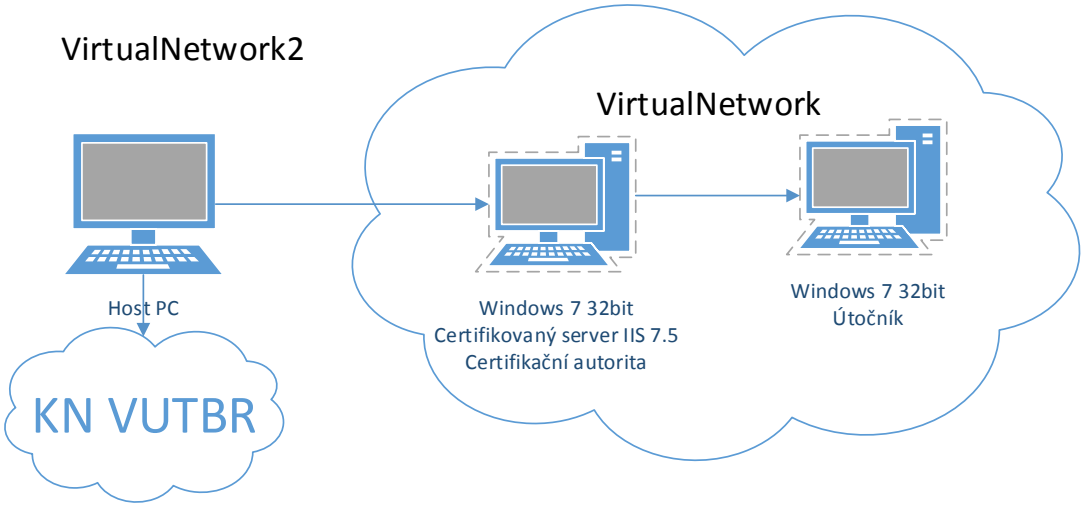


Obrázek 4-4 Zamýšlená topologie virtuálních stanic

Nejdůležitější prvek, MS Windows Server 2008 R2 byl záměrně vytvořen jako centrální bod virtuální topologie, kvůli možnostem správy a rozšíření. Bohužel, finální sestavení topologie a certifikační struktury byla ztížena množstvím problémů, např. ustavením důvěryhodnosti domény, sestavení Active Directory, Group Policy a dalších nutných služeb. Finálního stádia, tedy funkčnosti celkové certifikační autority, tzn. *CRL*, *CA* a *RA* na platformě Windows nakonec bohužel nebylo dosaženo z důvodu nedůvěryhodnosti domény v rámci celku struktury a kvůli nutnosti sestavovat pravidla a politiky pro tvorbu Active Directory a také Domain Services neobsažená problematikou zadání práce. Nicméně, pro ověření vlastností certifikačního systému ani není nutné její nasazení. Certifikační autorita poskytnutá Active Directory Domain Services může být nahrazena aplikací třetí strany.

4.3. Výsledná testovací topologie

Nakonec byla sestavena finální topologie odpovídající *obr. 4-5*. Nejjednodušší cesta k získání důvěryhodného certifikátu se jevila možnost získání certifikátu na dobu odvíjející se od investované částky. Snaha byla použít dočasný certifikát (90 dní platnosti) vydaný společností COMODO SSL, ten ale vyžadoval právě výše zmíněnou veřejně dostupnou doménu, jíž nebylo, ani přes použití IIS služeb doinstalovaných na stanici se systémem Windows 7 32bit, dosaženo. V rámci komerční Certifikační důvěry se však jedná o logický krok. Eliminuje se tím zneužití parametru DN – Domain Name. Bez ověřování by přeci bylo jednoduché vyžádat si certifikát pro doménu, která již existuje, duplikovat její přihlašovací stránku a prohlásit ji za veřejnou doménu firmy. Pomocí zmiňované metody *phishing* poté například sbírat data od uživatelů.



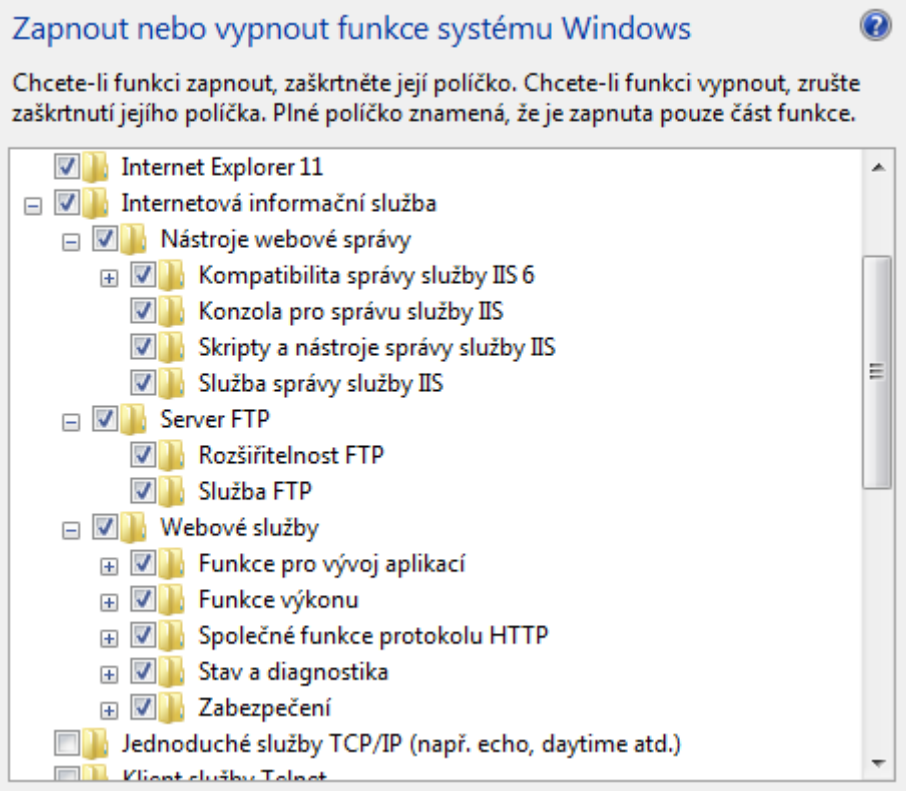
Obrázek 4-5 Finální použitá topologie

Certifikační autorita Comodo SSL, vyžaduje pro vydání certifikátu šifrovanou formu žádosti, CSR (certificate signing request). Tato šifrovaná žádost, může být vydána pomocí Microsoft IIS služby. Ta přímo umožňuje sestavení této žádosti pomocí grafického menu. Výstupem je textový soubor obsahující hash otisk žádosti ve tvaru odpovídajícím schématu:

-----BEGIN NEW CERTIFICATE REQUEST-----
MIIESzCCAzMCAQAwYzELMAkGA1UEBhMCQ1oxDTALBgNVBAgMB
EJSTk8xDTALBgNVkAvVqdRvorbFBT3fQxpSf5oIMWADbP8xuQkCfxW
……
AbwB2AGkAZABlAHIDAQAwgc8GCSqttsd5G4t5h+QJ4gJR0nT5S6TG
Wa4T75s6cTaDFB/cUd/8B/62n2FQA03eDrWct3j+Pf01kPoFjRtNE8H0
-----END NEW CERTIFICATE REQUEST-----

4.4. Vystavení žádosti CSR

Internetová Informační služba ve verzi IIS 7.5 byla nainstalována na stanici Windows 7 CA, v rámci topologie dle *obr. 4-5*. Webová služba byla instalována v plném rozsahu, se všemi vnořenými možnostmi a pod-službami. Společně s webovým rozhraním byla nainstalována i služba serveru FTP, viditelné na *obr. 4-6*. Nastavení webové služby bude detailně popsáno později. Nyní je cílem získat žádost o vydání certifikátu *CSR*.



Obrázek 4-6 Instalace webové služby IIS

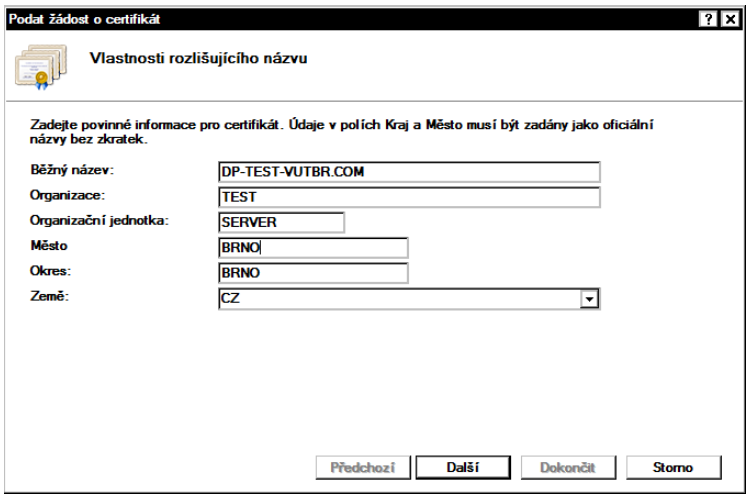
Žádost je vystavena pomocí funkce localhost serveru v menu IIS a možnosti Certifikáty Serveru. Zde je možné spravovat veškeré certifikáty serveru a vytvářet nové žádosti.

4.4.1 Informace pro certifikační žádost

V testovací konfiguraci byla pomocí služby Microsoft IIS vystavena standardní *CSR* žádost o komerční certifikát pro použití s webovou hostitelskou službou. Ta sestává z několika kroků začínajících *obr. 4-7*. Nejprve bylo potřeba vyplnit základní informace jako běžný název domény (*CN*), název organizace (*O*), organizační jednotu (*OU*) a lokalizační informace jako stát (*C*), nižší organizační jednotka (*SP*), v tomto případě okres. Poslední lokační údaje je název města, lokalita (*L*). Následujícím krokem je výběr šifrovacího algoritmu a délky klíče. Společnost Microsoft pro svou webovou službu využívá metodu RSA/Schannel a DH/Schannel.

4.4.2 Metoda RSA/Schannel

Metoda založená na principu RSA tvoří klíč za pomoci hash funkcí SHA a MD5. Tvoří otisk pomocí metody CALG_SSL3_SHAMD5, kdy se zřetězí dva hash otisky. Nalevo je použit hash otisk MD5 a vpravo SHA. To vytvoří 36 bytovou hodnotu (16 byte pro MD5 a 30 byte pro SHA).

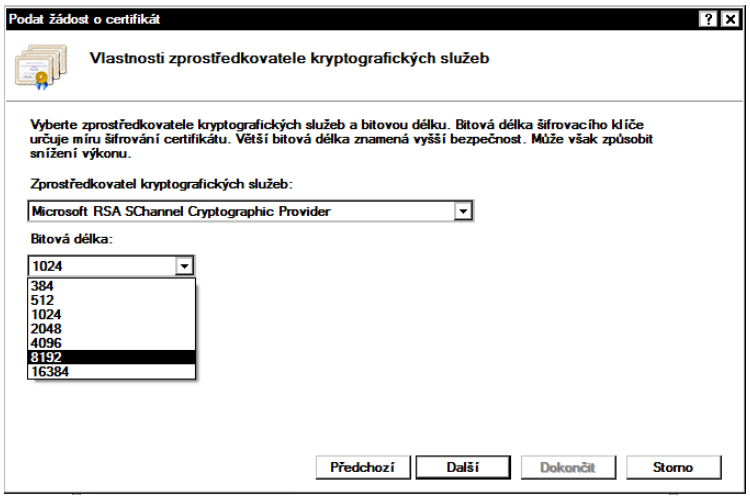


Obrázek 4-7 Základní informace pro tvorbu CSR

Tento řetězec je dále algoritmicky zpracován a výsledkem jsou šifrovaná data, podepsaná primárním klíčem RSA. Řetězený hash otisk je poté zničena [35]. Metoda ověřuje pomocí SSL 3.0 a TLS1.0 a bitové délka šifrovacího klíče může nabývat hodnot dle *obr. 4-8.*

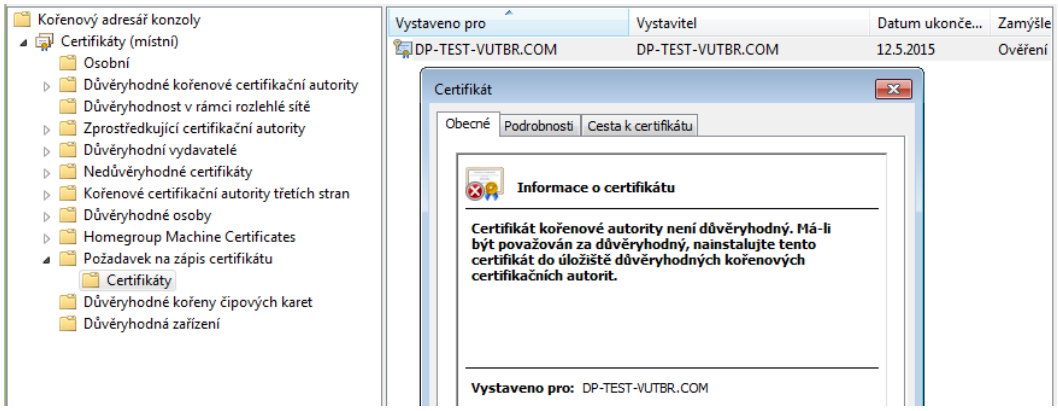
4.4.3 Metoda DH/Schannel

Metoda využívající standardní algoritmus pro ustanovení symetrického klíče DH. Tato metoda využívá hash funkci a zabezpečené podepisování dat pomocí Microsoft aplikačního model DSS – decentralizované softwarové služby. Pomocí této služby se sestavuje DH klíč a ten je dále pomocí ní spravován.



Obrázek 4-8 Výběr šifrovací metody

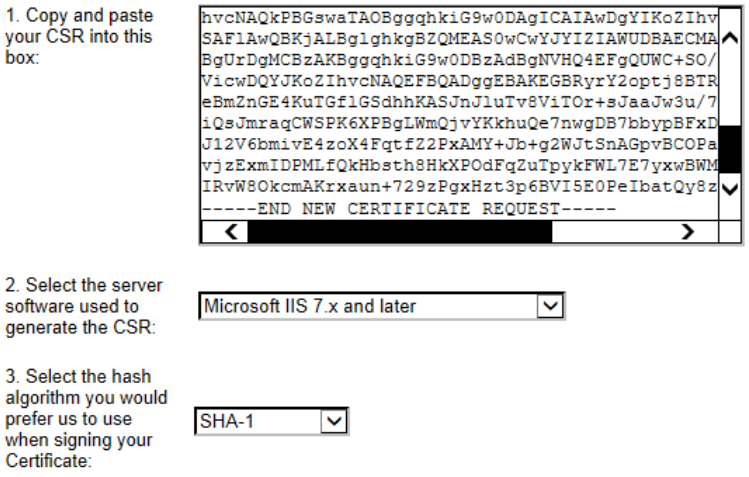
Tato metoda podporuje ověřování pomocí SSL 3.0 stejně jako TLS1 protokoly. [36] Metoda je doporučena, stejně jako DH algoritmus samotný, k ustanovení klíče na nezabezpečené lince. Bitová délka klíče je volitelná 512 nebo 1024 bit. Po dokončení žádosti *CSR* se šablona certifikátu promítne v záložce Požadavek na zápis certifikátu lokálního úložiště certifikátů. Je však nedůvěryhodná jak je vidět na *obr. 4-9*. Čeká se na vydání důvěryhodného certifikátu certifikační autoritou a jeho instalaci do systému. Systém poté porová informace, změní hodnotu žádosti na důvěryhodnou a vloží certifikát mezi Důvěryhodné v rámci rozlehlé sítě.



Obrázek 4-9 Záznam certifikační žádosti v lokálním počítači

4.5 Využití komerční certifikační autority

Jak již bylo zmíněno dříve, snaha byla testovat komerčně vydávaný certifikát. Pro tento účel by dokonale postačil dočasný certifikát, který většina komerčních certifikační autorit poskytuje pro testovací účely. Nejlepší podmínky z prozkoumaných dodavatelů poskytuje nejspíš autorita Comodo SSL. Její vstupní formulář je zobrazen na *obr. 4-10*. Jedná se o *CSR* otisk, informace o web serveru a žádaný hash algoritmus podpisu certifikátu.



Obrázek 4-10 Vstupní data pro generování certifikátu

První kontrolou při vydávání komerčního certifikátu může být ověřování pravomocí uživatele v rámci domény. Jde jednoduše o zadání standardizované administrátorské emailové adresy, na kterou by byl zaslán vydaný certifikát. Certifikační autorita k tomu využívá data ze služeb whois. Pokud indetifikuje konkrétního majitele domény, nabídne přímo možnost zaslání finálního certifikátu administrátorovi serveru. Pokud informace nenajde, jako v případě testovací domény, nabídne buď standardizované možnosti administrátorské emailové adresy, nebo zajímavější možnost, a to ověřené kontroly nad doménou - *DCV*. Toto ověření může mít probíhat trojím způsobem. [37]

4.5.1 Email DCV – tradiční

Tato metoda využívá již výše zmíněné zasílání na tradiční emailové adresy administrátora serveru. Email obsahuje unikátní potvrzovací kód a odkaz na zabezpečené stránky vydavatele certifikátu. Kliknutím na odkaz a zadáním potvrzovacího kódu prokáže dostupnost a dostatečná práva přístupu k doméně, jejíž název byl zadán pro účely certifikace. Výběr emailových adres probíhá pomocí protokolu whois fungujícím na portu 43, kdy certifikační autorita zkontroluje typické emailové adresy typu admin, tech nebo registrant. Pokud nenajde konkrétní typy adres pomocí kontroly whois, nabízí typické možnosti zobrazené na *obr. 4-11*.

4.5.2 DNS CNAME

Z poskytnutého CSR poskytovatel certifikátu spočítá MD5 / SHA hash. Je vyžadováno, aby finální poskytnutá hodnota byla uvedena jako CNAME zápis v DNS záznamu domény. Automatická funkce certifikační autority potom prohledává záznamy DNS pro požadovanou doménu. Výsledný tvar je potom ve tvaru:

$$< \textbf{MD5 hash} >.< \text{Doménové jméno} > . \ \textbf{CNAME} < \textbf{SHA1 hash} >.< \text{Doménové jméno CA} >.$$

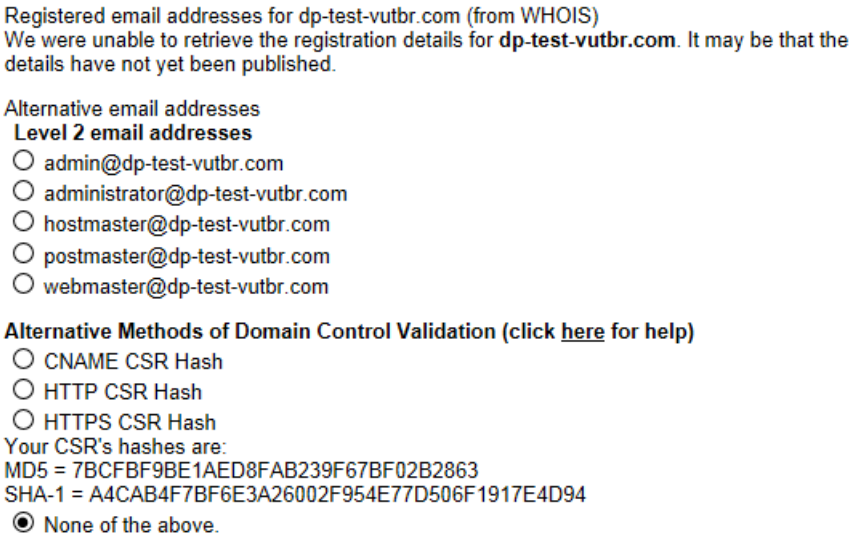
Při společné registraci více domén – *MDC / UCC,* je třeba přidat poskytnutý hash otisk před každou subdoménu.

4.5.3 HTTP DCV

Tato možnost je podobná předchozímu způsobu. Poskytnutý hash otisk je ovšem nutné umístit na server jako textový soubor. Vyžadováno je umístění přímo v root kořenovém adresáři. Musí se jednat o http server a struktura ověřovacího souboru je následná:

$$\text{http://} < \textbf{Doménové jméno} > / < \textbf{MD5 hash} >. \text{txt}$$

Obsah souboru je potom tvořen řetězcem otisku SHA1**.** Není připuštěno žádné přesměrování a verifikátor certifikační autority bude vyhledávat buď plně specifické doménové jméno *FQDN* obsažené v certifikační žádosti nebo primární doménovou úroveň serveru.



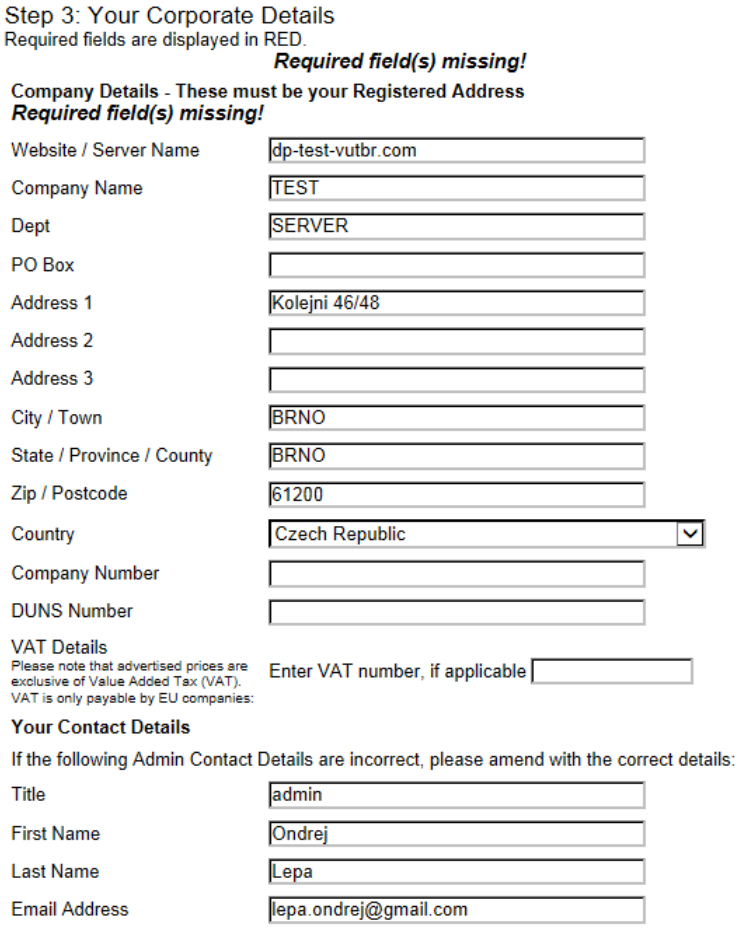
Obrázek 4-11 Identifikace domény pomocí WHOIS

4.6 Detaily žadatele

Nezbytnou součástí, kterou registrační autorita vyžaduje pro ověřování a udání totožnosti, je kontaktní adresa společnosti nebo žadatele. Jak je vidět na *obr. 4-12,* je vyžadována poměrně podrobná adresa a musí korespondovat s adresou uvedenou v žádosti *CSR*. Pokud žádost neobsahuje dostatečně konkrétní adresu, registrační autorita si ji vyžádá. Tento krok pevně svazuje žádost s konkrétním jménem, adresou a veřejně dostupnou doménou. Později není možné žádnou položku změnit, bez znehodnocení certifikátu. Po vyplnění adresy žadatele je u některých komerčních autorit vyžadováno ještě jméno a emailová adresa jednatele – Contact Details. To může být použito pro účely kontaktování ze strany *CA*. To se může týkat informací o změně certifikační polity, bezpečnostních sděleních, informacích o vydání důležitého *CRL* seznamu nebo pro komunikaci v případě kompromitace soukromého klíče vydaného certifikátu. Dalším důvodem je identifikace právní odpovědnosti. Dalším krokem je obvykle možnost (nutnost) souhlasit s certifikační politikou (*CP*) vydávající poskytující autority.

Ta obsahuje ujednání o podmínkách použití certifikátu. Zákaz využívání k protiprávní, nebo jinak nepřípustné činnosti. Zásady a ujednání správy soukromého klíče a definice vzájemné odpovědnosti v rámci smluvních podmínek používání veřejně důvěryhodného certifikátu.

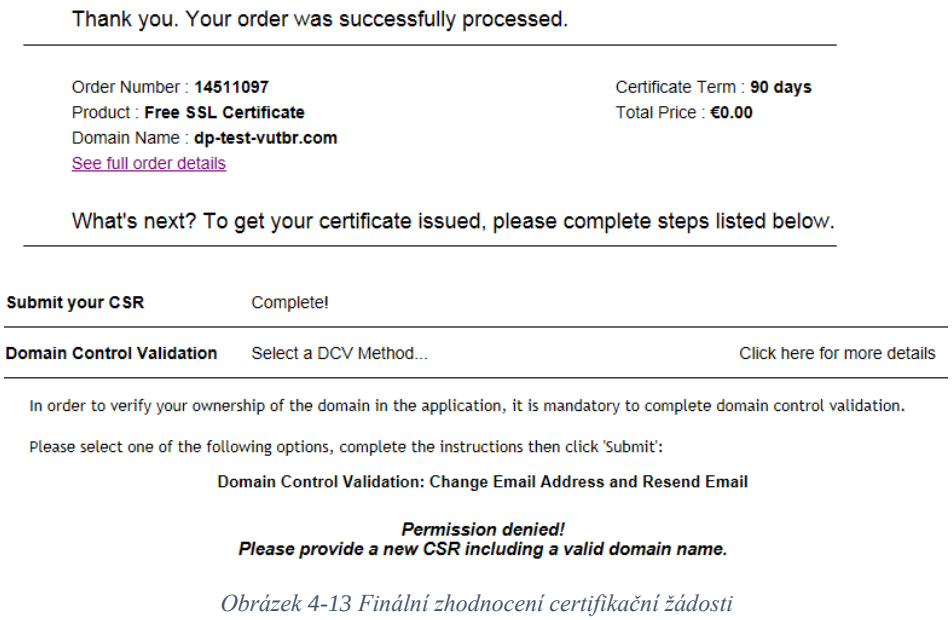
Obě strany tak na sebe berou závazek bezpečného užívání a ochrany citlivých dat. Ze strany poskytovatele certifikátu se jedná především o citlivé zacházení s kontaktními údaji žadatele, poskytnutí důvěry a veřejně důvěrohodného podpisu soukromým klíčem certifikační autority. Ze strany uživatele se potom jedná o již výše zmíněné využívání certifikátu k činnostem nepoškozujícím jméno vydávající certifikační autority. Dále se uživatel zavazuje ke korektní ochraně soukromého klíče před prozrazením. Konkrétní certifikační politiku certifikační autority Comodo SSL je možné nalézt v příloze v původním anglickém znění. Tato verze, platná k 1. květnu 2014, je vydaná 20. srpna 2009.


Step 3: Your Corporate Details
Required fields are displayed in RED.
Required field(s) missing!
Company Details - These must be your Registered Address
Required field(s) missing!
Website / Server Name: dp-test-vutbr.com
Company Name: TEST
Dept: SERVER
PO Box:
Address 1: Kolejni 46/48
Address 2:
Address 3:
City / Town: BRNO
State / Province / County: BRNO
Zip / Postcode: 61200
Country: Czech Republic
Company Number:
DUNS Number:
VAT Details
Please note that advertised prices are exclusive of Value Added Tax (VAT). VAT is only payable by EU companies:
Enter VAT number, if applicable
Your Contact Details
If the following Admin Contact Details are incorrect, please amend with the correct details:
Title: admin
First Name: Ondrej
Last Name: Lepa
Email Address: lepa.ondrej@gmail.com

Obrázek 4-12 Kontaktní adresa vyžádaná registrační autoritou

Posledním krokem je rekapitulace celé žádosti. Systém registrační autority vyhodnotí, je-li certifikační žádost kompletní, a jestli splňuje požadavky dané certifikační politikou (kontaktní údaje, prokazatelná kontrola nad doménou, atd.) a pokud je vše v pořádku, odešle žádost ke zpracování. Registrační autorita při kontrole nezaznamenala žádnou možnost ověření kontroly nad doménou a při kontrol pomocí protokolu whois nebyl nalezen záznam odpovídající uvedeným kontaktním údajům. Jak je vidět na *obr. 4-13*, žádost byla zamítnuta

na základě nedůvěryhodnosti uvedených údajů a vyzvala k poskytnutí nové žádosti s možností ověření. Tímto tedy pokus o získání certifikátu vydaného komerční certifikační autoritou zkolaboval. Toto se v podstatě dá považovat za první fázi ochrany v rámci certifikační struktury. Registrační autorita tedy správně vyhodnotila potenciální hrozbu zneužití a zabránila vydání veřejně důvěryhodného certifikátu nedůvěryhodnému žadateli.



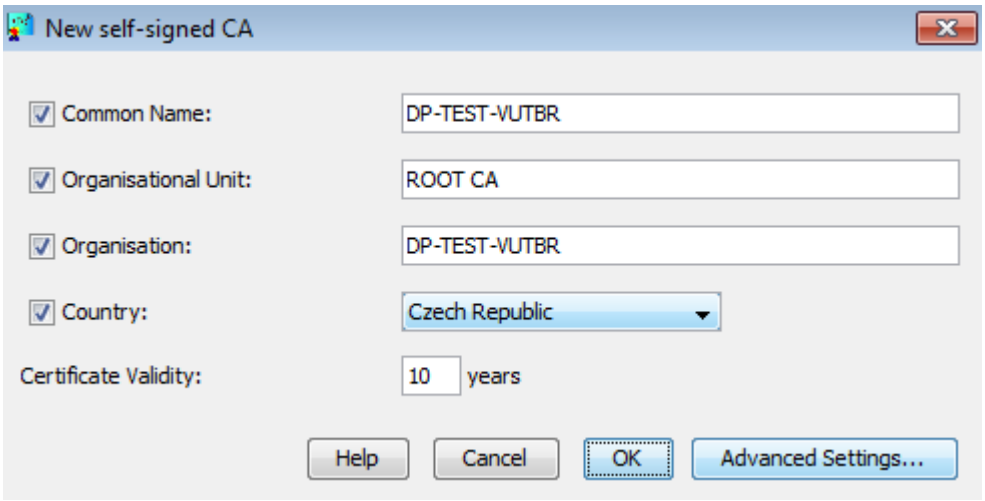
Obrázek 4-13 Finální zhodnocení certifikační žádosti

5 Práce s vlastním certifikátem

Dříve zmíněná testovací topologie byla vybavena Internetovou Informační službou IIS 7.5 a v rámci této služby byl sestaven lokální webový server s doménovým jménem podle původní certifikační žádosti. K vytvoření webového rozhraní byla využita volně dostupná šablona webových stránek a certifikáty pro testovací účely byly vydány pomocí veřejně dostupné aplikace Simple Authority. Ta, jak již bylo výše zmíněno, umožňuje v trial verzi vydat certifikáty až pro 5 uživatelů. Dále umožňuje sestavení průběžně aktualizovaného, online dostupného, listu odmítnutých certifikátů *CRL*. Aplikace nabízí dostatečně variabilní nastavení pro použití v menším podniku. V případě zakoupení placené plné verze lze hovořit o plnohodnotné náhradě jednoduché, ale i více úrovňové, certifikační autority pro nasazení v podnikovém sféře. Umožňuje totiž vydávat i certifikáty pro podřízené certifikační autority, webové servery a koncové uživatele.

5.1 Certifikační autorita Simple Authority

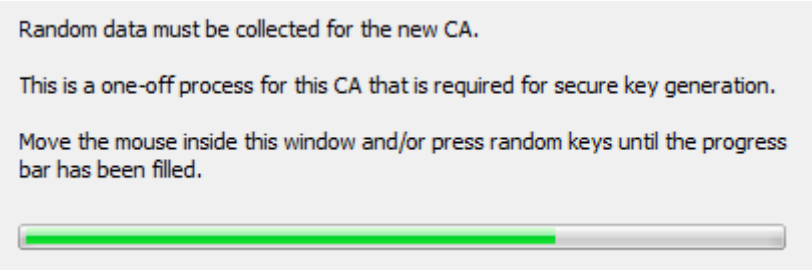
Při prvním spuštění aplikace kontroluje dostupné certifikační autority v systému. Pokud nenajde certifikát vydaný pro potřeby certifikační autority, automatický nabídne vytvoření nového certifikátu. Tvorba kořenového certifikátu v testovací verzi vyžaduje pouze základní informace o certifikační autoritě, jak je vidět na *obr. 5-1*. Rozšířené možnosti certifikátu kořenové autority jsou dostupné pouze komerční licenci. V této verzi je možné změnit délku klíče, *hash* algoritmus a další parametry certifikátu.



Obrázek 5-1 Detaily tvorby certifikátu CA

Zachytávání dat pro vytvoření soukromého klíče kořenového certifikátu využívá metody odečtu pohybu myši po určité ploše. Po teoretické stránce se sice jedná o metodu deterministickou, ale reálně je velice složité ji opakovat, protože není plně znám počáteční

stav a metoda je definována mnoha neúplně známými parametry. Jedná se tedy o reálně náhodná data, viz *obr. 5-2.*

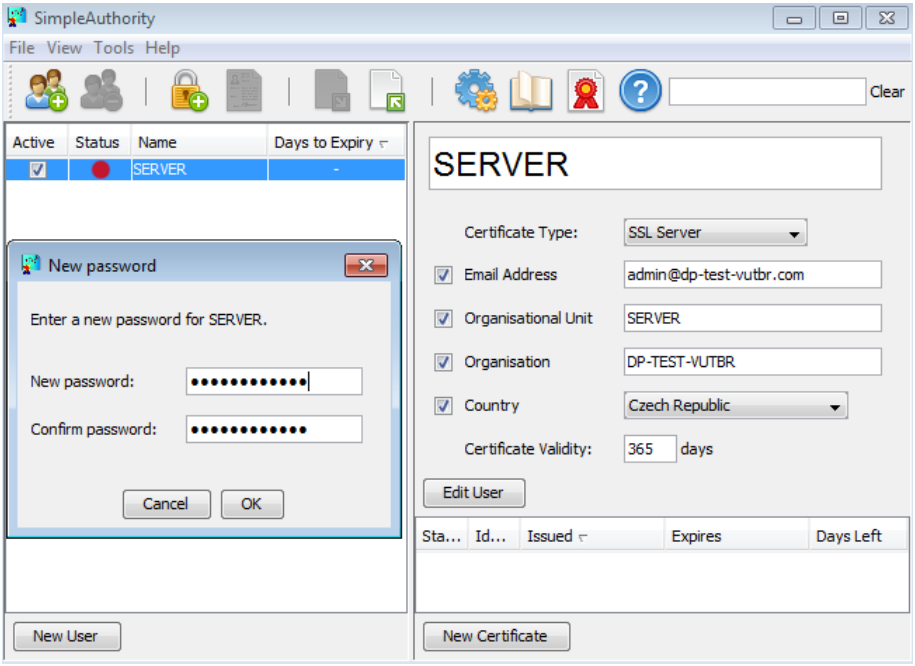


Obrázek 5-2 Zachytávání náhodných dat pro tvorbu primárního klíče

Výsledný certifikát je uložen v datové složce aplikace, automaticky generované pro uživatelský účet počítače. Ve skryté složce %User\Application Data\SimpleAuthority lze později nalézt veškeré vydané certifikáty, včetně certifikátu *CA*. Ten je doporučeno nainstalovat do systému jako Důvěryhodnou kořenovou autoritu.

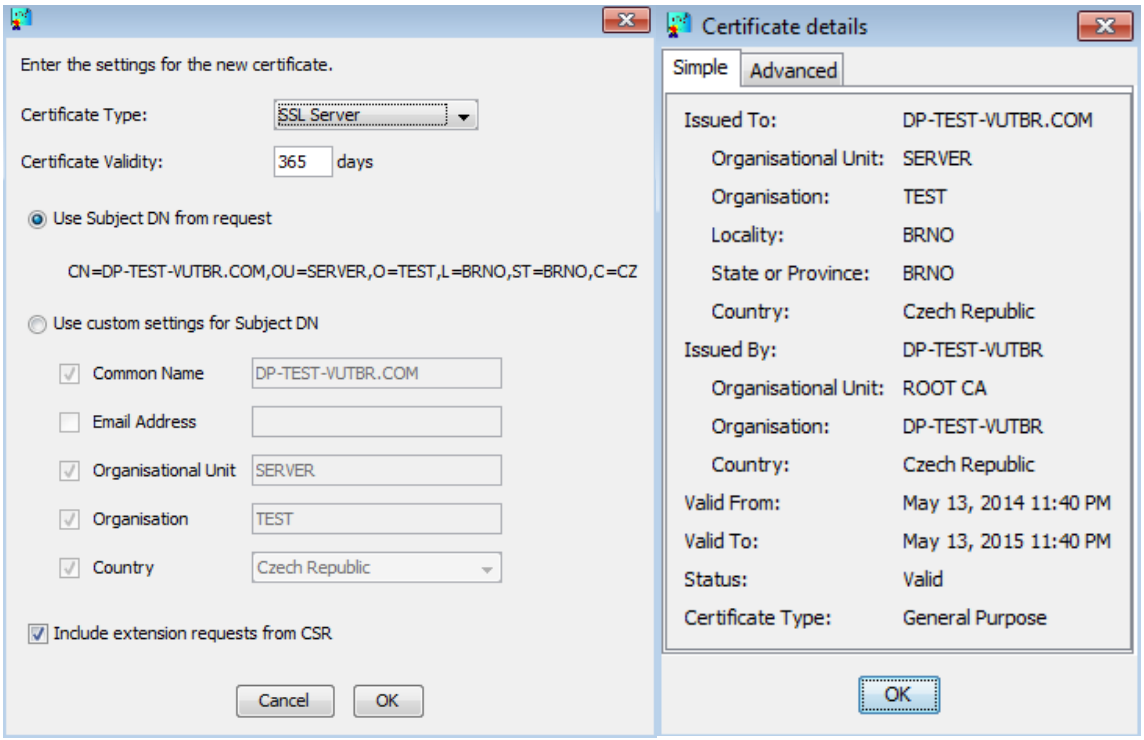
5.1.1 Vydání uživatelských certifikátů

Vydání podřízených certifikátů poskytuje základní typy pro SSL server, podřízenou *CA*, self-signed certifikát a univerzální certifikát pro předem neurčené využití. Identifikace uživatele, nebo služeb, je provedena pomocí emailové adresy, organizační jednotky, organizace a státu, ve kterém se organizace nachází. Pro testovací účely je vyžadován certifikát pro SSL server, proto byl vytvořen uživatel SERVER s nastavením odpovídajícím *obr. 5-3.*



Obrázek 5-3 Vytvoření certifikátu pro SSL server

Pro vytvoření certifikátu je nejprve nutné zadat heslo kořenové autority, a poté zvolit heslo pro certifikát nově vydávaného certifikátu SERVER. Aplikace potom nabízí jednoduchou správu vydaných certifikátů v grafickém prostředí. Na první pohled je vidět základní informace o certifikátu, tedy je-li aktivní, kolik dní zbývá do vypršení platnosti a kolik uživatel vlastní aktivních certifikátů. Vydané certifikáty jsou automaticky exportované do zvolené složky. Automaticky exportované jsou dva obecně nejpoužívanější formáty, jeden ve formátu *DER* a druhý ve formátu PKCS #12. Další možnosti exportu je možné zvolit ručně, přímo z grafického rozhraní. Podporované formáty, odpovídající specifikaci RFC 5280 definující standard X.509, jsou DER, P7C (PKCS #7) a PEM.

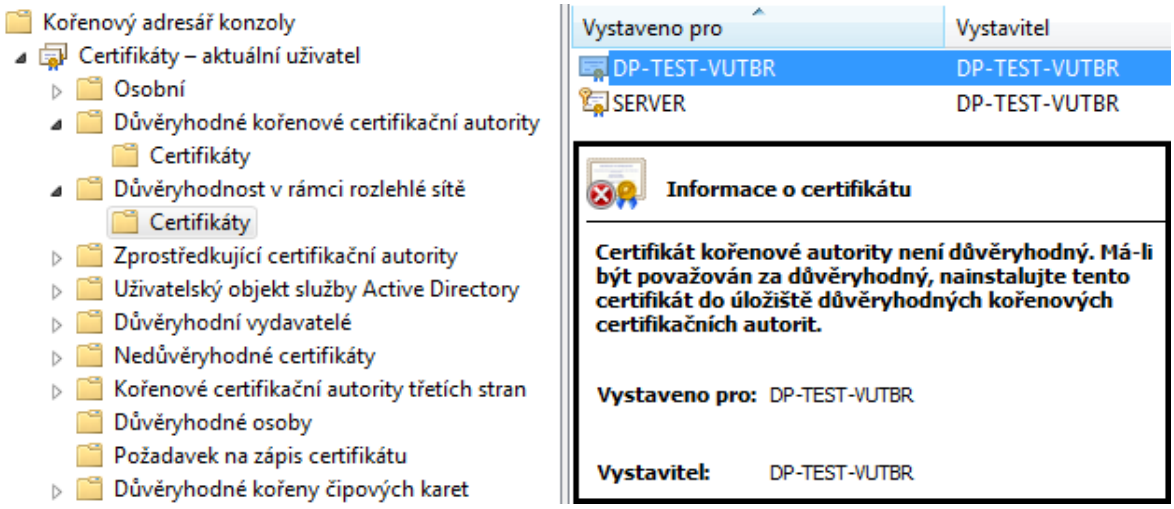


Obrázek 5-4 Import CSR do aplikace Simple Authority

Aplikace Simple Authority umožňuje také import žádosti *CSR* dodané ve formátu PEM. Díky ní je možné vytvořit certifikát serveru. Může tedy nahradit funkci komerční autority, tento certifikát však nikdy nebude uznán v rámci externí sítě. Bude uznán žádajícím klientem, ale ostatní jej budou považovat za nedůvěryhodný, protože je podepsán soukromým klíčem neznámé certifikační autority vytvořené v bodě *5.1*. V testovacím případě byla použita žádost vytvořená pro certifikační autoritu Comodo SSL. Ta byla zpracována a přijata jak aplikací Simple Authority, tak webovou službou IIS 7.5, která žádost sestavila. Certifikát je rozeznán jako typ General Purpose, viz *obr. 5.4*.

5.2 Import certifikátu do prostředí web-serveru

Prvním krokem k sestavení zabezpečeného spojení je import získaného digitálního certifikátu do Internetové Informační Služby. Ten probíhá přes záložku samotného localhost serveru ve skupině IIS a funkci Certifikáty Server. Zde je možné spravovat, přidávat nové, obnovovat, exportovat, odebírat veškeré certifikáty serveru. Samozřejmě je zde také možnost získávat *CSR* žádosti. Při pokusu o import certifikátu vytvořeného pro použití SSL serveru, viditelného již na *obr. 5-3*, došlo k neočekávané chybě. IIS vyžaduje pro ověřování certifikát, který obsahuje soukromý klíč. Ten není v defaultním certifikátu ve formátu DER obsažen, a proto je nutné použít export certifikátu ve formátu PKCS #12 s příponou **.p12*. Importem těchto dvou certifikátů do prostředí IIS serveru se certifikáty současně instalovaly také do systémového úložiště certifikátů, jak je vidět na *obr. 5-5*. Certifikát vydaný přímo za účelem certifikace SSL serveru se automaticky instaloval i do složky Důvěryhodné kořenové certifikační autority, certifikát vytvořený importováním žádosti CSR do aplikace Simple Authority byl však instalován pouze mezi certifikáty Důvěryhodnost v rámci rozlehlé sítě z důvodu nejistého původu certifikační žádost. Certifikát vydaný přímým zadáním informací do prostředí *CA* je automaticky zařazen mezi důvěryhodné.

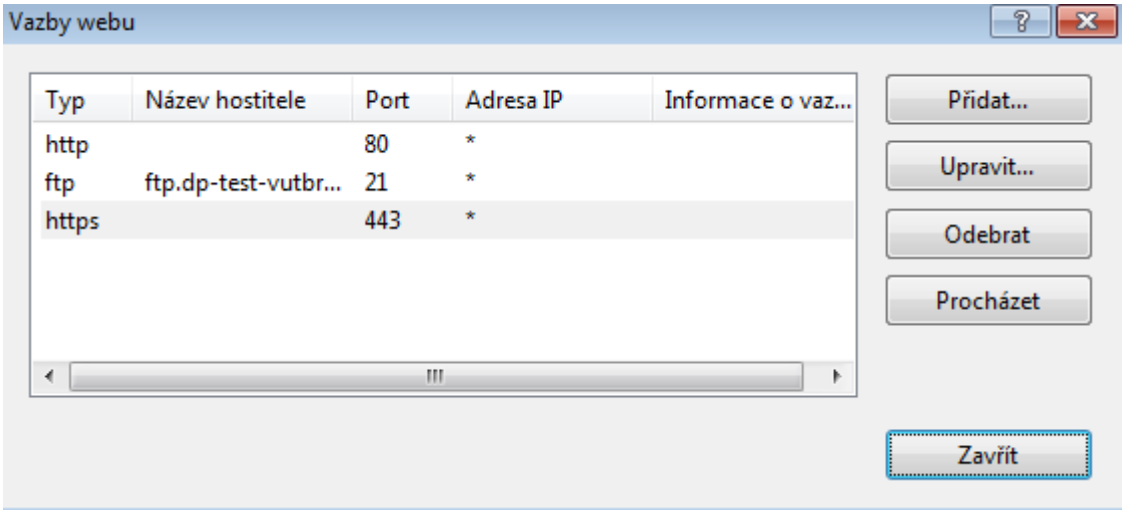


Obrázek 5-5 Úložiště certifikátů a rozdílný přístup k certifikátům

5.3 Nastavení zabezpečeného přístupu

V nastavení lokálního serveru pod nastavením Weby je třeba přidat publikování FTP. Zde je třeba nastavit vyžadování připojení *SSL* a vybrat importovaný certifikát. Dále je třeba vybrat importovaný certifikát SSL. Pro *FTP* zabezpečení byl vybrán certifikát DP-TEST-VUTBR. Možné je i možnost přiřazení IP adresy nebo URL. Dále je třeba označit možnost požadovat základní ověřování a autorizovat pouze určité uživatele. Pro tuto možnost byl

vytvořen uživatelský účet admin. Práva uživatele jsou Číst a Zapisovat. Po obnovení domácí stránky nastavení web serveru se zpřístupní skupina nastavení FTP serveru. V té je možné dále spravovat a nastavovat *FTP* server. Například přidat zprávy pro přihlášení atd.
Povolení zabezpečeného přístupu k web serveru je třeba nastavit ve více krocích. Prvním krokem je nastavení *vazby* přístupů. Tím se prováže použitý certifikát, v tomto případě certifikát SERVER vydaný pomocí Simple Authority, s nastaveným portem a IP adresou. Výsledné vazby potom odpovídají příkladu dle *obr. 5-6.*

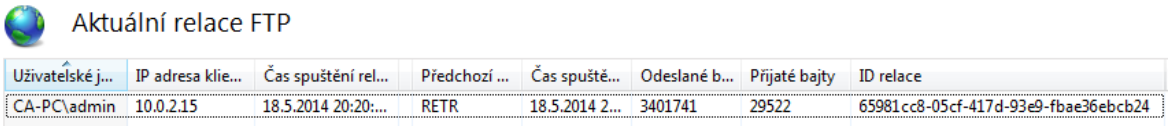


Obrázek 5-6 Vazby přístupu na web server IIS

5.4 Sledování komunikace

Při spuštěných službách serveru, sestavených podle předešlých kroků, byl spuštěn program WireShark, sloužící pro odposlech síťové komunikace. Odposlech byl spuštěn na jediném síťovém rozhraní virtuálního počítače a tímto byla monitorována veškerá komunikace virtuálního počítače s okolím. Při vyžádání a přístupu ke službám HTTPS a FTPS pomocí předinstalovaného prohlížeče Internet Explorer, resp. Total Commander, nebyla zachycena žádná *ssl* komunikace. Po chvilkovém vyšetřování a ověřování bylo vyjasněno, že použité, lokální, certifikáty slouží pouze k ověření identity a kvůli chybějící aplikaci vyžadující ověření v komunikaci, není sestaven žádný komunikační *ssl* kanál, viz *obr 5-7*. V rámci *ftp* komunikace služba IIS využívá explicitní zabezpečení, což jednoduše sestaví komunikaci na základě identity serveru, ale dále se hlásí k zabezpečení na základě ověřování uživatele. Jednoduše, stejně jako *https* služba využívá certifikát pouze k ověření identity, tak služba *ftps* bez vlastní aplikace, sestavující zabezpečené spojení, nepoužívá požadovaný *ssl* kanál, ale pouze symetrické šifrování na základě uživatelského hesla. Spojení bylo ustanoveno v rámci povoleného uživatelského účtu, a tím i přístupu k zabezpečenému *ftps* serveru byl

zabezpečen pouze nastaveným heslem. Zachycený datový tok tedy neobsahuje žádný packet zabezpečeného protokolu *ssl*.



Obrázek 5-7 Klienti FTP serveru

5.5 Zachycení certifikační komunikace

Jelikož se sestavení vlastního serveru s certifikátem zabezpečenou komunikací nezdařilo, byla zvolena náhradní možnost, a to zachytávání reálné komunikace. Pro tyto účely byl vybrán portál www.vutbr.cz a pro samotnou práci s certifikátem byl potom, kvůli potenciálním problémům s bezpečností, použit certifikát DP-TEST-VUBR a SERVER, oba použité pro původní testovací server.

5.5.1 Zachycení reálné komunikace

Při nezměněném topologii a zapnutém trasovacím programu WireShark byl v aplikaci Internet Explorer otevřen portál VUT. Při přístupu do intranetu bylo vyžadováno přihlášení a další komunikace probíhala v rámci *ssl* tunelu, sestavenému pomocí certifikátu. Jak již bylo psáno dříve, certifikát odkazuje na další certifikáty v rámci řetězce důvěry a jelikož byly při přístupu ověřovány i vyšší certifikační jednotky, jsou zachyceny i ty.

Source	Destination	Protocol	Info
10.0.2.15	147.229.2.90	TLSv1.2	Client Hello
10.0.2.15	147.229.2.90	TLSv1.2	Client Hello
147.229.2.90	10.0.2.15	TLSv1.2	Server Hello
147.229.2.90	10.0.2.15	TLSv1.2	Certificate
147.229.2.90	10.0.2.15	TLSv1.2	Server Hello
147.229.2.90	10.0.2.15	TLSv1.2	Certificate
10.0.2.15	147.229.2.90	TLSv1.2	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
147.229.2.90	10.0.2.15	TLSv1.2	Change Cipher Spec, Encrypted Handshake Message
10.0.2.15	147.229.2.90	TLSv1.2	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
147.229.2.90	10.0.2.15	TLSv1.2	Change Cipher Spec, Encrypted Handshake Message
195.113.232.72	10.0.2.15	TLSv1.2	Encrypted Alert
147.229.2.90	10.0.2.15	TLSv1.2	Encrypted Alert
147.229.2.90	10.0.2.15	TLSv1.2	Encrypted Alert

Obrázek 5-8 Záznam zachycené komunikace

Jak je vidět na *obr. 5-8*, při sestavování spojení s certifikovaným portálem VUT klientská stanice nejprve odesílá client hello packety.

Client Hello

Tyto packety obsahují, podle normy RFC5246, unikátní identifikátor *Random*. Ten se skládá ze dvou polí. Jedno o velikosti 28 byte jménem random_bytes a 32 bit blok obsahující přesný čas gmx_unix_time. [38] Ty jsou používané pro ověření platnosti certifikátu (čas) a další šifrování (náhodné hodnoty produkované zabezpečeným generátorem).

Další hodnoty obsažené v hello zprávě jsou klientem náhodně generované Session ID, CipherSuites, CompressionMethod, ClientVersion a Extensions.

Ty obsahují následující informace:

- Session ID – identifikátor relace, kterou volí klient. Je prázdné pokud se jedná o první komunikace nebo je potřeba sestavit novou žádost, např. z bezpečnostního hlediska.
- CipherSuites – seznam klientem podporovaných kryptografických služeb. První v seznamu je klientem preferovaný způsob šifrování
- CompressionMethod – seznam klientem podporovaných kompresních metod. Seřazené podle preferencí klienta.
- Extensions – klientem podporované rozšíření. V testovacím případě je například zmíněna podpora šifrování pomocí eliptických křivek a základní veřejné parametry klienta. Dále potom podporovaný podpisový algoritmus pro digitální podpis.

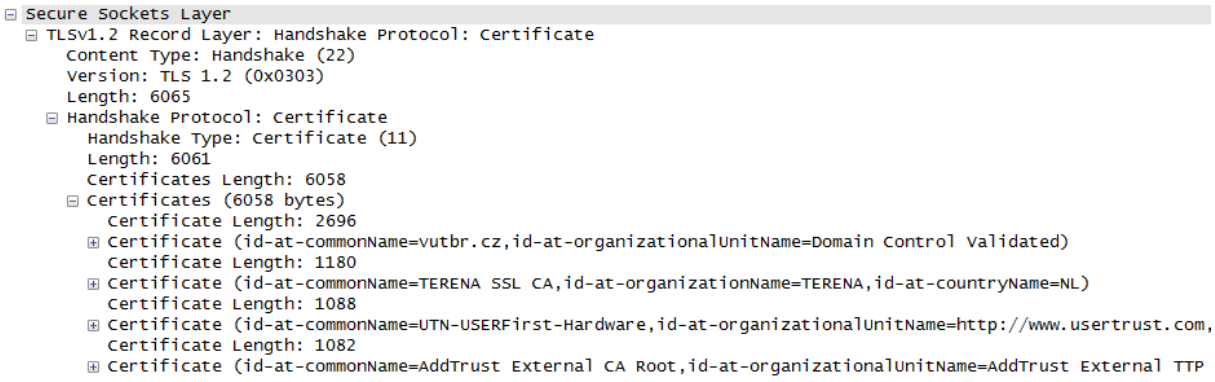
Server Hello

Tato zpráva je odeslána jako odpověď na klientovu výzvu. Vybírá z nabízených, a samotným serverem podporovaných, možností a potvrzuje využití konkrétních metod. Pokud nedojde k nalezení oboustranné shody, odešle zprávu handshake failure a zabezpečená komunikace se tím ukončí. Zpráva server hello obsahuje následující pole:

- ServerVersion – pole určuje nejvyšší, oboustranně podporovanou, verzi
- Random – obsahuje, stejně jako klientova zpráva, gmx_unix_time a random_bytes. Hodnota generovaná serverem musí být odlišná od klientovi
- SessionID – pokud v klientově zprávě nebyl zvolen identifikátor relace, server volí ID sám. Pokud klientova zpráva nějaké *ID* obsahovala, server prohledává cache, aby mohl navázat v předešlé komunikaci
- CipherSuites – zvolený mechanismus odpovídající klientovým schopnostem
- CompressionMethod – zvolená kompresní metoda podporovaná klientem
- Extensions – Využití klientem podporovaných rozšíření.

Certificate

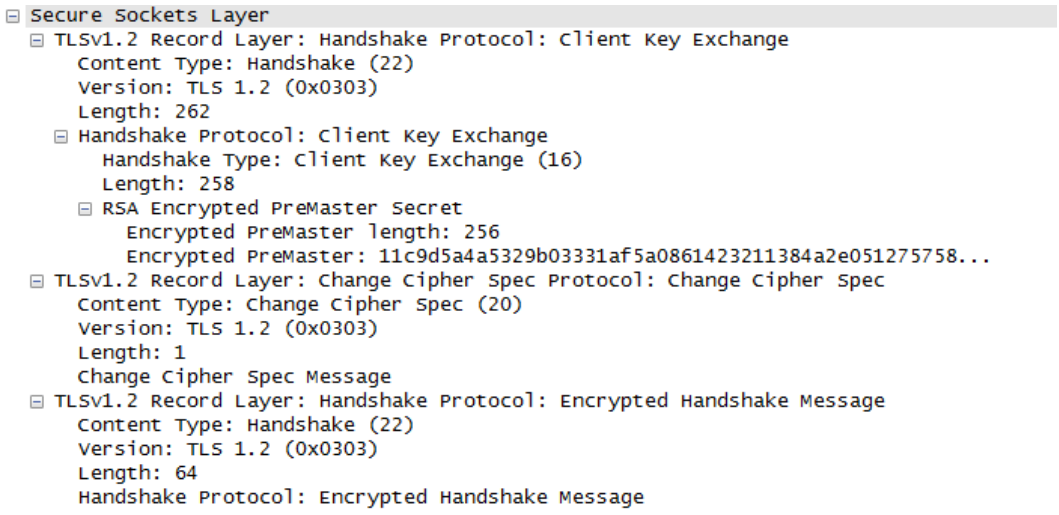
V této zprávě server zasílá klientovi svůj vlastní certifikát. Tato zpráva bezprostředně následuje po zprávě Server Hello a obsahuje všechny certifikáty v rámci řetězce, jak je vidět na *obr 5-9.*



Obrázek 5-9 Zachycené certifikáty

Při bližší prozkoumání zprávy lze zjistit např. identifikátory certifikátu, jeho délku, verzi, sériové číslo, podepisující autoritu a mnoho dalších informací. V rámci rozšíření certifikátu je potom uvedena například adresa *CRL* seznamu, identifikátor certifikační politiky, povolené použití klíče a záznamy podřazených *DNS* jmen. Seznam určuje, které podřízené domény jsou sdružené s prvotní doménou www.vutbr.cz. Protože například interní poštovní server VUT není obsažen v rozšíření, při přihlášení k serveru email.feec.vutbr.cz prohlížeč vypisuje potenciální nebezpečí.

Certifikáty lze ze zachycených dat exportovat pro další použití pomocí aplikace WireShark označením jednotlivých certifikátů a výběrem Export selected packet bytes v kontextové nabídce. Toto vybere certifikát v holé bytové formě, odpovídající *DER* kódování. Stačí přidat příponu **.der* a data se poté chovají jako importovatelný certifikát.



Obrázek 5-10 Zpráva sloužící k výměně klíče

Client Key Exchange, Change Cipher

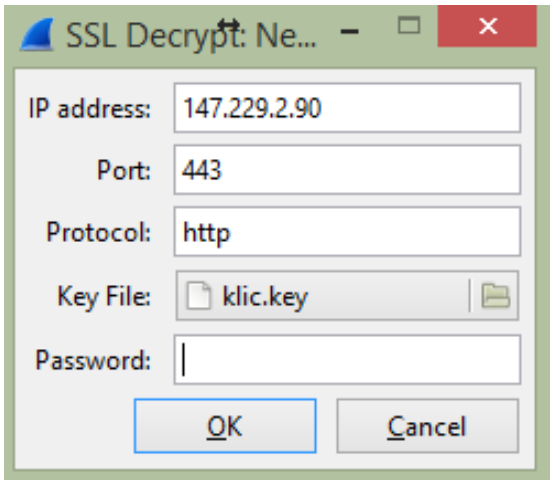
V této zprávě serveru zasílá předvolenou hodnotu k tvorbě šifrovacího klíče. V zachyceném případě šlo o RSA PreMaster Secret. Další možnost je DH Public Value při použití DH

šifrovacího algoritmu, tím se mírně změní i struktura zprávy. V rámci RSA PreMaster Secret se přenáší 48 bytů klientem generovaných dat, která jsou šifrována pomocí veřejného klíče obsaženého v certifikátu serveru.

Další informací je potom změna šifrovacích parametrů. Tato zpráva je již šifrovaná pomocí premastered klíče a proto není veřejně známo, jaká pravidla budou použita pro tvorbu finálního šifry, která bude zabezpečovat komunikaci. Potvrzením této zprávy ze strany serveru ustanovuje konečnou komunikaci a završuje ji Encrypted Handshake zpráva. Další komunikace je potom již šifrována a data jsou nečitelná. Jediná veřejná informace poskytnutá v rámci komunikace je verze používané protokolu *tls* a délka zprávy samotné.

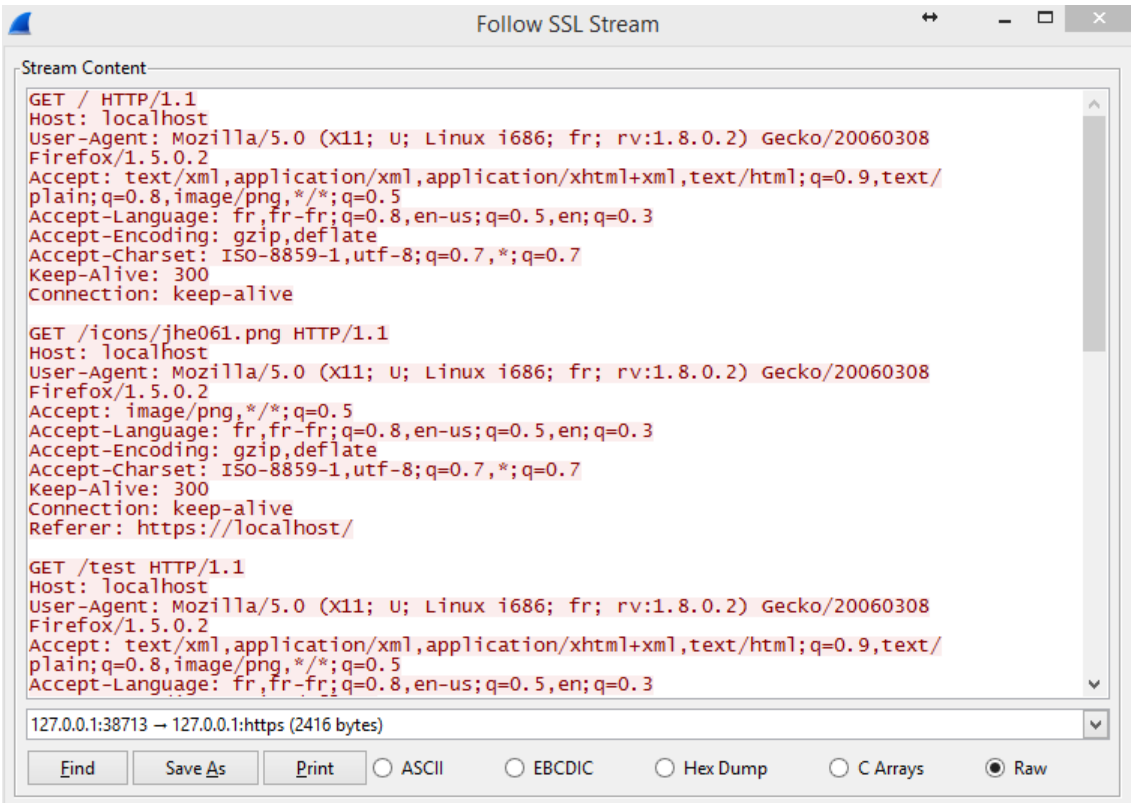
5.6 Dešifrování komunikace

Pomocí aplikace WireShark je možné také dešifrovat zachycenou *SSL / TLS* komunikaci. K tomu je ale potřeba mít informace o certifikátu. V zobrazeném datovém toku je třeba vyfiltrovat komunikaci podle protokolu *SSL*. V menu aplikace *Úpravy – Preference – Protokoly* je třeba vybrat *SSL* a přidat informace o zabezpečení, jako na *obr. 5-11*.



Obrázek 5-11 Potřebné informace o zachycené komunikaci

V podstatě se tedy jedná o v celku jednoduchý úkol. Jediná problematická část je získání souboru s klíčem klic.key. Jak z certifikátu extrahovat tento soubor bude ukázáno v další kapitole, nicméně u certifikátu, který není vlastněn samotným uživatelem je potřeba znát bezpečnostní heslo. To znemožňuje jednoduchou extrakci z veřejných, komerčních certifikátů třetích stran. Pokud jsou zadané informace správné a klíč odpovídá certifikátu, šifrovaná komunikace se stane čitelnou tak, jak jde vidět na *obr. 5-12*.



Obrázek 5-12 Dešifrovaný datový tok

Z takto dostupného datového toku je potom jednoduché získat detailní informace o uložené, ale i právě probíhající relaci. Opět je tedy potřeba zmínit vysokou citlivost na kompromitaci soukromého klíče.

6 Práce s certifikáty a openSSL

S pomocí aplikačního balíčku *openSSL* je možné pracovat s certifikáty a informacemi v nich obsažených. Tento projekt se zabývá především implementací protokoly *SSLv2/3* a *TLSv1* stejně tak, jako univerzální kryptografickou knihovnou doplňovanou a spravovanou internetovou komunitou. OpenSSL využívá veřejný licenční model, je tedy možné ji nasadit v komerčním i nekomerčním prostředí. Nejdůležitější vlastnosti openSSL projektou jsou správa a tvorba soukromých a veřejných klíčů, kryptografické operace s veřejným klíčem, tvorba certifikátů standardu X.509 [43], šifrování a dešifrování a *SSL*/*TLS* testy.

Pro analýzu struktury certifikátu je doporučeno certifikát nejdřív převést do formátu *PEM*, který používá ASCII (BASE64) data. To značně usnadňuje práci ve srovnání například s formátem *DER*. Formát *PEM* je využíván i pro certifikační žádosti a openSSL s ní pracuje nativně. Při použití formátu *DER* je v syntaxi příkazu potřeba přidat povel určující formát certifikačního souboru.

6.1 Analýza certifikátu

Prvním krokem při analýze certifikátu v prostředí openSSL je zobrazení jeho základního obsahu pomocí příkazu:

$$openssl\ x509\ \ -in\ název\_certifikatu.pem\ \ -text$$

Tímto příkazem prostředí openSSL vypíše detailní informace o certifikátu, vydávající autoritě a dalších vlastnostech. Extrakce soukromého klíče potom proběhne pomocí dalšího příkazu, který vybere otisk a uloží pouze ten:

$$openssl\ pkcs12\ \ -in\ certifikát.p12\ \ -nocerts\ \ -out\ název.pem\ \ -nodes$$

Tato operace však vyžaduje znalost hesla, které bylo použito při tvorbě certifikátu. Bez něj neproběhne správě ověření integrity a zabezpečení dat a extrakce je tím zamítnuta s hláškou:

$$"MAC\ verify\ error: invalid\ password?"$$

Kryptografický message authentication code je krátky kousek informace použité pro autentifikaci zprávy. *MAC* algoritmus akceptuje jako vstup tajný klíč a zprávu určené délky

pro ověření výstupu. Tím je chráněna jak integrita, tak i autentičnost, s možností ověření integrity a detekce změn pro toho kdo vygeneroval, nebo zaslal, tajný klíč. Při ověřování *MAC* došlo k chybě a openSSL není schopno vykopírovat otisk primárního klíče. Při zadání správného hesla potom proběhne extrakce do zvoleného formátu. Dalším krokem může být odstranění hesla ze souboru soukromého klíče, což způsobí, že při dalším zpracování není potřebě jej opakovaně zadávat. Tento příkaz uloží soubor pouze se soukromým klíčem.

$$opessl\ rsa\ \ -in\ \ název.pem\ \ -out\ klíč.key$$

S tím je poté možno dále pracovat, například pro dešifrování komunikace zachycené pomocí aplikace WireShark, jak bylo ukázáno v předchozí kapitole. Je tedy velice důležité chránit soukromý klíč k certifikátu, jinak je velmi jednoduché jej zneužít.

6.2 Úložiště certifikátů

Jelikož extrakci soukromého klíče není možné provést bez znalosti bezpečnostního hesla, zbývá možnost odebrání certifikátu, resp. nahrazení certifikátu přímo v úložišti klientského počítače. Certifikátu počítače lze zobrazit za použití správce certifikátů zadáním příkazu certmgr.msc v kolonce Spustit nabídky Start. Zde jsou vypsány všechny aktivní certifikáty v předem definovaných skupinách, jak je vidět již na *obr. 5-5*. Certifikáty jsou do skupin rozřazeny buď automaticky při instalaci, podle informací v certifikátu, nebo je možné uložit je do konkrétní skupiny, podle potřeby při importu certifikátu. Z toho ovšem není jasné, kde se nachází reálné logické úložiště certifikátů v rámci souborového systému.
Při bližším zkoumání certifikačního úložiště bylo zjištěno, že certifikáty používané systémem jsou uložené jako položky registru ve větvi

$$HKEY\_LOCAL\_MACHINE\backslash SOFTWARE\backslash Microsoft\backslash SystemCertificates$$

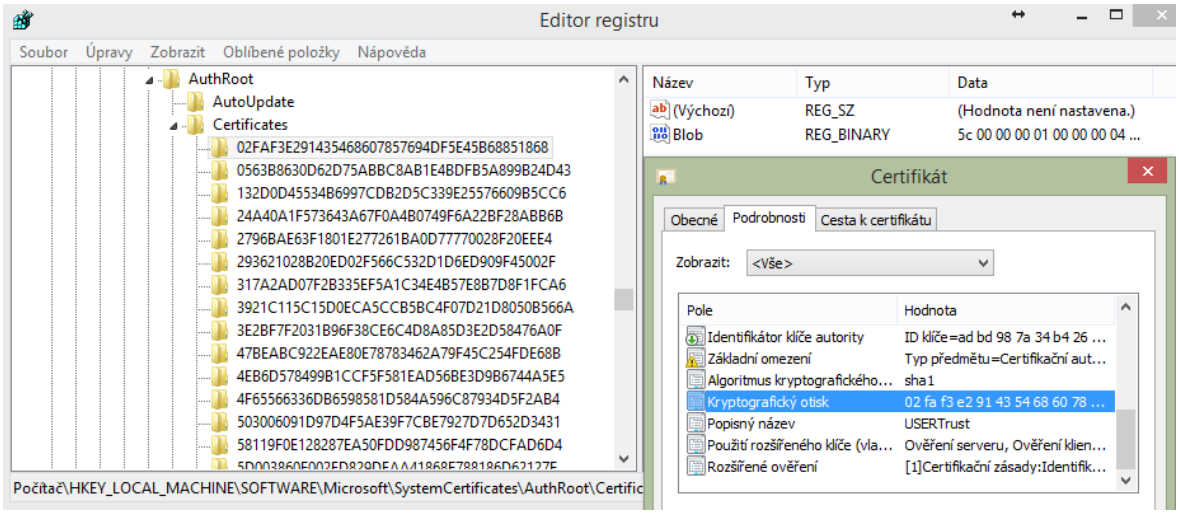
a zde jsou třízeny do přibližně stejných skupin jak je zobrazuje správce certifikátů.
Každá skupina je zde ještě rozdělena na tři podskupiny:

- Certificates – seznam samotných certifikátů.
- CRLs – seznam odmítnutých certifikátů.
- CTLs – seznam hash otisků obsahů certifikátů.

Uložení v registrech je důležitou součástí aktuálně běžícího systému, protože k nim reálně přistupuje. Pokud jsou registry upraveny, systém se začne chovat podle nastavení. Je tedy možné do registrů přidat i požadovaný certifikát pomocí exportu / importu záznamu registrů. K tomu slouží služba certutil, která importuje certifikáty z **.reg* souborů. Příkaz, který importuje certifikát je:

$$certutil - addstore - f\ zvolený\_typ\_uložiště\ název.reg$$

Nicméně, je třeba, aby tento příkaz byl proveden v *administrátorském režimu* příkazového řádku, a typ úložiště je třeba vybrat podle potřeby využití certifikátu. [39] Uložení certifikátu v registrech bohužel nevyužívá stejné pojmenování jako ve správci certifikátů, je proto nutné znát konkrétní unikátní identifikátor certifikátu. Tímto identifikátorem je v prostředí registrů hodnota kryptografický otisk ze struktury certifikátu jak je vidět na *obr 6-1*.



Obrázek 6-1 Logická struktura úložiště v registrech systému

6.3 Známé útoky na SSL

Jak již bylo ukázáno v předchozích pokusech, bezpečnost certifikátů nespočívá v ničem jiném než kumulativní bezpečnosti asymetrické kryptografie. Nejslabším článkem PKI tedy zůstává soukromý klíč certifikátu a zabezpečený kanál samotný. Pokud je útočník schopný prolomit komunikaci, na certifikační bezpečnosti založenou, zůstává certifikátu pouze vlastnost ověření identity. Nicméně komunikace může posléze převzít útočník.

Na světlo světa se dostalo několik druhů útoků na šifrovanou *SSL* komunikaci a tím byla narušena i důvěryhodnost v zabezpečenou komunikaci. Většinou se tedy jedná o převzetí

kontroly nad *https* komunikací prohlížeče klientského počítače a dalšího odchytu zabezpečených dat útočníkem. Útočník tam ani nemusí lámat samotné heslo, ale to přijde s dostatečným množstvím zachycených dat samo ve formě porovnaných dat.

6.3.1 Beast

Útok vedený proti *TLS1.0* a *SSLv3.0* oznámený v roce 2011 využívá slabosti webových prohlížečů a pomocí zachytávání relace ve smyslu převzetí odpovědnosti za probíhající komunikaci. Metoda je založena na předpovídání IV inicializačního vektoru, pomocí průniku do klientova prohlížeče. Při známosti tohoto inicializačního vektoru, vstupních dat a jejich výstupu, je z další komunikací snadné odhalit klíč.

Metoda je v podstatě založená na průniku do relace prohlížeče a odchytávání komunikace. Pro větší objem důležitých dat, je možné injektovat datové dotazy do prohlížeče a tím sledovat a upřesňovat hodnoty porovnávající vstup / výstup. Přestože je tato slabost inicializačního vektoru odstraněna již ve verzi *TLSv1.1* a dále, většina prohlížečů pouze pomalu přechází na verzi *TLSv1.1* a vyšší. [40]

6.3.2 Breach

Metoda využívající injekce zvoleného textu do klientových dotazů a meření velikosti šifrované komunikace. Tato metoda slibuje prozrazení již od několika tisíc dotazů a může být provedena v čas pod 1 minutu v závislosti na délce hesla.

Útok využívá postranní kanály a útočí tak na *https* spojení prohlížeče, kdy se zajímá především o hlavičky *http* žádostí. Jelikož tyto obsahují cookies a cookies jsou přednostním nositelem webových aplikací a tím i autentikačních informací, jedná se o závažný problém. Přestože se jedná o poměrně jednoduchý systém, je tuto metoda nutno modifikovat pro každý případ prolomení, kdy je třeba splnit následující podmínky:

- Server, na který se útočí, využívá http-kompresi.
- Server odpovídá na klientovy žádosti v těle *http* zprávy.
- Server reflektuje záznam hesla v těle *http* odpovědi.

Útoku navíc značně pomáhají odpovědi, které zůstávají stejné, např. modulo útočníkova pokusu.

Přesto, útok není cílen na určitou verzi *SSL / TLS* a nevyžaduje kompresi zabezpečené vrstvy a útok proti proudové šifře je o to jednodušší, že rozdílné velikosti těla odpovídající zprávy jsou mnohem řídší. Při použití blokové šifry, je třeba upravovat zarovnávání.

6.3.3 Crime

Metoda, stejně jako předchozí, útočí přímo na již zaběhnutou *https* relaci. Ze základu zabezpečené komunikace *SSL / TLS* většiny veřejných prohlížečů lze odvodit dostatek

informací k převzetí kontroly nad relací. Metoda *CRIME* je přímým následovníkem metody *BEAST* a popsala jej i stejná dvojice výzkumníků, J. Rizzo a T. Duong. [41] Metoda je opět založena na vyhledávání a zachytávání cookies a hledání opakující se šablony. Na rozdíl od *BEAST* není třeba kontrolovat dotazovaná data. Stačí zajistit, aby data byla odesílána společně s cookies. Útočník v podstatě využívá porovnávání zpráv při náhodných dotazech a při nižší velikosti odpovědi útočník vychází z poznatku, že útočníkův dotaz a heslo bylo komprimována společně.

Oba předešlé útoky poukazují především na slabosti ranných verzí *SSL / TLS* a výzkumníci se tím snažili urychlit prosazení především *TLSv1.1 / 1.2*. Konkrétní způsob prolomení bezpečnosti není veřejně dostupný. Obecná medotologie ale zůstává veřejná.

6.3.4 HeartBleed

V současnosti velmi diskutovaný útok zaměřený na *openSSL* a jeho kryptografické knihovny, který je o to zajímavější, že je sice známý až přibližně od dubna 2014, ale služba, která toto prolomení umožňuje, je v produkci používána již od konce roku 2011. [42]

Metoda využívá, paradoxně bezpečnostní, rozšíření *TLS / DTLS* komunikační linky, která zpřístupňuje klientovi odesílat heartbeat žádost serveru. To v podstatě znamená, že se klient optá serveru, jesli je stále dostupný, a jestli je, ať odpoví konkrétní textovou zprávou, stejnou jakou zaslal klient. HeartBleed potom využívá chyby v alokaci dočasné paměti serveru a ten nekontroluje velikost obrržené zprávy od klienta, ale reaguje pouze na ohlášenou velikost. Klient tedy odešle krátkou zprávy s informací, že se jedná o maximální možnou délku a čeká na odpověď od serveru. V normálním případě by server jednodušše přeopslal zpět klientův dotaz, ale jelikož ten je zanedbatelně krátký, server nemá na dotaz dostatečně dlouhý záznam od klienta, proto doplní zprávu informacemi z dočasné paměti. V té se ale mohou nacházet různě důležité informace, soukromým klíčem serveru počínaji, uživatelským jménem, heslem uživatele a dalšími informacemi konče. Navíc je tento útok proveditelný bez zaznamenatelné stopy. Server tedy ani nezaznamená, že byl potenciálně vystaven nebezpečí útoku a považuje toto chování za standardní.

Z povahy této metody se tedy jedná o velmi závažnou bezpečnostní díru v bezpečnosti, která byla odstraněna novější verzí a kolektiv vyvíjející openSSL prostředí nedoporučuje využívat knihovny ve verzi v1.0.1 do verze openSSLv1.0.1g, která je notována jako opravná.

Závěr

Hlavním cílem této práce bylo popsat problematiku digitálních certifikátů, certifikačních autorit a dále popis principu certifikáty zabezpečené komunikace a její sestavení. Součástí práce je komplexní popis možností certifikačního zabezpečení a sestavení certifikátem zabezpečeného servervu,včetně možnosti získání důvěrného a komerčně použitelného certifikátu, vydaného důvěrnou certifikační autoritou Comodo SSL.

První kapitola se zabývá teoretickým popisem funkcí sloužících k digitálnímu zabezpečení a jejich obecným pojmenováním. V rámci této kapitoly jsou taký vyjmenovány známé způsoby šifrování a věřejně používané standardy.

Druhá kapitola popisuje strukturu samotného certifikátu, jak je definován podle normy X.509, včetně popisu jednotlivých datových polí a jejich významu. Dále je potom popsán systematický význam certifikační autority a její teoretická funkčnost, včetně vnitřní struktury a podřízených certifikačních orgánů.

Ve třetí kapitole je nastíněn teoretický problém certifikačního systému. Jsou zde popsány teoretické slabé stránky certifikačního systému a jejich možné zneužití z hlediska kompromitace a zneužití soukromých informací.

Čtvrtá a pátá kapitola se zabývají konkrétním použitím certifikátu jako způsobu zabezpečení a ověření identity webové služby, vydáváním certifikátu komerční důvěryhodnou certifikační autoritou a v kontrastu s tím i soukromou self-signed certifikační autoritou pro laboratorní využití. Dále je zde rozebráno sestavení zabezpečené komunikace a její struktura v rámci zabezpečeného protokolu *ssl / tls* s ohledem na normu RFS 5246. Zachytávání komunikace a její dešifrování pomocí veřejně dostupných nástrojů pod platformou MS Windows. Důraz byl kladen také na logické úložiště certifikátů v rámci systému.

V poslední, šesté, kapitole je rozebrán vydaný certifikát pomocí volně dostupného prostředí openSSL. S jeho pomocí byl certifikát podroben rozboru z hlediska informací v něm obsažených a možností práce s ním. Dále jsou také zmíněny veřejně známé a popsané útoky na zabezpečenou komunikaci kanálu *ssl / tls* a jejich stručný popis. Tím byl splněn hlavní cíl práce splněny a dodržena struktura stanovených dílčích cílů.

Jako závěrečné zhodnocení bezpečnosti certifikačního systému a struktury PKI je nutné přiznat, že nejslabším článkem řetězce je lidský faktor, kdy se při prozrazení soukromého klíče certifikátu otevírá cesta pro široké spektrum možných útoků a zneužití jdoucích ruku v ruce s důvěrou, která je do komerčně užívaných certifikátů v kládána. Z hlediska

zabezpečení samotného systému se jedná o systém bezpečný, relativně jednoduše ovladatelný a s bohatou možností škálování a dělení. Samozřejmě s rostoucím stromem certifikačních autorit se zvyšuje režie potřebná k ověřování konečné důvěryhodnosti a její integrity, ale i tento neduh lze jednoduše vyřešit křížovým podpisem adekvátních certifikačních autorit. Navíc v dnešní době již většina operačních systémů obsahuje předem schválené certifikáty obecně známých certifikačních autorit, proti kterým se ověřují další, později získané certifikáty.

Nedílnou součástí je také pravidelná změna uživatelských certifikátů *C* a asymetrických klíčů. Opět je tedy třeba vrátit se k lidskému faktoru, který je, jak již bylo psáno výše, nejjednodušší cestou k prolomení certifikačního systému a celkově nejslabším článkem jinak velmi silného řetězce.

Zásadním zlomem v bezpečnosti asymetrické kryptografie, a tedy i celkového certifikačního systému a jako takového PKI bude nejspíš průlom v oblasti kvantových výpočtů a počítačů, kdy, jak je nastíněno v oddílu 1.3, většina dnes používaných asymetrických šifrovacích metod bude jednodušše a rychle prolomitelná v souvislosti s možností velkého množství paralelních výpočtů.

I v této oblasti jsou již ale popsány metody, které buď v kvantových výpočtech nemají výpočetní výhodu ve smyslu paralelních výpočtů, nebo nejsou algoritmizovatelné a tudíž nejdou zpracovt pomocí deterministického algoritmu. V budoucnu je tedy třeba se orientovat ne na hrubé zvyšování bezpečnosti pomocí prodlužování hesla, ale spíše na důmyslnější metody zpracování šifrovacích klíčů a jejich bezpečnou distribuci.

Seznam použité literatury

[1] MAREŠ, Amádeo. Staňte se programátorem: Prolomení MD5. *Staňte se programátorem: Prolomení MD5* [online]. [cit. 2013-12-29]. Dostupné z: http://www.zive.cz/clanky/stante-se-programatorem-prolomeni-md5/sc-3-a-148556/

[2] List of Rainbow Tables. *List of Rainbow Tables* [online]. 2013 [cit. 2013-12-29]. Dostupné z: http://project-rainbowcrack.com/table.htm

[3] KELSEY, John. NIST. *SHA-3: Past, Present and Future*. 2013. Dostupné z: https://docs.google.com/file/d/0BzRYQSHuuMYOQXdHWkRiZXlURVE/edit?pli=1

[4] Hash Function Louge: Hash Functions. *Hash Function Louge* [online]. 2012 [cit. 2013-12-29]. Dostupné z: http://www.larc.usp.br/~pbarreto/hflounge.html

[5] *The Electronic Frontier Foundation* [online]. 1998 [cit. 2013-12-29]. Dostupné z: http://w2.eff.org/Privacy/Crypto/Crypto_misc/DESCracker/HTML/19980716_eff_des_faq.html

[6] Bots and Botnets—A Growing Threat. NORTON BY SYMANTEC. *Bots and Botnets—A Growing Threat* [online]. 2013 [cit. 2013-12-29]. Dostupné z: http://us.norton.com/botnet/

[7] RIJMEN, Vincent a Joan DAEMEN. AES Proposal: Rijndael. In: *The Rijndael Block Cypher* [online]. 1999 [cit. 29.12.2013]. Dostupné z: http://csrc.nist.gov/archive/aes/rijndael/Rijndael-ammended.pdf

[8] NECHVATAL, James, Elaine BARKER, Lawrence BASSHAM, William BURR, Morris DWORKIN, James FOTI a Edward ROBACK. U.S. DEPARTMENT OF COMMERCE. *Report on the Development of the Advanced Encryption Standard (AES)*. 2000. Dostupné z: http://csrc.nist.gov/archive/aes/round2/r2report.pdf

[9] COURTOIS, Nicolas T. Is AES a secure cypher?. *Cryptosystem.net* [online]. 2007 [cit. 2013-12-29]. Dostupné z: http://www.cryptosystem.net/aes/

[10] ALEŠ, Vaverka. *DIGITÁLNÍ PODPIS* [online]. Brno, 2008 [cit. 2013-12-29]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=9229. Bakalářská práce. VUT Brno. Vedoucí práce doc. Ing. Václav Zeman, Ph.D.

[11] LEPKA, Karel. MUNI. *Malá Fermatova Věta*. Brno, 2000. Dostupné z: http://bart.math.muni.cz/~fuchs/ucitel/clanky/1_3_5.pdf

[12] HORDĚJČUK, Vojtěch. Čínská věta o zbytcích. *Čínská věta o zbytcích* [online]. 2013 [cit. 2013-12-29]. Dostupné z: http://voho.cz/wiki/matematika/cinska-veta-o-zbytcich/

[13] TROJÁK, Martin. *Realizace certifikační autority a digitálního podpisu*. Brno, 2008. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=7426. Diplomová. VUT Brno. Vedoucí práce Petra Lambertová.

[14] MCGREW, David, Kevin M. IGOE a SALTER. INTERNET ENGINEERING TASK FORCE. *Fundamental Elliptic Curve Cryptography Algorithms: RFC6090*. 2011. Dostupné z: http://tools.ietf.org/search/rfc6090

[15] KOUŘIL, Daniel. MUNI. *Správa soukromých klíčů pomocí hardwarových tokenů*. 2005. Dostupné z: http://www.ics.muni.cz/bulletin/articles/335.html

[16] GRAHAM, Paul. A plan for Spam. *Paul Graham* [online]. 2002 [cit. 2013-12-29]. Dostupné z: http://www.paulgraham.com/spam.html

[17] Phishing. *OnGuarOnline* [online]. 2011 [cit. 2013-12-29]. Dostupné z: http://www.onguardonline.gov/phishing

[18] DOSTÁLEK, Libor a Marta VOHNOUTOVÁ. *Velký průvodce infrastrukturou PKI a technologií ektronického podpisu*. vyd. 1. Brno: Computer Press, 2006, 534 s. ISBN 80-251-0828-7.

[19] KLÍMA, Vlastimil. Hašovací funkce, principy, příklady a kolize. *CryptoWorld* [online]. 2005 [cit. 2013-12-29]. Dostupné z: http://cryptography.hyperlink.cz/2005/cryptofest_2005.htm

[20] Certifikační politika. *I.CA* [online]. 2013 [cit. 2013-12-29]. Dostupné z: http://www.ica.cz/Certifikacni-politika

[21] Advanced Encryption Standard (AES). UNIVERSITY OF PITTSBURGH. *School of Information Science* [online]. 2005 [cit. 2013-12-29]. Dostupné z: http://www.sis.pitt.edu/~ir/Projects/is2470_fall2007/e/aes_applet.html

[22] TREMHOLM, Samuel. S-box used by the AES cryptographic algorithm. *Sam Tremholme's webpage* [online]. 2011 [cit. 2013-12-29]. Dostupné z: http://www.samiam.org/s-box.html

[23] Rozlišovací jméno. IBM. *ISeries Information Center, verze 5 vydání 3* [online]. 2005 [cit. 2013-12-30]. Dostupné z: http://publib.boulder.ibm.com/infocenter/iseries/v5r3/index.jsp?topic=%2Frzahu%2Frzahudistname.htm

[24] UHLMANN, Stephan. ARP cache poisoning / ARP spoofing. *Su2* [online]. 2004 [cit. 2013-12-30]. Dostupné z: http://su2.info/doc/arpspoof.php

[25] ALGOLOGIC RESEARCH AND SOLUTIONS. *Negotiating public keys in asymmetric key cryptography*. 2012. Dostupné z: http://www.freewebs.com/profpartha/publications/publikey.pdf

[26] JOHNSON, George. Zkratka napříč časem: cesta ke kvantovému počítači. 1. vyd. v českém jazyce. Překlad Jiří Podolský, Pavel Cejnar. Praha: Dokořán, Argo, 2004, 235 s. ISBN 80-865-6983-7.

[27] PQCRYPTO. Post-quantum cryptography [online]. 2013 [cit. 2014-05-26]. Dostupné z: http://pqcrypto.org/index.html

[28] Daniel J. Bernstein, Johannes Buchmann, Erik Dahmen (editors). Post-quantum cryptography. Springer, Berlin, 2009. ISBN 978-3-540-88701-0

[29] BENNETT, Charless H. a Gilles BRASSARD. ICC. *Quantum Cryptography: Public Key Distribution and Coin Tossing*. Bangalore, India, 1984. Dostupné z: http://www.cs.ucsb.edu/~chong/290N-W06/BB84.pdf

[30] GERSHON, Eric. New qubit control bodes well for future of quantum computing. Phys.org [online]. [cit. 2013-01-14]. Dostupné z: http://phys.org/news/2013-01-qubit-bodes-future-quantum.html

[31] LYABASHEVSKY, Vadim, Chris PEIKERT a Oded REGEV. A toolkit for Ring-LWE cryptography. 2013. Dostupné z: http://www.di.ens.fr/~lyubash/papers/toolkit.pdf

[32] BREAKEN, An, Christopher WOLF a Bart PRENEEL. A Study of the Security of Unbalanced Oil and Vinegar Signature Schemes. 2013. Dostupné z: http://www5.rz.rub.de:8032/imperia/md/content/wolf/uov_rsa.pdf

[33] DE FEO, Luca, David JAO. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. 2013. Dostupné z: http://cacr.uwaterloo.ca/techreports/2011/cacr2011-32.pdf

[34] KOHL, John a Clifford NEUMAN. The Kerberos Network Authentication Service (V5). In: IETF [online]. 1993 [cit. 2014-05-27]. Dostupné z: http://www.ietf.org/rfc/rfc1510.txt

[35] Creating a CALG_SSL3_SHAMD5 Hash. MICROSOFT. *MSDN* [online]. 2013 [cit. 2014-05-27]. Dostupné z:http://msdn.microsoft.com/en-us/library/windows/desktop/aa379865(v=vs.85).aspx

[36] Microsoft DSS and Diffie-Hellman/Schannel Cryptographic Provider. MICROSOFT. *MSDN* [online]. 2013 [cit. 2014-05-27]. Dostupné z: http://msdn.microsoft.com/en-us/library/windows/desktop/aa386984(v=vs.85).aspx

[37] Alternative Methods of Domain Control Validation (DCV). COMODO SSL. *Knowledgebase: SSL Validation FAQs*[online]. 2013 [cit. 2014-05-27]. Dostupné z: https://support.comodo.com/index.php?/Default/Knowledgebase/Article/View/791/16/alternative-methods-of-domain-control-validation-dcv

[38] RFC5246. *The Transport Layer Security (TLS) Protocol Version 1.2*. IETF, 2008. Dostupné z:http://tools.ietf.org/html/rfc5246#page-41

[39] Certificates Tools and Settings. MICROSOFT. *TechNet* [online]. 2013 [cit. 2014-05-27]. Dostupné z:http://technet.microsoft.com/en-us/library/cc787544(WS.10).aspx#w2k3tr_certs_tools_dgzz

[40] LODGE, David. BROWSER SUPPORT FOR TLS 1.1 AND 1.2. In: *PenTestPartners* [online]. 2013 [cit. 2014-05-27]. Dostupné z: http://www.pentestpartners.com/blog/browser-support-for-tls-1-1-and-1-2/

[41] BEAST & CRIME: How TLS was a2acked. In: *IETF* [online]. 2012 [cit. 2014-05-27]. Dostupné z:http://www.ietf.org/proceedings/85/slides/slides-85-saag-1

[42] OpenSSL: OpenSSL vulnerabilities. In: *OpenSSL* [online]. 2014 [cit. 2014-05-27]. Dostupné z: https://www.openssl.org/news/vulnerabilities.html

[43] RFC 5280. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. IETF, 2008. Dostupné z: http://tools.ietf.org/html/rfc5280

Seznam použitých zkratek

PIN	Personal identification number
MD5	Message-Digest verze 5
SHA1	Secure Hash Algorithm
DES	Data Encryption Standard
AES	Advanced Encryption System
NIST	National Institute of Standards and Technology
RSA	Rivest-Shamir-Adleman system
DH	Diffie-Hellman system
ECC	Elliptic Curve Cryptography
LCM	Least Common Denominator
Ring-LWE	Ring-Learn with Errors
UOV	Unbalanced Oil and Vinegar
PKCS	Public-Key Cryptography Standards
SSL	Secure Socket Layer
CA	Certifikační Autorita
ITU	International Telecommunication Union
RFC	Request For Comments
URL	Uniform Resource Locator
CRL	Certificate Revocation List
LDAP	Lightweight Directory Access Protocol
OCSP	Online Certificate Status Protocol
MIM	Man In the Middle
ARP	Address Resolution Protocol
MTA	Message Transfer Agent
SSID	Service Set IDentification
HSM	Hardware Security Module
MS	Microsoft
MSDNA	Microsoft Developer Network Academic Alliance
GNS3	Graphical Network Simulator
IIS	Internet Information Services
TLS	Transport Layer Security
DSS	Decentralized Software Services

DCV	Domain Control Validation
MDC	Multi Domain Certificate
UCC	Unified Communications Certificate
DNS	Domain Name Server
http	Hypertext Transfer Protocol
FQDN	Fully Qualified Domain Name
DER	Distinguished Encoding Rules
PEM	Privacy Enhanced Mail
FTP	File Transfer Protocol
IP	Internet Protocol
MAC	Message Authetication Code
BEAST	Browser Exploit Against SSL/TLS
CRIME	Compression Ratio Info-leak Made Easy
BREACH	Browser Reconaissance and Exfiltration via Adaptive Compression of Hypertext

Seznam použitých obrázků

Příloha č. 1

Seznam nejpoužívanějších jedinečných jmen

Common Name CN	commonName	Název objektu, pod kterým je místně znám. Např. u osob to může být jméno a příjmení. U serverů pak jejich DNS-jméno.
Surname	surname	Příjmení
Serial Number	*serialNumber*	Slouží k rozlišení různých certifikovaných objektů, kterým by jinak vyšlo stejné jedinečné jméno. Je doporučen používat u kvalifikovaných certifikátů.
Country C	countryName	Stát podle ISO 3166, tj. podle stejné normy, jaká se používá pro top level domény DNS (CZ=Česká republika, SK=Slovensko, FJ=Fidži, atp.)
Locality L	localityName	Místo (např. město)
State or Province SP	stateOrProvinceName	Nižší organizační jednotka státu. Např. spolková země.
Organization Organization Unit OU	organizationName organizationUnitName	Název firmy Název části firmy
Title	title	Titul
Postal Adres	*postalAdress*	*Poštovní adresa*
Name	name	Jméno
Given Name	givenName	Rodné jméno
Initilas	initials	Iniciály
Generation Qualifier	generationQualifier	Např. „Jr.“ či „IV“ pro Karel IV
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 Component DC	domainComponent	

Příloha č. 2

Certifikační politika autority Comodo SSL

COMODO CERTIFICATE SUBSCRIBER AGREEMENT

IMPORTANT - PLEASE READ THIS CERTIFICATE SUBSCRIBER AGREEMENT CAREFULLY BEFORE APPLYING FOR, ACCEPTING, OR USING A COMODO CERTIFICATE. BY USING, APPLYING FOR, OR ACCEPTING A COMODO CERTIFICATE OR BY CLICKING ON "I AGREE", YOU ACKNOWLEDGE THAT YOU HAVE READ THIS AGREEMENT, THAT YOU UNDERSTAND IT, AND THAT YOU AGREE TO ITS TERMS. IF YOU DO NOT ACCEPT THIS AGREEMENT, DO NOT APPLY FOR, ACCEPT, OR USE A COMODO CERTIFICATE AND DO NOT CLICK "I AGREE".
This agreement is between you ("Subscriber") and Comodo CA Limited ("Comodo"), a United Kingdom company. The agreement governs your application for and use of an SSL Certificate issued from Comodo. You and Comodo agree as follows:

1. Subscription Service.
1.1. Issuance. Upon Comodo's acceptance of Subscriber's application for a Certificate, Comodo shall attempt to validate the application information in accordance with the Comodo CPS and, for EV Certificates, the EV Guidelines. If Comodo chooses to accept the application and can validate Subscriber to Comodo's satisfaction, Comodo shall issue the ordered Certificate(s) to Subscriber. Comodo may refuse an application for any reason.

1.2. Multiple Certificates. This agreement applies to multiple future Certificate request and any resulting Certificates, regardless of when the Certificate is requested or issued.

1.3. License. After issuance, Comodo grants Subscriber a revocable, non-exclusive, nontransferable license to use the issued Certificates on the server hosting the domain name(s) listed in the Certificate. Comodo also grants Subscriber a non-exclusive, non-transferable, and revocable license to use Comodo's EV AUTO-Enhancer and EV Enhancer technology in connection with Comodo EV Certificates. All rights not expressly granted herein to Subscriber are reserved to Comodo.

1.4. TrustLogos. Comodo grants Subscriber a license to display purchased TrustLogos on domain(s) secured by a Comodo Certificate. When revoking a Certificate, Comodo may also revoke any TrustLogos issued to the same site. Subscriber shall not modify a TrustLogo in any manner. Subscriber shall not display or use a TrustLogo 1) to represent that Comodo guarantees any non-Comodo products or services, 2) on a site that is misleading, defamatory, libelous, disparaging, obscene or otherwise objectionable to Comodo, or 3) in a way that harms Comodo's rights to its trademarks or harms Comodo's business reputation.

1.5. Fee. Subscriber shall pay all applicable fees for the Certificate before it issues. Certificate fees are provided to Subscriber during the application process. All payments are nonrefundable, except that the Certificate's seller will refund a payment if, before twenty (20) business days after the Certificate's issuance, the Subscriber has 1) not used the Certificate and 2) made a written request to Comodo for the Certificate's revocation.

1.6. Subscriber Obligations. Subscriber shall:
(i) use the Certificates only for the purposes listed in the Comodo CPS;

(ii) only install an issued Certificate on the servers accessible at the domain name(s) listed in the Certificate and only use an issued Certificate for authorized business of the Subscriber;

(iii) be responsible for any computer hardware, telecommunications hardware, and software necessary to use the Certificate;

(iv) obtain and maintain any authorization or license necessary to use the Certificate;

(v) bind every Relying Party to Comodo's Relying Party Agreement;

(vi) keep Confidential Information confidential and uncompromised, and immediately inform Comodo and request revocation of any affected Certificates if Subscriber reasonably believes that Confidential Information is likely to be disclosed or compromised;

(vii) ensure that all information provided to Comodo is complete and accurate and does not include any information that would be unlawful, contrary to public interest, or otherwise likely to damage the business or reputation of Comodo if used in any way;

(viii) immediately cease using a Certificate and associated Private Key 1) if the Private Key is compromised or 2) after the Certificate is expired or revoked,

(ix) immediately notify Comodo of 1) any a breach of this agreement or 2) any information provided to Comodo changes, ceases to be accurate, or becomes inconsistent with the warranties made by Subscriber herein, and

(x) comply with all applicable local and international laws when receiving or using a Certificate, including all export laws. Subscriber shall not export or re-export, either directly or indirectly, any Certificate to a country or entity under United Kingdom or United States restrictions. SUBSCRIBER ASSUMES ALL LIABILITY FOR ITS VIOLATION OF EXPORT LAWS.

1.7. Restrictions. Subscriber shall not:
(i) impersonate or misrepresent Subscriber's affiliation with any entity,

(ii) modify, license, create a derivative work of, or transfer any Certificate (except as required to use the Certificate) or Private Key;

(iii) install or use an issued Certificate until after Subscriber has reviewed and verified the Certificate data's accuracy;

(iv) upload or distribute any files or software that may damage the operation of another's computer,

(v) use the Services to 1) engage in conduct that is offensive, abusive, contrary to public morality, indecent, defamatory, obscene, or menacing, 2) breach the confidence of a third party, 3) cause Comodo or a third party distress, annoyance, denial of any service, disruption or inconvenience, 4) send or receive unsolicited bulk correspondence or 5) create a Private Key that is substantially similar to a Comodo or third party's Private Key,

(vi) make representations regarding the Service to any third party except as agreed to in writing by Comodo.

2. Warranties and Representations. Subscriber warrants that:
(i) for EV Certificates, the subject named in the Certificate has exclusive control of the domain name(s) listed in the Certificate;

(ii) it has full power and authority to enter into this agreement and perform its obligations hereunder;

(iii) for EV Certificates, the individual accepting the Agreement is expressly authorized by Subscriber to sign the agreement for Subscriber.

3. Revocation. Comodo may revoke a Certificate if Comodo believes that:
(i) Subscriber requested revocation of the Certificate;

(ii) Subscriber did not authorize the Certificate and has not retroactively granted authorization;

(iii) Subscriber breached this Agreement;

(iv) Confidential Information related to the Certificate has been disclosed or compromised;

(v) the Certificate has been 1) misused, 2) used contrary to law, rule, or regulation or 3) used, directly or indirectly, for illegal or fraudulent purposes;

(vi) information in the Certificate is inaccurate or misleading,

(vii) for EV Certificates, Subscriber loses exclusive control over a domain name listed in the Certificate;

(viii) the Certificate was not issued or used in accordance with Comodo's CPS, industry standards, or, for EV Certificates, the EV Guidelines;

(ix) Comodo 1) ceased operations or 2) is no longer allowed to issue the Certificate, and no other certificate authority has agreed to provide revocation support for the Certificate;

(x) Subscriber is added as a denied party or prohibited person to a blacklist, or is operating from a prohibited destination under the laws of Comodo's jurisdiction of operation;

(xi) the Certificate was issued to publishers of malicious software;

(xii) the CPS authorizes revocation of the Certificate; or

(xiii) the Certificate, if not revoked, will compromise the trust status of Comodo.

After revoking the Certificate, Comodo may, in its sole discretion, reissue the Certificate to Subscriber or terminate the agreement.

4. Intellectual Property Rights.

4.1. Comodo IP Rights. Comodo retains, and Subscriber shall not obtain or claim, all title, interest, and ownership rights in:
(i) the Services, including issued Certificates,

(ii) all copies or derivative works of the Services, regardless of who produced, requested, or suggested the copy or derivative work,

(iii) all documentation and materials provided by Comodo, and

(iv) all of Comodo's copyrights, patent rights, trade secret rights and other proprietary rights.

4.2. Trademarks. Subscriber shall not use a Comodo trademark without Comodo's written consent. Comodo consents to use of trademarks in connection with provided TrustLogos.

4.3. Other Rights. EV AUTO-Enhancer™ for Windows uses Microsoft Detours Professional 2.1. Detours is Copyright 1995-2004, Microsoft Corporation. Portions of the Detours package may be covered by patents owned by Microsoft corporation. Microsoft, MS-DOS, Windows, Windows NT, Windows 2000, Windows XP, and DirectX are registered trademarks or trademarks of Microsoft Corporation in the U.S. and other countries.

5. Indemnification.

5.1. Indemnification. Subscriber shall indemnify Comodo and its affiliates and their respective directors, officers, employees, and agents (each an "Indemnified Person") against all liabilities, losses, expenses, or costs (including reasonable attorney's fees) (collectively "Losses") that, directly or indirectly, are based on Subscriber's breach of this agreement, information provided by Subscriber, or Subscriber's or its customers' infringement on the rights of a third party.

5.2. Indemnification Procedure. Comodo shall notify Subscriber promptly of any demand for indemnification. However, Comodo's failure to notify will not relieve Subscriber from its indemnification obligations except to the extent that the failure to provide timely notice materially prejudices Subscriber. Subscriber may assume the defense of any action, suit, or proceeding giving rise to an indemnification obligation unless assuming the defense would result in potential conflicting interests as determined by the Indemnified Person in good faith. Subscriber may not settle any claim, action, suit or proceeding related to this agreement unless the settlement also includes an unconditional release of all Indemnified Persons from liability.

5.3. Additional Liability. The indemnification obligations of Subscriber are not Comodo's sole remedy for Subscriber's breach and are in addition to any other remedies Comodo may have against Subscriber under this agreement. Subscriber's indemnification obligations survive the termination of this agreement.

6. Term and Termination.

6.1. Term. Unless otherwise terminated as allowed herein, this agreement is effective upon Subscriber's acceptance and lasts for as long as a Certificate issued under the agreement is valid.

6.2. Termination. Either party may terminate the agreement with 20 business days notice for convenience. Comodo may terminate this agreement immediately without notice if
(i) Subscriber materially breaches this agreement,

(ii) if Comodo revokes a Certificate as allowed herein,

(iii) if Comodo rejects Subscriber's Certificate application,

(iv) Comodo cannot satisfactorily validate Subscriber in accordance with section 1.1, or

(v) if industry standards change in a way that affects the validity of the Certificates ordered by Subscriber.

6.3. Events Upon Termination. After termination, Comodo may revoke any other Certificate's issued to Subscriber without further notice. Subscriber shall pay any amounts still owed for the Certificates. Comodo is not obligated to refund any payment made by Subscriber upon termination of this Agreement.

7. Disclaimers and Limitation of Liability.

7.1. Relying Party Warranties. Subscriber acknowledges that the Relying Party Warranty is only for the benefit of Relying Parties. Subscriber does not have rights under the warranty, including any right to enforce the terms of the warranty or make a claim under the warranty.

7.2. Exclusion of Warranties. THE SERVICES ARE PROVIDED "AS IS" AND "AS AVAILABLE". COMODO EXPRESSLY DISCLAIMS ALL IMPLIED AND EXPRESS WARRANTIES IN THE SERVICES. THIS DISCLAIMER INCLUDES ALL WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT AND IS EFFECTIVE TO THE MAXIMUM EXTENT ALLOWED BY LAW. COMODO DOES NOT GUARANTEE THAT 1) THE SERVICES WILL MEET SUBSCRIBER'S REQUIREMENTS OR EXPECTATIONS OR 2) THAT ACCESS TO THE SERVICES WILL BE UNINTERRUPTED, TIMELY, SECURE, OR ERROR-FREE.

7.3. Limitation on Liability. SUBJECT TO SECTION 7.4, THE TOTAL LIABILITY OF COMODO AND ITS AFFILIATES, AND EACH OF THEIR OFFICERS, DIRECTORS, PARTNERS, EMPLOYEES, AND CONTRACTORS, RESULTING FROM OR CONNECTED TO THIS AGREEMENT IS LIMITED TO THE AMOUNT PAID BY SUBSCRIBER FOR THE SERVICES GIVING RISE TO THE LIABILITY. SUBSCRIBER WAIVES ALL LIABILITY FOR ANY SPECIAL, INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES. THIS WAIVER INCLUDES ALL DAMAGES FOR LOST PROFITS, REVENUE, USE, OR DATA AND APPLIES EVEN IF COMODO IS AWARE OF THE POSSIBILITY OF SUCH DAMAGES. These limitations apply to the maximum extent permitted by law regardless of 1) the reason for or nature of the liability, including tort claims, 2) the number of any claims, 3) the extent or nature of the damages, and 4) whether any other provisions of this agreement have been breached or proven ineffective.

7.4. Exception. Nothing in this agreement excludes or limits the liability of either party for death or personal injury resulting from the negligence of that party or for any statements made fraudulently by either party.

8. Remedy.

8.1. Injunctive Relief. Subscriber acknowledges that its breach of this agreement will result in irreparable harm to Comodo that cannot adequately be redressed by compensatory damages. Accordingly, in addition to any other legal remedies which may be available, Comodo may seek and obtain an injunctive order against a breach or threatened breach of the agreement by Subscriber.

8.2. Limitation on Actions. Except for actions and claims related to a party's indemnification and confidentiality obligations, all claims and actions arising from this agreement must be brought within one (1) year from the date when the cause of action occurred.

8.3. Remedy. Subscriber's sole remedy for a defect in the Services is to have Comodo use reasonable efforts to correct the defect. Comodo is not obligated to correct a defect if (i) the Service was misused, damaged, or modified, (ii) Subscriber did not immediately report the defect to Comodo, or (iii) Subscriber breached any provision of this agreement.

9. Confidentiality. Except as allowed herein, a party ("Receiving Party") shall not use or disclose any Confidential Information provided by the other party (the "Disclosing Party") other than for the purpose of performing its obligations under this agreement. The Receiving Party shall take reasonable measures to prevent unauthorized disclosure and shall ensure that any person receiving Confidential Information complies with the restrictions in this section. The Receiving Party may disclose Confidential Information if the information:
(i) is already possessed by the Receiving Party before receipt from the Disclosing Party;

(ii) is or becomes public domain without fault of the Receiving Party;

(iii) is received by the Receiving Party from a third party who is not under an obligation of confidentiality or a restriction on the use and disclosure of the information,

(iv) is disclosed in response to the requirements of a law, governmental order, regulation, or legal process and the Receiving Party first gives prior notice to the Disclosing Party of the requirement to disclose the information, or

(v) is disclosed under operation of law to the public without a duty of confidentiality.

A party asserting one of the exceptions to Confidential Information above shall prove the assertion using verifiable documentary evidence. The restrictions contained in this section apply for the duration of the agreement plus five years after its termination.

10. Privacy.
(i) Comodo shall follow the privacy policy posted on its website when receiving and using information from the Subscriber. Comodo may amend the privacy policy at any time by posting the amended privacy policy on its website. Subject to Section 10(ii), Comodo shall use reasonable efforts in protecting Subscriber's information. Subscriber acknowledges that risks remain that are beyond Comodo's reasonable control and waives all liability of Comodo for these risks.

(ii) Subscriber consents to 1) Comodo disclosing Subscriber's information publicly by embedding the information in issued Certificates and 2) Comodo disclosing and transferring Subscriber's information to third parties located outside of the European Union as necessary to validate and issue Certificates.

(iii) Subscriber may opt-out of having information used for purposes not directly related to the Services by emailing a clear notice to optout@comodo.com. By clicking "I AGREE", Subscriber affirmatively consents to receiving Comodo's and its affiliates marketing material.

11. Miscellaneous.

11.1. Force Majeure and Internet Frailties. Other than for payment obligations by Subscriber, neither party will be liable for a delay or failure to perform an obligation to the extent that the delay or failure is caused by an occurrence beyond the party's reasonable control. Each party acknowledges that the operation of the Internet is beyond the other party's reasonable control, and neither party will be liable for a delay or failure caused by an interruption or failure of telecommunication or digital transmission links, Internet slow-downs or failures, or other such transmission failure.

11.2. Notices. You shall send all notices to Comodo by first class mail in English writing, with return receipt requested, to Comodo CA Limited, 26 Office Village, 3rd Floor, Exchange Quay, Trafford Road, Salford, Manchester M5 3EQ, United Kingdom. Comodo shall send all notices to Subscriber's contact information listed on its Certificate application. Comodo may send notices by mail, email, or facsimile.

11.3. Entire Agreement. This agreement and all documents referred to herein constitutes the entire agreement between the parties, superseding all other agreements that may exist with respect to the subject matter. Section headings are for reference and convenience only and are not part of the interpretation of this agreement.

11.4. Amendments. Comodo may amend this agreement, the CPS, the Relying Party Agreement, the Relying Party Warranty, its website, and any documents listed in its Repository at any time by posting either the amendment or the amended document in the Repository. Subscriber shall periodically review the Repository to be aware of any changes. Subscriber may terminate the agreement if Subscriber does not agree to the amendment. Subscriber's continued use of the Services after an amendment is posted constitutes Subscriber's acceptance of the amendment.

11.5. Waiver. A party's failure to enforce a provision of this agreement will not waive the party's right to enforce the same provision later or right to enforce any other provision of this agreement. To be effective, all waivers must be both in writing and signed by the party benefiting form the waived provision.

11.6. Assignment. Subscriber may not assign any of its rights or obligations under this agreement without the prior written consent of Comodo. Any transfer without consent is void. Comodo may assign its rights and obligations without Subscriber's consent.

11.7. Governing Law and Venue. The laws of England and Wales govern the interpretation, construction, and enforcement of this agreement and all proceedings arising out of it, including tort claims, without regard to any conflicts of law principles.

All proceedings or legal action arising from this agreement must be commenced in the courts of England and Wales. Both parties agree to the exclusive venue and jurisdiction of these courts.

11.8. Severability. Any provision determined invalid or unenforceable by rule of law will be reformed to the minimum extent necessary to make the provision valid and enforceable. If reformation is not possible, the provision is deemed omitted and the balance of the agreement remains valid and enforceable.

11.9. Survival. All provisions of the agreement related to confidentiality, proprietary rights, indemnification, and limitations of liability survive the termination of the agreement.

11.10. Rights of Third Parties. The Certificate Beneficiaries are express third party beneficiaries of Subscriber's obligations and warranties in this agreement.

12. Definitions.

12.1. "Certificate" means a digitally signed electronic data file issued by Comodo to a person or entity seeking to conduct business over a communications network which contains the identity of the person authorized to use the Digital Signature, a copy of their Public Key, a serial number, a time period during which the data file may be used, and a Digital Signature issued by Comodo.

12.2. "CPS" refers to the documents explaining Comodo's polices and procedures when operating its PKI infrastructure.

12.3. "Confidential Information" means all material, data, systems, technical operations, and other information concerning Comodo's business operations that is not known to the general public, including all information about the Certificate issuance services (such as all Private Keys, personal identification numbers and passwords).

12.4. "Certificate Beneficiaries" means the Subscriber, the Subject named in the Certificate, any third parties with whom Comodo has entered into a contract for inclusion of its root certificate, and all Relying Parties that actually rely on such Certificate during the period when it is valid.

12.5. "Digital Signature" means an encrypted electronic data file which is attached to or logically associated with other electronic data and which identifies and is uniquely linked to the signatory of the electronic data, is created using the signatory's Private Key and is linked in a way so as to make any subsequent changes to the electronic data detectable.

12.6. "EV AUTO-Enhancer" means Comodo's patent-pending process and software to enable EV functionality on web browsing computers using a modified Apache configuration file or the Comodo developed IIS plug-in.

12.7. "EV Certificate" means a Certificate signed to Comodo's EV root certificate that is designed for use with an SSL v3 or TLS v 1.0 enabled web browse and that complies with the EV Guidelines.

12.8. "EV Enhancer" means the process and software used by Comodo to enable EV functionality on web browsing computers by pointing the web browser on the web browsing computer to a beacon website designed to download and install a new EV root certificate.

12.9. "EV Guidelines" refers to the official, adopted guidelines governing EV Certificates as established by the CA/Browser Forum that are available online at http://www.cabforum.org.

12.10. "Private Key" means a confidential encrypted electronic data file designed to interface with a Public Key using the same encryption algorithm and which may be used to create Digital Signatures, and decrypt files or messages which have been encrypted with a Public Key.

12.11. "Public Key" means a publicly available encrypted electronic data file designed to interface with a Private Key using the same encryption algorithm and which may be used to verify Digital Signatures and encrypt files or messages.

12.12. "Relying Party" means an entity that acts in reliance on a Certificate or a Digital Signature.

12.13. "Relying Party Agreement" refers to an agreement located on the Comodo Repository that governs a Relying Party's use of the Certificate when transacting business with the Subscriber's website.

12.14. "Relying Party Warranty" refers to a warranty offered by Comodo to a Relying Party under the terms and conditions found in the Comodo Relying Party Agreement in connection with the Relying Party's use of a Certificate.

12.15. "Repository" means a publicly available collection of information and databases relating to Comodo's Certificate practices and which is available at http://www.comodo.com/repository.

12.16. "Services" means the Certificates ordered hereunder along with any related TrustLogos, software, and documentation.

12.17. "TrustLogo" means a logo provided by Comodo for use on a Subscriber's site in connection with an issued Certificate.

ACCEPTANCE

BY CLICKING "I AGREE", YOU ACKNOWLEDGE THAT YOU HAVE READ AND UNDERSTAND THIS AGREEMENT AND THAT YOU AGREE TO COMPLY WITH ITS TERMS. DO NOT CLICK "I AGREE" IF YOU DO NOT ACCEPT THIS AGREEMENT.