Infrastruktura veřejného klíče
Public key infrastructure

Mgr. Petr Soukup

Bakalářská práce
2012

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2011/2012

ZADÁNÍ BAKALÁŘSKÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Mgr. Petr SOUKUP**
Osobní číslo:	**A09791**
Studijní program:	**B 3902 Inženýrská informatika**
Studijní obor:	**Bezpečnostní technologie, systémy a management**
Téma práce:	**Infrastruktura veřejného klíče**

Zásady pro vypracování:

1. **Seznamte se s pojmy symetrická a nesymetrická šifra.**
2. **Charakterizujte pojem hašovací funkce.**
3. **Vysvětlete pojem elektronický podpis a zaručený elektronický podpis.**
4. **Objasněte pojem Certifikační Autority.**
5. **Prakticky vytvořte podpis pomocí kryptografických standardů, adresář a certifikační autoritu.**

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **MENEZES, Alfred J. Handbook of applied cryptography. Vyd. 1. Boca Raton: CRC Press, 1997, 780 s. ISBN 08-493-8523-7.**
2. **DOSTÁLEK, Libor, Marta VOHNOUTOVÁ a Miroslav KNOTEK. Velký průvodce infrastrukturou PKI a technologií elektronického podpisu. 2., aktualiz. vyd. Brno: Computer Press, 2009, 542 s. ISBN 978-802-5126-196.**
3. **PETERKA, Jiří. Báječný svět elektronického podpisu. Praha: CZ.NIC z.s.p.o., 2011. ISBN 978-80-904248-3.**
4. **BUDIŠ, Petr. Elektronický podpis a jeho aplikace v praxi. 1. vyd. Olomouc: ANAG, 2008, 157 s. ISBN 978-807-2634-651.**
5. **BOSÁKOVÁ, Dagmar. Elektronický podpis. Vyd. 1. Praha: ANAG, 2002, 141 s. ISBN 80-726-3125-X.**

Vedoucí bakalářské práce:	**Ing. Ján Ivanka** Ústav bezpečnostního inženýrství
Datum zadání bakalářské práce:	**24. února 2012**
Termín odevzdání bakalářské práce:	**25. května 2012**

Ve Zlíně dne 24. února 2012

L.S.

prof. Ing. Vladimír Vašek, CSc.
děkan

doc. Mgr. Milan Adámek, Ph.D.
ředitel ústavu

ABSTRAKT

Tato práce popisuje vybrané prostředky, které se používají v infrastruktuře veřejných klíčů. Bakalářská práce popisuje hašovací funkce, asymetrické šifry, elektronický podpis, certifikáty a certifikační autority. V textu jsou objasněny rozdíly mezi uvedenými pojmy a s pomocí jednoduchých příkladů jsou u vybraných pojmů vysvětleny jejich základní principy.

Klíčová slova: hašovací funkce, šifra, certifikační autority, PKI

ABSTRACT

This work describes the selected resources, which are used in public key infrastructure. Bachelor's thesis describes the hash function, an asymmetric cipher, the electronic signature certificates, and a certification authority. In the text clarified the differences between those concepts and with the help of simple examples are for selected terms explained their basic principles.

Keywords: hash function, cipher, certification authority, PKI

V úvodu své bakalářské práce bych chtěl poděkovat Ing. Jánu Ivankovi za odborné rady a připomínky poskytnuté v rámci konzultací. Dále bych chtěl poděkovat své rodině za podporu a trpělivost během studia.

Prohlašuji, že

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

Prohlašuji,

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

Ve Zlíně

……
podpis diplomanta

OBSAH

ÚVOD

Rychlý rozvoj internetu v posledních desetiletích s sebou přinesl i řadu bezpečnostních otázek. Mezi největší patří problematika zabezpečení vysoké důvěryhodnosti komunikace. V případě komunikace mezi jednotlivými právními subjekty je možné používat pojem e-commerce nebo mezi subjekty a vládou je používán pojem e-government. Velmi rozšířeným způsobem, kde je vyžadována vysoká míra zabezpečení je i komunikace na peněžních trzích, hlavně komunikace, kdy klient banky může pracovat se svým účtem. Zde hovoříme zpravidla o internetovém bankovnictví. Na jedné straně existuje velice silná poptávka po jednoznačně zabezpečených a nezpochybnitelných dokumentech Přičemž ve všech případech je nutné zabezpečit vysokou úroveň důvěryhodnosti v procesu předávání informací. V podstatě je nutné zabezpečit několik základních oblastí. Je nutné spolehlivě prokázat, že daný dokument nebyl pozměněn a existuje úplná shoda mezi odeslaným a přijatým dokumentem. Dále je potřeba zabezpečit, aby v případě podepsání určitého dokumentu, bylo zcela nezpochybnitelné autorství podpisu. A neméně důležitým prvkem je i otázka, kdy byl dokument vytvořen nebo zaslán. Samozřejmě, že musí existovat obecné organizační principy, které popisují způsoby zabezpečení dokumentů. Tyto principy by měly být co nejjednodušší, ale bezpečné a dále by měly být přijaty velkým počtem uživatelů. Jedna z možností, jak zabezpečit vysokou úroveň, je pomocí infrastruktury veřejného klíče tzv. PKI[1] .

Cílem mé práce je vysvětlit a na praktických příkladech ukázat na jakých základních principech funguje a tím přispět, alespoň trochu, ke zvýšení zájmu o tuto problematiku v průmyslu komerční bezpečnosti. Domnívám se, že v současné době převládá většinový názor, že využívání všech bezpečnostních prvků, které obsahuje struktura PKI je zbytečné. Často můžeme slyšet nebo číst jak je zdůvodňováno menší využívání. Možná to využije větší firma, než je naše. A kdyby jsme to i chtěli využít, tak proces registrace a proces zavedení je velice složitý. Ale na druhé straně, kdo by měl mít víc propracovaný systém bezpečnosti, než firmy zabývající se bezpečnostní problematikou? A není podstatné, zda se jedná o firmu poskytující hlídací služby, vždyť i takováto společnost může mít propracovaný systém například přístupových bodů. A každý mi dá za pravdu, že na

[1]PKI - Public key infrastructure - Infrastruktura veřejného klíče

zákazníka určitě bude lépe působit zaměstnanec firmy používající identifikační karty, než strážný neustále hledající klíče a to tak, že se dotazuje pomocí komunikačního prostředku prostřednictvím hlasitého poslechu svého nadřízeného. Takže celé okolí ví, kde se nacházejí tajné skrýše s klíči od chráněného prostoru. A to jsem vyzdvihnul pouze obchodní hledisko, záměrně jsem vynechal finanční efektivitu.

Lidé se většinou bojí nepoznaného, neznámého. Proto si myslím, že v prvním kroku musí dojít poznání základních principů PKI, k pochopení funkcí jednotlivých součástí infrastruktury veřejných klíčů. A následně dojde ke zvýšenému zájmu o popisovanou problematiku. Byl bych rád, kdyby i tato práce, alespoň malou mírou, přispěla k objasnění problematiky PKI.

I. TEORETICKÁ ČÁST

1 PKI

PKI si můžeme představit jako prostředí, které umožňuje ochranu informačních systémů, elektronických transakcí a komunikace. Zahrnuje veškerý software, technologie a služby, které umožňují šifrování veřejným a privátním klíčem. PKI je soustava technických a organizačních opatření spojených s vydáváním, správou, používáním a odvoláváním platnosti.

PKI obsahuje jednu nebo více řídících jednotek pro vytváření, distribuci a poskytování výstupní podpory pro certifikáty veřejného klíče. Termín PKI se začíná prvně užívat v polovině devadesátých let, přestože termín byl znám mnohem dříve. Skutečně, poznámky o veřejných klíčích, které jsou základem pro PKI jsou datovány rokem 1978. Cílem PKI je ustavit a ošetřovat důvěryhodné prostředí v otevřené síti jako je Internet. Prostředky PKI jsou především služby řídící práci s klíči a digitálními certifikáty, tedy jedná se o šifrování dat, o digitální podpisy elektronických zpráv. Základní nástroje PKI jsou vytvářeny na bázi asymetrické kryptografie. Jak systém vlastně funguje? Nejdříve z datové zprávy vytvoříme její jedinečný otisk (hash). Otisk zašifrujeme pomocí soukromého klíče a přidáme ke zprávě vznikne její digitální podpis. Každý, kdo má přístup k veřejnému klíči, může nyní s jeho pomocí ověřit takto vzniklý podpis. Aby vše fungovalo musí být zabezpečena důvěryhodná distribuce veřejných klíčů. To je v praxi prováděno pomocí digitálních certifikátů. Digitální certifikát je prostředek, jehož cílem je dát možnost ověřit propojení totožnosti stran s jejich veřejnými klíči. Ve skutečnosti se jedná o informaci, která obsahuje totožnost uživatele, jeho veřejný klíč a prostředky, které umožňují ověřit, že certifikátu lze důvěřovat. Přičemž je informace digitálně podepsána důvěryhodnou třetí stranou. Digitální certifikáty vydává specializovaný subjekt, kterému se říká certifikační autorita, která odpovídá za spolehlivost práce s digitálními certifikáty.

Aby bylo možné efektivně popsat problematiku PKI je nutné se nejdříve seznámit s prostředky, které jsou využívány v PKI.

2 PROSTŘEDKY POUŽÍVANÉ V PKI

2.1 Hash

Když bude někdo chtít pozměnit určitý dokument, má v podstatě dvě možnosti. Buď zachová text a zfalšuje podpis, nebo ponechá podpis a změní text ve svůj prospěch. Přičemž druhá možnost je těžko proveditelná, pokud je dokument vyhotoven v tištěné podobě, protože nahrazení části textu je téměř hned viditelné. Často bývají dokumenty vyhotoveny a podepisovány v několikerém provedení a každý z partnerů obdrží jednu kopii dokumentu. Pokud se jedná o dokument, který je vytvořen pouze v originále např. směnka, musí být vyplněna podle pevně daných pravidel, aby ji nebylo možné zpochybnit. A aby jakýkoliv neoprávněný zásah do textu byl lehce odhalitelný.

Obdobně elektronické dokumenty musí být opatřeny nástrojem, který zaznamená i malé změny v textu. Tím nástrojem je jednocestná funkce, která se nazývá otisk (hash). V české literatuře se používá častěji počeštěný název haš, popřípadě hašovací funkce (hash function). Základním principem je vygenerovat z řetězce libovolné délky krátký řetězec konstantní délky tzv. haš , který bude maximálně charakterizovat původní řetězec. Když je dána původní zpráva X hašovací funkce vygeneruje hash $h(X)$ o délce n bitů, přičemž $h(X)$ jednoznačně definuje původní zprávu X (někde nazývaná jako vstup).

Hašovací funkce má tyto základní vlastnosti:

1. libovolný řetězec konečné délky X transformuje na $h(X)$ fixní bitové délky
2. $h(X)$ je relativně lehce vypočitatelná
3. nelze z $h(X)$ zpětně určit X tj. odolnost argumentu (preimage resistance)
4. je nesnadné nalézt dva řetězce X a X' ($X \neq X$‘) takové, že $h(X) = h(X')$ tzv. odolnost druhého argumentu
5. je nesnadné nalézt k řetězci libovolnému X nalézt řetězec X' ($X \neq X$‘), takový aby platilo $h(X) = h(X')$ tzv. odolnost vůči kolizím (collision resistance) [1].

Rozdíl mezi podmínkou 4 a 5 je v tom, že u odolnosti vůči kolizím si vstupní řetězec volíme libovolně. Na první pohled se zdá, že se jedná o shodné definiční podmínky jako v předchozím případě, ale rozhodně tomu tak není. Danost řetězce X (prvního argumentu) je totiž při zkoušení odolnosti druhého argumentu určena nějakou, na útočníkovi nezávislou, podmínkou. Z pohledu útočníka je X již dáno a musí pro něj hledat X’.

Při obecné (silné) odolnosti proti kolizím si však útočník může volit X i X' s jediným cílem, a to aby $h(X)$ a $h(X')$ měly shodnou hodnotu – libovolnou.

Zajímavé rovněž je, že obě zmíněné situace mají rozdílný výpočet pravděpodobností náhodného nalezení kolizí. Druhá situace je v kryptologii známa jako tzv. narozeninový paradox. Jste-li na večírku, pak pro 50-ti procentní pravděpodobnost toho, že se na něm nachází někdo, kdo se narodil shodný den v roce, roky se neuvažují, jako právě vy, je zapotřebí, aby tam přišlo alespoň 254 osob. Pokud se však spokojíte s 50-ti procentní pravděpodobností toho, že se na večírku nachází libovolné dvě osoby se shodným narozeninami, pak dostačuje, aby večírek navštívilo 23 osob. Uvedená nevinná matematická hříčka má v kryptologii vážný dopad. Útoky hrubou silou jsou v případech použitelnosti narozeninového paradoxu totiž mnohem snazší a říká se jim pak narozeninový útok.

Pravděpodobnost, že náhodně vybraná dvojice textů X a Y bude mít stejnou haš tedy, že $h(X) = h(Y)$ bohužel existuje. Předcházet kolizím je obtížné a je nutné brát na tuto skutečnost zřetel již při konstrukci algoritmu hašovací funkce.

Existuje několik základních způsobů řešení problému.

1. Zřetězené hašování
2. Otevřené adresování a lineární vkládání
3. Otevřené adresování a dvojité hašování

Jaké je tedy využití hašovacích funkcí? Hašovací funkce přiřazuje každé libovolně dlouhé posloupnosti dat jedinečný identifikátor stejné délky. Této vlastnosti je využíváno v široké oblasti informačních technologií. Je nutné zde upozornit, na jeden zdánlivý rozpor v předcházejícím textu. Na jedné straně popisuji hašovací funkce jako vhodné funkce pro zakódování libovolné zprávy. Na straně druhé mluvíme o reálné možnosti kolize a zároveň je evidentní, že zpráv může být veliké množství a hašovacích kódů je pouze 2^n, kde n je bitová délka výstupu hašovací funkce. Vysvětlení je jednoduché, přestože existuje kolizí velké množství, je nalezení byť jen jediné kolize nad možnosti soudobé výpočetní techniky.

2.2 Algoritmy pro hašovací funkce

2.2.1 MD5[2]

MD5 je jednosměrná hašovací funkce, kterou vyvinul Ronald Rivest z MIT (Massachusetts Institute of Technology) v roce 1991. Postupně bylo v této funkci nalezeno několik vad, ale funkce je stále používána. V roce 2004 byly nalezeny další závažnější bezpečnostní chyby, nicméně i přesto všechno je hašovací funkce stále velmi používaná.

Název MD5 vznikl z anglického Message-Digest algorithm 5 a v současné době je popsána v internetovém standartu RFC 1321. Hlavní využití této funkce spočívá v malých aplikacích, například pro ověřování hesel v databázi.

MD5 je jednou z nejznámějších hašovacích funkcí a má 128 bitový výstup. V minulosti se, i přes její objevené slabiny, jednalo se spolu s funkcemi SHA o nejpoužívanější hašovací funkci vůbec. MD5 byla navržena Ronem Rivestem v roce 1991, aby nahradila dřívější hašovací funkci stejného řady-MD4, která byla shledána nevyhovující. Ale ani MD5 není dokonalá, v roce 1996 v ní byla nalezena vada a i když to nebyla chyba fatální, začalo být doporučováno používat spíše jiné hašovací funkce. Další vady byly odhaleny v srpnu roku 2004, což vyvolalo spory ohledně používání MD5 k bezpečnostním účelům. V březnu 2005, Arjen Lenstra, X.Wang a Benne de Weger předvedli konstrukci dvojice různých vstupů, jež mají společný kontrolní součet. Tím byla nalezena kolize. O pár dní později RNDr. Vlastimil Klima popsal dokonce vylepšený algoritmus této konstrukce, který je schopen nalézt kolizi během jedné minuty počítání na běžném počítači. Klíma nazval tuto metodu tunelování [9].

Hašovací funkce MD5 je široce využívána v počítačových programech, aby zajistila jistou záruku, že přenášena data byla doručena neporušena. Například databázové servery často poskytují spočítané kontrolní součty spolu s daty. Takže uživatel si s nimi může porovnat kontrolní součet, který provede na těch datech, která si stáhl. Hašovací funkce MD5 je prakticky (nejen teoreticky) zranitelná pro většinu svých použití a měla by být nahrazena jinou, standardní a ověřenou funkcí.

MD5 vždy pracuje s daty, která mají celkovou délku v bitech násobku 512-ti. Pokud není velikost násobkem tohoto čísla, musí se doplnit na požadovanou velikost.

---

[2] Message-Digest algorithm 5, skupina hašovacích funkcí

Doplňuje se následně:

- jeden bit hodnoty 1 je přidán tak, že délka zprávy je o 64 bitů menší než je konečný násobek 512
- chybějících 64 bitů má potom své uplatnění jako uchování délky zprávy pře doplněním, aby nebyla informace ztracena.

Doplňování se provádí vždy, tedy i v případě, že je velikost násobkem 512-ti. Když je vstupní řetězec upraven, nic nám nebrání zjistit hodnotu, která závisí na opakované modifikaci 128-bitové hodnoty popisující stav.

Pro zpracování každého 128-bitového stavu je stav rozdělen na 4 bloky po 32 bitech, označených A, B, C, D a na začátku se každému z nich nastaví výchozí hodnoty. Každý z těchto 4 bloků je potom zpracován nezávisle na ostatních a různě modifikován, přičemž modifikace probíhají ve 4 stupních, které se nazývají kola. Jednotlivá kola se skládají z 16 operací, což je celkem (4*16) 64 operací pro každý blok dat.512-bitový blok dat je rozdělen na 16 datových slov. (každé obsahuje 32 bitů) a uvnitř každého kola je jedna z následujících operací [1]:

F(X, Y, Z) = (X AND Y) OR (NOT(X) AND(Z))

G(X,Y,Z) = (X AND Z) OR (Y AND NOT(Z)),

H(X,Y,Z) = X XOR Y XOR Z,

I(X,Y,Z) = Y XOR (X OR NOT(Z)).

2.2.2 SHA-x

První specifikace algoritmu známého jako SHA - 0 byla zveřejněna v roce 1993 jako Secure Hash Standartd, FIPS PUB 180 (Federal Information Processing Standards Publication 180-1). Tuto specifikaci zveřejnila agentura NIST (National Institute of Standards and Technology) ze Spojených států amerických. Jednou z nejvýznamnějších událostí v roce 2004 bylo objevení kolizí pro skupinu hašovacích funkcí MD4, MD5, HAVAL-128 a RIPEMD čínským týmem. Jejich autoři (Wangová a kol.) však utajili metodu nalézání kolizí a zveřejnili pouze strohá data a informace. V říjnu 2004 se australský tým (Hawkes a kol.) pokusil tuto metodu zrekonstruovat ve skvělé práci.

Nejdůležitější princip se nepodařilo objevit, ale na základě dat bylo dobře popsáno diferenční schéma, kterým uveřejněné čínské kolize vyhovují [10].

V následujících letech je v hojné míře používán algoritmus SHA-0, u kterého je však velice brzy nalezena chyba a je nahrazen SHA-1. Hašovací funkce serie SHA jsou vytvořeny National Secutity Agency of USA a byly publikovány ve vládním standartu. Ten generuje jako svůj výstup 160 bitový řetězec, velikost vstupní zprávy pro algoritmus je omezena hodnotou 2^{64} .Algoritmus je využíván především v oblasti digitálního podepisování a v oblasti ověřování integrity dat.

Zatímco v prosinci 2008 se podařilo u algoritmu MD5 najít kolizi po 2^{51} proběhnuvších výpočtech, pro překonání SHA-1 bylo doposud potřeba 2^{62} výpočtů. Na konferenci Eurocrypt 2008 však tým výzkumníků z australské Macquarie University předvedl výsledky studie, podle které k prolomení SHA-1 a třeba k vygenerování pirátského SSL certifikátu uznávané autority, která jej ale nikdy nevydala, postačí 2^{52} operací. SHA-1 je tak v současné době téměř stejně rychle napadnutelný jako MD5 na počátku roku 2008. Na zjištění rychle reaguje Ministerstvo vnitra a na základě vyhlášky č. 378/2006 Sb., o postupech kvalifikovaných poskytovatelů certifikačních služeb a dokumentu ETSI TS 102 176-1 V2.0.0 (ALGO Paper) Ministerstvo vnitra stanoví: Kvalifikovaní poskytovatelé certifikačních služeb ukončí vydávání kvalifikovaných certifikátů s algoritmem SHA-1 do 31. 12. 2009. Od 1. 1. 2010 budou tito poskytovatelé vydávat kvalifikované certifikáty podporující některý z algoritmů SHA-2. Zároveň je od uvedeného data stanovena minimální přípustná délka kryptografického klíče pro algoritmus RSA na 2048 bitů [11].

Institut NIST[3] publikoval další čtyři hašovací funkce ve skupině SHA, všechny jsou označovány zkratkou SHA-2, přičemž zkratka nebyla nikdy standardizována. Jednotlivé varianty, označené podle bitové délky výsledného řetězce SHA-224, SHA-256, SHA-512. U funkcí SHA-2 dochází ke komplikovanějšímu výpočtu a k expanzi vstupního bloku zprávy, než je tomu u funkcí SHA-1 nebo u starších hašovacích funkcí.

---

[3] National Institute of Standards and Technology - Národní institut standardů a technologie je institut při Ministerstvu obchodu USA

2.2.3 Další hašovací funkce

- **RIPEMD**

Jedná se o skupinu hašovacích funkcí, které byly navrženy z důvodu kompatibility s aplikacemi, které pracují s délkou klíče 128 bitů

- **Haval**

Haval je hašovací algoritmus, který byl publikován v roce 1997. Funkce předpokládá jako vstupní blok data menší než 2^{128} bitů, pak je blok dat doplněn na určitou délku. Takto upravená data jsou rozdělena do bloků o délce 512 bitů. Výsledný hašovací řetězec vznikne výpočtem n cyklů

- **Tiger**

Hašovací funkce, která byla navržena v roce 1995 a je speciálně určena pro 64 bitové platformy. Funkce je založena na principu iterativního[4] výpočtu s nelineárními funkcemi.

- **Whirlpool**

Whirlpool je hašovací funkce, která byla postupně vyvíjena a měla tři verze. Operuje se zprávami nejvýše do velikosti 2^{256} bitů a produkuje klíč o celkové délce 512 bitů. Funkce využívá blokovou šifru, kde je vstupní řetězec nejprve doplněn sekvencí jedniček, poté sekvencí nul. Zarovnaná zpráva je rozdělena do bloků o celkové délce 512 bitů, bloky jsou následně využity pro generování pomocných hašovacích klíčů

- **Panama**

Panama může být využita jako hašovací funkce nebo jako proudová šifra[5], poprvé byla prezentována v roce 1998, pracuje na principu posuvného registru s posouváním bitového slova o velikosti 32 sx 32 bitů. V roce 2007 byl předveden útok na tuto funkci, který umožnil najít kolizní pár a funkce Panama se stala nevhodnou k praktickému použití.

- **Ghost**

Hašovací funkce, která byla poprvé zveřejněna v roce 1994. Původně byla definována jako ruský národní standart a je jedinou hašovací funkcí, která může být využita v ruském digitálním podpise. Jedná se o funkci, která pracuje na principu iterace a produkuje 256

---

[4] opakované přepočítávání až do té doby, než je splněna určitá podmínka

[5] Proudová šifra šifruje zvlášť jednotlivé znaky abecedy, zatímco bloková šifra zpracovává najednou bloky (řetězce) délky t znaků.

bitů dlouhý výsledný klíč. Funkce Ghost navíc kromě běžné struktury iterace definuje i výpočet kontrolního součtu nad všemi vstupními bloky zprávy..

- **Grindahl**

Hašovací funkce Grindahl byla poprvé představena v roce 2007, je založena na blokové šifře. Základním stavebním blokem této hašovací funkce je pole 4 x 13 bytů. Pole je pětkrát transformováno, dochází k cyklickému posouvání bitů řádků v závislosti na délce, dále je provedena transformace sloupců a řádků. Nakonec je ještě připojeno zakončení a následně je provedeno dalších osm cyklů funkce

- **Radiogatún**

Funkce Radiogatún byla poprvé představena v roce 2006, je založena na hašovací funkci Panama, ale funkce je navržena tak, aby odolávala všem známým útokům. Radiogatún obsahuje dva základní prvky Belt a Mill (Pás a Mlýn), operace, které jsou využity v této hašovací funkci jsou standardní bitové operace a cyklický posuv bitových slov. Jednou z výhod této funkce je vysoká výkonnost a díky kompaktnosti funkce je možná i její implementace přímo do hardwaru.

3 SYMETRICKÉ ŠIFRY

K úspěšnému zašifrování a předání zprávy od odesilatele k adresátovi je zapotřebí dvou klíčů. Prvním z nich je klíč šifrovací, který slouží odesilateli k převodu původní zprávy na zprávu zakódovanou a druhým z nich je dešifrovací klíč, který slouží adresátovi k převodu zakódované zprávy na původní. Na základě způsobu užití těchto dvou klíčů se dělí kryptografické metody na symetrické a asymetrické. U symetrické šifry jsme ze znalosti šifrovacího klíče schopni zjistit klíč dešifrovací a naopak. U asymetrické šifry tomu tak není. Oba dva klíče jsou na sobě nezávislé a ze znalosti šifrovacího klíče je velmi obtížné a velmi časově náročné zjištění klíče dešifrovacího. Nejprve popíšeme kryptografické systémy, které obecně zajišťují bezpečnostní službu důvěrnosti dat, a to pomocí kryptografického nástroje šifrování dat. Tyto systémy se nazývají šifrovací systémy neboli šifry. Nejprve si uvedeme definici, která platí jak pro symetrické, tak pro asymetrické šifry. Kryptografickým systémem pro šifrování zpráv (tzv. šifrou) budeme rozumět: Kryptografický systém pro šifrování zpráv je pětice (M, C, K, E, D), kde M je prostor otevřených zpráv, C prostor šifrových zpráv a K prostor klíčů. E, D je dvojice zobrazení, které každému klíči $k \in K$ přiřazují transformaci pro zašifrování zpráv E_k a transformaci pro dešifrování zpráv D_k, přičemž pro každé $k \in K$ a $m \in M$ platí $D_k(E_k(m)) = m$. Klasická kryptografie se zabývala především šiframi, tj. způsoby utajení zpráv. Patřily sem zejména šifrovací systémy jako jednoduchá záměna, jednoduchá a dvojitá transpozice. Tyto systémy zpravidla nazýváme historickými šifrovacími systémy. Druhá světová válka přinesla nebývalý zájem o kryptografii a o kryptoanalýzu. V poválečném období se začíná více rozvíjet i teorie. V rámci tohoto poválečného dění Claude E. Shannon nejprve v roce 1948 publikoval práci A Mathematical Theory of Communication, která je pokládána za základ teorie informace, a rok poté práci Communication Theory of Secrecy Systems, která je pokládána za základ moderní kryptologie. Shannon využil pojmů z teorie informace k ohodnocení bezpečnosti známých šifer. Definoval entropii jazyka, vzdálenost jednoznačnosti, dokázal absolutní bezpečnost Vernamovy šifry, zavedl pojmy difúze a konfúze a ukázal, jak posuzovat a konstruovat šifrové systémy kombinací různých typů šifer. Zavedl také model komunikačního kanálu, který se používá při popisu kryptografických systémů dodnes.

3.1 Symetrické šifry

Symetrickým kryptografickým systémem pro šifrování zpráv budeme nadále rozumět takovou šifru, kde pro každé k $\in$ K lze z transformace zašifrování E_k určit transformaci dešifrování D_k a naopak. Z důvodu této symetrie se tyto systémy nazývají symetrické systémy a jejich klíče symetrické klíče. Symetrické klíče jsou tajné, zatímco obě zobrazení E a D mohou být zcela veřejná, jako je tomu například u šifrovacích standardů DES[6] a AES[7].

3.2 Proudové šifry

Klasická definice proudových šifer zní, že zpracovávají otevřený text po znacích, zatímco blokové šifry po blocích t znaků. Proudové šifry by tedy mohly být chápány i jako blokové šifry s blokem délky t =1, avšak připomeňme, že tou podstatnou odlišností je, že u proudových šifer je každý "blok" zpracováván jiným způsobem, jinou substitucí. Nechť A je abeceda q symbolů, nechť M = C je množina všech konečných řetězců nad A a nechť K je množina klíčů. Proudová šifra se skládá z transformace (generátoru) G, zobrazení E a zobrazení D. Pro každý klíč k ϵ K generátor G vytváří posloupnost hesla h(1), h(2),... , přičemž prvky h(i) reprezentují libovolné substituce $E_{h(1)}$, $E_{h(2)}$, ... nad abecedou A. Zobrazení E a D každému klíči k ϵ K přiřazují transformace zašifrování E_k a odšifrování D_k. Zašifrování otevřeného textu m = m(1), m(2), ... probíhá podle vztahu

$c(1) = E_{h(1)}(m(1))$, $c(2) = E_{h(2)}(m(2))$, ...

a dešifrování šifrového textu c = c(1), c(2), ... probíhá podle vztahu

$m(1) = D_{h(1)}(c(1))$, $m(2) = D_{h(2)}(c(2))$, ... kde $D_{h(i)} = E_{h(1)}^{-1}$.

- **Vernamova šifra**

V roce 1917 si Gilbert Vernam dal patentovat vylepšení dřívějších způsobů šifrování. Vezmeme jednotlivá písmena tajné zprávy a každé z nich posuneme o několik pozic v abecedě. Například pokud ve zdrojovém řetězci FAI ZLÍN uplatníme posloupnost P =

[6] DES - Data Encryption Standard, první veřejný šifrovací standart

[7] AES - Advenced Encryption Standard, nahrazuje DES

{3,4,5,6,7,8,9}, tak získáváme zašifrovaný text IENFSQW. Klíčem k rozluštění je P, kdo ho zná, dokáže snadno posunout písmena opačným směrem a získat původní text. Bez znalosti klíče je luštění odposlechnuté zprávy krajně obtížné, i když útočník ví, o jakou šifru jde.

Aby byla Vernamova šifra spolehlivá, je nutno dodržet tři požadavky:

1. Klíč je stejně dlouhý jako přenášená zpráva. (Ve starších šifrách to bylo jinak.),
2. Klíč je dokonale náhodný. (Generátory pseudonáhodných čísel nepřipadají v úvahu, nejlepší je užití fyzikálních metod.),
3. Klíč nelze použít opakovaně. (Žádné dvě zprávy nesmí být šifrovány stejným klíčem.).

Porušení libovolného z těchto pravidel umožní útočníkovi odhalit tajný text, postupy jsou známé. Naopak když požadavkům vyhovíme, můžeme si být bezpečností svých dat velmi jisti. Ani útok tzv. hrubou silou nepomůže. Jeho výsledkem budou všechny možné zprávy dané délky, mezi nimiž nepoznáme, která byla odesílána. Lze matematicky dokázat, že bez znalosti klíče nelze zašifrovanou zprávu rozeznat od náhodné posloupnosti písmen. Vernamovu šifru nazýváme absolutně bezpečnou šifrou (perfect secrecy) , je dosud jedinou šifrou, jejíž neprolomitelnou byla exaktně dokázána (1949, C. E. Shannon). Vernamova šifra se používala a mnohde ještě používá pro ochranu nejdůležitějších, zejména diplomatických spojů, kde je nutné mít zajištěnu absolutní bezpečnost. Nevýhodou je nutnost distribuovat heslo na obě strany komunikačního kanálu. Dříve se heslo děrovalo do děrných pásek a bylo na zastupitelské úřady dopravováno v diplomatických zavazadlech, dnes mohou být nosiče těchto klíčových materiálů jiné, ale podstata zůstává stejná.

Popsané zacházení s klíčem je ale v praxi velice obtížné. Dlouhý náhodný klíč si člověk nedokáže zapamatovat, musí být zaznamenán. Jeho generování není jednoduché. Musí být zajištěno, že klíč zná pouze odesilatel a příjemce zprávy a nikdo jiný. Komunikující strany se tedy musí předem dohodnout na dlouhém klíči nějakým bezpečným způsobem a hned po odeslání první zprávy klíč zničit. Stojíme tak před problémem slepice a vejce: Abychom mohli bezpečně odeslat třeba 2 MB tajných dat, musíme předem bezpečně odeslat 2 MB dat (klíč). Vernamova šifra se tak i přes svou sílu používala jen výjimečně, respektive mnohem častěji se používají různé modifikace této šifry, kdy není dodržena podmínka č.1 pro délku klíče.

V roce 1984 navrhli Charles Bennett a Gilles Brassard kryptografický protokol postavený na kvantové mechanice. Podle počátečních písmen objevitelů a podle roku objevu se protokol označuje BB84 a je označován jako kvantový protokol výměny klíče. K přenosu informací se používají současně dvě přenosové cesty, dva kanály. Jeden je klasický, například telefon nebo Internet. Odposlouchávání na tomto kanále neprozradí nic tajného, půjde přes něj zašifrovaná zpráva a několik vedlejších informací. Druhý kanál je kvantový a slouží k domluvení tajného klíče. K přenosu informace využívá fotonů s různou polarizací. Možné jsou 4 roviny polarizací fotonu, dvě z nich reprezentují jedničkový bit, druhé dvě nulový. V souvislosti s kvantovou kryptografií se dostává do popředí pojem nepodmíněná bezpečnost. Značí, že bezpečnost komunikace není podmíněna žádnými předpoklady na schopnosti a technické možnosti útočníka. Bezpečnost dosud nejčastěji používaných kryptografických systémů je založena na výpočetní složitosti čili na faktu, že nejsou známy dostatečně rychlé postupy a dostatečně výkonné počítače na vyřešení určitých úloh. Kvantová kryptografie žádné takové předpoklady neobsahuje. Ani síla nejrychlejších, ještě nesestrojených počítačů ani jakýchkoliv jiných systémů nemůže porušit přírodní zákony, o které se uvedený systém opírá.

- **Šifra RC4**

RC4 je klasický symetrický algoritmus s tajným klíčem. Je to proudová šifra, kterou navrhl Ronald Rivest (RC znamená Rivest s Cipher), jeden z vynálezců algoritmu RSA a spoluzakladatel společnosti RSA DSI. Vstupem RC4 je klíč o volitelné délce, teoreticky až 256 bajtů. Klíč inicializuje konečný automat, který pak generuje posloupnost bajtů hesla h(0), h(1), ... Při zašifrování se heslo □xoruje□ na otevřený text a při odšifrování na šifrový text, tedy: □t(i) = ot(i) xor h(i), i = 0, 1, ...

3.3 Blokové šifry

3.3.1 Šifra PlayFair

Tuto šifru navrhl v roce 1854 britský vědec Charles Wheatstone (6. února, 1802 - 19. října, 1875) a to jako vhodnou šifru pro utajení telegrafických zpráv. Jméno však dostala až podle skotského barona Lyon Playfaira (1. května, 1818) - (29. května, 1898), který byl velkým propagátorem této šifry. Šifra se nakonec prosadila především jako vojenská šifra.

Britská armáda ji používala během obou Búrských válek (1880-81, 1899-1902) a byla jí používána i za I. světové války. Australská armáda ji dokonce používala i během svých válečných operací za druhé světové války. Výhodou této šifry je, že je daleko hůře luštitelná než jiné klasické „ruční šifry". Její hlavní předností je, že je odolná (na rozdíl např. od jednoduché záměny) proti frekvenční analýze[8]. Z hlediska kryptoanalýzy se vlastně jedná o bigramovou záměnu – tedy záměnu, kdy dvojice písmen otevřeného textu se zamění za jinou dvojici písmen. Je pochopitelné, že frekvenční analýza bigramů vyžaduje záchyt výrazně většího počtu šifrových textů než luštění jednoduché záměny. Vojenským expertům se proto zdál systém pro masové použití v armádě vhodný a spolehlivý. Zvláště při dodržení dalších bezpečnostních pravidel, jako pravidelná výměna klíčového slova, nezasílání velkého objemu korespondence apod. Dále jeho výhodou byla rychlá výuka uživatele a příprava šifrového textu, zároveň i rychlá dešifrace textu a celkově malá cena na masové nasazení.

Popis systému:

Uživatel pomocí šifry PlayFair vytvoří z otevřeného textu šifrový tak, že nejdříve otevřený text podle jednoduchých pravidel upraví a potom jej pomocí abecedního čtverce záměny podle pěti prostých pravidel transformuje (zašifruje). Abecední čtverec záleží na dohodnutém hesle - klíčovém slově. Celý text přepíšeme na text složený pouze z velkých písmen, bez diakritiky a interpunkce a pokud obsahuje text písmeno J, všude ho zaměníme na I (v angličtině se J vyskytuje velmi zřídka). Pokud by se v bigramu objevila dvě stejná písmena, musí být oddělena písmenem X a Z. Pokud má původní text lichý počet písmen, doplníme na konec textu opět písmeno X nebo Z.

Šifrování:

Šifrování systémem PlayFair je založeno na skutečnosti, že každý bigram v upraveném otevřeném textu se může vyskytnout pouze v jednom ze tří následujících stavů.Bigram může být společně v jednom řádku, jednom sloupci nebo je každé z písmen bigramu v

[8] Frekvenční analýza je analýza četnosti výskytu určité vlastnosti, procesu nebo jevu. V oblasti zpracování textu metoda analýzy formální struktury textu založená na měření četnosti výskytů slov či slovních spojení, vycházející z předpokladu, že výrazy s vyšší frekvencí jsou pro text významnější než výrazy s nízkým počtem výskytů a lze z nich proto usuzovat na jeho obsah.

jiném řádku a sloupci (statisticky nejběžnější situace). Samotné šifrování proto probíhá takto:

- Pokud leží obě písmena ve stejném řádku, je každé písmeno bigramu nahrazeno písmenem ležícím v tabulce vpravo od něj. Poslední písmeno v řádku (tedy pokud nemá vpravo od sebe písmeno) se nahradí prvním písmenem téhož řádku.
- Pokud leží obě písmena ve stejném sloupci, je každé písmeno bigramu nahrazeno písmenem pod ním. Je-li písmeno v posledním řádku (tedy pokud nemá pod sebou písmeno) je nahrazeno prvním písmenem téhož sloupce.
- Pokud je každé z písmen bigramu v jiném řádku a sloupci, je každé písmeno digramu nahrazeno písmenem nacházejícím se v průsečíku jeho vlastního řádku a sloupce obsahujícího druhé písmeno bigramu. Musí se dodržet pořadí: nejdříve se určí průsečík řádku prvního písmene se sloupcem druhého písmene, potom teprve průsečík řádku druhého písmene se sloupcem prvního písmene. S výhodou se používá představa, že dvě písmena upraveného otevřeného textu vytvářejí uvnitř abecedního čtverce dva vrcholy obdélníka a písmena zašifrovaného textu leží v opačných vrcholech tohoto obdélníka.

Výsledný šifrový text se zapisuje do pětic oddělených jednou mezerou.

4 ASYMETRICKÉ ŠIFRY

Asymetrická šifra můžeme definovat jako takovou šifru, kde pro skoro všechna $k \in K$ nelze z transformace pro zašifrování E_k určit transformaci pro dešifrování D_k. V praxi je u asymetrických šifer klíč k tajným nastavením, z kterého se vhodnou transformací G vygeneruje dvojice parametrů (e, d), které se nazývají po řadě veřejný (e) a privátní (d) klíč. Ty potom parametrizují transformace zašifrování a dešifrování, takže pro jednoduchost nepíšeme E_k a D_k, ale přímo E_e a D_d.

V roce 1976 publikovali Whitfield Diffie, Martin Hellman a Ralph Merkle článek o nových možnostech kryptografie, čímž byly položeny základy asymetrických kryptografických metod. Hned vzápětí (1978) vzniká první asymetrický šifrový systém - RSA. Tyto metody se snaží odstranit problémy a nedostatky symetrické kryptografie.

Algoritmy používají různé klíče pro šifrování a dešifrování (tzv. klíčový pár, obsahující veřejný a soukromý klíč). Veřejný klíč není třeba tajit a používá se pro šifrování zprávy či ověření podpisu. Soukromý klíč je využíván k dešifrování či k vytváření podpisu. Před začátkem komunikace je třeba předat pouze veřejné klíče. Jelikož neobsahují žádné tajemství, mohou být distribuovány libovolnou cestou. Důležité je ověřit autenticitu a integritu přijatého klíče . K zahájení komunikace není nutno přenášet ani sdílet žádné tajemství. S jediným párem je možno bezpečně komunikovat s neomezeným množstvím uživatelů. Asymetrická kryptografie je založena na jednosměrné funkci se zadními vrátky. Tedy na funkci, z jejíhož výsledku nelze získat vstupní hodnotu bez znalosti dodatečného tajemství. Taková ideální funkce není známa, ale blíží se k ní funkce založené na obtížných matematických problémech. Výpočet je však výrazně snazší pokud známe více údajů[9].

Síla kvalitní asymetrické šifry závisí stejně jako u symetrické na délce klíče. Se znalostí algoritmu však lze předem vyřadit některé možnosti, které nemohou být řešením matematického problému. Pro zachování bezpečnosti přenosu je třeba použít výrazně delší klíče než u symetrické kryptografie. Kvůli výpočetně složitějším algoritmům a delším klíčům je výrazně pomalejší než symetrická kryptografie.

---

[9] Příkladem je součin velkých prvočísel. Výpočet je relativně snadný. Rozložit výsledek zpět na prvočísla je však netriviální problém. Pokud je ovšem známo jedno z uvedených prvočísel, je to opět jednoduché. Na podobném principu fungují všechny asymetrické šifry.

I asymetrická kryptografie však má své slabé stránky. Za potenciální bezpečnostní riziko lze totiž považovat samotný princip založený na matematickém problému, jehož řešení není známo. Nelze vyloučit, že v budoucnu bude nalezeno snadné řešení onoho problému a všechny šifry na něm založené se ze dne na den stanou snadno rozluštitelnými. Na rozdíl od rychlosti vývoje výpočetní techniky nelze odhadnout, zda a kdy se to vědcům podaří. I přes uvedený fakt jsou asymetrické šifry s užitím dlouhých klíčů považovány za dostatečně silné.

4.1 Algoritmus RSA

Brzy po zveřejnění teoretického schématu asymetrické kryptografie (1978) se objevuje první šifrový systém založený na této myšlence. Vžil se pro něj název RSA. Systém se po malých úpravách, především prodloužení klíče a stanovení jistých pravidel, která musí klíče splňovat používá dodnes. Je založen na obtížném matematickém problému , na rozkladu velkých čísel na prvočísla.

Podrobněji se budu tímto algoritmem zabývat v praktické části této práce.

5 ELEKTRONICKÝ PODPIS

5.1 Právní vymezení pojmu elektronický podpis

Vymezení pojmu elektronický podpis se může zdát na první pohled velmi jednoduché, avšak velká část problematiky elektronického podpisu má převážně technický a technologický charakter. Elektronický podpis spočívá v nahrazení klasického podpisu na papíru podpisem elektronického dokumentu, při současném zachování nebo dokonce zvýšení bezpečnosti celé podpisové operace. Principem podepisování je připojení určitého identifikátoru ke zprávě, přičemž tímto identifikátorem může být heslo, obraz – například otisku prstu či celé ruky, kryptografický kód apod. – vše v digitálním tvaru. Dnes se používá prakticky výlučně metoda digitálního podpisu, pracující na bázi asymetrické kryptografie. Digitální tvar elektronického podpisu však nebyl založen na této bázi od počátku. Zprvu byla navrhována řešení elektronického podepisování dokumentů cestou symetrických šifrovacích algoritmů, což bylo později nahrazeno výše zmíněnou metodou asymetrické kryptografie. Je možné, že v budoucnosti může technologický vývoj přinést zcela novou metodu, nebo dokonce nahradit současný digitální tvar elektronického podpisu jiným než digitálním tvarem. Je rovněž třeba připomenout, že elektronický podpis je výstupem určitého procesu, do kterého jako parametry vstupuje soukromý klíč a podepisovaný dokument. Uživatel tedy nikdy nebude vlastnit svůj elektronický podpis, ale nástroje pro jeho vytváření a pro jeho ověřování [5]. Elektronický podpis totiž ve své podstatě není ničím jiným, než číslem. Které je tak velké, že by nebylo vhodné ho zapisovat jako binární číslo. Takže pokud je někdy třeba ho zapsat tak, aby to bylo alespoň trochu srozumitelné pro člověka, využívá se k tomu efektivnější reprezentace (kódování), tak aby se vystačilo s méně znaky. Aby byla situace více nepřehlednější, elektronický podpis definuje i Evropská směrnice.

5.2 Elektronický podpis – obecné vymezení

Vyjdeme z vymezení, které je obsaženo v § 2 písm. a) zákona o elektronickém podpisu: elektronickým podpisem (se rozumí pro účely tohoto zákona) údaje v elektronické podobě, které jsou připojené k datové zprávě nebo jsou s ní logicky spojené a které umožňují ověření totožnosti podepsané osoby ve vztahu k datové zprávě. Obdobně popisuje elektronický podpis i Směrnice v článku 2 odst. 1. Požadavky na námi definované a sledované kategorie jsou tedy zcela minimální. Nepožaduje se časové razítko, není

definován žádný konkrétní formát nebo standard, který by popisoval tvar vytvořených nebo předávaných dat. Není použit certifikát nebo jiný způsob zveřejnění pomocných dat ani uvedená data nejsou definována. Takový typ odpisu nemá pro příjemce příliš velkou vypovídací hodnotu a důvěra v něj je minimální. Slouží spíše pro informaci příjemce. Příkladem může být podpis vložený pod klasický e-mail, ale i vložené jméno autora článku. Skutečnost, že i popsaný podpis je podpisem ve smyslu zákona 227/2000 Sb., vyplývá z § 3 odst. 1, kde je napsáno: Datová zpráva je podepsána, pokud je opatřena elektronickým podpisem. Na tomto místě je ovšem nutné zdůraznit, že zákon o elektronickém podpisu upravuje především náležitosti zaručeného elektronického podpisu a obyčejným elektronickým podpisem se dále nezabývá, je v zákoně použit pouze podpůrně. To je také důvodem, proč se v praxi pod pojmem elektronický podpis většinou rozumí zaručený elektronický podpis. Ustanovení § 2 písm. a) zákona o elektronickém podpisu obsahuje zákonné vymezení pojmu elektronický podpis, kterým rozumí údaje v elektronické podobě, které jsou připojené k datové zprávě nebo jsou s ní logicky spojené, a které slouží jako metoda k jednoznačnému ověření identity podepsané osoby ve vztahu k datové zprávě. V tomto případě se jedná o obyčejný elektronický podpis. S tímto druhem podpisu se můžeme setkat například v bankách při porovnávání podpisu na papíru s podpisovým vzorem, neskenovaným a uloženým v paměti počítače. Jedná se však o postup ryze subjektivní, protože takovéto srovnání je pouze vizuální a záleží na momentální kondici podepisujícího i na schopnosti zaměstnance banky, aby odhalil, zda jde o padělek. V tomto pojetí se může elektronickým podpisem rozumět i podpis v textu e-mailové zprávy.

5.3 Elektronický podpis – historie

Právní aspekty elektronických podpisů byly řešeny velmi intenzivně na všech úrovních. V rámci Komise OSN pro mezinárodní obchodní právo (UNCITRAL[10]) byl zpracován tzv. Vzorový zákon o elektronickém obchodu, který byl Valným shromážděním OSN v r. 1996 schválen. V návaznosti na zákon se dokončila obecná pravidla UNCITRAL pro elektronické podpisy, která sjednotila zejména právní aspekty elektronických podpisů, certifikačních orgánů a certifikátů v celosvětovém měřítku. Zajímavé je, že již v návrzích

[10] United Nations Commission on International Trade Law – Komise OSN pro mezinárodní obchodní právo

pracovní skupiny pro Elektronický obchod z roku 1998 je definován podpis, elektronický a také zaručený elektronický podpis na základě podpisu digitálního. Poté došlo k řadě změn v pojetí podpisů ze strany UNCITRAL, který mění původní značně technologicky závislou koncepci elektronických podpisů, v podstatě pouze na bázi podpisů digitálních, na koncepci značně obecnou, umožňující použití i jiných technologií. Následně Evropský parlament a Rada Evropské unie schválily 13.12.1999 Směrnici 1999/93/EC pro elektronické podpisy v rámci společenství s cílem usnadnit používání elektronických podpisů a přispět k jejich právnímu uznání v prostředí členských států EU.

Na úrovni jednotlivých států přijímá i naše republika zákon č. 227/2000 Sb., o elektronickém podpisu, který v ČR nabyl účinnosti 1.10.2000. V tomto zákoně je definován zaručený elektronický podpis a podmínky jeho používání [6].

5.4 Nevhodné pokusy pro elektronický podpis

V následujících kapitolách se budeme zamýšlet nad tím, co by bylo možné používat jako elektronický podpis. Určitě nás na jednom z prvním míst napadne, že by se dal vytvořit soubor dat, který by měl každý jedinec u sebe uschovaná a v případě potřeby je připojil k souboru. Připojením této pečeti by byl jednoznačně identifikován podepisující. Takováto představa by stále mohla zajistit to, aby elektronický podpis byl spojen s konkrétní osobou, pokud by pečetě byly vytvářeny jako individuální a nikoli jako klasické známky, které jsou vždy stejné a nezávislé na tom, kdo si je koupí a na něco nalepí. Jinými slovy: každý by potřeboval své vlastní a individuální pečetě. Nebyl by splněn požadavek na to, aby elektronický podpis umožnil detekovat jakoukoli změnu podepsaného dokumentu čili nebylo by možné zaručit integritu dat a ani by nebylo možné z takto podepsaného dokumentu vyvozovat, že podepsaný s výše uvedeným textem souhlasí. Kvůli tomu, že elektronický podpis jako pečeť by byl zcela nezávislý na tom, co jím je podepsáno. Konkrétním příkladem toho typu elektronického podpisu by mohlo být naskenování vlastnoručního podpisu do podoby obrázku a přidání tohoto obrázku ke konkrétnímu textu v elektronické podobě, nejspíše v textovém editoru. Na tomto příkladu si můžeme vytvořit představu, že takto podepsaný dokument je nám k ničemu. A to i přestože je evidentně podpis pravý. Představu podpisu jako pečeti musíme vyloučit.

V myšlenkách, co by mohlo být elektronickým podpisem si můžeme vytvořit i následující model. Nebudeme mít v zásobě pečetě, ale budeme mít program, který vytvoří obdobu razítka, které známe z reálného života. Zde se již dostáváme k silnějšímu nástroji na tvorbu

našeho elektronického podpisu, protože již nemáme někde uloženo několik pečetí do zásoby. Razítka používáme až po přečtení dokumentu, je možné říci, že jediné co nám otisk razítka nezaručuje je integritu podepsaných dokumentů. Nemohli bychom zjistit, zda dokument byl či nebyl někdy v následném časovém úseku změněn či nikoliv. A to je samozřejmě podmínka veskrze zásadní.

5.5 Způsob tvorby elektronického podpisu

Když se budeme zabývat výše popisovanou představou s razítkem, narazíme na další problém. Bylo by asi velice složité vytvořit určitý program, který by vytvářel popisovaný otisk razítka. Protože by program musel být přísně individuální, tím pádem by mohl být použitelný pouze pro jednu fyzickou osobu. To by bylo určitě realizovatelné, ale ve svém důsledku by to bylo velice nepraktické. Program by nebylo možné pravděpodobně si volně zakoupit, protože by musel být vytvářen speciálně ke každé individualitě. Což by vedlo v důsledku znamenalo vysokou cenu produktu, zároveň by byla i drahá individuální distribuce produktu. Zároveň by se musel vytvořit i specifický systém ochrany programu proti zneužití i kopírování.

5.6 Schéma podpisu vytvořeného na základě symetrické šifry

Schématem elektronického podpisu budeme rozumět takové použití kryptografického algoritmu, které povede k autentizaci určitých dat. Nejdříve naznačím princip pro elektronický podpis, který vychází ze symetrických šifer a v další kapitole se budu věnovat podrobněji tvorbě podpisu vycházejícího z asymetrických šifer.

Nyní popíši, v podstatě jen teoretickou možnost, jak by bylo možné použít symetrickou šifru pro tvorbu elektronického podpisu. Budeme mít zprávu Z, symetrickou šifru f, důvěryhodný server T, který sdílí se všemi uživateli šifru f a s každým z uživatelů A, B, C po řadě, veřejně neznámý klíč k_A, k_B a k_C ...

Postup vytvoření podpisu:

1) Uživatel A zašifruje zprávu Z svým klíčem k_A a výslednou zprávu Z_A odešle serveru T,
2) Server T rozšifruje zprávu Z_A a v případě, že dává smysl, usoudí, že je od uživatele A. Připojí k rozšifrované zprávě svůj souhlas S, a zašifruje jednak Z_A, jednak Z a jednak S klíčem k_B uživatele B. Odešle tyto zašifrované texty uživateli B,
3) Uživatel B rozšifruje příchozí texty svým klíčem k_B.

Postup ověření podpisu další osobou:

1) Uživatel B zašle zprávu Z a zprávu Z_A opět serveru T,
2) Server T ověří, zda zašifrováním zprávy Z klíčem k_A vznikne zpráva Z_A,
3) Pokud ano, pošle uživateli C svůj souhlas.

Schéma podpisu vytvořeného na základě asymetrické šifry

Postup při vytváření dvojice veřejný a soukromý klíč pro RSA je následující:

1. Vygenerujeme náhodně dvě dostatečně velká prvočísla p a q, jejichž přibližná velikost, tím myslíme počet bitů, je předem dána.
2. Vypočítáme[11]

   $$n = pq$$

   $$\varphi(n) = (p-1)(q-1)$$

3. Zvolíme číslo e tak, že platí e $\perp$ φ(n)[12]
4. Vypočteme číslo d pro které platí[13]

Veřejným klíčem je potom dvojice [n,e], soukromým klíčem uživatele je dvojice [n,d], někdy též nazývána tajným klíčem.

V případě použití systému RSA se pro elektronický podpis v souladu s terminologií Zákona o elektronickém podpisu č. 227/2000 používá odlišné názvosloví. Veřejný klíč se nazývá data pro ověření podpisu a soukromý klíč se nazývá data pro vytváření

---

[11] *φ(n)* se nazývá Eulerovou funkcí a platí: hodnota φ(n) je definovaná jako počet přirozených čísel nepřevyšujících n, která jsou s n nesoudělná, tedy formálně φ(n) ≡{k ∈ N | k ≤ n, k⊥n} Pokud musíme určit čísla soudělná s pq. To jsou právě čísla lp pro l ∈ {1, 2, . . . , q} a čísla jq pro j ∈ {1, 2, . . . , p}. Jediné číslo, které je zároveň tvaru lp i jq je pq.

Proto φ(pq) = pq − q − p + 1 = q(p − 1) − (p − 1) = (p − 1)(q − 1)..

[12] Když je NSD(a, b) = 1, řekneme, že a a b jsou navzájem nesoudělná a značíme a $\perp$ b. Kde NSD značí největšího společného dělitele

[13] Nechť m ∈ N. Řekneme, že a, b ∈ Z jsou kongruentní modulo m, jestliže m dělí rozdíl a – b . Značíme a ≡ b mod m.

elektronického podpisu. Číslo n nazýváme modul, číslo e šifrovacím exponentem a číslo d dešifrovacím exponentem. Přitom znalost jednoho z čísel p, q, φ(n) vede k bezprostřednímu nalezení zbývajících tří a rozklad n na p a q vede k prolomení šifrování v RSA.

5.7 Vytvoření podpisu – algoritmus RSA

Pokusme se shrnout hlavní problémy, které nás při naší konstrukci elektronického podpisu nejvíce znepokojují. Problém obtížnosti dešifrování textu s podpisem jsme v podstatě vyřešili v ukázce tvorby podpisu. Zachování integrity dat budeme řešit ,vzhledem ke znalostem, které jsou popsány v kapitole 2.1, čili pomocí hašovacích funkcí. A skutečnost, že vlastně neznáme podpisující osobu budeme muset vyřešit pomocí nějaké instituce, která nám zajistí, že podepisující se osoba je ta daná osoba. Přičemž popisovaná instituce by mohla zároveň vydávat a kontrolovat soukromé a veřejné klíče. Nejdříve si popíšeme jednotlivé typy elektronických podpisů a následně se budeme zabývat problematikou institucí, které kontrolují a vydávají tyto klíče.

6 ZARUČENÝ ELEKTRONICKÝ PODPIS

Definici elektronického podpisu je nutno opět hledat v samotném zákoně č. 227/2000 Sb. o elektronickém podpisu. Zákon definuje v § 2 písm. b) následující:
zaručeným elektronickým podpisem (se rozumí pro účely tohoto zákona) elektronický podpis, který splňuje následující požadavky:

i. je jednoznačně spojen s podepisující osobou,
ii. umožňuje identifikaci podepisující osoby ve vztahu k datové zprávě,
iii. byl vytvořen a připojen k datové zprávě pomocí prostředků, které podepisující osoba může udržet pod svou výhradní kontrolou,
iv. je k datové zprávě, ke které se vztahuje, připojen takovým způsobem, že je možno zjistit jakoukoliv následnou změnu dat

Na druhé straně Směrnice definuje tzv. advanced electronic signature následovně vylepšeným elektronickým podpisem se rozumí elektronický podpis, který splňuje tyto požadavky:

i. je jednoznačně spojen s podepisující osobou;
ii. umožňuje identifikovat podepisující osobu;
iii. je vytvořen s využitím prostředků, které podepisující osoba může mít
iv. plně pod svou kontrolou;
v. je spojen s daty, ke kterým se vztahuje, tak, aby bylo možno zjistit jakoukoliv
vi. následnou změnu těchto dat

Pokud porovnáme tyto dvě definice, dospějeme k závěru, že se jedná v podstatě o jeden pojem.
Pokud si vysvětlíme požadavky na zaručený elektronický podpis můžeme říci následující: První dva požadavky (tj. i. a ii.) nám zajišťují identifikaci podpisujícího a o něm či o ní můžeme tvrdit, že je podepisující (resp. podepsanou) osobou. Další požadavek (iii.) hovoří o tom, že k podpisu stačí soukromý klíč , který má podpisující osoba u sebe. A poslední požadavek (iv.) nám zajišťuje integritu dat. Požadavky na zaručený elektronický podpis se vzhledem k předchozímu odstavci mění. Sice stále se ještě nevyžaduje časové razítko, ani se nevyžaduje použití certifikátu ke zveřejnění dat pro ověření podpisu. Nově se zavádí přesné formáty pro vytváření a přenos elektronických podpisů. To je nutné především z hlediska kompatibility se základním dokumentem v této oblasti Electronic Signature Formats (ETSI TS 101 733 V1.2.2, 2000-12). Nově se zavádí požadavek na důvěryhodnost operačního systému, ve kterém se dokument podepisuje. Nejsou kladeny

žádné specifické požadavky na podpisový prostředek nebo ověřovací prostředek. Bezpečnost těchto prostředků (použití, zabezpečení, ochrana) se zcela nechává na podepisující osobě případně na osobě, která se spoléhá na podpis. Popsaný typ podpisu nemá pro příjemce příliš velkou vypovídací hodnotu a důvěra v něj je minimální. Slouží spíše pro informaci příjemce. Příkladem může být podpis vložený za email, ale i jméno autora uvedené v záhlaví článku. Skutečnost, že i podpis je podpisem ve smyslu zákona číslo 227/2000 Sb., vyplývá z § 3 odst. 1, kde je řečeno, že datová zpráva je podepsána, pokud je opatřena elektronickým podpisem.

6.1 Zaručený elektronický podpis založený na kvalifikovaném certifikátu

K použití tohoto typu podpisu se zavádějí pojmy certifikát, kvalifikovaný certifikát a poskytovatel certifikačních služeb. Poskytovatelé certifikačních služeb se dělí na poskytovatele, kteří vydávají certifikáty dále na poskytovatele, kteří vydávají kvalifikované certifikáty a na akreditované poskytovatele.

Typy certifikátů jsou uvedeny v § 2 písm. g) a h) zákona o elektronickém podpisu:

g) certifikátem (se rozumí pro účely tohoto zákona) datová zpráva, která je vydána poskytovatelem certifikačních služeb, spojuje data pro ověřování podpisů s podepisující osobou a umožňuje ověřit její totožnost

h) kvalifikovaným certifikátem (se rozumí pro účely tohoto zákona) certifikát, který má náležitosti stanovené tímto zákonem a byl vydán poskytovatelem certifikačních služeb, splňujícím podmínky, stanovené tímto zákonem pro poskytovatele certifikačních služeb vydávající kvalifikované certifikáty.

Definice poskytovatelů certifikačních služeb jsou uvedeny v § 2 písm. e) a f) tohoto zákona:

e) poskytovatelem certifikačních služeb (se rozumí pro účely tohoto zákona) subjekt, který vydává certifikáty a vede jejich evidenci, případně poskytuje další služby spojené s elektronickými podpisy,

f) akreditovaným poskytovatelem certifikačních služeb (se rozumí pro účely tohoto zákona) poskytovatel certifikačních služeb, jemuž byla udělena akreditace podle tohoto zákona.

Povinnosti poskytovatele certifikačních služeb, který vydává kvalifikované certifikáty, jsou obsaženy v § 6 zákona o elektronickém podpisu a jsou dále upřesněny v prováděcí vyhlášce č. 366/2001 Sb. Každý poskytovatel certifikačních služeb může požádat Úřad pro ochranu osobních údajů o udělení akreditace pro výkon činnosti akreditovaného poskytovatele certifikačních služeb. Podmínky udělení akreditace jsou uvedeny v § 10 zákona o elektronickém podpisu. Požadavky se na zaručený elektronický podpis založený na kvalifikovaném certifikátu se rozšiřují. Stále se ještě nevyžaduje časové razítko. Zpřísňují se požadavky na přesné formáty pro vytváření a přenos elektronických podpisů. Používání formátů se rozšiřuje o stanovení požadavků na formáty kvalifikovaných certifikátů a o další související formáty. To je upraveno mimo jiné i dokumentem ETSI Qualified Certificates Profile (ETSI TS 101 862). Požadavek na důvěryhodnost operačního systému, ve kterém se datová zpráva podepisuje, je stejný jako u předchozího typu. Ani u tohoto typu podpisu není součástí profilu povinný požadavek na používání bezpečného podpisového nebo ověřovacího prostředku.

Uvedený typ podpisu je základním typem elektronického podpisu, kterým se zabývá zákon o elektronickém podpisu. Podpis má pro příjemce vysokou vypovídací hodnotu a důvěra v něj je vysoká; je také podpořena právními aspekty, které vyplývají z použití takovéhoto podpisu a které plynou ze zákona o elektronickém podpisu. Slouží pro styk příjemce a jiného subjektu, který vlastní kvalifikovaný certifikát. Příjemce podepsanou osobu nemusí osobně znát, data pro ověřování elektronického podpisu získá příjemce z kvalifikovaného certifikátu. Právní jistota v souvislosti s tímto způsobem komunikace vyplývá ze zákona o elektronickém podpisu, nemusí se tedy na rozdíl od předchozího případu uzavírat speciální smlouvy pro právní podporu této komunikace. Důvěra v obsah certifikátu je podmíněna důvěrou v poskytovatele certifikačních služeb, který certifikát vydal. Důvěra vyplývá i ze skutečnosti, že zákon o elektronickém podpisu stanoví poskytovatelům vydávajícím kvalifikované certifikáty celou řadu povinností. Podpis může být použit i k anonymnímu styku, místo jména podepisující osoby může být v kvalifikovaném certifikátu uveden pseudonym, ovšem s označením, že se jedná o pseudonym. V případě právního sporu může být anonymní držitel certifikátu dohledán prostřednictvím údajů, které má k dispozici poskytovatel certifikačních služeb. Uvedený typ podpisu lze použít všude tam, kde se v zákoně o elektronickém podpisu umožňuje použít elektronický podpis. Podpis je přímo vyžadován v § 11, který stanoví způsob komunikace v oblasti veřejné moci. Profil je, zpřísněn z tohoto důvodu nestačí, aby byl kvalifikovaný certifikát vydán

poskytovatelem, který vydává kvalifikované certifikáty, ale poskytovatelem, který byl akreditován Úřadem pro ochranu osobních údajů.

Obecně se považuje takový typ za vhodný pro přímou komunikaci mezi subjekty. Není vhodný k archivaci dat a tam, kde je nutné zpětně prokazovat, kdy přesně byl dokument podepsán.

6.2 Časová razítka

Dalším důležitým pojmem, se kterým se v praxi setkáme, je časové razítko. To je po technické stránce také zaručeným elektronickým podpisem, ale na rozdíl od něj je v něm uveden garantovaný údaj o čase jeho vzniku. A tak časové razítko neslouží ani tak k podpisu, jako k určení přesné doby kdy k podpisu došlo, respektive k prokázání skutečnosti, že to, co je časovým razítkem opatřeno, již v okamžiku vzniku časového razítka existovalo.

Časové razítko tedy stvrzuje, že to, co je jím označeno, vzniklo určitý, libovolně dlouhý úsek před časový okamžikem, uvedeným na časovém razítku. Už se ale neříká nic o tom, zda to bylo dříve o sekundy, minuty či třeba roky.

V praxi se ovšem používají spíše kvalifikovaná časová razítka, která jsou silnější než časová razítka (bez přívlastku), protože je vytváří kvalifikovaný poskytovatel služby časových razítek. A na jeho služby a produkty čili na časová razítka i v nich obsažené časové údaje se skutečně můžeme spolehnout. Zatím se zdá, že časové razítko je stejný produkt jako elektronický podpis. Ano, způsob jejich vzniku je principiálně shodný. Ale liší se ale to, kdo provádí jednotlivé úkony. Postup při vytváření časového razítka je následující. Z dokumentu, který má být opatřen časovým razítkem, se nejprve vytvoří otisk (hash) pevné velikosti. K tomu dochází na straně zdroje, a tedy úplně stejně, jako u elektronického podpisu. V dalším kroku je otisk odeslán poskytovateli časových razítek, a ten jej podepíše s využitím svého soukromého klíče. Poskytovatel časových razítek obvykle k zaslanému otisku nejprve připojí údaj o čase, z výsledku vytvoří nový otisk a teprve ten podepíše Výsledek již představuje samotné časové razítko jako takové – a to je zasláno zpět uživateli, který jej připojí k dokumentu.

Je potřeba zdůraznit ještě jeden důležitý rozdíl mezi podpisy a razítky: elektronický podpis už z principu věci vždy vytváří podepisující osoba . Nejčastěji tedy ten, kdo dokument vytvořil, případně ten kdo ho schvaluje, vydává apod. Obecně ten, kdo s ním vyjadřuje svůj souhlas. Naproti tomu časové razítko může k elektronickému dokumentu připojovat

kdokoli za účelem fixace v čase. Nejedná se tedy o vyjádření souhlasu, a to ani ze strany aktuálního držitele dokumentu, který si nechává časové razítko vystavit a může s obsahem dokumentu třeba i zásadně nesouhlasit, ani ze strany poskytovatele časových razítek. Na druhé straně je časové razítko vlastně shodné s elektronickým podpisem. Protože vzniká principiálně stejným výpočtem. Proto časové razítko zajišťuje například i integritu toho dokumentu, který je razítkem opatřen, přestože to zákon explicitně neříká. Uvedená skutečnost evidentně vyplývá z mechanizmu způsobu tvorby, když zaručuje, že data opatřená časovým razítkem již existovala v okamžiku jeho vzniku. Kdyby došlo k jakékoli změně těchto dat nebo samotného razítka, neboli k porušení jejich integrity, nebyla by to už ta původní data ani to samé razítko. V praxi to znamená, že i při vyhodnocování platnosti časového razítka se zjišťuje, zda je či není porušena integrita původního dokumentu . Z ryze praktického pohledu pak má vytváření časových razítek několik dalších zajímavých aspektů. Například ten, že probíhá v reálném čase a vyžaduje on-line přístup. Ten, kdo chce nějaký svůj dokument opatřit časovým razítkem tím míníme, kvalifikovaným časovým razítkem od kvalifikovaného poskytovatele, musí být schopen zaslat poskytovateli časových razítek příslušný otisk (hash). Poskytovatel na to musí reagovat v reálném čase , musí vygenerovat časové razítko okamžitě a také ho co nejrychleji vrátit zpět žadateli. Přitom jak uživatel, který o vytvoření časového razítka žádá, tak i poskytovatel časových razítek, musí být vzájemně propojeni takovým způsobem a přes taková rozhraní, aby si jejich systémy vzájemně rozuměly a dokázaly spolupracovat. Naproti tomu při podepisování, při vytváření elektronických podpisů, není nutné být on-line ani se propojit s dalšími subjekty, a není zde ani požadavek na fungování v reálném čase. Už vzhledem k tomu, že údaj o čase vzniku podpisu není fakticky relevantní, kvůli tomu že není dostatečně důvěryhodný.

6.3 Shrnutí kapitoly elektronický podpis

V této části jsme si ukázali, jakým způsobem můžeme vytvořit elektronický podpis. Známe základní principy pro tvorbu elektronických podpisů. Umíme vytvořit i vlastní modifikaci elektronického podpisu, takovou která splňuje všechny hlavní principy pro vznik elektronického podpisu. A bohužel také víme, že vše výše popsané nestojí na pevných základech. Elektronický podpis se svými hašovacími funkcemi, se svými soukromými a veřejnými klíči musí být nutně zpevněn pomocí autorizovaných institucí, které nám zajistí hlavně tyto skutečnosti. Budeme moci ověřit platnost certifikátů a bude

nám uvedená instituce garantovat, že podepsaná osoba je existující osoba. Čili nám ji bude moci identifikovat a v neposlední řadě nám tyto organizace budou poskytovat určité metodické vedení při správě elektronických podpisů. Jakým způsobem jsou tyto instituce organizovány to budu popisovat v následujících kapitolách.

7 CERTIFIKÁTY A CERTIFIKAČNÍ AUTORITY

Základní stavební kámen infrastruktury zajišťující důvěryhodnost vztahu jedince a jeho klíče, ať již šifrovacího klíče nebo klíče pro ověření podpisu je certifikační autorita, CA[14]. Certifikační autorita je důvěryhodná třetí strana, jejíž důvěryhodnost umožňuje důvěřovat vztahu páru klíčů, vzniklých jako produkt asymetrické kryptografie, a konkrétní identifikovatelné osoby. Typicky musí být certifikační autorita schopná důvěryhodně identifikovat osoby držící ve vlastnictví příslušný pár klíčů , může hodnoty takových klíčů i generovat, musí být schopná veřejně deklarovat platnost vztahu mezi osobou a párem klíčů držených touto osobou a musí být schopna odvolávat platnost vztahu mezi osobou a párem klíčů držených touto osobou. Pro mnohé aplikace musí CA umožnit, aby si párové hodnoty klíčů zúčastněné osoby generovaly samy. CA vydávají elektronické, tedy jimi podepsané dokumenty, které platnost vztahu osoby a jejího veřejného klíče důvěryhodně potvrzují digitálními certifikáty. Potvrzení vlastnictví veřejného klíče danou osobou současně potvrzuje, že osoba vlastní odpovídající soukromý klíč. Certifikáty jsou vydávány obvykle ve třech úrovních záruky za důvěryhodnost: nízká, střední a vysoká záruka. Vyšší úroveň poskytované záruky vyjadřuje skutečnost, že CA věnuje větší úsilí při potvrzování identity osob a při zajišťování své bezpečnosti. Propojením dvou a více CA s možností se vzájemně certifikovat, vzniká certifikační infrastruktura veřejných klíčů. Jejím hlavním přínosem je, že umožňuje provozovat komunikační procesy i mezi stranami, které se předem neznají, pokud CA zúčastněných stran lze nějakým způsobem začlenit do společné infrastruktury, často nazývané infrastruktura veřejných klíčů. Pravidla, která deklarují použitelnost certifikátů vydávaných danou CA v rámci jisté komunity se společnými shodnými požadavky na bezpečnost, definuji certifikační politiku (CP[15]) certifikační agentury. Daná CA může uplatňovat více CP, a to jak pro oblast šifrování, tak i pro oblast podpisování zpráv. Certifikační politika je základní stavební kámen pro budování důvěryhodnosti certifikátů veřejných klíčů. Je technologickou základnou pro vzájemnou certifikaci více CA vytvářejících společnou strukturu. CP vymezuje důkladnost prověřování autenticity žadatele o vydání certifikátu. Popis toho, jak

[14] CA – certifikační autorita

[15] CP – certifikační politika

jsou pravidla dané CP implementována, vyjadřuje další dokument z dokumentové základny CA. Certifikační prováděcí směrnice (CPS[16]). CA může danou CPS podporovat více certifikačních politik a naopak certifikační autority s různými prováděcími směrnicemi mohou podporovat identické CP. Certifikační politika stanovuje záruku, se kterou lze důvěřovat certifikátu, CPS dané CA stanovuje, jak certifikační autorita dané záruky dosahuje [2].

7.1 CERTIFIKACE VEŘEJNÉHO KLÍČE

Problematiku která se zabývá správou, distribucí a uchováváním klíčů je možné shrnout pod pojem certifikace veřejného klíče, certifikáty vydávají certifikační autority. Pokud se budeme dívat na celou bezpečnostní architekturu, která se zabývá vším co souvisí se soukromým a veřejným klíčem, budeme tuto oblast nazývat infrastrukturou veřejných klíčů často zkracovanou na PKI. Možná by bylo dobré na tomto místě si připomenout o jakou bezpečnostní strukturu se jedná. Proč je nutné dále se zabývat veřejnými a soukromými klíči? Vždyť už známe princip tvorby. To je pravda, ale jádro problému není ve způsobu tvorby klíčů. Hlavní problém spočívá v tom, jak zabezpečit klíče před zneužitím a jak vybudovat funkční strukturu.

7.1.1 Certifikační autorita

Pokud certifikační autorita vydala certifikát konkrétní osobě a svým podpisem potvrdila, že daný certifikát skutečně patří této osobě, lze věřit, že veřejný klíč obsažený v tomto certifikátu skutečně náleží této osobě, protože věříme certifikační autoritě. V závislosti na požadovaném stupni bezpečnosti je možné použít různé certifikáty. Pro vydání certifikátů s nižším stupněm bezpečnosti stačí, pokud se uživatel identifikuje svým e-mailem, zatímco u některých je vyžadována osobní přítomnost uživatele, který svým podpisem potvrdí dokumenty v papírové podobě přímo v kanceláři certifikační autority. Ne všechny certifikační autority musí být skutečně důvěryhodné. V praxi je možné se setkat s nepravou certifikační autoritou. Pokud máme věřit certifikační autoritě, měla by být všeobecně známá a prověřená. Ve světě digitální bezpečnosti tyto autority povinně používají speciální hardware, který garantuje, že nemůže dojít k úniku důležitých informací například

[16] CPS – certifikační prováděcí směrnice

privátních klíčů. Mezi nejznámější a prověřené autority patří firmyVeriSign Inc, Thawte Consulting, GlobalSign NV/SA, Baltimore Technologies, TC Trust Center AG, Entrust Inc a v České republice to jsou První certifikační autorita, a.s., Česká pošta, s.p. a eIdentity a.s. Každá certifikační autorita vlastní certifikát a k němu příslušný soukromý klíč, pomocí kterého podepisuje certifikáty svých zákazníků. Certifikační autorita může být různých úrovní, například nejvyšší úrovně (top-level CA, root CA). CA nejvyšší úrovně vydávají sami sobě certifikát na začátku své působnosti a podepíší jej tím samým certifikátem. Vznikne tak kořenový certifikát – root certificate. Kořenové certifikáty všech CA jsou veřejně dostupné na jejich webových stránkách a mohou být použity pro ověření dalších certifikátů. CA nižší úrovně závisí na CA vyšší úrovně, která jim vystaví certifikát, který jim dovoluje vydávat a podepisovat certifikáty pro jejich zákazníky. Je technicky možné použít každý certifikát k podpisu dalších certifikátů, ale prakticky je možnost podepsat certifikáty velice omezená. Každý certifikát obsahuje informaci, která není běžně dosažitelná, mohou-li být certifikáty použity k podpisu dalších certifikátů. CA vydávají certifikáty, které nemohou být použity k podpisu dalších certifikátů. Certifikáty, které lze použít k podpisu dalších jsou vydávány pouze jiným CA s vysokými bezpečnostními opatřeními. Jakýkoli vydaný certifikát může být podepsán certifikátem CA nebo může být podepsán sám sebou. Tyto certifikáty jsou nazývány self-signed. Vlastně každý kořenový certifikát je self-signed. Tyto certifikáty nemohou být použity k identifikaci uživatele, protože takovýto certifikát si může vytvořit kdokoliv. Přestože self-signed certifikáty nemohou být použity k identifikaci jeho uživatele, i tak pro ně najdeme uplatnění. Například ve firemním prostředí, kde je možné bezpečně distribuovat fyzicky certifikáty mezi jednotlivými zaměstnanci a vnitřními systémy firmy. self-signed certifikáty mohou úspěšně nahradit certifikáty vydávané certifikační autoritou.

Není zde nezbytně nutné, aby určitá CA potvrdila, že daný veřejný klíč patří dané osobě, protože to je garantováno systémem vydávání a distribucí certifikátu. V případě, že nový zaměstnanec nastoupí do firmy, administrátor mu vytvoří nový certifikát a předá jej na USB disku nebo jinou formou. Poté administrátor zanese bezpečně jeho certifikát do všech vnitřních systémů a garantuje, že zde bude správný certifikát pro určitého zaměstnance. Tyto self-signed certifikáty mohou být nahrazeny vlastní firemní CA. Pro tento účel musí administrátor firmy vydat self-signed certifikát a všechny další vydané certifikáty jsou podepsány tímto certifikátem. Certifikát se stane kořenovým certifikátem firemní CA a firma samotná je CA nejvyšší úrovně, ale pouze v rámci firmy.

7.1.2 Reálná možnost zneužití

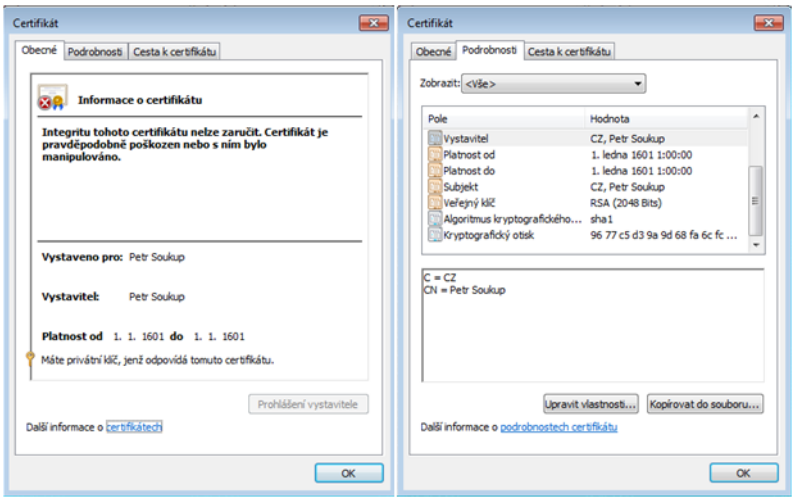
Celý systém, vycházející z infrastruktury veřejného klíče je na jedné straně docela praktický. Stačí nám jeden úkon a to vyjádřit důvěru kořenové autoritě, a naše důvěra se tím automaticky přenáší na celý podstrom. Jenže problematika má i druhou stranu mince. Pokud se někomu podaří zkompromitovat právě kořenovou autoritu, případně podřízenou certifikační autoritu, dochází k tomu, že je napadena celá struktura.

Nyní si představme následující scénář, někdo napadne autoritu a podaří se mu vystavit si od ní podvodné SSL[17] certifikáty. Takové, že serveru Y umožňují vydávat se za server X. Čili útočník si nechá vystavit podvodný SSL certifikát pro doménové jméno, které představuje X např. xxx.cz . Odpovídající soukromý klíč ale svěří svému serveru, který představuje server Y. Ten se díky tomu může vydávat například za server mail.xxx.cz. Kdo se bude zabývat serverem Y, když ve skutečnosti chce pracovat se serverem X? K tomu je nutná ještě další část scénáře. Musí totiž dojít k podvodnému přesměrování, když uživatel zadá například mail.xxx.cz, nesmí se dostat tam, kam by se správně měl dostat, na server X, ale musí být přesměrován na server Y. Teprve v tomto případě se server Y dostane k tomu, aby se prokázal oním podvodným certifikátem, a vydával se za server mail.xxx.cz. Uživatel to ale neví, protože jeho webovému prohlížeči se podvodný server Y jeví jako pravý tj. jako server X. Zbývající část je již zřejmá server Y, ochotně přepošle veškerou konverzaci skutečnému uzlu X. Přitom si ale odposlechne to, co ho zajímá – například přihlašovací údaje k uživatelským účtům, poštovním schránkám apod. [9].

7.2 CERTIFIKÁTY

Každý certifikát má svůj předmět to je zpravidla majitel certifikátu nebo doménový název serveru. Účel a možnost použití, svého vystavitele , období platnosti a další parametry. Pro základní rozdělení typů certifikátů nás budou zajímat hlavně první tři atributy.

[17] SSL – Secure Sockets Layer, vrstva poskytující zabezpečení komunikace



Obr. 1. Ukázka certifikátu

Předmětem, vlastníkem certifikátu může být buď osoba nebo počítač nebo serverový certifikát. Pokud přistupujeme k zabezpečenému webovému serveru, tím myslím elektronický obchod, elektronická podatelna úřadu apod., pak je vlastníkem daný server identifikovaný svým doménovým názvem a většinou současně i organizací, která daný webový server provozuje.

Osobní certifikáty podle typu a úrovně ověření identifikují vlastníka podle emailové adresy, případně i dalších údajů (jméno a příjmení, název zaměstnavatele atd.). Asi nejvíce detailů obsahují kvalifikované certifikáty určené zejména pro komunikaci s veřejnou správou.

U certifikátů se uvádí účely použití a atributy. Nejčastěji se můžete setkat s následujícími základními účely:

- přístupový certifikát - lze jej použít k přístupu k serveru, který vyžaduje ověření vlastníka certifikátu pomocí soukromého klíče
- podpisový certifikát - slouží k digitálnímu podepsání dat, která tak chrání proti jejich neoprávněnému pozměnění a zároveň potvrzuje, že je podepsal majitel soukromého klíče, ke kterému byl vystaven podpisový certifikát.

7.2.1 Typy certifikátů

Vystavitele certifikátů certifikační autority můžeme zjednodušeně rozdělit do dvou základních skupin: na důvěryhodné a nedůvěryhodné. Poskytovatelé operačních systémů, vývojáři webových prohlížečů a od nedávné doby i výrobci mobilních telefonů se svými

produkty dodávají i seznam certifikačních autorit, kterým operační systém, prohlížeč nebo jiný software v základním nastavení důvěřuje. Výchozí důvěra je odvozena z certifikační politiky, tj. pravidel a organizačních podmínek, na základě kterých autorita ověřuje žadatele a následně jim vydává certifikáty.
Seznam důvěryhodných autorit si může následně uživatel upravovat dle vlastních potřeb. Odstraňovat ty, kterým nedůvěřuje, nebo naopak přidávat vlastní, kterým důvěřuje – např. vnitrofiremní certifikační autorita. Certifikáty certifikačních autorit jsou nazývány jako kořenové certifikáty (root certificates) a většinou je najdete na webových stránkách příslušné autority.

Mezi důvěryhodné certifikáty patří následující:

1. komerční certifikát
2. komerční certifikát - instantní certifikát
3. komerční certifikát s rozšířeným ověřením (EV – extended validation)

Mezi nedůvěryhodné certifikáty patří následující:

1. kvalifikovaný certifikát
2. self-signed certifikát
3. certifikát, který vydala nedůvěryhodná certifikační autorita

Komerčním certifikátem se nazývá takový, který vystaví certifikační autorita za úplatu a ověří žadatele standardní procedurou, kterou stanovuje ve své certifikační politice – v případě osobních certifikátů identitu žadatele, v případě serverových certifikátů pak kromě identity žadatele i vlastnictví domény, pro kterou je certifikát vystavován.
Některé certifikační autority nabízejí instantní certifikáty, které jsou vystaveny během několika minut od podání žádosti. Ověřovací rutina je pak výrazně zjednodušena – identita se ověřuje pouze emailem zaslaným na adresu žadatele (osobní certifikáty) nebo na adresu uvedenou v registraci domény (serverové certifikáty). V certifikátu samotném pak není uvedeno jméno nebo název žadatele, ale text je nahrazen pouze informací jakým způsobem a na základě jak byl certifikát ověřen.
Některé certifikační autority nabízejí rozšířené ověření. Takový certifikát používají z důvodu jeho vysoké ceny většinou banky, pojišťovny nebo podobné větší instituce. Kromě jistoty důkladného ověření žadatele se v nových prohlížečích ihned při otevření zabezpečeného spojení v adresní řádce zobrazí jméno žadatele a jméno vystavitele.

Zákon č. 227/2000 Sb., o elektronickém podpisu stanovuje v ČR pravidla pro elektronickou komunikaci především mezi fyzickými i právnickými osobami a veřejnou správou. Kvalifikované certifikáty obsahují rozšířené identifikační údaje, v případě firemních nebo zaměstnaneckých certifikátů je jejich součástí identifikační číslo organizace. Na základě uvedeného zákona vystavuje tyto certifikáty kvalifikovaná certifikační autorita , přičemž certifikační politiku stanovuje zákon. Na stránkách Ministerstva informatiky ČR můžete najít seznam akreditovaných poskytovatelů. Kvalifikované certifikáty ve výchozím nastavení nepatří mezi důvěryhodné certifikační autority. Při přístupu na stránky zabezpečené certifikátem se proto objevuje hlášení, že certifikát vydala společnost, která není důvěryhodná. Aby k tomuto nedocházelo, musíme v počítači nainstalovat kořenový certifikát certifikační autority do seznamu důvěryhodných autorit.

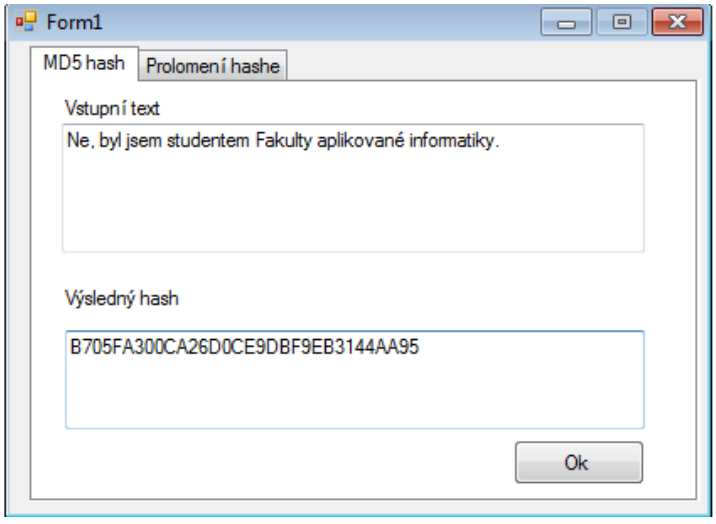
II. PRAKTICKÁ ČÁST

8 VYUŽITÍ V PRAXI

V praktické části si ukážeme nejdříve pojem hash a uvidíme základní způsob využití. Také si ukážeme rozdíl mezi hašovací funkcí a šifrováním. V druhé kapitole vytvoříme elektronický podpis pomocí algoritmu RSA, přičemž podpis budeme vytvářet na základě vlastních kryptografických standardů. Podpis budeme vytvářet na základě elementárních matematických operací tak, aby byl zcela jasně ukázán princip tvorby podpisu. V poslední kapitole budeme vytvářet certifikační autoritu a to pomocí programu makecert, který je součástí Windows SDK[18].

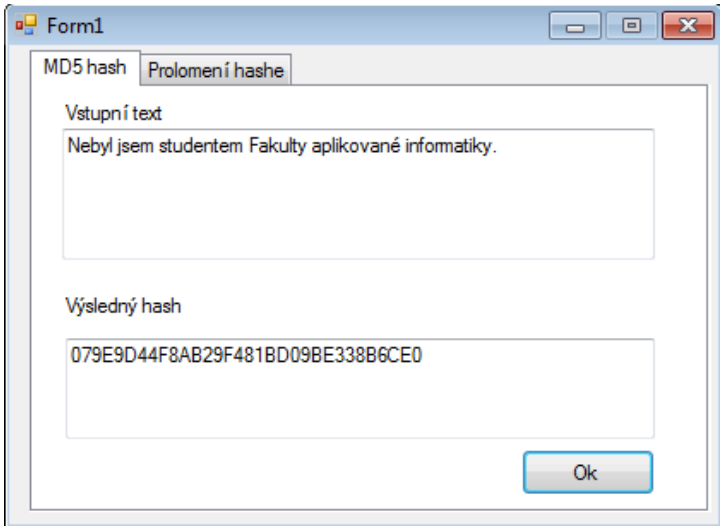
8.1 Hašovací funkce

Pro porovnání si můžeme předvést na jednoduchém příkladu. Budeme porovnávat rozdíl při malé změně ve vstupním řetězci je vidět na následujících obrázcích. Text byl úmyslně zvolený tak, aby obsahoval jen nepatrnou změnu (jen byla odstraněna čárka a mezera) a zároveň, aby došlo k významné změně ve smyslu sdělení. Jak je vidět v části "Výsledný haš" je zcela jasně patrný rozdíl mezi h(X) a h(Y).



Obr. 2. Hash h(X)

[18]Windows SDK - Software Development Kit, sada vývojových a programových rozhraní k tvorbě aplikací



Obr. 3. Hash h(Y)

X = Ne, byl jsem studentem Fakulty aplikované informatiky.

h(X) = B705FA300CA26D0CE9DBF9EB3144AA95

Y = Nebyl jsem studentem Fakulty aplikované informatiky.

h(Y) = 079EE9D44F8AB29F481BDO9BE338B6CE0

V této části zašifrujeme pomocí šifry PlayFair jednoduchý text a porovnáme ho s hašovacím řetězcem, který jsme vytvořili v první části. Takže budeme šifrovat text:

Z = Byl jsem studentem Fakulty aplikované informatiky.

S	O	K	U	P
A	B	C	D	E
F	G	H	I	L
M	N	Q	R	T
V	W	X	Y	Z

Obr. 4. Šifrovací tabulka

Po zašifrování výše uvedeného textu a jeho úpravě do bloku po pěti písmenech, získáváme tuto šifru :

DWIHP AVARP CDMRA TMFOK TZVDE THUSW BMDLM GUNVF

RLUXX (1)

Pokud vytvoříme haš stejného textu, získáme v případě algoritmu MD5:

$h(Z_{hex})$ = 0D849AA29AA7BCC2403A336E4326D3A3 (2)

$h(Z_{dek})$ = 41521556263161740000 (3)

Porovnáním výsledků (1) a (2) resp. (3), kde vidíme, že hašovací algoritmus nám vrátil opět stejně velkou skupinu znaků a nikdy nemůžeme dostat jinak dlouhý výsledek. A je zřejmé, že zašifrováním získáváme různě dlouhý text, jehož délka závisí na délce původního textu. Relativně snadno, pokud známe heslo, můžeme text rozšifrovat. Zde tedy velice názorně vidíme rozdíl mezi hašovací funkcí a šifrováním.

Nyní víme, co to je hašovací funkce, ale my si můžeme i naznačit jak se dá popisovaná funkce vytvořit. Prostě si vymyslíme hašovací algoritmus, protože příklad bude velice zjednodušený a nebude se jednat o hašovací funkci, pojmenuju ji jako skoro hašovací funkce a budu ji zkracovat na SAŠ funkci.

Nejdříve musíme mít zprávu Z_a.

Nechť Z_a = {Dnes je utery} (4)

Zprávu Z_a převedeme na číselný zápis a to velice jednoduše. Využijeme internetu a ASCII kódu, získáváme číslo

Z_a = 681101011153210610132117161011141211310

dále si zvolíme libovolné, velké prvočíslo p. Prvočíslo můžeme jednak vypočítat pomocí některé z mnoha metod, nebo lehce nalézt na internetu a následně ověřit zda se jedná doopravdy o prvočíslo pomocí programu WolframAlpha. Jedním ze způsobů výpočtu prvočísla jsou i Carolova prvočísla, která získáme pomocí následujícího vztahu

$4^n - 2^{n+1} - 1$. Nyní můžeme přistoupit ke konstrukci naší SAŠ funkce $S(Z_a)$, bude platit

$S(Z) = Z_a \ mod \ p$

Pokud zvolíme p = 16127 a ověříme prvočíselnost, tak dostaneme v našem případě SAŠ

$$S(Z_a) = 5189$$

Co nastane, když zaměníme vstupní řetězec na

Z_b = {Dnes neni utery} (5)

Postupujeme obdobně jako u předcházejícího řetězce, ale získáváme jiný otisk zprávy Z_b.

$$S(Z_b) = Z_b \ mod \ p = 7065$$

Na uvedeném příkladu vidíme, že naše SAŠ funkce také vytváří určitý otisk zadaného textu, dokáže také rozpoznat malé změny ve vstupním textu. Na druhé straně vidíme i nedostatky takto vytvořené funkce. První problém je v malém množství výstupních hodnot, kterých je *p-1*, pro naše *p* je to 16126 možností. Tento nedostatek by bylo možné lehce vyřešit tak, že by bylo zvětšeno prvočíslo *p*, což by možné bylo. Větší problém vidím v odolnosti vůči kolizím. Každá změna v původním řetězci taková, že součet všech změněných hodnot vyjádřených pomocí ASCII kódu bude roven *k p*, způsobí kolizi naší SAŠ funkce. Možnost jak vyřešit odolnost vůči kolizím na takto postavené funkci nevidím. Asi by bylo možné rozdělit vstupní řetězec na n-tice a z nich vypočítávat SAŠ funkci, ty následně mezi sebou násobit. Ale vždy se dostaneme ke konečnému počtu možností.
I přes uvedené nedostatky jsme si ukázali, jak je možné vytvářet skoro hašovací funkci. Naši SAŠ funkci by asi nebylo možné použít při komunikaci s bankou a provádět pomocí ní bankovní transakce. Na druhé straně si umím představit její využití například při firemní komunikaci, kdy mohu ověřovat, zda odeslaný email byl doručen v plném a nepozměněném stavu.

8.2 Podpis vytvořený pomocí algoritmu RSA

V této části vytvoříme elektronický podpis pomocí algoritmu RSA. Jak bylo výše uvedeno potřebujeme vytvořit soukromý a veřejný klíč. Dále musíme předem daným způsobem převést text, který budeme podepisovat, nějaké číselné soustavy. Text tedy zakódujeme.

8.2.1 Zakódování zprávy

Využijeme modifikovanou tabulku z PlayFair šifry , kdy každé písmeno nahradíme uspořádanou dvojicí číslic a to tak, že v záhlaví příslušného řádku nalezneme první číslici a podle sloupce určíme druhou číslici. Zároveň musíme vytvořit určitá pravidla formátování. V praxi se používají poměrně složitá pravidla, která se souhrnně nazývají standarty PKCS[19] a značí se PKCS #1 až PKCS #15 a přesně definují například mechanismy šifrování, standart syntaxe apod.

My si však zavedeme následující jednoduchá pravidla. Budu je nazývat standardem formátování SFAI #01.X[20] :

1. Délka modulu bude v dekadickém zápisu rovna $s = 3$
2. Pokud modul nemá $s = 3$, doplníme potřebný počet nul z pravé strany
3. Výsledný řetězec rozdělíme do skupin po 4 číslicích, chybějící číslice doplňujeme tak, že přidáme potřebný počet nul z levé strany.

V následujících řádcích vytvoříme elektronický podpis. Samozřejmě se nejedná o skutečný podpis, který by vytvořil algoritmus RSA, jedná se pouze o ukázkový podpis. Hlavním omezením je ta skutečnost, že nevolím dostatečně velká přirozená čísla p a q , důvodem je jednak větší názornost, větší jednoduchost a v neposlední řadě i větší praktičnost. Protože si neumím představit, jak zde vypisuji čísla o velikosti stovek řádů. Ale k popsání základních principů tvorby elektronického podpisu pomocí algoritmu RSA je následující příklad plně dostačující. Je dán textový řetězec M, na kterém budeme vytvářet elektronický podpis.

M = Byl jsem studentem FAI. Petr

[19] PKCS - Public-key Cryptography Standards, standardy kryptografie veřejných klíčů

[20] SFAI – Standards FAI, standardy FAI

	1	2	3	4	5
9	**S**	**O**	**K**	**U**	**P**
8	**A**	**B**	**C**	**D**	**E**
7	**F**	**G**	**H**	**I**	**L**
6	**M**	**N**	**Q**	**R**	**T**
5	**V**	**W**	**X**	**Y**	**Z**

m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9	m_{10}	m_{11}	m_{12}	m_{13}	m_{14}
825	475	749	185	619	165	946	265	856	171	817	495	856	564

Obr. 5. Zakódování textu

Pravidlo SFAI[21] #01.2 použít v tomto případě nemusíme.

Máme původní zprávu převedenou podle jistých, pro nás pevně daných pravidel. Pravděpodobně, čtenáře napadne, že by bylo možné převádět zprávu Z například jen pomocí ASCII tabulky. Ano, to by možné bylo. Ale myslím si, že je zajímavé využít výše uvedenou tabulku z šifry PlayFair a ukázat, že převod zprávy na dekadická čísla je možný i jinak než triviálně.

8.2.2 Nutné matematické základy

Zde by jsme si měli naznačit teoretická východiska, na jejichž základě budeme postupovat. Nejdříve si popíšeme Eukleidův algoritmus.

Největšího společného dělitele bychom zřejmě mohli najít tak, že bychom postupně kontrolovali všechna čísla menší než x i y a hledali první takové číslo, které beze zbytku obě čísla dělí. Existuje zde však podstatně efektivnější algoritmus, který byl publikován zhruba 300 let před naším letopočtem řeckým učencem Euklidem. Vychází z pozorování, že jestliže číslo r je zbytek po dělení čísla x číslem y, pak největší společný dělitel čísel x a

[21] KFAI – Standard Fakulty Aplikované Informatiky

y a čísel y a r je stejný. Výpočet největšího společného dělitele čísel x a y tak můžeme převést na výpočet největšího společného dělitele čísel y a r. Tím ovšem problém redukujeme na jednodušší, neboť číslo r je jistě ostře menší než číslo y. Pokud budeme algoritmus opakovat, musíme dříve či později dojít k situaci, kdy r bude rovno nule. Největší společný dělitel x a nuly je pak číslo x. Řečeno úplně jednoduše: Výpočet největšího společného dělitele tedy spočívá v tom, že neustále dělíme dělitele zbytkem po předchozím dělení. Ve chvíli, kdy nám vyjde zbytek nulový, podíváme se na zbytek v dělení předchozím, a to je právě náš hledaný největší společný dělitel.

Ukažme si Eukleidův algoritmus na následujícím příkladu výpočtu největšího společného dělitele

(NSD) čísel 40 a 6:

NSD(40, 6) = NSD(6, 4) = NDS(4, 2) = NSD(2, 0) = 2

Na tomto místě si nadefinujeme další pojem. Jestliže dvě celá čísla a, b mají při dělení přirozeným číslem m týž zbytek r, kde $0 \leq r < m$, nazývají se a, b kongruentní modulo m (též kongruentní podle modulu m), což zapisujeme takto:

$$a \equiv b \ (mod \ m)$$

Dalším teoretickým předpokladem je Malá Fermatova věta:

Nechť p je prvočíslo. Pak pro všechna přirozená čísla a platí

$$a^p \equiv a \ (mod \ p)$$

respektive

$$a^{\varphi(n)} \equiv 1 \big(mod(n)\big) \quad právě\ když \quad NSD(a,n) = 1$$ [7].

Dalším pojmem, který je nutné popsat je speciální typ polynomiální rovnice, a to rovnice diofantická lineární. Rovnici můžeme definovat následovně: Nechť $a_1, a_2, a_3, \ldots, a_n$, b jsou celá čísla, nechť $x_1, x_2, x_3, \ldots, x_n$ potom rovnici

$$a_1x_1 + a_2x_2 + a_3x_3 + \ldots + = b \qquad (6)$$

budeme nazývat lineární diofantickou rovnicí [8].

K řešení těchto rovnic je možné užít kongruencí, přičemž výše uvedená rovnice má celočíselné řešení, právě když číslo b je dělitelné největším společným dělitelem čísel $a_1, a_2, a_3, \ldots, a_n$.

Poslední pojmem je Bezoutova věta, která říká: Jsou-li a a b celá čísla, pak existují celá čísla x, y, že platí

$$xa + yb = NSD(a, b) \qquad (7)$$

Rovnici (5) budeme nazývat Bezounovou rovnicí a říká nám, že NSD dvou čísel můžeme zapsat jako lineární kombinaci těchto čísel a koeficientů *x* a *y* [8].

Nejlépe si řešení ukážeme na příkladu:

Řešte rovnici 5x + 7y = 8.

Libovolné řešení této rovnice musí splňovat kongruenci

$5x + 7y \equiv 8\ (mod\ 5)$

čili

$2y \equiv -2\ (mod\ 5)$ po úpravě $y \equiv -1\ (mod\ 5)$ z čehož plyne

$y = -1 + 5t$ kde $t \in Z$

po dosazení do rovnice dostáváme

$5x + 7(-1 + 5t) = 8$

odkud vypočítáme $x = 3 - 7t.$ Řešením naší rovnice je tedy

$x = 3 - 7t$

$y = -1 + 5t$

kde t je libovolné celé číslo.

A nyní již prakticky

8.2.3 Výpočet klíčů

V následujících odrážkách jsou vypočítány jednotlivé kroky algoritmu.

1. zvolíme prvočísla

$p = 47 \ \wedge \ q = 67$

2. vypočteme *n* a *φ(n)*

$n = 47 \cdot 67 = 3149$

$\varphi(n) = 46 \cdot 66 = 3036$

3. určíme šifrovací exponent *e*

$e = 19$

přičemž musí platit e $\perp$ φ(n) , o čemž se přesvědčíme prostým dělením, zbytek je 15.

4. vypočítáme d

$$e \cdot d \equiv 1 \, mod \, \varphi(n)$$

výpočet uskutečníme pomocí Eukleidova algoritmu.
Eukleidův algoritmus

$$3036 = 159 \cdot 19 + 15$$
$$19 = 1 \cdot 15 + 4$$
$$15 = 3 \cdot 4 + 3$$
$$4 = 1 \cdot 3 + 1$$

Z čehož plyne NSD(3036,15) = 1, takže čísla jsou nesoudělná. Ale to jsme si potvrdili co již víme z bodu 3. Dokonce jsme to zjistili prostým dělením, k čemu nám je Eukleidův algoritmus? Použijeme rozklad a vhodně ho upravíme.

$$1 = 4 - 3 = 4 - (15 - 3 \cdot 4) = 19 - 15 - \big(15 - 3(19 - 15)\big) =$$

po úpravě dostáváme

$$= 4 \cdot 19 - 5 \cdot 3036 + 5 \cdot 159 \cdot 19 = -5 \cdot 3036 + 799 \cdot 19$$

nyní můžeme psát

$$19 \cdot 799 = 15181 \implies 19 \cdot 799 \equiv 1 \, mod \, 3036 \implies d = 799$$

Získali jsme tedy

veřejný klíč $[n,e] = [3149, 19]$ a

soukromý klíč $[n,d] = [3149, 799]$.

Dále budeme pokračovat ve vytváření elektronického podpisu, protože máme vše co potřebujeme.
Postup si ukážeme podrobněji na prvním bloku m_1 , zdrojová data ve blocích značíme m_x a cílová data budeme nazývat c_y. Musí platit

$$c_1 \equiv m_1^e \, mod \, n$$

$c_1 \equiv 825^{19}\ mod\ 3149$

$c_1 = 395$

Výpočet můžeme uskutečnit buď pomocí matematických aplikací například MatLab nebo pomocí webového rozhraní WolframAlpha. Popřípadě můžeme použít i tabulky programu Excel, který však bohužel neumí počítat větší čísla a nepříjemně zaokrouhluje. Nicméně problém s velkými čísly se dá vyřešit pomocí úvahy, kterou vyjádřím následujícím vztahem

$mod(n; x^{k+q}) = mod(n; (mod(n;\ x^k) \cdot mod(n;\ x^q))$

Ať zvolíme jakýkoliv způsob, získáváme řetězec C = c_1 c_2 c_3 ..., který nazveme podpisem . V naší ukázce získáváme:

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{11}	c_{13}	c_{14}
395	668	2520	2661	2565	63	2733	2608	2286	305	2194	382	2286	2162

Doplníme podle pravidla KFAI#01.3

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{11}	c_{13}	c_{14}
0395	0668	2520	2661	2565	0063	2733	2608	2286	0305	2194	0382	2286	2162

Obr. 6. Podepsaná zpráva

Můžeme tedy psát, že zpráva M, která je podepsaná jménem Petr má finální tvar

C = 039506682520266125650063273326082286228622862162

Zprávu dešifrujeme opačným postupem. Za použití soukromého klíče

$m_1 \equiv c_1^d\ mod\ n$

$m_1 = 720^{799} mod\ 3149$

A uživatel, který zná soukromý klíč může zprávu dešifrovat. Co jsme vlastně získali ? Dokážeme zašifrovat zprávu, kterou bez znalosti soukromého klíče je nemožné dešifrovat. Víme, že je za zprávou připojené jméno. A to je vlastně vše. Nevíme. zda Petr zprávu M

podepsal. Nevíme, zda byla podepsanou osobou podepsaná či zda nebyla někým pozměněna. Dokonce ani nevíme, zda nějaký Petr vůbec existuje. Z výše uvedeného je možné usoudit, že budeme potřebovat silnější nástroje než jen prosté užití dvojice veřejný a soukromý klíč.

8.3 Vytvoření certifikátů

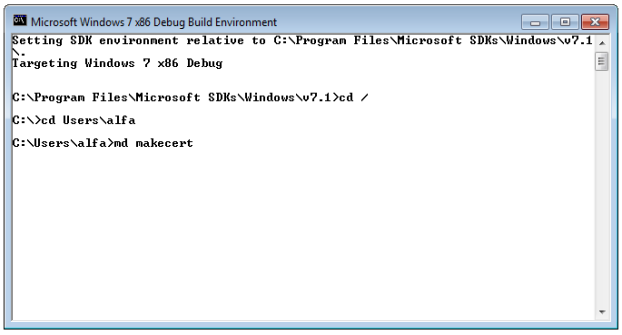
Ukážeme si jak je možné vytvořit certifikáty. Abychom mohli vyzkoušet jejich principy, můžeme si vytvořit testovací certifikáty. Tyto testovací certifikáty mají tu výhodu, že si nemusíme kupovat některý z komerčních a velmi drahých produktů. Pokud potřebujeme vytvořit testovací certifikáty nebo jejich logickou sestavu, je možné použít Open SSL. Open SSL implementuje protokoly SSL (Secure Sockets Layer) a TLS (Transport Layer Security). Protokol SSL nebo jeho modernější mírně pozměněná verze TLS jsou kryptografické protokoly, poskytující možnosti zabezpečené komunikace na Internetu pro služby jako WWW, elektronická pošta, internetový fax a další datové přenosy. Mezi protokoly SSL 3.0 a TLS 1.0 jsou drobné rozdíly, ale v zásadě jsou stejné. Další vhodný nástroj je součástí Windows SDK[22] a jmenuje se makecert.exe. Uvedený nástroj je možné používat jednak pro vytváření self-signed certifikátů[23], ale hlavně si můžeme vytvořit certifikační autoritu a jí podepsané serverové i klientské certifikáty. Nástroj lze nainstalovat společně s programem Visual Studio nebo z webových stránek firmy Microsoft. Nadále se budeme zabývat právě tímto nástrojem.

Po stažení a instalaci se program rozbalí v C:\Program Files (x86) \ Microsoft SDKs\Windows\v7.0A\Bin. Program spouštíme v příkazovém řádku. Nejdříve si připravíme adresář makecert, kde budeme ukládat certifikáty, pak pomocí příkazu makecert a.cer vytvoříme certifikát a.cer

---

[22] Windows SDK (software development kit) sada vývojářských utilit a programových rozhraní k tvorbě aplikací pro Windows, obvykle v jazyce C nebo C++.

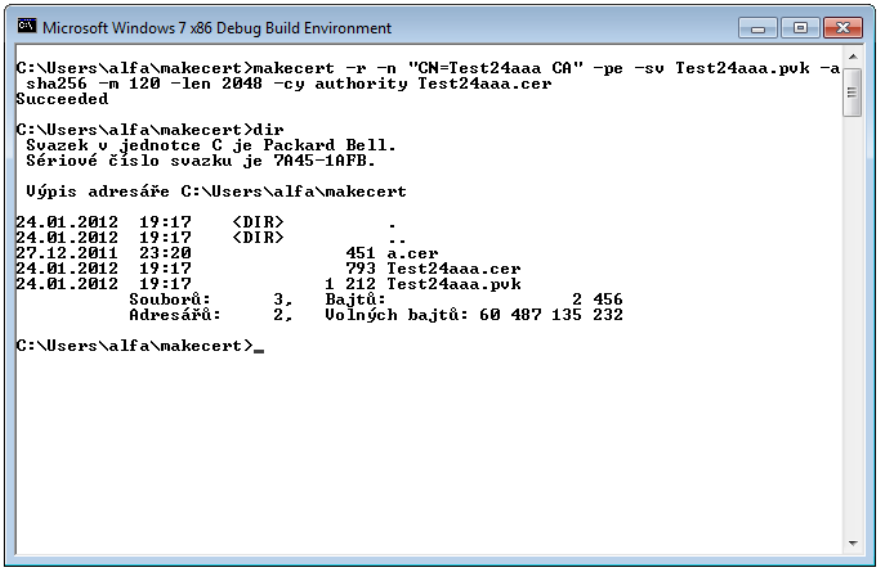
[23] Self – signed certifikát je v kryptografii specifická forma digitálního certifikátu , který podepsal sám jeho tvůrce, který se tak zároveň stal certifikační autoritou. Používá se pro testování nebo pro potřeby uzavřených okruhů uživatelů (škola, firma či jiná komunita). Ověření takového certifikátu se děje jiným způsobem, typicky kontrolou jeho otisku.



Obr. 7. Vytvoření adresáře

Následně již vytvoříme certifikát certifikační autority,k tomu použijeme následující příkaz[24]:

makecert -r -n "CN=Test24aaa CA" -pe -sv Test24aaa.pvk -a sha256 -m 120 -len 2048 -cy authority Test24aaa.cer



Obr. 8. Vytvoření certifikátu

[24] jedna z nejobsáhlejších manuálových stránek se nachází na těchto www stránkách

http://stuff.mit.edu/afs/athena/software/mono_v2.8/man/man1/makecert.1

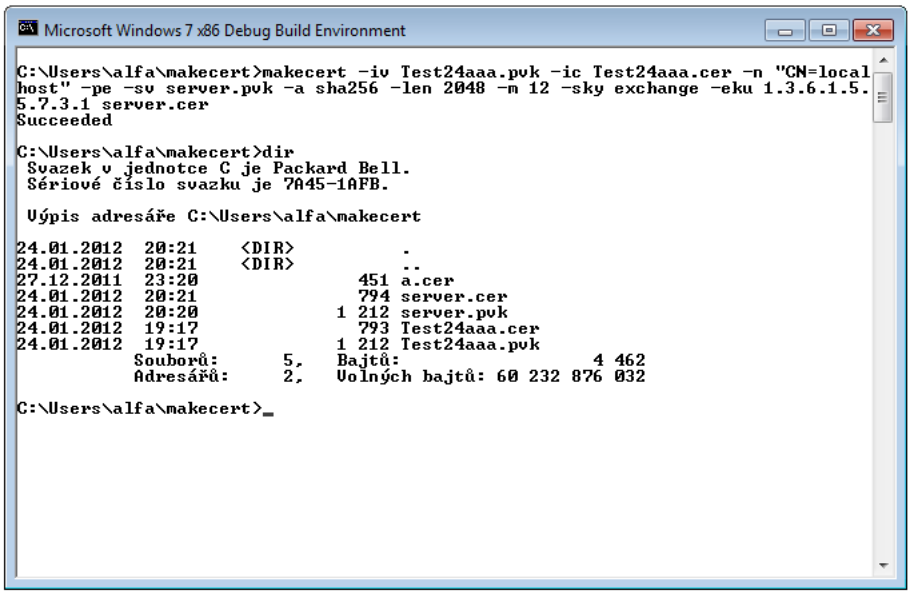
Význam parametrů je následující:

-r	Vytvoří self-signed certifikát, tedy vystavovatel je stejný jako subjekt
-n " CN "	"CN" je common name, nejdůležitější parametr, ale zde můžete napsat jakkoliv složité jméno.
-pe	Vygenerovaný soukromý klíč bude označen jako exportovatelný.
-sv Test24aaa.pvk	Název souboru, do nějž bude uložen soukromý klíč.
-a sha256	Algoritmus použitý pro podpis. Výchozí hodnota je sha1 (což je v nouzi ještě použitelné), pro nové certifikáty se doporučuje používat nejméně sha256.
-m 120	Doba platnosti certifikátu v měsících. Pro CA se obvykle nastavují delší hodnoty.
-len 2048	Délka klíče v bitech. Výchozí a dnes již stěží přijatelná je hodnota 1024 bitů, doporučuje se používat 2048 nebo 4096.
-cy authority	Typ certifikátu může být end = koncový uživatel nebo authority = certifikační autorita.
root.cer	Název souboru, kam bude uložen nový certifikát.

Výsledkem tohoto příkazu budou soubory Test24aaa.pvk a Test24aaa.cer. První z nich představuje soukromý klíč a měli byste ho střežit jako oko v hlavě. Druhý je kořenový certifikát. Ten musíte naopak obvyklým způsobem nainstalovat do všech počítačů, které mají tuto CA pokládat za důvěryhodnou. Na obr. 8 je vidět výše popsaný postup. V průběhu provádění příkazu jsme vyzvání, zda chceme vytvořit heslo privátního klíče. Pokud ano, musíme vyplnit heslo a to následně potvrzujeme. Než je příkaz dokončen, jsme vyzváni k zadání hesla privátního klíče a následně je nám vypsáno úspěšné vytvoření souborů *.pvk a *.cer

V dalším kroku použijeme tuto certifikační autoritu pro vytvoření serverového certifikátu, který je typický pro web server nyní použijeme příkaz:

```
makecert -iv Test24aaa.pvk -ic Test24aaa.cer -n "CN=localhost" -pe -sv server.pvk -a sha256 -len 2048 -m 12 -sky exchange -eku 1.3.6.1.5.5.7.3.1 server.cer
```



Obr. 9. Vytvoření serverového certifikátu

Význam parametrů je:

-iv Test24aaa.pvk	Cesta k soukromému klíči CA.
-ic Test24aaa.cer	Cesta k veřejnému klíči (certifikátu) CA.
-n "CN=localhost"	Opět distinguished name uvedený v certifikátu. Pro použití v rámci web serveru se musí obvykle CN rovnat DNS názvu serveru.
-pe	Vygenerovaný soukromý klíč bude označen jako exportovatelný.
-sv server.pvk	Název souboru, do nějž bude uložen soukromý klíč.
-a sha256	Algoritmus použitý pro podpis. Výchozí hodnota je sha1 , pro nové certifikáty se doporučuje používat nejméně sha256.
-m 12	Obecné certifikáty (serverové, klientské) se obvykle vystavují nak kratší dobu, v tomto případě na 12 měsíců.
-sky exchange	Typ klíče je typicky signature (pro elektronický podpis) nebo exchange (výměna klíčů, šifrování).
-eku 1.3.6.1.5.5.7.3.1	Numerické identifikátory (OID) účelů, pro které lze certifikát využít. V tomto případě OID1.3.6.1.5.5.7.3.1 znamená Server Authentication.

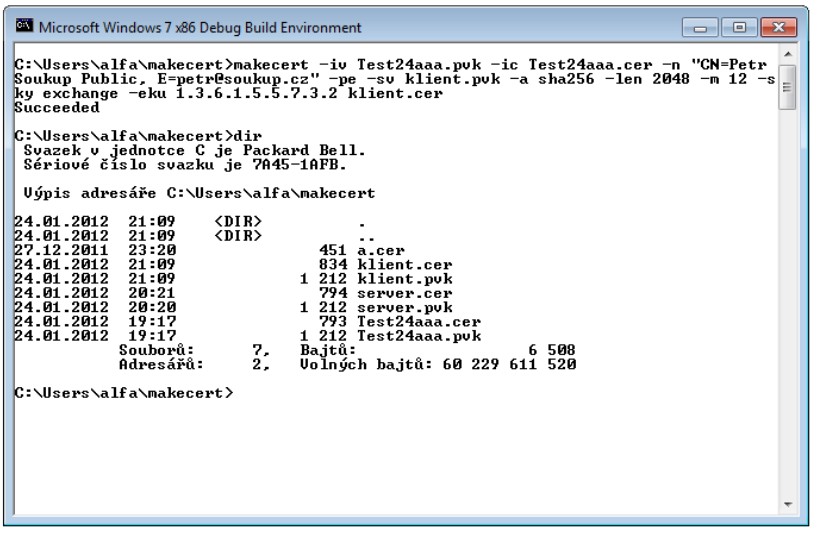
server.cer Název souboru, kam bude uložen nový certifikát.

Na obrázku 9 vidíme realizaci druhého kroku. Ve výpisu adresáře vidíme, že máme k dispozici další soukromý klíč server.pvk a další kořenový certifikát server.cer, ale vytvořeny na základě prvního kroku.

Nyní nám zbývá vytvořit klíč a certifikát pro klienta. Pro třetí krok použijeme příkaz:

```
makecert -iv Test24aaa.pvk -ic Test24aaa.cer -n "CN=PetrSoukup Public, E=petr@soukup.cz" -pe -sv klient.pvk -a sha256 -len 2048 -m 12 -sky exchange -eku 1.3.6.1.5.5.7.3.2 klient.cer
```

Příkaz pro vygenerování je prakticky stejný, jako u serverového certifikátu. Liší se jenom v jiném distinguished name (typicky udáváme nejméně jméno osoby a její e-mailovou adresu) a především v použitém OID, které má na konci dvojku - 1.3.6.1.5.5.7.3.2 znamená Client Authentication.



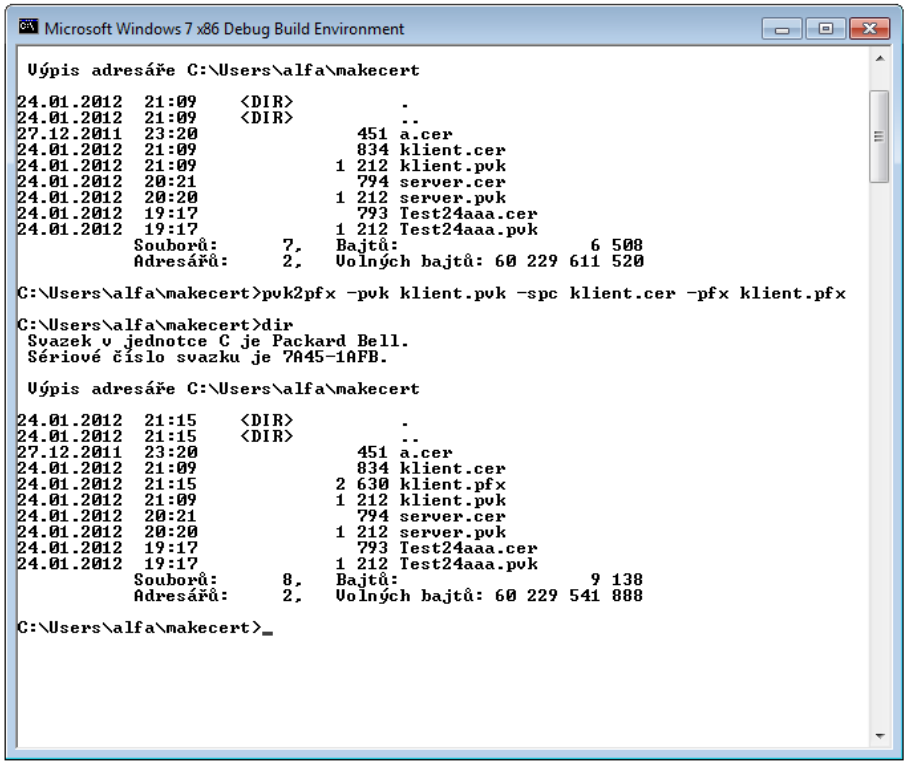
Obr. 10. Certifikát pro klienta

Pro serverovou a klientskou autentizaci potřebujeme certifikát importovat do úložiště včetně soukromého klíče. Samostatný PVK soubor nelze importovat přímo, oba klíče nejprve zkonvertujeme do formátu dle PKCS#12[25], tedy do souboru s příponou obvykle PFX nebo P12. K tomu použijeme program pvk2pfk, který je součástí Windows SDK.

[25] PKCS (Public Key Cryptographic Standards) je skupina standardů pro kryptografii s veřejným klíčem

Příkaz bude mít následující podobu: `pvk2pfx -pvk klient.pvk -spc klient.cer -pfx klient.pfx`

Vstupem je privátní klíč klient.pvk a certifikát klient.cer, výstupem soubor klient.pfx.

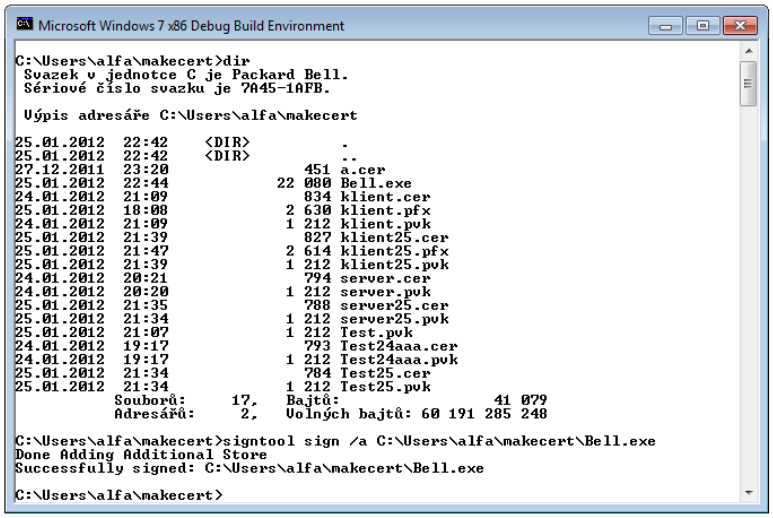


Obr. 11. Vytvoření klient.pfx

Nyní když už máme vytvořený soubor .pfx, můžeme si ukázat praktické využití vlastní certifikační autority. Pomocí vytvořeného souboru klient.pfx je možné podepsat libovolný spustitelného programu v operačním systému Windows , což jsou soubory typu *.exe, *.dll a další.

Použijeme příkaz signtool v tomto tvaru:

`signtool sign /a C:\Users\Alfa\manecert\Bell.exe`



Obr. 12. Podepsání souboru

kde

/a vybere nejlepší certifikát k podepsání

/f klient.pfx zadání certifikátu použitého k podepsání

/v vypíše zprávu o prováděném podepisování

Dále je možné podpis opatřit i časovým razítkem /t timestamp_url.

Jak již bylo uvedeno na začátku, pomocí signtool sign [options] <filename> jde podepsat libovolný program, nejenom náš. Toho lze s úspěchem využívat v bezpečnostní opatřeních podniků. A to tak, že povolíme používat pouze programy, které jsou podepsány pouze autoritou, které podnik důvěřuje, zpravidla někým z managementu. Toto důvěryhodná osoba nemusí zjišťovat, zda je používaný program digitálně podepsán a zde je podepsán důvěryhodnou osobou. A pokud není, nemusí vyhledávat autora programu a přesvědčovat ho k tomu, aby program podepsal. Stačí k libovolnému spustitelnému programu připojit podpis se svým vlastním certifikátem a následně má bezpečnostní manažer zajištěno, že nebude moci být používáno v podniku žádné jiné než prověřené programové vybavení.

ZÁVĚR

V bakalářské práci jsem popsal základní prostředky, které používá infrastruktura veřejného klíče. Na několika příkladech jsem demonstroval základní principy fungování některých prvků této struktury. Ukázal jsem hlavní princip hašovací funkce, vytvořili jsme podepsanou zprávu a ukázal jsem jak je možné pracovat s certifikačními autoritami. Myslím si, že jsem splnil hlavní cíl, kterým bylo vysvětlit a na příkladech ukázat základní principy s kterými se setkáváme v PKI. Na druhé straně je potřeba přiznat, že hlavním problémem při větším využívání prostředků veřejných klíčů není technická složitost této struktury. Ale hlavním důvodem nižšího zájmu o prostředky PKI jsou až nepřiměřeně vysoké nároky na elektronický podpis, ve smyslu autorizace a autentizace. Když se zamyslíme nad tím, v jakých je klasický podpis používán v absurdních a nesmyslných situacích, ale nikoho netrápí, že vypovídací hodnota klasických podpisů v některých případech je téměř nulová. Můžeme si uvést několik příkladů. Pokud budete omlouvat dítě ze školy, napíšete nějaký důvod a připojíte libovolný shluk znaků, který prohlásí dítě za podpis rodiče, přestože nikdo nikdy ve škole váš podpis neviděl, je vše v pořádku. Když někdo omluví své dítě ve škole prostřednictvím emailu, je zpravidla následně vyžadován i klasický podpis. Zajímavá je reakce většiny lidí, kterým výše popsanou událost popisuji. Hned mi řeknou, že je možné psát email za někoho jiného, že komunikace na internetu není bezpečná. Což je pravda i když, kolik lidí to umí? Ale podstatné je to, že málokdo se zamyslí nad tím, že i klasický podpis je možné zfalšovat. A nikoho již nezajímá autorizace podpisu. Lidé většinou argumentují tím, že v případě zfalšování klasického podpisu se jedná o protiprávní činnost. A zde se dostáváme k jádru problému. Stále přistupujeme ke komunikaci mezi počítači jako k něčemu, co se nachází někde vedle našeho reálného světa. Stále platí v podvědomí lidí, že když něco napíši na papír a podepíši má to větší váhu než jakýkoliv digitální podpis.

V celé struktuře PKI existuje i několik problematičtějších oblastí, jednou z nich jsou certifikační autority. Jakožto subjekty, které musí vytvářet zisk a tedy vydávání certifikátů rády a často zpoplatňují. Existoval sice elektronický podpis od certifikační autority Thawte, který byl v určitém období bezplatný, ale nyní již tato možnost není. Na druhé straně je uživatel, který musí za certifikát každý rok platit určitý obnos a zároveň opětovně procházet schvalovacím procesem. Proces schvalování může být pro některé uživatele nepřiměřeně složitý. Zároveň nezná uživatel odpovědi na několik základních otázek: Jak moc je certifikační autorita důvěryhodná? Jak je zabezpečený můj soukromý klíč? A

v neposlední řadě také uživatel neví zda jak hodnověrně byla provedena autentizace podepsané osoby.

Dalším problémovým okruhem je vlastní správa soukromých klíčů. Klíče používané v PKI musí být uloženy v nějaké elektronické podobě tak, aby jej mohla přečíst aplikace, kterou uživatel používá. Vždy se ale jedná o data, která jsou uložena na disku počítače a tudíž jsou teoreticky čitelná pro kohokoliv, kdo má oprávnění číst příslušnou část disku.

Jedním z dalších problémů při užívání PKI je i bezpečné zničení nepoužívaných klíčů. Prosté smazání souboru, v němž byl klíč uložen, v tomto případě nestačí. Často je třeba mít k dispozici podrobný popis postupu, jak toho docílit. Mimo jiné tak, že veškeré elektronicky uložené klíče by měly být po smazání kompletně přepsány, aby se o nich nikde neuchovala žádná informace, kterou by útočník mohl zneužít. To je velice důležité především u softwarových aplikací, jež ukládají klíče do paměti, kterou pak lze využít i k jiným účelům.

Širšímu využívání struktury veřejných klíčů nebrání ani tak zdánlivá složitost, ale pouze nedostatečná informovanost. Když dokážeme bez problému využívat strukturu veřejných klíčů v bankovnictví, není daleko ani doba, kdy budeme naplno využívat další produkty, založené na struktuře veřejných klíčů. K tomu, aby tato doba byla co nejkratší, by mohla přispět i moje bakalářská práce.

ZÁVĚR V ANGLIČTINĚ

In the Bachelor's work I described the basic resources, which uses the public key infrastructure. A few examples illustrate the basic principles of functioning of I certain elements of this structure. I showed the main principle of the hash function, we have created a signed message and I showed how you can work with certification authorities. I think that I have fulfilled the main objective, which was to explain and show examples of the basic principles of that occurring in the PKI. On the other hand, it is necessary to admit that the main problem in the more intensive use of resources of the public keys is not the technical complexity of this structure. But the main reasons for lower interest on the funds of the PKI are unreasonably high demands on electronic signature within the meaning of authorization and authentication. When we contemplate that, in what is a classic signature used in the ridiculous and absurd situations, but nobody bothers that the explanatory value of classic signatures in some cases is almost zero. We can give a few examples. If you'll excuse the child from school, write for a reason and you connect any cluster of characters, which declares the child for the signature of the parents, even though no one ever in school your signature seen, everything is all right. When someone will excuse your child in school is usually through email, and subsequently required the signature of the classic. Interesting is the reaction of most people, which is an event described above. I will say that it is possible to write email for someone else, that the communication on the Internet is not secure. Which is true though, how many people can do it? But the point is that few ponder over the facts that even the classic signature it is possible to falsify and already not interested in authorization signature. People often argue that in the case of falsification of classical signature is an illegal activity. And here we come to the core of the problem. Still treat the communication between the computer as to something that is situated somewhere next to our real world. Still in the subconscious of people, that when something I'll write on the paper and I'll sign it has greater weight than any digital signature. The whole structures of PKI there are several problematic areas, one of which is the certification authority. As the bodies required to make a profit and, therefore, issuing certificates and often like to be billed. Although there was an electronic signature from the certificate authority Thawte, who was in a period free, but now this option is. On the other hand, it is the user who must pay for the certificate every year a specific amount and at repeatedly through the approval process. The approval process may be unduly complex for some users. At the same time, the user does not know the answers to some basic questions: how

much is the certificate authority trusted? How secure is my private key? And last but not least also the user knows whether or how to prove to the authentication has been performed of the signatory. Another problem circuit is the management of the private keys. The keys used in PKI must be stored in any electronic form so that it can read the application that the user is using. But it was always the data that is stored on the hard disk of your computer and therefore are, in theory, to read for anyone who has permission to read the appropriate section of the disk. One of the other problems in the use of PKI is the safe destruction of unused keys. Simple deletion of the file in which the key is stored, in this case is not enough. Often, you must have available a detailed description of the procedure, as that. Inter alia, so that all electronically stored keys should be deleted completely overwritten, to value yourself anywhere, no information that an attacker could exploit. This is very important, especially for software applications, which store keys in memory, which can then be used for other purposes. wider use of the structures of public keys does not prevent even the apparent complexity, but only inadequate information. If we can no problem to use the structure of the public keys in the banking sector, is not far when we will fully use other products, based on the structure of public keys. To this period was as short as possible, could contribute to my thesis.

SEZNAM POUŽITÉ LITERATURY

[1] MENEZES, Alfred J. *Handbook of applied cryptography*. Vyd. 1. Boca Raton: CRC Press, 1997, 780 s. ISBN 08-493-8523-7.

[2] DOSTÁLEK, Libor, Marta VOHNOUTOVÁ a Miroslav KNOTEK. *Velký průvodce infrastrukturou PKI a technologií elektronického podpisu*. 2., aktualiz. vyd. Brno: Computer Press, 2009, 542 s. ISBN 978-802-5126-196.

[3] PETERKA, Jiří. *Báječný svět elektronického podpisu*. Praha: CZ.NIC z.s.p.o., 2011. ISBN 978-80-904248-3.

[4] BUDIŠ, Petr. *Elektronický podpis a jeho aplikace v praxi*. 1. vyd. Olomouc: ANAG, 2008, 157 s. ISBN 978-807-2634-651.

[5] BOSÁKOVÁ, Dagmar. *Elektronický podpis*. Vyd. 1. Praha: ANAG, 2002, 141 s. ISBN 80-726-3125-X.

[6] http://www.uncitral.org/pdf/english/workinggroups/wg_ec/wp-79.pdf [online]. [cit. 2012-02-10].

[7] http://bart.math.muni.cz [online]. [cit. 2012-02-16]. Dostupné z: http://bart.math.muni.cz/~fuchs/ucitel/clanky/1_3_5.pdf

[8] http://www.math.muni.cz [online]. [cit. 2012-01-28] Dostupné z: http://www.math.muni.cz/~bulik/vyuka/Algebra-2/alg2-screen.pdf

[9] http://cryptography.hyperlink.cz/ [online]. [cit. 2012-02-24]. Dostupné z: http://cryptography.hyperlink.cz/MD5_collisions.html

[10] http://www.crypto-world.info/. [online]. [cit. 2012-02-12]. Dostupné z: http://crypto-world.info/casop7/crypto03_05.pdf

[11] http://www.mvcr.cz. [online]. [cit. 2012-01-28]. Dostupné z: http://www.mvcr.cz/clanek/zmena-v-kryptografickych-algoritmech-ktere-jsou-pouzivany-pro-vytvareni-elektronickeho-podpisu.aspx

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

PKI	Public key infrastructure - Infrastruktura veřejného klíče
MD5	Message-Digest algorithm 5 - skupina hašovacích funkcí
NIST	National Institute of Standards and Technology - Národní institut standardů a technologie je institut při Ministerstvu obchodu USA
DES	Data Encryption Standard, první veřejný šifrovací standart
AES	Advenced Encryption Standard, nahrazuje DES
UNCITL	United Nations Commission on International Trade Law – Komise OSN pro mezinárodní obchodní právo
CA	Certifikační autorita
CP	Certifikační politika
CPS	certifikační prováděcí směrnice
SSL	Secure Sockets Layer, vrstva poskytující zabezpečení komunikace
PKCS	Public-key Cryptography Standards, standarty kryptografie veřejných klíčů
SDK	Software Development Kit, sada vývojových a programových rozhraní k tvorbě aplikací

SEZNAM OBRÁZKŮ