Libor Dostálek, Marta Vohnoutová, Miroslav Knotek

Velký průvodce infrastrukturou PKI

a technologií elektronického podpisu

2. aktualizované vydání

Zabezpečení elektronické komunikace
Certifikáty, veřejné klíče a bezpečnostní protokoly
Budujeme vlastní certifikační autoritu
Role ADCS ve Windows Serveru 2008 R2

CPRESS

Libor Dostálek, Marta Vohnoutová, Miroslav Knotek

Velký průvodce infrastrukturou PKI
a technologií elektronického podpisu
2. aktualizované vydání

Computer Press, a. s.
Brno
2009

Velký průvodce infrastrukturou PKI
a technologií elektronického podpisu
2. aktualizované vydání

Libor Dostálek, Marta Vohnoutová, Miroslav Knotek

Computer Press, a. s., 2009.

Jazyková korektura: Marie Schreinerová
Vnitřní úprava: Petr Klíma
Sazba: Petr Klíma, Dagmar Hajdajová
Rejstřík: Libor Dostálek
Obálka: Martin Sodomka
Komentář na zadní straně obálky: Libor Pácl
Technická spolupráce: Jiří Matoušek, Zuzana Šindlerová
Odpovědný redaktor: Libor Pácl
Technický redaktor: Jiří Matoušek
Produkce: Petr Baláš

Computer Press, a. s.,
Holandská 8, 639 00 Brno

Objednávky knih:
http://knihy.cpress.cz
distribuce@cpress.cz
tel.: 800 555 513

ISBN 978-80-251-2619-6
Prodejní kód: K1717
Vydalo nakladatelství Computer Press, a. s., jako svou 3426. publikaci.

Stručný obsah

Obsah

Kapitola 3

Kapitola 4

Kapitola 8

Kapitola 9

Kapitola 10

Kapitola 11

Kapitola 12

Kapitola 13

Kapitola 14

Kapitola 18

Kapitola 19

Kapitola 23

Kapitola 24

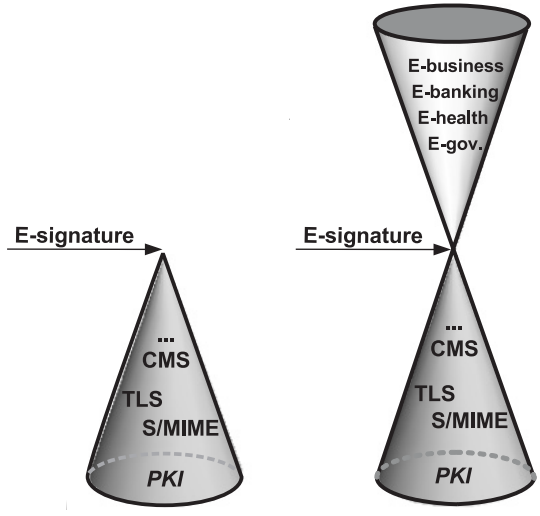
Kapitola 25

Kapitola 27

Úvod

Je to již několik let, kdy jsme byli naposledy v Paříži. I tenkrát jsme si vzpomněli na Petera Sylvestera. A hned nás napadlo, že se u něj opět zastavíme. P. Sylvester je spoluautor legendárního standardu-nestandardu RFC-3029 „*Internet X.509 Public Key Infrastructure: Data Validation and Certification Server Protocols*“, který už tehdy mnozí kritizovali, ale přitom nikdo nedokázal vymyslet nic lepšího. Což bohužel víceméně platí dodnes.

I přes stávku pařížských dopraváků jsme dorazili včas a začali naši diskusi. Uprostřed diskuse Peter namaloval kužel (obr. ú.1 vlevo), který komentoval slovy, že PKI si můžeme představit jako podstavu kužele, nad níž je vybudována řada protokolů (S/MIME, TLS, CMS, IPsec, EAP-TLS…). Na vrcholu kužele je pak elektronický podpis.



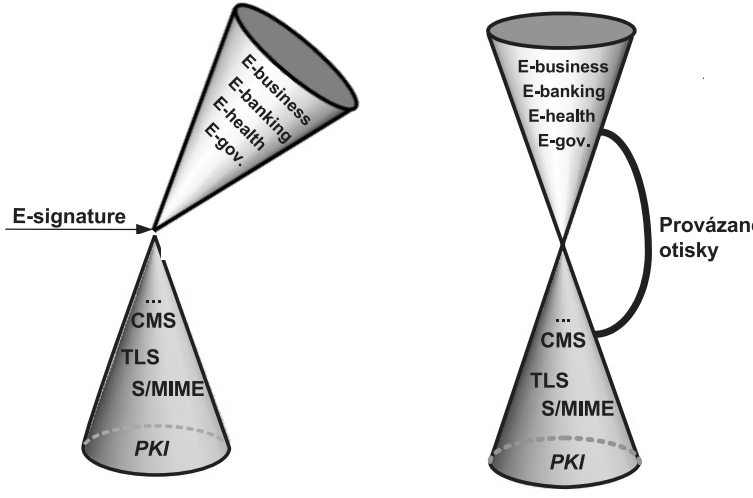
Obrázek ú.1: Sylvesterovy kužely

Vedle namaloval týž kužel, ale na jeho vrchol přidal ještě další kužel otočený vrcholem dolů (obr. ú.1 vpravo). A pokračoval tvrzením, že na tom jediném elektronickém podpisu stojí všechny nejrůznější aplikace jako E-government, E-health, E-banking, E-business, E-procurement a kdoví jaké další „E-“.

„No a nyní si stačí představit“, zaníceně pokračoval, „že někdo jen zpochybní ten elektronický podpis.“ A už maloval další kužely (obr. ú.2). Hned bylo vidět, jak se celý ten humbuk „E-“ kácí jako krabička sirek. Zdůrazňoval, že je třeba hledat i jiné algoritmy a postupy, které ty užitečné aplikace podepřou, a jako rozumný mu připadal systém provázaných otisků (viz kapitola 20).

Nás tyto Sylvesterovy kužely přímo nadchly. Avšak u mnohých kolegů jsme s nimi nepochodili. Připadalo jim to totiž nadnesené.

Cílem této publikace je začít zkoumat Sylvesterovy kužely od spodní podstavy, kterou je PKI. Dále si objasníme zejména protokoly popsané ve spodním kuželu a elektronický podpis. Pochopitelně že kužely rovněž pořádně zatřepeme, když si položíme otázku o platnosti elektronického podpisu po vypršení platnosti certifikátu určeného k ověření tohoto podpisu. A nebojte se, i na provázané otisky dojde.



Obrázek ú.2: Provázané otisky možná pomohou udržet Sylvesterovy kužely ve správné poloze nad sebou

Jak tuto knihu číst

Kniha je určena jak pro začátečníky v oblasti PKI, tak i pro odborníky, kteří se potřebují dozvědět řadu detailů. Aby začátečníci nebyli zahlceni, je prvních deset kapitol napsáno populární formou tak, aby byly dobře srozumitelné i pro ně. Těchto prvních 10 kapitol objasňuje princip certifikátu veřejného klíče a jeho životní cyklus.

Kapitoly 11, „Má první certifikační autorita“, a 12, „Wireshark“, jsou určeny šťouralům, kteří si chtějí pohrát s jednoduchou certifikační autoritou a připravit se na pitvání nejenom certifikátu po jednotlivých bitech.

Přelomovou kapitolou je kapitola 13, „ASN.1, BER, DER,UTF-8 a Base64“, zabývající se jazykem ASN.1 sloužícím k definování jednotlivých datových struktur. Dále se zabývá kódováním BER a DER těchto struktur pro počítačovou komunikaci. Pokud se laskavý čtenář seznámí s jazykem ASN.1 a kódováním BER a DER (tj. s obsahem této kapitoly), pak bez jakýchkoliv problémů může rozebírat dále popisované datové struktury po jednotlivých bitech. Stane se tak pokročilým čtenářem této publikace.

Kapitoly 14, „Žádost o vydání certifikátu pod lupou“, 15, „Certifikát pod lupou“, 16, „Žádost a odvolání certifikátu pod lupou“, a 17, „CMP a CMC“, jsou určeny pro pokročilé čtenáře. Mají obdobný obsah jako kapitoly 1–10, ale zaměřují se na detailní popis jednotlivých datových struktur.

Zbývající část publikace pak obsahuje tematicky zaměřené kapitoly (Atributové certifikáty, Časová razítka, Bezpečný web, Bezpečná pošta, Dlouhodobý digitální podpis a Dlouhodobá archivace). Tyto kapitoly jsou určeny jak začátečníkům, tak i pokročilým čtenářům. Začátečníci jen přeskočí popisy jednotlivých datových struktur.

Kapitola 27, „Budujeme PKI, TSA a důvěryhodné archivy", je pak závěrem celé publikace.

Poděkování

Chtěli bychom poděkovat všem, kteří nám zapůjčili nejrůznější zařízení, abychom mohli připravit jednotlivé příklady. Dále bychom chtěli poděkovat Luďku Raškovi za podnětnou odbornou korekturu a Michalu Hojsíkovi, který rukopis pozorně přečetl a opravil mnohé chyby.

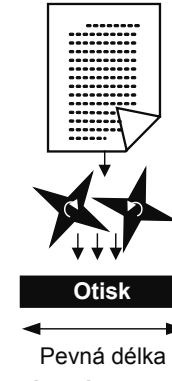
Kapitola 1

Symetrická a asymetrická kryptografie

Téměř v každé učebnici je kryptografická komunikace vysvětlována na komunikaci mezi Alicí a Bobem. Jenže naše drahá Alice a její Bob jsou již tak staří, že přestali kryptograficky komunikovat. Naštěstí mají potomka Aloise, který pokračuje v rodinné tradici kryptografické komunikace se svou milou Bohumilou.

Otisk (hash)

Nejprve si ukážeme, jak mocným nástrojem je otisk (*hash*). Otisk je jednocestná funkce, která nám z libovolně dlouhého textu vytvoří krátký řetězec konstantní délky. Výsledný řetězec (otisk) by měl maximálně charakterizovat původní text. Typická velikost výsledného textu je 16 B (např. algoritmus MD-5) nebo 20 B (algoritmus SHA-1). Dnes se již algoritmy MD-5 a SHA-1 vesměs považují za slabé, proto se stále častěji setkáváme s novými algoritmy, produkujícími ale delší otisky: SHA-224 (otisk dlouhý 28 B), SHA-256 (otisk 32 B), SHA-384 (otisk 48 B) a SHA-256, někdy též označovanou SHA-2 s otiskem dlouhým 64 B.



Obrázek 1.1: Otisk

Jednocestnou funkcí se rozumí algoritmy, které nejsou výpočetně náročné. Je však výpočetně velice náročné k výsledku nalézt původní text. Jednocestnou funkci lze přirovnat k manželství. Je přece jednoduché se oženit, ale často velice obtížné se rozvést.

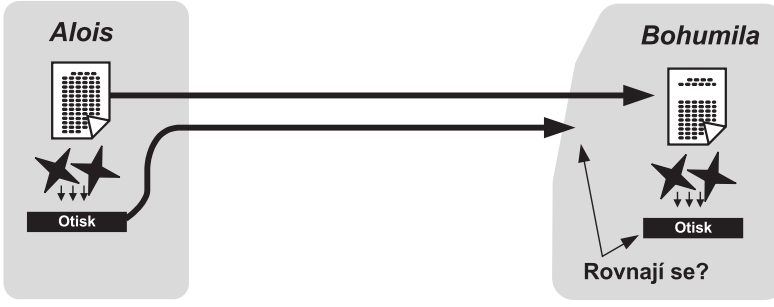
Kvalitní jednocestné funkce pro výpočet otisku by měly dát výrazně jiný výsledek při drobné změně původního textu. Počítáme-li např. otisk pro digitální podpis z textu nesoucího platební příkaz, pak by bylo nemilé, kdyby se nám po připsání nuly k převáděné částce otisk nezměnil.

Jelikož se otisk počítá z libovolně dlouhého textu, tak ke konkrétnímu otisku je teoreticky možné najít nekonečně mnoho původních textů. U některých algoritmů (např. MD-5) se již daří nacházet texty se stejným otiskem. Výsledkem je pak opouštění těchto algoritmů a jejich nahrazení jinými (algoritmy třídy SHA-2, FIPS PUB 180-2; algoritmus WHIRLPOOL – ISO/IEC 10118-3:2003).

Jednocestné funkce jsou konstruovány na výpočetních operacích nízké úrovně (především bitové operace a posuny), a jsou tedy výpočetně velmi rychlé a efektivní. Algoritmy pro výpočet otisku nejsou v žádném případě šifrovacími algoritmy (už vzhledem k nejednoznačnosti – obecně neexistuje inverzní funkce), ale používají se v roli kvalitního „otisku prstu dat" (fingerprint).

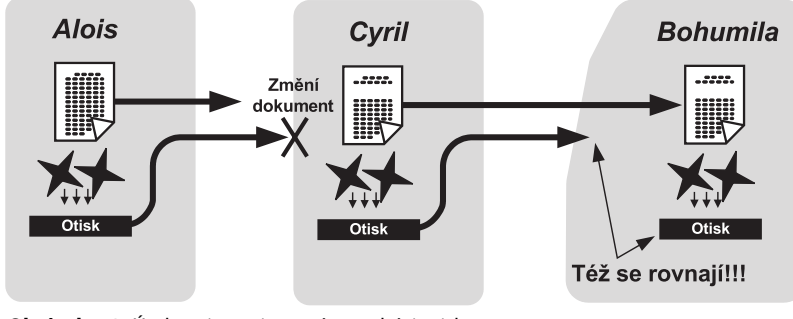
Jak využije Alois otisk ve své komunikaci se svou milou Bohumilou? No přece využije otisk jako důkaz, že zpráva na cestě od Aloise k Bohumile nebyla změněna, tj. využije otisk jako důkaz integrity zprávy. Alois neodešle pouze samotná data (text) zprávy, ale data doplní o patu zprávy (*trailer*) obsahující otisk z textu zprávy (obr. 1.2).

Bohumila, poté co přijme zprávu, spočte otisk z přijaté zprávy a porovná svůj výsledek s otiskem ze zápatí přijaté zprávy (tj. s otiskem spočteným Aloisem). Pokud jsou oba otisky shodné, zpráva nebyla cestou změněna. Tj. Bohumila provedla kontrolu integrity zprávy. Tento typ důkazu integrity přenášených dat využívají linkové protokoly (např. Ethernet) pro detekci chyb vzniklých poruchami linek (poruchami fyzické vrstvy komunikace v počítačové síti).



Obrázek 1.2: Využití otisku jako důkazu integrity zprávy

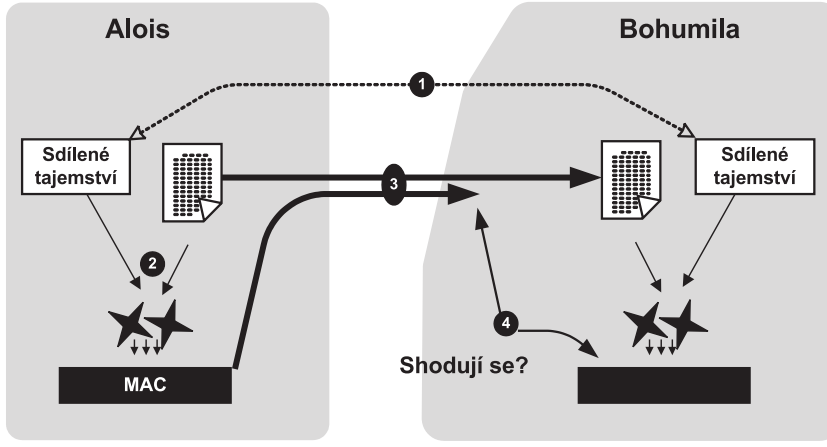
Jenže algoritmus výpočtu otisku používaný Aloisem je veřejně popsán v příslušné technické normě. Tuto normu si přečte i žárlivý Cyril, který zprávu od Aloise pozmění v jeho neprospěch a ten pošle Bohumile (obr. 1.3).



Obrázek 1.3: Útok na integritu zprávy na bázi otisku

Naštěstí Alois s Bohumilou Cyrila znají, proto si při své tajné schůzce vymění tajemství (šipka 1 na obr. 1.4), které sdílí pouze Alois s Bohumilou, a Cyril je tudíž nezná. Jako sdílené tajemství mezi Aloisem a Bohumilou stačí např. nějaký krátký textový řetězec.

Od chvíle, kdy si Alois s Bohumilou vyměnili sdílené tajemství, tak vždy, když bude Alois odesílat nějakou zprávu Bohumile, nebude otisk počítat pouze ze zprávy, ale do výpočtu otisku zahrne i sdílené tajemství (šipka 2 na obr. 1.4). Jelikož Cyril tajemství nezná, není schopen takovýto otisk spočíst, proto nemůže pozměňovat zprávu, aniž by to Bohumila nepoznala.



Obrázek 1.4: Zajištění integrity přenášených dat pomocí sdíleného tajemství. Na tomto principu je založen algoritmus HMAC (Keyed-Hashing for Message Authentication) – viz RFC-2104.

Otisk spočtený nejenom ze zprávy, ale ze zprávy nějakým způsobem zřetězené se sdíleným tajemstvím se často označuje jako MAC* (*Message Authentication Code*). MAC se využívá velice často, např. v protokolech SSL/TLS, protokolu IPsec, ale ve své podstatě jsou na této technice postaveny i autentizační kalkulátory pro tvorbu jednorázových hesel.

MAC se někdy označují jako „symetrický podpis". Z tohoto označení však mnohým kryptologům naskakují pupínky. Proč? Protože na rozdíl od digitálního podpisu se takto nedá zaručit pravost dokumentu („nepopiratelnost" – *non repudiation*), ale pouze jen integrita přenášených dat.

Vysvětlení je prosté. Kdyby Bohumila byla potvora, zprávu od Aloise by sama změnila a spočetla z ní „symetrický podpis". A Aloisovi by se vysmála. Kdyby se Alois divil, tak by mu ukázala změněnou zprávu a řekla mu: „Vidíš, co jsi zač, vždyť mě nemáš rád." A Alois by se spravedlnosti nedovolal, protože by nemohl dokázat, zdali „symetrický podpis" opravdu vytvořil on, nebo jej podvrhla Bohumila.

Replay attack, nonce

Uvedený mechanismus má jeden velký nedostatek. Útočník může odposlechnout a zaznamenat přenášená data zaslaná Aloisem Bohumile včetně MAC, a po chvíli to celé Bohumile zaslat znovu (zopakovat). Tento typ útoku se označuje jako *replay attack*.

Pokud Alois zasílá Bohumile milostný dopis, útočník Bohumilu nanejvýš potěší, když jí Aloisův dopis pošle ještě jednou. Avšak pokud Alois neposílá milostný dopis Bohumile, ale posílá platební příkaz do banky, pak bude jeho platební příkaz zaplacen dvakrát a Alois přijde o peníze (v bankovnictví se to označuje jako *dual spend attack*).

Tomuto útoku se Alois brání např. pomocí vzestupného číslování svých zpráv. Pokud Bohumila obdrží zprávu nižšího než očekávaného čísla, pochopí, že se jedná o zopakovanou starou zprávu. Obdobně banka nezpracuje dva příkazy stejného čísla.

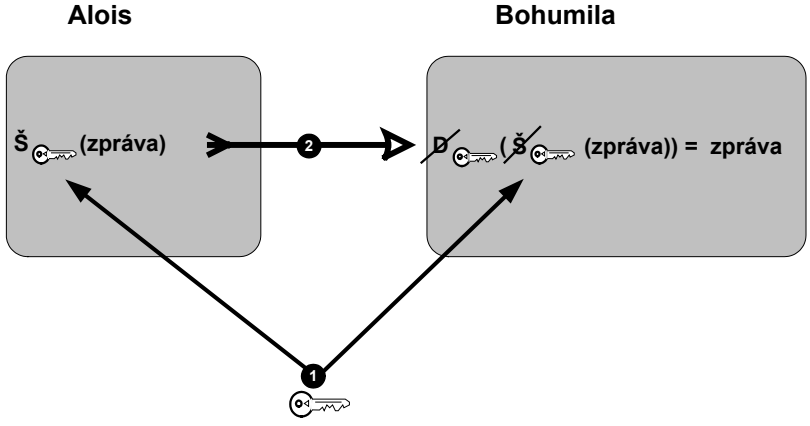
* MAC se někdy do češtiny překládá jako „kryptografický kontrolní součet".

Jinou obranou je nonce. Nonce je dostatečně dlouhé náhodné číslo (zpravidla více jak 16 B dlouhé), které Alois vždy přidává do své zprávy (k přenášeným datům). Tím zajistí, že je nepravděpodobné, aby Alois odeslal dvě stejné zprávy, a tudíž generoval dva stejné MAC. Avšak Bohumila musí kontrolovat, zdali již v minulosti neobdržela tutéž zprávou se stejnou nonce.

Jakou chybu může Alois udělat? Např. může použít chybný software, který negeneruje čísla náhodně. Pak může vytvořit dvě stejné nonce. Aby se i tomuto předešlo, tak se někdy nonce vytváří tak, že se skládá ze dvou částí: jedna část obsahuje náhodné číslo a druhá část obsahuje datum a čas, který je sám o sobě neopakovatelný.

Symetrické šifry

Jenže Alois s Bohumilou si mohou také přát, aby byl jejich vztah zachován v tajnosti (aby byla zachována privátnost jejich vztahu), proto svou komunikaci šifrují. K dispozici mají např. symetrické šifry (obr. 1.5). Při výběru této šifry si předem musí na své tajné schůzce vyměnit tajný šifrovací klíč (1). Ten budou sdílet podobně jako sdílené tajemství. Nesmí dopustit, aby se k němu dostala třetí osoba (např. Cyril).



Obrázek 1.5: Symetrická šifra

Alois zprávu šifruje tajným klíčem. Na výsledek pak Bohumila aplikuje dešifrovací algoritmus, pro který použije týž tajný klíč. Dešifrování se vyruší s šifrováním a Bohumila získá původní zprávu.

Symetrická šifra má v jistém smyslu autorizační účinek. Z pohledu Bohumily mohl zprávu zašifrovat pouze Alois, protože přece nikdo jiný než ona a Alois nemají k dispozici tajný klíč.

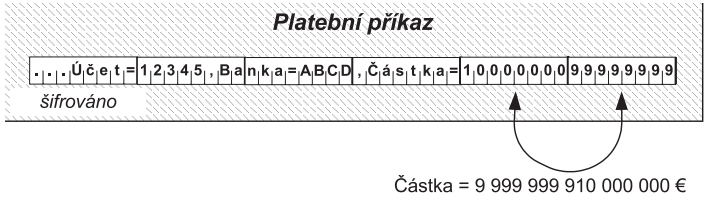
Symetrických šifrovacích algoritmů je celá řada. Snad nejrozšířenějším je algoritmus DES, používající šifrovací klíč délky 56 bitů. Dnes se však považuje za nedostatečný. Z algoritmu DES byl odvozen algoritmus 3DES s klíčem 112 bitů nebo 168 bitů*. Dále se používají algoritmy s délkou klíče 128 bitů (IDEA, RC2, RC4 atd.). Aktuálně doporučovaným algoritmem je však algoritmus AES s délkou klíče 128, 192 nebo 256 bitů.

* 112 = 2 x 56 a 168 = 3 x 56 (56 je délka šifrovacího klíče algoritmu DES)

Jedná se vesměs o blokové šifry. Tj. data se šifrují/dešifrují po blocích dlouhých zpravidla 8 B. Pokud jsou vstupující data kratší, musí se nějak dorovnat na 8 B. I když útočník nevidí do šifrovaného textu, mohl by hypoteticky útočit tak, že by zaměnil pořadí jednotlivých zašifrovaných bloků (obr. 1.6). Tomu se šifry brání vázáním po sobě následujících bloků. Hovoříme o tzv. módu šifry, který zahrnutím obsahu předchozího bloku do bloku následujícího zajišťuje, že nelze přehazovat jednotlivé šifrované bloky.

Velice zajímavou otázkou je, jakou informaci máme zahrnout do prvního šifrovaného bloku. Pokud bychom nezahrnovali nic, dva shodou okolností stejné texty by měly i shodné zašifrované texty. Útočník by sice nebyl schopen získat původní (nešifrovaný) text, ale informace, že se jedná o tutéž zprávu, pro něj také nemusí být k zahození. Proto se často před šifrováním zprávy vygenerují náhodná čísla (tzv. inicializační vektory), která se zahrnou do prvního šifrovaného bloku, pak nelze porovnáním dvou zašifrovaných textů získat informace o rozdílech mezi vstupními texty.

Často používanými módy jsou např. módy CBC (*Cipher block chaining mode*) či ECB (*Electronic codeblock mode*). Pokud chceme vyjádřit to, že máme na mysli šifrovací algoritmus s konkrétním módem, říkáme např. DES-CBC či DES-ECB nebo IDEA-CBC či IDEA-ECB atd.



Obrázek 1.6: Význam módu šifry (útočník ví, že 5. a 6. blok obsahuje částku)

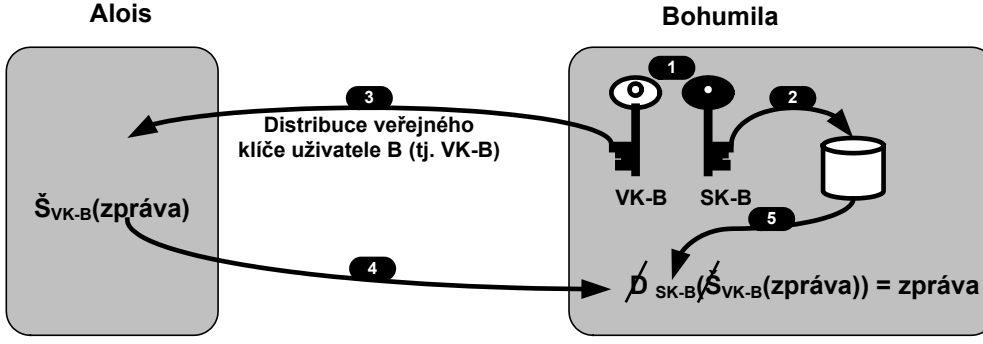
Asymetrické šifry

Jiným typem šifer jsou asymetrické šifry. Tyto šifry nepoužívají jeden tajný šifrovací klíč sdílený mezi odesilatelem a příjemcem, ale vždy se používá pár šifrovacích klíčů. Jeden klíč pro šifrování a druhý pro dešifrování. U digitálního podpisu pak uvedeme, že operace šifrování a dešifrování jsou u některých šifer zaměnitelné, proto u asymetrických šifer nemluvíme o šifrovacím a dešifrovacím klíči, ale o veřejném a soukromém klíči. Asi nejznámějším asymetrickým šifrovacím algoritmem je algoritmus RSA.

Pokud chce Alois šifrovat zprávu Bohumile asymetrickou šifrou, pak (obr. 1.7):

1. Bohumila, tj. příjemce zprávy, si musí vygenerovat dvojici klíčů: veřejný klíč (VK-B) a soukromý klíč (SK-B).
2. Bohumila si uloží svůj soukromý klíč do důvěryhodného úložiště klíčů. Např. na disk, na čipovou kartu atd. Soukromý klíč je aktivem Bohumily, které si musí střežit.
3. Bohumila distribuuje svůj veřejný klíč (VK-B) do celého světa. Klidně může svůj veřejný klíč poslat Aloisovi po žárlivém Cyrilovi.
4. Alois po obdržení veřejného klíče Bohumily šifruje zprávu Bohumile jejím veřejným klíčem (VK-B).
5. Bohumila (příjemce) dešifruje přijatou šifrovanou zprávu svým soukromým klíčem (SK-B) a získá původní zprávu.

Základní vlastností šifrování na bázi asymetrických algoritmů je skutečnost, že je relativně jednoduché za využití veřejného klíče šifrovat text, ale na základě znalosti veřejného klíče a veřejným klíčem šifrované zprávy je velice obtížné získat původní zprávu.



Obrázek 1.7: Asymetrická šifra

Délka šifrovacích klíčů pro algoritmus RSA se tč. považuje za ještě bezpečnou, pokud je alespoň 1 024 bitů. Často se však používají klíče dlouhé 2 048 nebo i 4 096 bitů (v závislosti na tom, jestli je protivníkem kolega na LAN, útočník z Internetu či NSA).

Existují i jiné asymetrické algoritmy. Dnes se často mluví o algoritmu ECC – *Elliptic Curve Cryptography* (eliptické křivky). Obecně se míní, že z bezpečnostního hlediska odpovídá 1 024 bitů dlouhému RSA klíči 160 bitů dlouhý ECC klíč, přičemž výpočetní náročnost je srovnatelná.

Jiným algoritmem je Diffie-Hellmanův (DH) algoritmus. Ten se vůbec nehodí k nějakému asymetrickému šifrování, ale k bezpečnému ustavení tajných klíčů či sdílených tajemství. Aby Alois s Bohumilou mohli komunikovat symetrickou šifrou, tak se nejprve za využití algoritmu DH dohodnou na tajném klíči, aniž by museli organizovat nějakou tajnou schůzku. Oba nejprve vygenerují dvojici: veřejné a soukromé DH číslo. Vzájemně si pak vymění (např. volně přes Internet) svá veřejná DH čísla. Obě strany jsou následně ze znalosti svého veřejného a soukromého DH čísla a veřejného DH čísla svého protějšku schopny spočíst sdílené tajemství. Od tohoto tajemství je pak snadno možné nějakou transformací odvodit symetrický šifrovací klíč, který se použije pro šifrování vzájemné komunikace např. algoritmem AES. Diffie-Hellmanův algoritmus hojně využívá např. IPsec.

Bez ohledu na délku klíčů obecně platí, že asymetrické šifrovací algoritmy jsou výpočetně mnohem náročnější než symetrické algoritmy.

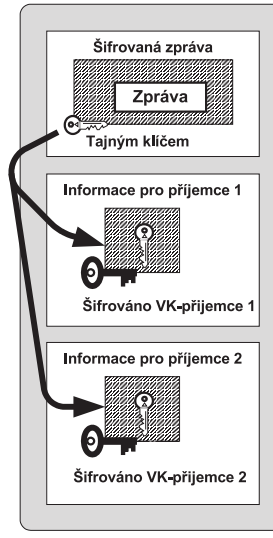
Elektronická obálka

Šifrování je vždy operací, do které vstupují data určená k zašifrování a šifrovací klíče (a někdy též inicializační vektory). V případě využití asymetrické kryptografie, kde se používají výpočetně náročné matematické postupy, je doba trvání výpočtu dlouhá.

Např. klíč RSA o délce 1 024 bitů se v desítkové soustavě zapíše jako číslo s více než 300 ciframi a nelze tak použít standardních operací mikroprocesoru.

Řešením tohoto problému je elektronická obálka (obr. 1.8). Odesilatel zašifruje zprávu náhodným tajným (symetrickým) klíčem, což je rychlá operace. A k takto šifrované zprávě jen přibalí

„informace pro příjemce“ (*Recipient info*) obsahující mj. náhodný tajný klíč zašifrovaný veřejným klíčem příjemce. Takže asymetricky se šifruje pouze kraťoučký tajný klíč. Výsledek je velice rychlý a efektivní. Má to ještě jeden pozitivní efekt. Pokud zprávu posíláme více adresátům, šifrujeme ji pouze jednou náhodným tajným klíčem a každému adresátovi ke zprávě přibalíme tajný klíč šifrovaný jeho veřejným klíčem. Tj. pokud náš Alois má kromě Bohumily ještě další milenky, může vyznání lásky rozeslat jen jednou, přičemž pouze pro každou z milenek šifruje tajný klíč veřejným klíčem odpovídající milenky.



Obrázek 1.8: Elektronická obálka

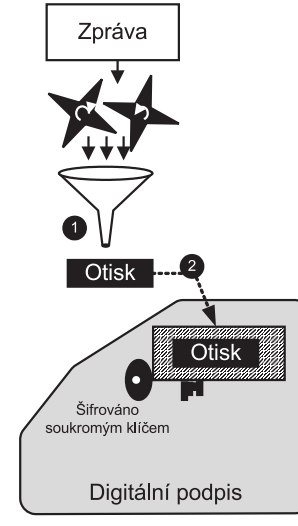
Digitální podpis

Digitální podpis je mechanismus, kterým se zajišťuje důkaz nepopiratelnosti dat (pravosti dokumentů). Digitální podpis se vytváří ve dvou krocích (obr. 1.9):

1. Spočte se otisk z dokumentu.
2. Výsledný otisk se šifruje soukromým klíčem uživatele, který podpis vytváří. Soukromým klíčem šifrovaný otisk ze zprávy se nazývá digitální podpis zprávy.

Na obr. 1.10 je pak znázorněno ověřování (verifikace) digitálního podpisu. To se provádí ve třech krocích:

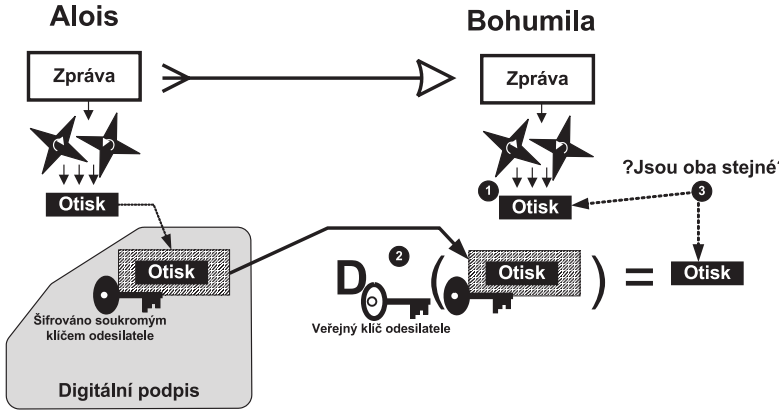
1. Příjemce samostatně spočte otisk z přijaté zprávy.
2. Příjemce dešifruje přijatý digitální podpis veřejným klíčem odesilatele.
3. Příjemce porovná výsledek získaný z bodu 1 s výsledkem získaným z bodu 2. Pokud jsou stejné, pak mohl digitální podpis vytvořit pouze ten, kdo vlastní soukromý klíč odesilatele – tedy odesilatel. A navíc tato skutečnost prokazuje, že zpráva nebyla během přenosu pozměněna, tj. zajišťuje i integritu zprávy.



Obrázek 1.9: Digitální podpis

Digitální podpis provádí důkaz pravosti na základě vlastnictví soukromého klíče. Je tedy nutné, abychom si své soukromé klíče dobře střežili. Ztráta soukromého klíče je pak obdobná výměně podobizny v občanském průkazu či záměně otisků prstů v evidenci zločinců. Neopatrnost ochrany soukromého klíče lze přirovnat k podepsání bianco šeků.

Na rozdíl od šifrování použije digitální podpis klíč odesilatele (nikoliv příjemce jako u šifrování). Mlčky jsme tedy předpokládali, že náš algoritmus umožňuje nejprve „dešifrovat“ soukromým klíčem a pak „šifrovat“ veřejným klíčem, tj. že operace šifrování a dešifrování jsou zaměnitelné. Algoritmem, který takovouto záměnu umožňuje, je právě algoritmus RSA.

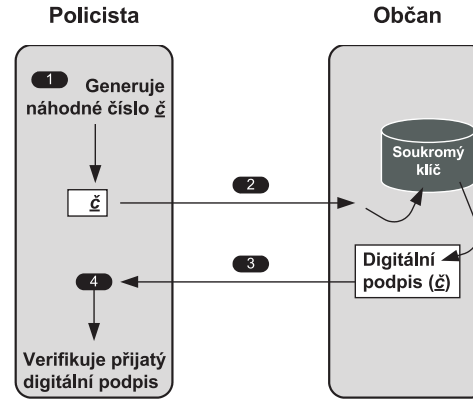


Obrázek 1.10: Verifikace digitálního podpisu

Prokazování totožnosti (autentizace) na základě asymetrické kryptografie

Digitální podpis může být využit jako nástroj pro ověření totožnosti (tj. k autentizaci) na základě prokázání vlastnictví soukromého klíče. Uživatel prokazuje svou totožnost tím, že dokáže, že vlastní příslušný soukromý klíč a je schopen jej použít. Problémem však je, jaký text má osoba k prokázání své totožnosti podepsat. Pokud by totiž podepisovala stále stejný text, pak by byl opět možný *reply attack*.

Autentizace na základě asymetrické kryptografie je proto vždy nějakou variací na situaci znázorněnou na obr. 1.11. Představme si, že policista chce, aby mu občan prokázal svou totožnost na základě asymetrické kryptografie.



Obrázek 1.11: Autentizace na bázi asymetrické kryptografie

V klasickém případě občan prokazuje svou totožnost na základě občanského průkazu, který předloží policistovi. Policista v klasickém občanském průkazu ověřuje totožnost na základě občanovy fotografie. V elektronickém případě pak občan prokazuje svou totožnost na základě vlastnictví svého soukromého klíče.

Princip prokazování totožnosti na základě asymetrické kryptografie je jednoduchý. Policista vygeneruje dostatečně dlouhé náhodné číslo ***č***. Toto číslo ***č*** předá občanovi, který jej za pomoci svého soukromého klíče digitálně podepíše. Digitálně podepsané číslo ***č*** předá občan policistovi, který provede verifikaci digitálního podpisu.

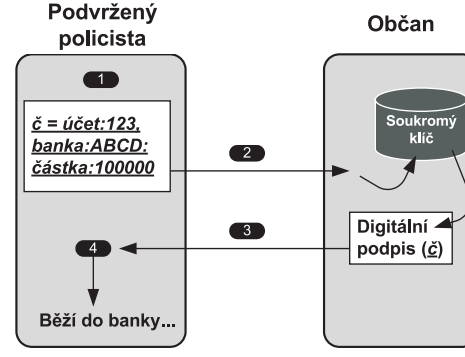
V případě, že nechceme využít digitální podpis, ale výhradně šifrování, pak policista náhodné číslo utají a občanovi jej zašle šifrované jeho veřejným klíčem, občan jej dešifruje soukromým klíčem a dešifrované vrátí policistovi. Policista zkontroluje, rovná-li se přijaté číslo tomu, které šifroval veřejným klíčem občana.

Pro občana se tak základem stává ochrana jeho soukromého klíče, neboť soukromý klíč je jeho cenným aktivem. Odcizení soukromého klíče by způsobilo to, že by se zloděj mohl elektronicky prokazovat místo majitele soukromého klíče. Opět připomeňme, že odcizení soukromého klíče lze přirovnat v případě klasických občanských průkazů k odcizení podoby z fotografie občanského průkazu.

Využíváme-li stejný pár veřejný/soukromý klíč k digitálnímu podpisu i k prokazování totožnosti na základě digitálního podpisu, koledujeme si o problém. Podvržený policista (obr. 1.12) totiž negeneruje náhodné číslo ***č***, ale místo něj řetězec obsahující například platební příkaz v neprospěch občana. Za autentizaci občanovi poděkuje a obratem uplatní platební příkaz v neprospěch občana.

Digitální podpis jako algoritmus je tak využíván jednak pro autentizaci uživatele a jednak pro digitální podpis dokumentů jako indicie pravosti dokumentu.

Někteří autoři pak rozlišují termín „digitální podpis" jako algoritmus (bez ohledu na to, je-li využit k podpisu či autentizaci) a termín „digitální podpis" jako důkaz pravosti dokumentu. Nepostřehneme-li, v jakém smyslu je termín „digitální podpis" v daném okamžiku použit, pak může dojít k docela nepříjemným nedorozuměním.



Obrázek 1.12: Zneužití autentizace k vylákání digitálního podpisu z nechtěného dokumentu

Tři typy asymetrických klíčů

V předchozím paragrafu jsme obhajovali nutnost samostatných párů soukromý/veřejný klíč pro:

- Digitální podpis dokumentů
- Autentizaci uživatele

Nicméně potřebujeme ještě další pár:

- Pro šifrování (resp. elektronickou obálku)

Proč potřebujeme třetí pár pro šifrování? Přinejmenším máme dva pádné důvody:

- První důvod je kryptografický. Dobrou zprávou pro hackera lámajícího šifru totiž je, že má k dispozici známý text šifrovaný soukromým klíčem (digitální podpis) a jiný známý text šifrovaný veřejným klíčem (např. náhodný symetrický klíč v elektronické obálce).
- Druhým, podle našeho názoru pádnějším, důvodem je praktické používání soukromého klíče. Jestliže uživatel ztratí soukromý klíč (nikoliv vyzradí) určený k digitálnímu podpisu nebo k autentizaci, pak se vcelku nic neděje. Soukromý klíč je totiž třeba pouze pro vytváření nového podpisu (existující podpisy se verifikují pomocí veřejného klíče). Uživatel si v takovém případě vygeneruje nový pár klíčů a může podepisovat další dokumenty. V případě ztráty soukromého klíče určeného pro šifrování (přesněji pro dešifrování) ztratíme veškeré tímto klíčem zabezpečené dokumenty, protože je nemáme čím dešifrovat. Soukromý klíč určený k vytváření podpisu

zpravidla uchováváme na nosičích, které soukromý klíč nikdy nemůže opustit (např. na čipové kartě). Nikdy takový klíč nedáváme z ruky, protože by jím mohly být podepisovány dokumenty v náš neprospěch. Naopak šifrovací klíč je mnohdy dobré zálohovat, abychom jej měli i v případě ztráty primárního úložiště soukromého klíče. Jelikož mají šifrovací klíče jiný životní cyklus než podepisovací/autentizační klíče, je praktické mít samostatný pár šifrovacích/dešifrovacích klíčů.

Elektronický podpis, digitální podpis a kvalifikovaný podpis

Setkali jsme se s dvěma termíny: elektronický podpis a digitální podpis. V této publikaci budeme pod pojmem elektronický podpis chápat veškeré elektronicky vytvořené důkazy o tom, že dokument byl vytvořen konkrétní osobou nebo konkrétním systémem. Jedná se tedy o důkaz autenticity dokumentu. Takovými důkazy může být zmíněný MAC, podpis vytvořený autentizačním kalkulátorem, ale i digitální podpis.

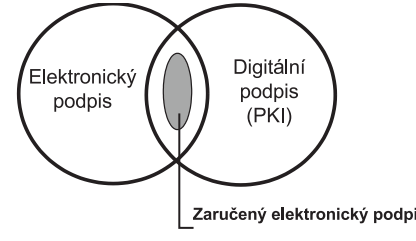
Digitálním podpisem budeme rozumět podpis vytvořený na základě asymetrické kryptografie, tak jak je popsán v této kapitole.

Jelikož digitální podpis může sloužit jako důkaz pravosti dokumentu („nepopíratelnosti" – *non repudiation*), za jistých podmínek jej můžeme využít jako plnohodnotnou náhradu rukou psaného podpisu. Takový podpis pak označujeme jako zaručený elektronický podpis.

Podmínky, za kterých vytváříme zaručený elektronický podpis, jsou dány nejenom kryptografickými parametry a organizačními opatřeními spojenými s bezpečnou generací a správou páru klíčů, ale zejména legislativními podmínkami státu, ve kterém chceme příslušný zaručený podpis uplatnit.

Je třeba podotknout, že na elektronický podpis existují v různých zemích různé pohledy:

- V některých zemích je elektronický podpis chápán jako plnohodnotná náhrada rukou psaného podpisu. V těchto zemích je pak v právním řádu zaváděn zaručený elektronický podpis.
- V jiných zemích je digitální podpis chápán výhradně jen k autentizaci dokumentu, nikoliv jako plnohodnotná náhrada rukou psaného podpisu.



Obrázek 1.13: Zaručený elektronický podpis

V členských zemích EU je problematika náhrady elektronického podpisu za rukou psaný podpis řešena pomocí *SMĚRNICE EVROPSKÉHO PARLAMENTU A RADY 1999/93/ES* ze dne 13. prosince 1999, která byla např. do české legislativy zapracována zákonem „O elektronickém podpisu" č. 227/2000 Sb. Tento zákon byl později novelizován zákony 226/2002 Sb., 517/2002 Sb., 440/2004 Sb., 635/2004 Sb., 501/2004 Sb., 110/2007 Sb. a 124/2008 Sb.

Zaručeným elektronickým podpisem se míní elektronický podpis, který splňuje následující požadavky:

1. je jednoznačně spojen s podepisující osobou,
2. umožňuje identifikaci podepisující osoby ve vztahu k datové zprávě,

3. byl vytvořen a připojen k datové zprávě pomocí prostředků, které podepisující osoba může udržet pod svou výhradní kontrolou,
4. je k datové zprávě, ke které se vztahuje, připojen takovým způsobem, že je možno zjistit jakoukoliv následnou změnu dat.

Blíže se této problematice budeme věnovat v kap. 10.

Autentizační metody založené na jiných principech

Mnohdy se autentizace (prokazování totožnosti) spojuje s přihlášením uživatele k počítačovému systému. Klasická autentizace v počítačovém světě byla orientována na autentizaci pomocí jména a stálého hesla. V dnešní době je problém autentizace uživatele aktuální pro internetové aplikace, které používá široká veřejnost. A tak pro volbu autentizace začínají platit i kritéria, která byla dříve spíše v pozadí. Jedná se např. o cenu autentizačních pomůcek či pracnost autentizace pro klienta. Stačí se nad problémem zamyslet prakticky. Je rozdíl v tom, nakupuje-li firma autentizační kalkulátory v ceně několika set Kč pro padesát zaměstnanců nebo pro padesát tisíc klientů.

Na pracnost autentizace je možno se dívat ze dvou pohledů: z pohledu pracnosti a z pohledu nároků na znalosti nutné k provádění autentizace. Mechanická pracnost vstupuje do hry např. při použití autentizačních kalkulátorů, kdy klient musí do kalkulátoru a z kalkulátoru přepisovat data. Při použití digitálního podpisu je autentizace pro uživatele mechanicky jednoduchá, avšak uživatel musí pochopit princip digitálního podpisu, musí obnovovat certifikáty atd.

Autentizaci uživatele je možné provést na základě prokázání:

- **že uživatel něco má** (autentizační kalkulátor, čipovou kartu či v poslední době mobilní telefon);
- **že uživatel něco ví** (heslo, PIN);
- **že uživatel něčím je** – má např. nějaké biometrické vlastnosti (otisky prstů, struktury oční sítnice či duhovky, tvar obličeje atp.);
- **že uživatel něco umí** (podepsat se).

Snahou je jednotlivé uvedené metody kombinovat, pak hovoříme o vícefaktorové autentizaci.

Vedle autentizace (prokazování totožnosti) budeme používat ještě termín autorizace. Zatímco autentizací prokážeme, o koho se jedná, autorizací budeme konkrétnímu subjektu přiřazovat role a z nich plynoucí oprávnění, která má v jednotlivých aplikacích. Např. uživateli Fr. Novákovi, poté co prokáže svou totožnost (autentizuje se), je v aplikaci XY poskytnuta role administrátora. Tj. F. Novák je pro aplikaci XY autorizován jako administrátor.

V PKI se pro autentizaci využije certifikát veřejného klíče a pro autorizaci pak atributové certifikáty. Tato kapitola je však věnována jiným autentizačním metodám než certifikátům, aby čtenář získal pokud možno objektivní porovnání jednotlivých autentizačních metod.

Stálá hesla

Přístupové heslo je typickým příkladem stálého hesla. Na straně serveru nebývá uchováváno v čisté textové podobě, ale znehodnocené jednocestnou funkcí proti zneužití správcem systému.

V okamžiku, kdy uživatel zadá heslo, aby se autentizoval, systém zavolá na zadané heslo jednocestnou funkci a výsledek se porovná s údajem uloženým v systému. Algoritmů jednocestných funkcí je celá řada. Nejčastěji jsou založeny buď na výpočtu otisku nebo na symetrické šifře.

Stálé heslo může být:

- Odposlechnuto v případě, že je přenášeno po nezabezpečených spojích.
- Vylákáno pomocí podvrženého serveru (např. i v relacích zabezpečených pomocí SSL/TLS).

Jednorázová hesla

Jednorázová hesla řeší problém odposlechu hesla během jeho přenosu sítí a následným použitím odposlechnutého hesla. Pokud je jednorázové heslo využito pouze pro počáteční autentizaci, pak ještě po úspěšně proběhlé autentizaci existuje nebezpečí v převzetí relace útočníkem. Proto se jednorázová hesla zpravidla ještě následně využívají pro další zabezpečení relace (např. pomocí doplňování MAC k blokům přenášených dat či jako součást symetrických klíčů pro šifrování relace).

Jednorázová hesla se nepoužívají pouze u aplikací provozovaných v počítačových sítích, ale i u tak odlišných aplikací, jako je CallCentrum, kdy je nutné autentizovat uživatele, který požaduje služby běžným telefonem.

Jak je vlastně možné, že uživatel může pokaždé zadat jiné heslo? Algoritmů na tvorbu jednorázových hesel je celá řada.

Seznam jednorázových hesel

Nejjednodušší metodou jednorázových hesel je seznam jednorázových hesel. V tomto případě je vygenerován seznam hesel, který může být vytištěn na papír a předán uživateli (resp. zaslán uživateli bezpečnou elektronickou poštou). Stejný seznam existuje i na straně systému, kde mohou být i jednotlivá jednorázová hesla znehodnocena jednocestnou funkcí.

Uživatel pak pro svou autentizaci zadává jedno heslo po druhém. Po zadání hesla si jej škrtne ze seznamu.

Jednorázová hesla mohou být v seznamu i očíslována. Systém pak může ve výzvě pro zadání hesla napovědět uživateli, jaké heslo má zadat.

Jednou z nevýhod tohoto způsobu je, že po vyčerpání seznamu musí být uživateli vygenerován a předán další seznam. Další nevýhodou seznamu hesel je, že si jej uživatel těžko může zapamatovat, a tak jej musí nosit s sebou v tištěné či elektronické podobě. Může se tak snadno stát, že uživatel seznam jednorázových hesel někde zapomene.

Seznamy jednorázových hesel se často kombinují s klasickým heslem. Uživatel pak zadává heslo skládající se ze dvou částí: ze stálého hesla („PIN“) a z jednorázového hesla. Tím se komplikuje využití seznamu jednorázových hesel zapomenutého v internetové kavárně a na druhou stranu se i komplikuje použití odposlechnutého hesla.

Jinou možností používání jednorázových hesel je jednotlivá hesla očíslovat a nevyžadovat hesla jedno po druhém, ale náhodně. Náhodný výběr hesel může být i s opakováním, tj. pak se v podstatě nejedná o „jednorázové heslo“, ale požadavek na výběr dvou stejných hesel v čase zajímavém pro útočníka je málo pravděpodobný.

Pro některé aplikace je dostatečná i autentizační karta. Jedná se o plastikovou kartičku bez magnetického proužku a bez čipu s několika předtištěnými sadami čísel (obr. 1.14).

Princip použití spočívá v tom, že aplikace vygeneruje dotaz na zadání několika náhodně vybraných čísel vytištěných na kartě. Např. v podobě „zadejte třetí čtveřici čísel ze čtvrté sady vytištěných na vaší Autentizační kartě".

Autentizační karta je forma použití vícenásobných hesel s jednoduchým doplňkem vzdáleně připomínajícím heslo na jedno použití.

Výhodou tohoto prostředku jsou jeho zanedbatelné pořizovací náklady, kterými jsou získány velmi zajímavé bezpečnostní vlastnosti.

Obrázek 1.14: Autentizační karta

Rekurentní algoritmus

Rekurentní algoritmus využívá jednocestné funkce (např. otisk). Zvolme si konkrétní jednocestnou funkci, kterou označíme jako F. Dále si uživatel musí sám zvolit nějaký počáteční řetězec *násada*. Tento řetězec uživatel nikomu nesděluje – je to uživatelovo tajemství.

Jednocestnou funkci F aplikovanou na řetězce *násada* vyjádříme jako:

F(*násada*).

Použijeme-li algoritmus F dvakrát opakovaně na tutéž zprávu, tj. F(F(*násada*)), pak budeme psát:

F_2(*násada*)

A obdobně:

F_n(*násada*)

bude znamenat, že jsme použili algoritmus F na řetězec *násada* celkem n-krát.

Použití této metody spočívá též nejprve v inicializačním kroku. Uživatel si pořídí text *násada*.

V inicializačním kroku se uživatel a správce aplikace dohodnou na čísle *n*, např. 1 000. Uživatel vyrobí: F_{1000}(*násada*) a předá jej správci aplikace. Správce aplikace si do databáze k našemu uživateli poznamená název algoritmu jednocestné funkce (tj. F), číslo 1 000 a hodnotu F_{1000}(*násada*). Správce však nezná hodnotu řetězce *násada* (je to uživatelovo tajemství).

Při autentizaci pošle uživatel na server jméno, server ve své databázi zjistí, jakou uživatel používá autentizační metodu (F). Obratem server uživateli pošle dotaz obsahující číslo (*n-1*), tj. nyní 999. Uživatel vygeneruje odpověď F_{999}(*násada*) a odešle ji jako jednorázové heslo serveru. Server prověří totožnost uživatele tím, že provede porovnání:

$F(F_{999}(násada)) = F_{1000}(násada)$

Algoritmus F je mu znám, hodnotu F_{1000}(*násada*) má uloženu v konfiguračním souboru a hodnotu F_{999}(*násada*) obdržel v odpovědi uživatele.

Po úspěšné autentizaci uživatele uloží server do databáze místo hodnoty F_{1000}(*násada*) hodnotu F_{999}(*násada*) a místo čísla 1 000 číslo 999. Při další autentizaci se vše provádí s číslem o jedničku nižším, tj. provádí se autentizace:

$F(F_{998}(\textit{násada})) = F_{999}(\textit{násada})$

Uživatel mohl tedy vygenerovat celkem 999 hesel na jedno použití, pak musí změnit hodnotu řetězce *násada* a správci serveru předat nový $F_{1000}(\textit{násada})$.

S/KEY

Algoritmus S/KEY je implementací rekurentního algoritmu. S/KEY je popsán v RFC-1760. Jádrem je použití algoritmu pro výpočet otisku MD4 (MD4 je popsán v RFC-1320).

Násadu si klient volí sám tak, aby byla dlouhá minimálně 8 bajtů.

Algoritmus MD4 produkuje 16 bajtů dlouhý otisk. Ten se v tomto případě dělí na dvě poloviny po 8 bajtech, které se spojí operací XOR do výsledných osmi bajtů. Použijeme-li terminologii z předchozího paragrafu, pak algoritmem F je algoritmus MD4, jehož výsledek se dělí na dvě poloviny. Ty, které jsou operací XOR sloučeny do výsledných osmi bajtů.

S/KEY má nápaditě rozšíření umožňující použít stejný algoritmus (včetně stejné násady) pro více aplikací (např. pro více serverů). Princip spočívá v tom, že aplikace (server) vyzývající uživatele k prokázání své totožnosti (k autentizaci) klientovi zobrazí tři údaje:

- Informaci, že se používá algoritmus S/KEY.
- Číslo *n*, kolikrát má uživatel aplikovat algoritmus F.
- Sůl, což je řetězec vygenerovaný serverem a zasílaný uživateli nezabezpečeně jako součást výzvy. Právě solí se budou výzvy jednotlivých aplikací lišit.

Uživatel nejprve spojí násadu se solí. Výsledný řetězec teprve použije jako násadu pro algoritmus F.

OTP (One Time Password)

OTP je popsáno v RFC-1938, které rozšiřuje mechanismus S/KEY o možnost použití dalších algoritmů pro výpočet otisku, jako jsou např. algoritmy MD5 (popsaný v RFC-1321) a SHA-1.

Sdílené tajemství

Na obr. 1.4 je znázorněn princip autentizace za využití sdíleného tajemství. Jedná se o důkaz pravosti dokumentu. V tomto případě je využit MAC (*Message Authentication Code*). Pokud chceme MAC využít nikoliv pro autorizaci dokumentu, ale pro autorizaci osoby, pak základním problémem bude, s jakými daty řetězit sdílené tajemství, tj. z čeho počítat otisk.

Cílem je tedy generovat jednorázová hesla, jež budou spočtena jako MAC. Vlastně jsme v absurdní situaci. Víme, jaký má být výsledek, víme, že jej budeme počítat z něčeho, co budeme řetězit se sdíleným tajemstvím, ale nevíme z čeho. Navíc by byla nepříjemná pravděpodobnost, že generovaná hesla se budou opakovat.

Musíme tedy najít něco, co zná jak autentizovaný (klient), tak i server, který bude autentizaci verifikovat. Takovými veličinami jsou např.:

- Datum a čas. Stačí si představit digitální hodiny s minutovou přesností. Pokud se na nich zobrazuje datum a čas, pak se tato hodnota nikdy neopakuje a platí nejvýše jednu minutu. Drobnou závadou je časová odchylka hodin uživatele a serveru.

- Počet přihlášení uživatele k systému je rovněž stále monotónně vzrůstající posloupností. Drobnou potíží jsou stavy, kdy se klientovi přeruší komunikace během autentizace.
- Náhodné číslo generované serverem. V podstatě se jedná o symetrickou obdobu obr. 1.12. Jedná se o dialog dotaz-odpověď (*challange-response*):
  - Server generuje náhodné číslo a odešle jej klientovi jako dotaz.
  - Klient zřetězí dotaz se sdíleným tajemstvím a na výsledek aplikuje jednocestnou funkci (např. pro výpočet otisku). Výsledkem je jednorázové heslo, které klient vrátí jako odpověď serveru. Klient prokazuje svou totožnost serveru pomocí tohoto jednorázového hesla. Server má k dispozici veškeré údaje pro verifikaci tohoto jednorázového hesla: dotaz, sdílené tajemství a algoritmus pro jednocestnou funkci. Princip sdíleného tajemství často využívají autentizační kalkulátory (viz Autentizační kalkulátory).

Symetrická šifra

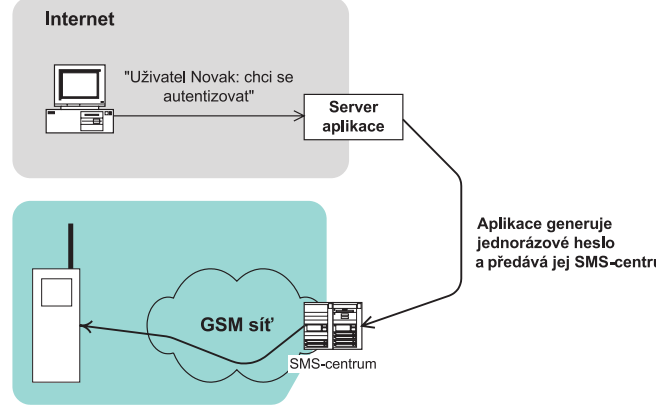
Ke generování jednorázových hesel můžeme místo otisku využít též symetrickou šifru. Namísto sdíleného tajemství sdílí klient se serverem symetrický šifrovací klíč, který využije k šifrování: času, počtu přihlášení či náhodně generovaného dotazu.

Jednorázové heslo doručované přes nezávislý kanál

Principem této metody je, že dotaz generuje server, který jej uživateli doručí nezávislým kanálem. Druhým kanálem může být fax, e-mail, mobil apod. Každý z použitých kanálů přitom nemusí být příliš bezpečný, útočník by ale musel prolomit dva na sobě nezávislé komunikační kanály současně (v jednom okamžiku), což je velice těžko uskutečnitelné. Tento princip se také označuje jako princip dvou zámků. Předpokládejme, že druhým kanálem je mobil.

Nevýhodou autentizačních kalkulátorů na výrobu jednorázových hesel je totiž samotná existence kalkulátoru, tj. z hlediska provozovatele aplikace se kalkulátor musí pořídit, což není laciné. Z hlediska uživatele je zase nepříjemné, že se kalkulátor musí stále nosit s sebou, a přitom je dobré jej nezničit či neztratit. Naopak mobilní telefon má dnes téměř každý. Uživatelé jsou zvyklí jej s sebou nosit a starat se o něj a navíc si mobilní telefon pořizuje uživatel sám.

Přímé nahrazení autentizačních kalkulátorů mobily je málo běžné (existují např. emulátory autentizačních kalkulátorů jako Java aplikace v mobilu). Nabízí se ale jiné použití. V okamžiku, kdy se má klient autentizovat, oznámí aplikaci své přihlašovací jméno. Aplikace vygeneruje jednorázové heslo a v databázi uživatelů najde číslo jeho mobilního telefonu, na které pomocí SMS-zprávy jednorázové heslo zašle (obr. 1.15).



Obrázek 1.15: Jednorázové heslo zasílané přes mobil

Tento způsob autentizace kombinuje dva nezávislé komunikační kanály. Tato skutečnost podstatným způsobem omezuje možnost zneužití, protože případný útok by musel být veden společně na oba nezávislé kanály, což je vysoce náročné. Další podstatnou výhodou jsou nízké pořizovací náklady a relativně jednoduchá obsluha. Jistým omezením tohoto řešení je, že klient musí být vybaven mobilním telefonem a že tento pracuje pouze v místě, kde má dostupný signál.

Biometrika

Biometrických vlastností člověka je možné měřit celou řadu. Ekonomicky nejpřijatelnější jsou zatím stále otisky prstů. Navíc data popisující otisk prstů lze omezit na 300 až 600 bajtů. Nevýhodou otisků prstů je skutečnost, že se jedná o osobní data, a musí tak s nimi být i zacházeno. Navíc se v komerční praxi používá řada formátů dat popisujících otisky prstů, a tak zařízení různých výrobců nemusí být vzájemně kompatibilní.

Na problematiku však nahlížíme z pohledu sítě. A z tohoto pohledu není ani tak zajímavé, že je možné snímat i otisk z useknutého prstu, ale fakt, že mezi otiskem prstu a stálým heslem není z našeho pohledu příliš velký rozdíl. Snad jen v tom, že běžné heslo má řádově jednotky znaků a otisk prstů až 1 000 bajtů. Jelikož se otisk nemění, bylo by jej možné na síti odchytit a následně zneužít.

Význam biometrických vlastností je spíše v umožnění přístupu k lokálním zařízením: k otevření dveří, k přístupu k PC apod. A právě kombinace přístupu pomocí otisků prstů k čipové kartě a následné využití čipové karty je již špičkovou technologií. Pomocí otisků prstů a PIN otevřeme přístup k soukromému klíči na čipové kartě a následně využijeme soukromého klíče na této kartě pomocí asymetrické kryptografie k autentizaci.

Shamirův algoritmus

Nyní z trochu jiného soudku. Na základní škole jsme se učili, že české korunovační klenoty jsou zajištěny několika zámky. Pro přístup ke klenotům je pak nutná přítomnost všech držitelů příslušných klíčů i s jejich klíči.

Shamirův algoritmus je určen pro ochranu nějakého aktiva (šifrovacího klíče, sdíleného tajemství) tak, aby pro získání tohoto aktiva bylo nutné sestavit tajemství, jehož jednotlivé části jsou distribuovány mezi n uživatelů. Rozdíl oproti přístupu ke korunovačním klenotům spočívá v tom, že pro sestavení celého tajemství se nemusí sejít všech n uživatelů, ale stačí jen libovolných k z nich ($0 < k \leq n$).

Prakticky to znamená, že části sdíleného tajemství či šifrovacího klíče uložíme např. na n čipových karet, které rozdáme n různým držitelům. Pokud potřebujeme tajemství rekonstruovat, musí se sejít alespoň k držitelů se svými kartami.

Kapitola 2

Prostředky pro bezpečné ukládání aktiv

Soukromé klíče, sdílená tajemství či jiný kryptografický materiál tvoří aktiva, která je třeba proti případným hrozbám chránit odpovídajícími protiopatřeními. Nejběžnějším typem ochrany těchto aktiv je jejich uložení do hardwarových klíčů.

Uložení aktiv na disk

Před tím, než se budeme věnovat hardwarovým klíčům, si připomeneme, že ukládání aktiv na lokální disk je nejjednodušší metodou uložení aktiv. Nevýhodou ale je, že data lokálního disku lze poměrně snadno zcizit. Což není až takové riziko v prostředí kontrolovaném uživatelem. Nebezpečí ale stále spočívá v aplikacích typu „trojský kůň", které mohou být schopny zjistit přístupové heslo k aktivu nebo přečíst přímo rozšifrovanou podobu aktiva ve chvíli, kdy je v paměti počítače používáno. Takové trojské koně mohou být staženy např. z Internetu nebo získány elektronickou poštou.

V sítích Windows je na disku udržovaný soukromý klíč součástí tzv. uživatelského profilu. Uživatelské profily jsou často konfigurovány tak, aby „cestovaly za uživatelem". Např. v případě tzv. *roaming profile* se uživatelský profil natáhne na každý počítač, ze kterého se uživatel kdy přihlásil do domény Windows (!). Dalších komentářů už asi netřeba.

Velké nebezpečí hrozí také v případě, kdy je disk vyjmut z řízeného prostředí, ve kterém je používán, a čten/zapisován v prostředí jiném (např. disk se systémem souborů NTFS je čten z prostředí Linuxu, kdy nejsou respektována přístupová oprávnění).

Jiným způsobem útoku je modifikace aktiva na lokálním disku a řada dalších.

Samostatnou kapitolou je pak ochrana aktiv uložených na lokálních discích proti počítačovým správcům. Tady si může být uživatel jist, pouze pokud svá aktiva uloží mimo pevný disk – např. na čipovou kartu. Čipová karta zajišťuje přístup k aktivům pouze držitelům karty. Host Security Moduly (HSM moduly) pak zamezují přístup k aktivům i správcům. Aktiva uložená v HSM modulu jsou totiž dostupná výhradně bezpečnostním manažerům. HSM modul může totiž být konfigurován tak, aby bez jejich přítomnosti neprovedl žádnou bezpečnostně citlivou operaci.

Autentizační kalkulátory

Autentizačními kalkulátory rozumíme samostatné technické zařízení přímo nepropojené s počítačem, které slouží pro generování jednorázových hesel pro autentizaci držitele kalkulátoru nebo autentizaci dat zasílaných držitelem kalkulátoru.

Autentizační kalkulátory jsou tak elektronické pomůcky pro autentizaci klienta (případně pro autentizaci dat zadaných klientem). Autentizační kalkulátory zpravidla umí některý z kvalitních algoritmů pro výpočet otisku (méně často symetrický šifrovací algoritmus). Do autentizačních kalkulátorů se ukládá sdílené tajemství. Autentizační kalkulátor umí zřetězit zadaná data se sdíleným tajemstvím a z výsledku spočíst jednorázové heslo (viz též obr. 1.4). Pro tvorbu jednorázových hesel se pak jako vstupní data používá např. čas či počet dosud vygenerovaných jednorázových hesel (viz též Sdílené tajemství).

Uživatel obdrží od správce aplikace autentizační kalkulátor, avšak nejdřív je třeba autentizační kalkulátor připravit (tj. je třeba provést management autentizačních kalkulátorů). Příprava spočívá např. v tom, že se do kalkulátoru vloží sdílené tajemství, které se nazývá násada. Sdílené tajemství je řetězec, který bude uložen v kalkulátoru a na serveru aplikace (nikdo jiný toto sdílené tajemství mezi klientem a aplikací nezná). V databázi na serveru tak musí být pro každého uživatele udržována informace obsahující mj. identifikaci uživatele, sdílené tajemství a postupy, jak se používá.

Rozlišujeme tak kalkulátory:

- Určené pouze pro autentizaci klienta (často nemívají ani klávesnici – např. zobrazují stále se měnící jednorázová hesla v závislosti na změně času či počtu dosud vygenerovaných jednorázových hesel).
- Určené též pro autorizaci dat zadávaných uživatelem (pak mají klávesnici na pořízení autentizovaných dat). Cílem těchto kalkulátorů je vytvořit MAC.

Kalkulátory mnohdy využívají patentované jednocestné funkce – např. funkce pro výpočet otisku apod. Důvodem, proč se namísto všeobecně známých jednocestných funkcí využívají patentované algoritmy, je skutečnost, že vytvořené jednorázové heslo (resp. MAC) musí být uživatelem přepsáno z kalkulátoru do počítače, nemůže být tedy příliš dlouhé.

Detailní popis použitého algoritmu nebývá u autentizačních kalkulátorů zveřejňován – výrobci kalkulátorů jej často považují za své vlastnictví. Dodavatel kalkulátorů zpravidla dodává nejen samotné kalkulátory, ale i software pro autentizační server, který je volán aplikací v případě autentizace.

Hardwarové klíče

Hardwarovým klíčem se nazývá technické zařízení, které poskytuje bezpečnostní funkce spojené s ukládáním soukromých klíčů, tajných klíčů, sdílených tajemství a jiných aktiv držitele hardwarového klíče. Na rozdíl od autentizačních kalkulátorů je hardwarový klíč propojen s počítačem, který je vybaven příslušným rozhraním. Takovým rozhraním může být: sériový port, USB, SCSII, PCI, PCMCIA apod.

Pro generování a přechovávání párových dat (asymetrická kryptografie) jsou hardwarové klíče často uzpůsobeny tak, že soukromý klíč nikdy neopouští hardwarový klíč. Hardwarový klíč tedy zajišťuje následující funkce:

- generuje dvojici veřejný/soukromý klíč
- generuje podklady pro žádost o certifikát
- vydaný certifikát lze uložit opět do hardwarového klíče
- v případě použití soukromého klíče aplikace vyšle data do hardwarového klíče a hardwarový klíč provede šifrování soukromým klíčem uloženým v hardwarovém klíči

Hardwarové klíče je možno používat pouze v prostředí kontrolovaném samotným uživatelem (např. uživatelův osobní počítač, který se opravdu provozuje jako *osobní* počítač) nebo v prostředí kontrolovaném správcem aplikace (např. bankomat). Použití hardwarových klíčů v prostředí kontrolovaném třetí stranou se považuje za nebezpečné. Útok v prostředí kontrolovaném třetí stranou je jednoduchý: třetí strana na svém počítači podvrhne software, který se navenek tváří jako aplikace uživatelem běžně používaná. Uživatel předá této falešné aplikaci data, která má aplikace zabezpečit za využití hardwarového klíče. Podvržený software však hardwarovému klíči nepředá k zabezpečení původní informace, ale informace změněné ve prospěch útočníka. Hardwarový klíč tyto údaje zpracuje, jako by je zadal uživatel.

Čipové karty

Nejčastějším druhem hardwarového klíče je **čipová karta**. Čipová karta je plastiková karta, která má ve svém těle vložen čip. Nejčastější technologií vložení čipu do karty je vyfrézování dutiny v kartě o rozměru čipu a následné vlepení čipu do dutiny. V případě bezkontaktních čipových karet se čip zalévá včetně antény přímo do karty.

V současnosti se většinou využívají karty dle **ISO 7816-1**. Jedná se o dva rozměry karty. S oběma se běžně setkáváme. Velký rozměr mají platební karty a malý rozměr SIM-karty mobilních telefonů. Přitom rozměr kontaktů je shodný. Malá karta tak vznikne jakoby vyříznutím z velké. Existují proto i plastikové redukce, do kterých lze vmáčknout malou kartu, a vznikne velká karta.

Nově se v blízké budoucnosti budeme setkávat s bezkontaktními čipovými „kartami" ve formě elektronické části nově zaváděných cestovních dokladů (e-pasů). Zde bude využito bezkontaktního čipu pro uložení biometrických údajů o držiteli pasu. Tento čip je v e-pasu zabudován buď ve speciální plastové stránce nebo v deskách e-pasu.

Kontaktní čipové karty mají dle ISO 7816-2 osm kontaktů. Tento standard totiž předepisuje osm plošek C1 až C8 o rozměru 2 x 1,7 mm, kde kontakty musí být umístěny (viz obr. 2.1). Výrobci pak vytváří kontakty o trochu větším rozměru tak, aby tvořily módní design.

ISO 14443 – standard pro bezkontaktní karty na frekvenci 13,56 MHz pracující do vzdálenosti 10 cm. Tento komunikační standard pro komunikaci se čtečkami využívají karty MIFARE®.

ISO 15693 – standard pro bezkontaktní karty na frekvenci 13,56 MHz pracující do vzdálenosti 1 m.

ISO 7816 – standard pro kontaktní karty rozšiřující se v lecčems i na bezkontaktní karty. Tento standard se skládá z následujících částí:

- **ISO 7816-1** specifikuje fyzikální charakteristiky karty (tepelnou odolnost, ohebnost karty, odolnost proti rentgenovému záření, UV záření, elektromagnetickému poli, minimální počet zasunutí karty do čtečky apod.).
- **ISO 7816-2** specifikuje umístění kontaktů na kartě, jejich rozměr a funkci.
- **ISO 7816-3** specifikuje elektrické signály a přenosové protokoly. Specifikuje již zmíněné protokoly T = 0, T = 1 až T = 15.
- **ISO 7816-4** specifikuje datové příkazy pro komunikaci s kartou, přístupové metody k datům na kartě, zabezpečení komunikace (*secure messaging*) mezi čtečkou a kartou atd.

Nejjednodušší čipové karty jsou osazené pouze paměťovými registry, které je možné nastavovat, přičítat k nim, odečítat od nich apod. (např. telefonní karty). Na rozdíl od nich mají **procesorové karty** kromě paměti také jednočipový procesor schopný vykonávat příkazy. Procesor je řízen operačním systémem karty. Operační systém procesorové karty je operační systém pro řízení jednočipového procesoru v kartě.

Z hlediska spouštění aplikací v čipových kartách můžeme též karty dělit na statické s pevně nahranými aplikacemi (většina firemních operačních systémů jednotlivých výrobců karet) a dynamické, do kterých je možné zapisovat nejen data, ale i spustitelný kód. Nejznámějšími technologiemi dynamických karet jsou systémy JavaCard či Multos.

ISO 7816-5 Registrace aplikací

ISO 7816-6 Aplikační datové elementy

ISO 7816-7 Příkazy jazyka Structured Card Query Language (SCQL)

ISO 7816-8 Příkazy pro bezpečnostní operace

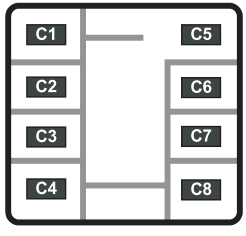
ISO 7816-9 Příkazy pro správu karty

ISO 7816-10 Elektronická signalizace pro synchronní karty

ISO 7816-11 Osobní identifikace pomocí biometrických metod

ISO 7816-12 Komunikace s kontaktní kartou s využitím USB

ISO 7816-15 Standardní aplikace pro uložení kryptografických informací



C1:Vcc = 5 V
C2:Reset
C3:Hodiny
C4:Rezerva
C5:Zem
C6:Vpp (progr. EEPROM)
C7:I/O
C8:Rezerva

Obrázek 2.1: Kontakty čipové karty dle ISO 7816 zakrývají standardem předepsané plošky C1 až C8

Bezkontaktní čipové karty obsahují čip a anténu zalitou v těle karty. Pro komunikaci se čtečkou nepotřebují galvanický spoj, komunikují pomocí elektromagnetických vln. Napájení rovněž obstará čtečka (terminál) na bázi přenosu energie pomocí elektromagnetického vlnění. Kontaktní čipové karty mají na sobě zpravidla kontakty, pomocí kterých se propojují se čtečkou. Napájení rovněž obdrží ze čtečky.

Podskupinou procesorových čipových karet jsou **PKI čipové karty**. Jsou to procesorové čipové karty schopné provádět příkazy nejenom symetrické kryptografie, ale i asymetrické kryptografie a často i výpočet otisku. PKI čipové karty zpravidla mají kryptografické moduly pro urychlení kryptografických operací. PKI čipové karty mohou být nejenom kontaktní, ale i bezkontaktní.

Zejména soukromé klíče určené pro vytváření elektronického podpisu je nutné chránit proti kompromitaci. Jednou z možných cest této ochrany je generování dvojice veřejný/soukromý klíč samotnou PKI čipovou kartou a uložení soukromého klíče do paměťové oblasti karty, která není přímo dostupná vně karty. Je výhradně využitelná přes příkazy karty pro vytvoření elektronického podpisu. Tj. otisk z podepisovaných dat musí být k podpisu zaslán do karty. Jeho šifrování soukromým klíčem pak zajistí sama karta.

Generování párových dat samotnou kartou je velice náročná operace. Není-li karta zajištěna např. proti **útokům postranním kanálem** založeným na sledování elektromagnetického vyzařování karty, může útočník z elektromagnetického pole karty zjistit, že dochází ke generování párových dat. V takovém okamžiku stačí na některé karty působit předem definovaným magnetickým polem a výsledkem je, že vygenerovaná párová data nejsou náhodná, ale jen pseudonáhodná, a to na omezené množině. Výrobci proto garantují, proti jakým postranním kanálům je karta testována.

Ochrana proti **kopírování obsahu čipové karty** patří k dalším základním parametrům čipových karet. Je vynucována splněním bezpečnostních standardů (např. hodnocením karty dle ČSN ISO/IEC 15408, hodnocením dle standardu FIPS či hodnocením dle příslušného standardu ITsec).

Spíše zajímavostí je, že PKI i platební čipová karta může též emulovat autentizační kalkulátor. V takovém případě držitel karty obdrží miniaturní čtečku, která se nepřipojuje k počítači. Na čipové kartě jsou pak uložena aktiva, ze kterých se generují jednorázová hesla, a v miniaturní čtečce pak vlastní logika a display se zobrazovaným jednorázovým heslem.

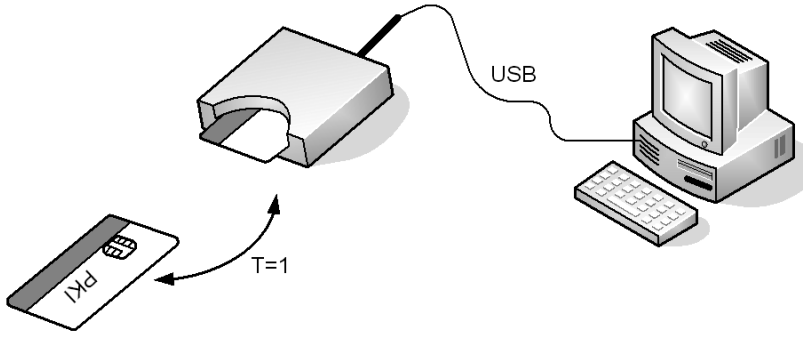
Čtečky čipových karet (terminály)

Čtečka čipových karet se správně označuje jako terminál. Jedná se o zařízení, které zprostředkovává komunikaci s čipovou kartou. Čtečka může být jako samostatné zařízení nebo může být propojena např. s počítačem. Pro propojení čtečky s počítačem se často využívá jiný komunikační protokol než ten, který využívá čtečka pro komunikaci s kartou. Např. na obr. 2.2 probíhá komunikace mezi kartou a čtečkou protokolem T = 1, kdežto čtečka s počítačem komunikuje protokolem USB.

Pro zajištění konverze komunikačních protokolů jsou čtečky často realizovány jednočipovým procesorem, což patřičně navyšuje cenu čteček. Navíc na rozdíl od standardizovaného protokolu mezi kartou a čtečkou jsou protokoly mezi čtečkou a počítačem většinou proprietární, a proto každá čtečka vyžaduje specializovaný ovladač.

Nejčastějšími komunikačními protokoly mezi čtečkou a kartou jsou:

- T = **0** – jedná se o znakově orientovanou asynchronní polo-duplexní sériovou výměnu dat mezi čtečkou a kartou.
- T = **1** – jedná se rovněž o asynchronní polo-duplexní sériový protokol mezi čtečkou (terminálem) a kartou, ale tentokrát blokově orientovaný, tj. mezi terminálem a kartou se přenáší data po celých blocích dat. Tento protokol zrychluje výslednou komunikaci s kartou, avšak karta musí disponovat větší RAM pro vyrovnávací paměti. Dnešní karty umí současně jak protokol T = 0, tak i protokol T = 1. Takové karty se neoznačují jako duální (v obou případech se jedná o sériovou výměnu dat).
- T = **CL** – jedná se o bezkontaktní sériový přenos (CL = *Contact Less*).
- T = **USB** – jedná se o protokol USB. Karta však nedisponuje konektory protokolu USB, takže čtečka je „redukcí" mezi dvěma tvary konektoru. Nemusí tak sama obsahovat čip, a tudíž je výrazně levnější než čtečky pro předchozí protokoly.



Obrázek 2.2: Mezi kartou a čtečkou je jeden komunikační protokol (např. T = 1) a mezi čtečkou a počítačem je druhý komunikační protokol (např. USB)

Zmíněné protokoly řeší pouze fyzickou komunikaci, nicméně v případě absence vyrovnávacích pamětí („bufferů“) v kartě ovlivňují i logiku odesílání příkazů do karty. Nad těmito protokoly se přenáší datové pakety, tzv. **APDU (*Application Protocol Data Unit*)**. Pomocí APDU se zasílají instrukce kartě, která vrací odpovědi. APDU je nízkoúrovňové rozhraní, pomocí kterého již s kartou mohou komunikovat specializované aplikace v počítači (CSP, PKCS#11 modul, může probíhat personalizace karet apod.).

Čipová karta po svém vložení do čtečky a sepnutí napájení vrací tzv. **ATR řetězec**. ATR je maximálně 33 B dlouhý řetězec, který nastaví již výrobce karty. Na základě ATR je často možné odlišit různé druhy karet, není to však pravidlem. Jestliže má operační systém pracovat s konkrétním typem karty, znamená to, že pracuje s kartou o tom a tom ATR řetězci. ATR řetězec by tak měl být předem registrován v operačním systému, aby operační systém byl schopen s konkrétní kartou pracovat. Může nastat situace, kdy dvě karty se shodným ATR a rozdílně provedenou personalizací mohou působit problémy, pokud se operační systém pokouší vybrat příslušný ovladač právě na bázi ATR.

Velice důležitým parametrem čteček se stává **test na vytažení karty ze čtečky**, který oceníme zejména v případě přihlašování k počítači (do Windows, k Linuxu apod.) pomocí čipové karty. V tomto případě je velice důležité, aby čtečka bezpečně signalizovala, že došlo k vyjmutí karty ze čtečky. V takovém případě totiž automaticky dojde k zablokování stanice.

Čtečky, které nemají garantovánu tuto signalizaci, může pak zaměstnanec opouštějící své pracoviště obejít jednoduchým trikem: do čtečky pod čipovou kartu vloží vizitku a z čtečky vytáhne jen kartu (ve čtečce ponechá vizitku). Právě test na signalizaci vytažení karty odlišuje profesionální čtečky od laciných domácích čteček. Trik s vizitkou lze mnohdy úspěšně provádět až po určitém opotřebení čtečky, proto je nutná garance výrobce, že čtečka byla testována proti tomuto útoku.

Se čtečkami jejich výrobci dodávají i příslušné ovladače pro operační systémy. Ovladač čtečky je SW knihovna, pomocí které operační systém komunikuje se čtečkou. Dnes je de facto standardem pro tuto oblast standard PC/SC.

Hybridní a duální karty

Hybridní čipová karta v sobě obsahuje dva čipy: kontaktní i bezkontaktní. Rozměr „velké“ karty dle ISO 7816 totiž umožňuje umístit do karty oba čipy, aniž by si překážely.

Duální čipová karta oproti hybridní čipové kartě obsahuje jen jeden čip s dvěma vstupně/výstupními rozhraními. Zpravidla jedno bývá kontaktní a druhé bezkontaktní. Existují však i čipové karty se dvěma typy kontaktních rozhraní (např. T = 1 a T = USB).

Dnes existují i čipové karty s jedním čipem a více než dvěma vstupně/výstupními interface (např. T = 0 i 1; T = USB a T = CL).

Výroba karty a její životní cyklus

Plastikové karty rozměrů vizitky byly původně zavedeny bankami jako platební karty vyjadřující kredibilitu jejího držitele. Plastikové karty jsou opatřeny ochrannými prvky (logo vydavatele, texty čitelné v infračerveném spektru, hologramy apod.). Plastikové karty jsou zpravidla potištěny (personalizovány) osobními údaji držitele karty. Týž potisk se využívá i v případě čipových karet, pouze potiskem nesmíme zakrýt kontakty nebo zničit čip. Nevhodným potiskem může být zničen i bezkontaktní čip, proto místo, pod kterým je uložen bezkontaktní čip, nepotiskujeme pomocí mechanicky zatěžujících tiskařských technologií.

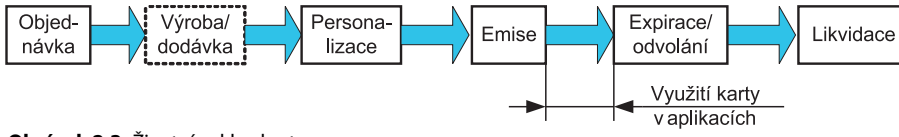
V poslední době se využívají technologie potisku, které nemohou poškodit čip. Vzniká proto i opačný problém. Karta v místě bezkontaktního čipu zpravidla vykazuje jisté nerovnosti, které by naopak mohly deformovat tisk. Výsledkem je tedy stejné ponaučení, aby se nad/pod bezkontaktní čip netisklo.

Výroba čipových karet spočívá ve vlepení/zalití čipu do zpravidla bílého plastu, na který se tisknou/lepí různé potisky a ochranné prvky. Výsledkem je, že z výroby vypadne dávka karet, které jsou více či méně identické. Vydavatel karet (instituce emitující karty) však zpravidla potřebuje pro každého držitele potisknout kartu jeho personálními údaji. Tento proces, kdy z více či méně stejných karet vzniknou karty personalizované pro každého konkrétního držitele, se označuje jako personalizace.

V případě čipových karet se personalizace skládá nejenom z personalizace potisku, ale mnohdy se na kartu nahrávají personální data. Čipová karta se tak často personalizuje nejenom zvenku, ale i „zevnitř". Jelikož se karty vydávají ve větších sériích, musíme zajistit evidenci karty během celého jejího životního cyklu. Za tímto účelem si obstaráme aplikaci pro řízení životního cyklu karet obíhajících v naší firmě.

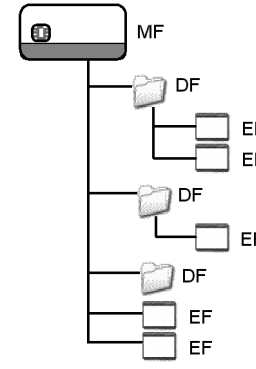
Životní cyklus karet se tak skládá z (obr. 2.3):

- Objednávky dávky karet u výrobce.
- Výroby karet s případným barevným potiskem na spodních vrstvách karty.
- Vytvoření personalizačních dat.
- Personalizace karty (jak zvenku, tak i případná personalizace dat v čipu).
- Emise (předání karty držiteli).
- Případné odvolání kary (ztráta karty, rozvázání pracovního poměru, poškození karty apod.).
- Vypršení platnosti karty a její odevzdání vydavateli.
- Likvidace karty.



Obrázek 2.3: Životní cyklus karty

Struktura dat uložených v kartě



Obrázek 2.4: Datová struktura karty

Data uložená na čipové kartě tvoří souborovou strukturu vzdáleně připomínající souborovou strukturu disku. Na kartě (obr. 2.4) je kořenový adresář nazývaný **MF (*Master File*)**, ve kterém mohou být podadresáře **DF (*Dedicated File*)** nebo datové soubory **EF (*Elementary File*)**.

Čipová karta se netváří jako disk, protože uživatel nemá možnost zjišťovat strukturu karty, tj. nemá k dispozici nějakou obdobu příkazu DIR.

Představa byla taková, že karta může sloužit nejenom PKI, ale současně i zcela jiným aplikacím (např. věrnostní systémy, elektronické peněženky apod.). Každá z takových aplikací by měla na kartě svůj adresář – své DF. Proto se DF také někdy označuje jako aplikace.

Architekt navrhne datovou strukturu karty, která se na kartě vytvoří při její výrobě nebo nejpozději při její personalizaci. Při personalizaci pak mohou být některé EF naplněny (např. identifikačními daty držitele či EF sloužící jako čítač elektronické peněženky, který může být předem nabit na stanovenou částku).

Další vlastností je, že na celou kartu, na určitý DF či na konkrétní EF mohou být nastavena přístupová práva. Přístupová práva mohou být vztažena k autentizaci pomocí PIN nebo může být využit tzv. ***Secure Messaging***. Přístupová práva mohou být nastavena jen na určitý DF, pak např. EF v kořenovém adresáři jsou přístupná bez omezení. Do takových EF pak můžeme např. uložit doma vytvořenou žádost o certifikát, kterou pak na čipové kartě odneseme na registrační autoritu. Čipová karta v tomto případě slouží jako datový nosič. Paměťová kapacita čipových karet dodávaných v současné době se zpravidla pohybuje kolem 64 kB, přičemž část kapacity je rezervována služebním datům operačního systému karty, část je spotřebována při personalizaci a zbytek lze využít pro uživatelská data (certifikáty, klíče apod.).

Na používání PIN/PUK jsou dnešní uživatelé zvyklí a bylo by asi nošením dříví do lesa tuto problematiku dále pitvat. Je třeba jen připomenout, že existují dvě základní strategie generování PIN:

- PIN se generuje při personalizaci karty a vytiskne do Pinové obálky, která je držiteli dopravena často i jinou cestou než karta.
- PIN si držitel karty zadává sám při prvním použití karty.

Při autentizaci pomocí PIN se předpokládá zásah držitele karty. Jak to ale udělat, když potřebujeme provést nějakou operaci s kartou a nechceme, aby uživatel zadával PIN? Tj. vyžadujeme, aby se kartě autentizoval a s ní komunikoval software. K tomuto způsobu autentizace slouží

Secure Messaging. Ten je určen nejenom k autentizaci, ale též k šifrování komunikace s kartou. Stačí si jen uvědomit, že pokud provádíme PKI operace pomocí bezkontaktní karty, tak to asi bez zabezpečení komunikace dost dobře nepůjde.

Secure Messaging bývá realizován tak, že na kartě je uložen symetrický šifrovací klíč. Autentizovaná strana pak musí mít k dispozici týž klíč. Autentizace pak probíhá pomocí dialogu dotaz/odpověď. Pošle se šifrovaná výzva, která je vrácena dešifrovaná (nebo obráceně).

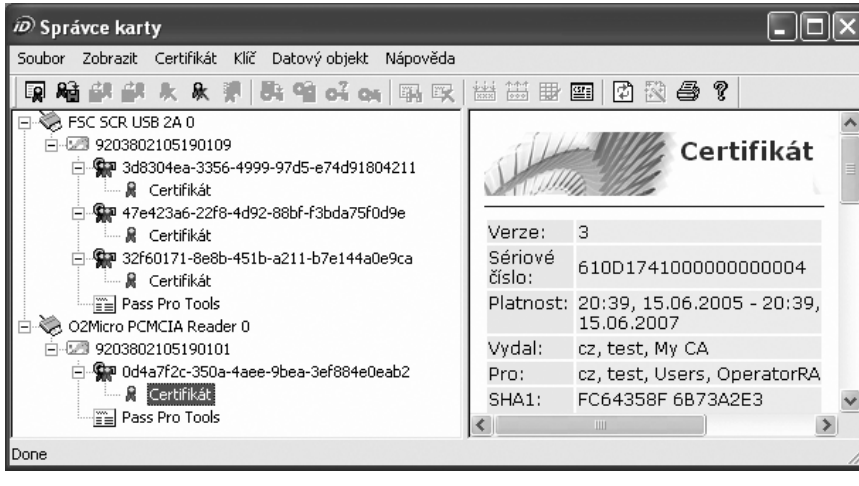
Praktické využití *Secure Messaging* spočívá např. v tom, že zatímco doma se bude držitel vůči kartě autentizovat pomocí PIN, na registrační autoritě od něj PIN nemusí být vyžadován, když s kartou uživatele manipuluje pracovník registrační autority.

Secure Messaging také využijeme, pokud se držiteli zablokuje na kartě PIN; touto cestou je mu totiž možné (i vzdáleně) nastavit nový PIN (pokud ovšem architekt kartu navrhl s touto vlastností). V našich zeměpisných šířkách je spíše zvykem vybavit pro tyto případy držitele PIN ještě kódem PUK.

V případě PKI čipových karet budou v některých EF uloženy soukromé klíče, v jiných veřejné klíče, případně certifikáty. Držitele karty však nebude zajímat, v jakém EF hledat konkrétní soukromý klíč, ale bude chtít mít k dispozici dvojici soukromý/veřejný klíč (případně celý certifikát). Proto PKI čipové karty mají kromě fyzické struktury (MF, DF, EF) ještě strukturu logickou, tvořenou tzv. kontejnery. Příslušející soukromé klíče, veřejné klíče včetně případných certifikátů tvoří jeden konkrétní kontejner.

Jeden z kontejnerů může být označen jako kontejner nesoucí kryptografický materiál pro přihlašování se k počítači pomocí čipové karty.

Dodavatelé PKI čipových karet pak s kartami většinou dodávají i utilitu, která zobrazuje logickou strukturu karty, tj. kontejnery, a pak případné další soubory, které mohou např. obsahovat certifikáty certifikačních autorit apod. Na obr. 2.5 je znázorněno okno takové utility. Všimněte si, že k počítači máme připojeny dvě čtečky. Jedna obsahuje tři kontejnery a druhá jeden kontejner.



Obrázek 2.5: Výpis obsahu karty utilitou dodávanou firmou Monet+

Výsledkem je tedy situace, kdy architekt má k dispozici nepersonalizovanou čipovou kartu. Nyní musí navrhnout souborovou strukturu na kartě. Vytvoří MF a v něm DF a EF. Definuje, kde budou uloženy soukromé klíče, veřejné klíče, certifikáty atd. Dále navrhne využití autentizačních metod a případné zabezpečení komunikace mezi čtečkou a kartou. Nakonec navrhne postup personalizace (včetně potisku karty), který formalizuje tak, aby jej mohly využívat personalizační linky, které budou chrlit personalizované čipové karty.

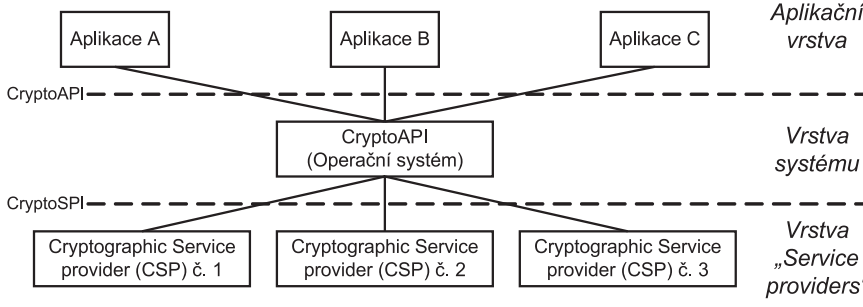
Kromě návrhu samotné čipové karty musí architekt navrhnout i obslužný software (middleware) pro podporu karty v jednotlivých operačních systémech.

Potíž je v tom, že architekt jiného dodavatele navrhne kartu trochu jinak, a pak musí být vyvinut i jiný middleware. Tento problém měla odstranit norma PKCS#15, specifikující, jak mají být na PKI čipové kartě uloženy kryptografické údaje. Také standardizuje strukturu aplikace (aplikace = DF) na kartě, identifikátory souborů a jejich obsah. Jenže se ukázalo, že není implementace PKCS#15 jako implementace PKCS#15. Takže dodavatelé ke své implementaci PKCS#15 stejně museli dodávat vlastní middleware. Navíc implementace PKCS#15 je až zbytečně složitá, takže výsledkem je, že norma PKCS#15 se stala spíše jen inspirací než zákonem pro architekty.

Pokud je potřeba vytvářet aplikace pracující s více druhy karet, doporučuje se sjednotit přístup ke kartě o úroveň výše použitím standardizovaného API pro přístup k middleware karty (Windows CSP, PKCS#11, Java OpenCardFramework).

Čipová karta a operační systém (middleware)

Základním problémem je skutečnost, že čipové karty včetně middleware dodává často jiný výrobce než výrobce operačního systému. Operační systém tedy musí umožňovat do sebe začlenit softwarové moduly třetích stran.



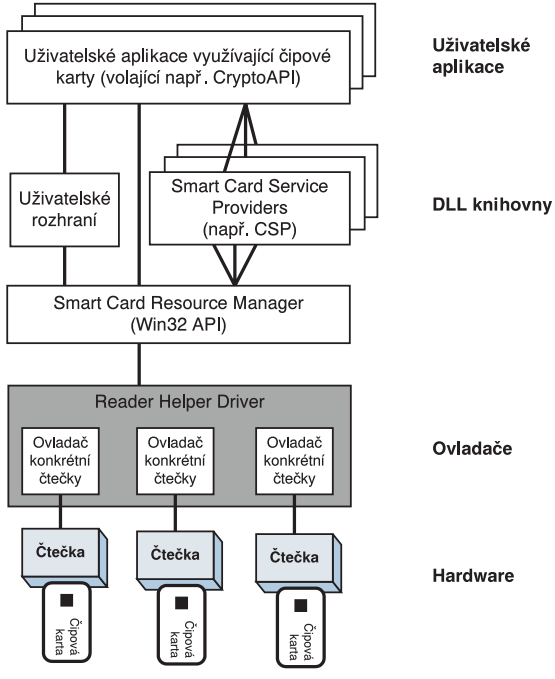
Obrázek 2.6: Microsoft řeší podporu čipových karet pomocí CSP

Na obrázku 2.6 je znázorněno řešení tohoto problému firmou Microsoft. Aplikace požadující kryptografické funkce je logicky požadují od operačního systému. K tomuto účelu operační systém poskytuje rozhraní CryptoAPI. Takovými funkcemi může být generování párových dat, šifrování, podepisování apod. Jenže ve výsledku tyto operace mohou být zajištěny jak prostředky dodávanými s operačním systémem, tak i prostředky třetích stran. Proto CryptoAPI vysokoúrovňové kryptografické požadavky aplikací dekomponuje na jednotlivé kryptografické operace, o jejichž provedení požádá konkrétní specializovaný modul – CSP (*Cryptografic Service Provider*). CSP je pak onen kýžený modul, který může být dodán i třetí stranou. Máli se např. vytvořit digitální podpis, pak je od CryptoAPI vyžadováno dodání CMS zprávy SignedData (CMS viz kap. 22). Do této CMS zprávy je nutné vložit soukromým klíčem šifrovaný otisk. Do CSP tak propadne požadavek: „soukromým klíčem“ šifruj zaslaných 20 B

dat otisku. Je-li CSP realizován pouze softwarově, nalezne se soukromý klíč na disku a spočte výsledek. Pokud se jedná o CSP pro čipovou kartu, transformuje se tento požadavek na APDU příkaz pro čipovou kartu a skrze čtečku se pošle do čipové karty.

Má-li být kryptografická operace provedena pomocí konkrétní čipové karty, musí být s touto čipovou kartou dodán příslušný CSP (DLL knihovna), který musí být předem nainstalován. Microsoft umožňuje do svých operačních systémů začleňovat jen CSP, které jsou digitálně podepsány firmou Microsoft.

Jestliže chceme využívat čipovou kartu také pro přihlašování do systému, musíme současně s instalací CSP v operačním systému registrovat též ATR, aby systém naše karty znal. Modul CSP je většinou dodáván ve formě instalačního programu, který provede všechna potřebná nastavení.



Obrázek 2.7: Celková architektura podpory karet v systémech Microsoft

Microsoft s operačním systémem dodává sadu svých vlastních softwarových CSP (*Microsoft Basic CSP* s omezenými kryptografickými klíči, *Enhanced Microsoft CSP* s plnými klíči atd.), které nevyužívají čipové karty, ale veškerý kryptografický materiál je uložen na disku. Otázkou do pranice je slovo disk. Soukromé klíče a certifikáty bývají uloženy na lokálním disku. Avšak v případě, že využíváte ActiveDirectory a cestovní profily (*Roaming Profile*), může se i soukromý klíč stát součástí cestovního profilu a bude distribuován po celé síti v závislosti na tom, ze kterého počítače se budete přihlašovat do sítě.* Pokud tedy chcete mít soukromý klíč i v síti Windows pod vlastní kontrolou, pak se jeho umístění na čipovou kartu jeví jako dobré řešení.

* Cestování soukromého klíče v cestovním profilu je relativně bezpečné; soukromý klíč je šifrován a bez znalosti hesla se k soukromému klíči nelze dostat.

Dále Microsoft přibaluje do své distribuce CSP některých výrobců čipových karet, které jsou však většinou nepoužitelné, protože i kdybyste náhodou použili čipovou kartu, pro niž je CSP dodáván jako součást systému, stejně ji pravděpodobně nepoužijete se souborovou strukturou přímo od výrobce, ale souborovou strukturou navrženou lokálním dodavatelem, tudíž potřebujete i pro tuto kartu CSP od lokálního dodavatele.

Problém je ještě s tím, že CSP musí komunikovat s kartou skrze čtečku, takže situace je ještě komplikovanější. Každá čtečka musí mít v operačním systému příslušný ovladač. V tom není problém. Potíž je v tom, že k počítači mohu mít připojeno více čteček a kartu mohu vsunout do libovolné z nich, proto je mezi ovladače čteček a CSP vložen ještě manažer čteček (*Smart Card Resource Manager*). Kromě CSP skrze manažer čteček budou ke kartám přistupovat i ostatní programy, jako např. uživatelské rozhraní či utilita pro práci s kartami – viz obr. 2.7. Tato architektura je od verze 2 rozpracována jako průmyslový standard PC/SC (původně se jednalo o standard Microsoftu).

Když aplikace pomocí standardu PC/SC hledá konkrétní čipovou kartu (konkrétní ATR), manažer čteček jí pro každou čtečku připojenou k systému sdělí:

- Jestli je čtečku možné využívat v této aplikaci.
- Jestli je ve čtečce vložena nějaká karta, když ano, pak jí sdělí ATR této karty.
- Jestli je požadovaný ATR registrován v systému.

Standard PC/SC se využívá nejenom pro PKI čipové karty, ale i pro platební čipové karty EMV a rovněž pro hardwarové klíče ve formě USB tokenu, přičemž všechna tato zařízení spojuje využití komunikace pomocí APDU definovaných v ISO 7816. Standard PC/SC zavádí již zmíněný manažer čteček, který umožňuje snadno implementovat a více aplikacím sdílet čtečky a podobná zařízení. Pro dnešní výrobce čteček je již samozřejmostí dodávat ovladače pro PC/SC architekturu. Výsledkem je, že pomocí PC/SC ovladačů pak snadno začleníme čtečky jednotlivých výrobců.

Standard PC/SC se rozšířil natolik, že z původní mateřské platformy Windows byl portován i do prostředí operačních systémů z rodiny UNIX, a to v balíku PCSCLite.

Jiným řešením než CSP je standard PKCS#11. Jedná se o obecně zaměřený standard na zpřístupnění kryptografického hardwaru a mimo jiné jím lze obsluhovat i čipové karty. Tento standard používají pro obsluhu čipových karet zejména operační systémy UNIX. V operačních systémech Microsoft pak standard PKCS#11 využívají alternativní internetové prohlížeče. Standard PKCS#11 neřeší problematiku manažera čteček, proto i v systémech UNIX se pro správu čteček využívá standard PC/SC.

Mini klíč (*USB token*)

Obdobné technologie jako v případě čipových karet se používají i v případě tzv. mini klíčů. Mini klíč se nepřipojuje k PC prostřednictvím čtečky, ale pomocí USB portu, který je součástí všech nových typů PC.

Obliba mini klíčů však nenaplnila očekávání jejich výrobců. Hlavním argumentem pro jejich nasazení byla totiž cena. V moha případech se často dodává sada: co karta, to čtečka. Výsledná sada pak není zrovna lacinou záležitostí. Mini klíče jsou v tomto případě alternativou, která nepotřebuje čtečku. Jenže díky malé sériovosti mini klíčů je jejich cena vysoká

a zejména čtečky pro protokol T = USB jsou laciné. Navíc uživatelé nejsou zařízeni, aby se mohli o mini klíče starat. Pokud se mini klíč strčí do kapsy, lze jej rozsednout. A na čipové karty jsou již uživatelé zvyklí, neboť v peněženkách mají místo pro platební karty stejného rozměru.

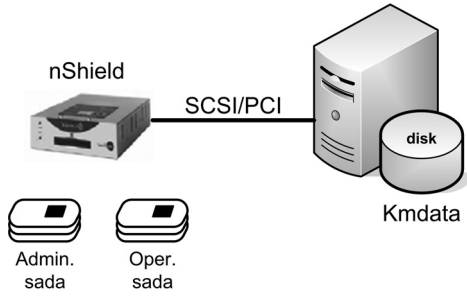
HSM (*Host Security Modul*)

Soukromé klíče důležitých serverů (např. soukromý klíč samotné certifikační autority) nebývají ukládány na čipových kartách, ale ve specializovaných „černých skříňkách" (HSM), které jsou vybaveny speciální fyzickou bezpečností umožňující např. vymazání uloženého kryptografického materiálu v případě mechanické manipulace s boxem (klíč nemusí být smazán, ale může být např. šifrován symetrickou šifrou). Náročnější boxy mohou v případě mechanického útoku aktivovat výbušninu, která box i se soukromým klíčem zcela zničí…

Přínosem HSM však není jen ochrana kryptografického materiálu proti fyzickému útoku. Mnohem přínosnější vlastností HSM je zamezení přístupu ke kryptografickému materiálu nepovolaným osobám – zejména správcům serverů, k nimž je HSM připojen. Použití HSM tak přináší oddělení role správce serveru od role bezpečnostního administrátora, který jediný má přístup ke kryptografickému materiálu. Navíc často může být i oddělena role bezpečnostního operátora, který může kryptografický materiál využívat. Pro ještě větší bezpečnost může být role bezpečnostního administrátora a operátora navíc rozdělena mezi více osob.

Rozdíl mezi bezpečnostním administrátorem a bezpečnostním operátorem spočívá v tom, že administrátor může zálohovat HSM, zaměňovat HSM za jiný HSM, obnovovat obsah HSM apod. Kdežto operátor nemůže dělat žádnou z těchto akcí, ale zase může vydávat svolení k tomu, aby HSM provedl konkrétní kryptografickou operaci (např. vytvořil digitální podpis).

Software s HSM zpravidla komunikuje pomocí CSP nebo PKCS#11 modulu v závislosti na operačním systému počítače, kde je provozován.



Obrázek 2.8: HSM nShield

Činnost HSM si ukážeme na asi nejznámějším HSM, kterým je nShield od firmy nCipher. Jedná se o HSM modul, který se k počítači připojuje buď pomocí SCSI sběrnice nebo je realizován jako PCI karta. HSM modul obsahuje čtečku čipových karet. Součástí dodávky HSM je i balíček čipových karet, z nichž si během instalace HSM vytvoříme sadu administrátorských karet a případně i sadu operátorských karet.

Během instalace vznikne tzv. bezpečný svět HSM modulu (Security World), ve kterém si poklidně bude žít náš kryptografický materiál. Během instalace k vytvořenému bezpečnému světu vygenerujeme sadu administrátorských čipových karet a volitelně i sadu operátorských čipových karet. Na počítači, k němuž je HSM připojen, vznikne adresář, který kromě obslužného softwaru obsahuje podadresář Kmdata. Do tohoto podadresáře je průběžně zálohován celý bezpečný svět (obsah HSM modulu). Záloha je však na disku šifrována dostatečně dlouhými klíči.

Při výměně HSM modulu nám stačí mít k dispozici obsah podadresáře Kmdata a administrátorskou sadu čipových karet. Do bezpečného světa také můžeme přidávat další HSM moduly, které pak sdílí náš kryptografický materiál.

K obnovování/přidávání HSM modulů do bezpečného světa je třeba mít k dispozici ***k*** z ***n*** administrátorských čipových karet (viz též Shamirův algoritmus). Kdežto ***k*** z ***n*** operátorských čipových karet bude třeba pro každé využití kryptografického materiálu v HSM modulu (např. pro vytvoření digitálního podpisu pomocí klíče uloženého v HSM).

Nyní již máme funkční bezpečný svět a dejme tomu máme též nainstalován CSP pro připojení k Windows serveru, na kterém budeme instalovat certifikační autoritu (součást Windows serveru). Spustíme instalaci, vše probíhá, jak jsme zvyklí, bez použití HSM. Až při instalaci certifikační autority dojdeme k okamžiku, kdy bude nutné vygenerovat dvojici veřejný/soukromý klíč CA. Pokud v tomto okamžiku zvolíme CSP pro HSM modul, začne se s generací párových dat v modulu. Nyní je velice důležité, máme-li vytvořenu sadu operátorských karet. V případě, že ano, aktivuje se okno middleware HSM modulu a jsme vyzváni k postupnému zasunutí ***k*** z ***n*** operátorských čipových karet. Po zasunutí ***k*** operátorských čipových karet nám instalace CA pokračuje dále.

Prostředky pro bezpečné vytváření elektronického podpisu (SSCD)

Termín SSCD (anglicky: *Secure Signature Creation Device*, česky: prostředek pro bezpečné vytváření elektronického podpisu) byl zaveden evropskou směrnicí 1999/93/ES o elektronickém podpisu, a to konkrétně v její Příloze III. Tuto evropskou směrnici jednotlivé členské země EU implementovaly do své národní legislativy formou zákona o elektronickém podpisu. Součástí těchto zákonů pak bývá ustanovení, že některý z orgánů státní moci vyhodnocuje, je-li mu předložené zařízení ve shodě s Přílohou III směrnice 1999/93/ES. Výsledkem je pak národní registr zařízení, která jsou ve shodě s uvedenou Přílohou III.

Hodnotit bezpečnost zařízení lze podle různých norem: FIPS, ITsec, ISO15408

SMĚRNICE EVROPSKÉHO PARLAMENTU A RADY 1999/93/ES z 13. prosince 1999

O zásadách Společenství pro elektronické podpisy

…..

PŘÍLOHA III Požadavky na prostředky pro bezpečné vytváření elektronických podpisů

Prostředky pro bezpečné vytváření podpisů musí vhodnými technickými prostředky a postupy přinejmenším zajistit, aby:

- se data pro vytváření podpisu mohla vyskytnout pouze jedenkrát a aby bylo dostatečně zajištěno jejich utajení;

(*Common Criteria*) atp. Běžný občan ale nemá kvalifikaci posoudit výsledky hodnocení podle jednotlivých norem, proto je služba, kterou za něj provede stát, docela praktická. Občanovi stačí podívat se do příslušného registru a ví, kolik uhodilo. Bohužel, v České republice tato hodnocení nějak usnula. Díky našemu členství v EU nám však stačí tyto informace získat z registru libovolného jiného člena EU (např. na Slovensku).

- bylo dostatečně zajištěno, že data pro vytváření podpisu nelze odvodit a že podpis je dostupnými technickými prostředky chráněn proti jakémukoli padělání;
- podepisující osoba měla možnost data pro vytváření podpisů spolehlivě chránit proti jejich zneužití třetí osobou.

Prostředky pro bezpečné vytváření podpisů nesmí měnit data, která mají být podepsána, ani bránit tomu, aby tato data byla podepisující osobě předložena před procesem podepisování.

Tato směrnice 1999/93/ES se však zabývá jen problematikou vytváření kvalifikovaného elektronického podpisu. Neřeší problematiku autentizace, šifrování, dokonce ani problematiku archivace dokumentů opatřených kvalifikovaným elektronickým podpisem. Přesto informace, že zařízení je na uvedeném registru, může být referencí velice důležitou zejména při nákupu těchto zařízení.

Porovnání jednotlivých prostředků

Nyní se zamysleme nad otázkou, kdy a jaký prostředek zvolit pro ochranu našeho kryptografického materiálu (našich aktiv). Asi nejdůležitějšími kritérii takové volby jsou:

- Technická náročnost (malá, střední, velká). U autentizačních kalkulátorů je obdobou pracnost, se kterou ručně připisujeme data mezi kalkulátorem a počítačem (malá, pracné, příliš pracné).
- Hodnota chráněných aktiv (malá, střední, velká)
- Cena prostředku (nízká, střední, velká)
- Prostředí, ve kterém budeme prostředek využívat. Z hlediska bezpečnostních charakteristik rozlišujeme následující tři typy prostředí, v nichž jsou provozovány aplikace:
  1. **Prostředí kontrolované provozovatelem aplikace** – toto prostředí nemůže uživatel ani útočník ovlivnit, protože prostředí je předem nakonfigurováno. Může do něj zasáhnout jen správce aplikace nebo výjimečně výrobce. Takovým prostředím je např. mobilní telefon či bankomat.
  2. **Prostředí kontrolované uživatelem** – uživatel si sám zodpovídá za konfiguraci a údržbu prostředí (např. osobní počítač na pracovišti či doma).
  3. **Prostředí kontrolované třetí stranou** – klient pracuje na PC, které je veřejně přístupné (např. v internetové kavárně). Toto prostředí se vyznačuje tím, že za jeho bezpečnost nemůže odpovídat ani uživatel, ani provozovatel aplikace. Takovýmto prostředím je např. internetová kavárna nebo školní počítačová učebna.

				Prostředí		
	Cena	Pracnost	Chráněná aktiva	Kontrolované provozovatelem	Kontrolované uživatelem	Pod kontrolou třetí strany
Stálé heslo	**Nízká**	**Malá**	**Malá**	☺	☺	-
			Střední	-	-	-
			Velká	-	-	-
Seznam hesel na jedno použití; autentizační karta, jednorázové heslo přes nezávislý kanál (SMS)	**Nízká**	**Pracné**	**Malá**	☺	☺	☺
			Střední	☺	☺	-
			Velká	-	-	-
Rekurzivní algoritmus nebo jiná softwarová autentizace jednorázovým heslem	**Nízká**	**Malá**	**Malá**	☺	☺	☺
			Střední	-	☺	-
			Velká	-	-	-
Autentizační kalkulátory	**Střední**	**Pracné**	**Malá**	☺	☺	☺
			Střední	☺	☺	☺
			Velká	☺	☺	☺
Soukromý klíč na disku	**Nízká**	**Malá**	**Malá**	☺	☺	-
			Střední	-	☺	-
			Velká	-	-	-
PKI čipová karta, mini klíč	**Střední**	**Střední**	**Malá**	☺	☺	-
			Střední	☺	☺	-
			Velká	-	˘	-
HSM	**Vysoká**	**Střední**	**Malá**	☺	☺	-
			Střední	☺	☺	-
			Velká	☺	☺	-

Z uvedené tabulky jako by vyplývalo, že nejlepším nástrojem je autentizační kalkulátor. Avšak díky větší pracnosti při práci s autentizačním kalkulátorem a nemožnosti jej využít pro automatickou počítačovou komunikaci se z něj stává doplňkový nástroj, který má tu výhodu, že jej můžeme použít v libovolném prostředí (např. cestujeme-li na konferenci do zahraničí a není tak k dispozici nic jiného než internetové kiosky apod.).

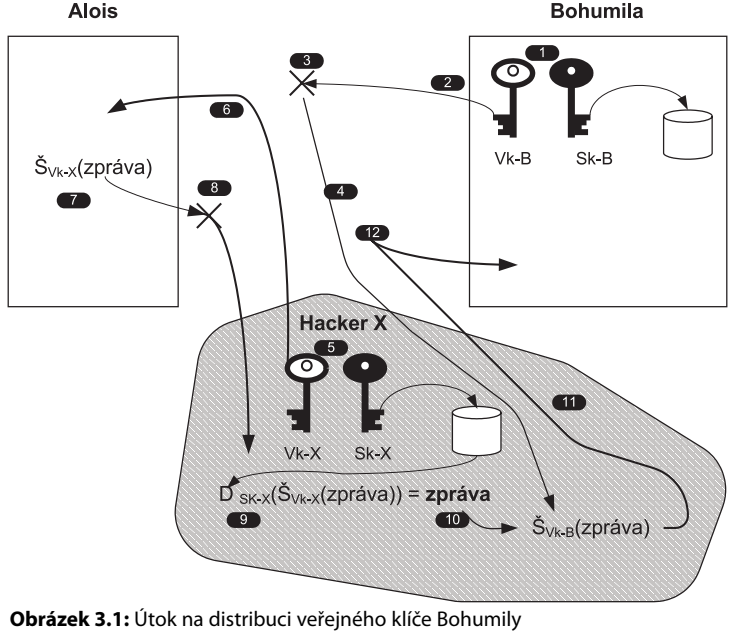
Zajímavé je, že mnohdy se volí PKI čipová karta anebo autentizační kalkulátor a hledají se důvody pro jednu nebo druhou pomůcku. Je jasné, že PKI čipovou kartu autentizačním kalkulátorem nelze nahradit, a obráceně, ani kalkulátor nelze nahradit PKI čipovou kartou. Cestující by tak měl být vybaven oběma prostředky osobní autentizace. Základním nástrojem osobní autentizace by tak měla být PKI čipová karta a jen v případech, kdy není jiná volba, pak autentizační kalkulátor (např. emulovaný na čipové kartě).

Kapitola 3

Certifikáty a certifikační autority

Vraťme se zpět k našim známým: k Aloisovi, Bohumile a Cyrilovi (sledujte obr. 3.1). Bohumila vygenerovala dvojici asymetrických klíčů (1); vygenerovaný veřejný klíč Vk-B poslala Aloisovi po dotěrném Cyrilovi (2). Bohumila doufá, že Alois zašifruje svou odpověď jejím veřejným klíčem, jejž mu přinese Cyril. Tím Alois zamezí, aby si zprávu přečetl kdokoliv jiný než Bohumila. I kdyby Alois tuto šifrovanou zprávu poslal po Cyrilovi, tak si ji ani Cyril nepřečte.

Cyril se pomalu vleče k Aloisovi a tu mu v hlavě uzraje plán (3). Zastaví se u své kamarádky – hackerky označující se jako slečna X. Třeba mu poradí, každopádně tím nic nezkazí. Slečna X si prohlédne veřejný klíč Bohumily a okamžitě odvětí Cyrilovi, že zlomit RSA algoritmus je i nad její síly. Ale existuje přece úplně jednoduchá šance, jak Cyrilovi pomoci! Vezme si od Cyrila veřejný klíč Bohumily a schová si jej pro pozdější využití (4). Nyní sama vygeneruje svou dvojici: veřejný klíč Vk-X a soukromý klíč Sk-X (5). Právě vygenerovaný veřejný klíč Vk-X dá Cyrilovi a vyzve ho: Dones jej Aloisovi a řekni mu, že to je ten veřejný klíč, který mu posílá Bohumila. Cyril tak neprodleně učiní (6).



Obrázek 3.1: Útok na distribuci veřejného klíče Bohumily

Nyní Alois v dobré víře, že svou odpověď šifruje veřejným klíčem Bohumily, zašifruje odpověď veřejným klíčem Vk-X a pošle ji po Cyrilovi Bohumile (7). Cyril rovnou pospíchá za slečnou X (8). Ta na něj již netrpělivě čeká. Vytrhne mu jejím veřejným klíčem Vk-X šifrovanou Aloisovu odpověď. Dešifruje ji (9) svým soukromým klíčem Sk-X a získá čistou zprávu. Tu ukáže překvapenému Cyrilovi, kterému dokonce umožní zprávu změnit v jeho prospěch. Výsledek pak šifruje veřejným klíčem Bohumily Vk-B (10) a po Cyrilovi ho pošle Bohumile (11). Nic netušící Bohumila dešifruje zprávu svým soukromým klíčem Sk-B (12).

Všichni jsou šťastní. Alois zprávu šifroval, tak jak mu kázal dobrý mrav. Bohumila obdržela zprávu od Aloise, kterou dešifrovala, tak jak měla. Cyril si nejenom mohl přečíst obsah zprávy, ale dokonce jej mohl i změnit ve svůj prospěch. A Slečna X se rovněž realizovala. Všichni jsou tudíž šťastní! Ve světě businessu bychom řekli, že všichni postupovali dle ISO 27001.

Jaká je obrana?

Jak se tedy bránit proti útokům na distribuci veřejného klíče? Před tím, než Alois použije nějaký veřejný klíč, by si měl obstarat nějaký důkaz, že veřejný klíč je opravdu jeho, tj. že není podvržen.

Alois má následující možnosti pro ověření pravosti Bohumilina veřejného klíče:

- Alois obdrží veřejný klíč osobně přímo od Bohumily na jejich vzájemné schůzce. Je tedy nad všechny pochybnosti jasné, že veřejný klíč je Bohumily.
- Alois si ověří, že veřejný klíč je opravdu Bohumily. Např. před prvním použitím klíče zavolá Bohumile, autentizuje ji, aby si byl jist, že telefonuje opravdu s ní a ne s nějakým útočníkem, a požádá ji, aby mu zarecitovala otisk z veřejného klíče nebo jeho část. Autentizaci provede např. na základě společně prožitého zážitku: „Jak se ti líbil ten film, na kterém jsme spolu včera byli?“ A Bohumila odpoví: „Co blbneš, vždyť včera jsme spolu byli v divadle.“
- Bohumila si nechá stvrdit nezávislou třetí stranou (např. notářem) pravost svého veřejného klíče.

Vlastní Bohumila odpovídající soukromý klíč?

I když si je Alois zcela jist, že veřejný klíč dostal od Bohumily, je pro něj důležité si ověřit, že Bohumila má ke svému veřejnému klíči příslušný soukromý klíč. Proč? Představme si situaci, že Bohumila má podezření, že Alois miluje kromě ní ještě její, dnes již bývalou, kamarádku Hanu. Když Alois posílá milostná psaní Bohumile, šifruje je veřejným klíčem Bohumily. Když posílá psaní Haně, šifruje je veřejným klíčem Hany. Alois si je natolik jist asymetrickou kryptografií, že dokonce po Bohumile posílá Haně Haniným veřejným klíčem šifrované zprávy. Alois přitom Bohumile tvrdí, že to je čistě obchodní věc – nic soukromého. Bohumile ani nezáleží na tom, co Alois Haně píše, to je jí vcelku jasné. Musí tomu udělat přítrž.

A tak sdělí Aloisovi, že z kryptografických důvodů přejde na nový pár veřejný/soukromý klíč. Jenže nevygeneruje novou dvojici veřejný/soukromý klíč, ale vezme Hanin veřejný klíč a předá jej Aloisovi, jako by to byl její veřejný klíč. Alois si ničeho nevšimne a vesele komunikuje dál s Hanou i s Bohumilou. Bohumila oželí, že se nedozví, co jí Alois píše, a když po krátkém čase Aloise potká, mezi řečí mu sdělí: „Stala se nám s Hanou taková až

neuvěřitelná věc. Zjistily jsme, že máme stejný veřejný klíč. To asi máme stejný i soukromý klíč! Takže si obě můžeme dešifrovat tvé zprávy."

Bylo by tedy nanejvýš vhodné, aby Bohumila také podala Aloisovi důkaz o tom, že vlastní i odpovídající soukromý klíč (*private key possession*). Takovým důkazem může být např. elektronický podpis vytvořený odpovídajícím soukromým klíčem z veřejného klíče nebo z nějaké datové struktury, která veřejný klíč obsahuje. Tento typ důkazu je však možné provádět jen tehdy, když je pomocí odpovídajícího soukromého klíče možné vytvářet digitální podpis.

V případě asymetrických algoritmů neumožňujících digitální podpis, ale umožňujících šifrování se důkaz o vlastnictví příslušného soukromého klíče postaví na šifrování. Např. Alois šifruje Bohumile zprávu jejím veřejným klíčem a čeká, zdali jí porozuměla, tj. jestli má k dispozici odpovídající soukromý klíč.

Důkaz o vlastnictví soukromého klíče

Žárlivá Bohumila je obeznámená se základy kryptografie. Instaluje program Wireshark (viz kap. 12) a čeká, až bude Hana posílat svůj veřejný klíč včetně důkazu o vlastnictví příslušného soukromého klíče Aloisovi. Bohumila při trošce štěstí odchytne jak veřejný klíč, tak i důkaz o vlastnictví soukromého klíče vytvořený jako elektronický podpis zasílané zprávy. Nyní může Bohumila zaslat totéž i Aloisovi (*replay attack*). Kdyby se chtěl Alois bránit proti tomuto typu útoků, měl by kontrolovat, jestli už v minulosti náhodou neobdržel stejný veřejný klíč, čímž by ošetřil i případ, že opravdu si obě vygenerovaly stejné klíče (nejspíše chybou v softwaru).

Důkaz o vlastnictví soukromého klíče má kromě kryptologických aspektů i aspekty čistě technické. Když chce Alois šifrovat za využití veřejného klíče, nesmí se splést ani v jednom bitu veřejného klíče. Jinak by Bohumila nedokázala zprávu dešifrovat. A pokud Alois správně verifikuje důkaz vlastnictví soukromého klíče, pak si je zpravidla jist, že správně interpretuje všechny bity veřejného klíče. Tj. ví, že když Bohumila nedešifruje zprávu, není chyba na jeho straně.

Generovala Bohumila svá párová data na bezpečném zařízení?

Máme důkaz o tom, že veřejný klíč patří konkrétní osobě, a jiný důkaz o tom, že daná osoba má k tomuto veřejnému klíči příslušný soukromý klíč. A aby toho nebylo dost, tak máme ještě třetí důkaz: o tom, že příslušný pár klíčů byl generován a chráněn zařízením odpovídající bezpečnostní úrovně.

Soukromý klíč je aktivum, které může mít značnou hodnotu. Takové aktivum se snažíme odpovídajícím způsobem střežit. Soukromý klíč proto mnohdy udržujeme v odpovídajících zařízeních. Takovým zařízením mohou být čipové karty, USB tokeny či HSM, které samy generují pár klíčů a soukromý klíč tato zařízení nikdy neopouští. Bylo by nemilé, kdybychom např. omylem vygenerovali pár klíčů mimo toto zařízení a soukromý klíč uložili na disk. Investice do speciálního zařízení by vešla vniveč a navíc bychom naše střežené aktivum vystavili nebezpečným rizikům.

Jako důkaz, že soukromý klíč je střežen odpovídajícím zařízením, zpravidla využíváme jednorázová hesla, která toto zařízení generuje společně s odpovídajícím párem klíčů. Jak je ale

možné, že nám stačí jedno jednorázové heslo při generování páru klíčů? Tento důkaz nám totiž stačí pouze jednou – v okamžiku vytváření žádosti o certifikát.

Závěr

Jestliže Alois přímo použije Bohumilin veřejný klíč, měl by od Bohumily obdržet:

1. Samotný veřejný klíč.
2. Důkaz o tom, že tento veřejný klíč opravdu patří Bohumile.
3. Důkaz o tom, že Bohumila má k tomuto veřejnému klíči opravdu příslušný soukromý klíč.
4. V případě, že se jedná o zabezpečení dat značné ceny, důkaz o tom, že párová data byla generována na odpovídajícím zařízení a příslušný soukromý klíč je odpovídajícím způsobem chráněn.

Pokud si Bohumila nechá ověřit platnost svého soukromého klíče u nezávislé třetí strany (např. notáře), předvede notáři čtyři zmíněné úkony a notář vystaví ověření o platnosti veřejného klíče Bohumily. Bohumila pak Aloisovi předá veřejný klíč a příslušné ověření tohoto klíče vydané notářem (vytištěné na papíře např. v hexadecimálním tvaru).

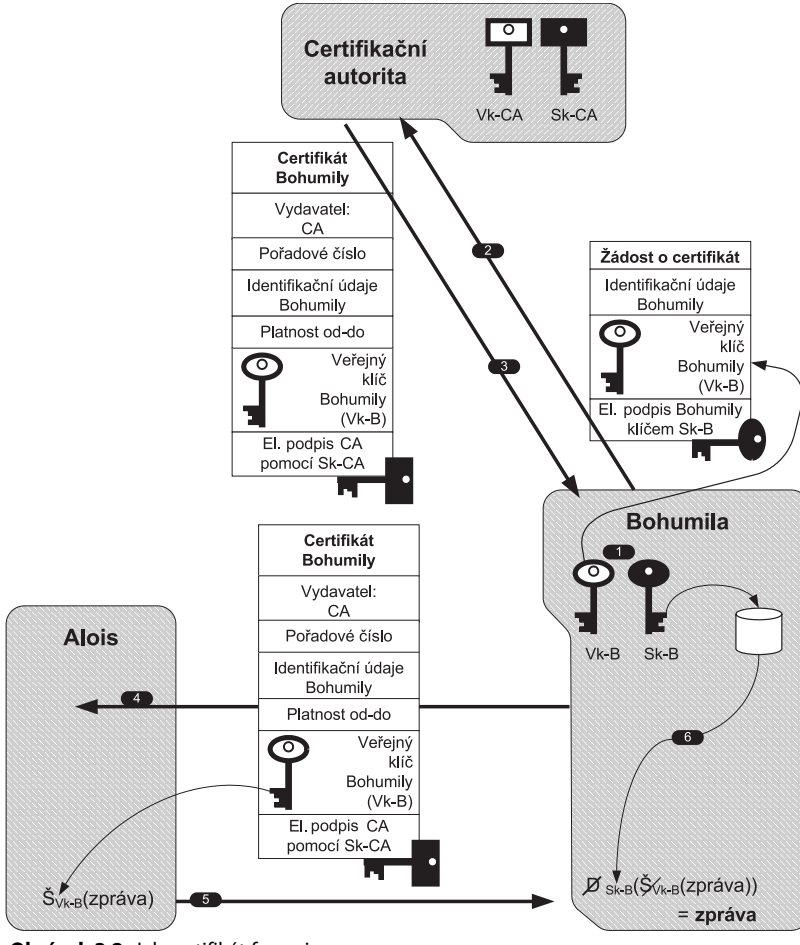
Certifikace veřejného klíče

Proti podvržení veřejného klíče je však praktičtější se bránit certifikací veřejného klíče nezávislou třetí stranou – certifikační autoritou (obr. 3.2). Bohumila si vygeneruje dvojici veřejný/soukromý klíč, přičemž soukromý klíč si jako své tajné aktivum pečlivě uloží a střeží. Veřejný klíč nezašle Aloisovi samotný, ale až jako součást certifikátu vydaného certifikační autoritou. Avšak nepředbíhejme.

Po vygenerování dvojice klíčů (1) Bohumila sestaví strukturu „žádost o certifikát". Tato struktura obsahuje identifikační údaje Bohumily, její veřejný klíč a případně další data, o kterých si povíme později. Tuto strukturu digitálně podepíše svým právě vygenerovaným soukromým klíčem (= důkaz o vlastnictví soukromého klíče) a výsledek předá certifikační autoritě (2). Certifikační autorita může ověřit totožnost Bohumily. V každém případě však verifikuje elektronický podpis na žádosti o certifikát, aby si CA ověřila, že Bohumila opravdu vlastní příslušný soukromý klíč. Pokud je žádost certifikační autoritou shledána v pořádku, pak certifikační autorita vystaví certifikát.

Certifikát je datová struktura obsahující veřejný klíč Bohumily a její identifikační údaje. Dále certifikát obsahuje název vydavatele certifikátu (jedinečné jméno certifikační autority), pořadové číslo vydaného certifikátu, platnost certifikátu atd. Spojení identifikačních údajů Bohumily a jejího veřejného klíče stvrzuje certifikační autorita svým digitálním podpisem.

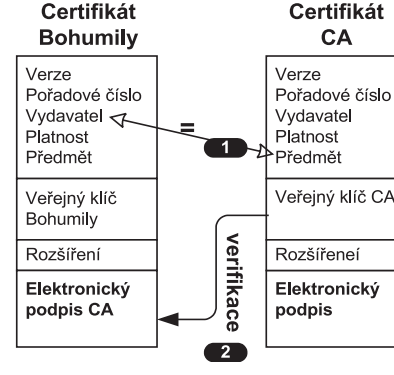
Bohumile je certifikační autoritou vrácen vystavený certifikát (3). Nyní již může Bohumila svůj veřejný klíč poslat i Aloisovi jako součást právě vystaveného certifikátu (4). Alois tento certifikát ověří a v případě, že je vše OK, extrahuje z tohoto certifikátu veřejný klíč, který následně využije k šifrování zprávy Bohumile (5). Bohumila pak pomocí svého soukromého klíče zprávu dešifruje (6) a získá tak původní zprávu.



Obrázek 3.2: Jak certifikát funguje

Co Alois na certifikátu Bohumily ověřuje? Později si povíme, že ověřování (verifikace) certifikátu je docela komplikovanou procedurou. V tomto okamžiku jen připomeňme, že Alois musí ověřit, zdali byl certifikát Bohumily vydán pro něj důvěryhodnou certifikační autoritou (jejíž identifikace je v položce vydavatel certifikátu). Jelikož certifikát je digitálně podepsaná struktura, musí Alois ověřit digitální podpis na Bohumilině certifikátu.

Na obr. 3.3 je znázorněna vazba mezi certifikátem Bohumily a certifikátem CA, která Bohumile certifikát vydala. Alois si musí nalézt příslušný certifikát certifikační autority ve svém úložišti certifikátů důvěryhodných certifikačních autorit nebo jiným



Obrázek 3.3: Zjednodušené ověření (verifikace) certifikátu Bohumily

mechanismem (např. pomocí rozšíření Přístup k informacím úřadu). Postupuje tak, že nejprve prohledá své úložiště důvěryhodných certifikátů a vyhledá v něm certifikát certifikační autority, která vydala certifikát Bohumile. Ten pozná podle toho, že má identickou položku Předmět s položkou Vydavatel v Bohumilině certifikátu (1). Poté vyextrahuje veřejný klíč z certifikátu certifikační autority a pomocí něj verifikuje elektronický podpis v Bohumilině certifikátu (2).

Certifikáty certifikačních autorit jsou podobné uživatelským certifikátům. Svůj certifikát si certifikační autorita může nechat vydat:

- U jiné certifikační autority, pak certifikát této CA má různé položky Vydavatel a Předmět. Takový certifikát CA nazýváme křížovým certifikátem (*cross certificate*).
- Může jej vydat sama, tj. podepisuje si jej sama svým soukromým klíčem, pak certifikát této CA má shodné položky Vydavatel a Předmět. Takový certifikát označujeme jako kořenový (*self signed certificate*).

Achillova pata certifikátu

Certifikát je veřejná listina. Cílem držitele certifikátu je maximálně certifikát šířit. Nepříjemností by ale bylo, kdyby certifikační autorita vystavila nějakému uživateli certifikát pro konkrétní veřejný klíč a záhy by se objevil jiný uživatel s žádostí o certifikaci téhož klíče. Protože když tito uživatelé mají stejný veřejný klíč, navzájem si znají i soukromé klíče.

Certifikační autorita by každopádně měla sledovat, zdali již stejný veřejný klíč necertifikovala, a další žádost o certifikaci téhož klíče odmítnout. Zde je vidět, jak praktické je doplňovat žádost o certifikát důkazem o vlastnictví soukromého klíče. Bez tohoto důkazu by mohl o certifikaci již certifikovaného veřejného klíče žádat každý, komu se nějaký certifikát dostane do ruky.

Generování shodných párových dat se zamezuje využíváním kvalitních generátorů náhodných čísel při generování dvojice veřejný/soukromý klíč. Používají se tzv. pravé generátory náhodných čísel (*true random*). Je pak nepravděpodobné, že by si dva různí uživatelé vygenerovali stejná párová data. Ale chybou softwaru se může stát, že generátor náhodných čísel čísla negeneruje zcela náhodně. Jsou popsány i případy, kdy chyba generátoru náhodných čísel byla navozena uměle, pomocí tzv. útoků postranními kanály.

Certifikát

Certifikát se často přirovnává k občanskému průkazu či pasu. Zatímco občanský průkaz se vydává v tištěné podobě, certifikát je digitálně podepsanou datovou strukturou, jejíž základní součástí je veřejný klíč držitele certifikátu.

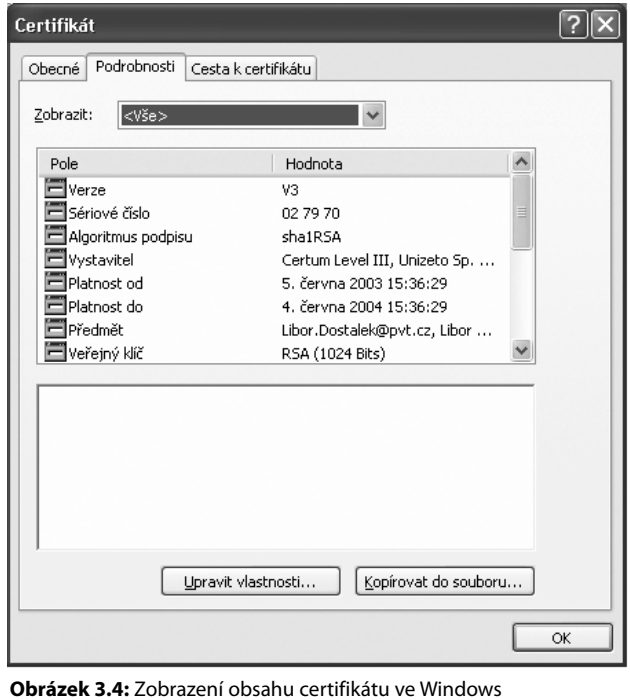
Můžeme i porovnat jednotlivé položky občanského průkazu a certifikátu (později se dozvíme, že toto porovnání je základem práce zaměstnance registrační autority při ověřování totožnosti žadatele o certifikát):

Položka certifikátu	Položka občanského průkazu
Verze (Version)	**Verze formátu občanského průkazu (knížka, karta apod.)**
Pořadové číslo (Serial number)	**Číslo občanského průkazu**
Algoritmus podpisu (Signature Algorithm)	**Způsob podpisu úředníka, typy ochranných prvků**
Vydavatel (Issuer)	**Vydal**
Platnost (Validity)	**Platnost**
Předmět: jméno, adresa, … (Subject)	**Jméno a adresa**
Veřejný klíč (Subject Public Key)	**-**
-	**Fotografie**
Rozšíření certifikátu (Extension)	**Nepovinné údaje**
Elektronický podpis (Digital signature)	**Rukou psaný podpis, aplikace ochranných prvků**

V mnoha evropských zemích se dokonce již vydávají občanské průkazy ve tvaru čipové karty, na které jsou certifikáty držitele karty.

Máme několik norem definujících strukturu certifikátu (X.509, EDI, WAP apod.). V Internetu se vychází ze standardu X.509 verze 3, který vydal ITU. Pro potřeby Internetu je vytvořen internetový profil standardu X.509 v příslušném RFC. Aktuálním internetovým profilem certifikátu je dnes standard RFC-5280.

Nyní se zastavíme u jednotlivých položek certifikátu.



Obrázek 3.4: Zobrazení obsahu certifikátu ve Windows

Verze certifikátu

Verze certifikátu souvisí s tím, je-li certifikát odvozen od normy X.509 verze 1, 2 nebo 3*. Položka *Version* má v případě verze jedna hodnotu nula, v případě verze 2 hodnotu 1 a v případě verze 3 hodnotu 2. Dnes se zásadně používají pouze certifikáty verze 3.

Pořadové číslo certifikátu

Pořadové číslo certifikátu (*Serial Number*) je definováno jako celé kladné číslo, které musí být jednoznačné v rámci konkrétní certifikační autority. Tj. certifikační autorita nesmí vydat dva certifikáty, které by měly stejné pořadové číslo. Dvojice položek *Serial Number* + *Issuer* jednoznačně identifikují certifikát.

Algoritmus podpisu

Položka Algoritmus podpisu (*Signature Algorithm*) specifikuje algoritmy použité CA pro vytvoření elektronického podpisu certifikátu. Tato položka vždy specifikuje dvojici algoritmů:

- Jeden pro výpočet otisku (hash).
- Druhým algoritmem je asymetrický algoritmus, kterým je otisk šifrován.

Platnost

Položka Platnost (*Validity*) určuje platnost certifikátu od (*Not Before*) do (*Not After*).

Častou otázkou je, proč je omezena doba platnosti certifikátu. Důvody jsou dva:

- **Organizační**, tj. aplikace má určitou životnost. Bezesporu je i obchodně zajímavé vydávat certifikáty častěji atd.
- **Bezpečnostní**, což je pádnější důvod. Životnost certifikátu by měla být výrazně kratší než doba nutná k prolomení certifikovaného veřejného klíče. To je ovšem problém zejména u certifikátů certifikačních autorit, které by měly být vydávány na dobu alespoň pětkrát delší, než je životnost uživatelských certifikátů (při kratší době se silně zvyšuje režie obnovování uživatelských certifikátů).

Je třeba vždy znovu a znovu zdůrazňovat, že certifikát po vypršení doby platnosti není k nepotřebě, a tudíž nemůže být bezstarostně zahozen. Pomocí soukromého klíče příslušejícího k veřejnému klíči uvedenému v certifikátu po vypršení doby platnosti pouze nepodepisujeme nové zprávy. Avšak k ověření elektronického podpisu zpráv vytvořených v době platnosti certifikátu budeme v budoucnu vždy potřebovat i prošlé certifikáty. A budeme je potřebovat tak dlouho, dokud se ověřování bude provádět.

Obdobně, pokud si budeme archivovat zprávy zašifrované veřejným klíčem z certifikátu, pak k jejich dešifrování opět budeme potřebovat soukromý klíč k certifikátu, který byl v době vytvoření zprávy platný. Při zpracování zprávy je proto praktické zprávu dešifrovat a před ukládáním do archivu ji případně znovu šifrovat, ale tentokrát šifrovacím klíčem archivu.

Položky Vydavatel a Předmět

Položka Vydavatel (*Issuer*) specifikuje toho, kdo certifikát vydal, tj. certifikační autoritu. Položka Předmět (*Subject*) specifikuje držitele certifikátu.

* Norma X.509 verze 4 již strukturu certifikátu nemění, proto též využívá v položce *Version* hodnotu 2.

Obě položky Vydavatel i Předmět používají stejný datový formát označovaný jako jedinečné jméno (*Distinguished Name*).

Jedinečné jméno

Jedinečné jméno (*Distinguished Name*) bylo původně zavedeno v normách ITU řady X.500, konkrétně v normě X.501. Cílem norem řady X.500 je vytvořit celosvětovou adresářovou strukturu. Adresářem se přitom nerozumí adresář souborů, ale adresář jako seznam adres v telefonním seznamu. Cílem je tak vytvořit celosvětovou obdobu telefonního seznamu. Jeden záznam v takovém seznamu pak odpovídá jedinečnému jménu.

Jedinečné jméno by v takovém seznamu bylo tvořeno dílčími informacemi o tomto subjektu: názvem země, názvem telefonní společnosti, telefonního obvodu, jménem, adresou a konečně telefonním číslem. Takovou konkrétní dílčí informaci, ze které se skládá jedinečné jméno, nazýváme relativním jedinečným jménem (*Relative Distinguished Name*).

Jedinečné jméno je tak tvořeno posloupností relativních jedinečných jmen. Přitom v jedinečném jméně se mohou i relativní jedinečná jména opakovat (např. s jinou hodnotou).

Samotné relativní jedinečné jméno je množina atributů. Atribut je pak dvojice tvořená identifikátorem objektu (např. Country, Organization, Common Name apod.) a hodnotou (např. CZ). Relativní jedinečné jméno zapisujeme např.:

```
Common Name=Libor Dostalek
```

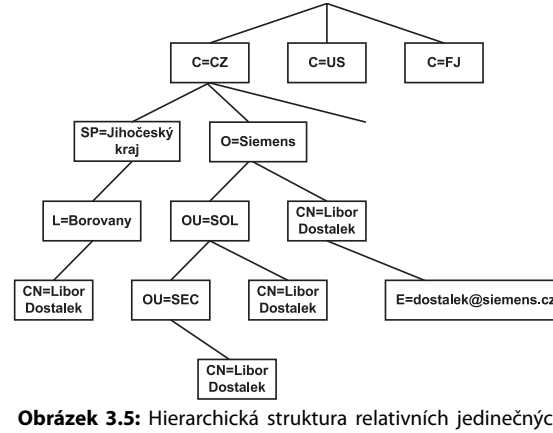
Jedinečné jméno popisující jedince je pak sekvencí relativních jedinečných jmen. Např.:

```
CommonName=Libor Dostalek, Organization=Siemens, Country=CZ
```

Tento zápis se často zkracuje pomocí zkratek pro identifikátory objektů relativních jedinečných jmen:

```
CN=Libor Dostalek, O=Siemens, C=CZ
```

I když relativní jedinečné jméno je množinou atributů, v praxi bývá tato množina jednoprvková, tj. obsahuje jen jeden atribut (jednu dvojici identifikátor + hodnota). Jedinečná jména jsou tvořena vždy větví ve stromu relativních jedinečných jmen (obr. 3.5).



Obrázek 3.5: Hierarchická struktura relativních jedinečných jmen

Zajímavé je, že Libor Dostálek může být ve struktuře uveden mnoha způsoby. (Pokaždé se jedná o jiné jedinečné jméno!)

Např.:

- Jako obyvatel ČR, konkrétně Borovan:

  ```
  CN=Libor Dostalek, L=Borovany,SP=Jihočeský kraj C=CZ
  ```

- Jako zaměstnanec firmy Siemens:

  ```
  CN=Libor Dostalek,O=Siemens,C=CZ
  ```

Uvedli jsme, že relativní jedinečné jméno se zpravidla skládá z jedné dvojice identifikátor objektu a hodnota (např. `C=CZ`). Jedinečné jméno je posloupnost tvořená těmito dvojicemi.

Jednotlivé prvky této posloupnosti se oddělují čárkou. Jenže jak se zapíše, když by teoreticky bylo relativní jedinečné jméno tvořeno dvěma dvojicemi? Takové dvojice se oddělí znakem plus. Např.:

```
OU=Siemens+CN=Libor Dostálek,C=CZ
```

specifikuje jedinečné jméno tvořené dvěma relativními jedinečnými jmény. Přičemž první obsahuje dvě dvojice (dva atributy `OU=Siemens` a `CN=Libor Dostálek`).

Jestliže se blíže zajímáte o to, jak zapsat jedinečné jméno jako textový řetězec, doporučuji vám prohlédnout si krátkou normu RFC-4514, která tuto problematiku řeší pro LDAP.

Přehled atributů relativních jedinečných jmen používaných PKI:

Atribut	Zkratka	Význam
Common Name	**CN**	**Název objektu, pod kterým je místně znám. Např. u osob to může být jméno a příjmení. U serverů pak jejich DNS-jméno apod.**
Surname	**SN**	**Příjmení**
Country	**C**	**Stát podle ISO 3166, tj. podle stejné normy, jaká se používá pro top level domény DNS (CZ = Česká republika, SK = Slovensko, FJ = Fidži...)**
Locality	**L**	**Lokalita (např. město)**
State or Province	**SP nebo ST**	**Nižší organizační jednotka státu (např. kraj či spolková země)**
Organization	**O**	**Název firmy**
Organizational Unit	**OU**	**Organizační jednotka (např. oddělení)**
Title	**T**	**Pozice (např. hejtman, jednatel společnosti, ředitel apod.)**
Name		**Jméno**
Given Name	**G**	**Jméno**
Initials		**Iniciály**
Generation Qualifier		**Např. „Jr." či „IV" pro Karel IV.**
DNQualifier		**Slouží k rozlišení různých certifikovaných objektů, kterým by jinak vycházel stejný předmět.**
Serial Number	**-**	**Slouží k rozlišení různých certifikovaných objektů, kterým by jinak vycházel stejný předmět. (Nezaměňovat se sériovým číslem certifikátu.)**
Pseudonym	**P**	**Pseudonym**
E-mail Address	**E**	**Adresa elektronické pošty (dle RFC-822).**
Domain Component	**DC**	**Jednotlivé řetězce z doménového jména (např. domény Windows). Např. domain Controller www.cpress.cz je DC = www, DC = cpress, DC = cz. Pozor! Tento atribut nemá přímou souvislost s DNSname – jedná se o jiný prostor jmen.**

Pomocí jedinečného jména specifikujeme osobu, systém či obecně nějakou entitu. Zajímavá situace je u jmen fyzických osob. Různé země mají své zvyklosti pro pojmenování svých občanů, proto konkrétní využití jednotlivých atributů závisí na certifikační politice konkrétní certifikační autority.

Uživatel uvede své jedinečné jméno do žádosti o certifikát a certifikační autorita toto jedinečné jméno zkopíruje do certifikátu. Obecně by certifikační autorita neměla modifikovat jedinečné jméno při kopírování z žádosti o certifikát do certifikátu. Výjimkou jsou pouze atributy *DNQualifier* a *SerialNumber*, ty naopak bude certifikační autorita s největší pravděpodobností do certifikátu doplňovat. Tyto dva atributy totiž slouží k rozlišení dvou různých osob, které by jinak měly stejné jedinečné jméno.

Rozdíl mezi *DNQualifier* a *SerialNumber* je v tom, že atributem *DNQualifier* rozlišujeme osoby, které by měly shodou okolností jinak stejný předmět. Kdežto atributem *SerialNumber* můžeme

rozlišit dva certifikáty téže osoby. Např. má-li osoba vydány dva podepisovací certifikáty jiné bezpečnostní úrovně: jeden má soukromý klíč např. na disku a druhý na čipové kartě. Avšak používání *DNQualifier* a *SerialNumber* není ustálené. Certifikační autority často *DNQualifier* nepoužívají a atribut *SerialNumber* pak použijí pro rozlišení osob. Konkrétní význam atributů *DNQualifier* a *SerialNumber* tak musíme hledat v příslušné certifikační politice konkrétní certifikační autority.

Dalším častým zvykem certifikačních autorit je přesunutí atributu *E-mail Address* z předmětu žádosti o certifikát do příslušného rozšíření certifikátu, protože dnešní standardy přímo vyzývají certifikační autority, aby atribut *E-mail Address* neuváděly v předmětu certifikátů.

Vydavatel certifikátu

Položka Vydavatel (*Issuer*) obsahuje jedinečné jméno certifikační autority. Je třeba, aby certifikační autorita měla jednoznačnou identifikaci (jedinečné jméno) v rámci všech certifikačních autorit.

Útočník bude mít snahu vytvořit certifikační autoritu stejného jména, ale za využití své podvržené dvojice veřejný/soukromý klíč. Zejména pokud naše certifikační autorita používá kořenový certifikát, musíme být na jeho distribuci obzvláště opatrní.

Předmět certifikátu

Pokud používáme certifikáty dle X.509 verze 3, musí být předmět certifikátu jedinečný v rámci všech objektů certifikovaných danou certifikační autoritou. Tj. certifikační autorita nesmí vydat dvěma různým osobám certifikát se stejným předmětem. Na druhou stranu je velice praktické, že certifikační autorita může vydávat jedné osobě certifikáty se stále stejným předmětem (stejným jedinečným jménem). Tj. Václav Vopička může mít více různých certifikátů se stejným předmětem, protože se jedná o stejného Václava Vopičku. Ale jeho jmenovec, který se jen shodou okolností také jmenuje Václav Vopička, musí mít jiný předmět. Může mít např. jinou lokalitu (město), ale kdyby všechny ostatní údaje byly stejné, pak CA použije k rozlišení jedinečné jméno *dnQualifier* či jedinečné jméno *serialNumber* (nezaměňovat s číslem certifikátu – to musí být v každém případě různé).

V předmětu certifikátu zpravidla využíváme širší paletu atributů jedinečných jmen než u jedinečného jména vystavitele, kde bychom měli být střídmí, i když software má podporovat nejrůznější atributy.

Mnohé identifikační údaje, které se „nevejdou“ do předmětu certifikátu, je možno uložit do rozšíření certifikátu.

Zajímavostí je, že předmět certifikátu může být i prázdný (prázdná sekvence relativních jedinečných jmen). V takovém případě ale certifikát musí povinně obsahovat rozšíření Alternativní jméno předmětu, které musí být označeno jako závažné.

Veřejný klíč

Položka Veřejný klíč (*Subject Public Key*) je sekvencí dvou informací: identifikátorem algoritmu, pro nějž je veřejný klíč určen, a samotným veřejným klíčem.

Zmíněný algoritmus na rozdíl od položky Algoritmus podpisu (*Signature Algorithm*) specifikuje algoritmus, pro který je určen certifikovaný veřejný klíč.

Např. chce-li Bohumila certifikovat své veřejné Diffie-Hellmanovo číslo, položka Veřejný klíč obsahuje identifikátor pro Diffie-Hellmanův algoritmus a Bohumilino veřejné Diffie-Hellmanovo číslo. Certifikační autorita stvrzující svým podpisem platnost Bohumilina certifikátu použije pro digitální podpis certifikátu např. algoritmus RSA. Certifikační autorita pak do položky Algoritmus podpisu vyplní identifikaci algoritmů použitých pro podpis samotného certifikátu (např. *RSA with SHA-1*).

Rozšíření certifikátu

To, co se nevešlo do předchozích položek certifikátu, se snažíme uložit do některého z rozšíření. Neměli bychom to však přehánět. Zásada je taková, že do certifikátu vkládáme informace týkající se identifikace držitele certifikátu. Někteří IT architekti se snaží do certifikátu vyznačit i role držitele certifikátu v aplikacích včetně přístupových práv. K tomuto účelu ale slouží atributové certifikáty.

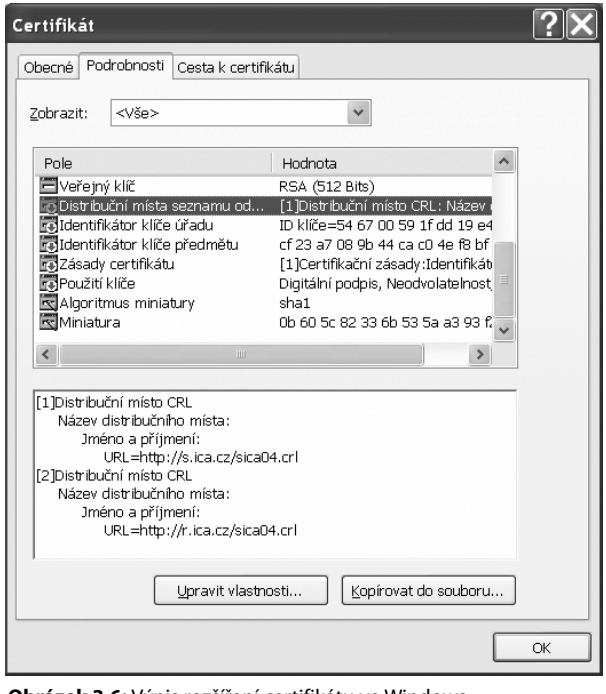
I když rozšíření certifikátu je definováno zcela obecně, je i u něj potíž (podobně jako u atributů předmětu certifikátu) spočívající v tom, že některým rozšířením nebude aplikace rozumět – nebude vědět, k čemu toto konkrétní rozšíření slouží. Tento problém řeší položka závažnost rozšíření (*critical*). Tato položka sděluje, je-li rozšíření závažné či nikoliv. V případě, že je položka závažnost nastavena na TRUE, je rozšíření označeno jako závažné.

Software pracující s certifikátem musí rozumět všem závažným rozšířením – musí si být vědom závažnosti informací v nich uvedených. V případě, že některé z rozšíření v certifikátu je označeno jako závažné a software neví, k čemu toto rozšíření slouží, musí celý certifikát odmítnout.

RFC-5280 specifikuje standardní rozšíření uvedené v následující tabulce. Zajímavé je, že ne každé z těchto standardních rozšíření musí být podporováno konkrétními aplikacemi. Některá dokonce RFC-5280 vůbec nedoporučuje používat.

	Rozšíření certifikátu	**V certifikátech CA**	**V certifikátech koncových uživatelů**
Standardní internetová rozšíření	**Identifikátor klíče úřadu** *(Authority Key Identifier)*	**Povinné ve všech certifikátech, které nejsou kořenové. Nesmí být označeno jako závažné**	
	Identifikátor klíče předmětu *(Subject Key Identifier)*	**Povinné; nesmí být závažné**	**Mělo by být; nesmí být závažné**
	Použití klíče *(Key Usage)*	**Povinné v certifikátech, pomocí kterých se verifikuje elektronický podpis certifikátů a CRL; mělo by být závažné**	
	Rozšířené použití klíče *(Extended Key Usage)*	**Je povinné pro některé certifikáty (TSA, DVCS apod.)**	
	Platnost soukromého klíče *(Private Key Usage Period)*		
	Certifikační politiky, též někdy Zásady certifikátu *(Certificate Policies)*	**Volitelné**	
	Mapování zásad *(Policy Mappings)*	**Jestliže se použije, pak by mělo být závažné**	**–**
	Subject Directory Attributes	**Jestliže se použije, pak nesmí být závažné**	
	Alternativní jméno předmětu *(Subject Alternative Name)*	**Certifikát CA nemůže mít prázdný předmět, proto toto rozšíření je volitelné a nemusí být závažné**	**Jen v případě, že certifikát má prázdný předmět, tak musí být použito a navíc označeno jako závažné**

	Rozšíření certifikátu	V certifikátech CA	V certifikátech koncových uživatelů
	Alternativní jméno úřadu *(Issuer Alternative Name)*	**Nemělo by být závažné, aplikace jej nemusí rozeznávat**	
	Základní omezení *(Basic Constraints)*	**Musí být použito, a to jako závažné**	–
	Omezení jmen *(Name Constraints)*	**Je-li použito, pak jako závažné**	–
	Omezení politik *(Policy Constraints)*	**Je-li použito, mělo by být závažné**	–
	Distribuční místa seznamu odvolaných certifikátů *(CRL Distribution Points)*	**Nemělo by být závažné**	
	Omezení *Any-Policy* *(Inhibit Any-Policy)*	**Je-li použito, pak jako závažné**	–
	Nejčerstvější seznam CRL *(Freshest CRL)*	**Nesmí být závažné**	
Privátní internetová rozšíření	**Přístup k informacím úřadu** *(Authority Information Access)*	**Nesmí být závažné**	
	Přístup k informacím předmětu *(Subject Information Access)*	**Nesmí být závažné**	
Kvalifikované certifikáty	**Biometrické informace** *(Biometric Information)*	**Nesmí být závažné**	
	Qualified Certificate Statements	**Může i nemusí být závažné**	
Microsoft	**Název šablony certifikátu** *(Certificate Template Name)*		



Obrázek 3.6: Výpis rozšíření certifikátu ve Windows

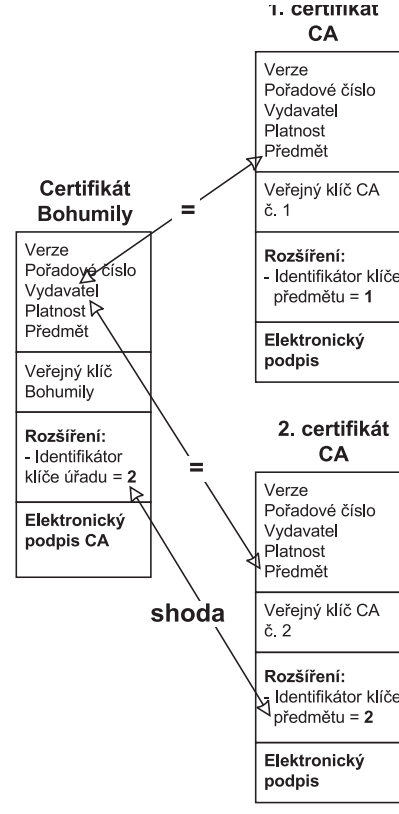
Průvodce některými rozšířeními certifikátu

Ve zbytku této kapitoly se zmíníme o některých rozšířeních certifikátu. Hlavním cílem této kapitoly je poskytnout čtenáři základ, aby byl schopen porozumět těmto rozšířením, když si zobrazí obsah certifikátu např. v MS Windows. Takový výpis je např. na obr. 3.6, kde závažná rozšíření jsou označena vykřičníkem ve žlutém trojúhelníčku, kdežto ostatní rozšíření jsou označena zelenou šipkou v bílém terčíku. Vyčerpávající popis rozšíření certifikátu obsahuje kapitola 15.

Identifikátor klíče předmětu a Identifikátor klíče úřadu

Jak jsme se již zmínili, táž osoba může mít vydáno více certifikátů. V tomto smyslu je osobou i sama certifikační autorita. Co kdybychom tedy měli např. certifikační autoritu, která by měla dva certifikáty se stejným předmětem, ale s různými veřejnými klíči?

Pokud by Alois chtěl ověřit certifikát Bohumily, narazil by. Nevěděl by totiž, který z certifikátů certifikační autority zvolit pro ověření certifikátu Bohumily. Možná si řeknete, že je to jednoduché: nejprve použije jeden, a když verifikace selže, tak druhý. Ale co když selže i druhý certifikát CA? Co to znamená? Je certifikát Bohumily podvržen nebo naše milá certifikační autorita má ještě třetí certifikát, vhodný pro verifikaci certifikátu Bohumily?



Obrázek 3.7: Alois nalezne správný certifikát CA podle shody v obsahu rozšíření Identifikátor klíče předmětu a Identifikátor klíče úřadu

A právě rozšíření Identifikátor klíče předmětu spolu s rozšířením Identifikátor klíče úřadu nám řeší tento problém. Certifikační autorita si označí své jednotlivé veřejné klíče (např. je očísluje, ale praktičtější je klíče identifikovat otiskem z nich). Do svého certifikátu pak certifikační autorita doplní rozšíření Identifikátor klíče předmětu, do něhož uvede označení svého veřejného klíče.

Jestliže certifikační autorita vydává certifikát svým uživatelům, do každého vydaného certifikátu uvede rozšíření Identifikátor klíče úřadu, do něhož vloží označení veřejného klíče CA, který se má použít pro verifikaci tohoto certifikátu.

Vraťme se k Aloisovi. Když Alois prohledává své úložiště důvěryhodných certifikačních autorit, aby nalezl veřejný klíč pro ověření certifikátu Bohumily, pak vždy, když vyhledá certifikát certifikační autority, který má Předmět shodný s položkou Vydavatel Bohumilina certifikátu, ještě navíc zkontroluje, jestli se shoduje obsah položky Identifikátor klíče úřadu v Bohumilině certifikátu s obsahem položky Identifikátor klíče předmětu v příslušném certifikátu certifikační autority.

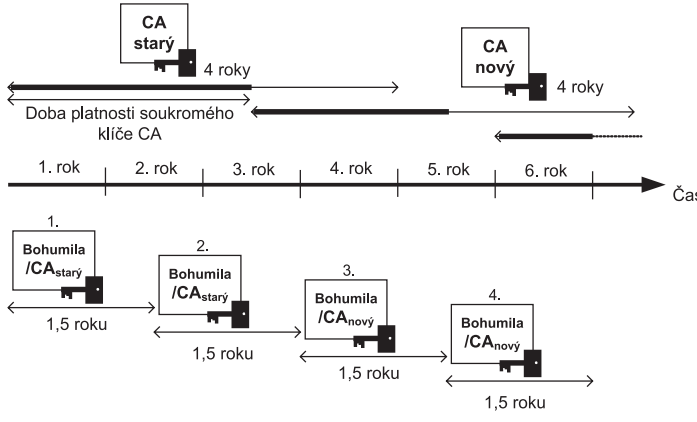
Jinými slovy: Dvojice rozšíření Identifikátor klíče předmětu a rozšíření Identifikátor klíče úřadu slouží k identifikaci klíče certifikační autority, kterým byl certifikát podepsán. To je důležité zejména v případě, že certifikační autorita má více dvojic veřejný/soukromý klíč. Pokud vám připadá zbytečné, aby certifikační autorita měla více dvojic klíčů, jen jste si neuvědomili, že i certifikát certifikační autority má svou dobu platnosti. Tj. i certifikáty certifikačních autorit je třeba obnovovat. Po jistou dobu tak má certifikační autorita certifikáty dva: starý a nový (později uvidíme, že správně by měla mít po tuto dobu ještě další dva: starý-nový a nový-starý).

Rozšíření Identifikátor klíče předmětu může být užitečné i v případě certifikátů koncových uživatelů. Obsahem tohoto rozšíření lze totiž efektivně identifikovat certifikát. V dalších kapitolách se pak setkáme s tím, že certifikát je buď identifikován dvojicí Vydavatel + Pořadové číslo certifikátu nebo právě obsahem rozšíření Identifikátor klíče předmětu, jehož velkou výhodou je pevná délka.

Platnost soukromého klíče

Toto rozšíření umožňuje vyznačit kratší dobu platnosti soukromého klíče, než je doba platnosti celého certifikátu, což umožňuje takto označeným klíčem digitálně podepisovat např. 1 rok, kdežto ověřovat můžeme tento podpis bez potíží např. 5 let.

Zastavme se nyní u trochu jiné otázky: Proč by při obnově certifikátu certifikační autority měly po jistou dobu platit oba certifikáty?



Obrázek 3.8: Platnost soukromého klíče

Představme si, jak pracuje Bohumilina oblíbená certifikační autorita (obr. 3.8), jejíž certifikáty mají platnost 4 roky a svým uživatelům (např. Bohumile) vystavuje certifikáty nejvýše na 1,5 roku. V horní části obrázku je znázorněna platnost certifikátů certifikační autority a ve spodní části pak platnost uživatelských certifikátů.

Zaměřme se na obr. 3.8. Certifikační autorita vydá Bohumile první certifikát. Když se blíží vypršení tohoto certifikátu, vydá certifikační autorita Bohumile druhý certifikát. Oba dva certifikáty se ověřují certifikátem certifikační autority CA-starý (*CAold*). Když se blíží vypršení platnosti druhého Bohumilina certifikátu, chce si Bohumila opět obnovit svůj certifikát. Může jí certifikační autorita vydat certifikát, který se ověřuje certifikátem CA-starý? Nemůže! Pokud by to totiž udělala, po jednom roce platnosti Bohumilina certifikátu by byl tento certifikát sice formálně platný, ale jeho ověření by selhalo, jelikož certifikát CA-starý by již byl neplatný.

Závěr je tedy takový, že certifikační autorita si musí obnovit svůj certifikát nejpozději tak dlouho před vypršením svého starého certifikátu, na jak dlouho vydává certifikáty svým uživatelům. CA totiž nemůže vydat uživateli certifikát podepsaný klíčem CA, který by v době platnosti vydaného certifikátu vypršel. Tj. nastala by situace, že uživatel má sice platný certifikát, který je však podepsán neplatným (expirovaným) certifikátem.

CA tak má poměrně dlouhou dobu dva certifikáty, které se překrývají. Někteří uživatelé mají svůj certifikát podepsán „starým" certifikátem CA a jiní „novým" certifikátem CA. Oba certifikáty CA budou mít stejný předmět. Budou se lišit pořadovým číslem a veřejným klíčem. V certifikátu uživatele je v položce Vydavatel (*Issuer*) uveden předmět z certifikátu CA, kterým je certifikát uživatele podepsán. A ten je pro nový i starý certifikát certifikační autority stejný. A opět jsme u problému, který se přece řeší pomocí rozšíření zmíněného v předchozím paragrafu – Identifikátor klíče úřadu a Identifikátor klíče předmětu.

V horní části obr. 3.8 je vyznačena „Doba platnosti soukromého klíče". Po tuto dobu je využíván klíč certifikační autority k podpisování vydávaných certifikátů. Po uplynutí této doby začne certifikační autorita využívat nový certifikát. Soukromý klíč příslušející starému certifikátu certifikační autority je po uplynutí této doby dobré zlikvidovat.

Dobu platnosti soukromého klíče je též možné uvést do stejnojmenného rozšíření, které však nebylo doporučeno využívat. RFC–5280 na rozdíl od předchozích standardů toto rozšíření nezakazuje používat (ale ani nedoporučuje). Toto rozšíření muže též omezit platnost soukromého klíče i na počátku platnosti certifikátu.

Podle RFC-5280 by se rozšíření Platnost soukromého klíče nemělo používat v certifikátech pro internetové aplikace.

Použití klíče

Pomocí tohoto rozšíření lze omezit způsob použití veřejného klíče obsaženého v certifikátu, tj. omezit použití certifikátu. Toto rozšíření obsahuje bitový řetězec. Každý bit z řetězce pak odpovídá konkrétnímu způsobu použití certifikátu. Je-li příslušný bit nastaven na TRUE, je certifikát k danému použití možno používat. (Pokud se daný bit v řetězci nevyskytuje, předpokládá se jeho nastavení na TRUE.)

Význam jednotlivých bitů:

- **Digitální podpis (Digital Signature)** – certifikát je určen k elektronickému podpisu dat (jako algoritmu). Nastavení tohoto bitu **ne**opravňuje k:
  - Ověřování pravosti (či jestli chcete nepopiratelné odpovědnosti – k tomu je určen až následující bit, *Non Repudiation*). To vás asi překvapilo.

Zřejmě si říkáte, k čemu tedy může takový certifikát sloužit. **Může** sloužit k:

- Autentizaci uživatelů.
- K ověřování integrity dat.

- **Neodvolatelnost (Non Repudiation)** – certifikát je určen k ověřování pravosti či jestli chcete nepopiratelné odpovědnosti. Názvy bitu *Digital Signature* a bitu *Non Repudiation* jsou naprosto zavádějící, a existují dokonce dva způsoby jejich interpretace:
  - Podle první interpretace bit *Digital Signature* signalizuje libovolné použití, které vychází z algoritmu digitálního podpisu, nikoliv tedy z konkrétního využití pro ověřování pravosti v případě digitálního podpisu. Bit *Non Repudiation* pak signalizuje konkrétní využití algoritmu digitálního podpisu pro ověřování pravosti. Takže chceme-li např.

využít certifikát k ověření digitálního podpisu listiny, který má nahrazovat rukou psaný podpis, musí být nastaveny oba bity. První signalizuje podporu algoritmu a druhý jeho konkrétní nasazení pro ověřování pravosti.

- Podle druhé interpretace bit *Digital Signature* signalizuje využití certifikátu k autentizaci a bit *Non Repudiation* k digitálnímu podpisu. Takže pokud má být využit certifikát k verifikaci digitálního podpisu listiny, který má nahrazovat rukou psaný podpis, musí být nastaven jen bit *Non Repudiation*, aby takový certifikát nebylo možné zneužít při autentizaci (viz obr. 1.12). Bohužel první interpretace je pravděpodobně ta pravá.

- **Zakódování klíče (Key Encipherment)** – certifikát je určen k šifrování klíčů. Klasickým případem je elektronická obálka, kdy data jsou šifrována náhodným symetrickým šifrovacím klíčem, který je ke zprávě přibalen a zašifrován právě veřejným klíčem z takto označeného certifikátu.
- **Zakódování dat (Data Encipherment)** – veřejný klíč z takto označeného certifikátu je určen pro šifrování dat (jiných než šifrovacích klíčů).
- **Key Agreement** – certifikát je určen pro výměnu klíčů (např. DH výměna klíčů).
- **Podepisování certifikátu (Key Certificate Sign)** – veřejný klíč uvedený v tomto certifikátu je určen pro verifikaci certifikátů. Tj. jedná se o certifikát certifikační autority.
- **Podepisování CRL (CRL Sign)** – veřejný klíč uvedený v tomto certifikátu je určen k verifikaci CRL.
- **Encipher Only** – Tento bit se používá ve spojení s bitem *Key Agreement* (výměna klíčů). Výsledný dohodnutý symetrický klíč může být využit pouze k šifrování.
- **Decipher Only** – Tento bit se používá ve spojení s bitem *Key Agreement* (výměna klíčů). Výsledný dohodnutý symetrický klíč může být využit pouze k dešifrování.

Rozšíření se označí jako závažné. Tím se zamezí použití certifikátu k jiným účelům než k účelům vyznačeným v certifikátu.

Rozšířené použití klíče

Je obecnějším řešením pro určení účelů, k jakým je certifikát určen. Toto rozšíření může obsahovat sekvenci identifikátorů objektů specifikujících způsoby konkrétního použití veřejného klíče.

Toto rozšíření je předepsáno používat u některých dalších autorit. Např. autority pro vydávání časových razítek (TSA), OCSP serveru, DVCS, vyžadují, aby certifikáty jejich serverů měly pomocí tohoto rozšíření explicitně vyjádřeno, že se mohou používat k tomuto účelu.

Alternativní jméno předmětu

Toto rozšíření umožňuje vložit do certifikátu další jedinečná jména držitele certifikátu: např. jedinečné jméno ve světě e-mailové komunikace (e-mailovou adresu); jedinečné jméno v DNS světě (DNS jméno), důležitém zejména pro certifikáty počítačů; další jedinečné jméno stejného tvaru, jaký se používá pro předmět certifikátu (*Directory name*), atd. Důležité je, že alternativních jmen může být uvedeno i více.

Při vydávání certifikátu nesmí být opomenuta ani kontrola údajů uvedených v tomto rozšíření.

Měli bychom upustit od zlozvyku uvádět adresy elektronické pošty a DNS jména do předmětu certifikátu, ale uvádět je výhradně zde v rozšíření Alternativní jméno předmětu.

Může zde být uvedeno:

- **Jiný název (Other Name)** – jiný identifikační údaj,
- **Název RFC822 (rfc822 Name)** – adresa elektronické pošty dle RFC-822 (např. `dostalek@siemens.com`),
- **DNS Name** – DNS jméno (např. jméno serveru `www.firma.cz`),
- **X.400 Address** – adresa elektronické pošty podle norem řady X.400,
- **Directory Name** – adresářové jméno podle norem řady X.500, tj. má stejný formát, jako má předmět nebo vydavatel certifikátu,
- **EDI Party Name** – jméno podle norem EDI,
- **Uniform Ressource Identifier** – URI (např. `http://www.firma.cz`),
- **IP Address** – pro IPv4 obsahuje přesně 4 bajty IP-adresy verze 4, pro IPv6 obsahuje 16 bajtů IP-adresy verze 6, tj. jednotlivé bajty nejsou odděleny oddělovači ani převedeny do desítkové soustavy,
- **Registered ID** – identifikátor objektu.

Certifikační politiky (certifikační zásady)

Toto rozšíření obsahuje identifikátor dokumentu Certifikační politika. Navíc může obsahovat i hypertextový odkaz na tento dokument, který není součástí certifikátu. Tj. do výpočtu digitálního podpisu certifikátu je zahrnuto pouze toto rozšíření a nikoliv celý dokument. Není tedy zaručeno, že certifikační autorita tento dokument nezmění po vydání certifikátu.

Certifikační politika je dokument specifikující postupy, praktiky a cíle sloužící k ověření certifikátu před tím, než je použit, tj. pravidla, za kterých CA vydává certifikáty, a zejména jak za vydané certifikáty ručí. Tato pravidla jsou zpravidla sepsána v dokumentu Certifikační politika, vydaném certifikační autoritou. Na rozdíl od některých jiných dokumentů CA je certifikační politika veřejným dokumentem, zpravidla vystaveným na Internetu (na webu provozovatele certifikační autority).

Rozšíření Certifikační politiky v certifikátu:

- Není uvedeno. To je např. případ certifikačních autorit Microsoft Enterprise* (MSCA), které nepředpokládají tvorbu pracné certifikační politiky při instalaci certifikační autority. Namísto certifikační politiky vyplňují do certifikátů své privátní rozšíření: *Ceritificate template*.
- V certifikátu je identifikátor certifikační politiky a případný hypertextový odkaz na tuto politiku, která je vystavena na Internetu. Propracovaná certifikační politika je totiž o několik řádů větší, než je velikost samotného certifikátu, takže její umístění do certifikátu by bylo neefektivní.
- V certifikátu je jen prohlášení vydavatele nepřesahující 200 znaků, které nahrazuje certifikační politiku (např.: „*Pouze pro testovací účely*“).

* I certifikační autority na serverech Windows mohou být konfigurovány s funkčností „*Qualified Subordination*“, pak se i v prostředí Microsoft hraje na rozšíření „Certifikační zásady“, „Mapování zásad“, „Constrains“ apod.

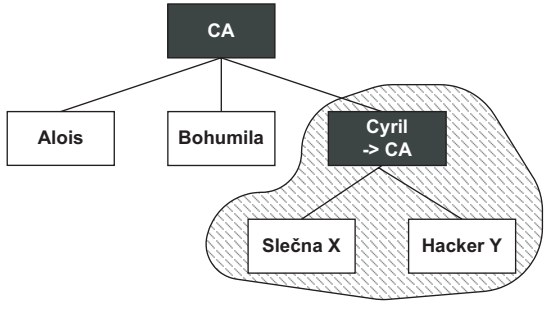
Mapování zásad

Toto rozšíření se používá pouze v certifikátech certifikačních autorit. Má význam v případě, že certifikát certifikační autority je podepsán jinou CA. V takovém případě je pravděpodobné, že nadřízená certifikační autorita si vydá jinou certifikační politiku než podřízená certifikační autorita. Při ověřování řetězce certifikátů je pak důležité, aby certifikační politiky jednotlivých certifikátů v řetězci byly konzistentní (vzájemně si odpovídaly).

Jelikož každá CA má pro své certifikační politiky své identifikátory objektu, úkolem rozšíření Mapování zásad je sdělit, že ta a ta certifikační politika vydavatele (nadřízené CA – *issuerDomainPolicy*) je srovnatelná s tou a tou certifikační politikou předmětu (podřízené CA – *subjectDomainPolicy*).

Omezení využívání certifikátu (Constrains)

CA svým elektronickým podpisem ručí za údaje uvedené ve vydaných certifikátech. Na obr. 3.9 je znázorněno nebezpečí, když CA vydává certifikáty svým uživatelům: Aloisovi, Bohumile a Cyrilovi. Cyril se v záchvatu žárlivosti svévolně prohlásí za certifikační autoritu a vydá certifikáty dalším uživatelům: slečně X a hackerovi Y. Problém je v tom, že CA nepřímo ručí i za takto vydané certifikáty pro uživatele X a Y. Např. slečna X získá řetězec certifikátů obsahující: certifikát slečny X, certifikát Cyrila a certifikát CA. Certifikát slečny X se ověřuje pomocí certifikátu Cyrila. Certifikát Cyrila se následně ověřuje pomocí certifikátu CA. Není tedy problém ověřit platnost takto vydaného certifikátu Cyrilem, ale problém je ve zodpovědnosti CA za takto vydané certifikáty.



Obrázek 3.9: Cyril se svévolně prohlásil za certifikační autoritu

Jeden mechanismus, jak takovémuto počínání uživatele zamezit, jsme již popsali pomocí rozšíření Použití klíče. Tímto rozšířením v certifikátech vydávaných uživatelům omezíme použití jejich klíče tak, že jej není možné používat k ověřování certifikátů.

Rozšíření mající v názvu české slovo „omezení" nebo anglické „*constrains*" omezuje nejen svévolným prohlášením uživatelů za certifikační autority, ale v případě, že i když cílevědomě vystavuje certifikát podřízené certifikační autoritě, omezíme její působnost výhradně na sjednané oblasti.

Základní omezení

Rozšíření Základní omezení (*Basic constrains*) umožňuje označit certifikát tak, aby bylo zřejmé, zdali se jedná o certifikát CA nebo koncového uživatele. V případě certifikátu CA umožňuje určit, kolik může mít tato certifikační autorita podřízených CA.

Toto rozšíření se využívá výhradně u certifikátů certifikačních autorit.

Omezení jmen

Certifikát certifikační autority se tímto omezuje na vydávání certifikátů uživatelům splňujícím podmínky pro jedinečná jména či alternativní jména předmětu, zadané v jejich certifikátech.

Omezovat lze např. na DNS jména patřící do domény .firma.cz. Lze rovněž omezovat jména poštovních schránek na poštovní adresy patřící určité doméně. Lze omezovat i IP-adresy apod. Toto rozšíření může být využito pouze v certifikátech CA.

Distribuční místa seznamu odvolaných certifikátů

Seznam odvolaných certifikátů (CRL) může vystavovat buď sama certifikační autorita nebo může vystavováním CRL pověřit jinou autoritu (autoritu pro vydávání CRL).

Toto rozšíření obsahuje seznam distribučních míst, na kterých je vystaven seznam odvolaných certifikátů (CRL). Distribuční místo je zpravidla reprezentováno URI.

Subject directory attributes

Jedná se o rozšíření obsahující další atributy (atributy viz jedinečné jméno). Rozšíření obsahuje sekvenci těchto atributů. Norma RFC-3739 pro kvalifikované certifikáty zavádí některé specifické atributy pro toto rozšíření.

Pomocí tohoto rozšíření lze řešit tzv. problém uživatelských práv. Problém spočívá v tom, že certifikát slouží k prokázání totožnosti držitele certifikátu (na základě faktu, že uživatel má k dispozici odpovídající soukromý klíč). Prokázání totožnosti ještě nic neříká o tom, jestli mám nějaká přístupová práva ke konkrétní aplikaci. Podobně jako na základě občanského průkazu mohu prokázat svou totožnost, ale to ještě nic nevypovídá o tom, jestli jsem např. oprávněn vstupovat do nějaké budovy. V případě občanského průkazu může být takové přístupové právo vyznačeno v občanském průkazu. Jiným řešením je vydání průkazu ke vstupu. V takovém průkazu pak jsou opsány mé identifikační údaje z občanského průkazu a dále jsou vyznačena má přístupová práva. Já se mohu prokázat občanským průkazem a následně podle průkazu ke vstupu jsem buď vpuštěn, nebo ne.

V případě certifikátů máme opět obě možnosti. Právě v rozšíření „*Subject directory attributes*" můžeme vyznačit např. přístupová práva přímo v certifikátu. Jinou možností je využít atributové certifikáty, které jsou pak obdobou dalšího průkazu.

Přístup k informacím úřadu (*Authority Information Access – AIA*)

Toto rozšíření může mít několik funkcí. V praxi se zpravidla nejčastěji využívají následující dvě možnosti:

- Máme-li ověřit platnost certifikátu, potřebujeme mít k dispozici certifikát certifikační autority, která jej vydala. Ten můžeme mít např. již předem uložen na našem počítači, ale pokud tomu tak není, dostáváme se do frapantní situace. Odkud jej vzít? A právě v rozšíření AIA může být odkaz (URL), na němž se potřebný certifikát certifikační autority nachází.
- Druhou možností je uvést do tohoto rozšíření odkaz na OCSP server, pomocí kterého můžeme OnLine zjišťovat, není-li náhodou certifikát odvolán.

Název šablony certifikátu

Certifikační autority dodávané firmou Microsoft vydávají certifikáty k různým účelům, např.: certifikáty CA, certifikáty podřízených CA, certifikát pro přihlášení uživatele do Windows, certifikát pro podpis uživatele (*UserSignature*) atd. Obecně by asi bylo možné pro každý účel napsat certifikační politiku a tu vyznačit v certifikátu. Microsoft šel jinou cestou – vytvořil výčet všech účelů. Pro každý účel ze seznamu pak má konkrétní šablonu.

Biometrické informace

Jedná se o rozšíření pro kvalifikované certifikáty. V certifikátu nejsou uloženy samotné biometrické vlastnosti držitele certifikátu, ale pouze otisky z příslušných biometrických vlastností držitele certifikátu. U příslušného otisku může být uvedeno rovněž URI, na kterém je celý biometrický údaj kompletně k dispozici.

Qualified Certificate Statements

Toto rozšíření by mělo vyjadřovat, že se jedná o kvalifikovaný certifikát. Rozšíření obsahuje sekvenci příznaků kvalifikace certifikátu.

Každý příznak se skládá z identifikátoru objektu a parametrů definovaných při zavedení tohoto identifikátoru objektů. RFC-3739 zavádí identifikaci pro registrační autority. Další příznaky by měl zavést zákon (resp. související vyhlášky) země, ve které se kvalifikované certifikáty vydávají.

Označit certifikát jako kvalifikovaný je též alternativně možné pomocí rozšíření Certifikační politiky. V certifikační politice pak napíšeme, že se jedná o kvalifikované certifikáty.

Kvalifikované certifikáty

Kvalifikovaný certifikát je zvláštní typ certifikátu, které používá ve své legislativě Evropská unie. Zvláštní není ani svou syntaxí (ta ve své podstatě vychází ze syntaxe certifikátu popsaného v předchozí kapitole, tj. certifikátu dle RFC-5280). Zvláštnost se týká právní oblasti. Cílem je po právní stránce nahradit rukou psaný podpis elektronickým podpisem, který se ověřuje právě kvalifikovaným certifikátem. Jádro myšlenek ohledně kvalifikovaných certifikátů je uvedeno v RFC-3739.

Kvalifikovaný certifikát obsahuje identifikaci držitele certifikátu založenou na oficiální identifikaci člověka nebo na jeho pseudonymu. Certifikační autorita vždy zná konkrétní osobu, které certifikát vydala.

Předmět certifikátu musí být jednoznačný pro konkrétní osobu, tj. dvě různé osoby nemohou mít vydán certifikát se shodným předmětem. Tato podmínka musí být splněna po celou dobu existence konkrétní CA. Pro docílení této podmínky je možné použít atribut *serialNumber* (nezaměňovat s položkou sériové číslo certifikátu). Kdyby dvě osoby měly mít stejné předměty, pak se odliší hodnotou v položce *serialNumber*.

U kvalifikovaných certifikátů nestačí, aby byl pouze předmět jednoznačný pro konkrétní osobu, ale certifikační autorita nesmí vydat dvěma různým osobám certifikát, který by měl stejný veřejný klíč. Tj. certifikační autorita musí po dobu své existence archivovat i veřejné klíče, které uživatelům podepsala. U veřejných klíčů musí mít i informaci, pro jaké algoritmy se budou používat, aby mohla porovnávat klíče, zdali již nejsou použité.

Viz též kap. 10.

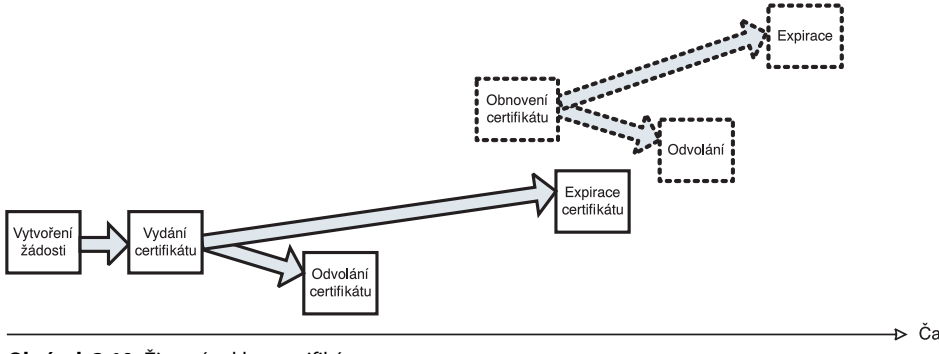
Životní cyklus certifikátu

Certifikát v průběhu času prochází několika fázemi, tvořícími životní cyklus certifikátu (obr. 3.10). Životní cyklus certifikátu se skládá z následujících fází:

1. **Vytvoření žádosti o certifikát** – vytvoření žádosti může, ale i nemusí předcházet generování párových dat (je to sice málo běžné, ale párová data může generovat až certifikační autorita po obdržení žádosti o certifikát).
2. **Vydání certifikátu** a jeho případná publikace.
3. **Platnost certifikátu** – poté co byl certifikát vydán, nemusí být ještě automaticky platný. Platnost certifikátu začíná v době uvedené v položce „od" (*Not Before*) a končí buď vypršením platnosti certifikátu nebo odvoláním certifikátu.
4. **Vypršení platnosti certifikátu** (expirace certifikátu) nastane po uplynutí doby „do" (*Not After*) uvedené v certifikátu.
5. **Odvoláním certifikátu** před uplynutím jeho původně deklarované doby platnosti. Certifikát odvolává certifikační autorita zpravidla tím, že identifikaci certifikátu zveřejní na seznamu odvolaných certifikátů (CRL). Odvolaný certifikát se uvádí na všech CRL po dobu jeho původní platnosti.

   Certifikační autorita odvolává certifikát:

   - Buď ze svého rozhodnutí, např.:
     - Jiný uživatel požádal o certifikaci již certifikovaného veřejného klíče.
     - Certifikační autorita zjistila, že údaje v certifikátu nadále nejsou pravdivé (např. zaměstnanec rozvázal pracovní poměr a vlastní zaměstnanecký certifikát, tj. certifikát s uvedeným atributem „Organization").
   - Nebo na žádost držitele certifikátu, např.:
     - Uživatel si již nepřeje, aby certifikát dále platil z osobních důvodů.
     - Byl kompromitován soukromý klíč uživatele.
     - Byl zničen soukromý klíč uživatele.



Obrázek 3.10: Životní cyklus certifikátu

Existuje i možnost pozastavení platnosti certifikátu.

Je-li certifikát ještě platný, můžeme jej využít k obnovení certifikátu. Co to znamená? Žádáme-li o vydání certifikátu, musíme zpravidla prokázat svou totožnost, aby certifikační autorita

mohla ověřit údaje v žádosti o certifikát. Pakliže máme platný certifikát, certifikační autorita již ověřila naši totožnost, a tak stačí, když využijeme platný certifikát k opětovnému prokázání naší totožnosti. Pokud jsme se např. v případě vydání prvního certifikátu museli osobně dostavit k prokázání totožnosti, v případě obnovení certifikátu to už zpravidla není nutné – prokážeme se elektronicky za využití platného certifikátu.

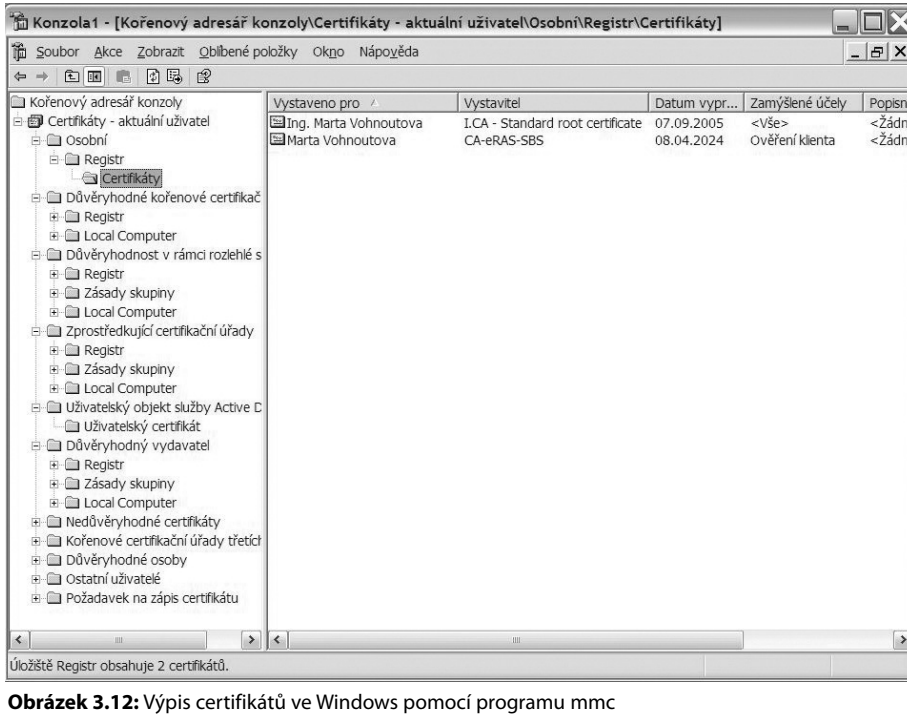
Certifikační autorita zpravidla uvede stejný předmět obnoveného certifikátu, jako měl původní certifikát. (Přesné podmínky, jak certifikační autorita postupuje při vydávání a obnovování certifikátu, ale najdeme v certifikační politice příslušné certifikační autority.)

Obdobná situace nastane i v případě, jestliže používáme více certifikátů současně (např. šifrovací, podpisový, autentizační apod.); v tomto případě stačí, když je nám osobně vydán např. podpisový certifikát a ostatní certifikáty si vydáme jako další certifikáty s tím, že totožnost již nemusíme prokazovat osobně, ale postačí ji prokázat elektronicky, např. na bázi digitálního podpisu pomocí platného podpisového certifikátu.

Certifikát ve Windows

Ve Windows zpravidla bývají přípony souborů .cer, .crt, .der atp. asociovány s manažerem certifikátů. Po klepnutí na soubor s touto příponou se zobrazí obsah certifikátu (viz obr. 3.6).

Jiným řešením, jak zobrazit certifikáty v systému Windows, je spustit program `mmc`. Volbou „Přidat nebo odebrat modul snap-in" přidáme modul „Certifikáty", kde jsou zobrazena úložiště certifikátů aktuálně přihlášeného uživatele (viz obr. 3.12). Kromě toho si správce může zobrazit i úložiště certifikátů místního počítače a úložiště služeb, která jsou důležitá zejména pro servery běžící na tomto počítači.



Obrázek 3.12: Výpis certifikátů ve Windows pomocí programu mmc

Na obr. 3.12 vidíme jednotlivá úložiště certifikátů:

- **Osobní** – obsahuje osobní certifikáty aktuálně přihlášeného uživatele. Důležité je, že k osobním certifikátům zpravidla budeme mít i příslušné soukromé klíče.
- **Důvěryhodné kořenové certifikační autority** – obsahuje důvěryhodné kotvy.
- **Důvěrnost v rámci rozlehlé sítě** – obsahuje CTL (*Certificate Trusted List*).
- **Zprostředkující certifikační autority** – obsahuje certifikáty mezilehlých certifikačních autorit, tj. certifikáty CA, které nejsou kořenové.
- **Uživatelský objekt služby Active Directory** – obsahuje certifikáty, které má aktuální uživatel registrovány v ActiveDirectory.
- **Ostatní uživatelé** – obsahuje certifikáty ostatních uživatelů, které byly vydány důvěryhodnými CA. Toto úložiště může též využívat elektronická pošta pro vyhledávání certifikátů adresátů.
- **Požadavek na zápis certifikátu** – obsahuje žádosti o certifikáty.

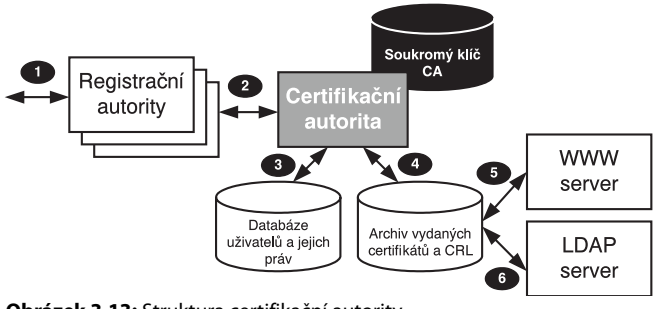
V případě osobních certifikátů zpravidla budeme mít k certifikátu i soukromé klíče. Soukromé klíče se v operačních systémech Microsoft ukládají:

- Na disk. V tomto případě mohou být uloženy lokálně nebo jako součást uživatelského profile (ActiveDirectory).
- Na čipovou kartu, USB token či podobné zařízení – viz též Čipové karty a Mini klíč.

V případě certifikátů počítače (serveru) se soukromé klíče ukládají do obdobných úložišť. Mohou být uloženy na disku serveru nebo v HSM modulu – viz též odstavec HSM.

Certifikační a registrační autority

Certifikační autorita (CA) je nezávislá třetí strana, která vydává certifikáty. Slovní spojení „certifikační autorita“ lze ale chápat dvojím způsobem: buď jako aplikaci (vydávající certifikáty) nebo jako instituci (zajišťující proces vydávání certifikátů). Jako instituce může být CA realizována jako samostatná firma nebo jako samostatný útvar v rámci firmy.



Obrázek 3.13: Struktura certifikační autority

Certifikační autorita jako instituce se skládá z několika základních částí:

1. Registračních autorit (RA), které jsou často realizovány podobně jako bankovní přepážky. Na RA se dostavují žadatelé o certifikáty se svými žádostmi. RA mohou ověřovat totožnost žadatelů. RA následně zprostředkují vydání certifikátu. Vydaný certifikát

bývá skrze RA předán žadateli (ze kterého se tím stane držitel certifikátu). RA mohou být též organizovány i jako servery a uživatel s nimi komunikuje výhradně elektronicky.

2. Jádrem CA je certifikační autorita jako aplikace vydávající certifikáty, které jsou elektronicky podepisovány soukromým klíčem CA. Soukromý klíč CA je tak největším aktivem CA, které je nutné odpovídajícím způsobem chránit. Pro ochranu soukromého klíče CA se často využívají HSM.
3. CA udržuje databázi uživatelů a auditní záznamy o činnosti CA včetně případných účtovacích informací pro fakturaci poskytovaných služeb.
4. CA udržuje archiv vydaných certifikátů a CRL.
5. Archiv vydaných certifikátů a CRL může být dostupný skrze webové rozhraní.
6. Archiv vydaných certifikátů a CRL může být dostupný skrze protokol LDAP.

Kromě vydávání certifikátu by měla certifikační autorita zajišťovat též mechanismus odvolávání certifikátů.

Kapitola 4

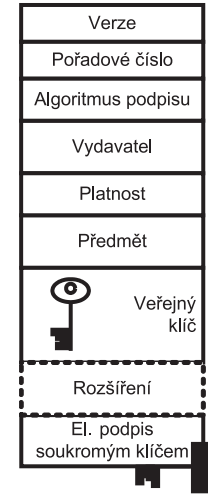
Žádost o certifikát

Na obrázku (obr. 4.1) je schematicky znázorněna datová struktura certifikátu. Jednotlivé položky certifikátu musí certifikační autorita řádně naplnit před tím, než výsledek digitálně podepíše. O certifikát může žadatel žádat dvojím způsobem: Buď předloží datovou strukturu nazývanou žádost o certifikát nebo nikoliv.

Zabývejme se nejprve druhým případem, kdy žadatel žádnou datovou strukturu nepředkládá a certifikační autorita vše zařídí za něj. Tento postup využívají některé firmy na svých intranetech pro vystavování certifikátů svým zaměstnancům.

Zajímavý je případ některých členských zemích EU, které takto vystavují dokonce občanské průkazy. Jak certifikační autority postupují, když žadatelé nepředkládají žádost o certifikát v elektronické podobě? Žadatel se dostaví na úřad, předloží své osobní údaje, tj. prokáže svou totožnost, a žádá o vystavení nového občanského průkazu.

Během výroby občanského průkazu jsou za žadatele vygenerována párová data a je mu vystaven certifikát. Uživatel následně obdrží občanský průkaz ve formě čipové karty s párovými daty i vystavenými certifikáty. Zvenku karta nahrazuje stávající tištěné občanské průkazy a uvnitř je čip se soukromými klíči a vydanými certifikáty. Jedná se v podstatě o obdobu vydávání platebních karet. Tento způsob vyžaduje vysoké nároky na bezpečnost zařízení, na kterých se generují párová data, i maximální bezpečnost všech procesů manipulujících s kartou, než se karta dostane do ruky svému oprávněnému držiteli.



Obrázek 4.1: Struktura certifikátu

Dále se budeme zabývat již jen případem, kdy žadatel předkládá elektronickou žádost o certifikát. Žádost o certifikát pak obsahuje informace (nebo jejich části), které by si žadatel přál mít uvedeny v certifikátu. Je pak na certifikační politice CA, jestli obsah těchto položek zkopíruje (nebo nezkopíruje) do certifikátu. Nakonec některé informace do certifikátu doplní certifikační autorita ze své iniciativy a vydá kýžený certifikát.

Máme několik možností digitální žádosti o certifikát: PEM, PKCS#10, CRMF, kořenový certifikát a SPK.

Údaje v žádosti o certifikát

Žádost o certifikát by měla obsahovat:

- **Identifikační údaje žadatele** – ve výsledném certifikátu budou uvedeny v předmětu certifikátu nebo v rozšíření Alternativní jméno předmětu.
- **Veřejný klíč** včetně příslušné identifikace asymetrického algoritmu, ke kterému je veřejný klíč určen.

- **Důkaz o držení příslušného soukromého klíče.**
- **Další údaje**, které si přeje uživatel do certifikátu vložit (např. použití klíče, rozšířené použití klíče apod.).
- Může obsahovat **důkaz o generování párových dat** bezpečným zařízením (např. čipovou kartou). Pro tento důkaz nemívají žádosti o certifikát separátní položku. Vždy se však najde nějaké rozšíření, do kterého je ho možné vložit.
- Pokud je vydání certifikátu zpoplatňováno, pak je třeba též předat **údaje potřebné pro fakturaci** (fakturační adresa, IČO, DIČ, číslo bankovního účtu pro inkaso apod.).
- **Hesla pro komunikaci s certifikační autoritou –** jedná se zejména o:
  - Jednorázové heslo pro vystavení certifikátu, které je užitečné zejména v případě, že vydavatel certifikátů nechce uživatele obtěžovat častou návštěvou registračních autorit a chce využít strategii „jedna návštěva stačí". Uživatel se totiž zpravidla poprvé bez problémů dostaví na registrační autoritu bez žádosti o certifikát pro informace. Avšak podruhé (s žádostí o certifikát) se mu tam už osobně nechce. Takže již při první návštěvě s ním registrační autorita sepíše veškeré podklady a místo vydání certifikátu mu vydá jednorázové heslo pro vydání certifikátu. Uživatel si pak z tepla domova či kanceláře odešle žádost přes Internet a doplní ji jednorázovým heslem o vydání certifikátu. CA následně zkontroluje, zdali se shodují údaje, které vyplnil při první návštěvě, s údaji v žádosti. A navíc zkontroluje jednorázové heslo. Když je vše v pořádku, pak vydá certifikát, který opět uživateli zašle přes Internet.
  - Jednorázové heslo pro odvolání certifikátu je velice užitečné v okamžiku, kdy je uživateli zcizen soukromý klíč. Např. mu byla odcizena PKI čipová karta, na níž měl fixem napsán PIN. V takovém případě je rychlé využití jednorázového hesla pro zneplatnění certifikátu přímo k nezaplacení. Uživatel se pak může na CA obrátit libovolným komunikačním kanálem (telefonicky, faxem, e-mailem, webem apod.) a požádat o odvolání svého certifikátu. Autentizuje se přitom jednorázovým heslem pro zneplatnění certifikátu.
  - Stálé heslo pro osobní (neelektronickou) komunikaci uživatele s CA. Na základě autentizace tímto heslem může CA poskytovat různou podporu svým uživatelům.
  - Kromě stálého hesla si uživatelé často volí ještě tzv. frázi. Fráze je užitečná v okamžiku, kdy uživatel přijde úplně o vše včetně jednorázových a stálých hesel. Jako fráze se volí např. dívčí jméno tchýně apod. Pokud i to zapomene, má se zpravidla koho zeptat, postupuje-li obezřetně.

Důkaz o vlastnictví soukromého klíče

Prvním dotazem je, proč by měl žadatel o certifikát předkládat důkaz o držení příslušného soukromého klíče. Na první pohled se může zdát nesmyslné, proč by někdo žádal o vystavení certifikátu veřejného klíče, když k němu nemá odpovídající soukromý klíč!

Představte si, že by žadatel o certifikát vygeneroval párová data shodná, jako vygeneroval jiný žadatel, kterému již byl vystaven certifikát veřejného klíče. Pokud by certifikační autorita vydala certifikáty se stejným veřejným klíčem dvěma různým osobám, bylo by to velice frapantní. Oba by měli stejný soukromý klíč a mohli by jeden za druhého např. podepisovat dokumenty.

Případy, kdy si dva uživatelé vygenerují stejná párová data, jsou téměř nemožné. Avšak to platí jen za předpokladu, že uživatelé používají kvalitní generátory náhodných čísel. Vinou nekva-

litního softwaru se nemožné může stát realitou. Certifikační autority zpravidla kontrolují, jestli předkládaný veřejný klíč již necertifikovaly (musí být shodný i algoritmus). Pokud ano a certifikát je platný, pak rozumná certifikační autorita původní certifikát okamžitě odvolá a novou žádost zamítne.

Takže kdyby CA nekontrolovala držení příslušného soukromého klíče, útočník by se dostavil s žádostí o certifikát s veřejným klíčem uživatele, kterému chce uškodit, a CA by příslušný certifikát ke škodě uživatele odvolala.

Nejčastějším typem důkazu o držení příslušného soukromého klíče je digitální podpis ze struktury obsahující veřejný klíč. V anglické terminologii se žádosti o certifikát obsahující důkaz o držení příslušného soukromého klíče na bázi digitálního podpisu označují jako **CSR**, *Certificate Signing Request*.

Důkaz o vlastnictví soukromého klíče je třeba vytvářet na jiném principu pro klíče určené k digitálnímu podepisování, pro klíče určené k šifrování a pro klíče určené pro výměnu klíčů (např. Diffie-Hellmanův algoritmus).

Důkaz založený na digitálním podpisu

V případě, že žádost obsahuje veřejný klíč určený pro ověření digitálního podpisu, vytvoří se jako důkaz digitální podpis z žádosti. Problém je jen u žádostí, které neobsahují veřejný klíč, pak se jako důkaz musí použít digitální podpis z nějakého jiného řetězce.

Verifikaci důkazu provedla RA jinou cestou

Je také možné, že důkaz o vlastnictví soukromého klíče provede registrační autorita jinou cestou. Pak žádost o certifikát nemusí obsahovat žádný důkaz. Praktické ale je, aby skutečnost, že RA ověřila důkaz vlastnictví soukromého klíče, byla v žádosti vyznačena.

Důkaz pro šifrovací klíče

V případě, že žádost obsahuje šifrovací klíč, je nutné provést důkaz na základě šifrování. V tomto případě máme následující možnosti:

- Žadatel může poskytnout celou dvojici veřejný/soukromý klíč. Jelikož se jedná o šifrovací klíče, může využívat i možnost archivace soukromého klíče na CA.
- Žadatel využije dešifrování jako důkaz, že vlastní příslušný soukromý klíč. Jsou zde dvě metody:
  - Přímá metoda: CA vygeneruje náhodný dotaz, který šifruje veřejným klíčem. Žadatel jej dešifruje svým soukromým klíčem a nešifrovaný výsledek vrátí CA. Když CA obdrží řetězec, který vygenerovala, nemohl jej dešifrovat nikdo jiný než držitel soukromého klíče.
  - Nepřímá metoda: CA vydá certifikát, který šifruje veřejným klíčem uvedeným v certifikátu, a výsledek předá žadateli. Jedině žadatel mající k dispozici příslušný soukromý klíč je schopen takto šifrovaný certifikát dešifrovat a publikovat.

Důkaz na základě výměny klíčů

Pro výměnu klíčů (např. Diffie-Hellmanův algoritmus) je možné využít jak přímou, tak i nepřímou metodu (uvedenou v předchozím odstavci). Obě strany (žadatel i CA/RA) si nejprve na základě výměny svých klíčů ustaví společný tajný šifrovací klíč. Tento klíč je pak možné využít

pro symetrickou šifru, kterou je šifrován náhodný dotaz (přímá metoda) nebo vystavený certifikát (nepřímá metoda).

V případě algoritmů pro výměnu klíčů je možná ještě jedna alternativa. Výměnou klíčů si obě strany ustaví společné sdílené tajemství, které následně žadatel o certifikát použije pro výpočet kryptografického kontrolního součtu (MAC). CA/RA pak tento kryptografický kontrolní součet verifikuje.

Všechny metody vyžadovaly, aby obě strany měly vygenerována svá párová data pro výměnu klíčů.

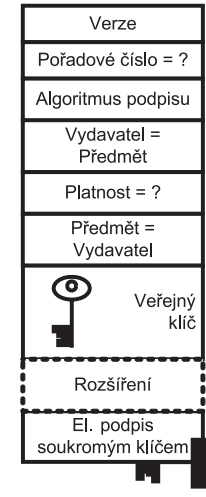
Kořenový certifikát

Kořenový* (*selfsigned*) certifikát je takový certifikát, který si vydává sám žadatel o certifikát. Kořenový certifikát má stejnou datovou strukturu jako certifikát, takže máme kořenové certifikáty X.509 verze 1 či X.509 verze 3.

A protože si kořenové certifikáty vydává sám uživatel, poznají se tyto certifikáty podle toho, že mají položku Předmět identickou s položkou Vydavatel.

Chceme-li vytvořit žádost o certifikát ve tvaru kořenového certifikátu, zjistíme, že to není až tak jednoduché. Certifikát totiž musí mít některá pole, s jejichž vyplněním bychom mohli mít problém. Např. pořadové číslo certifikátu, platnost certifikátu apod.

Jako důkaz o držení soukromého klíče uživatelem se v kořenovém certifikátu použije digitální podpis certifikátu, který byl vytvořený soukromým klíčem příslušejícím k veřejnému klíči uvedenému v certifikátu. Ověření tohoto podpisu se tedy provádí veřejným klíčem uvedeným v samotném kořenovém certifikátu.



Obrázek 4.2: Kořenový certifikát

Kořenový certifikát se v praxi nepoužívá jako žádost o certifikát, s níž by žadatelé přicházeli na veřejnou CA žádat o vystavení certifikátu. Software jej však často používá vnitřně. Pokud žadatel vygeneruje žádost o certifikát, aby ji uplatnil na veřejné CA, po dobu mezi vygenerováním párových dat uživatelem a okamžikem, kdy CA vystaví certifikát, musí být veřejný klíč udržován na počítači žadatele. V tomto mezidobí bývá veřejný klíč často udržován ve formátu kořenového certifikátu. Kořenový certifikát bývá pak přepsán vydaným certifikátem (se stejným veřejným klíčem).

Poznámka: Pozor! Není kořenový certifikát jako kořenový certifikát.

V předchozím odstavci jsme uvedli, že kořenový certifikát si vydává uživatel jako zvláštní typ žádosti o certifikát.

Termín kořenový certifikát v případě certifikátů CA má zcela jiný význam. Kořenové certifikáty jsou též certifikáty kořenových CA – jsou tou nejvyšší autoritou (tzv. důvěryhodnou kotvou). Tyto certifikáty mají rovněž shodné položky Vydavatel a Předmět, ale mají řádně vyplněny i všechny ostatní položky (pořadové číslo, platnost…).

Formát obou typů kořenových certifikátů je týž, ale význam je jiný.

* Překlad slova *selfsigned* jako kořenový v tomto případě trochu kulhá, ale nevymysleli jsme lepší.

PEM

PEM (*Privacy Enhancement for Internet Electronic Mail*) byl z dnešního pohledu prvním systémem PKI na Internetu, který pracoval s certifikáty X.509 (tehdy verze 1). Jeho počátky jsou v RFC-989 z února 1987. Postupně byl přepracováván až do RFC-1421 až RFC-1424.

Protokol PEM jako žádost o certifikát využíval X.509 kořenový certifikát verze 1. Jakými hodnotami vyplnit sporná pole kořenového certifikátu, předepisuje RFC-1424, které je pro tento případ inspirací dodnes.

Jenže certifikát X.509 je binární struktura a protokol PEM byl orientován na komunikaci elektronickou poštou, která je principiálně sedmibitová. Takže nejenom žádost o certifikát, ale všechna další binární data protokol PEM kódoval algoritmem Base64 (viz kap. 13) do sedmibitového tvaru. Výsledek byl uvozen a ukončen vždy řádkem, který měl mezi skupinou pomlček napsáno, o jakou strukturu se jedná:

```
——-BEGIN PRIVACY-ENHANCED MESSAGE——-
Proc-Type: 4,MIC-ONLY
Content-Domain: RFC822
Originator-Certificate: <requestor's self-signed certificate>
MIC-Info: RSA,RSA-MD2,<requestor's signature on text>

<text>
——-END PRIVACY-ENHANCED MESSAGE——-
```

Protokol PEM se v praxi neujal (v té době byl masově používán systém PGP). Zbylo nám po něm ale označení, že nějaká struktura je ve formátu PEM. Míní se tím, že původní binární struktura byla kódována Base64 do sedmibitového tvaru. Takže v diskuzi můžeme slyšet „pošli mi tu PKCS#10 žádost (či ten certifikát, to CRL…) v PEM formátu". A není tím míněno, že by žádost měla být ve tvaru kořenového certifikátu verze 1, ale pouze to, že binární data mají být kódována Base64.

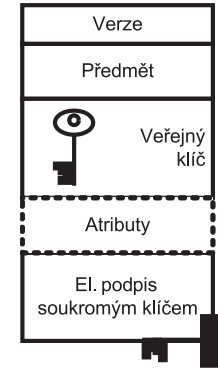
PKCS#10

PKCS#10 (obr. 4.3) je konečně již opravdovou žádostí o certifikát, tak jak je znázorněno na obr. 3.2. Velice zjednodušeně řečeno, PKCS#10 je obdobou kořenového certifikátu s tím rozdílem, že sporná pole, která způsobovala komplikace při vyplňování, byla vypuštěna.

Zbyla nám tak následující pole:

- Verze, do které vyplňujeme verzi struktury PKCS#10. T. č. má tato položka hodnotu 0.
- Předmět
- Veřejný klíč
- Atributy – tato položka obsahuje jednotlivé atributy. Jedním z atributů jsou rozšíření, která si žadatel přeje uvést do certifikátu.

Do položky Atributy můžeme navrhnout rozšíření, která si přejeme mít ve výsledném certifikátu. Pochopitelně závisí na certifikační politice CA, jak se zde uvedenými atributy naloží. Dalším typem atributu je jednorázové heslo pro odvolání certifikátu. Tento atribut se z pochopitelných důvodů ve vydaném certifikátu neobjeví.



Obrázek 4.3: Žádost o certifikát formátu PKCS#10

Žádost PKCS#10 je podepsána soukromým klíčem, který přísluší k veřejnému klíči uvedenému v žádosti. To je obdobně jako v případě kořenových certifikátů důkaz, že žadatel vlastní příslušný soukromý klíč.

PKCS#10 je podniková norma firmy RSA. Vyšla tedy ze světa algoritmu RSA. Algoritmus RSA umožňuje jak šifrování, tak i digitální podpis. To je ale také základním omezením i toho jinak velice populárního formátu žádosti o certifikát. Je totiž obtížné tento formát použít pro žádost o certifikaci veřejných klíčů algoritmů, které neumožňují digitální podpis. Proč se tento formát nehodí pro žádosti o certifikaci veřejných klíčů, které nebudou určeny k ověřování digitálního podpisu? Asi si říkáte, proč takové komplikace s šifrovacími certifikáty. Ať to prostě uživatel podepíše a hotovo, a pak ať si ten certifikát používá jen pro šifrování. Jenže pokud žádáme o certifikaci Diffie–Hellmanova veřejného čísla, tak ze soukromého DH čísla digitální podpis prostě neuděláme, ať se budeme snažit sebevíc. A asymetrických algoritmů nepodporujících digitální podpis je více.

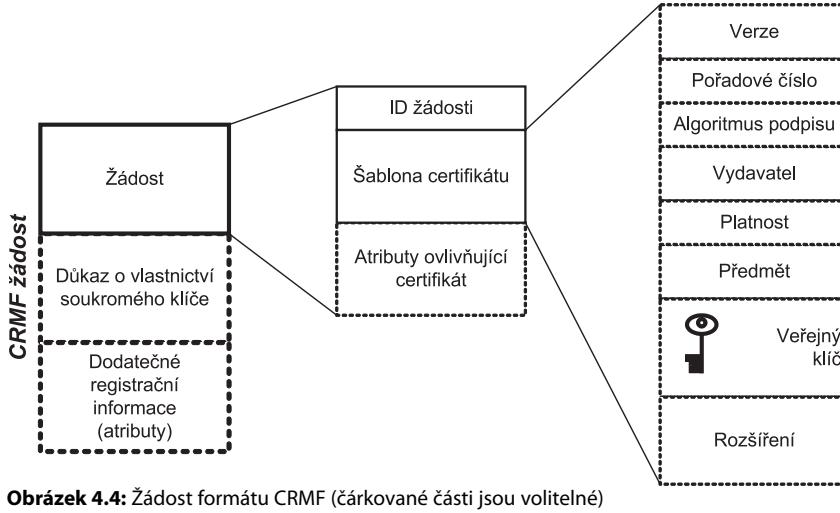
Dále se PKCS#10 žádost nehodí pro případy, kdy jsou párová data generována až CA, tj. kdy uživatel v okamžiku žádosti párová data nemá k dispozici vůbec.

Obtížné je i obnovování certifikátů žádostí PKCS#10. Při obnovování certifikátů musíme předvést důkaz o držení nejenom nového soukromého klíče, ale i starého klíče (čímž prokazujeme svou totožnost – lépe řečeno kontinuitu totožnosti). To lze ale vyřešit vložením žádosti PKCS#10 např. do zprávy CMS, která je podepsána starým klíčem.

Norma PKCS#10 byla převzata nejprve jako RFC-2314 a později jako RFC-2986.

CRMF

CRMF (*Certificate Request Message Format – RFC 4211*) je podstatně bohatší žádost než žádost PKCS#10. Řeší totiž některé již zmiňované problémy, s nimiž se žádost PKCS#10 nedokáže vyrovnat. Žádost tvaru CRMF umožňuje jako součást žádosti zaslat dokonce i informace pro fakturaci.



Obrázek 4.4: Žádost formátu CRMF (čárkované části jsou volitelné)

CRMF se skládá z následujících částí:

- Žádost, která obsahuje vlastní žádost o certifikát. Žádost pak obsahuje tzv. šablonu certifikátu (nezaměňovat s šablonami certifikátu podle terminologie MSCA). Do šablony

certifikátu si žadatel navrhne položky svého certifikátu. Je pak na politice certifikační autority, jak s těmito položkami naloží.

- Důkaz o vlastnictví soukromého klíče, do kterého žadatel vyplní důkaz, že vlastní soukromý klíč příslušející k veřejnému klíči, jenž má být certifikován.
- Dodatečné registrační informace (atributy) obsahující dodatečné informace, např. kontakt na žadatele, údaje o účtování za vydání certifikátu apod. Asi vás okamžitě napadne otázka, proč tyto informace nelze vložit již do první části struktury – do položky Žádost. Odpověď je jednoduchá. Informace, které se vkládají do žádosti, např. na registrační autoritě během ověřování totožnosti žadatele, již nelze vložit do položky, aby se nepoškodil její digitální podpis nebo jiný důkaz o vlastnictví soukromého klíče. Takové informace se proto vloží do položky Dodatečné registrační informace (atributy).

SPK

Žádost o certifikát typu SPK je atypická žádost zavedená firmou Netscape. Cílem bylo umožnit jednoduché vytváření žádostí o certifikát pomocí webových stránek.

Firma Netscape zavedla zvláštní HTML tág KEYGEN, který může být součástí popisu jednoho pole webového formuláře. Odeslání formuláře obsahujícího tág KEYGEN klientem z jeho webového prohlížeče způsobí:

- Vygenerování dvojice klíčů veřejný/soukromý klíč.
- Vytvoření digitálního podpisu veřejného klíče vygenerovaným soukromým klíčem.
- Vložení Base64 kódovaného veřejného klíče včetně jeho digitálního podpisu do obsahu pole HTML stránky, jež obsahuje tág KEYGEN.

Tato žádost o certifikát se označuje jako žádost formátu SPK. Jedná se o SPK nestandardní typ žádosti o certifikát, protože neobsahuje digitální podpis celé žádosti, ale pouze podpis veřejného klíče. Normy PKI tento formát žádosti v podstatě neznají, ale certifikační autority jej někdy podporují.

Žádosti generované webovou stránkou

Jenže certifikační autority zpravidla požadují žádosti o certifikát formátu PKCS#10 nebo CRMF. Takovou žádost je možné vyrobit i pomocí webové stránky obsahující softwarovou komponentu, která žádost takového formátu vytvoří. Takovými komponentami jsou ActiveX nebo Java Applety.

Problém je ale s tím, že komponenta musí umět vygenerovat párová data, vytvořenou žádost digitálně podepsat a párová data uložit do příslušných úložišť na počítači žadatele. To ale běžná komponenta stažená z webu z bezpečnostních důvodů udělat nemůže. Může to však provést komponenta digitálně podepsaná důvěryhodnou CA (byť je stahována z Internetu).

A komu ani komponenta digitálně podepsaná důvěryhodnou certifikační autoritou nepřipadá dostatečně bezpečná, ten si nainstaluje program, který příslušné žádosti vytváří.

CMC

Protokol CMC kompletně řeší dialog pro vystavování certifikátu. Pomocí tohoto protokolu je tedy rovněž možné vystavovat certifikáty.

Tento protokol je v některých případech využíván i Microsoft CA, oficiálně nazývanou Active Directory Certificate Services (ADCS). A to zejména v případě, kdy je požadováno, aby žádost o certifikát obsahovala více podpisů: jeden pomocí klíče žadatele a druhý pomocí klíče RA (např. v případě tzv. *Smartcard Enrolment Station*).

Protokolu CMC se více věnuje kap. 17.

Kapitola 5

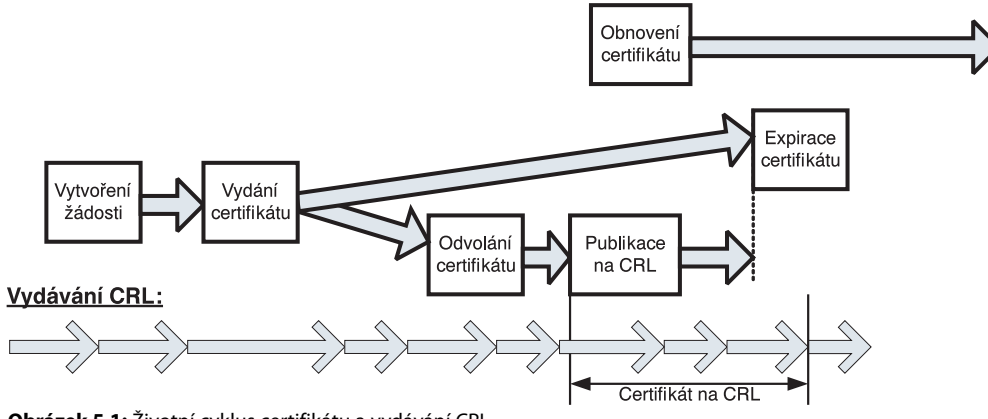
Odvolávání certifikátu

Certifikát může ztratit svou platnost tak, že vyprší jeho původně deklarovaná doba platnosti, tj. uplyne čas `notAfter` uvedený v certifikátu. Před tím, než vyprší původně deklarovaná doba platnosti certifikátu, může být certifikát odvolán nebo jeho platnost může být pozastavena. Pozastavení certifikátů je však jen zřídka používaným institutem.

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

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

Odvolané certifikáty jsou publikovány na seznamu odvolaných certifikátů (CRL) po celou dobu jejich původně deklarované platnosti. To, jestli certifikační autorita vůbec umožňuje certifikáty odvolávat či s jakou periodou vydává CRL, deklaruje certifikační autorita v dokumentu „Certifikační politika".



Obrázek 5.1: Životní cyklus certifikátu a vydávání CRL

Rozlišujeme několik druhů CRL:

- Přímá CRL vydává certifikační autorita, která vydala i odvolávané certifikáty.
- Nepřímá CRL vydává jiná než certifikační autorita, která vydala odvolávané certifikáty. Taková autorita se nazývá autoritou pro vydávání CRL.

- Úplné CRL obsahuje identifikace všech odvolaných certifikátů, jejichž původní doba platnosti dosud nevypršela.
- Rozdílové (přírůstkové) CRL obsahuje identifikace jen těch odvolaných certifikátů, které byly odvolány od posledního úplného CRL.
- Částečné (omezené) CRL obsahuje podmnožinu úplného CRL. Částečné CRL obsahuje rovněž kritérium, podle kterého byly vybrány identifikace certifikátů do částečného CRL.

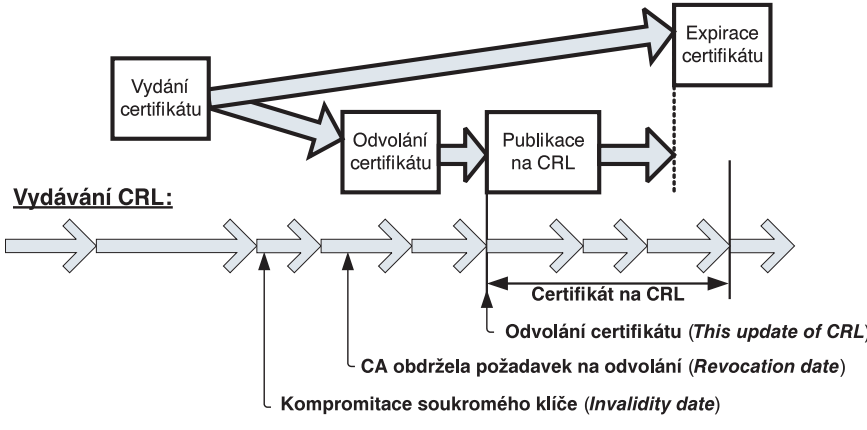
V drtivé většině případů se budeme setkávat s přímými, úplnými CRL. Jestliže nebude uvedeno jinak, pak pokud budeme uvádět termín CRL, budeme mít na mysli přímé, úplné CRL.

Žádost o odvolání certifikátu nemusí být prováděna elektronickou formou, ale CA může vyžadovat např. osobní účast uživatele. Pokud se provádí elektronickou cestou, může být žádost digitálně podepsána soukromým klíčem odvolávaného certifikátu. (V případě, že by takovou žádost provedl hacker neoprávněně, tj. uměl ji digitálně podepsat, čili znal soukromý klíč, pak je pro uživatele jen dobře, že ji podal.) Případně může být použito jednorázové heslo pro odvolání certifikátu.

Na obr. 5.2 je znázorněn proces odvolání certifikátu. Nejprve byl kompromitován privátní klíč. Tento okamžik se označuje jako okamžik Zcizení (kompromitace) soukromého klíče (*Invalidity date*). Poté si toho uživatel všiml a nahlásil to na CA. Tento okamžik se označuje jako Obdržení požadavku na odvolání (*Revocation date*). Jestliže CA žádost o odvolání shledala za oprávněnou, pak identifikaci certifikátu zařadila na nejbližší vydávané CRL (Vydání CRL, *This update of CRL*).

I když si postižený uživatel kompromitace svého soukromého klíče nepovšiml třeba i řadu dní, od CA očekává okamžitou publikaci svého certifikátu na CRL. Certifikační autority většinou volí jednu ze dvou strategií vydávání CRL:

- CRL vydávají v podstatě v pravidelných intervalech, a tak uživatel s kompromitovaným klíčem musí přetrpět interval do vydání dalšího CRL.
- CRL vydávají vždy po odvolání nějakého certifikátu. To však vyžaduje, aby si ostatní uživatelé skutečně před použitím každého certifikátu stáhli a zpracovali CRL, což je zase zdržuje při práci, protože stažení CRL vždy nějakou dobu trvá.



Obrázek 5.2: Jízdní řád odvolávání certifikátu

Realita však není tak romantická. Nejčastějším důvodem pro odvolání certifikátu kupodivu není kompromitace soukromého klíče, ale ukončení pracovního poměru zaměstnance, který měl vystaven zaměstnanecký certifikát. Zaměstnavatelé jsou schopni takový certifikát odvolat v dostatečném předstihu.

Vydávání CRL je ve své podstatě dávkovou záležitostí. Vznikají tak různé protokoly pro On Line zjišťování stavu certifikátů. Základní otázkou však zůstává, může-li certifikační autorita jiným kanálem (např. On Line kanálem) poskytnout informaci, že certifikát byl odvolán, když ještě není uveřejněn na CRL. Na tuto otázku neexistuje jednoznačná odpověď, ale většina lidí na to má vyhraněný názor.

Žádost o odvolání certifikátu

Při odvolávání certifikátu nejde o to, postupovat podle nějakých technických norem, ale jde především o rychlost, proto certifikační politiky jednotlivých CA bývají v této oblasti poměrně pružné. Neexistuje nějaká standardizovaná struktura s žádostí o odvolání certifikátu. Je několik možných způsobů odvolání certifikátu, jejichž použití většinou závisí na okolnostech, za kterých se certifikát odvolává:

- Jestliže byl kompromitován soukromý klíč, ale máte jej k dispozici, pošlete žádost o odvolání certifikátu na CA např. elektronickou poštou. Zprávu elektronické pošty podepíšete soukromým klíčem příslušejícím odvolávanému certifikátu. Pokud se díváte, že něco podepisujete kompromitovaným klíčem, uvědomte si, že kdyby o odvolání požádal útočník, pak mu jen poděkujete. Omezení této metody:
  - Soukromý klíč musí být k dispozici.
  - Tuto metodu nelze použít, jestliže odvolávaný certifikát není určen k digitálnímu podpisu. Ovšem máte-li vydán jiný podpisový certifikát, který je platný, pak může CA požádat o odvolání digitálně podepsanou zprávou za využití jiného platného podpisového certifikátu. Tuto metodu využívají zejména firmy pro odvolávání certifikátů svých zaměstnanců – viz následující paragraf.
- Pokud se odvolává certifikát zaměstnance rozvazujícího pracovní poměr, pravděpodobně jej odvolává pověřený pracovník zaměstnavatele. K takovému odvolání stačí digitálně podepsaná zpráva. Omezení této metody:
  - Pověřený pracovník zaměstnavatele musí mít vystaven odpovídající certifikát pro digitální podpis.
  - Zaměstnavatel by měl mít uzavřenu smlouvu s CA o tom, kteří jeho zaměstnanci mohou žádat o odvolání certifikátů ostatních zaměstnanců.
- Některé CA s vydáním certifikátu též vystaví jednorázové heslo pro odvolání certifikátu. Jestliže toto heslo znáte, lze odvolat certifikát telefonicky, faxem pomocí webového formuláře nebo nepodepsanou elektronickou poštou s uvedením zmíněného jednorázového hesla. Omezení této metody:
  - S vydáním certifikátu musí být vydáno i příslušné jednorázové heslo pro odvolání certifikátu (či obecně heslo pro ne-elektronickou komunikaci uživatele s CA).
  - Uživatel toto heslo nesmí zapomenout.

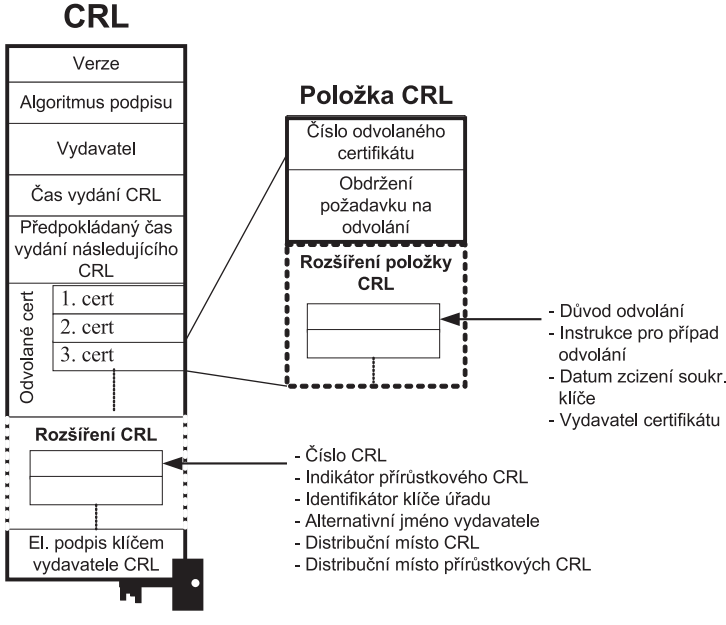
- Neznáte-li ani jednorázové heslo, nezbývá, než se s doklady totožnosti vydat na registrační autoritu. Pokud přijdete úplně o vše, máte problém. V takovém případě to ale asi nebude ten největší problém, který máte.

Nesmíme zapomenout na případ, že CA odvolává certifikát ze své iniciativy. Takovým případem může být situace, kdy se objeví žádost o vydání certifikátu s již certifikovaným veřejným klíčem. Dalším případem je skutečnost, kdy CA zjistí, že údaje uvedené v certifikátu (např. v předmětu certifikátu) již nejsou platné. Takovým případem je i skutečnost, když o odvolání certifikátu požádá odcházející zaměstnanec. Pověřený pracovník zaměstnavatele vlastně neodvolává přímo certifikát, ale jen informuje CA o tom, že údaje v uvedeném certifikátu již nadále nebudou platné.

CRL

CRL je ve své podstatě něco jako úřední deska CA, na níž CA zveřejňuje odvolané certifikáty. Je-li jednou certifikát odvolán (zveřejněn na CRL), pak by měl být zveřejňován i na všech následujících CRL, dokud by nevypršela jeho původní platnost.

CRL zpravidla vydávají certifikační autority, které to však nemusí dělat samy, touto službou mohou pověřit jinou autoritu – autoritu pro vydávání CRL. Ta pak vydává tzv. nepřímá CRL. U nepřímých CRL se zpravidla postupuje tak, že CA vydá certifikát serveru na vydávání svých CRL. Tento certifikát má v rozšíření Použití klíče nastaven bit `CRL Sign`.



Obrázek 5.3: Struktura CRL

Odvolané certifikáty jsou v CRL specifikovány podle svých sériových (tj. pořadových) čísel, takže pro běžného uživatele je CRL docela odtažitou záležitostí. CRL je určeno pro počítačové zpracování. V případě nepřímých CRL musí být odvolaný certifikát specifikován kromě sériového čísla ještě názvem CA, tj. pomocí položky Vydavatel.

Tvar CRL specifikovala norma X.509. Na Internetu používáme CRL, která sice vycházejí ze specifikace CRL podle normy X.509 verze 2, ale opět je tato struktura pro Internet popsána v RFC-5280. Struktura CRL je obdobná struktuře certifikátu (obr. 5.3).

- První položkou CRL je Verze (*Version*); RFC-5280 předepisuje, že v CRL musí být použita a musí mít hodnotu 1 (tj. CRL dle X.509 verze 2).
- Položka Algoritmus podpisu (*Signature Algorithm*) obsahuje identifikátor algoritmů použitých pro digitální podpis CRL (obdobně jako v certifikátu).
- Položka Vydavatel (*Issuer*) identifikuje, kdo tento CRL vydal (obdobně jako v certifikátu).
- Položka čas vydání CRL (*This Update*) specifikuje čas vydání tohoto CRL.
- Položka Předpokládaný čas vydání následujícího CRL (*Next Update*) určuje nejpozdější datum a čas vydání následujícího CRL. Následující CRL může být vydáno dříve nebo nejpozději v čase uvedeném v této položce. Následující CRL nesmí být vydáno později, než je datum a čas, které jsou uvedené v této položce předchozího CRL.
- Položka Odvolané certifikáty (*Revoked Certificates*) obsahuje seznam odvolávaných certifikátů. Pro každý odvolávaný certifikát obsahuje sekvenci údajů:
  - Pořadové číslo (*Serial Number*) odvolávaného certifikátu, které společně s položkou Vydavatel jednoznačně identifikuje odvolávaný certifikát (nejedná-li se o nepřímé CRL, pak se vydavatel bere z rozšíření položky odvolaných certifikátů).
    - Datum a čas, kdy držitel podal žádost o odvolání certifikátu, obsahuje položka Obdržení požadavku na odvolání certifikátu (*Revocation Date*).
    - Volitelná položka rozšíření odvolávaného certifikátu obsahuje jednotlivá rozšíření týkající se odvolání tohoto certifikátu.
- Nepovinná položka Rozšíření CRL (*CRL Extension*) obsahuje rozšíření CRL.
- Všimněte si, že v CRL jsou dva typy rozšíření:
    - Rozšíření položky CRL (*CRL Entry Extensions*), které obsahuje informace o odvolávání konkrétního certifikátu.
    - Rozšíření CRL (*CRL Extensions*) – rozšíření CRL jako celku, které je obdobou rozšíření certifikátu.

Příklad výpisu CRL ve Windows je uveden na obr. 16.1.

Rozšíření CRL

Rozšíření CRL jsou obdobou rozšíření certifikátu. Tato rozšíření mohou být opět označena jako závažná; při jejich zpracování si zpracovávající software musí být vědom, co takové závažné rozšíření znamená.

RFC-5280 explicitně popisuje šest rozšíření:

Rozšíření	RFC-5280 předepisuje:		Význam
Identifikátor klíče úřadu (*Authority Key Identifier*)	**Musí být použito**	**Není závažné**	**Viz stejnojmenné rozšíření certifikátu**
Alternativní jméno úřadu (*Issuer Alternative Name*)	**Není vyžadováno**	**Nemělo by být závažné**	**Viz stejnojmenné rozšíření certifikátu**
Číslo seznamu CRL (*CRL Number*)	**Musí být použito**	**Není závažné**	**Certifikační autorita pomocí tohoto rozšíření čísluje vydávaná CRL. Číslování se provádí pomocí monotónně rostoucí sekvencí čísel. Pokud zjistíme, že certifikát je na CRL, okamžitě dostaneme strach, jestli nebyl na předchozích CRL, a my jsme jej přesto používali. Musíme tedy zjistit, které bylo předchozí CRL. To provedeme právě podle rozšíření Číslo seznamu CRL. Předchozí CRL musí mít předchozí pořadí v sekvenci čísel přidělovaných CRL.**
Indikátor rozdílového seznamu CRL (*Delta CRL Indicator*)	**Musí být v rozdílovém CRL**	**Závažné**	**Toto rozšíření indikuje, že se jedná o rozdílové CRL. Rozdílové CRL obsahuje jen změny od předchozího úplného CRL. Rozdílové CRL obsahuje také číslo úplného CRL, od kterého jsou přírůstky brány.**
Vystavování distribučního místa (*Issuing Distribution Point*)	**Nemusí být podporováno**	**Závažné**	**Na první pohled by se mohlo zdát, že toto rozšíření je identické s rozšířením certifikátu *CRL Distribution Points*. To je však omyl. Rozšíření Vystavování distribučního místa je závažným rozšířením a indikuje, že se nejedná o kompletní CRL vydavatele, ale o částečné CRL. Případně obsahuje odkazy na další dílčí částečná CRL.**
Nejčerstvější seznam CRL (*Freshest CRL*)	**Nesmí být uvedeno v rozdílovém CRL**	**Není závažné**	**Obsahuje odkaz, kde obdržet rozdílová CRL k úplnému CRL.**
Přístup k informacím úřadu (*Authority Information Access*)		**Není závažné**	**Obdobně jako stejnojmenné rozšíření certifikátu odkazuje na certifikát CA, kterým se má ověřovat toto CRL.**

Rozšíření položky CRL

Kromě samotného odvolání certifikátu mohou být zajímavé i další informace související s jeho odvoláním. Tyto informace mohou obsahovat rozšíření položky CRL; tato rozšíření by neměla být označována jako kritická.

Důvod odvolání certifikátu (*Reason Code*)

Toto rozšíření identifikuje důvod odvolání certifikátu. CRL by měl obsahovat vždy konkrétní důvod odvolání certifikátu. Je však možné v odůvodněných případech toto rozšíření neuvést, což je stejné, jako by byl uveden kód pro „důvod nespecifikován" (*unspecified reason*).

Čas kompromitace soukromého klíče (*Invalidity Date*)

Toto rozšíření obsahuje datum a čas, kdy byla zjištěna ztráta či prozrazení soukromého klíče, jež je důvodem odvolání certifikátu (viz obr. 5.2).

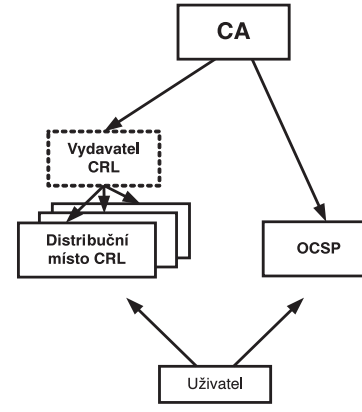
Vydavatel certifikátu (*Certificate Issuer*)

Toto rozšíření položky nepřímého CRL obsahuje jedinečné jméno vydavatele odvolaného certifikátu. Pokud toto rozšíření neobsahuje první položka CRL, jedná se o přímé CRL. Pokud toto rozšíření neobsahuje další položka CRL, bere se za vydavatele týž vydavatel jako v předchozí položce.

On Line zjišťování statusu certifikátu

Jestliže uživatel používá certifikát pouze v jedné aplikaci (např. při styku s bankou), nejspíše okamžitě po kompromitaci soukromého klíče kontaktuje provozovatele této aplikace a certifikát si nechá v této aplikaci zablokovat. V tomto případě snad ani CRL nepotřebuje.

Jenže pokud uživatel používá certifikát k více účelům, vyžaduje často službu, která by o odvolání jeho certifikátu co nejrychleji informovala ostatní uživatele. Jako řešení se nabízí On Line služba organizovaná CA. Obecně nehovoříme jen o odvolání certifikátu, ale o změně statusu certifikátu.



Obrázek 5.4: On Line služby jsou paralelními službami k CRL

On Line služby jsou vesměs konstruovány na základě architektury klient/server. Uživatel, který chce certifikát použít, se On Line dotáže serveru poskytujícího tyto služby na status certifikátu. Když server odpoví, že certifikát je platný, uživatel jej použije. Pokud server odpoví, že certifikát je odvolán, uživatel jeho použití zamítne. A jestliže mu server sdělí, že status tohoto certifikátu je mu neznámý, pak je na uživateli (resp. jeho softwaru), aby se rozhodl.

Jiným aspektem je otázka, jestli vůbec může CA tvrdit o nějakém certifikátu, že je odvolán, když není uveden na oficiálním CRL. Nebo On Line informace má snad jen informativní charakter? To už si ale musí vyřešit tvůrce certifikační politiky konkrétní CA. Na první pohled to vypadá jednoduše, jenže stačí si uvědomit, jak informace o tom, že certifikát byl odvolán, ale nebyl na CRL, bude vypadat po několika dnech odstupu a v případě, že nebudeme mít On Line odpověď archivovánu. Pak se objeví dohady typu: A opravdu to tak bylo?

Protokolem určeným pro On Line zjišťování statusu certifikátu je např. protokol OCSP (*Online Certificate Status Protocol*). On Line status certifikátu poskytuje OCSP server. CA deleguje pravomoc poskytovat On Line status svých certifikátů tím, že OCSP serveru vystaví speciální certifikát, který má v rozšíření Rozšířené použití klíče (*Extended Key Usage*) explicitně vyznačeno, že se jedná o certifikát OCSP serveru.

Pomocí protokolu OCSP lze řešit dva problémy:

- CRL vystavuje CA zpravidla s nějakou minimální periodou – tj. dávkově. V případě, že držitel certifikátu zjistí, že jeho soukromý klíč byl ztracen nebo odcizen, nebude

mu mechanismus CRL příliš vyhovovat, protože do vydání dalšího CRL může být jeho soukromý klíč zneužit. Protokolem OCSP (alespoň teoreticky) může certifikační autorita poskytovat informace o odvolání certifikátu rychleji než pomocí CRL.

- CRL může být značně rozsáhlá datová struktura, jejíž stažení a zpracování může značně zdržovat. OCSP dotaz a následná odpověď tak leckdy mohou být rychlejší a méně zdržovat uživatele. OCSP pak nenahrazuje CRL, ale zrychluje zpracování certifikátu.

Platnost certifikátu k uvedenému datu

Máme-li např. digitálně podepsaný dokument a chceme si ověřit platnost jeho digitálních podpisů, bude nás zajímat platnost příslušných certifikátů nikoliv teď, ale v okamžiku podepsání dokumentu. Potřebujeme tedy On Line dotaz na status certifikátu doplnit o datum, ke kterému chceme zjistit platnost certifikátu.

Protokol OCSP verze 1 však není určen pro řešení problému platnosti certifikátu ke konkrétnímu datu v minulosti.

Vzdálené ověřování platnosti certifikátu

CRL mohou být značně rozsáhlá a jejich stahování a následné zpracování může náš počítač značně zaměstnat. Navíc pokud chceme ověřit platnost certifikátu, musíme ověřit platnost všech certifikátů v řetězci certifikátů až k důvěryhodné kotvě. Tj. musíme nejprve získat všechny certifikáty až k důvěryhodné kotvě a pak zjistit status všech certifikátů.

V úřadech už každý uživatel nemívá vlastní tiskárnu, ale pro skupinu uživatelů je jeden printserver, který je rychlejší a efektivnější. Co kdybychom měli také server, který by za nás ověřoval certifikáty? Ocenili by to jistě tvůrci miniaturních počítačů, bylo by to výhodné i v rámci intranetu – poskytli bychom jeden server, který by příslušné informace zjišťoval a poskytoval centrálně ostatním PC.

RFC-3379 zavádí dvě strategie (dva protokoly), jak takovou situaci řešit:

- DPV (*Delegated Path Validation*) protokol umožní plně delegovat ověřování certifikátu na DVP server. Server provádí ověřování sám a klientovi vrací výsledek ověření.
- DPD (*Delegated Path Discovery*) protokol. DPD server slouží k tomu, aby poskytoval veškeré informace nutné pro ověření certifikátu. Tj. klient požádá DPD server o informace, DPD server je ve své odpovědi poskytne a klient si sám lokálně provede ověření certifikátu.

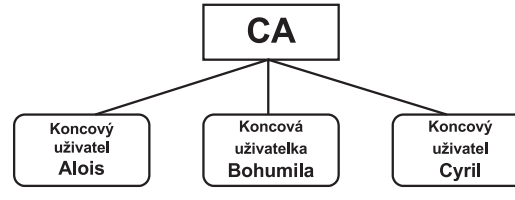
Toto RFC specifikuje sadu požadavků, které by nový protokol(y) měl naplňovat. RFC–5055 pak specifikuje protokol Server–Based Certificate Validation Protocol (SCVP), který dokonce umožňuje i verifikaci certifikátu k zadanému času v minulosti, tj. implementuje protokoly DPV a DPD. Protokol SCVP je blíže popsán v kapitole 21.

Kapitola 6

Certifikační cesta a důvěryhodné kotvy

Dosud jsme předpokládali, že máme samostatnou (*stand alone*) certifikační autoritu, která vystavuje certifikáty svým koncovým uživatelům (obr. 6.1). Tato představa předpokládá, že certifikační autorita má kořenový (*self signed*) certifikát. Kořenový certifikát certifikační autority má několik zvláštností:

- Je podepsán soukromým klíčem příslušejícím k veřejnému klíči uvedenému v témže certifikátu CA (viz obr. 6.2).
- Pokud by se ověřoval digitální podpis kořenového certifikátu, využil by se veřejný klíč z téhož certifikátu.
- Ověření digitálního podpisu kořenového certifikátu tedy nepřinese nic než kontrolu integrity datové struktury certifikátu. Digitální podpis kořenového certifikátu má tedy z pohledu jeho ověření nanejvýš hodnotu jako otisk.
- Podvrhnout kořenový certifikát je tudíž velice snadné. Útočníkovi stačí vygenerovat libovolnou dvojici veřejný/soukromý klíč a k takto vygenerovanému veřejnému klíči vytvoří kořenový certifikát s libovolným obsahem.
- Kořenové certifikáty musí být proto distribuovány důvěryhodnou cestou. Takovou cestou je např. osobní předání kořenového certifikátu s vydaným certifikátem koncového uživatele na registrační autoritě (RA) či distribuce kořenového certifikátu v rámci distribuce softwaru (např. distribuce operačního systému).
- Kořenovým certifikátům tedy důvěřujeme, protože jsme je získali z důvěryhodného zdroje. Jestliže obdržíme kořenový certifikát jinou cestou, explicitně důvěřovat mu nemůžeme a jeho důvěryhodnost si musíme ověřit. Nejběžnějším postupem ověření důvěryhodnosti takového certifikátu je, že se jinou cestou (např. telefonicky) spojíme s důvěryhodnou osobou (např. operátorkou CA), které se dotážeme na důvěryhodnost certifikátu. Jak to uděláme? Požádáme ji, aby nám recitovala otisk z tohoto certifikátu, a recitovaný tisk pak porovnáme s námi spočteným otiskem ověřovaného certifikátu. Nutno dodat, že je třeba, aby obě strany použily stejný algoritmus pro výpočet otisku. A také je třeba poznamenat, že mnohdy není nutné vyžadovat celý otisk, ale stačí jen některé cifry z otisku (fragment otisku).
- Certifikátům CA (nebo veřejným klíčům), jejichž platnost ověřujeme



Obrázek 6.1: Samostatná CA vydávající certifikáty koncovým uživatelům

jinou cestou, a tudíž jim explicitně důvěřujeme, říkáme **důvěryhodné kotvy**. Kořenový certifikát, který jsme ověřili jinou cestou, je příkladem důvěryhodné kotvy.

- Při ověřování certifikátu důvěryhodným kotvám věříme, a proto je samotné již neověřujeme – pouze využijeme údaje z nich (zejména informace týkající se veřejného klíče). Důvěryhodná kotva tak není součástí tzv. certifikační cesty (viz kap. 7).
- Jelikož důvěryhodná kotva není součástí certifikační cesty, tak v případě, že má tvar certifikátu (nejenom kořenového), se z tohoto certifikátu berou v úvahu jen některé údaje. Např. většina rozšíření certifikátu se v případě důvěryhodné kotvy ignoruje!
- Jiným způsobem distribuce důvěryhodné kotvy je její distribuce jako důsledek, že naše PC je součástí podnikové sítě např. na bázi Active Directory, pak certifikáty důvěryhodných CA mohou být distribuovány skrze Active Directory. Např. skrze zásady skupin (*Group Policy*) či `NTAuthSore`.
- Součástí novějších operačních systémů Microsoft je komponenta „*Update Root certificates*". Pokud si ji nainstalujete, Microsoft vám pravidelně přes Internet aktualizuje seznam důvěryhodných kotev registrovaných společností WebTrust (viz též paragraf Microsoft). Každá mince má však dvě strany. Na intranetu bude taková funkčnost patrně nevítaná.

Podvržení kořenového certifikátu

Když Bohumila pošle Aloisovi digitálním podpisem stvrzený rozchod (k podpisu přidá svůj certifikát i certifikát kořenové CA, jež jej vydala), nesmí Alois mechanicky ověřit certifikát Bohumily včetně kořenového certifikátu CA zaslaného Bohumilou, ale musí zjistit, jestli se kořenový certifikát zaslaný Bohumilou shoduje s kořenovm certifikátem, který má uložen mezi svými důvěryhodnými kotvami!

Kdyby totiž takové ověření neprovedl, bylo by také možné, že zprávu vytvořil žárlivý Cyril. Jak by to udělal? Jednoduše:

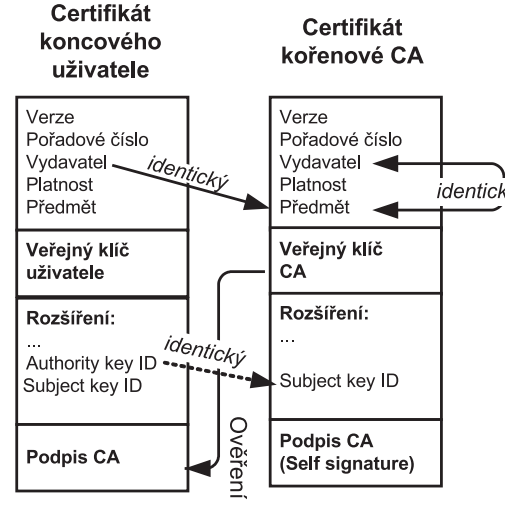
1. Vygeneroval by dvojici veřejný/soukromý klíč.
2. Tuto dvojici by Cyril podvrhl za veřejný/soukromý klíč CA (vytvořil by podvržený certifikát kořenové CA):
   a. Do podvrženého certifikátu by z certifikátu Bohumiliny oblíbené CA zkopíroval veškeré údaje CA kromě veřejného klíče.
   b. Jako veřejný klíč by Cyril použil právě vygenerovaný veřejný klíč.
   c. Tvorbu podvrženého certifikátu CA by završil digitálním podpisem certifikátu podvržené CA, k němuž by Cyril použil právě vygenerovaný soukromý klíč.
3. Nyní by Cyrilovi již nic nebránilo podvrhnout i certifikát Bohumily:
   d. Vygeneroval by další dvojici veřejný/soukromý klíč, která mu umožní vytvořit podvržený certifikát Bohumily.
   e. Do podvrženého certifikátu Bohumily by zkopíroval veškeré identifikační údaje z pravého certifikátu Bohumily kromě veřejného klíče.
   f. Do certifikátu by vložil vygenerovaný (podvržený) veřejný klíč.
   g. Certifikát Bohumily podepíše podvrženou CA.
   h. Nyní již může vytvořit podvržený dokument, který bude verifikovatelný podvrženým certifikátem Bohumily.

Ověření certifikátu Bohumily

Jak jsme se již zmínili, kořenový certifikát CA se ověřuje jinou cestou – je totiž důvěryhodnou kotvou. Nadále tedy budeme předpokládat, že kořenový certifikát CA je důvěryhodnou kotvou, o které nepochybujeme.

Ověření certifikátu Bohumily má tři polohy:

- Samotné ověření certifikátů vůči důvěryhodné kotvě. Je třeba mít na paměti, že naše CA může mít více certifikátů. Tj. musí být shoda nejenom v tom, že položka Vydavatel (*Issuer*) z certifikátu Bohumily je shodná s položkou Předmět (*Subject*) v certifikátu CA, ale zejména hodnota rozšíření Identifikátor klíče úřadu (*Authority Key Identifier*) v certifikátu Bohumily musí být shodná s rozšířením Identifikátor klíče předmětu (*Subject Key Identifier*) certifikátu CA*. Toto ověření je podmnožinou ověření certifikátu v řetězci certifikátů, které je věnována následující kapitola, proto se bližšími detaily zde již zabývat nebudeme.



Obrázek 6.2: Verifiace certifikátu koncového uživatele vydaného CA mající kořenový certifikát

- Ověření, zdali certifikát nebyl odvolán. To je možné pomocí CRL nebo pomocí On Line metod.
- Certifikát je možné využít v konkrétní aplikaci:
  1. Je pro konkrétní aplikaci použitelný příslušný asymetrický algoritmus?
  2. Umožňují využití certifikátu jeho rozšíření: Použití klíče (*Key Usage*), Rozšířené použití klíče (*Extended Key Usage*) atd.?
  3. Umožňuje to certifikační politika vyznačená v certifikátu?

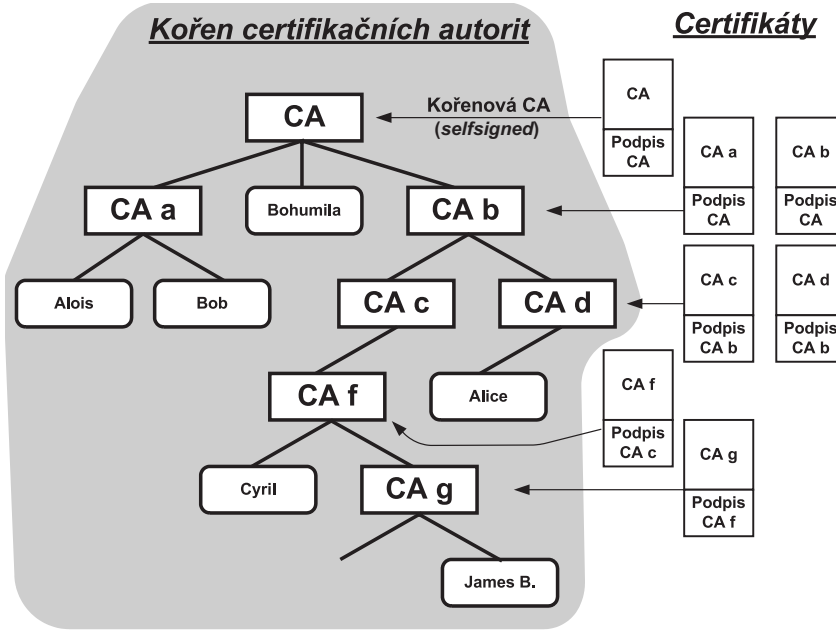
Ověřování certifikátu se budeme podrobněji věnovat v kap. 8.

Strom certifikačních autorit

Realita však může být složitější. Naše certifikační autorita CA může být jen kořenem ve stromu certifikačních autorit.

Certifikační autorita CA může vydat kromě certifikátů koncových uživatelů i certifikáty podřízeným certifikačním autoritám CAa a CAb. A obdobně např. certifikační autorita CAb může vytvořit své podřízené certifikační autority CAc a CAd tak, že jim vydá certifikáty. Výsledkem je stromová struktura certifikačních autorit (obr. 6.3).

* Jinou variantou je do rozšíření Identifikátor klíče úřadu uložit jedinečné jméno CA a číslo certifikátu CA, kterým se ověřovaný certifikát ověřuje. V takovém případě certifikát CA nemusí mít ani rozšíření Identifikátor klíče předmětu.



Obrázek 6.3: Strom certifikačních autorit

Jak vypadá certifikát podřízené certifikační autority? Nejdůležitější změna oproti certifikátu kořenové CA spočívá v tom, že tento certifikát má různé hodnoty v položkách Vydavatel (*Issuer*) a Předmět (*Subject*). X.509 označuje takové certifikáty jako křížové certifikáty (*cross-certificate*). Termín křížový certifikát se však nesmí zaměňovat s termínem křížová certifikace certifikačních autorit, o níž bude zmínka dále (křížová certifikace je oboustranná, tj. vyžaduje dva křížové certifikáty – každý v jednom směru).

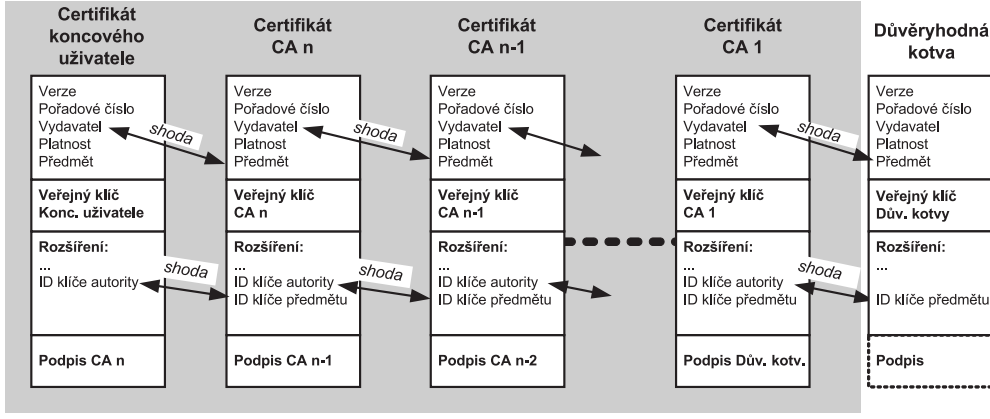
Na první pohled je tedy křížový certifikát obdobný certifikátu koncového uživatele. Jenže při podrobnějším zkoumání zjistíme, že křížový certifikát se značně liší zejména v rozšířeních certifikátu. Proč? Důvod je prostý. Tím, že někomu vydáme křížový certifikát, mu dáváme do rukou moc vydávat další certifikáty. A pokud nějakým způsobem ručíme za vydávané certifikáty, přeneseně budeme ručit i za certifikáty vydané osobou, které jsme vydali křížový certifikát. Jelikož se trochu bojíme, aby nezvlčil a vydával certifikáty zodpovědně, budeme se jej snažit omezit v neuváženém vydávání certifikátů.

Pomineme-li rozšíření certifikátu Použití klíče (*Key Usage*), můžeme držitele křížového certifikátu omezovat ve vydávání certifikátů zejména promyšleným využitím rozšíření certifikátů, které mají v názvu slovo „omezení“ (*constrain*). Pomocí rozšíření Základní omezení (*Basic Constrains*) jej můžeme omezovat ve vydávání dalších křížových certifikátů, pomocí rozšíření Omezení jmen (*Name Constrains*) jej můžeme omezit na vydávání certifikátů subjektům majícím např. jména pouze z konkrétních DNS domén atd.

Řetězec certifikátů

V případě, že máme strom certifikačních autorit, bude ověření certifikátu koncového uživatele složitější než v případě kořenové certifikační autority. Je třeba provést nejen ověření certi-

fikátu koncového uživatele, ale i ověření certifikátu certifikační autority, která tento certifikát vydala. A pokud i tato certifikační autorita má křížový certifikát, musíme verifikovat i ten. Neověřujeme pak jeden certifikát, ale řetězec certifikátů (obr. 6.4).



Obrázek 6.4: Certifikační cesta. Při jejím sestavování postupujeme „zleva doprava", ale při jejím ověřování naopak „zprava doleva"!

Řetězec certifikátů je zakončen důvěryhodnou kotvou, která bývá zpravidla ve tvaru kořenového certifikátu. Teoreticky můžeme i jako důvěryhodnou kotvu využít také samotný veřejný klíč (včetně jeho případných parametrů a jména jeho vydavatele).

Máme-li k dispozici množinu certifikátů, prvním problémem bude vytvořit řetězec certifikátů až k důvěryhodné kotvě. Tvorbu řetězce certifikátů zpravidla začínáme od ověřovaného certifikátu, např. od certifikátu koncového uživatele. K němu hledáme certifikát certifikační autoritu, která jej vydala, tak, že najdeme všechny certifikáty, jejichž položka Předmět (*Subject*) je identická s položkou Vydavatel (*Issuer*). Takových certifikátů může být i více. Mezi nimi vybereme ten, jehož hodnota rozšíření Identifikátor klíče předmětu (*Subject Key Identifier*) je shodná s hodnotou rozšíření Identifikátor klíče úřadu (*Authority Key Identifier*) ověřovaného certifikátu. Pak zjistíme, jestli certifikát této certifikační autority náhodou nemáme mezi důvěryhodnými kotvami. Pokud ano, další prvek řetězce certifikátů již nehledáme, pokud nikoliv, postupujeme obdobně dále, až dosáhneme důvěryhodné kotvy.

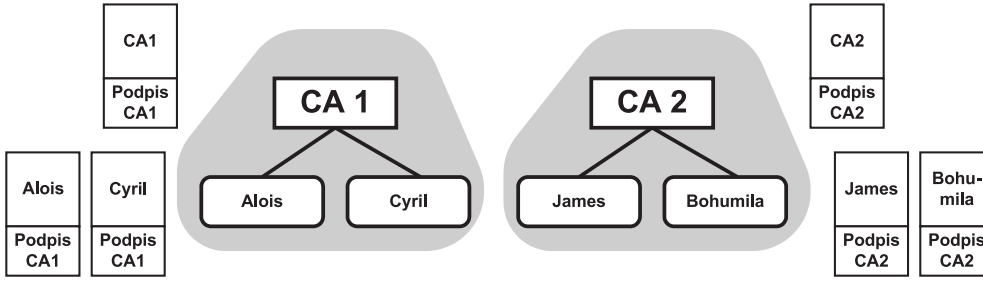
Máme-li postavený řetězec certifikátů, přikročíme k ověřování (verifikaci) certifikátů. Při ověřování certifikátu postupujeme zpět od důvěryhodné kotvy přes všechny křížové certifikáty až k ověřovanému certifikátu. Pro každý certifikát v řetězci musíme ověřit, jestli předchozí (nadřízená) certifikační autorita byla oprávněna vydat v řetězci uvedený certifikát, jestli některý certifikát z řetězce certifikátů není uveden na CRL atd. Pokud selže ověření jednoho certifikátu v řetězci certifikátů, selže také ověřování ověřovaného certifikátu a ověřovaný certifikát musí být odmítnut. Verifikaci certifikátu v řetězci certifikátů se budeme podrobně věnovat v kap. 7.

Ověříme-li platnost křížových (mezilehlých) certifikátů v certifikační cestě, můžeme si informaci o jejich ověření uložit a po jistou dobu těmto certifikátům důvěřovat obdobně jako důvěryhodné kotvě. Nebo naopak je můžeme považovat za odvolané. Tento mechanismus může pak značně zrychlit ověřování dalších certifikátů. Je však rozdíl mezi důvěryhodnou kotvou tvořenou kořenovým certifikátem a těmito důvěryhodnými mezilehlými, křížovými certifikáty.

Rozdíl spočívá v množství rozšíření certifikátu, která mají. V případě důvěryhodné kotvy se totiž jiná rozšíření než Alternativní jméno předmětu nevyužijí, kdežto rozšíření z mezilehlých certifikátů se zpravidla ověřují.

Vzájemná důvěra mezi certifikačními autoritami

Může se však stát, že dva uživatelé mají certifikáty vydány od různých certifikačních autorit, které navíc nejsou součástí téhož stromu certifikačních autorit (obr. 6.5).



Obrázek 6.5: Dva stromy certifikačních autorit

Jak pak může Alois důvěřovat např. digitálně podepsaným zprávám od Bohumily, když sám důvěřuje jen certifikační autoritě CA1 a Bohumila důvěřuje jen certifikační autoritě CA2?

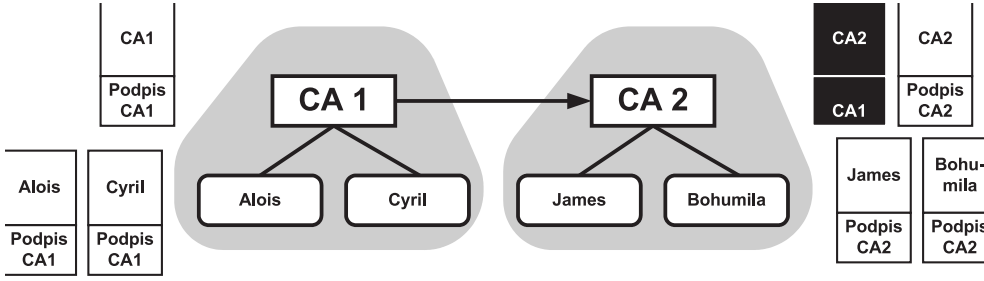
Řešení tohoto problému máme několik:

1. Alois si řekne: Když Bohumila důvěřuje CA2, tak já jí také budu důvěřovat, a doplní si certifikát CA2 mezi své důvěryhodné kotvy.
2. Vzájemná křížová certifikace CA1 a CA2.
3. Kořenová certifikační autorita tvořící most (*Bridge*) mezi stromy certifikačních autorit.
4. CTL (*Certificate Trusted List*), tzv. „Seznam důvěryhodných certifikátů".

Křížová certifikace

Doposud jsme předpokládali, že certifikační autority tvoří stromovou strukturu. Avšak vše může být ještě komplikovanější. Dvě certifikační autority, které nejsou zařazeny do téže stromové struktury, si mohou vzájemně podepsat své certifikáty – provést křížovou certifikaci. (Pro úplnost je třeba dodat, že křížově se teoreticky mohou certifikovat i CA patřící do společného stromu – to má však význam, když je strom příliš košatý a CA leží na velmi vzdálených větvích stromu; pak lze křížovou certifikací jejich vzdálenost zmenšit. Je to ale spíše jen teoretická úvaha, kterou můžeme ještě doplnit o skutečnost, že v takovém případě by v certifikační cestě mohly vznikat dokonce smyčky…)

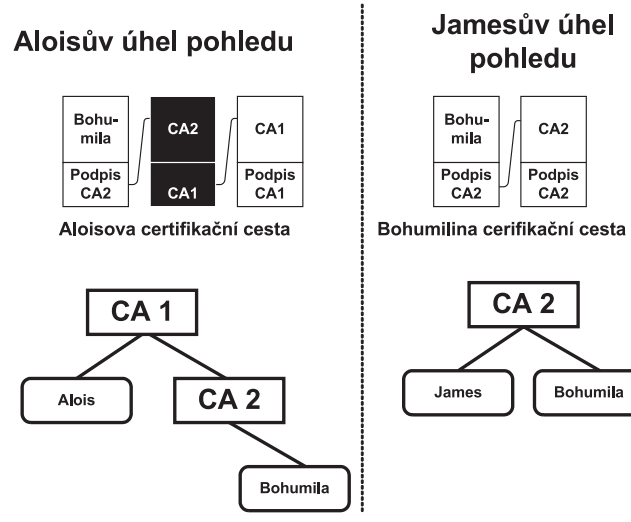
Avšak vraťme se na počátek k vysvětlení podstaty křížové certifikace (obr. 6.6). Aby Alois mohl důvěřovat zprávám podepsaným Bohumilou a nemusel si certifikát CA2 uložit mezi své důvěryhodné kotvy, CA2 si kromě svého kořenového certifikátu nechala vydat certifikát od CA1.



Obrázek 6.6: CA1 vydala CA2 další certifikát certifikační autority

Takže CA2 nyní má dva certifikáty certifikační autority: (1) kořenový, (2) vydaný CA1. Technicky to CA2 provedla jednoduše. Když si vystavovala svůj kořenový certifikát, vytvořila k tomu příslušnou žádost o certifikát. Nyní tutéž starou žádost uplatní u CA1.

Na obr. 6.7 je vidět, jak se nově vydaný certifikát CA2 prakticky uplatní. Aby James, důvěřující podobně jako Bohumila certifikační autoritě CA2, ověřil certifikát Bohumily, vytvoří řetězec certifikátů bez nově CA1 vydaného certifikátu CA2. Jamesovi takový řetězec certifikátů stačí, protože pro něj je kořenový certifikát CA2 důvěryhodnou kotvou.

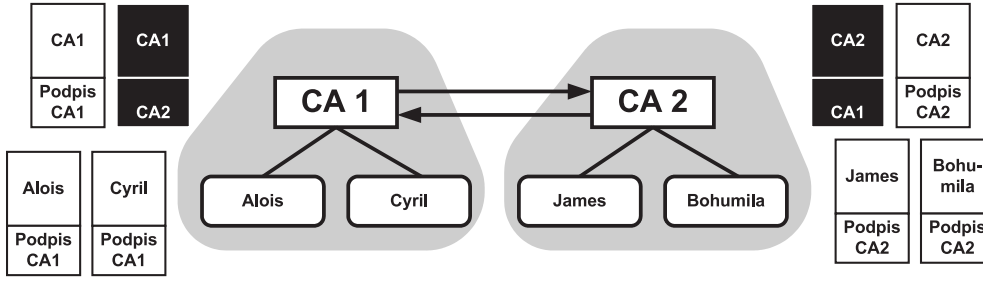


Obrázek 6.7: Alois a Bohumila vytvoří různé řetězce certifikátů – vždy ke svým důvěryhodným kotvám

Jenže pro Aloise není kořenový certifikát CA2 důvěryhodnou kotvou, pro něj je důvěryhodnou kotvou pouze kořenový certifikát CA1. Ale pomocí certifikátu CA2 vydaného CA1 je schopen vytvořit řetězec certifikátů až k jeho důvěryhodné kotvě. Takže i Alois nyní může důvěřovat také certifikátu Bohumily.

Jenže v případě, že Alois pošle Bohumile podepsanou odpověď, není Bohumila schopna nalézt řetězec až k jeho důvěryhodné kotvě. Je tedy nutné, aby si nejenom CA2 nechala vydat certifikát od CA1, ale naopak, aby i CA1 si nechala vydat certifikát od CA2 (obr. 6.8). Říkáme, že CA1 se s CA2 vzájemně křížově certifikovaly. Výsledkem je, že máme čtyři certifikáty:

1. CA1 podepsaný CA1, tj. kořenový certifikát CA1.
2. CA2 podepsaný CA2, tj. kořenový certifikát CA2.
3. CA1 podepsaný CA2, tj. křížový certifikát.
4. CA2 podepsaný CA1, tj. křížový certifikát.



Obrázek 6.8: Vzájemná křížová certifikace certifikačních autorit CA1 a CA2

Nyní mohou Alois s Bohumilou oboustranně důvěryhodně komunikovat. Pro ověření certifikátů bude každý z nich vytvářet jiné řetězce certifikátů (každý ke „své" důvěryhodné kotvě), ale komunikace bude možná.

V případě digitálně podepsaných zpráv odesilatelé zpravidla neposílají samotnou digitálně podepsanou zprávu, ale též certifikáty, o nichž se domnívají, že je příjemce bude potřebovat pro ověření digitálního podpisu (tj. bude je potřebovat k sestavení certifikační cesty). Odesilatel ale neví, jaké má adresát důvěryhodné kotvy, tak mu ke zprávě zabalí množinu certifikátů. Je pak na adresátovi, aby byl schopen z prvků této množiny vytvořit řetězec certifikátů až k některé z jeho důvěryhodných kotev. Adresát si však může potřebné křížové certifikáty obstarat i jinou cestou (např. využitím rozšíření Přístup k informacím úřadu (*Authority Information Access*) či obecně na LDAP serverech certifikačních autorit).

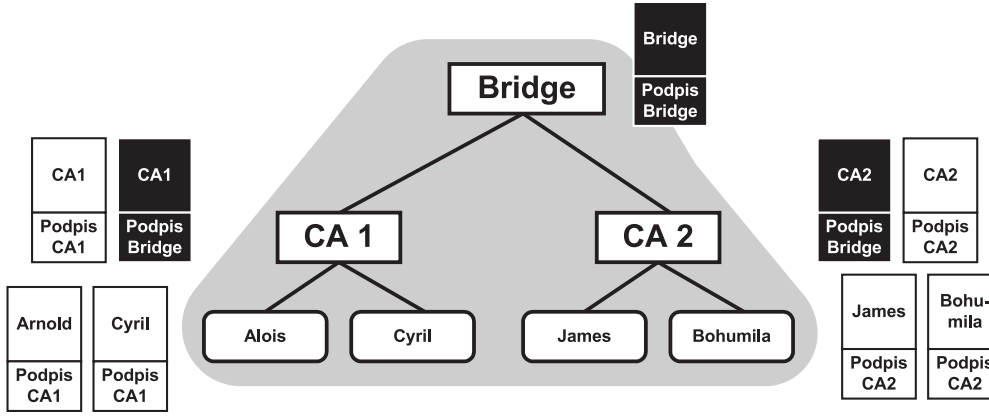
V případě, že se křížově certifikují dvě certifikační autority, křížová certifikace na obrázku vypadá jednoduše. Jenže pokud chcete vzájemně křížově certifikovat 10 certifikačních autorit, bude množství křížových certifikátů značné a problém je netriviální. A pak se jednoho dne jako naschvál stane, že potřebujete obnovit certifikát některé z těchto certifikačních autorit – a máte o zábavu postaráno.

Most certifikačních autorit (*Bridge*)

Vzájemná křížová certifikace může proběhnout po vzájemné dohodě dvou certifikačních autorit. Problém však nastane v případě, když by se mělo vzájemně křížově certifikovat větší množství certifikačních autorit. Pokud chceme propojit více certifikačních autorit, křížová certifikace není tím pravým ořechovým. Možným řešením je most (*Bridge*).

Certifikační autority, které si chtějí vzájemně důvěřovat, společně vybudují kořenovou certifikační autoritu – mostní kořenovou certifikační autoritu, *Bridge* (obr. 6.9).

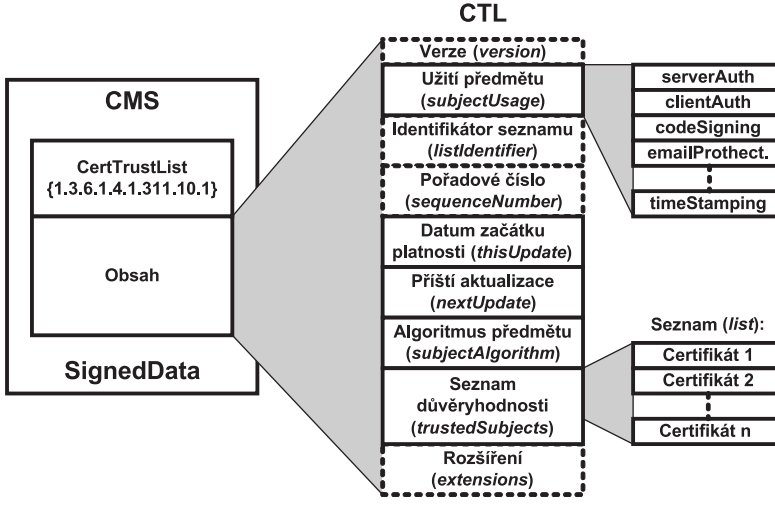
V evropských zemích vznikají na základě zapracování směrnice Evropského parlamentu a Rady 1999/93/ES do národních legislativ jednotlivých členských zemí akreditované certifikační autority, které vydávají tzv. kvalifikované certifikáty, na jejichž základě lze vytvářet digitální podpis, nahrazující rukou psaný podpis. Akreditované certifikační autority jsou často soukromé podnikatelské subjekty. V některých evropských zemích (Polsko, Slovensko atd.) stát provozuje kořenovou CA, která na základě akreditace vydá akreditované CA křížový certifikát. Tato kořenová CA pak tvoří most. To usnadňuje vzájemnou komunikaci i mezi uživateli s certifikáty vydanými různými akreditovanými CA v rámci jedné země.



Obrázek 6.9: Most (*Bridge*)

CTL (*Certificate Trusted List*)

Ne vždy je možné jako most použít certifikační autoritu. Jednodušším řešením je vytvořit a distribuovat jen seznam certifikátů důvěryhodných kořenových certifikačních autorit. Pokud je navíc takový seznam digitálně podepsán důvěryhodnou osobou, nic nám nebrání v tom, abychom si z něj bez obav důvěryhodné kotvy nainstalovali. Kromě toho ušetříme mostní certifikační autoritu.



Obrázek 6.10: Struktura CTL

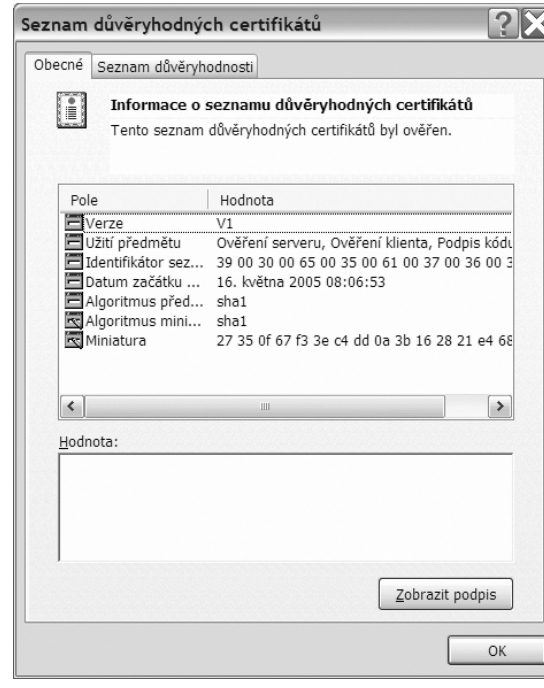
Microsoft zavedl seznam důvěryhodných kotev, označovaný jako CTL (*Certificate Trusted List*). Jedná se o seznam kořenových certifikátů, který je podepsán administrátorem. Sítě postavené na bázi Windows pak lze nakonfigurovat tak, že akceptují jako důvěryhodné kotvy certifikáty z CTL. Tento mechanismus se však nemusí využívat jen v rámci intranetu.

Dosud neexistuje žádný oficiální standard pro specifikaci CTL. Následující informace jsme získali rozborem CTL vydaného CA Windows. Digitální podpis CTL je zajištěn tím, že CTL

je vloženo do CMS zprávy `SignedData` (viz kap. 23). To umožní, aby CTL kontrasignovaly i další osoby (tj. aby se pod CTL přidávaly další podpisy). Struktura CTL je znázorněna na obr. 6.10.

Nejdůležitějšími položkami CTL jsou:

- **Užití předmětu** (*Subject Usage*) – obsahuje identifikátory objektů účelů, k jakým jsou důvěryhodné kotvy v tomto CTL určeny (autentizace serverů, autentizace klientů, zabezpečení elektronické pošty apod.).
- **Identifikátor seznamu** (*List Identifier*) – obsahuje identifikaci CTL.
- **Datum začátku platnosti** a **Příští aktualizace** – mají obdobný význam jako stejnojmenné položky CRL. Tentokrát však nesouvisí s platností CRL, ale s platností CTL.



Obrázek 6.11: Výpis CTL v prostředí Windows XP

- **Seznam důvěryhodnosti** (*Trust List* nebo *Trusted Subject*) – buď obsahuje identifikace důvěryhodných kotev nebo obsahuje samotné důvěryhodné kotvy. K identifikaci důvěryhodných kotev se používá otisk z důvěryhodné kotvy, který může být doplněn jedinečným jménem vydavatele důvěryhodné kotvy.

Pokud si ve Windows XP otevřeme CTL (obr. 6.11), zjistíme, že CTL se do češtiny vcelku logicky překládá jako „Seznam důvěryhodných certifikátů".

Distribuce veřejných důvěryhodných kotev

CTL je praktický nástroj pro distribuci důvěryhodných kotev v rámci intranetu. Jak se ale distribuují důvěryhodné kotvy veřejných certifikačních autorit? Jak to již bývá, existuje řada na sobě více či méně nezávislých aktivit. Cílem každé z těchto aktivit je buď vytvořit CA typu most nebo alespoň distribuovat digitálně podepsaný seznam důvěryhodných kotev ve tvaru např. CTL.

Mezi nejzajímavější aktivity patří:

- Federal Bridge Certification Authority (viz http://www.cio.gov/fbca/)
- Evropská můstková certifikační autorita (*European Bridge Certification Authority*) viz http://www.bridge-ca.org/
- http://www.openvalidation.org
- Bridge/Gateway Certification Authority – t. č. pouze projekt mostu evropských akreditovaných CA, též známý pod zkratkou IDA (viz http://europa.eu.int/idabc/).

WebTrust

Asi jste si všimli, že po instalaci operačních systémů Microsoft máte v úložišti pro důvěryhodné kořenové CA zpravidla předem instalovánu celou řadu důvěryhodných kotev. Navíc pokud si nainstalujete již zmíněnou komponentu „*Update Root certificates*“, seznam důvěryhodných kotev se vám automaticky přes Internet aktualizuje. Odkud se tyto kořenové certifikáty vzaly?

Odpověď je jednoduchá: Microsoft spravuje „svůj“ seznam důvěryhodných kotev. Na tento seznam se může certifikát vaší CA dostat poté, co projde auditem nezávislé třetí strany – např. WebTrust (www.webtrust.org). Přičemž WebTrust vystaví pouze dobrozdání, vlastní audit certifikační autority je požadován od některé z renomovaných konzultačních firem. Samotný WebTrust nebyl založen Microsoftem (jak by se mohlo zdát), ale dvěma organizacemi: *Canadian Institute of Chartered Accountants* (*CICA.ca)* a *American Institute of Chartered Public Accountants* (*AICPA)*.

Microsoft svůj seznam ve formě CTL distribuuje jako součást instalace a zejména upgradu svých systémů. V novějších verzích Windows je pak upgrade seznamu důvěryhodných kotev oddělen od upgradu systému. Aktivujeme si jej jako samostatnou komponentu systému „*Update Root Certificates*“.

Pro nás je zajímavé, že v programu Microsoft Root Update je zařazena i naše I. Certifikační Autorita (I.CA).

Kapitola 7

Ověřování platnosti certifikátu a poznámka k ověřování digitálního podpisu

V předchozí kapitole jsme sestavovali certifikační cesty od certifikátu, jejž chceme ověřovat, až k důvěryhodné kotvě. Všimli jsme si, že pro konkrétní certifikát můžeme teoreticky sestavit i více certifikačních cest. Dokonce se v certifikačních cestách mohou vyskytovat i cykly*. Je pak na konkrétní implementaci, jak se k této problematice postaví.

Nyní je naším cílem již sestavenou certifikační cestu ověřit. Tj. budeme se zabývat ověřováním (verifikací) konkrétní sestavené certifikační cesty.

Mlčky předpokládáme, že chceme ověřit, je-li certifikát platný právě teď – v tomto okamžiku. Jinou úlohou je ověřování platnosti certifikátu k nějakému okamžiku v minulosti. To je podstatně složitější problém, který nastane v případě ověřování pravosti archivovaných digitálně podepsaných dokumentů. Tento problém dodnes není zcela vyřešen a my se jím budeme zabývat v kapitolách 25, 26 a 27. Nyní se vraťme do horké současnosti – ověřujme platnost certifikátu k aktuálnímu času.

Ověřování cesty začíná od důvěryhodné kotvy!

Při sestavování certifikační cesty jsme postupovali od ověřovaného certifikátu. V případě ověřování pak postupujeme v opačném směru – od důvěryhodné kotvy až k ověřovanému certifikátu. Jelikož důvěryhodné kotvě věříme, a tudíž ji nemusíme ověřovat, postupně ověřujeme platnost prvního certifikátu pod důvěryhodnou kotvou, pak druhého atd., až dojdeme k certifikátu, jehož platnost máme ověřit. Šťouralové by tedy podotkli, že nadpis tohoto odstavce (Ověřování cesty začíná od důvěryhodné kotvy) není zcela pravdivý, protože důvěryhodná kotva přece není součástí certifikační cesty!

Představa, že pro ověření certifikátu nejprve sestavíme certifikační cestu, a teprve až ji máme sestavenu, začneme s jejím ověřováním, je představa školská, se kterou se též setkáváme

* Pokud je při sestavování certifikační cesty vyžadován certifikát, který v cestě již je (tj. jedná-li se o náznak cyklu), pak se zpravidla v sestavování takové cesty již nepokračuje a hledá se jiná certifikační cesta.

ve standardech. Jelikož sestavení a ověření certifikační cesty často trvá vteřiny nebo i desítky vteřin (což obtěžuje uživatele), implementátoři se snaží tento čas urychlit. Již při sestavování certifikační cesty startují ověřování všeho, co je možné ověřit (např. je-li certifikát publikován na CRL). Standard pak výslovně říká, že implementován může být i jiný algoritmus (než že se nejprve sestaví certifikační cesta, která se až následně ověřuje), ale takový jiný algoritmus musí vést k témuž výsledku.

Zde je pak možné nalézt i odpověď na otázku: Jak je možné, že ověřování certifikátů je ve Windows 2003 rychlejší než ve Windows 2000 – Microsoft vytvořil efektivnější ověřovací algoritmus (v jeho terminologii: rychlejší ověřovací stroj).

Ověřujeme certifikační cestu

Certifikát ověřujeme proto, abychom zjistili, zda veřejný klíč uvedený v certifikátu je platný a opravdu náleží držiteli uvedenému v předmětu certifikátu. A dále, že tento klíč může být využit k danému účelu. Proto jsme vytvořili certifikační cestu až důvěryhodné kotvě a nyní ji budeme ověřovat. Certifikát vydaný důvěryhodnou kotvou označujeme jako první certifikát v certifikační cestě. Certifikát vydaný prvním certifikátem označujeme jako druhý certifikát v certifikační cestě atd. Důvěryhodná kotva se často označuje jako certifikát číslo 0.

Důvěryhodná kotva pravděpodobně bude ve tvaru kořenového certifikátu. Z tohoto certifikátu se pro ověřování vezme:

- Jméno vydavatele.
- Veřejný klíč včetně algoritmu veřejného klíče a případných parametrů veřejného klíče.

Jelikož ostatní informace se v důvěryhodné kotvě při ověřování certifikátu ignorují, tak důvěryhodné kotvy, pokud vůbec jsou ve tvaru certifikátu, pak zpravidla obsahují minimum rozšíření certifikátu. Pokud vydavatel důvěryhodné kotvy vydal více certifikátů se stejným předmětem, bude jediným rozšířením s reálným efektem rozšíření Identifikátor klíče předmětu, které má efekt pro sestavení certifikační cesty (nikoliv pro ověření certifikátu).

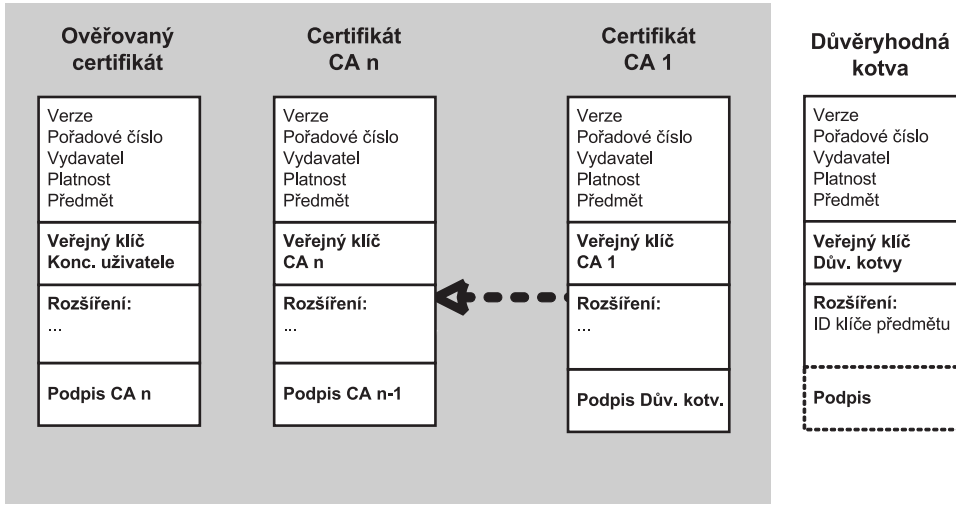
Do procesu ověřování tak vstupuje:

- Sestavená certifikační cesta.
- Důvěryhodná kotva.
- Aktuální čas.
- Inicializační informace týkající se certifikačních politik v případě, že si při ověřování certifikátu hodláme hrát na certifikační politiky.

Využití mnohých rozšíření certifikátu je volitelné. Takže pokud aplikace tato rozšíření nepodporuje, nemusí je využívat a jejich ověření může přeskočit. Nicméně pokud je nepodporované rozšíření označeno jako závažné, musí být certifikát odmítnut.

Při ověřování certifikátu půjdeme krok po kroku po certifikační cestě od certifikátu číslo 1 až k certifikátu, jejž chceme ověřit. Představme si, že ověřujeme certifikát číslo ***x*** v certifikační cestě, pak (blíže viz RFC-5280):

- Ověříme, že vydavatel certifikátu je předmětem certifikátu ***x-1***.
- Ověříme digitální podpis certifikátu ***x*** pomocí certifikátu ***x-1***.
- Ověříme, že aktuální čas padne do období platnosti certifikátu.



Obrázek 7.1: Ověřování certifikační cesty probíhá od důvěryhodné kotvy k ověřovanému certifikátu

- Ověříme, že certifikát nebyl odvolán (viz následující odstavec).
- Ověříme, že jména uvedená v předmětu certifikátu a v rozšíření Alternativní jména předmětu odpovídají omezením vyplývajícím z rozšíření Omezení jmen v předchozích certifikátech certifikační cesty.
- Ověřují se rozšíření spojená s certifikačními politikami (viz popis jednotlivých rozšíření certifikátu v kap. 15).

Pokud neselhalo ověřování žádného certifikátu certifikační cesty až k ověřovanému certifikátu, je certifikát platný.

Byl certifikát odvolán?

Informace o tom, zda byl certifikát odvolán, můžeme získat z CRL nebo jinou cestou (např. pomocí OCSP protokolu). Kde (na jakém URI) tuto informaci získat, je uvedeno v následujících rozšířeních ověřovaného certifikátu:

- V rozšíření Distribuční místa seznamu odvolaných certifikátů (*CRL Distribution Points*) jsou uvedeny URI, na kterých je publikováno CRL.
- V rozšíření Nejčerstvější CRL (*Freshest CRL*) jsou uvedeny URI, na kterých je publikováno přírůstkové CRL (*Delta CRL*).
- V rozšíření Přístup k informacím úřadu (*Authtority Info Access*) může být publikováno i URI, na kterém naslouchá OCSP server.

V případě ověřování, není-li certifikát publikován na CRL, musíme mít k dispozici:

- Aktuální CRL.
- Ověřovaný certifikát, ze kterého vezmeme: jméno vydavatele (*issuer*) a číslo certifikátu jako identifikace certifikátu pro jeho vyhledávání v CRL.

- Informaci o tom, budeme-li využívat přírůstková CRL (*Delta CRL*). Tato informace je signalizována pomocí rozšíření CRL nazývaného Indikátor rozdílového seznamu CRL.

Nyní můžeme začít s ověřováním, nebyl-li certifikát uveden na CRL:

1. Získáme neprošlé CRL (i přírůstkové CRL). Proto postupně procházíme lokální vyrovnávací paměť (*cache*), kontaktujeme všechna URL uvedená v rozšíření Distribuční místa seznamu odvolaných certifikátů v certifikátu, až získáme CRL, jehož položka Příští aktualizace (*Next Update*) obsahuje pozdější čas, než je aktuální čas. Pozor: Výsledkem je, že nemusíme získat aktuální CRL, ale neprošlé CRL!
2. Ověřme, zda CRL bylo vydáno vydavatelem odpovědným za vydávání CRL, tj. v případě přímého CRL musí být vydavatel ověřovaného certifikátu a vydavatel CRL týž.
3. V případě rozdílového (přírůstkového) CRL:
   - Opět ověřujeme, přísluší-li rozdílové CRL k úplnému CRL, které máme k dispozicii a jež využijeme ke zpracování.
   - Ověřujeme, zdali obsah rozšíření Identifikátor klíče úřadu v rozdílovém CRL se shoduje s obsahem téhož rozšíření v úplném CRL.
4. V případě omezeného CRL pak ověříme, jestli se toto omezené CRL týká ověřovaného certifikátu. Např. pokud ověřujeme certifikát certifikační autority, musí omezené CRL obsahovat odvolané certifikáty CA.
5. Vytvoříme certifikační cestu pro CRL (i případné přírůstkové CRL) až k důvěryhodné kotvě.
6. Ověřujeme certifikáty v certifikační cestě CRL (i případného přírůstkového CRL) (viz předchozí odstavec).
7. Ověříme digitální podpis CRL (i přírůstkového CRL).
8. Prohledáme přírůstkové CRL, není-li na něm uveden ověřovaný certifikát.
9. V případě, že ověřovaný certifikát nebyl uveden na přírůstkovém CRL (nebo se přírůstkové CRL vůbec nepoužívá), prohledáme úplné CRL.
10. Nebyl-li certifikát nalezen ani na úplném CRL, považujeme jej za neodvolaný (pomocí mechanismu CRL).

Microsoft

V případě operačních systémů firmy Microsoft se setkáváme s některými zajímavostmi, jež jsou předmětem následujících dvou paragrafů.

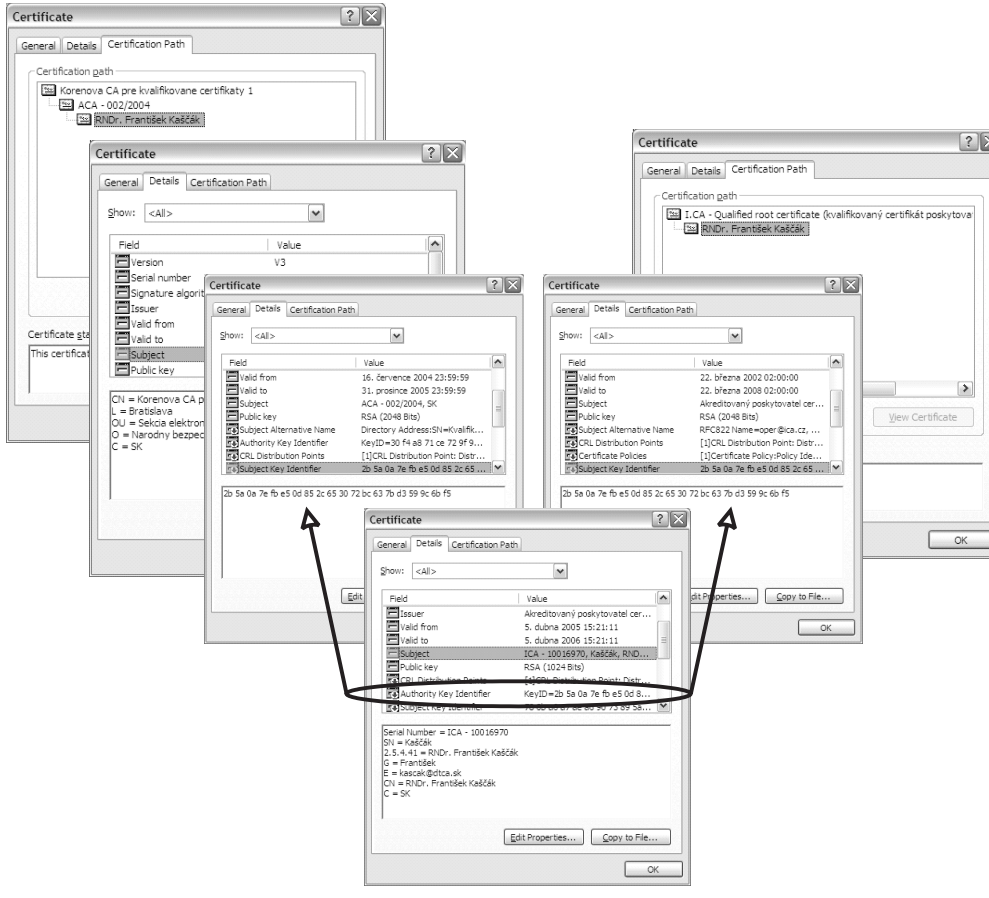
Sestavování certifikační cesty

Microsoft kombinuje mechanismus vytvoření certifikační cesty s ověřením této cesty. Velice zajímavým je mechanismus vyhledávání CA, která ověřovaný certifikát vydala (vyhledáním certifikátu tzv. vydávající CA). Pro vyhledání tohoto certifikátu jsou důležité dvě otázky:

- Kde hledat. Pro tyto účely přece slouží rozšíření Přístup k informacím úřadu (*Authority Info Access*) obsahující URI, na kterém je tento certifikát vystaven. To není nic mimořádného. V PKI je to obecně využívaný mechanismus.
- Ověřit shodu, že nalezený certifikát je opravdu certifikát certifikační autority, která ověřovaný certifikát vydala. A zde je právě ta zajímavost.

Zajímavost spočívá v tom, že se z ověřovaného certifikátu vezme rozšíření Identifikátor klíče úřadu a podle jeho obsahu se nejen vyhledává, ale i ověřuje shoda, zda se opravdu jedná o certifikát CA, která ověřovaný certifikát vydala. Přičemž rozšíření Identifikátor klíče úřadu nám skýtá tři možnosti:

1. Rozšíření Identifikátor klíče úřadu obsahuje jak předmět certifikátu CA, tak i pořadové číslo certifikátu CA, která ověřovaný certifikát vydala. Pak se ověřuje tzv. **přesná shoda** (*Exact match*). Přesná shoda spočívá v tom, že se zkontroluje shoda předmětu a čísla certifikátu CA uvedených v rozšíření ověřovaného certifikátu s předmětem a číslem z nalezeného certifikátu CA v certifikační cestě.
2. Rozšíření Identifikátor klíče úřadu ověřovaného certifikátu obsahuje pouze identifikátor veřejného klíče CA, která ověřovaný certifikát vydala. Identifikátorem zpravidla bývá SHA-1 otisk veřejného klíče (nebo jeho část), pak se ověřuje tzv. **shoda klíčů** (*Key match*). Zajímavost je v tom, že se v tomto případě kontroluje jen shoda veřejného klíče, tj. jestli obsah rozšíření Identifikátor klíče úřadu v ověřovaném certifikátu se shoduje s obsahem rozšíření Identifikátor klíče předmětu v nalezeném certifikátu CA. Nekontroluje se tedy shoda Vydavatel (ověřovaného certifikátu) – Předmět (vydávající CA).



Obrázek 7.2: Využití shody klíčů na Slovensku (OID 2.5.4.41 identifikuje atribut „Name")

3. Ověřovaný certifikát vůbec neobsahuje rozšíření Identifikátor klíče úřadu. V takovém případě je možné jen ověřit vazbu Vydavatel (ověřovaného certifikátu) – Předmět (vydávající CA). Hovoříme pak o **Shodě jedinečných jmen** (*Name natch*).

Mechanismus shody klíčů nepřipadá čtenáři na první pohled asi příliš zajímavý. Ukažme si však, jak jej velice efektivně využili pro akreditované CA na Slovensku, kde zákon předepisuje, že kvalifikovaný certifikát musí být ověřitelný až k důvěryhodné kotvě, kterou je na obr. 7.2 certifikát „Korenova CA pre kvalifikované certifikaty 1" vydaný NBÚ. Jenže jednotlivé akreditované CA mají na Slovensku i své kořenové certifikáty. Např. certifikační autorita DTCA využívá kořenový certifikát certifikační autority I.CA akreditované v České republice.

Všimněte si (na obr. 7.2), že kvalifikovaný certifikát („RNDr. František Kaščák") vydaný certifikační autoritou DTCA tak může být ověřitelný jak k slovenské kotvě, tak i k české kotvě pro kvalifikované certifikáty. Pro správné fungování se předpokládá, že v rozšíření Identifikátor klíče úřadu je otisk z veřejného klíče vydávající CA a není tam jméno vydavatele a pořadové číslo certifikátu.

Certifikační politiky, nebo certifikační šablony?

Microsoft v případě tzv. Enterprise CA nejde primárně cestou certifikačních politik a omezení jmen a politik, ale jde cestou šablon. Vcelku tak lze očekávat, že místo ověřování certifikačních politik ověřuje využití certifikátu podle šablon.

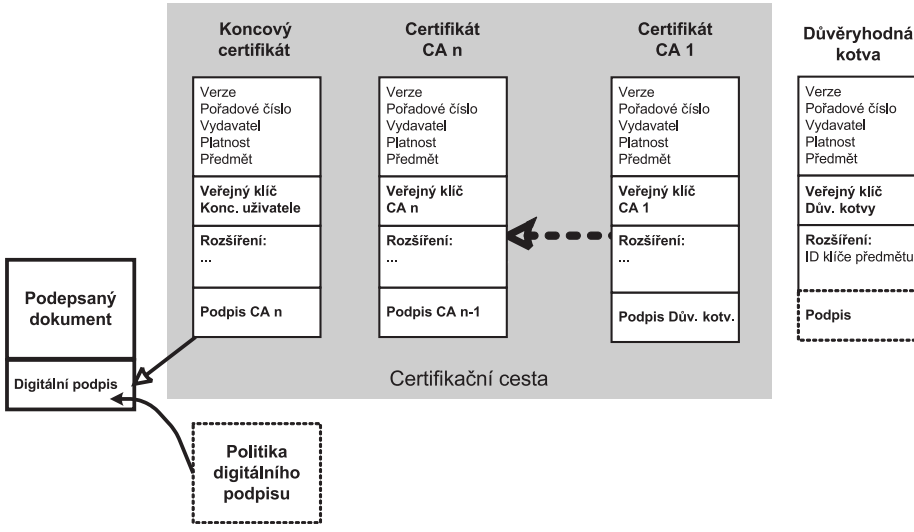
Microsoft ale také umožňuje využít i mechanismus certifikačních politik a omezení pomocí mechanismu „Qualified subordination", který se konfiguruje pomocí konfiguračního souboru `CAPolicy.inf` (umístěného ve %WinRoot%). V případě využití tohoto mechanismu pak lze i v prostředí Microsoft CA vydávat certifikáty mající:

- Rozšíření „Základní omezení" v plné syntaxi
- Rozšíření „Omezení jmen"
- Rozšíření „Certifikační politiky"
- Rozšíření „Mapování zásad (*Policy Mapping*)"
- Privátní rozšíření firmy Microsoft s názvem *Application policy* – jedná se o obdobné rozšíření jako „Rozšířené použití klíče". Tj. obsahuje identifikátory objektů účelů, ke kterým je certifikovaný veřejný klíč určen. V případě, že by certifikát obsahoval jak rozšíření *Application policy*, tak i „Rozšířená použití klíče", měla by obě rozšíření obsahovat identifikátory týchž objektů.

Ověřování podpisu

V případě ověření platnosti digitálního podpisu musíme nejprve ověřit certifikát koncového uživatele, který podpis vytvořil. Ověření koncového certifikátu je znázorněno na obr. 7.3. Po ověření koncového certifikátu máme k dispozici certifikát pro ověření digitálního podpisu. Můžeme tedy ověřit i podpis dokumentu.

Při ověřování platnosti certifikátu jsme při zjišťování, není-li certifikát uveden na CRL (tj. není-li odvolán), brali s povděkem CRL, jehož platnost dosud nevypršela (dosud nenastal čas *notAfter*). To mnohdy stačí, ale v případě digitálního podpisu nahrazujícího rukou psaný podpis může jít i o peníze. Pokud tedy chceme akceptovat koncový certifikát, nezapomínejme na přísloví, že je lépe měřit dvakrát. Možná vás napadne: Co když existuje novější CRL, na



Obrázek 7.3: Ověřování digitálního podpisu

kterém je koncový certifikát již odvolán? To bychom pak přišli k újmě. Takže si obstaráme to nejčerstvější CRL.

Ale stále nemáme vyhráno. Co když držitel koncového certifikátu již požádal o odvolání certifikátu, ale CA to jen ještě nestačila zpracovat a mezitím vydala CRL, na kterém koncový certifikát uveden ještě není, ale měl by (!). Prostě než ty peníze vyplatíme, tak si jednoduše počkáme na vydání dalšího CRL, abychom měli jistotu, a pak teprve provedeme výplatu.

Otázkou je, od jakého okamžiku máme čekat na vydání dalšího CRL. Potřebujeme také mít důkaz, že jsme opravdu čekali a podpis jsme akceptovali ve vhodném okamžiku. Takovým důkazem je vytvoření časového razítka z digitálního podpisu, které k podpisu může připojit podepisovaná osoba i příjemce dokumentu.

Při ověřování platnosti certifikátu jsme ověřovali certifikační politiky a nejrůznější omezení. K čemu nám je toto ověřování, když jej využijeme pouze pro ověření koncového certifikátu? Bylo by přece užitečné, kdybychom tyto informace mohli využít i pro ověření podpisu (např. toho, byl-li uživatel vůbec oprávněn takový podpis vytvořit). V kap. 25 si ukážeme, že pro ověření podpisu můžeme využít tzv. politiku digitálního podpisu. Politika digitálního podpisu může být textový dokument, ale též se může jednat o počítačově zpracovatelnou ASN.1 strukturu. Do digitálního podpisu pak lze vložit **podepisovaný** atribut obsahující otisk z politiky digitálního podpisu. Dále sem můžeme přidat i atributové certifikáty vyjadřující podpisová oprávnění (oprávnění právního výkonu). Tím se však budeme zabývat v kap. 25, která se pokouší řešit i problém platnosti archivovaných digitálně podepsaných dokumentů.

Kapitola 8

Obnovování certifikátů

Certifikát má omezenou dobu platnosti. Co si počít v okamžiku, když se blíží vypršení platnosti certifikátu, se budeme zabývat v této kapitole. V takovém okamžiku si totiž uživatel svůj certifikát obnovuje. Přirovnáme-li opět certifikát k občanskému průkazu, pak obnovování občanského průkazu též není věcí, na kterou by se občané těšili. Ještě zajímavější je však situace, když se blíží vypršení platnosti certifikátu certifikační autority.

V případě občanského průkazu si před koncem jeho platnosti občan dojde na příslušný úřad, prokáže se starým, ale ještě platným občanským průkazem, na základě čehož je mu vydán občanský průkaz nový. V případě, že se občan na úřad dostaví s již prošlým občanským průkazem, pak mu již úřad nedůvěřuje a občan musí absolvovat stejnou proceduru vydávání občanského průkazu, jakou musel projít, když mu byl občanský průkaz vydáván poprvé.

Vraťme se však k certifikátům. Držitel certifikátu má svůj jednoznačný Předmět (*Subject*) certifikátu. Certifikáty podle standardu X.509 od verze 3 umožňují, aby držitel certifikátu měl ve všech svých certifikátech tutéž hodnotu předmětu certifikátu. A to i v obnovených certifikátech. Držitel tak může mít např. jeden certifikát pro digitální podpis a jiný certifikát pro šifrování. A v obou certifikátech má tutéž hodnotu položky Předmět certifikátu. Pochopitelně, pokud oba tyto certifikáty obnoví, bude mít 4 certifikáty s toutéž hodnotou položky Předmět certifikátu.

Na tomto příkladu je vidět, že do položky Předmět bychom měli dát jen takové atributy, jejichž hodnoty se příliš nemění. Je pochopitelné, že při změně jména nebo bydliště je i tak jako tak nutné vystavit zcela nové certifikáty, tj. certifikáty s jinou hodnotou položky Předmět certifikátu. Pokud např. uvádíme e-mailovou adresu též do položky Předmět, tak se změnou e-mailové adresy musíme vydat i nový certifikát. A to je opravdu zbytečné. Takže se nesmíme divit, že standardy nám doporučují e-mailovou adresu neuvádět v položce Předmět, ale v rozšíření certifikátu Alternativní jméno předmětu.

Doporučení: Při obnově certifikátu neměníme obsah položky Předmět certifikátu. Drobné změny v rozšířeních certifikátu jsou ale v opodstatněných případech možné.

Snahou je, aby si koncový uživatel mohl obnovit certifikát ze svého počítače, aniž by se osobně dostavil na registrační autoritu. Pokud má uživatel ještě platný certifikát, může jej využít ke své autentizaci a nemusíme uživatele nutit se osobně s žádostí dostavit na registrační autoritu. Součástí žádosti o obnovení certifikátu musí tedy být i autentizace uživatele. Nejjednodušším způsobem je uložit žádost o nový certifikát do zprávy a tu digitálně podepsat platným certifikátem. Předpokladem je, že se musí jednat o certifikát určený k ověření podpisu (lze si představit i mechanismus pro „šifrovací“ certifikáty).

Tento princip je možné i zobecnit: Mnohé certifikační autority nehovoří o obnovených certifikátech, ale obecně o dalších certifikátech téhož předmětu. Certifikační autority jednoduše k předmětu vydají certifikát pro ověření digitálního podpisu. Držitel takového certifikátu si pak vytváří žádosti o další certifikáty (šifrovací, autentizační, libovolné obnovené atd.), které autentizuje svým digitálním podpisem. Certifikační autority tím řeší problém vydávání jiných certifikátů než těch, které jsou určené pro verifikaci digitálního podpisu (pomocí digitálního podpisu je asi autentizace nejpraktičtější).

A došlo to dokonce až tak daleko, že mnohé firmy si vytvoří smlouvu o vydávání zaměstnaneckých certifikátů s certifikační autoritou, kde stanoví konkrétní osoby (tj. předměty jejich certifikátů), které jsou zodpovědné za vystavování certifikátů serverů, jiných boxů apod. Tyto subjekty pak svým digitálním podpisem stvrzují žádosti o certifikáty pro tyto boxy. Ve své podstatě svými podpisy stvrzují žádosti o certifikáty jiných subjektů. Pomocí tohoto certifikátu také mohou žádat o odvolání certifikátu, např. zaměstnanců ukončujících pracovní poměr.

Renew, nebo Rekey?

V češtině se mohou pod obnovením certifikátu skrývat dva odlišné mechanismy:

- **Obnovení certifikátu téhož veřejného klíče (*Renew*):** Tj. v podstatě prodloužení certifikace téhož veřejného klíče. Nový certifikát se pak liší od původního pořadovým číslem, dobou platnosti a případně obsahem některých rozšíření. Tento mechanismus může být použit jak k prodloužení platnosti certifikace veřejného klíče, tak k vydání certifikátu jiného formátu téhož klíče (zpravidla o téže době platnosti). Můžeme např. vydat certifikát téhož klíče, ale např. standardu EDI. Nebo vydat certifikát s vypuštěním některých rozšíření, aby byl kratší a bylo jej možné využívat např. v přenosných zařízeních.
- **Obnovení certifikátu s vygenerováním nových párových dat (*Rekey*):** Pokud budeme dále uvádět spojení „obnovení certifikátu", budeme mít na mysli tuto eventualitu, kdy žadatel generuje nová párová data. Žádost o tento způsob obnovení certifikátu musí obsahovat:
  - Důkaz, že má žadatel v držení nový soukromý klíč – např. digitálním podpisem vytvořeným pomocí „nového" soukromého klíče.
  - Autentizaci uživatele. Tato autentizace může být provedena na základě důkazu o držení původního soukromého klíče, např. digitálním podpisem vytvořeným pomocí „starého" soukromého klíče.

I v případě obnovy certifikátů se vytváří žádost o certifikát. Všimněte si, že v případě *Renew* pro vydání nového certifikátu stačí původní (stará) žádost o certifikát. Na skutečnost, že jedna žádost o certifikát se uplatňuje vícekrát, jsme si už zvykli v případě křížové certifikace CA. Tam se ale táž žádost o certifikát uplatňovala u různých CA.

Avšak v případě *Rekey* je třeba, aby žadatel prokázal držení jak starého, tak nového páru klíčů. Z toho je vidět, že např. holá žádost tvaru PKCS#10 pro obnovení typu *Rekey* certifikátu nestačí.

Vydání dalšího certifikátu koncového uživatele

Jestliže CA nějakým způsobem ručí za údaje uvedené v certifikátu, bude se minimálně jednat o údaje v položce Předmět, tj. o identifikační údaje koncového uživatele.

Vraťme se však na počátek, kdy je uživateli vydán první certifikát. Uživatel se musí:

- Identifikovat, tj. musí předložit své identifikační údaje a dokázat jejich důvěryhodnost (např. předložením příslušných papírových dokladů).
- Autentizovat, tj. musí dokázat, že předložené identifikační údaje jsou jeho (např. shodou jeho podoby s fotografií na předložených dokumentech).
- Musí prokázat držení příslušného soukromého klíče.
- Případně musí dokázat, že párová data byla vygenerována zařízením splňujícím předepsané bezpečnostní požadavky (např. byla vygenerována konkrétní čipovou kartou či HSM modulem).

Pokud má koncový uživatel k dispozici platný certifikát určený např. pro digitální podpis, pak (lze to obdobně postavit i např. na certifikátech určených pro šifrování či autentizaci):

- Jeho identifikace byla provedena při vystavování původního platného certifikátu.
- Autentizaci může provést např. na základě digitálního podpisu vytvořeného pomocí starých, ale stále platných párových dat.
- V případě *Rekey* se navíc provede důkaz:
  - O držení „nového“ soukromého klíče (obdobně jako v případě vydání původního („starého“) certifikátu).
  - Je-li to předepsáno certifikační politikou, provede se též důkaz o vygenerování párových dat na příslušném zařízení (obdobně jako v případě vydání „starého“ certifikátu).

Pokud se koncový uživatel musel v případě vydání prvního certifikátu osobně dostavit na CA a prokázat svou totožnost, v případě vydání dalších certifikátů tomu tak již být nemusí. Jediné, co se ale takto neověří, je skutečnost, zda uživatel nezměnil některé identifikační údaje (např. jméno nebo adresu).

Aby šlo takto hladce certifikát obnovit (resp. vydat další certifikát), musí být starý certifikát ještě platný. Proto většina CA nějakým způsobem (např. e-mailem) upozorňuje koncového uživatele, že se blíží vypršení jeho certifikátu (např. e-mailem měsíc před vypršením certifikátu).

Poznámka: Skutečnost, že starý certifikát není platný, může být dána dvěma příčinami:

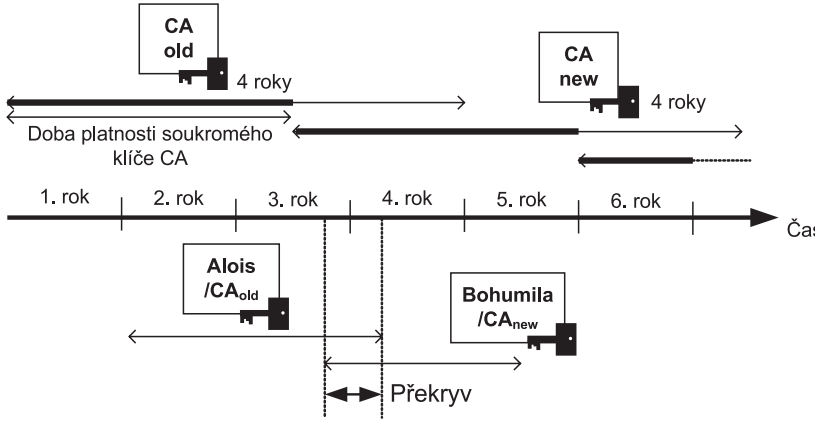
- Vypršela jeho přirozená platnost, pak před vypršením jeho platnosti jej lze popsaným způsobem obnovit.
- Certifikát byl odvolán. V takovém případě je třeba vystavovat další certifikáty koncovému uživateli stejným způsobem, jako když žádal o první certifikát! Tj. odvolaný certifikát nelze použít k prokázání totožnosti při obnovování certifikátu.

Obnovení certifikátu CA

Již při objasňovaní rozšíření certifikátu „Doba platnosti soukromého klíče“ (obr. 3.8 a 8.1) jsme si objasnili, že nový certifikát CA je nutné vydat s takovým předstihem, aby platnost certifikátů vydávaných koncovým uživatelům neskončila později než platnost certifikátu CA, kterým se tyto certifikáty koncových uživatelů ověřují.

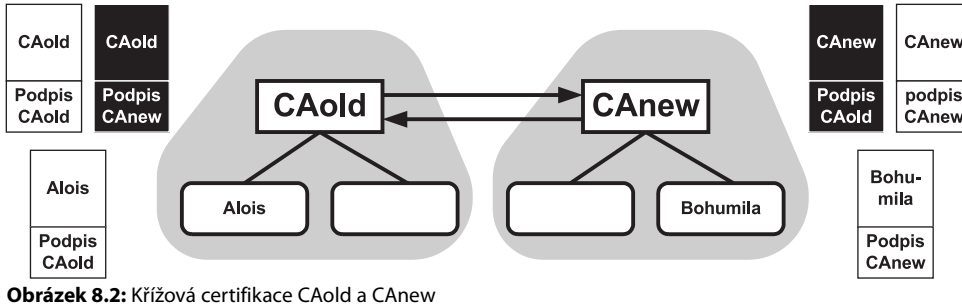
Starý certifikát CA budeme označovat jako CAold a nový certifikát CA budeme označovat jako CAnew.

Jenže v době, kdy platí oba certifikáty CA (CAold i CAnew), má Alois certifikát podepsán CAold a Bohumila CAnew.



Obrázek 8.1: Alois a Bohumila mají po překryvnou dobu každý vydán certifikát jakoby jinou CA

Pokud by Alois důvěřoval pouze CAold a Bohumila pouze CAnew, nemohou spolu důvěryhodně komunikovat. Řešením je křížová certifikace CAold s CAnew (obr. 8.2).



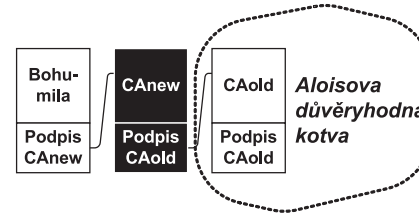
Obrázek 8.2: Křížová certifikace CAold a CAnew

I když má Bohumila certifikát vydaný certifikační autoritou CAnew, Alois – důvěřující pouze certifikační autoritě CAold – je schopen nalézt řetězec až ke své důvěryhodné kotvě (obr. 8.3).

Vzniknou nám tak čtyři certifikáty téže certifikační autority, které se často označují jako:

- OldOld – „stará“ certifikační autorita CAold
- NewNew – „nová“ certifikační autorita CA new

- OldNew – křížová certifikační autorita, jejíž certifikát vznikne tak, že se původní žádost o certifikát certifikační autority CAold vezme a uplatní se na CAnew.
- NewOld – křížová certifikační autorita, jejíž certifikát vznikne tak, že se žádost o certifikát nové certifikační autority CAnew vezme a uplatní se na staré certifikační autoritě CAold.



Obrázek 8.3: I přesto, že Bohumila má certifikát vydaný certifikační autoritou CAnew, je Alois schopen nalézt řetězec až ke své důvěryhodné kotvě, kterou je certifikát CAold (OldOld)

Všimněte si několika zajímavostí:

- Poslední certifikát vydaný CAold by měl být NewOld.
- Pro vystavení certifikátu OldNew potřebuje původní žádost, pomocí které byl vystaven OldOld (pokud je naše CA kořenovou CA, můžeme její certifikát OldOld považovat za žádost ve tvaru kořenového certifikátu).
- Doba platnosti certifikátů OldNew a NewOld by neměla přesahovat dobu, po kterou se překrývá platnost certifikátu OldOld s certifikátem NewNew.
- Všechny uvedené certifikáty mohou mít stejnou hodnotu položky Předmět.
- NewOld i OldNew se nemusí distribuovat důvěryhodnou cestou, protože nejsou kořenovými certifikáty.

CRL

Jestliže se při obnovování certifikátu CA změní předmět tohoto certifikátu, není o čem hovořit. CAold pokračuje ve vydávání CRL až do vypršení původní platnosti všech certifikátů, které vydala. CAnew si vydává od počátku svá CRL. Microsoft např. doporučuje u svých CA v případě, že CRL jsou příliš rozsáhlá, obnovit certifikát CA, aby se CRL již dále příliš nezvětšovalo. Nazývá to *CRL partitioning*.

Avšak pokud je předmět staré i nové certifikační autority totožný, vyvstává otázka: Nestačilo by vydávat jednu řadu CRL? Od jistého okamžiku by se CRL jen verifikovalo certifikátem „nové“ certifikační autority. Jenže to by museli mít všichni „staří“ uživatelé okamžitě k dispozici „nový“ certifikát. Praxe je taková, že se zpravidla vydávají dvojí CRL, tj. certifikáty vydané „starou“ CA mají jiné distribuční místo CRL než certifikáty vydané „novou“ CA.

Doba platnosti certifikátu

Jedno z nejtěžších rozhodnutí provozovatele certifikační autority je určení délky platnosti certifikátu CA. Nejjednodušší je stanovit tuto dobu co nejdelší. Jenže pak si zahráváme s kryptografickou nedostatečností párových dat a funkce pro výpočet otisku po uplynutí dlouhé doby.

Bezpečnější se tedy jeví naopak stanovit dobu platnosti certifikátu CA co nejkratší. Jenže to zase přináší nutnost časté obnovy certifikátu CA, která je organizačně náročná a může během ní dojít i k nejrůznějším útokům na distribuci obnovených certifikátů CA.

Je tedy nutné stanovit kompromis. Ten ale bude jiný u ryze komerčních certifikačních autorit a jiný u autorit vydávajících kvalifikované certifikáty, které mají nahrazovat občanské průkazy. Neexistuje nějaké univerzální doporučení. Rozhodnutí je na vás.