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

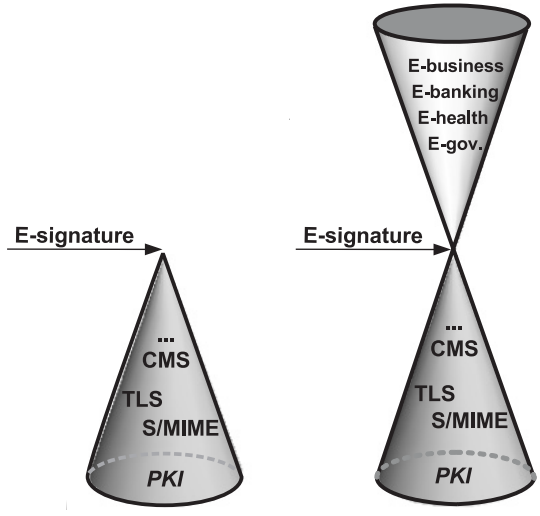
Kapitola 25

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.



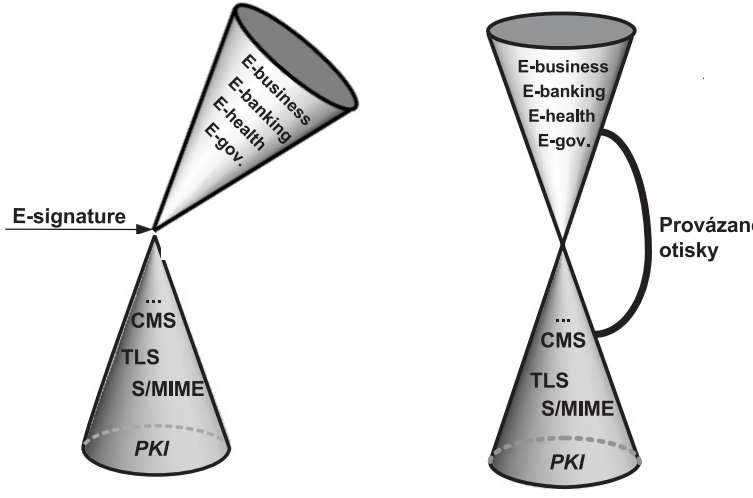
Obrázek ú.1: Sylvesterovy kužely

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.

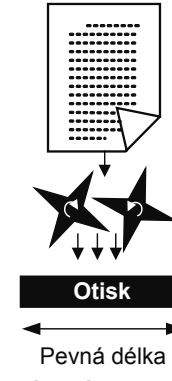
Kapitola 1

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.

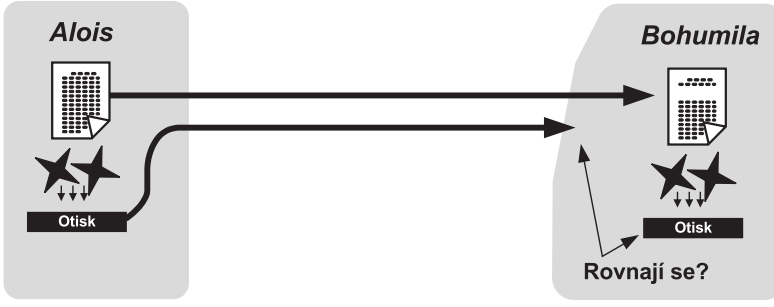
Kvalitní jednocestné funkce pro výpočet otisku by měly dát výrazně jiný výsledek při drobné změně původního textu. Počítáme-li např. otisk pro digitální podpis z textu nesoucího platební příkaz, pak by bylo nemilé, kdyby se nám po připsání nuly k převáděné částce otisk nezměnil.

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).

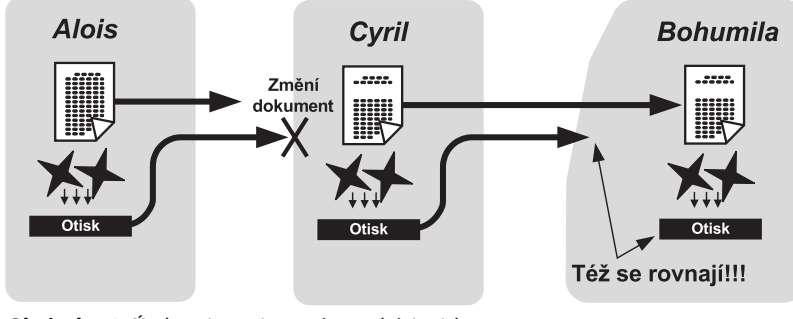
Jak využije Alois otisk ve své komunikaci se svou milou Bohumilou? No přece využije otisk jako důkaz, že zpráva na cestě od Aloise k Bohumile nebyla změněna, tj. využije otisk jako důkaz integrity zprávy. Alois neodešle pouze samotná data (text) zprávy, ale data doplní o patu zprávy (*trailer*) obsahující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

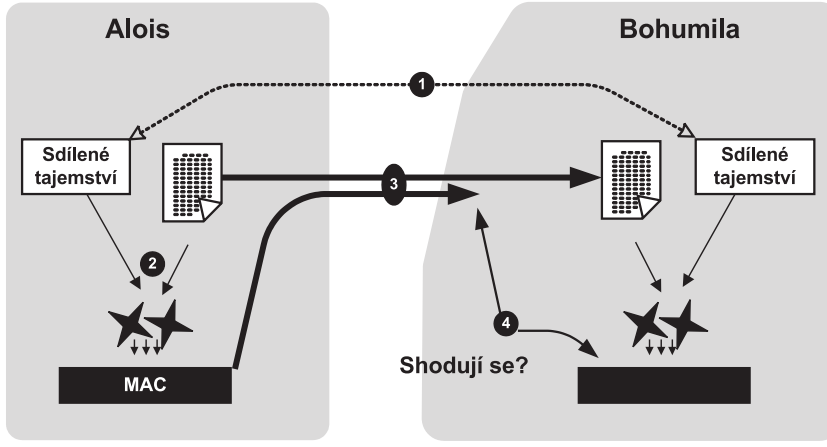
Jenže algoritmus výpočtu otisku používaný Aloisem je veřejně popsán v příslušné technické normě. Tuto normu si přečte i žárlivý Cyril, který zprávu od Aloise pozmění v jeho neprospěch a ten pošle Bohumile (obr. 1.3).



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.

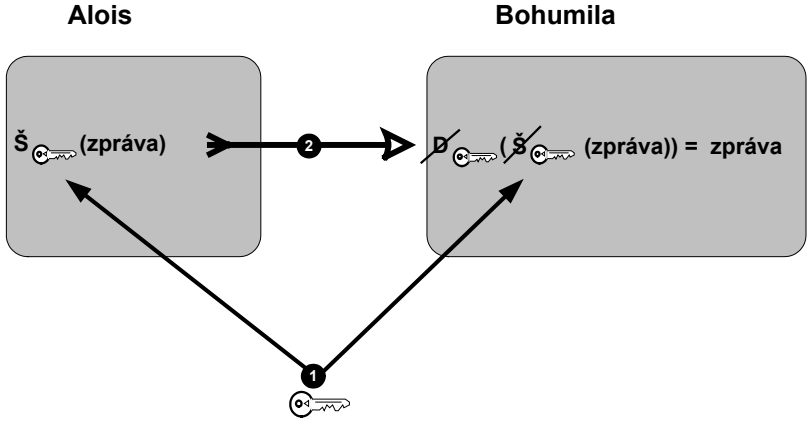
* MAC se někdy do češtiny překládá jako „kryptografický kontrolní součet".

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.

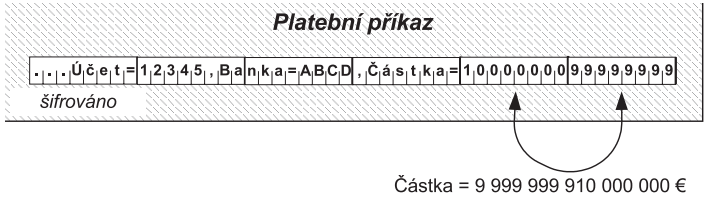
Symetrických šifrovacích algoritmů je celá řada. Snad nejrozšířenějším je algoritmus DES, používající šifrovací klíč délky 56 bitů. Dnes se však považuje za nedostatečný. Z algoritmu DES byl odvozen algoritmus 3DES s klíčem 112 bitů nebo 168 bitů*. Dále se používají algoritmy s délkou klíče 128 bitů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

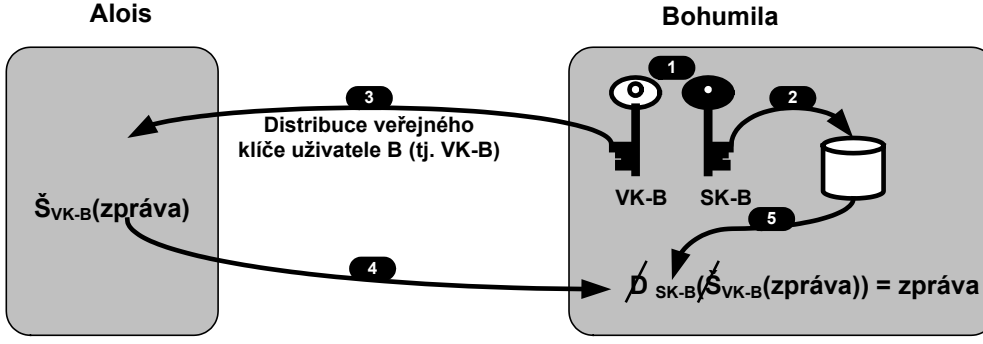
Asymetrické šifry

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.

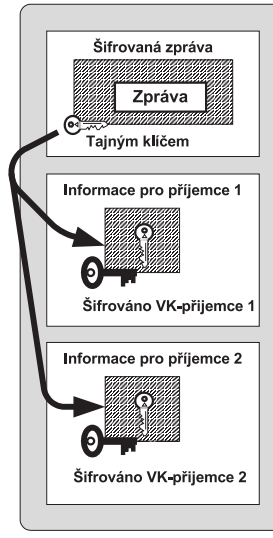
Elektronická obálka

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

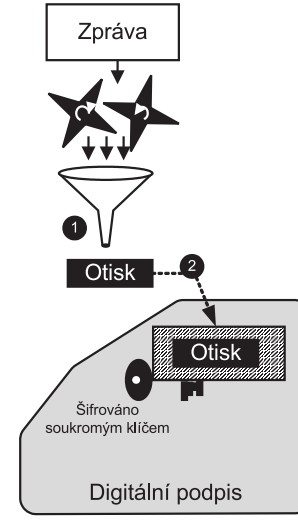
Digitální podpis

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:

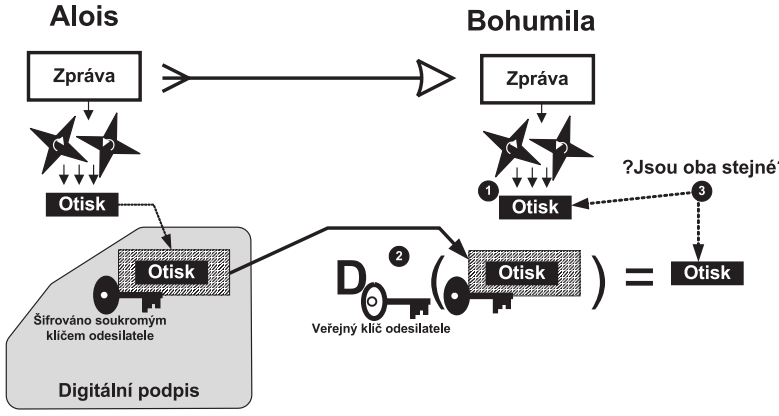
1. Příjemce samostatně spočte otisk z přijaté zprávy.
2. Příjemce dešifruje přijatý digitální podpis veřejným klíčem odesilatele.
3. Příjemce porovná výsledek získaný z bodu 1 s výsledkem získaným z bodu 2. Pokud jsou stejné, pak mohl digitální podpis vytvořit pouze ten, kdo vlastní soukromý klíč odesilatele – 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.

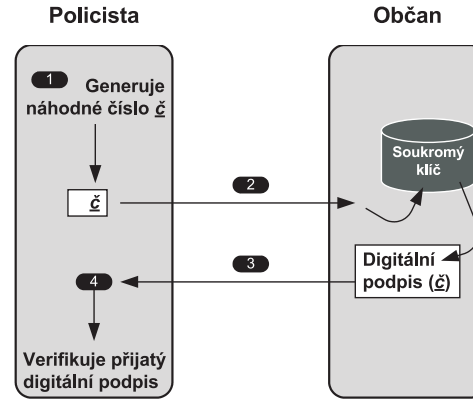


Obrázek 1.10: Verifikace digitálního podpisu

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.

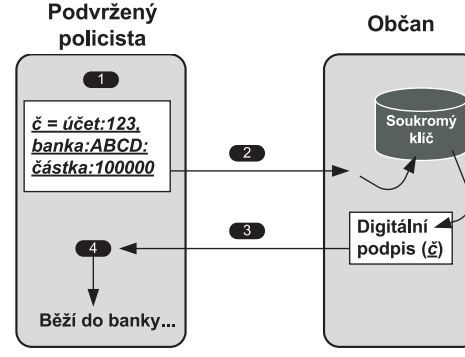
V případě, že nechceme využít digitální podpis, ale výhradně šifrování, pak policista náhodné číslo utají a občanovi jej zašle šifrované jeho veřejným klíčem, občan jej dešifruje soukromým klíčem a dešifrované vrátí policistovi. Policista zkontroluje, rovná-li se přijaté číslo tomu, které šifroval veřejným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

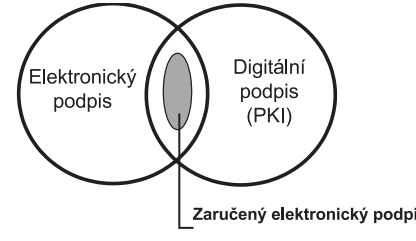
Digitálním podpisem budeme rozumět podpis vytvořený na základě asymetrické kryptografie, tak jak je popsán v této kapitole.

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

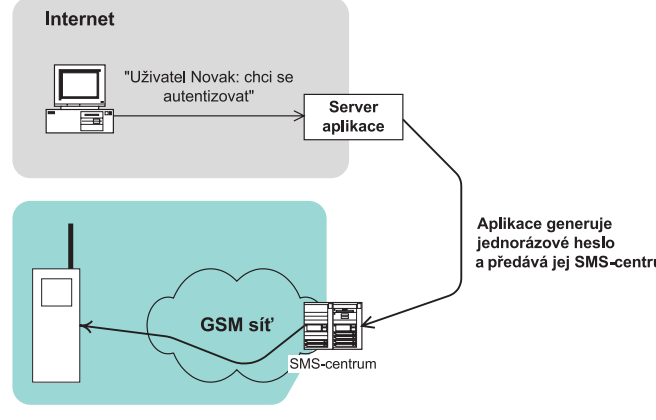
Ke generování jednorázových hesel můžeme místo otisku využít též symetrickou šifru. Namísto sdíleného tajemství sdílí klient se serverem symetrický šifrovací klíč, který využije k šifrování: času, počtu přihlášení či náhodně generovaného dotazu.

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

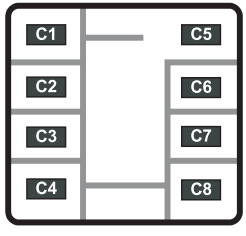
ISO 7816-9 Příkazy pro správu karty

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.

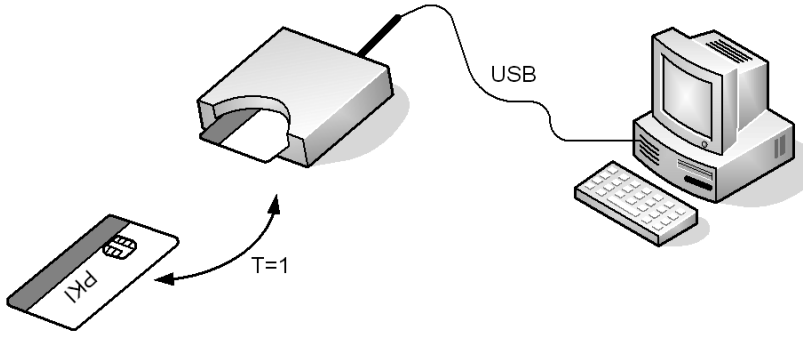
Čtečky čipových karet (terminály)

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.

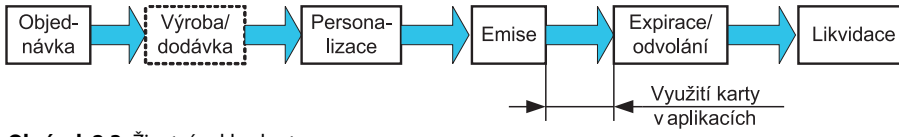
V poslední době se využívají technologie potisku, které nemohou poškodit čip. Vzniká proto i opačný problém. Karta v místě bezkontaktního čipu zpravidla vykazuje jisté nerovnosti, které by naopak mohly deformovat tisk. Výsledkem je tedy stejné ponaučení, aby se nad/pod bezkontaktní čip netisklo.

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.

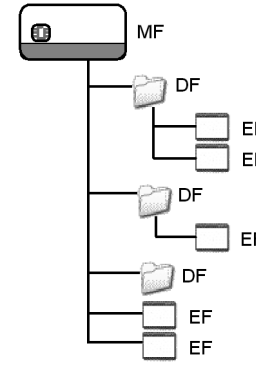
Životní cyklus karet se tak skládá z (obr. 2.3):

- 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).

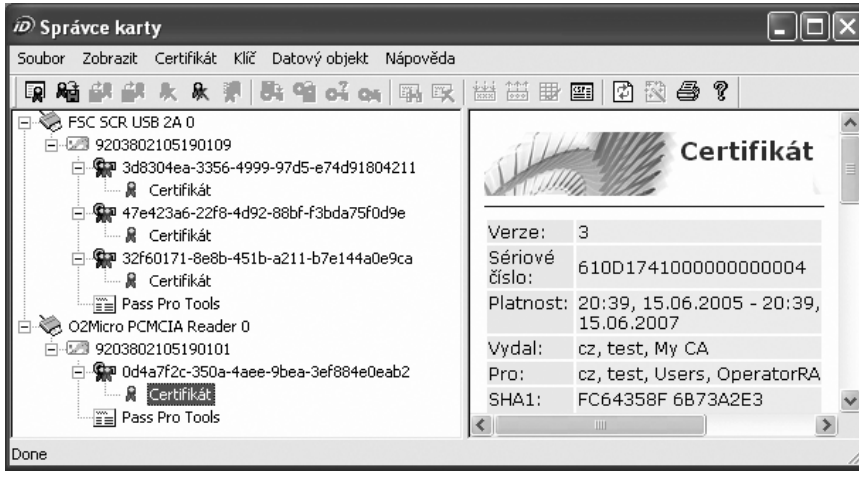
Praktické využití *Secure Messaging* spočívá např. v tom, že zatímco doma se bude držitel vůči kartě autentizovat pomocí PIN, na registrační autoritě od něj PIN nemusí být vyžadován, když s kartou uživatele manipuluje pracovník registrační autority.

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.

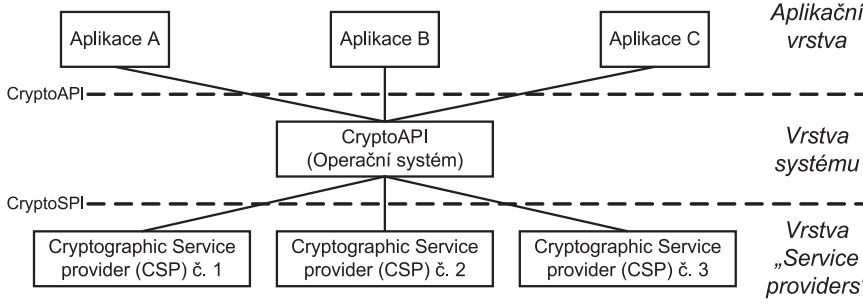
Kromě návrhu samotné čipové karty musí architekt navrhnout i obslužný software (middleware) pro podporu karty v jednotlivých operačních systémech.

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.



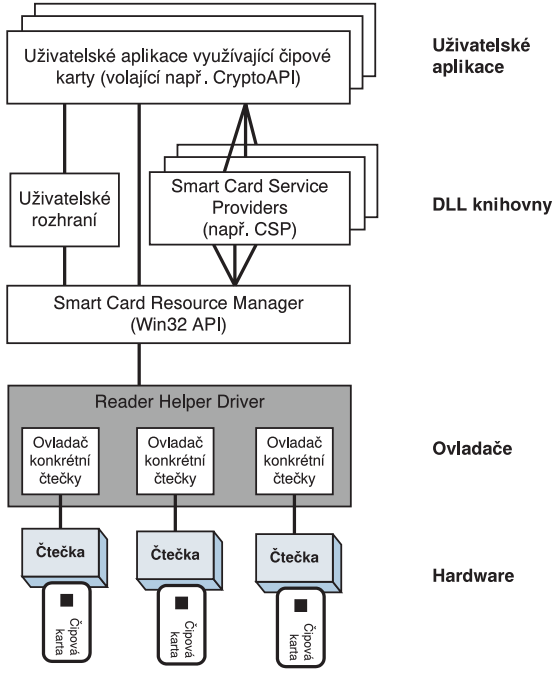
Obrázek 2.6: Microsoft řeší podporu čipových karet pomocí CSP

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.

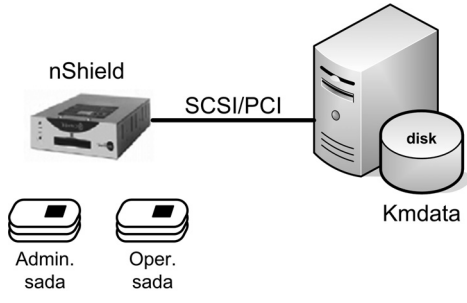
HSM (*Host Security Modul*)

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.).

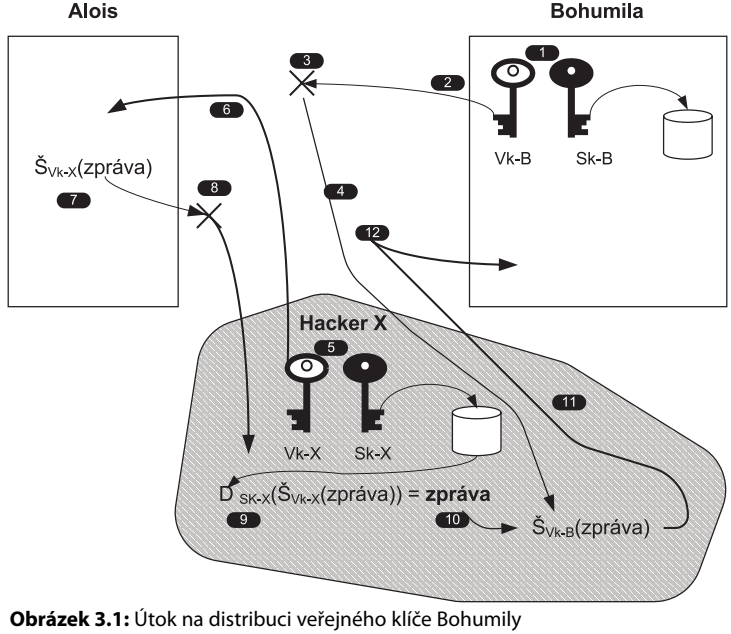
Zajímavé je, že mnohdy se volí PKI čipová karta anebo autentizační kalkulátor a hledají se důvody pro jednu nebo druhou pomůcku. Je jasné, že PKI čipovou kartu autentizačním kalkulátorem nelze nahradit, a obráceně, ani kalkulátor nelze nahradit PKI čipovou kartou. Cestující by tak měl být vybaven oběma 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