Motto

Existují dva typy šifrování: takové, které znemožní vaší mladší sestře číst vaše soubory a takové, které znemožní vládám velmocí číst vaše soubory.

Bruce Schneider, Applied Cryptography

Co je to bezpečný informační systém?

Počítačová bezpečnost či bezpečnost dat jsou pojmy značně obsáhlé a mohou být chápány v několika úrovních. Informační systém je systém, kde jsou zpracovávána a uchovávána data, která jsou nositeli informací. Informační systém zahrnuje hardware, software a vlastní data. Tyto tři složky jsou aktiva, která je nutno zabezpečit proti hrozbám a útokům pasivním i aktivním. Bezpečný IS lze definovat jako systém, který chrání informace během jejich vstupu, zpracování, uložení, přenosu a výstupu proti ztrátě dostupnosti, integrity a důvěrnosti a po jejich likvidaci proti ztrátě důvěrnosti. Bezpečnost IS je velmi rozsáhlý problém, který tvoří řetěz složený z článků - jednotlivých podoblastí bezpečnosti. Síla řetězu je rovna síle nejslabšího článku.

Problematika bezpečnosti informačních systémů je u nás často podceňována, protože dokáže spolykat značné množství zdrojů, aniž přitom generuje jakýkoli zisk. investice do bezpečnosti sice nevydělávají, ale chrání společnost před možnou ztrátou.

Kdo je typickým útočníkem?

• ***hackeři***, usilují prokázat své mimořádné schopnosti úmyslným prolomením ochrany systému, konají pro vlastní zábavu, útočí přes internet, nejsou příliš nebezpeční

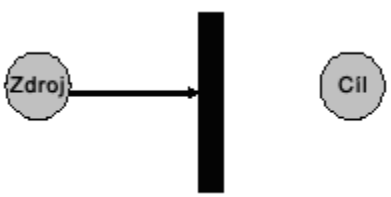
• ***amatéři,*** dostanou se do IS prostřednictvím Internetu přes náhodně objevená slabá místa

• ***profesionální zločinci***, slouží zájmům cizích organizací (průmyslová nebo obchodní špionáž). Tato napadení nejsou častá, ale přinášejí obrovské ztráty, mohou vést až ke zničení společnosti.

• ***„důvěryhodné“ osoby***, naprostá většina útoků připadá na osoby, kterým byla dána určitá důvěra - vlastní zaměstnance, obchodní partnery, specializované firmy najaté na konkrétní práci.

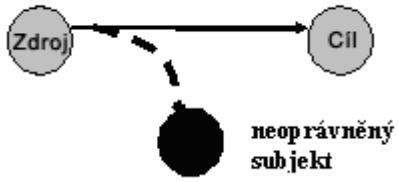
V oblasti bezpečnosti dat jsou specifikovány čtyři typy hrozeb namířených proti bezpečnosti informačního systému:

Přerušení
Aktivum IS se ztratí, stane se nepoužitelným nebo nepřístupným. Příkladem je zničení hardware zařízení, výmaz programu nebo datového souboru nebo selhání operačního systému při vyhledávání souboru na disku.



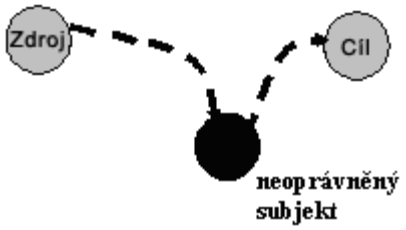
Odposlech

Neoprávněná strana získá přístup k aktivu. Neoprávněná strana může být jak osoba, tak program nebo informační systém. Příkladem může být nepovolené kopírování programů nebo datových souborů nebo tajný odposlech prováděný při datových přenosech po síti. Ztráta aktiva může být odhalena bezprostředně, ale "tichý" odposlech nemusí zanechat stopy, podle nichž by byl odhalitelný.



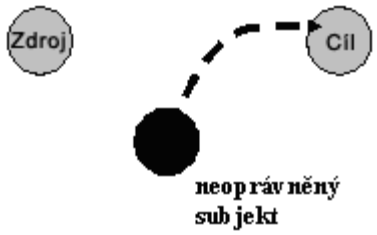
Pozměnění

Neoprávněná strana nejenže získá přístup k aktivu, ale také tohoto přístupu využije k jeho pozměnění. Příkladem může být nelegální provedení změn v datech databází, pozměnění programů tak, že se ovlivní jejich běh, nebo provedení změn na datech v průběhu jejich přenosu v síti. V některých případech se pozměnění aktiva projeví ihned, ale některé případy je téměř nemožné detekovat.



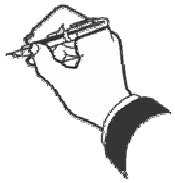
Vytvoření falsifikátu

Určitý objekt byl vytvořen neoprávněnou stranou. Příkladem může být neoprávněné vložení falešných záznamů do databází nebo vytvoření falešné zprávy a její následné odeslání po síti.



V reálných situacích dochází často ke kombinaci dvou i více základních způsobů ohrožení dat.

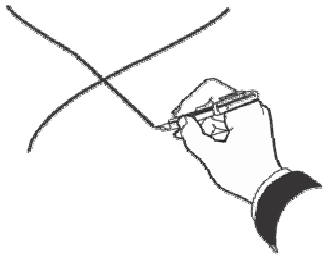
Úloha digitálního podpisu



Společným jmenovatelem, který umožňuje dálkový přenos dokumentů a informací je digitalizace, tedy proces, který z klasických "papírových" dokumentů vytvoří dokumenty digitální, dokumenty přeměněné do "řeči" čísel. Informační dálnice, např. městská počítačová nebo globální síť sítí - Internet, dopraví digitalizované dokumenty bezpečně na místo určení, na příslušnou digitální adresu.

Dokumenty, které se touto formou přenášejí mají většinou informativní, osobní či služební charakter. V případě právních dokumentů procházejí elektronickou poštou často návrhy smluv, které účastníci právního řízení vzájemně dolaďují a připravují jejich konečnou verzi. Připravené dokumenty pak ve svém konečném stádiu realizace opouštějí kybernetický prostor, aby na sebe vzaly klasickou papírovou podobu. Ta totiž jako jediná umožní provést poslední akt, kterým je právoplatný podpis dokumentu. A jsme u jádra věci. Dokumenty zatím nedokážeme ošetřit tak, aby je bylo možné právoplatně podepsat a stvrdit svoji právní odpovědnost za obsah dokumentu.

Kvalitativně vyšší úroveň práce s elektronickými dokumenty zajistí poslední zdánlivá maličkost – právně správný a ověřitelný digitální nebo chcete-li elektronický podpis.



Současný stav podepisování elektronických dokumentů či pošty je důvěrně znám. Dopisy včetně těch obchodních a služebních podepisujeme formou textového řetězce, sady znaků, kterou zapíšeme svoje jméno. Dalším způsobem je vložit do dopisů obrázek obsahující skenovaný podpis odesilatele. Ani ten však právní problém neřeší. Takový podpis lze velmi snadno kopírovat a vložit do dokumentu zcela jiného. Samozřejmě, že hlavička elektronické pošty obsahuje informace, ze kterých lze vystopovat odesilatele, ale po stránce právní jsou i takto zdánlivě "ručně" podepsané dopisy naprosto bezcenné. Navíc dáváme všanc grafickou podobu svého podpisu, které může být zneužito mnoha způsoby.

Problém vyřeší až nasazení digitálních podpisů. Digitální podpis je velmi složitý, zašifrovaný číselný kód, který je pro každého uživatele ojedinělý obdobně jako otisk prstu, a který je právně ověřitelný. Podstata digitálního podpisu spočívá v "označkování" elektronického dokumentu, ze kterého je zřejmá nezpochybnitelná identita autora. K podepisování dokumentu slouží privátní, tajné klíče. Ke čtení dokumentu a ověření podpisů slouží veřejné klíče.

Vidina digitálního podpisu je docela příjemná a lákavá. V rozvinuté informační společnosti lze digitální podpis uplatnit např. v bankách, u lékaře a v mnoha dalších případech. Výhody digitálního podpisu ocení především Ti, kterým se ne vždy a napoprvé podaří vykouzlit stejnou grafickou podobu podpisu. U digitálního podpisu tento problém odpadá, navíc ověření podpisu je nesrovnatelně jednodušší a rychlejší.

Hlavní cíle a zásady bezpečnosti elektronické komunikace

Obvykle si lidé slučují pojem bezpečnost se šifrováním, resp. s nerozluštitelností důvěrných elektronických dat. Ve skutečnosti je problém poněkud složitější. Zpravidla cíle zabezpečení dat při jejich výměně a použití rozdělujeme na tři zásady:

- zásada důvěrnosti
- zásada neodmítnutelnosti odpovědnosti
- zásada integrity

Zásada důvěrnosti vyjadřuje potřebu uložit data tak, aby jejich obsah mohl přečíst jen ten, komu jsou určena, přičemž kdokoliv další nemá šanci obsah rozluštit ani za pomoci nejmodernějších technologií.

Zásada neodmítnutelnosti odpovědnosti vyjadřuje neméně důležitou potřebu možnosti dokázat, kdo je autorem zprávy. Zde nejde o utajení, ale naopak o průkaznost původu dat. Požadavek neodmítnutelnosti odpovědnosti bývá často v praxi splněn digitálním, elektronickým podpisem.

Zásada integrity má na starosti, aby data došla nejen úplná, ale též prokazatelně nezměněná.

Pokud zprávy budou všechny tři zásady respektovat, pak je jejich bezpečnost stoprocentně zaručena.

Integritu lze vyřešit pomocí elektronického podpisu. Ten beze zbytku zajistí, že v dokumentu během přenosu nebyly provedeny žádné změny. Elektronický podpis ovšem nic neříká o fyzické identitě autora dokumentu, protože kdokoli může zveřejnit svůj veřejný klíč pod jakýmkoliv jménem. Příjemce dokumentu potřebuje spolehlivou informaci o identitě vlastníka veřejného klíče. Tuto informaci může získat např. fyzickým kontaktem s odesílatelem. Uvědomme si ale, že jednotlivé entity se často vůbec neznají (např. při styku instituce-firma, firma-klient), nebo se znají jen „na dálku“ prostřednictvím telefonického spojení, faxu či Internetu. Je tedy nutné potvrdit identitu vlastníka elektronického podpisu pomocí třetí důvěryhodné a nezávislé strany.

Funkci třetí strany zde plní certifikační autorita. Certifikační autorita tedy zajišťuje autenticitu dokumentů. Z toho logicky vyplývá, že systém veřejných klíčů spolu s certifikační autoritou řeší i otázku **neodmítnutelnosti** dokumentů. **Důvěrnost** dokumentů je pak záležitostí vhodného šifrování (kryptografie). Dnes hojně používaným způsobem je protokol SSL – Secure Sockets Layer. Ten je určen primárně pro přenos dat Internetem a poznáte jej podle přidaného „s“ v https:// nebo „zamčeného zámku“ v prohlížeči.

Kryptografie

Kryptografie je věda, zabývající se šifrováním – tedy utajením informací. V počítačovém prostředí slouží kryptografie k ochraně dat před neautorizovaným odhalením. Může sloužit k autentizaci uživatele či procesu, požadujícího nějakou službu, nebo může zabránit neautorizovaným modifikacím. Kryptografie představuje nezanedbatelnou součást moderní počítačové bezpečnosti.

Šifrování a dešifrování

Šifrování je proces, při němž se zpráva (nešifrovaný text) transformuje na jinou zprávu (zašifrovaný text) pomocí matematické funkce a speciálního šifrovacího hesla, tzv. klíče.

Dešifrování je opačný proces: zašifrovaný text se pomocí matematické funkce a klíče převede zpět na text nešifrovaný.

Šifrování může hrát významnou úlohu při každodenní komunikaci a práci s počítačem:

- pomocí šifrování můžeme chránit informace uložené na našem počítači před neautorizovaným přístupem – a to dokonce i před lidmi, kteří jinak mají k našemu počítačovému systému přístup.
- šifrováním můžeme chránit informace při přenosu z jednoho počítače na druhý.
- šifrováním můžeme zabránit či detekovat náhodné nebo úmyslné změny dat.
- pomocí šifrování je možno ověřit, zda autorem dokumentu je opravdu ten, kdo myslíme.

I přes tyto výhody má šifrování i určitá omezení:

- šifrováním nemůžeme útočníkovi zabránit v úplném vymazání našich dat.
- útočník může porušit samotný šifrovací program. Může jej modifikovat tak, že bude používat jiný klíč než zadáme, nebo může šifrovací klíče někam zaznamenávat pro pozdější použití.
- útočník může objevit dříve neznámý relativně snadný způsob dekódování zpráv, zašifrovaných naším algoritmem.
- útočník může získat soubor před nebo po jeho zašifrování nebo dešifrování.

Pro všechny tyto důvody je třeba šifrování chápat pouze jako součást celkové bezpečnosti našeho systému, ne jako náhradu za ostatní metody, například za správné řízení přístupu.

Součásti šifrování

Existuje celá řada způsobů, jak můžeme počítač použít k zašifrování nebo dešifrování dat. Bez ohledu na algoritmus však všechny šifrovací systémy používají shodné základní prvky:

Šifrovací algoritmus
Šifrovací algoritmus je funkce, obvykle sestavená na nějakém matematickém základě, která provádí samotné šifrování a dešifrování dat.

Šifrovací klíč
Šifrovací klíč říká šifrovacímu algoritmu, jak má data šifrovat nebo dešifrovat. Klíče se podobají počítačovým heslům: jakmile informaci zašifrujete, musíte k jejímu dešifrování zadat správný klíč. Na rozdíl od kontroly hesel však šifrovací program neporovnává klíč s dříve zadaným klíčem a nepovoluje přístup tehdy, pokud se klíče shodují. Namísto toho šifrovací algoritmus přímo používá klíč při transformaci zašifrovaného textu zpět do nezašifrované podoby. Zadáte-li správný klíč, dostanete zpět původní zprávu. Budete-li data šifrovat špatným klíčem, dostanete nesmysly.

Délka klíče
Stejně jako hesla, i klíče mají nějakou předem určenou délku. Delší klíče jsou bezpečnější než kratší klíče, protože při použití útoku hrubou silou skýtají více kombinací. Různé šifrovací systémy umožňují použití klíčů různých délek, některé dovolují použití klíčů s proměnnou délkou.

Nešifrovaný text
Informace, které chceme zašifrovat.

Zašifrovaný text (šifra)
Informace po zašifrování.

Síla šifry

Různé kryptografické metody nejsou ekvivalentní. Některé systémy se dají velmi snadno obejít či prolomit. Jiné daleko lépe vzdorují i mnohem systematičtějšímu útoku. Schopnost kryptografického systému chránit informace před útokem se označuje jako jeho síla.

Síla systému závisí na mnoha okolnostech, například:

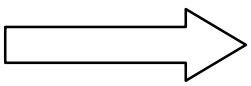
- utajení klíče.
- obtížnost uhodnutí klíče nebo vyzkoušení všech možných klíčů (tzv. hledání klíče). Obecně platí, že čím delší klíč, tím hůře se dá uhodnout či najít.
- obtížnost otočení šifrovacího algoritmu bez znalosti klíče (prolomení šifrovacího algoritmu).
- existence (či neexistence) zadních vrátek, tedy metody, jak je možno data relativně snadno dešifrovat i bez znalosti klíče.
- možnost dešifrovat celý text v případě, že z části znáte jeho nezašifrovanou podobu (tzv. útok se znalostí textu).
- vlastnosti nezašifrovaného textu, které jsou známy útočníkovi. Například některé systémy se dají snadno prolomit, pokud všechny jimi šifrované zprávy začínají nebo končí stejným kusem textu.

Cílem kryptografického návrhu je vyvinout algoritmus, který se dá bez klíče prolomit tak těžko, že to bude zhruba odpovídat námaze nutné ke zjištění klíče prohledáváním celého prostoru klíče. Tato schopnost by měla zůstat zachována i v případě, že útočním má nějaké informace o zašifrované zprávě. Na návrzích šifer obvykle spolupracují špičkoví matematici.
Jednoduchý (klasický) případ pro šifrování je historická Césarova šifra – o níž je známo, že byla používána samotným G. J. Césarem, císařem římským. Text je zašifrován pomocí posunutí písmen, například změnou písmene A na D, B na E, C na F a tak dále. V tomto případě je klíčem počet míst, o který jsou písmena posunuta v abecedě (např. tedy 3).

Jiným případem je použití blokové substituční metody, která převádí n-bitové bloky otevřeného textu na n-bitové bloky šifrového textu. K šifrování se používá substituční tabulka. Do historie se zapsalo polyalfabetické šifrování (několik substitučních tabulek), které používal za 2. světové války německý šifrovací stroj Enigma. Stroj prováděl poměrně složité operace se vstupním textem, ale zároveň se dal poměrně snadno ovládat. Poláci ještě před vypuknutím války pracovali na prolomení šifry a jejich zjištění byla později nedocenitelná pro spojenecké armády. Šifra byla již během války zlomena a poskytovala tak německé straně pouze falešný pocit bezpečí. Enigma měla původně 3 kolečka, která se otáčela, podobě jako mechanické počítadlo a každé mělo jinak propojené vstupní a výstupní kontakty, tím se měnil průběh proudu a i výsledné písmeno zašifrovaného textu. Později začalo ponorkové námořnictvo používat čtyřrotorové Enigmy.

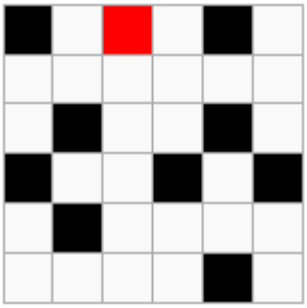
Velkou popularitu si také získaly tzv. transpoziční metody, které zachovávají abecedu a mění pouze pořadí znaků podle určitého pravidla. Například tak, že otevřený text je zapsán do tabulky pořádcích a šifrový text vznikne čtením sloupců téže tabulky.

š	i	f	r	o
v	á	n	í	p
o	m	o	c	í
t	r	a	n	s
p	o	z	i	c
e	.	.	.	.



špaionfmsropocovízá intcíre

Jiným případem šifrovací metody je použití transpoziční mřížky, v níž jsou některá políčka vystřižena a do těchto je vpisován otevřený text. Po zaplnění všech políček je tabulka otočena o 90° a postup se opakuje. Tuto šifru použil Jules Verne ve své knize Matyáš Sandorf (Nový hrabě Monte Christo).



Steganograpfie

Zajímavým způsobem zabezpečení textu je jeho vnoření do jiného předmětu. Už v antickém Řecku se používaly k přenosu zpráv dřevěné destičky zalité voskem, do kterého bylo vyryto písmo. Jeden z prvních zaznamenaných případů steganografie popisuje vyrytí zprávy přímo do dřevěné destičky, která byla následně zalita voskem a vypadala jako nepopsaná. Jindy byla zpráva vytetována na oholenou hlavu otroka a následně se nechala zarůst vlasy. Přečíst ji bylo možné až po dalším oholení. Za II. světové války se používala například technika mikroteček. Šlo o malé nezřetelné tečky na pásku filmu, které teprve při prohlédnutí pod mikroskopem obsahovaly drobné písmo. V současnosti se steganografie stále uplatňuje, ale změnila podobu. Tajná zpráva může být zakódována na místo nepodstatného šumu v souborech se zvuky, obrázky, videem a podobně. V následujícím pravém obrázku je uložen kompletní text pěti divadelních her Williama Shakespeara, přičemž levý obrázek tuto informaci neobsahuje.

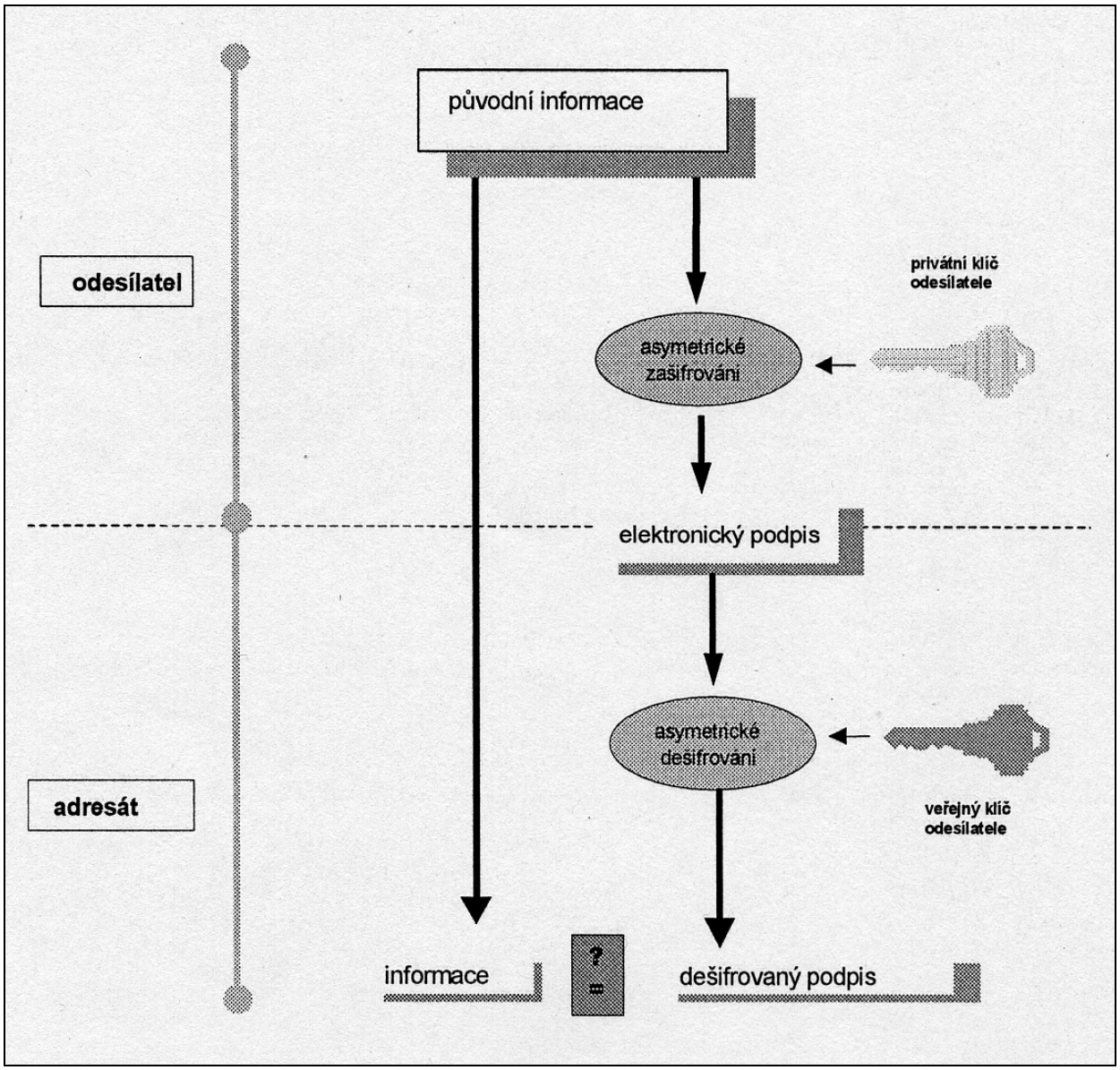
Kryptografické algoritmy

Algoritmy s privátní klíčem, v nichž se pro zašifrování i dešifrování zprávy používá stejný klíč. Těmto algoritmům se rovněž někdy říká šifry se symetrickým klíčem. Symetrické šifry jsou historicky nejstarší. Jejich hlavní nevýhoda spočívá v tom, že dojde-li k prozrazení klíče, pak nelze jednoznačně určit, došlo-li k úniku na straně autora nebo příjemce. Kryptografie s privátním klíčem se obvykle používá k ochraně informací uložených na disku počítače nebo k zašifrování dat při přenosu mezi dvěma systémy.

Algoritmy s veřejným klíčem, v nichž se pro zašifrování zprávy používá tzv. veřejný klíč, pro dešifrování zprávy slouží privátní klíč. Termín veřejný klíč vychází ze skutečnosti, že šifrovací klíč můžete klidně zveřejnit, aniž by se porušila bezpečnost zprávy nebo dešifrovacího klíče. Systémy s veřejným klíčem se někdy také označují jako kryptografie s asymetrickým klíčem. Kouzlo asymetrických šifer spočívá v tom, že to, co bylo zašifrováno jedním z klíčů privátní/veřejný, lze rozšifrovat právě a pouze druhým klíčem z dané dvojice. Tedy zašifruji-li něco svým klíčem privátním, rozšifruje to někdo pouze, má-li můj klíč veřejný. A obráceně: zašifruji-li něco vaším klíčem veřejným, pak to můžete rozšifrovat jen vaším klíčem privátním.

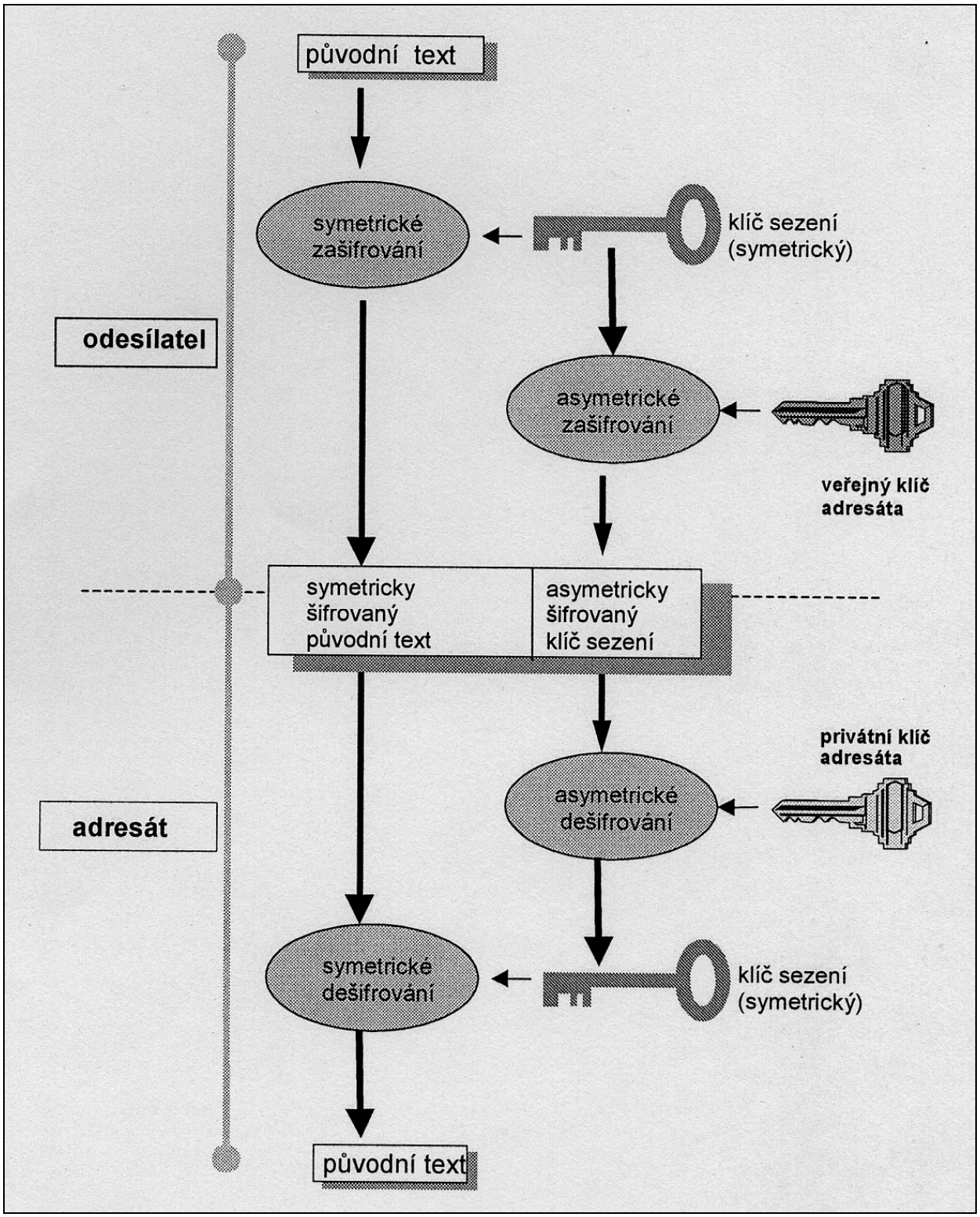
U asymetrické kryptografie je při prozrazení vždy jasné odkud byl klíč prozrazen. Privátní a veřejný klíč tvoří vždy nerozlučnou dvojici. Logicky patří totiž jeden k druhému. Dokonce je známo, že jeden lze vypočítat jednoznačně z hodnoty druhého.
Problém je ale v tom, že zatímco veřejný je možno z privátního vypočítat snadno a rychle, obráceně je to prakticky nemožné. Nikoliv ale proto, že by nebylo známo jak, ale proto, že výpočet by byl natolik náročný na kapacitu počítače, že výsledek by nebylo možné získat ani v horizontu let. Privátní klíč si každý musí chránit jako oko v hlavě, ale klíč veřejný by měl naopak zveřejnit. Zveřejnění je vhodné, aby každý mohl rozluštit soubory zašifrované privátním klíčem a zjistit tak, kdo je opravdu autorem.

Zašifrování dokumentu privátním klíčem a následné dešifrování klíčem veřejným za účelem ověření, kdo svým privátním klíčem dokument zašifroval, se nazývá metodou elektronického podpisu.



Kromě toho ještě existuje třetí typ šifrovacího systému, a to ***hybridní veřejné/privátní kryptosystémy***. V těchto systémech se pomalejší algoritmy s veřejným klíčem použijí k předání náhodně generovaného klíče sezení, který se pak použije jako základ pro algoritmy s privátním klíčem. Klíč se používá pouze pro jediné sezení a poté se zruší.

Výhodou takovéhoto způsobu je rychlost šifrování obsáhlých zpráv symetrickým algoritmem, při zachování bezpečnosti při distribuci asymetricky zašifrovaného klíče. Prakticky většina implementací systémů s veřejným klíčem jsou ve skutečnosti hybridní systémy.



Přehled systémů s privátním klíčem

DES, 3DES

Data Encryption Standard (DES) je šifrovací algoritmus, vyvinutý v 70. letech Národním úřadem pro standardy a technologie (dnes National Institute of Standards and Technology, NIST) a firmou IBM. DES používá klíč o délce 56 bitů. Bezpečnost algoritmu DES je možno vylepšit provedením více šifrování, takzvaným superšifrováním. Nejběžnější způsoby jsou dvojí šifrování (Double DES) a trojí šifrování (Triple DES). I když dvojitý DES zdánlivě výrazně zvyšuje bezpečnost, ukázaly se některé možnosti prolomení této metody a proto odborníci doporučují pro náročnější použití trojitý DES.

RC2, RC4

Blokové šifry původně vyvinuté Ronaldem Rivestem a uchovávané jako obchodní tajemství firmy RSA Data Security. Algoritmy byly v roce 1996 anonymně popsány na Usenetu a jsou rozumně bezpečné (i když některé klíče mohou být slabé). RC2 a RC4 se prodávají v implementacích s délkou klíče od 1 do 248 bitů.

RC5

Bloková šifra vyvinutá Ronaldem Rivestem, publikovaná v roce 1994. RC5 umožňuje uživateli definovat délku klíče, délku bloku dat a počet šifrovacích průchodů.

IDEA

International Data Encryption Algorithm (IDEA), vyvinutý Jamesem L. Massey a Xuejia Lai v Curychu, publikovaný v roce 1990. IDEA používá klíč o délce 128 bitů a je považován za poměrně silný. K šifrování souboru a elektronické pošty jej používá populární program PGP. Širšímu použití algoritmu IDEA bohužel brání řada softwarových patentů, které vlastní Ascom-Tech AG v Solothurnu (Švýcarsko).

Skipjack

Algoritmus s klasifikací Tajný, vyvinutý Národní bezpečnostní agenturou (NSA). Chcete-li vidět zdrojový kód algoritmu a jeho popis, musíte mít oprávnění pro nahlížení do přísně tajných materiálů. Používá klíč o délce 80 bitů.

Přehled systémů s veřejným klíčem

Diffie-Hellman

Systém pro výměnu kryptografických klíčů mezi dvěma stranami. Nejedná se vlastně o šifrovací algoritmus, ale o metodu pro vyvinutí a výměnu sdíleného privátního klíče přes veřejné komunikační kanály. V zásadě se obě strany dohodnou na nějaké společné číselné hodnotě a pak vytvoří klíč. Pak si strany vymění použité matematické transformace. Poté každá strana spočítá třetí klíč, který už není možno jednoduše odvodit ani při znalosti předchozí komunikace.
Existuje několik verzí tohoto protokolu pro různý počet komunikujících stran a s různými matematickými transformacemi. Je nutné pečlivě volit čísla a kalkulace, pro některé kombinace je totiž snadné algoritmus prolomit. Diffie-Hellmanův algoritmus se často používá jako základ při výměně kryptografických klíčů při zakódování komunikační linky. Podle implementace může mít klíč jakoukoliv délku. Obecně platí, že delší klíče jsou bezpečnější.

RSA

RSA je nejznámější kryptografický algoritmus s veřejným klíčem. Je pojmenován po svých objevitelích, Ronaldu Rivestovi, Adi Shamirovi a Leonardu Adlemanovi, kteří jej vymysleli v roce 1977. RSA je možno použít jednak jako šifrovací algoritmus a také jako základ pro systém digitálních podpisů. Podle použité implementace může mít klíč libovolnou délku.

ElGamal

Další algoritmus založený na exponenciální a modulární aritmetice. Podobně jako RSA algoritmus se dá použít k šifrování a digitálním podpisům.

DSA

Digital Signature Algorithm, vyvinutý v NSA a převzatý NISTem jako federální standard pro zpracování informací (FIPS). Přestože algoritmus DSA může používat klíče libovolné délky, podle FIPS je možno použít pouze klíče o délce 512 a 1024 bitů. Jak vyplývá z názvu, DSA slouží pouze pro digitální podpisy, dá se však upravit i pro potřeby šifrování. Tento algoritmus se občas označuje také zkratkou DSS, obdobně jako se algoritmus DEA označuje jako DES.

RSA a kryptografie s veřejným klíčem

Na rozdíl od algoritmů s privátním klíčem, používá RSA dva kryptografické klíče: veřejný klíč a privátní klíč. Veřejný klíč slouží k zašifrování zprávy, privátní klíč k jejímu dešifrování. Systém může fungovat i naopak, privátním klíčem se data šifrují, veřejným se dešifrují.

Síla algoritmu RSA je založena na obtížnosti faktorizace velmi velkých čísel. Následující popis neobsahuje matematické nuance algoritmu v plné šíři.

RSA je založen na vlastnostech modulární aritmetiky celých čísel. Jedním z použitých prvků je Eulerova funkce (n). Tato funkce je definována jako počet nesoudělných čísel, menších než n. Funkce pro prvočíslo je o jednu menší než toto prvočíslo: každé celé kladné číslo menší než naše prvočíslo je s ním nesoudělné.

Vlastnost využívaná algoritmem RSA byla objevena už Eulerem a říká: pro každé celé číslo i nesoudělné s n platí:

$$i^{\Phi(n)} \bmod n \equiv 1$$

Dále předpokládejme náhodná celá čísla e a d, která vyhovují následující kongruenci:

$$ed \equiv 1 \bmod \Phi(n)$$

Další využívanou vlastnost rovněž objevil Euler. Jeho teorém říká, že pro jakékoliv M nesoudělné s n platí:

$$(M^e)^d \bmod n \equiv M \; and \; (M^d)^e \bmod n \equiv M$$

Řečeno kryptografickým jazykem, bude-li M část zprávy, pak ji budeme jednoznačně šifrovat následující funkcí:

$$s \circ M^e \bmod n$$

Dešifrování se provede další funkcí:

$$M \circ s^d \bmod n$$

Jak nyní zvolit správné hodnoty pro n, e, d ? Nejprve pomocí nějaké vhodné metody zvolíme dvě velká prvočísla p a q o přibližně stejné velikosti. Tato čísla by měla být veliká – řádově stomístná – a je třeba je uchovávat v tajnosti.

Dále vypočítáme Eulerovu funkci $F(pq)$. Je-li číslo n součinem dvou prvočísel, pak platí:

$$(pq) = (p-1)(q-1) = F(n).$$

Dále zvolíme hodnotu e, která je nesoudělná s $F(n)$. Vhodná hodnota leží někde v intervalu $\max(p+1,q+1) < e < F(n)$. Pak vypočítáme d tak, aby platilo $ed \bmod F(n) \circ 1$. Jinými slovy hledáme modulo převrácenou hodnotu k $e \bmod F(n)$. Pokud by d vyšlo příliš malé (tedy menší než asi $\log_2(n)$), zvolíme jinou dvojici e a d.

Teď už tedy známe klíče. Při šifrování zprávy m postupujeme tak, že ji rozdělíme na stejná celá čísla M menší než n. Pro každou část zprávy hledáme hodnotu $M^e \bmod n = s$.

Tento výpočet je možno provést velmi rychle buď hardwarově, nebo softwarově s použitím speciálních algoritmů. Získané hodnoty uspořádáme do formátu zašifrované zprávy. Při dešifrování zprávu rozdělíme do bloků a každý blok dešifrujeme jako $\boldsymbol{s^d \bmod n = M}$.

Příklad algoritmu RSA

Předpokládejme například, že jsme zvolili prvočísla p a q takto:

p = 251
q = 269

Číslo n tedy bude n = 251 * 269 = 67519

Hodnota funkce je $\Phi(n) = (251 - 1)(269 - 1) = 67000$

Zvolíme si e = 50253, d potom bude $d = e^{-1} \bmod 67000 = 27917$

protože platí 50253 * 27917 = 1402913000 = 20939 * 67000 + 1 = 1 (mod 67000)

Pomocí čísla n = 67519 můžeme kódovat libovolnou zprávu M mezi 0 a 67518. Tímto systémem tedy můžeme kódovat textovou zprávu po dvojicích znaků. (Dva znaky mají 16 bitů, tedy 65536 kombinací.) Jako klíč použijeme hodnotu e a zakódujeme zprávu „RSA works!“. ASCII hodnoty zprávy „RSA works!“ a její zakódovanou podobu vidíte v následující tabulce:

ASCII	**Dekadická hodnota**	**Šifrovaná hodnota**
„RS“	21075	48467
„A “	16672	14579
„wo“	30575	26195
„rk“	29291	58004
„s!“	29473	30141

Jak vidíte, zakódované hodnoty nijak neodpovídají původní zprávě. Při dešifrování se každá hodnota umocní na d-tou a s výsledkem se provede operace mod n. Po převedení zpět do ASCII máme původní text.

Při použití algoritmu RSA ve skutečných aplikacích se pracuje s čísly, která jsou dlouhá řádově stovky míst. Protože početní operace se stoznakovými řetězci jsou časově velmi náročné, jsou moderní aplikace navrženy tak, aby minimalizovaly počet RSA operací, které bude třeba provést. Namísto použití RSA k šifrování celé zprávy se tímto algoritmem šifruje pouze klíč sezení, a teprve tímto se pak šifruje samotná zpráva pomocí nejakého rychlého algoritmu s privátním klíčem, jako jsou třeba DES nebo IDEA.

Síla algoritmu RSA

Čísla *n*, *e* a dokonce i *d* mohou být odhalena, aniž by se tím výrazně snížila bezpečnost šifry. Aby mohl útočník zprávu dekódovat, musel by zjistit hodnotu (*n*), což, podle všech současných vědomostí, vyžaduje faktorizovat číslo *n*.

Faktorizace velkých čísel je velmi obtížná – není dosud známa žádná metoda, kterou by bylo možno faktorizaci efektivně provést. Čas potřebný k faktorizaci může být, podle velikosti faktorizovaného čísla, třeba sto let nebo i několik miliard let, a to na těch nejrychlejších počítačích. Pokud je *n* dostatečně velké, je bez ohledu na vynaložené úsilí nefaktorizovatelné. Šifrovací algoritmus RSA je tedy poměrně velmi silný za předpokladu, že byly zvoleny vhodné hodnoty čísel *n*, *e* a *d*.

Abychom si ukázali, jak obtížná je faktorizace velkého čísla, zkusíme odhadnout, jak dlouho by trvala faktorizace dvousetmístného čísla – což je asi o 70 cifer delší, než doposud nejdelší faktorizovatelné číslo.

Všechna dvousetmístná čísla jsou reprezentovatelná na maximálně 655 bitech.

(2^x) má $(x \log_{10} 2) + 1$ číslic

Faktorizace čísla o délce 655 bitů si, s použitím nejrychlejšího známého algoritmu, vyžádá přibližně $1{,}2 \times 10^{23}$ operací. Předpokládejme nyní, že máme k dispozici počítač který je schopen provést 10 miliard operací za sekundu[10]. K provedení $1{,}2 \times 10^{23}$ operací pak budeme potřebovat $1{,}2 \times 10^{13}$ sekund, neboli 380,267 let počítačového času. Pokud se vám nezdá dost bezpečná šifra, kterou je možno rozluštit za 380 let, pak použijte číslo o čtyřech stech cifer – jeho faktorizace si vyžádá zhruba $8{,}6 \times 10^{15}$ let. To už by mělo být dost – podle Stručné historie času Stephena Hawkinga má samotný vesmír stáří jen asi 2×10^{10} let.

Ukažme si ještě jiný pohled na velikost takovýchto čísel. Předpokládejme, že se vám (nějak podaří vypočítat rozklady všech dvousetmístných dekadických čísel . Čistě k uložení samotných rozkladů budete potřebovat $(9 \times 10^{200}) \times 655$ bitů paměťového prostoru (bez jakékoliv režie nebo indexace). Tyto hodnoty budete ukládat na disk s kapacitou 100 GB (100×1024^4 nebo přibližně $1{,}1 \times 10^{14}$). Pak byste potřebovali $6{,}12 \times 10^{189}$ takovýchto disků. Dále uvažujme, že každý z těchto disků by vážil pouhou miliontinu gramu. Hmotnost celého potřebného diskového pole by byla $6{,}75 \times 10^{177}$ tun.

Planeta Země váží pouhých $6{,}588 \times 10^{21}$ tun. *Chandrasekharova mez*, hmotnost, která stačí hvězdě ke kolapsu do černé díry, je asi 1,5 násobek hmotnosti Slunce, tedy $3{,}29 \times 10^{27}$ tun. Takže vaše diskové pole by svou vlastní vahou spolehlivě zkolabovalo na černou díru.

Opakuji tedy, že pokud nedojde k nějakým zásadním objevům na poli teorie čísel, je algoritmus RSA prakticky stoprocentně odolný proti útoku hrubou silou.

Výtahy zpráv a digitální podpisy

Výtahy zpráv (message digest, nazývaný také kryptografický kontrolní součet nebo kryptografický hash-kód, cyptographic checksum, cryptographic hashcode) není nic jiného, než číslo – speciální číslo, vytvořené nějakou funkcí, která se jen velmi obtížně invertuje.

Digitální podpis (digital signature) je nejčastěji výtah zprávy zašifrovaný něčím privátní klíčem. Tomuto procesu se říká podepsání. Digitální podpis první dvě funkce, obě jsou pro bezpečnost systému důležité:

- ***integrita*** – digitální podpis indikuje, zda nedošlo k modifikaci souboru nebo zprávy
- ***autentizace*** – digitální podpis umožňuje matematicky ověřit, kdo zprávu podepsal

Kromě toho může plnit ještě třetí funkci, která může být někdy velmi důležitá – ***nepopiratelnost***. Nepopiratelnost znamená, že jakmile zprávu podepíšete a odešlete, nemůžete nikdy v budoucnu tvrdit, že nejste autorem této zprávy. Nemůžete svou zprávu zapřít, protože byla podepsána vaším privátním klíčem, o němž se předpokládá, že jej vlastníte pouze vy.

Výtah zprávy

Jednoduchá hashovací funkce dostane nějaký vstup, obvykle proměnné délky, a jejím výsledkem je číslo, výrazně menší než vstup. Funkce má charakter zobrazení „mnoha do jednoho“ – totiž větší množství vstupů (pravděpodobně nekonečné množství) produkuje stejný výstup. Funkce je kromě toho deterministická, což znamená, že stejná vstupní hodnota dává vždy stejný výstup. Hashovací funkce se velmi často používají v aplikacích, které potřebují rychle prohledávat velký objem dat – například u tabulek symbolů v překladačích nebo u spellcheckerů.

Výtah zprávy je též hashovací funkce. Má vstup libovolné délky – obvykle celý diskový soubor – a vytváří malou hodnotu (typicky o délce 128 nebo 512 bitů). Ze stejného vstupu vytváří vždy stejný výstup. Protože prostor výstupu je mnohem menší než prostor potencionálního vstupu, musí pro nejméně jednu výstupní hodnotu existovat více vstupních kombinací, které ji vyprodukují. U dobrého hashovacího algoritmu by to mělo platit pro všechny výstupní hodnoty. Kromě toho by měl dobrý algoritmus pro generování výtahu zprávy další dvě důležité podmínky. Algoritmus by neměl být snadno odvoditelný nebo invertovatelný. To znamená, že pokud máme výstupní hodnotu, neměli bychom být schopni jednoduše vymyslet vstupní text, který by takovýto výstup generoval ať už reverzací hashovací funkce nebo vypozorováním nějakých závislostí mezi vstupem a výstupem. Má-li výstup alespoň 128 bitů, celkem nepřichází v úvahu útok hrubou silou, protože bychom v průměru museli vyzkoušet $1,7 \times 10^{38}$ možných vstupů stejné délky, než bychom nalezli ten vstup, z něhož mohl vzniknout známý výstup. Když si to porovnáte s odhady v části „Síla algoritmu RSA“, zjistíte, že je to úloha, kterou se současnou technikou nemůže nikdo nikdy zvládnout. Při použití takto obrovských čísel je dokonce velmi nepravděpodobné, že by za celou historii lidstva mohly vzniknout dva různé dokumenty, které by produkovaly stejný 128 bitový výtah. Druhá důležitá podmínka dobrého algoritmu je ta, že malá změna ve vstupu bude mít za následek velké změny ve výstupu. Při změně jediného vstupního bitu by se měla změnit asi polovina výstupních bitů. Je to vlastně důsledek první podmínky, protože nechceme, aby se dala odvodit závislost chování vstupu a výstupu. Nicméně jedná se o vlastnost, která je důležitá i sama o sobě.

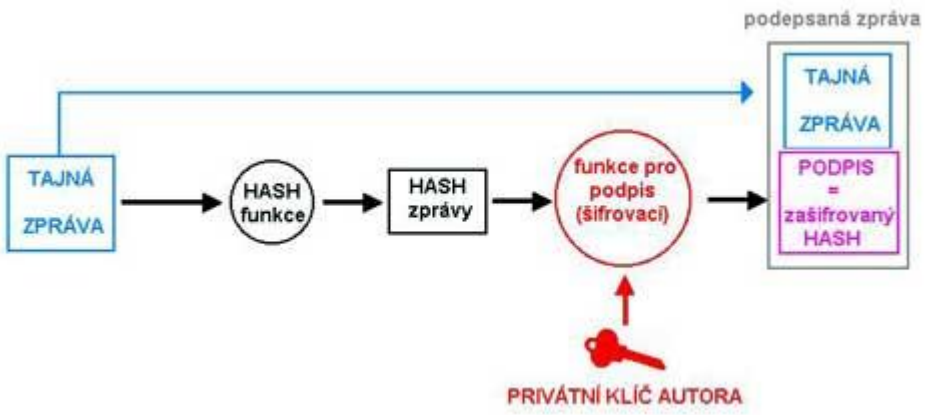
Digitální podpisy

Jak už jsme zjistili v předchozí části, funkce výtahu zprávy představuje pouze polovinu řešení spolehlivého digitálního podpisu. Druhá polovina spočívá v šifrování s veřejným klíčem – provozovaném ovšem opačným směrem. Připomeňme si, že když jsme dříve v této kapitole hovořili o kryptografii s veřejným klíčem, řekli jsme si, že je založena na dvou klíčích:

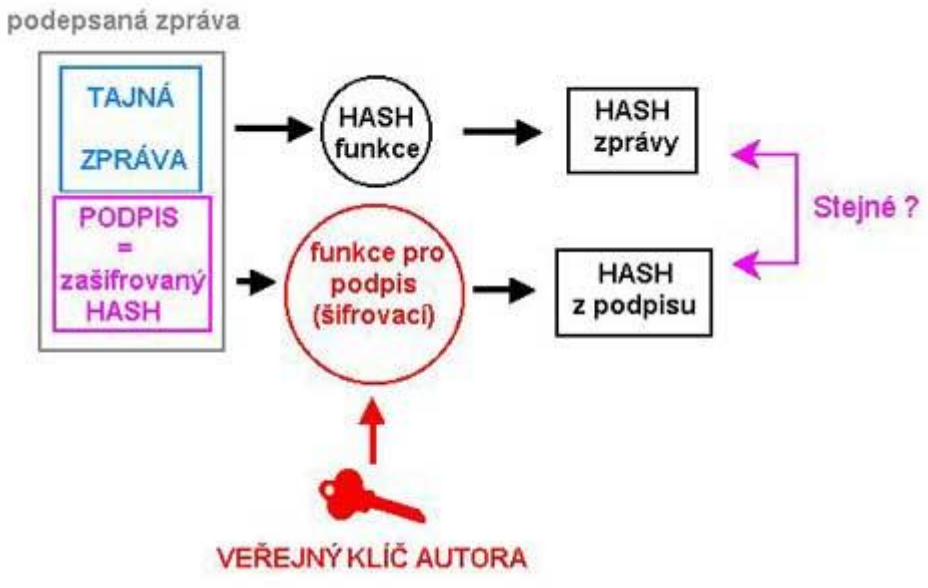
- ***veřejný klíč*** – používá se k zašifrování tajné zprávy. Obvykle je tento klíč široce znám,
- ***privátní klíč*** – je držen v tajnosti a slouží k dešifrování přijaté zprávy.

Pomocí trochy matematické gymnastiky se dá celý postup převrátit. Zprávu zašifrujete svým privátním klíčem a kdokoliv ji bude moci dešifrovat veřejným klíčem. Proč by to někdo dělal? Každému veřejnému klíči vyhovuje pouze jediný privátní klíč. Pokud je možno daným veřejným klíčem zprávu dešifrovat, s jistotou to znamená, že byla zašifrována správným privátním klíčem. No a to je princip činnosti digitálních podpisů.

Jakmile použijete na zprávu svůj privátní klíč, podepisujete ji. Když použijete privátní klíč a funkci výtahu zprávy, můžete vypočítat digitální podpis odesílané zprávy. Principiálně můžete použít algoritmus s veřejným klíčem bez algoritmu výtahu zprávy: je možno zašifrovat privátním klíčem celou zprávu. Každý známý algoritmus s veřejným klíčem však i u středně velkých zpráv trvá poměrně dlouho. Pokud byste tedy použili algoritmus s veřejným klíčem na soubor o velikosti několika megabajtů, mohlo by šifrování trvat také několik hodin či dní.



Namísto toho používáme rychlý algoritmus pro vytvoření výtahu zprávy a potom podepisujeme privátním klíčem tuto malou hodnotu. Když příjemce zašifrovanou hodnotu obdrží, může ji dešifrovat veřejným klíčem. Ze vstupního souboru se rovněž snadno vytvoří hashovaná hodnota. Pokud se obě hodnoty shodují, máte jistotu (téměř), že jste obdrželi stejnou zprávu, která byla odesílána.



V současné době se pro vytváření digitálních podpisů nejčastěji používají kombinace algoritmu pro výtah zprávy MD5 a kryptografického mechanismu s veřejným klíčem RSA. Další možnost spočívá v použití algoritmu SHA (Secure Hash Function) a ElGamalova mechanismu veřejného klíče – tyto algoritmy dohromady vytvářejí algoritmus DSA (Digital Signature Algorithm).

Algoritmy pro generování výtahu zprávy

V současnosti je k dispozici řada algoritmů pro generování výtahu zprávy. Všechny fungují v zásadě na stejném principu, liší se však v rychlosti a dalších vlastnostech.

MD2, MD4 a MD5

Jedním z nejrozšířenějších algoritmů pro generování výtahu zprávy je algoritmus MD5 Ronalda Rivesta, distribuovaný společností RSA Data Security, který je možno použít volně bez licenčních poplatků. Je založen na algoritmu MD4, který zase pro změnu vychází z algoritmu MD2. Všechny tyto tři algoritmy generují ze vstupního textu libovolné délky výtah o délce 128 bitů. Všechny algoritmy nejprve text rozšíří na fixní délku a poté provedou sérii matematických operací s celým vstupním blokem.

Algoritmus MD2 byl navržen Ronaldem Rivestem a publikován jako RFC 131913. Nemá žádné známé slabiny, je však velmi pomalý. Z toho důvodu vyvinul Rivest algoritmus MD4, který byl publikován jako RFC 1186 a 1320. Tento algoritmus byl navržen jako rychlý, kompaktní a optimalizovaný pro procesory s architekturou „little-endian“. V kryptografické literatuře se objevily možnosti potencionálních útoků na algoritmus MD4, a proto dr. Rivest vyvinul algoritmus MD5, publikovaný jako RFC 1321. Jedná se o přepracovaný algoritmus MD4, do kterého byla přidána navíc jedna série interních operací a bylo provedeno několik zásadních změn v algoritmu. Kvůli těmto změnám je MD5 o něco pomalejší než MD4. Je ovšem daleko šířeji používán, než algoritmus MD4. Na počátku roku 1996 byly odhaleny zásadní slabiny algoritmu MD4. Z toho důvodu by tento algoritmus neměl být používán.

SHA

Algoritmus Secure Hash Algorithm byl vyvinut NISTem ve spolupráci s NSA. Algoritmus vypadá jako blízce příbuzný algoritmu MD4, produkuje však výstup o délce 160 bitů, nikoliv 128 bitů. Analýza algoritmu ukazuje, že některé změny algoritmu MD4 plní podobnou funkci, jako vylepšení v algoritmu MD5 (i když se jedná o zcela jiné zásahy).

HAVAL

Algoritmus HAVAL je modifikací algoritmu MD5, vyvinuli jej Yuliang Zheng, Josef Pieprzyk a Jennifer Seberry. Dá se modifikovat tak, že vytváří výstup o délce od 92 do 256 bitů. Rovněž se dá volit počet „kol" (průchodů interním algoritmem). Výsledkem je, že HAVAL může být rychlejší než MD5, i když z toho plyne jisté snížení bezpečnosti výstupu. Na druhé straně může HAVAL vytvářet i delší a potencionálně bezpečnější hashovací kód.

Kontrolní součty

Kontrolní součet je funkce, která se vypočítává ze vstupu ke zjištění, zda nedošlo k porušení vstupu. Nejčastěji se kontrolní součty používají ke kontrole, zda data přenášená modemem nebo sítí nebyla poškozena nějakým prohozením bitů nebo šumem. Často se rovněž používají v diskových zařízeních pro kontrolu dat, zapisovaných a čtených z disku: pokud kontrolní součet dat nesouhlasí, vyskytl se zřejmě nějaký problém na disku nebo na pásce. Kontrolní součet se obvykle počítá jako jednoduchá lineární nebo polynomická funkce vstupu a výsledkem bývá malá hodnota (16 nebo 32 bitů). Často používanou formou kontrolních součtů jsou CRC – cyklické kontrolní redundantní součty. Kontrolní součty se snadno počítají a dají se rovněž snadno ošidit. Je možno modifikovat soubor tak, aby měl stejný kontrolní součet jako před modifikací. Řada „hackerských utilit", které kolují v hackerském podsvětí, obsahuje nástroje, jež obnovují hodnotu sum součtu systémových příkazů po jejich modifikaci. Z toho důvodu by se kontrolní součty nikdy neměly používat jako ochrana proti záměrné modifikaci.

Autentizační kódy zpráv

Autentizační kód zprávy (Message Athentication Code, MAC) je v zásadě funkce výtahu zprávy doplněná o heslo. Smyslem je, aby hodnota MAC kódu nebyla obnovitelná osobou, která má k dispozici sice stejný vstup, nezná však příslušný tajný klíč (heslo). Tento algoritmus může a nemusí být bezpečnější než prostá funkce výtahu zprávy – záleží na použitém algoritmu, síle klíče a délce MAC kódu.

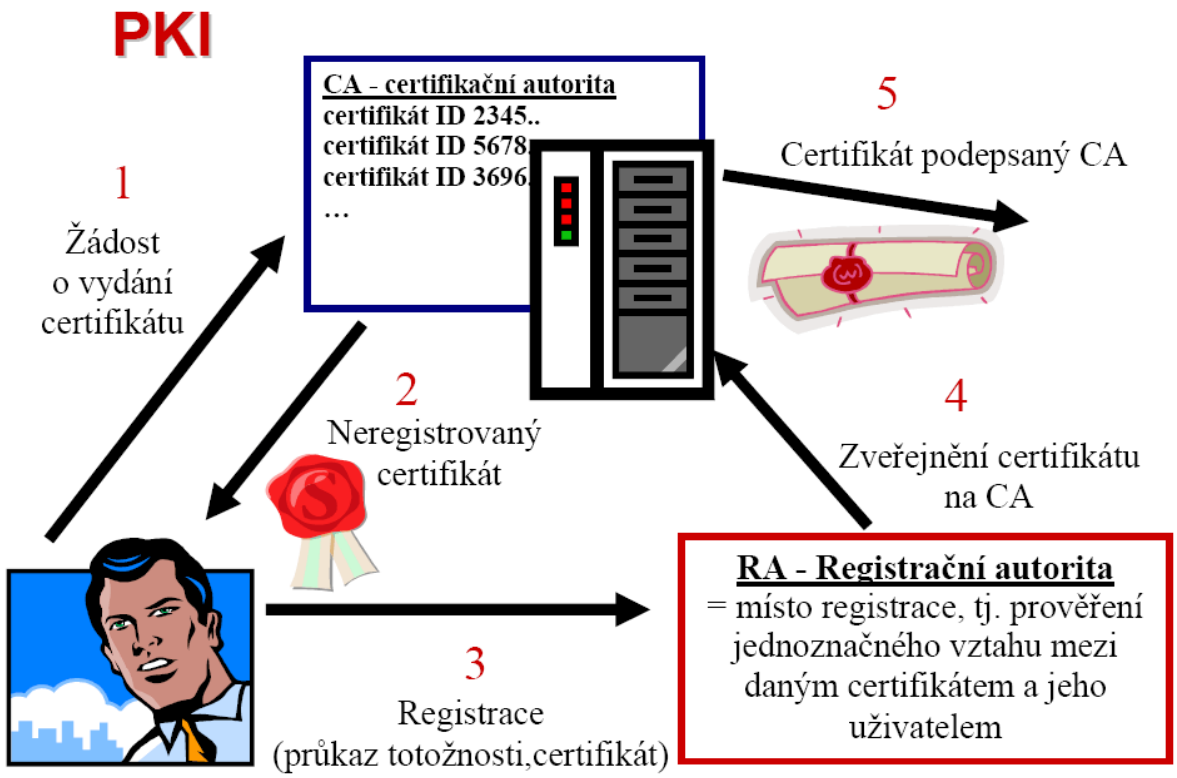
Jednoduchá metoda MAC algoritmu přidá ke zprávě klíč a poté generuje výtah zprávy. Protože klíč je součástí vstupu, ovlivňuje hodnotu výsledného kódu neobnovitelným způsobem. Protože dva různé klíče generují pro stejná data velmi rozdílné výstupy, plní tato metoda úspěšně funkci heslem řízeného výtahu zprávy.

Druhá používaná metoda vychází z nějaké proudové šifry, jako je třeba RC4 nebo DES. Klíč je v tomto případě šifrovací heslo a MAC kód je poslední blok bitů šifrovacího algoritmu. Protože výstup šifry závisí na všech bitech vstupu a na hesle, bude poslední blok výstupu různý pro různé vstupy i pro různá hesla. Pokud je ale velikost šifrovacího bloku malá (například 64 bitů), může být MAC kód snáze rozluštitelný hrubou silou než podstatně delší výtahy zpráv.

Digitální podpisy s veřejným klíčem mohou být také chápány jako určitá forma MAC kódu, protože závisejí jednak na výtahu zprávy a jednak na tajném klíči. Změna kterékoliv hodnoty má vliv na celkový výsledek funkce.

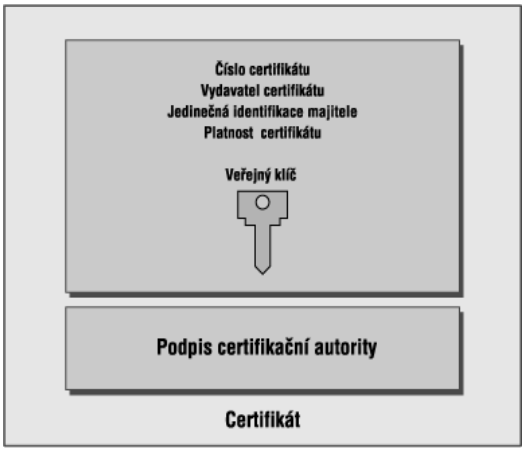
Certifikáty

Problémem asymetrické kryptografie je způsob, jak ověřit pravost zveřejněných veřejných klíčů. K tomu slouží digitální či elektronický certifikát. Digitální certifikát je elektronická obdoba cestovního pasu nebo občanského průkazu. Jedná se v podstatě o uživatelův veřejný klíč plus další údaje popisující držitele certifikátu (jméno, bydliště, fotografie apod.) To vše je zašifrováno (elektronicky podepsáno) privátním klíčem, jehož veřejný klíč je znám a dostupný z nezaměnitelných zdrojů. Pomocí digitálního certifikátu pak lze ochránit nejen elektronickou poštu, ale zajistit bezpečnou komunikaci např. i po Internetu.
Držitelem a vydavatelem privátního klíče je tzv. certifikační autorita (v ČR např. I. CA), tedy instituce nebo útvar, který tyto certifikáty neboli elektronické občanské průkazy vydává. Každý může požádat certifikační autoritu o digitální certifikát.



Certifikační autorita a certifikáty

Řešením problému správy, distribuce a uchování klíčů je využití služeb certifikační autority (tzv. PKI - Public Key Infrastructure, neboli Infrastruktura veřejného klíče). Instituce CA se podobají státním notářům, kteří při vzájemné komunikaci dvou subjektů vystupují jako třetí nezávislý důvěryhodný subjekt. Prostřednictvím jím vydaného certifikátu jednoznačně svazuje identifikaci subjektu s jeho dvojicí klíčů respektive s jeho digitálním podpisem. Certifikáty obsahují ve své nejjednodušší formě veřejný klíč, jméno a další údaje zajišťující nezaměnitelnost subjektů. Běžně používané certifikáty též obsahují datum počátku platnosti, datum ukončení platnosti, jméno certifikační autority, která certifikát vydala, sériové číslo a některé další informace. Certifikační autorita garantuje jedinečnost subjektů podle užité identifikace subjektu. To je zajištěno legislativními a technickými pravidly provozu instituce certifikační autority. Splnění těchto požadavků potvrdí certifikační autorita podepsáním dokumentu svým privátním klíčem a následným vydáním tohoto certifikátu.

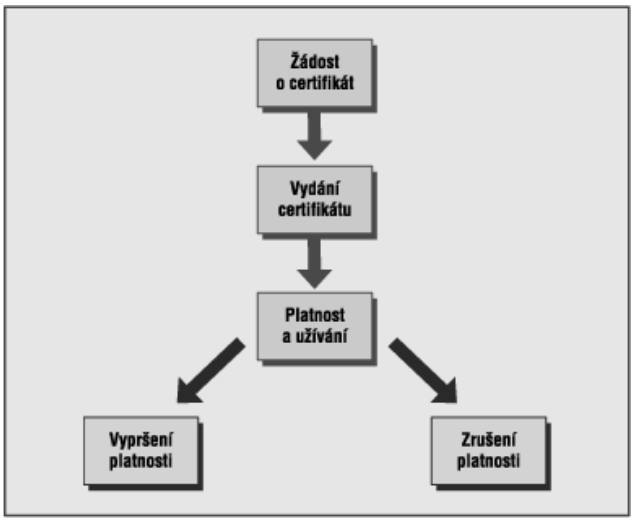


Znamená to, že certifikát je podepsaným dokumentem se všemi důsledky z toho plynoucími, tedy zejména autorizace (certifikační autorita jako garant pravosti dokumentu) a integrita dat (nelze zaměnit klíč nebo identitu klienta). Tím, že certifikační autorita zaručuje správnost jí vydaného certifikátu, odstraňuje nutnost smluvní důvěryhodné výměny klíčů mezi dvěma subjekty navzájem a jejich dohoda spočívá pouze v domluvě o společně uznávané certifikační autoritě. Důležité je, že se utajovaná data na straně klienta redukují pouze na bezpečné uchovávání privátního klíče, protože ostatní je řešeno certifikáty. Ty si můžeme kdykoliv ověřit se znalostí veřejného klíče certifikační autority, repektive jejího certifikátu. Existence certifikační autority také umožňuje důvěryhodnou komunikaci i subjektů, jenž se navzájem fyzicky nikdy nepotkali nebo neabsolvovali složitou proceduru vzájemné důvěryhodné výměny svých klíčů.

Tvorba a životnost certifikátů

Tvorba certifikátu má 6 kroků:

1. Generování klíčů. Každý potenciální žadatel o certifikát si nejprve sám pomocí dostupného SW vybavení vygeneruje dvojici klíčů pro použití v asymetrické kryptografii.
2. Příprava identifikačních dat. Žadatel o certifikát shromáždí podle požadavků certifikační autority osobní identifikační materiály nutné pro vydání certifikátu, jako IČO, DIČO, resp. číslo OP, rodné číslo a podobně.
3. Předání veřejných klíčů a identifikačních údajů certifikační autoritě. Žadatel předá certifikační autoritě data nutná pro vydání certifikátu spolu s doklady o jejich pravosti.
4. Ověření informací. Certifikační autorita si na příslušných místech ověří, že může vydat žadateli certifikát.
5. Tvorba certifikátu. Certifikační autorita vytvoří digitální dokument příslušného formátu a ten poté podepíše svým privátním klíčem.
6. Předání certifikátu. Podle dohody je certifikát žadateli předán (disketa), zaslán, nebo zveřejněn.



Doba platnosti certifikátů je omezená a je uvedena v každém certifikátu. Tato veličina je velmi důležitá. Pokrok ve zvyšování výkonnosti výpočetní techniky a možnost objevení mezer v protokolech nebo algoritmech by ve velkém časovém horizontu mohl způsobit, že by se certifikáty staly nespolehlivé. Běžné certifikáty jsou proto vydávány s platností 6 měsíců, nejvíce 1 rok. I během této doby je možné zrušit platnost certifikátu. Důvodem pro toto opatření může být například vyzrazení privátního klíče.

Klíče a certifikáty - kam s nimi?

Digitální podpis je ekvivalentem podpisu klasického. Ten je jako doklad totožnosti využíván velmi hojně. Vzhledem k jednoduché napodobitelnosti, je však velmi vysoké riziko jeho zneužití. Výhodou digitálního podpisu je téměř nulová možnost jeho zneužití. Riziko je zejména v nedodržovaní bezpečné manipulace s ním. S digitálním podpisem, přesněji řečeno se soukromým klíčem (část digit. podpisu) je třeba zacházet velmi obezřetně, nejlépe jako s PINem platební karty či mobilního telefonu. Důležitým prvkem, který brání zneužití digitálního podpisu, je uchovávání soukromého klíče na bezpečném místě.

První možností uložení klíčů je využít čistě softwarové řešení a svoje klíče skladovat přímo na pevném disku počítače v bezpečně zašifrovaném tvaru. Takto uložený klíč je odšifrován až bezprostředně před použitím na základě zadání správného hesla. Tato operace je z hlediska bezpečnosti kritická a klade velké nároky na zabezpečení vlastního počítače.

Dalším (nepříliš používaným) řešením mohou být různé hardwarové šifrovací a zabezpečovací doplňky k PC, které se většinou montují do volného PC slotu. Tento hardware může odstranit část rizik softwarových řešení (např. bezpečná paměť pro uložení klíčů apod.). I v tomto případě však zůstává potřeba zabezpečit počítač proti neautorizovanému zásahu (například záměna karty za jinou, upravenou ve prospěch narušitele).

Asi nejlepším řešením je použít hardwarové zařízení umístěné mimo vlastní počítač. Typickým představitelem jsou čipové karty, jejichž mechanické provedení se může lišit. Ač se to na první pohled nezdá, čipové karty dnes najdeme takřka všude: telefonní karty, SIM-karty do mobilních telefonů, různé průkazkové karty sloužící k identifikaci jejich nositele, elektronická peněženka, atp. Méně často už potkáme karty použité jako zabezpečovací.

„Čistokrevné“ čipové karty bývají vybaveny buď kontaktní ploškou (s osmi kontakty) nebo mohou být bezkontaktní (s VF-přenosem dat). Ať již je „balení“ jakékoliv, čipová karta se většinou skládá z těchto základních částí:

- paměť (ROM s firmwarem, RAM pro ukládání mezivýsledků, PROM pro nastavení některých nevratných parametrů karty, EEPROM pro další rozšiřující funkce karty)
- procesor nebo řadič
- komunikační zařízení (kontakty/VF)
- případně CryptoProcesor(včetně generátoru náhodných čísel)

Pokud je čipová karta používána pouze jako paměťové médium, pak v zásadě nemusí být vybavena procesorem, ale pro přístup do její paměti postačí pouze řadič. Z tohoto pohledu rozlišujeme karty na „procesorové“ a ostatní. Existence procesoru ale ještě nezaručuje podporu zabezpečovacích funkcí, kterých může být na kartě implementováno bezpočet. Ne všechny je ale optimální přímo na kartě využívat. Pro vlastní šifrování rozsáhlých dat je daleko efektivnější použít výkonný procesor počítače.

Karta s kryptografickým procesorem se hodí hlavně pro realizaci podpisu zde uloženým soukromým klíčem nebo k získání symetrického klíče použitého při hybridním šifrování rozsáhlého datového bloku. Důvodem je potřeba zajistit, aby soukromý klíč nemusel hardwarové zařízení nikdy opustit a dostat se do samotného počítače, kde by mohl být zneužit. Přístup ke klíčům na čipové kartě bývá většinou chráněn PINem (PersonalIdentificationNumber). Toto není zbytečné obtěžování uživatele, ale rozumné opatření pro případ, že by se karta dostala mimo jeho přímou kontrolu.

Některé karty mohou podporovat „administrátorský PIN“, který se obvykle označuje jako PUK (PrivilegedUserKey). Většina karet umožňuje zadat PIN o délce až 16 znaků. Karta je zajištěna i pro případ, že záškodník bude zkoušet zadávat různé hodnoty PINu s cílem najít správnou kombinaci. Tímto zajištěním je omezení počtu neúspěšných zadání PINu, po jehož překročení se karta uzamkne.

Další postup může být u jednotlivých karet různý:

- zadáním PUK se karta odblokuje a data na ní zůstanou
- zadáním PUK se karta odblokuje a zadá se nový PIN, přičemž klíče a data na kartě lze dále používat
- zadáním PUK kartu vymažeme a znovu naformátujeme
- zadáním PUK je karta nevratně zničena.

Uživatel má zpravidla k dispozici software, kterým může PIN na kartě měnit. Tuto možnost by měl pravidelně využívat, aby omezil nebezpečí jeho prozrazení. Zadávání PINu na klávesnici počítače je slabým místem, protože zde existuje reálné riziko, že případný záškodník dokáže nainstalovat například trojského koně. Tento problém řeší speciální klávesnice (PIN-Pad), která může být integrována například do čtečky. PIN potom neprochází zranitelným prostředím počítače.

Karta po svém vyrobení absolvuje několik „životních fází“. Nová karta obsahuje firmware, který většinou ještě nemá zadány všechny volitelné parametry a definovány vlastnosti. Podle požadavků zákazníka mohou být karty donastaveny (fáze „customizace“). Pro celou objednanou sérii může být nastaven PUK, minimální délka PINu, počet možných neúspěšných zadání apod. U uživatele pak následuje fáze inicializace (formátování) karty Může zadat např. jméno karty (label), uživatelský PIN apod. Tento krok lze většinou provést i u karty, která není nová, ale byla již používána. Následuje užívání karty.

Druhým typickým představitelem hardwarových zařízení jsou tzv. USB tokeny. Token iKey je identifikační předmět umožňující bezpečnou autentizaci uživatelů. Je určen pro zabezpečení přístupu do počítačových sítí, VPN, intranetu nebo extranetu, pro uložení digitálních certifikátů, pro nasazení v aplikacích e-business, digitálního podpisu a PKI. Každý uživatel je vybaven pouze svým iKey - můžete si jej přidat na svůj svazek s běžnými klíči (klíč máte stále při sobě). iKey podporuje, obdobně jako smart karty, zabezpečení pomocí PIN kódu, navíc ale nabízí kryptografické funkce jako - hardwarovou podporu algoritmů MD5 a RSA, včetně možnosti vygenerování privátního klíče přímo v tokenu.

iKey na portu USB je jednoduchým, okamžitě použitelným bezpečnostním řešením na každém dnešním počítači bez nutnosti investice do speciálního čtecího zařízení.

Standardní formáty souborů certifikátů

Do svého prohlížeče můžete importovat a exportovat certifikáty v následujících formátech:

- **Personal Information Exchange (PKCS #12)**

  Personal Information Exchange formát (PFX, také nazývaný PKCS #12) umožňuje přenos certifikátu a jeho odpovídajícího privátního klíče z jednoho počítače na jiný nebo z počítače na vyjímatelné médium. PKCS #12 (Public Key Cryptography Standard #12) je průmyslový standard, který se hodí pro přenos nebo zálohu a obnovení certifikátu a s ním asociovaného privátního klíče. Přenos může být proveden mezi produkty od různých dodavatelů.

- **Cryptographic Message Syntax Standard (PKCS #7)**

  PKCS #7 umožňuje přenos certifikátu a všech certifikátů v jeho certifikační cestě z jednoho počítače na druhý nebo z počítače na vyjímatelný disk. Soubory PKCS #7 typicky používají příponu .p7b a jsou kompatibilní se standardem ITU-T X.509.

- **DER – binárně kódované X.509**

  DER (Distinguished Encoding Rules)pro ASN.1, jak je definováno v ITU-T doporučení X.509, je více omezující kódovací standard než alternativní BER (Basic Encoding Rules) pro ASN.1, jak je definováno v ITU-T doporučení X.209, na kterém je DER založen. Oba BER i DER poskytují metodu kódování objektů (jako jsou certifikáty a zprávy) pro přenos mezi zařízeními a aplikacemi nezávislou na platformě. Při kódování certifikátu většina aplikací užívá DER, protože část certifikátu (info žádosti o certifikát) musí být DER-kódována, aby mohla být podepsána. Tento formát může být užíván certifikačními autoritami, které nejsou na serverech s Windows 2000 pro svoji nezávislost na platformě. Soubory s certifikáty DER mají typicky příponu .der.

- **Base64 kódované X.509**

  Toto je metoda kódování vyvinutá pro užití s S/MIME (Secure/Multipurpose Internet Mail Extensions), což je populární a standardní metoda pro přenos binárních příloh přes internet. Base64 kóduje soubory v ASCII textovém formátu, který snižuje pravděpodobnost znehodnocení souborů, které jsou posílány přes internetové brány, zatímco S/MIME poskytuje některé kryptografické zabezpečovací služby pro aplikace elektronické pošty včetně nepopiratelnosti původu užitím digitálního podpisu, důvěrnosti a bezpečnosti dat užitím šifrování, autentizace a integrity zprávy. MIME specifikace (RFC 1341 a následující) definuje mechanismus pro kódování jakýchkoliv binárních informací pro přenos elektronickou poštou. Protože všechny klientské aplikace podporující MIME umí dekódovat Base64 soubory, tento formát může být užíván certifikačními autoritami, které nejsou na serverech s Windows 2000 pro svoji nezávislost na platformě. Soubory s certifikáty Base64 mají typicky příponu .cer.

Snaha o co nebezpečnější informační systém

Nutnost zavést nějakou formu ochrany dat informačních systémů, které jsou v současnosti provozovány často veřejně v síti Internet, si postupně uvědomuje každý uživatel. Pokud se jedná o ochranu jednoho počítače, je možné to zajistit programem pro šifrování dat na disku a pro spolehlivé ověření uživatelů, kteří se k počítači přihlašují.

První problémy se ale už objevují v případě, kdy se má chránit několik počítačů v malé počítačové síti. To by bylo ještě možné vyřešit za použití nějakého na zakázku vytvořeného programu. I v ČR bylo několik takových výrobců. Jak ale postupovat v případě, že se má řešit bezpečnost dat v prostředí, které se nepřetržitě mění? Zajistit bezpečnost dat v lokální síti s několika tisíci počítači nebo v síti Internet, kde se mnou chtějí komunikovat stále noví a noví obchodní partneři, může být téměř neřešitelný problém.

Požadavek na zabezpečení moderních informačních technologií je naprosto pochopitelný, neboť například elektronická pošta je v současné podobě vlastně elektronickou podobou pohlednice či korespondenčního lístku. To znamená, že každý zvědavý listonoš si může pohlednici přečíst. Byrokrat může opět namítnout, že čtení pohlednic i elektronické pošty zakazuje zákon. Co je to platné, když podezření, že vaši elektronickou poštu čte konkurence, zjistíte, až když ztratíte několik zajímavých zakázek.

Autentizace

Autentizace je proces, při kterém dochází ke kontrole totožnosti uživatele. Existují tři možné způsoby, jak se můžeme počítačovému systému autentizovat, a při každé autentizaci jeden nebo více z nich používáme:

- můžeme počítači říct něco, co víme (například heslo).
- můžeme počítači „ukázat“ něco co máme (například identifikační kartu).
- mecháme počítač, aby si něco našeho zkontroloval (například otisk prstu).

Žádný z těchto systémů není dokonalý. Například tajným odposlechem vaší terminálové linky může někdo získat vaše heslo. Pokud vás přepadnou, mohou vám sebrat vaši identifikační kartu. A pokud bude mít útočník nůž, můžete klidně přijít o prst! Obecně platí, že čím je autentizační metoda spolehlivější, tím obtížněji se používá a narušitel musí být při jejím překonání více agresivní.

Základní autentizace – uživatelské jméno + heslo

Každý uživatel systému musí mít účet. Účet je identifikován uživatelským jménem. Obvykle má každý účet rovněž tajné heslo, které chrání před neoprávněným přístupem. Uživatelským jménům se občas také říká jména účtů. Abysme se mohli k systému přihlásit musíme znát jak uživatelské jméno, tak i své heslo. Hesla představují nejjednodušší způsob autentizace: je to tajemství, které spolu sdílí uživatel a počítač. Když se přihlašujeme, zadáváme heslo, čímž počítač přesvědčíme, že jsme opravdu tím, za koho se vydáváme. Počítač zkontroluje, zda námi zadané heslo odpovídá zadanému účtu. Pokud se shodují, můžeme začít pracovat v systému. Když heslo zapisujeme, systém jej nezobrazuje. Tím jsme chráněni pro případ, že používáme tiskárnový terminál nebo, že nám někdo hledí přes rameno.

Hesla obvykle představují první obranou linii systému před těmi, kteří se chtějí do něj vloupat. I když je možné se do systému dostat nebo ukrást nějaké informace po síti i bez znalosti hesla, řada průniků do počítačů uspěla zejména kvůli špatně zvoleným nebo špatně chráněným heslům.

Volba hesla

Přestože hesla představují základní prvek počítačové bezpečnosti, uživatelé o jejich volbě dostanou často jen velmi základní instrukce.

Špatné heslo je každé, které je možno snadno uhodnout. Nejprimitivnější metoda pro uhodnutí hesla je vyzkoušení všech možností. Když bychom vytvořili program, který by zkoušel všechny šestipísmenné kombinace počínaje od AAAAAA a konče u ZZZZZZ, museli bychom vyzkoušet 308 915 776 různých hesel. Pokud bychom zkoušeli jedno heslo za sekundu, potřebovali bychom k tomu skoro deset let. Skuteční narušitelé postupují daleko chytřeji. Namísto ručního zadávání hesel se svým počítačem připojí telefonicky nebo přes síť a zkoušejí hesla. Když je vzdálený systém odpojí, připojí se znovu a zkoušejí dále. Namísto, aby zkoušeli všechny možné kombinace písmen počínaje od AAAAAA (nebo čehokoliv jiného), zkoušejí pouze obecně často používaná hesla. I nepříliš výkonný domácí počítač s dobrým programem pro hádání hesel může vyzkoušet tisíce různých hesel za necelý den. Některé seznamy hesel, které různí útočníci používají, obsahují i několik set tisíc slov. Takže jakékoliv heslo, které by mohl používat kdokoliv jiný, je pravděpodobně heslo špatné.
Dobrá hesla jsou takové hesla, která se dají jen špatně uhodnout. Nejlepší hesla se hádají špatně z následujících důvodů:

- používají malá i velká písmena
- kromě písmen obsahují i číslice a/nebo interpunkční znaky
- mohou obsahovat nějaké řídící znaky a/nebo mezery
- snadno se pamatují, takže je není potřeba zapisovat
- jsou dlouhá sedm nebo osm znaků
- dají se rychle napsat, takže je nikdo nemůže zjistit tím, že by vám hleděl přes rameno

Hesla na více počítačích

Pokud máme účty na několika počítačích, možná budeme chtít používat na všech stejné heslo, abychom si jich nemuseli pamatovat tolik. Pokud však máme na více počítačích stejné heslo a dojde k průniku do jednoho z nich, jsou potencionálně narušeny naše účty na všech počítačích. Jedna obvyklá metoda, kterou volí uživatelé s účty na více počítačích, spočívá v tom, že mají základní heslo, které se na jednotlivých počítačích lehce modifikuje. Můžeme mít například základní heslo kxyzzy následované prvním písmenem jména počítače. Takže na počítači athena budeme mít heslo kxyzzya, a na počítači ems heslo ksyzzye. Pochopitelně, že nebudeme pro modifikaci hesel používat přesně tuto metodu.

Jednorázová hesla

Nejúčinnější způsob jak minimalizovat riziko plynoucí z použití špatného hesla je úplně vyloučit požití konvenčních hesel. Namísto toho bychom mohli nainstalovat software a/nebo hardware, který umožní práci s jednorázovými hesly. Jednorázové heslo je přesně to, co napovídá jeho název – heslo určené pro jediné použití.

Jako uživatel můžeme dostat vytištěný seznam hesel. Vždy, když heslo použijeme, na seznamu jej škrtneme a při příštím přihlášení použijeme následující heslo ze seznamu. Nebo můžeme dostat malou kartu, kterou budeme nosit s sebou; na kartě se bude každou minutu objevovat jiné číslo. Nebo dostaneme malou kalkulačku. Když se budeme přihlašovat, počítač nám zobrazí nějaké číslo.

Toto číslo napíšeme na kalkulačce, přidáme k němu naše osobní číslo a zobrazený výsledek pak použijeme jako heslo.

Každý z těchto systémů výrazně zvyšuje bezpečnost celého systému. Protože však vyžadují instalaci nějakého speciálního softwaru nebo pořízení speciálního hardwaru, nejsou tyto systémy v současnosti příliš rozšířené.

Shrnutí

Hesla představují jednu ze základních forem autentizace. Uživatel se musí prokázat správným heslem při zřizování spojení z důvodu autorizace přístupu do systému, aplikací nebo k informacím. Bezpečnostní systémy, které jsou závislé na heslech vyžadují, aby hesla byla udržována vždy v tajnosti. Avšak tato ochrana bývá často napadena krádežemi a následným zneužitím (zvláště v internetovém prostředí).

Autentizace uživatele v internetovém prostředí

World Wide Web představuje jedno z nejzajímavějších využití internetu. Zároveň však přináší i potencionální problémy s bezpečností. Autentizace uživatele ve webovém prostředí je stále nejčastěji založena na kombinaci uživatelského jména a hesla. Při přenosu těchto důvěrných informací mezi Web serverem a prohlížečem je však možné tyto informace zachytit.

Vyloučení možnosti odposlechu

Riziko odposlechu se týká všech internetových protokolů, je však významné zejména u služeb WWW, kdy se mohou přenášet důvěrné dokumenty a další údaje, například čísla kreditních karet. Existují pouze dva způsoby, jak informace před odposlechem chránit. První možnost je přenášet je pouze po fyzicky bezpečných linkách (což Internet nesplňuje). Druhá možnost je zašifrovat informace tak, aby je mohl dešifrovat pouze autorizovaný příjemce. Jinou možností odposlechu je analýza provozu. Při tomto odposlechu se útočník dozví o transakcích, které sledovaný objekt provádí, aniž by však znal jejich obsah. Tento typ útoků bývá nejčastěji namířen na logovací soubory Web serverů.

Analýza využití digitálního podpisu pro autentizaci přístupu

Ve webovém prostředí se objevuje riziko odposlechu citlivých informací. Tedy útočník může uživatelské jméno a heslo odposlouchat. Musíme se tedy zamyslet nad silnějším zabezpečením přihlašování.

Jak již bylo řečeno v úvodu této kapitoly, pro ověření totožnosti uživatele je možno použít něco, co víme (heslo) nebo něco, co vlastníme (identifikační kartu). Majitel digitálního podpisu má také něco, co vlastní pouze on – privátní klíč. Smyslem této podkapitoly je zjistit možnost získání potřebných údajů k autentizaci digitálním podpisem. Cílem je zvýšení bezpečnosti přístupu k citlivým informacím. To znamená, že nebude stačit pouhé přihlašování uživatelským jménem a heslem, ale bude vyžadován i jeho digitální certifikát.

Digitálním podpisem lze podepsat vše od e-mailů až po databázové položky. To znamená, že by mělo jít podepsat i uživatelské jméno a heslo při přihlašování. Pro získání podpisu by uživatel použil

nějaký externí program, do kterého by zadal své uživatelské jméno, heslo a cestu k souboru se svým privátním klíčem. Program by potom vypočítal digitální podpis přihlašovacích položek, který by uživatel schránkou zkopíroval do pole pro digitální podpis v prohlížeči. Toto řešení by určitým způsobem zvýšilo bezpečnost přihlašování. Ovšem pokud útočník odposlouchává komunikaci mezi klientem a serverem, získá uživatelské jméno, heslo a z nich vypočítaný digitální podpis. Pokud se tedy potom bude chtít do systému dostat, jednoduše tyto položky při přihlášení použije.

Předchozí případ se tedy musí vylepšit o určitou unikátní informaci při každém přihlášení. Server tedy před přihlášením musí vygenerovat nějakou náhodnou sekvenci, která se navíc zadá do programu pro vypočítání digitálního podpisu. Tedy podpis bude při každém přihlášení unikátní. To znamená, že i kdyby útočník odposlouchal uživatelské jméno, heslo i podpis, není mu to nic platné, protože při novém přihlášení server vygeneruje jinou náhodnou sekvenci. Bez znalosti privátního klíče se útočník do systému nedostane. Tato varianta přihlašování už kritéria bezpečnosti splňuje. Ovšem nyní si ji musíme rozebrat z pohledu řadového uživatele. Ten musí udržovat soubor s privátním klíčem v tajnosti. Dále musí vygenerovat podpis v externí aplikaci, do které musí předtím zadat svoje uživatelské jméno, heslo a serverem vygenerovanou unikátní sekvenci. Nakonec musí vygenerovaný podpis přenést schránkou do prohlížeče. Takto složitou proceduru přihlašování by však málokterý uživatel uvítal!

Dalším možným řešením daného problému je využití služeb vrstvy SSL (Secure Societ Layer), která je z části založena na předchozí úvaze. Hlavní roli při ověřování identity zde hraje digitální podpis. Řešení užitím SSL přináší navíc další výhody.

Protokol SSL je pro některé typy dokumentů ještě nedostatečný, proto byl vyvinut protokol SET – Secure Electronic Transaction. Je to otevřený protokol původně navržený pro bezpečnost plateb bankovní kartou přes Internet. Je schopen zaručit Integritu dat, ověřit elektronickou totožnost jednotlivých entit i zašifrovat dokumenty.

Secure Socket Layer (SSL)

SSL technologie umožňuje dvě funkce: šifruje informační tok mezi klientem a serverem a vytváří základ pro vzájemnou klient/server autentizaci. SSL bylo vyvinuto firmou Netscape Communication v roce 1994. Je podporováno populárními klientskými aplikacemi (Netscape Navigator, Microsoft Internet Explorer), většinou serverových aplikací (Netscape, Microsoft, Apache, Oracle, NSCA a dalších) a certifikačními autoritami jako je ve světě VeriSign a u nás I. CA. Dřívější verze SSL 2.0 umožňovala pouze autentizaci serveru (pouze server potřeboval certifikát). Současná verze SSL 3.0 zajišťuje i autentizaci klienta (server i klient se musí prokázat certifikátem).

Základ SSL

SSL je založeno na vytvoření zabezpečeného kanálu mezi klientem (prohlížečem) a serverem. Tento kanál garantuje důvěrnost každé zprávy, která se jím přenáší. SSL nešifruje žádné informace uložené na klientu ani na serveru. SSL zabezpečuje aplikační protokoly jako jsou HTTP, NNTP a Telnet. SSL umožňuje bezpečnou výměnu informací při inicializaci TCP/IP spojení, při kterém se klient a server domluví na konkrétní užité bezpečnosti a vykonají vzájemnou autentizaci certifikáty. Od tohoto bodu, pokud je šifrování aktivováno (počáteční stav), SSL šifruje a dešifruje proud bytů použitého aplikačního protokolu. To znamená, že všechny informace v HTTP požadavku i v HTTP odpovědi jsou plně zašifrované včetně URL, které klient požaduje, potvrzených obsahů formulářů19, přístupových autorizačních informací a všech dat vrácených serverem klientovi.

Protože HTTP+SSL (nebo „HTTPS“) a HTTP jsou dva různé protokoly, které typicky fungují na různých portech (443 a 80). Na stejném systému může běžet zabezpečený i nezabezpečený HTTP server zároveň. To znamená, že můžeme zpřístupnit některé informace všem uživatelům bez zabezpečení a jiné informace již určitým uživatelům užitím zabezpečení. Například v internetovém obchodu by měl být katalog produktů přístupný nezabezpečeně a objednávání a placení zabezpečeně.

SSL používá kryptografii s veřejným klíčem pro výměnu klíče sezení mezi klientem a serverem. Tento klíč je unikátně generován pro každé spojení mezi serverem a klientem a je použit pro šifrování HTTP transakcí (požadavku a odpovědi). Kryptografie s veřejným klíčem je užita jen pro vzájemné ověření a k zašifrování klíče sezení. SSL používá šifrování s privátním klíčem při šifrování další výměny informací. Každá transakce se provádí s různým klíčem sezení, takže i kdyby útočním „ukradl“ jeden klíč, v další komunikaci mu to nijak nepomůže.

Nové webové klientské aplikace mají již zabudovány klíče některých certifikačních autorit jako je VeriSign, další je možno samozřejmě doinstalovat. Tyto zabudované klíče umožňují klientské aplikaci ověřit legitimitu kontrolou identity serveru a identity CA, která danému serveru certifikát vydala. SSL vyžaduje, aby server měl digitální certifikát vydaný důvěryhodnou certifikační autoritou. Tento verifikační proces probíhá transparentně. Pokud klient (prohlížeč neboli uživatel) nedůvěřuje certifikační autoritě (nemá nainstalován její certifikát), která vydala serveru certifikát, klientská aplikace vypíše varovné hlášení. Přidání každé bezpečnostní vrstvy zpomaluje serverové procesy. SSL není výjimkou. Avšak v aplikacích, kde je nutná zvýšená bezpečnost, SSL vysoce převyšuje risk.

SSL proces

SSL používá sérii klient/server výměn a vytvoření zabezpečeného sezení. Dvě hlavní fáze SSL protokolu jsou:

- vytvoření privátní komunikace
- klientská autentizace

SSL proces se může zdát dlouhý a komplikovaný, ale pracuje konkrétně transparentně k uživateli a probíhá hned po vytvoření spojení mezi klientem a serverem. V následujícím výkladu budeme jako certifikační autoritu jmenovat VeriSign, což je světově uznávaná CA.

Následující grafika užívá tuto legendu:

certifikát serveru

hlavní klíč sezení

certifikát klienta

klíč{fráze} zašifrování *fráze klíčem*

klientská aplikace se zabudovaným kořenovým certifikátem VeriSign

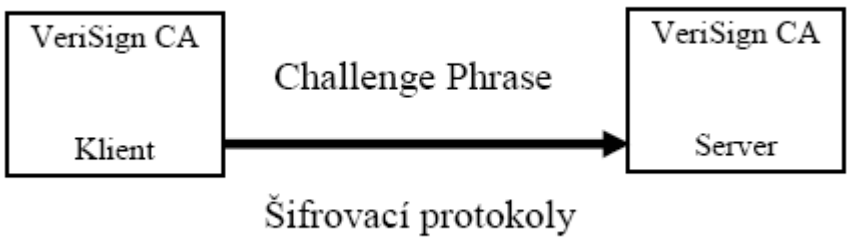
{fráze}klíč dešifrování *fráze klíčem*

aplikace serveru se zabudovaným kořenovým certifikátem VeriSign

výtah zprávy (hash)

Fáze 1: „Klientské ahoj“ (Client Hallo)

Tento první krok nastane, když se klientská aplikace zkouší připojit k zabezpečené stránce. Aplikace nejprve posílá ***řetězec náhodné výzvy*** (*Random Chalange String* nebo *Challenge Phrase*) serveru, potom vybírá, kterou sadu šifrovacích protokolů použít (závisí na protokolech nainstalovaných v aplikaci). Klientská aplikace musí vybrat algoritmus pro výměnu klíče sezení (fáze autentizace serveru), jako je například DES. Dále musí vybrat algoritmus šifrování privátního klíče (jako je RC2 nebo RC4), a hešovací algoritmu zajišťující integritu zprávy (jako je MD5 nebo SHA). Tyto algoritmy se použijí při zabezpečeném přenosu. Je mnoho jiných algoritmů, které se mohou použít.

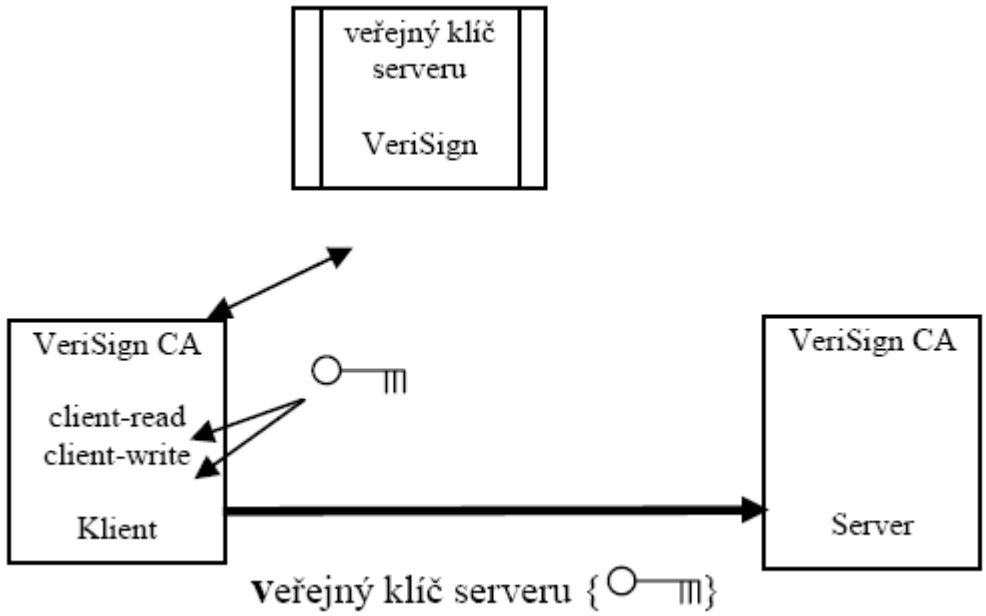


Fáze 2: „Serverové ahoj“ (Server Hallo)

Server potvrzuje svoji identitu navrácením svého certifikátu plus prohlášením, že podporuje sadu algoritmů vybranou klientem. Navíc generuje náhodný identifikátor spojení, který bude použit při komunikační fázi.

Fáze 3: „Hlavní klíč klienta“ (Client Master Key)

Klientská aplikace ověřuje certifikát serveru porovnáním podpisu certifikační autority (CA) v certifikátu serveru s veřejným klíčem CA zabudovaným v klientské aplikaci. Pokud klient nemá klíč CA nebo certifikát CA v klientské aplikaci nesouhlasí s CA serveru, uživatel přijme varovné hlášení, že tento server vlastní certifikát neznámý klientské aplikaci. Uživatel má možnost ukončit spojení, vždy věřit certifikátům nové CA serveru nebo věřit certifikátu pouze při tomto spojení.



Po ověření serveru klient generuje ***hlavní klíč sezení*** (*Master Session Key*). Tento klíč je použit jako semínko k vygenerování komunikačních klíčů klienta i serveru. Dvě symetrické sady párů klíčů jsou vytvořeny: jeden pro příchozí a jeden pro výchozí komunikaci. Protože k vytvoření klíčů byl jako semínko použit pouze jeden klíč, ***zápisový klíč serveru*** (*server write-key*) je shodný jako ***čtecí klíč klienta*** (*client read-key*) a ***čtecí klíč serveru*** (*server read-key*) odpovídá ***zápisovému klíči klienta*** (*client write-key*). Důležité je, že hlavní klíč sezení je vygenerován klientem a ne serverem, což uživateli zajišťuje další vrstvu bezpečnosti.

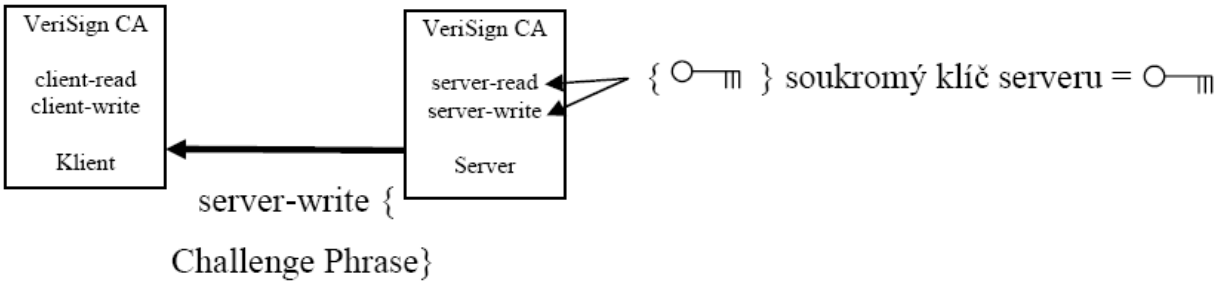
Nakonec klient šifruje klíč sezení veřejným klíčem serveru (certifikát serveru jej obsahuje) a posílá jej serveru.

Fáze 4: „Klient skončil“ (Client Finish)

Klient končí svou část privátní komunikační fáze zašifrováním identifikátoru spojení serveru klientským zápisovým klíčem. Potom čeká na obdržení zprávy „Server skončil“, aby se ujistil, že zabezpečení kanálu je hotové.

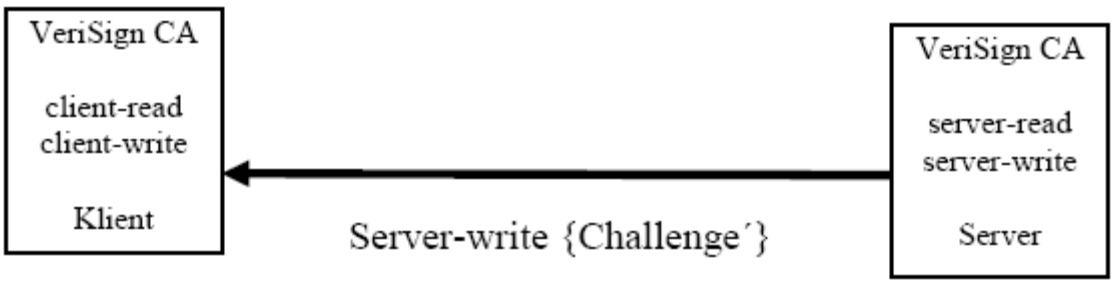
Fáze 5: „Ověření serveru“ (Server Verify)

Server dešifruje hlavní klíč sezení svým privátním klíčem. Klíč sezení použije k vytvoření odpovídajícího párů klíčů (*server-read* + *server-write*). Potom server vrací klientovi inicializační frázi (*Challenge Phrase*) zašifrovanou zápisovým klíčem serveru. Toto je potvrzení autenticity serveru, protože pouze hlavní klíč sezení mohl vytvořit klíč, kterým byla zašifrována inicializační fráze.



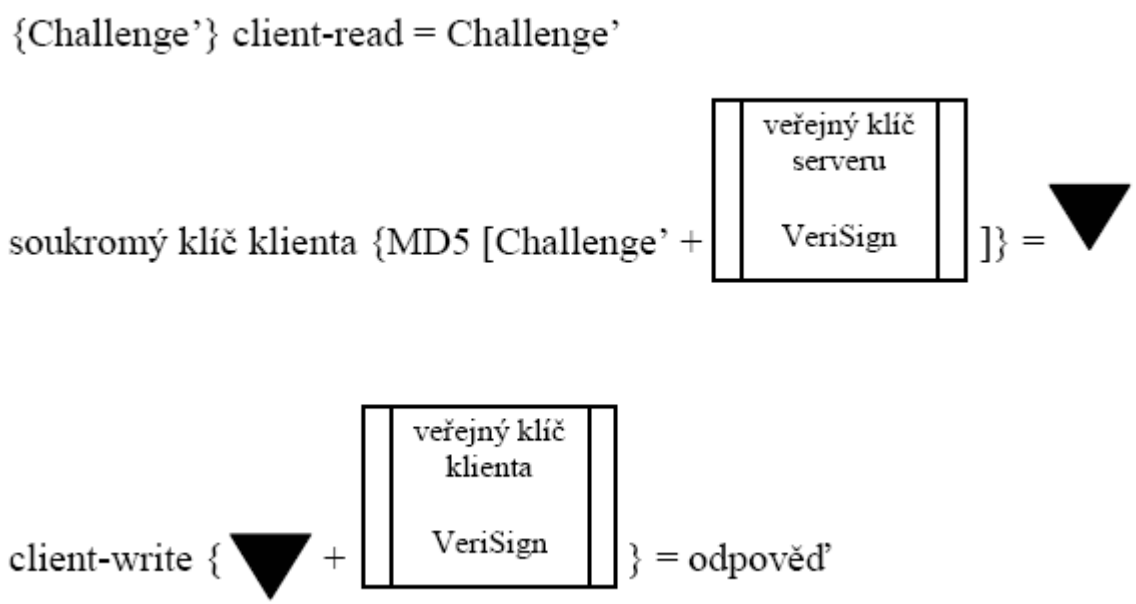
Fáze 6: „Požadavek certifikátu“ (Request Certificate)

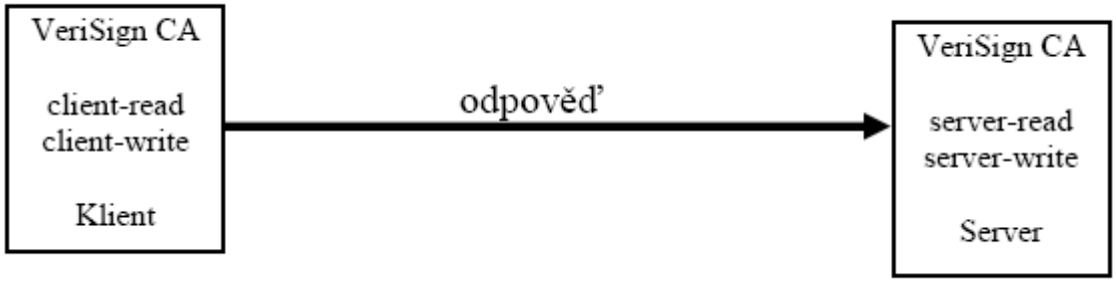
Server nyní požaduje, aby se klient prezentoval platným certifikátem a posílá klientovi novou ***výzvu*** (*Challenge´*) zašifrovanou zápisovým klíčem serveru.



Fáze 7: „Certifikát klienta“ (Client Certificate)

Pokud klient nemá certifikát, odpovídá chybovou zprávou. Jinak dešifruje výzvu serveru (svým čtecím klíčem) a vytváří odpověď, která se skládá z výtahu výzvy serveru (např. algoritmem MD5) plus certifikátu serveru. Tato odpověď plus certifikát klienta je digitálně podepsána klientovým soukromým klíčem a potom odeslána serveru.

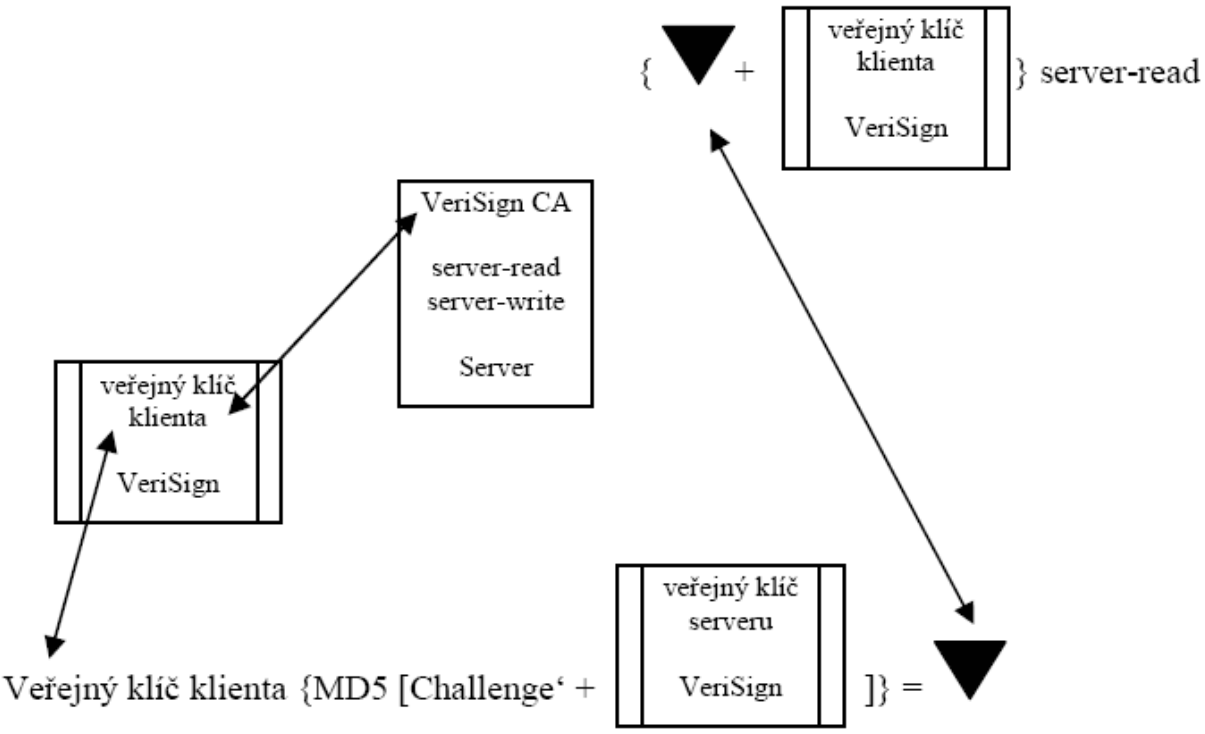




Fáze 8: „Ověření certifikátu klienta“ (Client Cerificate Verification)

Server ověřuje autenticitu klienta dvěma způsoby:

- nejprve kontrolou, že certifikát vydala důvěryhodná CA – podpis certifikátu ověří proti zabudovanému seznamu kořenových CA,
- potom vytvořením výtahu výzvy (*Challenge´*) plus certifikátu serveru a porovnáním



Fáze 9: „Server skončil“ (Server Finish)

Server končí posláním unikátního identifikátoru sezení zašifrovaného zápisovým klíčem serveru. Tento unikátní identifikátor je použit po zbytek sezení, což eliminuje potřebu další výměny, která by dále zpomalovala komunikaci.

Závěr

Problém autentizace přístupu do webových aplikací není jednoduchý. Přístup nikdy nebude stoprocentně bezpečný. Digitální podpis však přináší další neopomenutelnou vrstvu zabezpečení.

Řešení užitím SSL přináší navíc následující výhody:

- SSL probíhá transparentně – uživatel není ničím zatěžován, pouze musí mít v prohlížeč nainstalován certifikát od certifikační autority, kterou server „uznává“
- autentizace probíhá vzájemně – server se autentizuje klientovi a naopak
- komunikace je šifrována
- SSL podporuje většina moderních prohlížečů i serverů

Tvorba certifikátů

První možností je získávání certifikátů od autorizovaných certifikačních autorit jako je v ČR 1. Certifikační autorita. Je možno získat testovací certifikát s platností 14 dní, avšak standardní certifikáty s delší platností již bezplatné nejsou.

Další možností je využít software určený pro vytvoření vlastní certifikační autority a jejích certifikátů. Pro realizaci této diplomové práce bylo zvoleno prostředí příkazové řádky OpenSSL.

Výhody generování certifikátů s OpenSSL:

- standardní součást systému WebServeru Apache
- OpenSSL je volně dostupné na adrese: http://www.openssl.org

Dalším programovým nástrojem určeným pro tvorbu a správu certifikátů je program KeyTool, který je součástí Java 2 Standard Developer Kit 1.4. Hojně využívaný je také kryptografický produkt PGP neboli Pretty Good Privacy, který se umí úzce integrovat do systému a dílčích programů (např. Microsoft Outlook), které mohou šifrování potřebovat.

OpenSSL

OpenSSL je kryptografický nástroj, který implementuje Secure Socket Layer (SSL v2/v3) a Transport Layer Security (TLS v1) a k nim přidružené kryptografické standardy. Openssl je program příkazové řádky pro využití funkcí z knihovny crypto, která je napsána v jazyce C.

Může být použit pro:

- vytváření RSA, DH a DSA klíčů
- vytváření X.509 certifikátů
- počítání výtahů zpráv
- šifrování a dešifrování
- SSL/TLS – testování klienta a serveru
- podepisování, šifrování a následná verifikace a dešifrování S/MIME emailů