

# 1 Základní pojmy bezpečnosti

## Cíle bezpečnosti IS:

- *Důvěrnost* (Confidentiality) – ochrana proti neoprávněnému získání informace
- *Integrita* (Integrity) – ochrana proti neoprávněné modifikaci informace
- *Dostupnost* (Availability) – ochrana proti neoprávněnému odepření přístupu k informaci

**Zranitelná místa** (Vulnerabilities) jsou slabiny v IS, které je možné použít k útoku na IS.

- Při návrhu – chyba v architektuře, analýze, návrhu
- Při implementaci – chyba v kódu
- Při provozu – špatný postup nebo nastavení

**Hrozby** (Threats) jsou situace, které mají potenciál způsobit útok (hacker, prozrazení informace o vstupu).

- *Neúmyslné (pravděpodobnostní) hrozby* – živly, poruchy, chyby v SW, selhání člověka
- *Úmyslné (algoritmické) hrozby*
  - Cílem nejsou data (krádež HW, poškození HW, neoprávněné užití HW)
  - Cílem jsou data (krádež SW, krádež dat, neoprávněná manipulace s daty)
  - Cílem je uškodit (škodlivé programy)

**Aktiva** (Assets) jsou složky IS, které mají hodnotu (fyzická i abstraktní).

**Opatření** (Measures) redukuje pravděpodobnost vzniku útoku.

**Riziko** je kombinace zranitelného místa a hrozby (tedy musí existovat slabina a také příležitost nebo motiv ji využít).

**Systémy HoneyPot** jsou systémy bez bezpečnostních opatření a záplat, které jsou zapojeny v sídi a nechávají na sebe útočit (analýza).

## 2 Škodlivé programy

**Malware** (škodlivý software) bez vědomí a souhlasu uživatele provádí neautorizované činnosti.

**Virus** je program, který se replikuje, aby infikoval co největší část cílového systému. Typicky potřebuje hostitelský program a musí být spuštěn, často provádí destrukční činnost.

- *Boot-sector virus* – infikuje MBR
- *Souborový infektor* – infikuje spustitelné programy (původní koncept)
- *Makrovirus* – infikuje dokumenty s makry
- *Skriptovací virus* – virus psaný ve skriptovacím jazyce, šíří se jako zdrojový kód
- *Polymorfní virus* – modifikuje sám sebe, aby byl hůře detekován
- *Multipartní virus* – kombinuje více typů

**Červ** je samostatný program (bez hostitele), který se replikuje ze systému na systém (ne mezi soubory) v síti.

**Trojský kůň** je program, který na pozadí viditelné činnosti vykonává ještě skrytou škodlivou činnost (instalace hry, ale i krádeže apod.)

**Logická bomba** nic neinfikuje, ale na základě jisté podmínky provede destrukční činnost (zašifruje data, odšifruje až po zaplacení).

**Spam** je zákonem definováno jako nevyžádaná obchodní nabídka, pojem znám jako veliké množství nevyžádaných e-mailů.

**Phishing** je využití sociálního inženýrství k získání dat podvodem. Převážně vizuální simulace známých stránek. **Spyware** sbírá osobní informace a hesla, posílá je útočníkům.

**Bot/Zombie** je infikovaný počítač, který se posléze využije k útoku, často spojeny do sítě (DDoS).

**Rootkit** je virus, který běží pod OS, je tedy špatně detekovatelný.

### 3 Bezpečnostní opatření

**Omezující bezpečnostní opatření** snižují dopad útoku a maximalizují zotavení po útoku.

**Preventivní bezpečnostní opatření** snižují pravděpodobnost útoku, zvyšují cenu útoku (možnost odhalení, čas, náklady).

**Omezení opatření** vzniká hlavně cenou – snížení výkonu, údržba, obsluha, porízení, ...

**Typy bezpečnostních opatření:**

- *Fyzická* – budovy, ploty, zámky, stráže, ...
- *Administrativní* (procedurální) – evidence vstupu, přihlašování, zálohování, ...
- *Personální* – osvěta a školení, procedura přijímání a propouštění, ...
- *Technická* (logická) – identifikace, autentizace, řízení přístupu, protokoly, šifrování, ...

### 4 Bezpečnostní politika

**Celková bezpečnostní politika** popisuje globální cíle organizace a IS z hlediska bezpečnosti.

- Cílem je ochrana majetku, pověsti a činnosti celé organizace.
- Dokument je nadčasový, nezávislý na konkrétní IT implementaci, vedený jako norma, veřejný a závazný.
- Stanovuje citlivé informace, aktiva, klasifikaci, hierarchii zodpovědnosti, práva a minimální sílu bezpečnostních mechanismů.

**Systémová bezpečnostní politika** definuje *implementaci* celkové bezpečnostní politiky v konkrétní IT situaci.

- Volba technických, procedurálních, logických a administrativních opatření.
- Principy a pravidla ochrany IS.
- Pro rozsáhlejší systémy lze vybudovat SBP pro jednotlivé části.

**TR 13335** je norma pro řízení bezpečnosti, definuje proces sestavení CBP a SBP.



**BS 7799** je britský standard pro řízení bezpečnosti (posléze přijat ISO), definuje 10 řídicích principů.

**Modely bezpečnosti** formálně vyjadřují bezpečnostní politiku. Dělí se na DAC/MAC, jedno/víceúrovňové, podle funkce. 

**Referenční monitor** je bezpečnostní model, který lokalizuje bezpečnostní funkce do jediného místa, které nelze obejít.

**Bell-LaPadulův model důvěrnosti** stanovuje stupeň důvěry v subjekt  $C(s)$  a úroveň důvěrnosti objektu  $C(o)$ . Platí, že subjekt  $s$  může číst objekt  $o$ , pouze pokud  $C(s) \geq C(o)$ . Zároveň pokud může číst  $o$ , pak může zapisovat do  $p$ , pouze pokud  $C(p) \geq C(o)$  (aby nevyzradil).

**Bibův model integrity** stanovuje stupeň integrity subjektu  $I(s)$  a úroveň integrity objektu  $I(o)$ . Platí, že subjekt  $s$  může modifikovat objekt  $o$ , pouze pokud  $I(s) \geq I(o)$ . Zároveň pokud může číst  $o$ , pak může zapisovat do  $p$ , pouze pokud  $I(p) \leq I(o)$  (aby nepokazil).

## 5 Bezpečnostní funkce

**Požadované funkce:** důvěrnost, integrita, dostupnost, účtovatelnost (monitoring událostí), audit, anonymita, pseudonymita, nemožnost sledování

**Řízení přístupu** – zajišťuje důvěrnost a integritu.



- *Nepovinné řízení přístupu* (DAC) – práva uživatele k objektům, jeden stupeň utajení
- *Povinné řízení přístupu* (MAC) – stupeň utajení a kategorie tvoří atribut uživatele i objektu, pravidla přístupu
- *Minimální řízení přístupu* – k některým objektům přístup pouze pomocí privilegovaných procesů
- *Základní řízení přístupu* – uživatel má právo k objektům i procesům
- *Vyšší řízení přístupu* – přístup dle kombinace uživatel/objekt/proces



**Skryté kanály** porušují důvěrnost, předávají informaci v rozporu s bezpečnostní politikou.

- *Paměťové* – existence/neexistence souboru, atributy, délka, stavy zařízení
- *Časové* – zatížení procesoru, doba přístupu, zatížení I/O
- *Kombinované* – pohyb diskové hlavy



**Opětovné použití prostředků** je důležité pro důvěrnost. Přidělený objekt nesmí obsahovat data nebo práva jiného vlastníka (útok typu scavenging).

**DVB** (Důvěryhodná výpočetní báze) zajišťuje integritu sama sebe a svěřených objektů. Nesmí existovat možnost ji obejít.

**Fyzická integrita** je zajištěna tzv. tamper-resistant hardware. Buďto se útok pouze eviduje (pečeť), nebo se reaguje (zničení), nebo je odvrácen (útok je nemožný).



**Návrat** zajišťuje integritu schopností navrátit se do předchozího platného stavu. Příkladem je zálohování nebo transakce.



**Oddělení rolí** zajišťuje integritu definicí rolí pro různé funkce (správce, uživatel, auditor, operátor apod.).



**Autonomní testování** je schopnost systému poznat neintegritní stav. Systém si testuje HW i SW, manuálně, automaticky při startu nebo průběžně.



**Přidělování prostředků** je problém dostupnosti. Systém zajistí dostupnost kvótami, omezeními nebo prioritou.

**Systém opravitelný za provozu** zvyšuje dostupnost, může být opravitelný plně nebo jen zčásti.

**Robustnost** je schopnost systému být dostupný i při poruše (odolnost pouze některých částí, částečná funkčnost až plná funkčnost).

**Zotavení po chybě** zvyšuje dostupnost, buďto je potřeba manuální zásah, nebo automaticky.



**Identifikace a autentizace** jsou základem účtování. Silná autentizace je kryptografická autentizace.



**Audit** provádí evidenci a analýzu účtovaných událostí. Auditní data je třeba oddělit a chránit, musejí mít správnou granularitu. Je možné prostředky rozšířit o možnost poplachu a detekce útoku.

**Důvěryhodný kanál** je zaručené propojení mezi uživatelem a DVB, pokud je ustaven pouze na počátku spojení je autentizační, jinak je úplný.

## 6 Kritéria hodnocení bezpečnosti informačních systémů

**Kritéria se nezabývají** opatřeními (logické i fyzické), metodologií hodnocení, dohodami o uznávání, akreditací a kryptografickými algoritmy.

**Orange Book (TCSEC)** jsou první kritéria od DoD.

**Úrovně TCSEC:**

- D – *minimální ochrana*
- C1 – *nepovinná ochrana*  
identifikace, autentizace, nepovinné řízení přístupu
- C2 – *ochrana řízeného přístupu*  
opětne použití a audit
- B1 – *víceúrovňová ochrana*  
povinné řízení přístupu pro některé objekty a subjekty  
neformální model bezpečnostní politiky
- B2 – *strukturovaná ochrana*  
povinné řízení přístupu pro všechny objekty a subjekty  
formální model bezpečnostní politiky  
bezpečná cesta přihlášení  
princip minimálních privilegií  
analýza paměťových skrytých kanálů  
správa konfigurace
- B3 – *bezpečnostní domény*  
analýza všech skrytých kanálů  
mechanismus validace referencí (ref. monitor)  
omezení na vytváření kódu  
požadavky na dokumentaci a testování

- A1 – *verifikovaný návrh*  
formální analýza a verifikace  
důvěryhodná distribuce

#### Úrovně TCSEC v praxi:

- C1, C2 – mírně vylepšené současné operační systémy, aplikace nepoznají
- B1 – operační systémy se musejí pozměnit více (MAC), některé aplikace vyžadují změny (málo)
- B2 – OS jsou změněny zásadně, aplikace nefungují
- B3 – systémy, které nezvládly A1 (stejná funkčnost, ale formální návrh)
- A1 – systémy navržené od základu, netradiční metody

#### Nedostatky TCSEC:

- Chybí integrita dat
- Nezná počítačovou síť
- Směšuje funkčnost a zaručitelnost
- Různé úrovně abstrakce v dokumentu

UNIX ve třídě C2 může být, jelikož jeho řízení přístupu vyhovuje, pouze stačí přidat znovupoužitelnost (disk, paměť, obrazovka), zajistit možnosti auditu, tři dokumentace pro hlavní role a šifrovat hesla (shadow).

**Index rizika** se používá k vyjádření stupně bezpečnosti systému. Index rizika  $I = R_{max} - R_{min}$ , kde  $R_{max}$  je citlivost dat (neklasifikovaná až přísně tajná) a  $R_{min}$  je prověření uživatele (neprověřený až prověřen pro přísně tajné). Podle indexu rizika se pak určuje požadovaná úroveň dle TCSEC (stupnice pro otevřené a uzavřené prostředí).



ITSEC jsou evropská kritéria, která vznikla spojením národních kritérií jako alternativa k TCSEC.

- Rozlišuje produkty a systémy.
- Není lineární, ale má dva rozměry (funkčnost a zaručitelnost).
- Funkčnost: F-C1 až F-B3, definuje i další a říká jak definovat vlastní.
- Zaručitelnost: E1 až E6, což odpovídá C1 až A1.



**Síla mechanismů** je v ITSEC vágně popsána, ITSEM již specifikuje přesněji:

- *Znalosti* – jak moc útočník zná produkt (začátečník, zkušený, expert)
- *Prostředky* – čas a vybavení
- *Příležitost* – neovlivněno útočníkem (komplot, šance, možnost detekce)

## 7 Kritéria CC (Common Criteria)



**Common Criteria** jsou standardizovaná kritéria hodnocení bezpečnosti systémů. Existuje dohoda vzájemného uznávání a státy mají národní schémata pro použití CC.

CC se dělí na třídy (FDP), rodiny (ACC) a komponenty (1)  $\rightarrow$  FDP\_ACC.1.

**Třídy funkčnosti Fxx:** AUdit, COmmunication, Cryprographic Support, Data Protection, Identification and Authentization, security ManagemenT, PRivacy, functionality ProTectiion, Resource Usage, Trusted Paths

**Úrovně zaručitelnosti** jsou kompatibilní s TCSEC – EAL1 až EAL7.

**Třídy zaručitelnosti Axx:** Configuration Management, Delivery and Operation, Guidance Documents, Life Cycle support, Vulnerability Assesment, DeVelopment documentation, TEsting

CEM (Common Evaluation Methodology) popisuje aktivity hodnotitele CC.

## 8 Kryptografie

**Kryptografie** je transformace otevřeného textu na šifrovaný a naopak.

**Kryptoanalýza** je transformace šifrovaného textu na otevřený bez znalosti klíče.

**Kryptologie** je věda zabývající se kryptografií a kryptoanalýzou.

**Šifra (šifrovací algoritmus)** je algoritmus pro provedení kryptografie.

**Útoky:**

1. *Ciphertext-only attack* – znalost pouze šifrovaného textu, hledá se klíč nebo otevřený text
2. *Known-plaintext attack* – znalost šifrovaného i otevřeného textu, hledá se klíč
3. *Chosen-plaintext attack* – jako výše, ale útočník si volí otevřený text

**Typy šifer:**

1. Klasické
  - (a) Steganografické **ukryjí přenášený text uvnitř jiného textu**
  - (b) Transpoziční **mění pořadí znaků v textu**
  - (c) Substituční **nahrazují jednotlivé znaky jinými znaky**
    - i. Monoalfabetické
    - ii. Polyalfabetické **více různých substitucí, pro každý znak jiná sub. funkce**
2. Moderní
  - (a) Blokované
  - (b) Proudové
    - (a) Symetrické
    - (b) Asymetrické

**Kerckhoffův princip** říká, že bezpečnost musí záviset pouze na utajení klíče, nikoliv algoritmu nebo postupu – "Security by obscurity doesn't work".

## 9 Klasické šifrovací algoritmy

**Caesarova šifra** je první známá algoritmická šifra, posunuje znaky v abecedě o 3, lze vylepšit klíčem (1 až 26).

**Monoalfabetická substituční šifra** nahrazuje jednotlivá písmena abecedy jinými podle klíče (převodní tabulky).

Klíče jsou všechny možné permutace abecedy, tedy  $26! = 4.10^{26}$  možností, ale to by se špatně pamatovalo, používá se klíčové slovo doplněné ostatními písmeny v abecedním pořadí, někdy také zapsány do matice a čteny jistým průchodem.

**Frekvenční analýza** je zjištění počtu výskytů jednotlivých písmen v textu. Frekvence písmen se monoalfabetickou substitucí nezmění, využívá se při kryptoanalýze.

**Kódová kniha** je monoalfabetická polygramová šifra, nahrazuje skupiny znaků (slova) jinými (čísla). Frekvence výskytů lze zarovnat (homofonní vlastnost) více možnostmi pro jedno slovo a náhodným výběrem. Příkladem je Zimmermanův telegram.

**Polyalfabetická šifra** šifruje každý znak jinou substitucí, výsledný text je tak blíže náhodnému šumu, nelze použít statistické metody jako u monoalfabetické šifry.

**Vigenerova šifra** využívá caesarova principu, ale nepoužívá jediné klíčové písmeno. Klíč se replikuje do potřebné délky a pak se "sčítá" s textem (původně posunutí o znaky, dnes se používá spíše XOR). Výhodou je mj. snadné vytvoření Vigeněrový tabulky pro rychlou práci se šifrou.

**Útok na Vigeněrovu šifru** je založen na nalezení délky klíče  $k$  a poté rozdělení zprávy na  $k$  zpráv, které se analyzují jako monoalfabetické nezávislé zprávy. Délka klíče se dá zjistit ze vzdáleností stejných znaků v textu (je větší šance že se stejné písmeno šifrovalo stejným klíčem než že se jiné písmeno zašifrovalo pomocí jiného klíče na stejný výsledek).

**Vernamova šifra** je jediná šifra, která má matematicky dokázanu nerozluštitelnost. Principem je XOR zprávy stejně dlouhým klíčem. Klíč musí být (opravdu) náhodný, použit pouze jednou a musí mít délku zprávy. Obrovský problém je pak transport klíče. Využita pro komunikaci mezi Pentagonem a Kremlem.

**Sloupcová transpozice** je jednoduchý zápis textu do matice a poté čtení po sloupcích, většinou se provádí vícekrát.

**Složené šifry** kombinují substituční a transpoziční šifry za sebe. Kombinace stejného typu nepřináší zlepšení, ale kombinace substituce a transpozice ano (viz moderní algoritmy).

**Posouvané a rotované abecedy** berou klíč a pro každý znak jej posunou buďto horizontálně nebo vertikálně (pokračují v abecedě). Vhodné pro mechanické stroje.

**Rotorové stroje** používají rotory s kontakty, které implementují substituci, pootočením se posouvá nebo rotuje abeceda klíče. Rotory jsou zapojeny za sebou, po protočení prvního se otočí další atd.

### Enigma

- Nejznámější rotorový stroj.
- Původně pro komerční účely, pak převzal a upravil Wehrmacht.
- Tři rotory (námořní čtyři), pak přidány dva další pro výběr.

- Reflektorová deska (měla virtuálně zvýšit počet rotorů), vnesla chybu – znak se nemůže kódovat sám na sebe.
- *Nastavení* – výběr rotorů, pozice rotorů, nastavení posunutí rotorů.
- Přidána propojovací deska (až s 10 kabely), která provede substituci před výstupem.
- Při ukradení enigmy je počet možností 17576 plus  $10^{11}$  kvůli propojovací desce!
- Zpočátku lámána kvůli opakování klíče na počátku zprávy.
- Posléze lámána v Bletchley parku "počítačem" kvůli existenci hlášení počasí.


## 10 Symetrické šifrovací algoritmy

**Symetrické algoritmy** jsou algoritmy se sdíleným klíčem. Jsou bezpečné, pokud nejlepší útok je útok silou (vyzkoušení všech klíčů). Zajišťují důvěrnost a autentizaci, dá se zajistit i integrita (přidáním kontrolního součtu/hashe). Nelze zajistit nepopiratelnost, jelikož klíč existuje ve dvou exemplářích. Dá se jednoduše vytvořit nový algoritmus. Jsou relativně rychlé.

**Bloková šifra** pracuje se 64b (někdy 128b nebo 256b) dat. Existuje pak  $2^{64}$  možných zobrazení, ale nelze náhodně, jelikož by tabulka byla příliš veliká. Náhodné zobrazení se aproximuje pomocí klíče.

**Feistelova šifra** je základem několika symetrických šifer. Je iterativní, probíhá v kolech a dělí blok na dvě části – prohodí si pořadí, jedna zůstane nezměněna, druhá je XORována s výsledkem transformace části první (transformaci ovlivní klíč).

**Subklíče** jsou "klíče generované z hlavního klíče", používají se v jednotlivých kolech šifrování. Problémem je existence slabých (samé nuly nebo jedničky) a poloslabých (opakují se co jedno kolo) subklíčů.

 **Algoritmus DES** vytvořilo IBM a modifikovalo NSA, zašifroval nejvíce bitů, jelikož je certifikován pro banky a státní správu.

- Klíč má pouze 56b, subklíče dokonce jen 48, nenaplní všechny  $2^{64}$  permutací bloku.
- Dnes lámatelný hrubou silou kvůli krátkému klíči.
- Možná zadní vrátka pro NSA.
- Útoky využívají slabé a poloslabé klíče, komplementární klíče, ale zrychlení je minimální.
- Vytvořen pro efektivitu v HW (permutace).

**Algoritmus DES-EDE (3DES)** nahrazuje DES hlavně tam, kde je potřeba jeho certifikace. Jedná se o stejný algoritmus, ale 3x za sebou se dvěma klíči (Encrypt-Decrypt-Encrypt). Dva algoritmy za sebou nejsou použitelné, kvůli "meet-in-the-middle" útoku. Další výhodou je, že se dá jednoduše převést na jednoduchý DES, pokud se obě části klíče rovnají.

**Další algoritmy:**

- *IDEA* – evropský protivník DESu, 128b klíč, efektivní v SW
- *AES* – nový standard, náhrada DESu, 128b, 192b a 256b
- *RC2, RC4, RC5* – Ron Rivest, malý a rychlý

**Režimy blokových šifer** – způsob řetězení bloků



- *ECB* (Electronic CodeBook) – každý se šifruje nezávisle, možnost slovníkového útoku a nahrazování
- *CBC* (Cipher Block Chaining) – výstup se XORuje s dalším blokem (první blok XORován s IV)
- *CFB* (Cipher FeedBack) – šifruje se 64b blok, ale výstupem jsou bloky o 8b, které se XORují se zprávou a pak vloží na vstup, který se o 8b posune
- *OFB* (Output FeedBack) – oboba CFB, ale na vstup se dává 8b výstupu ještě před XORem se zprávou
- Další mají hojně využití v šifrování disků apod.

## 11 Asymetrické šifrovací algoritmy

**Asymetrické algoritmy** jsou algoritmy s veřejným a soukromým klíčem. Neútočí se silou, ale na princip vytvoření klíče. Zajišťují důvěrnost nebo autentizaci, integritu a nepopíratelnost. Pro všechny vlastnosti najednou je třeba šifrovat dvakrát. Algoritmy se nedají vytvořit, musí se najít matematický princip. Jsou pomalé.

**Klíče** jsou dva – soukromý a veřejný

**Šifrování veřejným klíčem** zajišťuje pouze důvěrnost, kdokoli může text nahradit jiným, jelikož veřejný klíč je přístupný. Dešifrovat lze pouze soukromým klíčem.

**Šifrování soukromým klíčem** zajišťuje integritu, autentizaci a nepopíratelnost, existuje jediná osoba, která mohla šifrovat soukromým klíčem. Veřejným klíčem se dešifruje a také ověří, že je vše v pořádku.

**Algoritmy** mohou podporovat obě operace nebo pouze jednou z nich. Pokud podporují obě je nebezpečné používat pro obě operace stejný klíč (možnost podstrčení textu).

**Faktorizace čísel** je matematický problém, na kterém staví RSA. Číslo  $n$  se skládá z modulů  $p$  a  $q$  (prvočísla). Nalézt tato prvočísla pouze z  $n$ , které je velké ( $\geq 1024b$ ) je velmi pomalé.

**Diskrétní logaritmus** je obdobný problém, kde se hledá exponent  $k$  takový, aby platilo  $m^k \equiv p \pmod{q}$ .

**Knapsack** je matematický problém (a také název algoritmu) kterak do batohu naskládat obsah s nejvyšší cenou. Složitost je dána nesourodostí jednotlivých kusů zboží. Nepoužívá se, jelikož byl několikrát prolomen.

**Eliptické křivky** je zobrazení čísel na křivkách, které zesložituje faktorizaci čísel a diskretní logaritmus. Algoritmy mají kratší klíče, používaly se hodně v době, kdy ostatní byly patentovány.

**Algoritmus RSA** (Rivest, Shamir, Adelman) pracuje jako bloková šifra, blok představuje velké celé číslo  $n$  (1024 až 4096 bitů).

- $n$  je veřejný modulus, složky  $p$  a  $q$  se generují a pak zahodí.
- $e$  je veřejný exponent, typicky 3 nebo  $2^{16} + 1$ .
- $d$  je soukromý exponent.
- Veřejný klíč je  $(n, e)$ , soukromý je  $(n, d)$ .
- Vygenerované  $p$  a  $q$  musejí být téměř prvočísla.
- Zbytek se počítá ze vztahu  $d \times e \pmod{(p-1)(q-1)} = 1$ .
- Šifrování:  $c = m^e \pmod{n}$  (veřejným)
- Dešifrování:  $m = c^d \pmod{n}$

- Podpis:  $s = m^d \bmod n$  (soukromým)
- Ověření:  $m = s^e \bmod n$

**Hashování** je algoritmus, který je ireverzibilní, poskytuje možnosti pro zajištění integrity zprávy. Bylo by možné použít asymetrickou kryptografii, ale ta je pomalá, proto se nejprve udělá hash a jen ta se zašifruje.

- Důraz na rychlost.
- Může a nemusí mít klíč.
- Nesmí jít nalézt původní zpráva jen z hashe (ireverzibilitnost).
- Nesmí jít nalézt zpráva se stejnou hashí jako studovaná zpráva.
- Nesmí jít nalézt dvě libovolné zprávy se stejnou hashí.
- Algoritmy MD2, MD4, MD5, SHA, SHA-1, SHA-2, RIPEMD, MAC

## 12 Certifikáty veřejných klíčů

**Distribuce veřejných klíčů** pro asymetrickou kryptografii má základní problém – jak ověřit, že veřejný klíč opravdu patří dané osobě/organizaci. Je potřeba aby byl volně získatelný (je zapotřebí k ověření podpisu a odeslání šifrované zprávy), ale zároveň bylo jasné, komu patří (kdo si zprávu přečte, kdo ji podepsal).

**Kryptografické certifikáty veřejných klíčů** zajitují možnost ověření správnosti dvojice *jméno-klíč*.

**X.509** je nejrozšířenější standard pro certifikáty, vznikl jako součást (dnes nepoužívané) adresářové služby X.500, implementován v Novellu a LDAP.

**Certifikační autorita** je třetí osoba, které obě strany důvěřují, ta podepíše certifikát, který svazuje jméno a klíč. Uživateli pak stačí pouze ověřit podpis certifikátu, tedy musí znát jediný klíč – klíč CA.

**Certifikát obsahuje:**

- Jméno vlastníka (původně kanonické jméno podle X.500).
- Veřejný klíč.
- Jednoznačnou identifikaci – vydavatel (CA) a sériové číslo.
- Typ algoritmu podpisu, typ použití (podpis, šifrování).
- Platnost certifikátu.
- Další...

**Strom certifikace** – má hierarchickou strukturu, kořenová CA podepisuje certifikát nižším CA a ty teprve podepisují certifikát uživatele. Koncovému uživateli je třeba poslat všechny certifikáty na této cestě, aby mohl použít pouze veřejný klíč kořenové CA. Stromů může existovat více, nejtěžší je pak bezpečně zjistit kořenový veřejný klíč jiné CA.

**Křížové certifikáty** jsou způsobem pro certifikaci mezi více stromy, které se nemohou spojit.

**Vydání certifikátu** probíhá osobně v CA na základě prototypového certifikátu (vyplněny hlavní položky), který je podepsán soukromým klíčem, aby bylo jasné, že osoba má soukromý klíč (tedy neváže své jméno na cizí veřejný klíč).

**CRL** (Certificate Revocation List) je způsob jak předčasně zneplatnit certifikát, problém je, že od nahlášení až po stažení CRL u koncového uživatele může uplynout až 48h.

**OCSP** (Online Certificate Status Protocol) umožňuje zjištění platnosti on-line a minimalizuje prodlevu revokace.

**Elektronický podpis** silně souvisí s certifikáty, bez nich nelze právně použít (je potřeba právě ona vazba klíče a jména).

**Digitální podpis** je přesnější termín pro podpis pomocí kryptografie, elektronický je vlastně i obrázek podpisu a podobně, ale je rozšířen pojem elektronický podpis.

## 13 Bezpečnost operačních systémů

**TPM** (Trusted Platform Module) zajišťuje bezpečnost pomocí bezpečného úložiště klíčů v tamper-resistant hardware a umí ověřovat podpisy programů a OS. Implementuje kryptografické operace a obsahuje podpis výrobce, který je možné ověřit.

**Řetěz důvěry pro zavedení OS:**

- V HW je uložen klíč, který je podepsán výrobcem.
- HW certifikuje firmware modulu.
- Firmware certifikuje bootloader.
- Bootloader certifikuje a zavádí OS.
- OS přebírá kontrolu a certifikuje veškeré programy.
- OS na počátku zajistí bezpečný čítač hodin a synchronizuje se servery, stahuje HCL a SRL (povolený a revokovaný seznam).

**Před zpřístupněním aplikace** se aplikace zavede, zkontroluje se hash, pak ID platformy (zda odpovídá), platnost licence a revokační seznamy.

**Bezpečné úložiště** poskytuje dlouhodobou úschovu dat, lze specifikovat i jiný příjemce tajemství než původce dat.

**Vzdálená atestace** spočívá v odeslání stavu (schválená konfigurace HW a SW, platné licence apod.) vzdálenému uzlu (poskytovatel připojení apod.).

**Vstup a výstup** musejí být také zabezpečeny, jinak by si uživatel mohl myslet, že vše je v pořádku, ve skutečnosti jde o podvržený výstup.

**Bezpečnost FS – Unix:** starý model, 3 úrovně práv a 3 práva, možnost Set UID nebo Set GUID, sticky-bit.

**Bezpečnost FS – Windows:** až od NT, model subjekt/objekt (lze rozšířit i mimo FS), subjekt je proces s UID, generuje token (obsahuje přístupové atributy), lze nastavit i privilegia (schopnosti provádět systémové operace), objekt má ACL, na základě kterého se určí akce, také SACL pro audit.

**Unix login** uchovává hesla jako jednosměrné šifry na osoleném textu.

**Nástroje pro analýzu bezpečnosti OS** – CIS Benchmark, L0phtcrack, Nessus, MBSA aj. komerční; kontrolují porty, přístupy na služby, verze, nastavení a zranitelnosti

## 14 Bezpečnost databází

**Bezpečnostní mechanismy v DB:** DAC, MAC, inference, šifrování

**DAC v DB** – nastavují se přístupová práva uživatele k objektům a operacím nad nimi:

- `GRANT privileges ON object TO users [WITH GRANT OPTION]`
- `REVOKE [GRANT OPTION FOR] privileges ON object FROM users`
- Pomocí kombinace příkazů (hlavně s `GRANT`) vzniká nepřehlednost.
- Jisté zjednodušení pomocí pohledů a přístupu pouze k nim.

**MAC v DB** – stupeň ověření pro uživatele a stupeň utajení pro objekty:

- Záznamy vedeny jako *polyinstance* (několik instancí dat, ale pro různé stupně utajení s vynecháním informace).
- Možnost použití RBAC.
- Problém *inference* – získání dat z agregovaných/statistických operací (SUM).

**Významné zranitelnosti DB:**

- Implicitní hesla (admin/admin apod.) nebo slabá hesla
- DoS, buffer overflow
- Nesprávná konfigurace/práva
- Špatná správa

## 15 Čipové karty

**Čipová karta** je objekt, který obaluje potřebné operace a nemůže být ve svém výpočtu ovlivněn (obaluje snížení telefonních jednotek, sumy peněz, obaluje ověření podpisu apod.).

**ISO 7816** specifikuje karty typu Smart-card – rozměry, vývody, napětí, protokol APDU.

**Zabezpečení:**

- Přístupová práva k datům zajišťuje mikrokontroller.
- Přístup k paměti je kontrolován v HW.
- Čtení a zápis pomocí kryptografického klíče.
- Nutná autentizace pomocí PIN.
- Pomalé rozhraní zamezuje útoku na PIN.
- ROM nelze modifikovat ani mazat.
- Obsahuje tamper-resistant HW.
- Zajištěna unikátnost.
- Automatické mazání při útoku sondou.

**Protokol APDU** (Application Protocol Data Unit) specifikuje dvě zprávy – *příkaz* a *odpověď*. Mohou obsahovat kromě třídy a instrukce s operandy také dodatečná data.

### Fyzické útoky:

- *Přístup k čipu* – odleptání kyselinou, jemné obrušování
- *Čtení obsahu* – obnovení programátorské pojistky (povolí zápis), např. UV
- *Mikrojechly* – tenké sondy, napojí se na obvody
- *Elektronový mikroskop* – zkoumá strukturu čipu
- *Spodní rentgenování* – struktura tranzistorů
- *Elektrooptické vzorkování* – laser a krystal nad obvodem, odchylí se při průchodu proudem
- *Laserový nůž* – přerušení obvodů
- *Iontový paprsek* – ”pájení”

### Levné útoky:

- *Zákmit* – při nečekané změně hodin nebo resetu se nemusí provést instrukce (přeskočí se JMP a např. se vypíše paměť). Změny teploty, silné záření, změny napájení mohou vyvolat tyto zákmity.
- *Zabránění zápisu* – staré čipy pro zápis potřebovaly vyšší napětí (pin Vpp), stačí jej utrhnout (staré telefonní karty).
- *Diferenciální chybová analýza* – analýza dopadu chyb na algoritmy
- *Časový kanál* – instrukce mají různou dobu provádění, klíč také ovlivňuje
- *Výkonová analýza* – průběh napájení se pro instrukce a bity klíče liší
- *Logické útoky* – na chyby implementace a návrhu (zadní vrátka apod.)
- *Odposlouchování a replay* – bezdotykové karty

