Obsah

Shannonův model kryptosystému 6
Kerchhoffův princip 6
Kategorie útoků na kryptosystém 7
Kryptoanalýza 7
Základní rozdělení šifer 8
Vernamova šifra 9
Teorie informace 9
- entropie jazyka, krytptosystému 10
- redundance 10
- jednotkový odstup 11
Teorie složitosti 11
- klasifikace problémů, třídy složitosti 11
- vztah mezi třídami P a NP a důležitost pro kryptografii 11
Steganografie 13
- obecné principy 13
- příklad použití dříve a dnes 14
Iterované blokové šifry 14
- Feistelova šifra a EFN 14
- DES - včetně opearací v rundě 18
- 3DES 19
- AES (Rijndael) – všechno 19
AES operace v rundě 20
AES – šifrování 23
AES - dešifrování 23
AES - optimalizace 24
AES – bezpečnost 24
- finalisté na AES - základní charaktetistiky + poznat algoritmus podle jeho vnitřní struktury 25
Obecné charakteristiky finalistů 25
Kandidáti na AES – MARS 26
Kandidáti na AES – RC6 – 32/20/16 26
Kandidáti na AES – Twofish 27

Kandidáti na AES – Serpent 28
- generování rundových klíčů - pouze DES,AES 31
Lineární a diferenciální kryptoanalýza - základní principy 31
Proudové šifry - synchronní, asynchronní 32
Shanonnova podmínka nepodmíněné bezpečnosti: 32
Synchronní proudová šifra 33
Asynchronní proudové šifry 33
- algoritmy RC4,A5/1 34
- projekt eStream - obecně, cíle, příklad algoritmů (ty není potřeba detailně znát) 36
E-stream Profile 2 - Trivium 37
E -stream Profile 1 - Rabbit 37
E-stream Profile 1 - SOSEMANUK 37
E-stream Profile 1 – Salsa20/12 37
ChaCha20 38
Režimy činnosti blokových šifer 38
- obecné informace: 38
- ECB, CBC,OFB,CFB, CTR - nejenom poznat, ale i umět nakreslit 38
- režimy AEAD pouze základní rozdělení + příklady 40
Asymetrické algoritmy 40
- rozdělení (IFP,DLP,ECDLP) 40
- RSA, DSA (není v přednášce?), DH 41
- ECDH 41
- srovnání IFP x DLP x ECDLP 42
- doporučené délky klíčů, suite B 42
Suite B 43
Hashovací funkce 43
- SHA-1, SHA-2, SHA-3 44
- CMAC,HMAC 45
Autentizace 46
- rozdělení 46
- principy jednotlivých technik 46
Biometrika 46
Tokeny 46

- biometrické autentizační metody - principy, rozdělení 47
- PAP,CHAP/MSCHAPv2 47
- TACACS+, RADIUS, DIAMETER, Kerberos 48
- EAP obecně , EAP-TLS, EAP-PEAPv0 49
VPN 49
- obecně 49
Protokoly: 50
- autentizace a dojednání klíčů (ISAKMP, IKEv1,IKEv2) 50
- funkce SPD, SAD, SPI 51
- DH grupy, PFS 52
SSL/TLS 52
- SSLv3 / TLSv1.0,1.1, 1.2 52
- dojednání klíčů 53
- DTLS 53
- X.509 certifikát struktura, princip, ... 54
Certifikační autorita 55
- certifikační cesta, validace certifikační cesty 55
- životní cyklus certifikátu 56
- kvalifikovaný certifikát 56
- CRL, OCSP 56
- PKI 57
Elektronický podpis 57
- princip 57
-EP, zaručený EP, kvalifikovaný EP 59
- elektronická pečeť 59
- časové razítko 59
HSM - princip, použití 59
SSH - princip, vlastnosti S/MIME, PGP - principy, vlastnosti 60
Zabezpečení v bezdrátových sítích - pouze v 802.11 60
- WEP, WPA, WPA2 60
- autentizace - PSK, 802.1x 61
- útoky 62
Zabezpečení v mobilních sítích - GSM, UMTS 62

Obsah - horní část dokumentu

Co nemusíte umět:
- kreslit vnitřní struktury většiny šifrovacích algoritmů (kromě těch, kde to budu explicitně vyžadovat, jako třeba AES)
- znát konstanty typu "počáteční hodnoty registrů u SHA-1" nebo vnitřní struktura S-boxů u DESu
- znát nazpamět záhlaví protokolů (ale znát význam důležitých polí typu k čemu je pole SPI u protokolu ESP)
- různé modifikace DESu (GDES, DESX...)
- algoritmus IDEA
- průběh volby AES
- algoritmus CAMELLIA
- podrobné informace a vnitřní strukturu algoritmů z projektu eStream
- matematické detaily o Rabinově kryptosystému, LUC, El-Gamal, McElice
- IEEE P1363
- algoritmy ECIES,ECMQV
- volba SHA-3
- algoritmy MDC-2,MDC-4, Hirose, MD-4, SHA-0, NMAC
- konkrétní hodnoty FAR, FRR různých biometrických technik
- broadcastové autentizační protokoly - TESLA,TIK
- VPN - L2F,PPTP, IP-over-IP, L2TP
- přesnou strukturu zpráv ISAKMP
- AuthIP
- protokol WTLS
- zabezpečení IEEE 802.16 a 802.15

Přehled důležitých věcí, které naopak musíte umět:

Základní cíle informační bezpečnosti

CIA – Confidentiality, Integrity, Availibility (důvěrnost, celistvost, dostupnost)
DOSTUPNOST - AVAILIBILITY
• zajištění, že informace je pro **oprávněné** uživatele přístupná v okamžiku její potřeby
• uvádí se v %
    • např. dostupnost 99,999% znamená výpadek 5 minut za rok
• Kdo je, ale **oprávněný** uživatel?
    • Autentizace
    • Autorizace
AUTENTIZACE - AUTHENTICATION
• Proces ověření identity entity (člověk, program, systém).
• Dvě možné formy
    • Verifikace - entita se aktivně identifikuje, systém pouze potvrdí shodu
    • Identifikace - systém aktivně vyhledá v databázi odpovídající záznam
• Může být vzájemná nebo jednostranná.

AUTORIZACE **–** AUTHORIZATION
- Vždy až po autentizaci
- nejdřív se uživatel musí ověřit a teprve pak zjišťovat, jaká má práva
• Oprávnění přístupu k systémovým zdrojům.
• V průběhu autorizace se určuje k jakým zdrojům má uživatel přístup.

DŮVĚRNOST/UTAJENÍ – CONFIDENTIALITY

• Informace je dosažitelná pouze autorizovaným subjektům. (ti, kteří mají klíč)
• Utajení zajišťují šifrovací (kryptografické) algoritmy

INTEGRITA **–** INTEGRITY

- Schopnost systému zjistit, že se něco změnilo (detekce neautorizované změny)

• Vlastnost systému zajišťující, že přenášená informace nebyla zničena, ztracena nebo modifikována, resp. schopnost detekce takovéto změny.

NEPOPIRATELNOST **–** NON-REPUDATION

• Subjekt nemůže důvěryhodně popřít své minulé požadavky nebo činy.

Prostředky používané k dosažení základních cílů:

• šifrovací algoritmy
• hashovací funkce, MAC/HMAC
• kryptografické protokoly
• časová razítka (známky)
• digitální podpisy

čeho chceme dosáhnout…

Nepodmíněná bezpečnost

• Šifru nelze prolomit bez ohledu na dostupné množství výpočetního výkonu, protože ŠT neposkytuje dostatek informací nutných k jednoznačnému rozpoznání odpovídajícího OT

- chceme dosáhnout

Podmíněná bezpečnost

• Šifru nelze prolomit, protože nemáme k dispozici dostatečné prostředky (čas, výpočetní výkon).

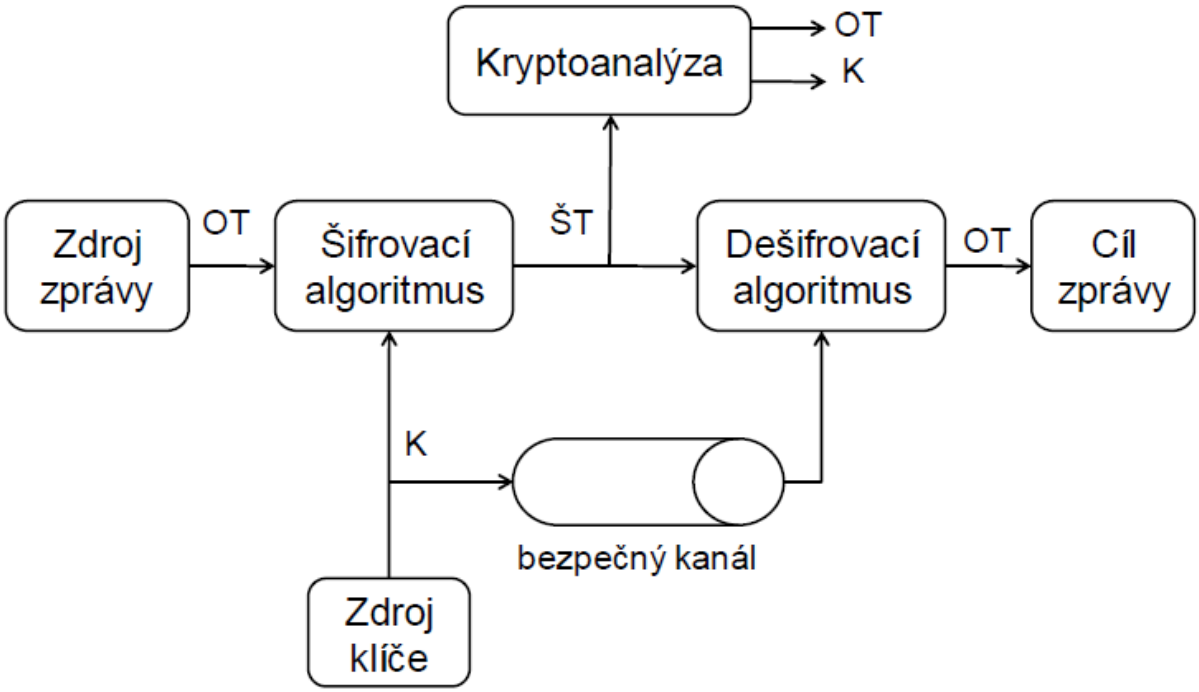
<u>• Prokazatelná bezpečnost</u>

• Problém, na kterém je šifra založena spadá do třídy NP.

<u>• Výpočetní bezpečnost</u>

• Cena za prolomení šifry přesahuje cenu chráněné informace.
• Čas nutný k prolomení šifry přesahuje dobu životnosti chráněné informace.

Shannonův model kryptosystému



Kerchhoffův princip

„Utajení šifrovacího algoritmu nesmí sloužit jako opatření nahrazující nebo garantující kvalitu šifrovacího systému.“

- 1883 Auguste Kerckhoff (holadnský kryptograf)
- Základní předpoklad při konstrukci kryptosystémů
  - útočník zná celý kryptosystém
  - kryptografické algoritmy nejsou tajné, pouze klíč je tajný
- Proč by měl tento předpoklad platit?
  - utajované algoritmy nikdy nezůstanou tajné navždy
  - praxe ukazuje, že v utajovaných algoritmech jsou po odhalení často nalezeny bezpečnostní chyby
  - je lepší odhalit chyby dříve nežli později…

- V dnešní době je většina informací vytvářena, udržována, a přenášena v elektronické podobě.
- Informace mohou být cílem různých útoků, které souvisí s elektronickou povahou dat a proto je přenášená data nezbytné chránit.
- Existuje několik základních cílů, které je potřeba splnit, aby byl systém manipulující s (elektronickými) daty považován za důvěryhodný.

Jaké jsou základní cíle?

Tyto cíle pomáhá plnit vědní disciplína – **kryptografie**.

Kategorie útoků na kryptosystém

- Kryptosystém lze prolomit využitím bezpečnostní chyby v:
  - šifrovacím algoritmu
  - komunikačním protokolu využívajícím daný šifrovací algoritmus
  - schématu pro správu klíčů

útok hrubou silou (brute-force)
  - prohledání celého prostoru klíčů
  - nejjednodušší útok
  - složitost je úměrná množství klíčů
  - předpokládáme, že jsme schopni detekovat nalezení OT
- pendreková (rubber-hose) kryptoanalýza (dokud ho člověk neprozradí)
- korupční kryptoanalýza (uplacení)
- social engineering

kryptoanalýza postranními kanály
- „alternativní“ způsoby útoků na kryptosystémů
- neútočí se na samotný algoritmus, ale jeho fyzickou implementaci
- zachycení informace v průběhu zpracování

Klasické postranní kanály:
  - **Timing analysis** – útok založený na analýze doby trvání různých matematických operací
  - **Power monitoring analysis** – viz předchozí případ, ale sleduje se spotřeba
  - **Radiation monitoring analysis** - sleduje se vyzařování v různých částech E-M spektra
  - **Fault analysis** – získávání informací z chybových hlášení…
- lze realizovat i adaptivní postranní kanály

Kryptoanalýza

- Věda o hledání slabých míst a/nebo prolamování matematických metod informační bezpečnosti
- Cílem je získání OT bez znalosti *klíče* (případně proces získání klíče nebo obojího).

- Kryptosystém lze prolomit využitím bezpečnostní chyby v:
  - šifrovacím algoritmu
  - komunikačním protokolu využívajícím daný šifrovací algoritmus
  - schématu pro správu klíčů

útok hrubou silou (brute-force)
  - prohledání celého prostoru klíčů
  - nejjednodušší útok
  - složitost je úměrná množství klíčů
  - předpokládáme, že jsme schopni detekovat nalezení OT
- pendreková (rubber-hose) kryptoanalýza (dokud ho člověk neprozradí)
- korupční kryptoanalýza (uplacení)
- social engineering

kryptoanalýza postranními kanály

- „alternativní“ způsoby útoků na kryptosystémů
- neútočí se na samotný algoritmus, ale jeho fyzickou implementaci
- zachycení informace v průběhu zpracování

Klasické postranní kanály:

- **Timing analysis** – útok založený na analýze doby trvání různých matematických operací
- **Power monitoring analysis** – viz předchozí případ, ale sleduje se spotřeba
- **Radiation monitoring analysis** - sleduje se vyzařování v různých částech E-M spektra
- **Fault analysis** – získávání informací z chybových hlášení…

- lze realizovat i adaptivní postranní kanály

Základní rozdělení šifer

- **symetrická šifra** / symmetric key cryptosystem – kryptografický algoritmus, který pro šifrování i dešifrování používá tentýž klíč
    - Proudové
    - Blokové
- **asymetrická šifra /** public key cryptosystem – kryptografický algoritmus, který používá dva odlišné klíče, jeden pro šifrování a jeden pro dešifrování
    - Integer Factorization Problem (IFP)
    - Discrete Logarithm Problem (DLP)
    - Elliptic Curve Discrete Logarithm Problem (ECDLP)
    - Další moderní kryptosystémy
        - Kryptografie založená na mřížkách – Lattice based
        - Kryptografie založená na bilineárním párování – Weil pairing
        - Homomorfní šifrování
- **veřejný klíč /** public key – jeden z dvojice klíčů asymetrického šifrovacího algoritmu, obvykle slouží k šifrování a nemusí být utajován
- **soukromý klíč** / private key – druhý z dvojice klíčů asymetrického šifrovacího algoritmu, obvykle slouží k dešifrování a musí být vždy utajován

- Substituční šifry
    - Monoalfabetické šifry
        - Césarova šifra
        - Affiní šifra
    - Polygrafické šifry
    - Polyalfabetické šifry
    - Playfair
    - Jednorázový heslář (One-Time Pad)
- Transpoziční šifry (Permutace)
    - Blokové (sloupcové) transpozice
    - Cardanova mřížka
    - Rail Fence
- Kombinované šifry (product cipher)
    - C. Shannon

- šifra obsahující jak substituční, tak transpoziční část
- větší bezpečnost

Součinové šifry (Product cipher)

- Shannon navrhl metodu vytváření šifer opakovaným skládáním (součinem) několika šifer různých typů
- kombinace různé konfúze a difúze

Iterované šifry

– v moderních algoritmech jedna aplikace konfúze a difúze nestačí (z hlediska bezpečnosti)
– řešení - opakovaná aplikace konfuze a difúze
– cíl - důkladné promíchání bitů
– iterované šifry – runda (stejné dílčí šifry v jednotlivých iteracích)

Vernamova šifra

Vernamova šifra (One-time Pad)

- Gilbert Vernam, 1917
- zaměstnanec AT&T
- jediný absolutně bezpečný kryptosystém
- původní verze pracovala s děrnou páskou
- matematický důkaz provedl C. E. Shannon v roce 1949
- používal se pro zabezpečení horké linky mezi Moskvou a Washingtonem
- problém s generováním a distribucí klíče

One-time Pad - požadavky nutné pro správnou funkci

Klíč je minimálně stejně dlouhý jako přenášená zpráva.

- jiné šifrovací systémy používají kratší klíče, což znamená, že počet možných klíčů je menší než počet možných zpráv
- kratší klíč umožňuje útok hrubou silou

Klíč je dokonale náhodný.

- nelze použít klasické počítačové generátory pseudonáhodných posloupností
- nejvhodnější je užití fyzikálních metod, například tepelného šumu nebo ještě lépe kvanvých procesů (poločas rozpadu atd.)

Klíč nelze použít opakovaně.

Tato podmínka je vychází z předchozí, protože opakovaný klíč není náhodný. Dostane-li útočník do ruky dvě zprávy zašifrované stejným klíčem, má často velmi snadnou cestu k rozluštění.

Klíč zná pouze odesílatel a příjemce.

Šifrový text neposkytuje žádnou informaci o otevřeném textu.

Porušení libovolného z těchto požadavků umožní útočníkovi odhalit tajný text. Při dodržení těchto podmínek nelze takto zašifrovaný text dešifrovat ani útokem hrubou silou. Jeho výsledkem budou všechny možné zprávy dané délky, mezi nimiž nepoznáme tu, která byla odeslána.

Šifrování: Znak otevřeného textu se přičítá na znak hesla pomocí operace XOR

Dešifrování: Znak šifrového textu se přičítá na znak hesla pomocí operace XOR

- Projekt VENONA

Teorie informace

- Teorie informace definuje jak změřit množství informace, obsažené ve zprávě.

- Definice „množství informace“ ve zprávě:
  - Množství informace ve zprávě je dáno minimálním počtem dvojkových signálových prvků (bitů) potřebných k zakódování všech možných stavů této zprávy, za předpokladu, že výskyt každého stavu má stejnou pravděpodobnost.
- Množství informace je vyjádřeno pomocí entropie zprávy.

- entropie jazyka, krytptosystému

Entropie jazyka

• Mějme zdroj zpráv, který generuje zprávy délky k bitů, všechny se stejnou pravděpodobností.
• Zpráv je tedy celkem 2^k a každá má pravděpodobnost 2^{-k}
• Entropie tohoto zdroje je potom

$$\sum_{i=1\ldots 2^k}(-p_i)\log_2(p_i) = 2^k\left(-2^{-k}\right)\log_2(2^{-k}) = k$$

• Každá zpráva z takového zdroje tak nese informaci k bitů.
• Obsažnost jazyka pro zprávy délky N (=množství informace ve zprávě) definujeme jako průměrnou entropii na jeden znak zprávy, tj. jako číslo

$$r_N = \frac{H(M)}{N}$$

Maximální entropie jazyka

• Má-li jazyk zpráv L stejně pravděpodobných znaků a všechny zprávy jsou stejně pravděpodobné, pak entropie zdroje zpráv délky N v tomto jazyce je

$$L^N(-L^{-N})\log_2(L^{-N}) = Nlog_2L$$

a obsažnost jazyka zpráv pro zprávy délky N je tedy $\log_2 L$.
• Toto je teoretická maximální možná obsažnost jazyka o L stejně pravděpodobných znacích.
• Maximální entropie jazyka se označuje R . Přirozený jazyk jí nikdy nedosahuje.
• Max. entropie angličtiny (L=26) pro zprávy délky N je $R = \log_2 2L = \log_2 26$= 4,7bit/symbol

Entropie kryptosystému

$$H(K) = \log_2 K \ \ (K \ldots počet\ klíčů)$$

- Jednotkový odstup (Unity distance): $U = \frac{H(K)}{D}$
- Delší klíč → větší entropie kryptosystému → větší jednotkový odstup → bezpečnější kryptosystém.

$$U = \frac{H(K)}{D} = \log_2\frac{2^{128}}{8-1{,}3} = \frac{128}{6{,}7} = 19{,}7$$

- redundance

Redundance

• Pro rostoucí N obsažnost přirozeného jazyka pro zprávy délky N klesá.
• Pro N→∞ se limitně blíží konstantě r .
• Tuto konstantu **r** nazýváme **průměrná obsažnost (entropie) jazyka** vzhledem k jednomu znaku .
• Udává průměrný počet bitů informace obsaženém v jednom znaku jazyka.
• Průměrná entropie angličtiny je **1,3-1,5** bit/symbol
• Rozdíl **D = R – r** pak nazýváme **nadbytečnost** (redundance) jazyka vzhledem k jednomu znaku(symbolu).

- Poměr **D / R** udává v procentech, kolik bitů jazyka je nadbytečných.
- Delší klíč → větší entropie kryptosystému → větší jednotkový odstup → bezpečnější kryptosystém.

- jednotkový odstup

- Jednotkový odstup (Unity distance): $U = \frac{H(K)}{D}$ (H(K)…entropie kryptosystému, D…redundance jazyka vzhledem k 1 znaku)
- Odhad minimálního množství šifrového textu nutného k tomu, aby výsledkem luštění metodou totálních zkoušek byl pravděpodobně pouze jeden smysluplný otevřený text.
- Delší klíč → větší entropie kryptosystému → větší jednotkový odstup → bezpečnější kryptosystém.

$$U = \frac{H(K)}{D} = \log_2 \frac{2^{128}}{8 - 1{,}3} = \frac{128}{6{,}7} = 19{,}7$$

Teorie složitosti

- zabývá se výpočetními úlohami
- zkoumá jejich řešitelnost a náročnost na zdroje
- rozlišujeme:
  - teorii vypočitatelnosti (má úloha řešení?)
  - teorii výpočetní složitosti (pokud má úloha řešení, jaké má nároky na výpočet?)
- úlohy se ve výpočetní teorii označují jako problémy (problém obchodního cestujícího, zavazadlový problém,…)
- problém je dán množinou zadání, kterých může existovat nekonečně mnoho
- jedno konkrétní zadání označujeme jako instanci

Co můžeme zkoumat

- **výpočet funkce** – má vždy jednoznačný výsledek
- **rozhodovací problémy** – speciální případ výpočtu funkce, kdy nám stačí odpověď ano/ne
- **optimalizační problémy** – hledáme v množině přípustných řešení, takové které splní nějaké kriteriální funkce
- **vyhledávací problémy** – hledáme libovolný objekt, který splňuje nějakou vlastnost

- klasifikace problémů, třídy složitosti

- **výpočet funkce** – má vždy jednoznačný výsledek
- **rozhodovací problémy** – speciální případ výpočtu funkce, kdy nám stačí odpověď ano/ne
- **optimalizační problémy** – hledáme v množině přípustných řešení, takové které splní nějaké kriteriální funkce
- **vyhledávací problémy** – hledáme libovolný objekt, který splňuje nějakou vlastnost

Třída složitosti

- množina problémů s podobnou složitostí

  Definice: Třída složitosti je množina problémů, které můžou být vyřešeny abstraktním strojem M, za použití $O(f(n))$ zdrojů R, kde n je velikost vstupu.
- v současné době je evidováno kolem 450 tříd (BPP, BQP, PSPACE, NPSPACE, EXPTIME, EXPSPACE,…)

Nejdůležitější třídy složitosti

- nejdůležitější je třída P - obsahuje problémy, jejichž řešení je nalezitelné v polynomiálním čase a v praxi zvládnutelné
- zajímavá je třída NP, u které nebylo potvrzeno ani vyvráceno, zda je rovna třídě P

- vztah mezi třídami P a NP a důležitost pro kryptografii

Třída P (PTIME)

- obsahuje zvládnutelné problémy

- existují pro ni dostatečně rychlé, v praxi použitelné algoritmy
- třída P obsahuje problémy, které jsou řešeny deterministickými Turingovi stroji v polynomiálním čase a za použití neomezeného množství paměti

Příklady P problémů

- test prvočíselnosti, hledání společného dělitele

Příklad: Výstupní hodnota logického obvodu

- známe strukturu logického obvodu, hodnoty vstupních proměnných
- v obvodu je jedna výstupní brána
- jaká je výstupní hodnota logického obvodu?

Třída NP (NPTIME)

- obsahuje problémy, které jsou řešeny nedeterministickými Turingovi stroji v polynomiálním čase
- výpočet může větvit do n cest
- při rozvětvení z nichž se každá možnost ověřuje deterministickým Turingovým strojem, když jedna uspěje, pak známe řešení
- cílem je efektivně najít možnou odpověď, která se dá v polynomiálním čase ověřit

Příklady NP úplných problémů

Příklad 1: problém obchodního cestujícího

- zadáno n-měst, vzdálenosti mezi nimi, délka požadované trasy
- lze navštívit všechny města právě jednou, tak aby nebyla překročena délka požadované trasy?

Příklad 2: problém batohu

- zadán objem batohu, objemy předmětů různé velikosti, požadované procento zaplnění
- je možné zaplnit objem batohu z X procent?

Třída co-NP

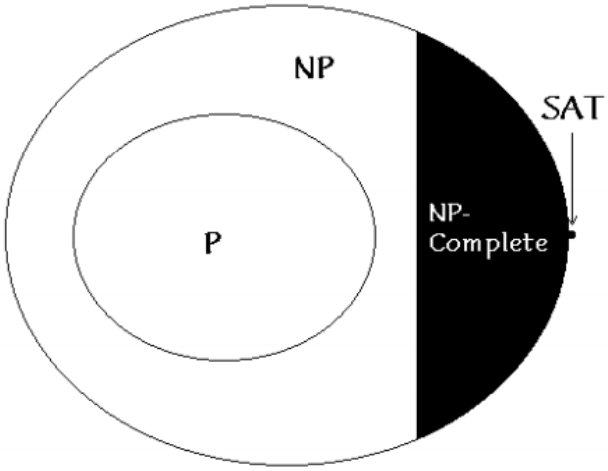
- problém X je členem třídy co-NP pouze tehdy, pokud jeho doplněk leží ve třídě NP
- vztahy k ostatním třídám:
  - P je podmnožinou NP a co-NP, rovnost je málo pravděpodobná
  - NP a co-NP by se rovněž neměly rovnat, což by znamenalo, že NP-úplný nemůže být v co-NP a naopak co-NP-úplný nemůže být v NP

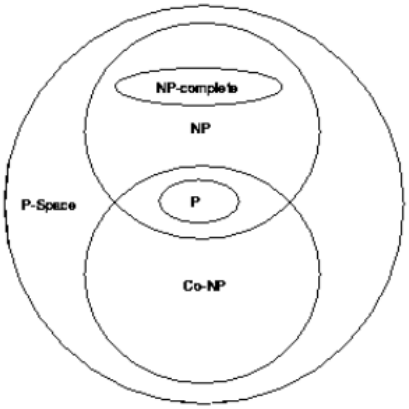
Příklad Co-NP úplného problému:

Určení, zda-li je daná boolovská formule tautologií (tzn. bez ohledu na vstupní kombinací proměnných je výstup pravdivý) .

Obsah - horní část dokumentu

Předpokládané vztahy P a NP k ostatním třídám





SAT - Boolean satisfiability problem (rozhodovací problém, pravdivost logické formule s operátory AND, OR, NOT)

P=NP?

1) Riemmanova hypotéza
2) vztah mezi třídami P a NP
    - pravděpodobně se nerovnají
    - pro vyřešení jsou důležité ty nejtěžší problémy v NP a to NP-úplné
    - nalezení efektivního řešení NP-úplného problému znamená, že P = NP
    - říká se, že „rovnost by měla fatální dopady na kryptografii veřejného klíče“

Steganografie

- obecné principy

- věda o skrývání existence zprávy a nikoliv nutně obsahu vlastní zprávy
- dříve neviditelné inkousty, značky, mikrotečky,…
- steganografická zpráva obecně vypadá jako něco jiného: obrázek, článek, nákupní seznam, nebo nějaká jiná zpráva – krycí text
- v praxi se šifrování a steganografie kombinuje (i když se odhalí přítomnost zprávy, je zpráva pořád nečitelná)
- Steganografická informace vložená do přenášených dat má být vizuálně (a ideálně i statisticky) neodhalitelná.
- Změny v přenášených datech mají být ukryty v šumovém pozdí nosiče.
- Digitální obraz - šum zobrazovacího elementu
- Digitální zvuk - šum z nahrávacího zařízení nebo šum vznikající při digitalizaci.
- Pokud je před digitalizací použit nějaký analogový zesilovač, je v signálu vždy přítomen tepelný šum součástek, který lze opět využít k ukrytí tajné informace.
- Ztrátová komprese (např. JPEG) vždy přináší určitou chybu do dekomprimovaných dat, kterou lze opět využít ke steganografickým účelům.
- neukrytá šifrovaná zpráva bez ohledu na to jak silná šifra je , vzbudí podezření a sama o sobě může být obviňující (protože v některých zemích je používání šifrovacích algoritmů nezákonné.)

- hlavní výhoda steganografie oproti kryptografii spočívá v tom, že zprávy nepřitahují pozornost (sami k sobě, k poslům, nebo k příjemcům).
- Lingvistická steganografie x Technická steganografie

- příklad použití dříve a dnes

Dříve

- dříve neviditelné inkousty, značky, mikrotečky,…
- Řecko: oholit hlavu → napsat zprávu → nechat zarůst
- Čína – zpráva na hedvábí → kulička → vosk → spolknout

Dnes

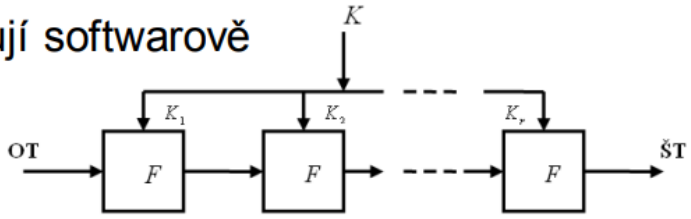
- grafických soborů (JPG, GIF,BMP,…)
- audio souborů (WAV, MP3,…)
- obecně jakýchkoliv binárních formátů (doc, xls)
- nepoužitých sektorů na disketě / HDD
- síťových protokolů (např. TCP, IP,…)
- binárního kódu spustitelných souborů
- HTML kódu
- SPAMu

Iterované blokové šifry

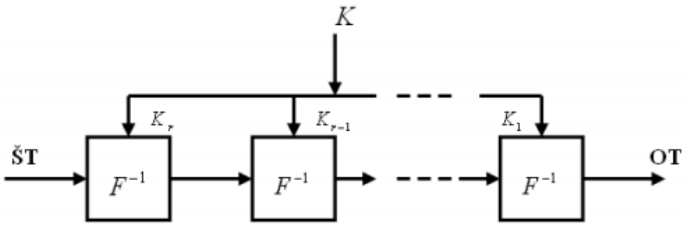
Iterované blokové šifry

- otevřený text i šifrový text mají pevnou délku
- ŠT ze získá z OT pomocí opakované **rundové funkce**
- vstupem do rundové funkce je klíč a výstup z předchozí rundy
- obvykle se implementují softwarově

Šifrování pomocí iterované šifry



Dešifrování iterované šifry

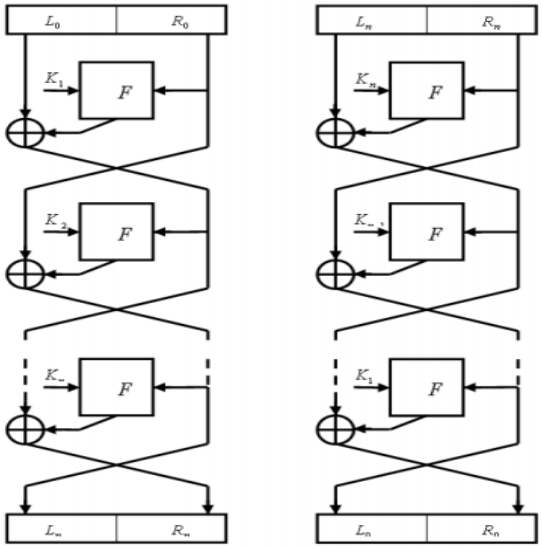


Lavinový efekt

- vhodná (a vyžadovaná) vlastnost šifrovacích algoritmů
- změna jednoho bitu v bloku vstupních dat nebo v klíči vede ke změně přibližně jedné poloviny výstupních bitů
- ztěžuje kryptoanalýzu
- vymyslel Horst Feistel

- Feistelova šifra a EFN

Feistelova šifra



- Feistelova šifra představuje určitý typ blokových šifer
- nejedná se o konkrétní algoritmus
- hodně moderních symetrických blokových šifer má Feistelovu strukturu

Feistelova šifra

- Rozdělíme blok otevřeného textu na levou a pravou část: OT = (L_0, R_0)
- Pro každou rundu $i=1,2,\ldots,n$, spočítáme

  $L_i = R_{i-1}$

  $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$

  kde f je rundová funkce a K_i je podklíč

  ŠT = (L_n, R_n)
- Dešifrování: ŠT = (L_n, R_n)
- V každé rundě i $i=n, n-1, \ldots, 1$, spočteme

  $R_{i-1} = L_i$

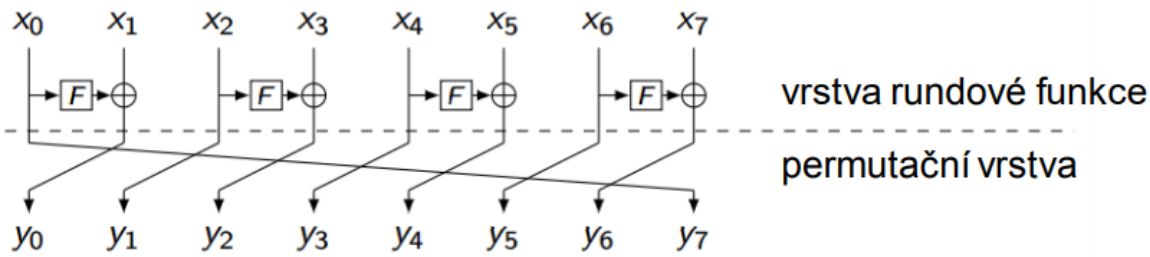
  $L_{i-1} = R_i \oplus f(R_{i-1}, K_i)$

  kde f je rundová funkce a K_i je příslušný podklíč

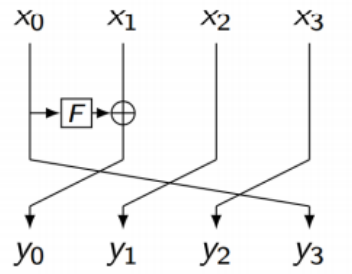
  OT = (L_0, R_0)
- Tento „vzorec“ funguje pro jakoukoliv funkci f()
- Bezpečné šifry jsou, ale pouze pro některé funkce f()

EFN - Zobecněné Feistelovy šifry

- představeny v 1989
- rozdělení bloku OT do $k \geq 2$ n-bitových částí
- obsahuje dvě vrstvy:
  - vrstva rundové funkce
  - permutační vrstva

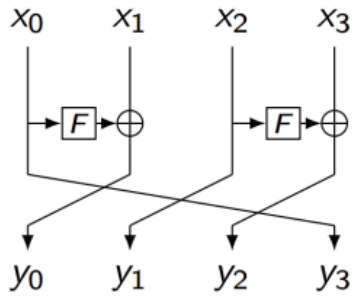


+ vhodné pro malé implementace (velikost podbloku = velikost S-boxu)
- horší difúze mezi podbloky pro rostoucí k



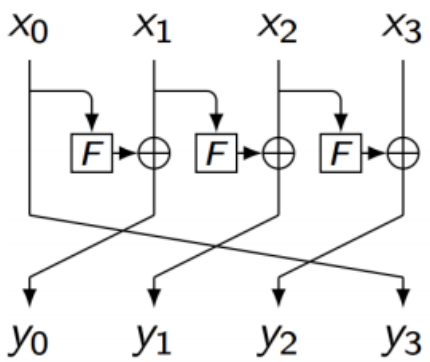
Typ I

- pouze jedna F funkce
- $y_0, y_1, \ldots, y_{n-1}, y_n = x_1 \oplus F(x_0), x_2, \ldots, x_n, x_1$
- Algoritmy: CAST-256, Lesamnta



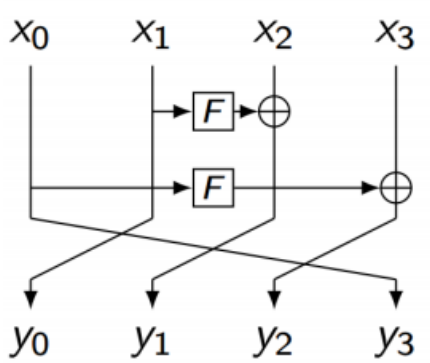
Typ II

- jedna F funkce pro každé dva sousedící bloky
- lepší difúze než Typ I
- $y_0, y_1, \ldots, y_{n-1}, y_n = x_1 \oplus F(x_0), x_2, x_3 \oplus F(x_2), \ldots x_n, x_{n-1} \oplus F(x_0)$
- Algoritmy: HIGHT, CLEFIA



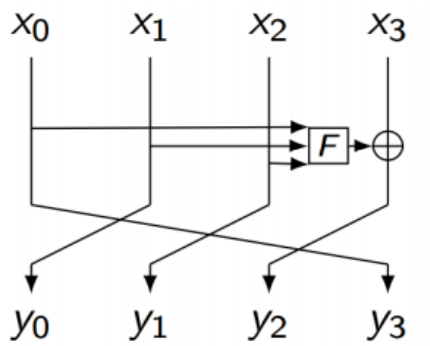
Typ III

- pro každý blok existuje vlastní *F* funkce
- difúze skoro stejně rychlá jako u Typu II
- $y_0, y_1, \ldots, y_{n-1}, y_n = x_1 \oplus F(x_0), x_2 \oplus F(x_3), \ldots x_n \oplus F(x_{n-1}), x_1$



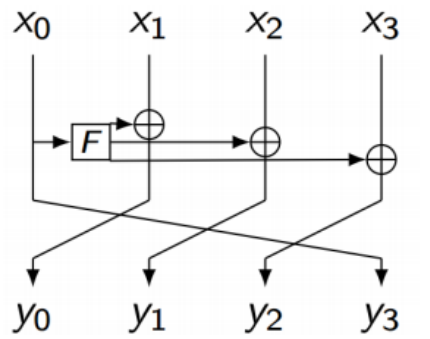
Nybergova EFN

- dvě *F* funkce
- $y_0, y_1, \ldots, y_{n-1}, y_n = x_1, x_2 \oplus F(x_1), x_3 \oplus F(x_0), x_0, \ldots, x_n \oplus F(x_{n-3}), x_{n-3}$



EFN s přetíženým vstupem

- pro každý blok existuje vlastní $y_0, y_1, \ldots, y_{n-1}, y_n = x_1, x_2, x_3 \oplus F(x_0, x_1, x_2), x_0, \ldots, x_n \oplus F(x_{n-3}, x_{n-2}, x_{n-1}), x_{n-3}$
- Algoritmy: MD-x, SHA-1,2

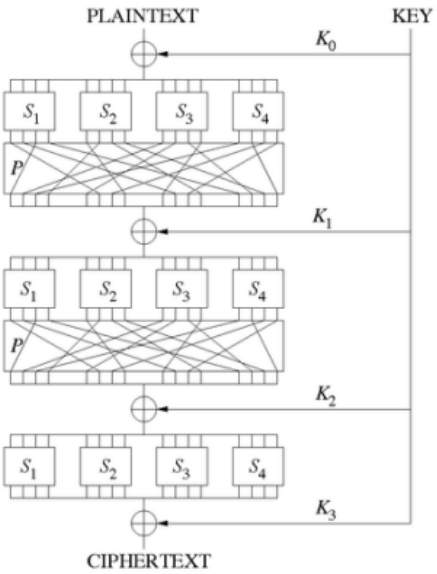


EFN s přetíženým výstupem

- $y_0, y_1, \ldots, y_{n-1}, y_n = x_1 \oplus F(x_0), x_2 \oplus F(x_0), x_3 \oplus F(x_0), x_0 \ldots, x_{n-2} \oplus F(x_{n-3}), x_{n-3}$
- Algoritmy: MARS

SP (Substitučně-permutační) síť

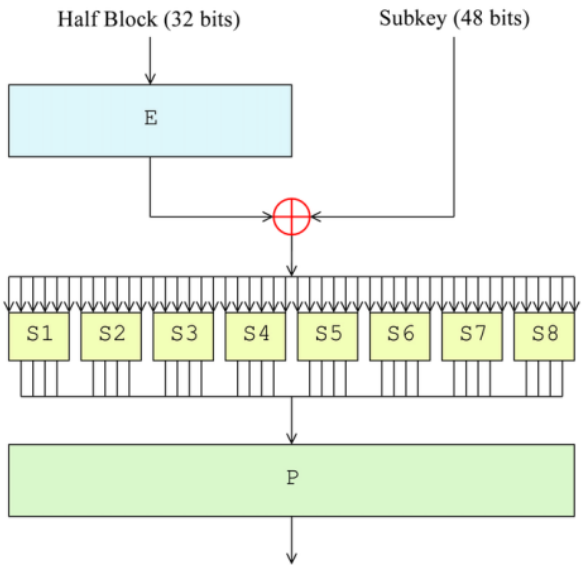
- SPN – Substitution-Permutation Network
- AES
- 2vrstvy
  - S-boxy
    - symetrické (1:1); lavinový efekt
  - P-box
    - výstupní bity z jednoho S-boxu jsou přivedeny na vstup co nejvíce S-boxů



- DES - včetně opearací v rundě

- šifra Feistela typu
- délka bloku 64 bitů
- délka klíče 56 bitů
- 2 56 = 72,057,594,037,927,900
- 16 rund
- v každé rundě je použito jiných 48 bitů klíčů (podklíč, rundový klíč)
- v každé rundě se s blokem vykonávají stejné jednoduché operace
- bezpečnost DESu závisí na konstrukci “S-boxu”
- z pohledu bezpečnosti nejdůležitější část
- jediný nelineární prvek v DESu

DES - operace v rundě – 8 paralelních S-boxů



DES S-box č.1

- 8 různých S-boxů
- 6 vstupů, 4 výstupy – nelineární operace
- první a poslední bit vstupního řetězce určují řádek
- vnitřní čtyři bity určují sloupec
- v průsečíku se nachází výsledek Příklad: na vstupu je hodnota **1**0001**1** zvolíme řádek **11**, sloupec 0001 → průsečík 12 (1100)

Bezpečnost DESu

- nejjednodušší útok
- hrubá síla
- najít správný klíč z celkem 2^{56} klíčů
- DES je komplementární → stačí uhodnout pouze 2^{55} (zmenšení problému na polovinu)
  - TMTO (Time/Memory TradeOff) – 1TB paměti, 5 dní výpočtů v 1980
- **bezpečnost DESu závisí především na S-boxech**
  - vše ostatní v DESu jsou lineární operace
  - lineární = „lehce“ odstranitelné

- 3DES

Triple DES (3-DES)

- Klíč o délce 56 bitů přestal v 90. letech postačovat
- DES byl široce rozšířen a nešlo ho rychle nahradit něčím zcela novým
- 3-DES-EDE2 byl standardizován v normách ANSI X9.17 a ISO 8732.
- **Délka klíče** 3DESu je **168 bitů** (3x 56 bitů), ale díky možnosti tzv. meet-in-the-middle útoku (luštění současně z obou stran) je efektivní délka pouze 112 bitů
- **Nejčastější varianta: Triple DES-EDE2**
  - C = E(D(E(P,K1),K2),K1)
  - P = D(E(D(C,K1),K2),K1)
- **Proč používat 3DES-EDE se 2 klíči?**
  - Zpětná kompatibilita: E(D(E(P,K),K),K) = E(P,K)
  - Klíč délky 112 bitů stačí
- **Praktické použití 3DES dnes**
  - PKCS#8 - šifrované soubory s privátními klíči (součást .pfx/.p12 blobů)
  - 3DES-EDE3-CBC

- AES (Rijndael) – všechno

- volba nového algoritmu AES (konec 90. let)
- nástupce DESu
- NIST vyhlásil 2.1.1997 plně veřejnou a otevřenou soutěž s cílem najít silnou šifra pro vládní i komerční použití
  - bloková symetrická šifra
  - otevřený algoritmus, nechráněný patenty
  - bezpečnost algoritmu je důležitější než jeho rychlost

AES – hodnotící kritéria

Bezpečnost	Cena	Algoritmické & implementační charakteristiky
• Skutečná bezpečnost ve srovnání s ostatními soutěžícími algoritmy • Kvalita výstupního ŠT - výstup musí být nerozeznatelný od náhodné permutace stejného vstupního bloku • Bezpečnost algoritmu musí být podložena solidními matematickými základy • Jiné bezpečnostní otázky vzešlé od (odborné) veřejnosti, včetně praktické demonstrace odolnosti vůči kryptoanalytickým útokům	• Licenční požadavky – celosvětově dostupný, neexklusivní licence, poskytovaný zdarma (royalty-free basis) • Licenční požadavky se musí týkat jak HW, tak SW implementace; rychlost algoritmů pod vybranými platformami • Pamětové požadavky – týká se jak HW tak SW implementace; roli hraje např. počet hradel, velikost kódu, požadavky na RAM	• Flexibilita – schopnost pracovat s delšími klíči bloky dat (klíče v rozsahu 128-256bitů po 32bitových krocích, bloky dat po 64 bitech) • Schopnost bezpečné a efektivní implementace v širokém spektru prostředí a aplikací (8bitové procesory, bankomaty, sítě, hlasová a satelitní komunikace, HDTV) • Možnost implementace AES jako proudové šifry, generátoru MAC, generátoru PRN, … • HW a SW přiměřenost • Jednoduchost návrhu

- Ostatní kandidáti: Twofish, Serpent, RC6,…

Rijndael (AES) Autoři: Vincent Rijmen a Joan Daemen

• Belgie
• iterovaná bloková šifra (stejně jako DES)
• není šifra Feistelova typu (na rozdíl od DESu)
• substitučně-permutační síť
• veškeré matematické operace v AESu se odehrávají v konečném poli F_2^8 s nerozložitelným polynomem $F(x)= x^8 + x^4 + x^3 + x + 1$
• **Délka bloku**: 128* , 192 nebo 256 bitů
• **Délka klíče**: 128* , 192 nebo 256 bitů (*v AES standardu)
    • nezávisle na délce bloku
• 10 až 14 rund (v závislosti na délce klíče a bloku)

AES operace v rundě

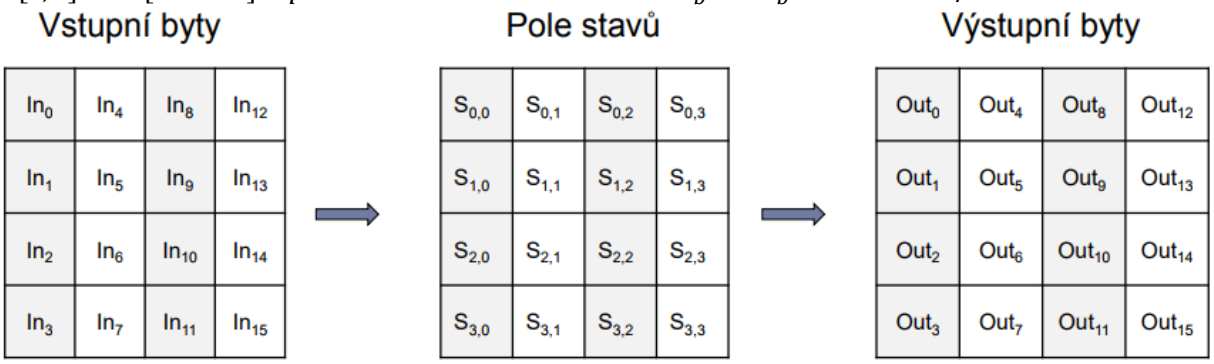
• v každé (kromě poslední) rundě se provádějí čtyři operace:
    • **ByteSub** (nelineární operace)
    • **ShiftRow** (lineární operace)
    • **MixColumns** (nelineární operace)
    • **AddRoundKey** (lineární operace)

Nr	N_b=4	N_b=6	N_b=8
N_k=4	10	12	14
N_k=6	12	12	14
N_k=8	14	14	14

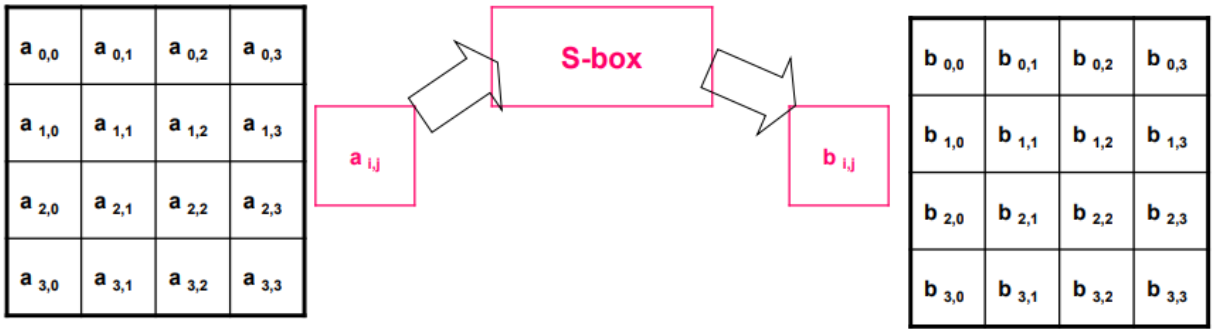
• všechny operace v AES se provádějí na 2-D poli označovaném jako Stav (State)
• pole má vždy 4 řádky a 4 reps.6 resp. 8 sloupců

- počet sloupců závisí na velikosti bloku
- každá buňka pole obsahuje 1 byte dat
- celková velikost stavu je 128/192/256 bitů

$S[r,c] = in[r + 4c] \quad pro\ 0 \leq r < 4 \ \ a\ \ 0 \leq c < N_b$ $\quad N_b$ = délka bloku/32



Rijndael (AES) – ByteSub



ByteSub - Cíl: vnesení nelinearity do procesu šifrování

ByteSub je nelineární operace ve dvou krocích:

1) na každý byte se v F_2^8 aplikuje multiplikativní inverze

2) na každý byte se aplikuje affiní transformace (nad F_2) ve tvaru

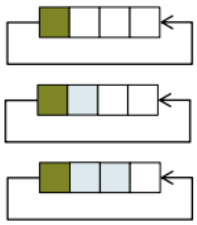
$$8F \bullet a_{i,j} \oplus A6$$

- operace SubByte má v AESu stejný význam jako „S-box“ v DESu
- může být implementován jako tabulka pro každý byte
  - Pro operaci ByteSub – tabulka (máme S_{in}={7b} → odpovídající hodnota (pro průsečík *7* a *b*))

Rijndael (AES) – ShiftRow

$b_{0,0}$	$b_{0,1}$	$b_{0,2}$	$b_{0,3}$
$b_{1,0}$	$b_{1,1}$	$b_{1,2}$	$b_{1,3}$
$b_{2,0}$	$b_{2,1}$	$b_{2,2}$	$b_{2,3}$
$b_{3,0}$	$b_{3,1}$	$b_{3,2}$	$b_{3,3}$

žádná rotace



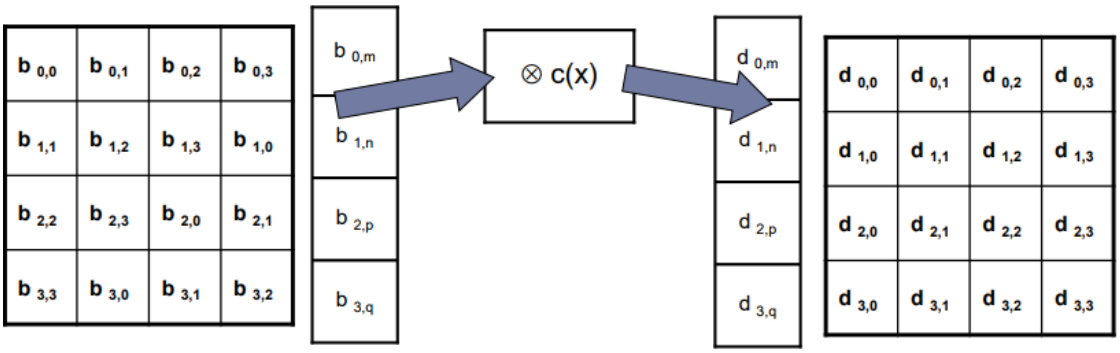
$b_{0,0}$	$b_{0,1}$	$b_{0,2}$	$b_{0,3}$
$b_{1,1}$	$b_{1,2}$	$b_{1,3}$	$b_{1,0}$
$b_{2,2}$	$b_{2,3}$	$b_{2,0}$	$b_{2,1}$
$b_{3,3}$	$b_{3,0}$	$b_{3,1}$	$b_{3,2}$

Operace ShiftRow se provádí na jednotlivých řádcích.

Cíl: difúze mezi sloupci

	C_0	C_1	C_2	C_3
$N_b = 4$	0	1	2	3
$N_b = 6$	0	1	2	3
$N_b = 8$	0	1	3	4

AES – MixColumns



- operace MixColumn pracuje se sloupci
- každý sloupec se uvažuje jako polynom nad F_2^8 a je vynásoben s polynomem $c(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$ mod x^4+1
- Implementuje se pomocí XOR

Cíl: zajištění difúze mezi jednotlivými byty.
Společně s ShiftRow zajistí tzv. lavinovitý efekt
Koeficienty matice byly zvoleny také s ohledem na možnost efektivní implementace.

$$c(x) = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

Obsah - horní část dokumentu

AES – AddRoundKey

$d_{0,0}$	$d_{0,1}$	$d_{0,2}$	$d_{0,3}$
$d_{1,0}$	$d_{1,1}$	$d_{1,2}$	$d_{1,3}$
$d_{2,0}$	$d_{2,1}$	$d_{2,2}$	$d_{2,3}$
$d_{3,0}$	$d_{3,1}$	$d_{3,2}$	$d_{3,3}$

$\oplus$

$k_{0,0}$	$k_{0,1}$	$k_{0,2}$	$k_{0,3}$
$k_{1,0}$	$k_{1,1}$	$k_{1,2}$	$k_{1,3}$
$k_{2,0}$	$k_{2,1}$	$k_{2,2}$	$k_{2,3}$
$k_{3,0}$	$k_{3,1}$	$k_{3,2}$	$k_{3,3}$

$=$

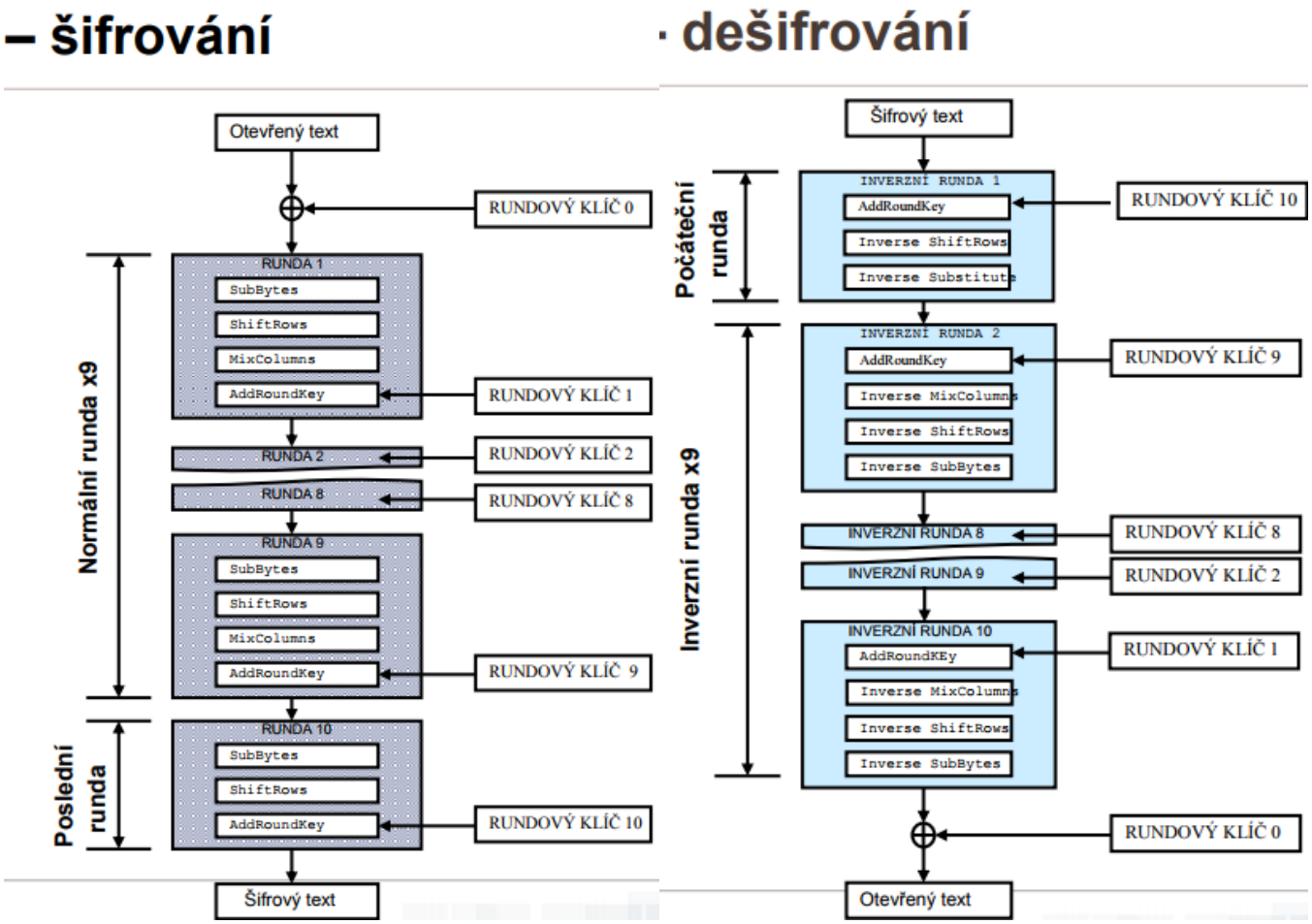
$e_{0,0}$	$e_{0,1}$	$e_{0,2}$	$e_{0,3}$
$e_{1,0}$	$e_{1,1}$	$e_{1,2}$	$e_{1,3}$
$e_{2,0}$	$e_{2,1}$	$e_{2,2}$	$e_{2,3}$
$e_{3,0}$	$e_{3,1}$	$e_{3,2}$	$e_{3,3}$

- Operace AddRoundKey provádí přičtení rundového klíče ke Stavu.
- Klíč rundy je určen pomocí plánovacího algoritmu (key schedule).

Cíl: operace v rundě musí být klíčově závislé.

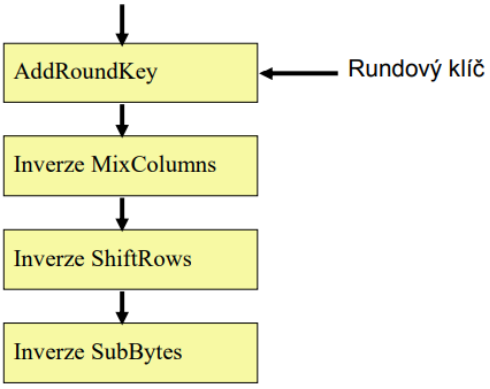
- XOR OT (nebo ŠT) a klíče se nazývá bílení (whitening) klíče. Je to jednoduchý postup zvyšující bezpečnost. Brání útočníkovi vytvářet odpovídající páry OT-ŠT. U **Rijndaelu** je realizován **před první rundou**.

AES – šifrování



AES - dešifrování

Při dešifrování probíhají **inverze jednotlivých operací** v opačném pořadí než při šifrování:



- **inverzní operace** se provádějí v **opačném pořadí** než při šifrování
- kromě nelineární operace SubBytes je inverze zbylých operací velmi jednoduchá
- Rijndael je navržen tak, že lze použít stejný kód na šifrování i dešifrování
- pouze se zamění příslušné tabulky a polynomy (v každém ze 4 kroků)
- zbytek operací probíhá jako při šifrování

AES – dešifrování jednotlivých operací

- Operace AddRoundKey je invertibilní
  - operace ⊕ je inverzní sama k sobě tzn. po provedení opětovného přičtení polynomu mod 2 dostaneme původní polynom
- MixColumn je invertibilní
  - inverzi se realizuje pomocí násobení inverzním polynomem

$c(x) = \{0b\}x^3 + \{0d\}x^2 + \{09\}x + \{0e\}$

- ShiftRow je invertibilní
  - Inverze se realizuje jako cyklický posun doleva
- Nelineární operace ByteSub je také vratná
  - inverze je implementována pomocí vyhledávání v tabulce

AES - optimalizace

- Určeno pro systémy pracující s 32 bitovým vstupem (nebo větším).
- Urychlení algoritmu zkombinováním operací SubBytes a ShiftRows spolu s MixColumns a jejich transformací do několika vyhledání v tabulkách.
- Je nutné sestavit čtyři tabulky s rozměry 16x16, což zabere celkem 4x16x16x8=4096 bytů paměti.
- Rundu pak můžeme realizovat pomocí 16 vyhledání v tabulce a 12 32bitových operací XOR
- Poté následují čtyři 32bitové operace XOR s klíčem (operace AddRoundKey)

AES – bezpečnost

AES-128

- 2011 – útok na plný AES se složitostí $2^{126.1}$ kroků
- 4x efektivnější než útok hrubou silou

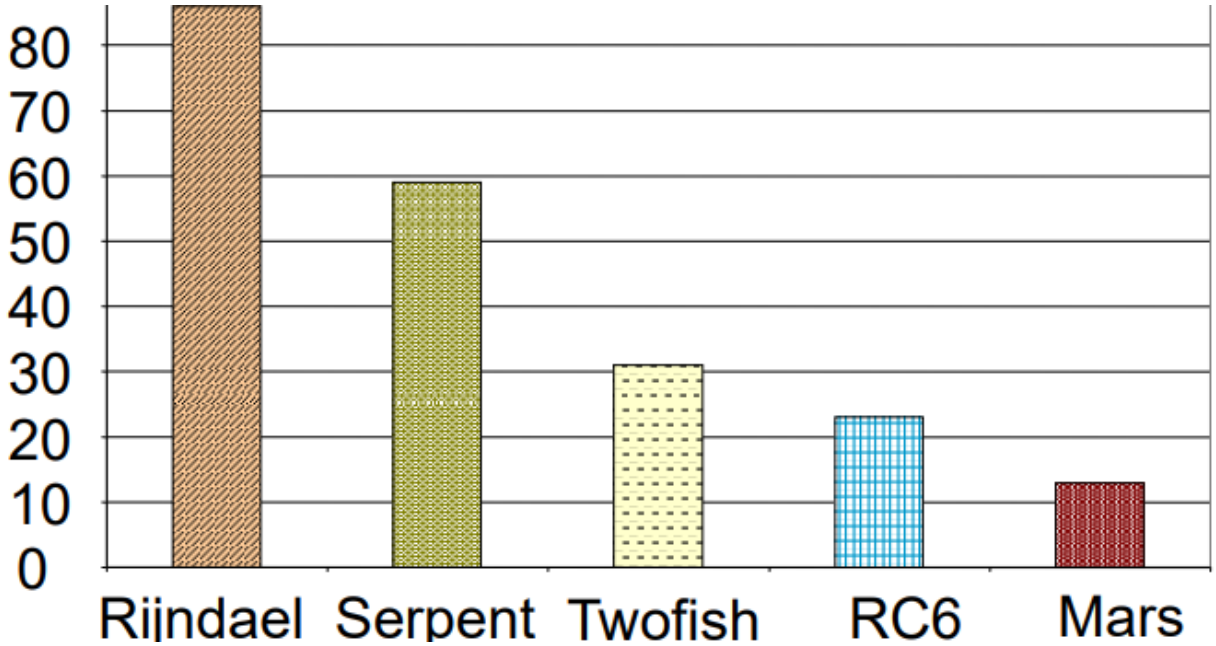
AES-192, AES-256

- popsány útoky s časovou složitostí 2^{172} a 2^{119} kroků
- funguje pouze se speciálním typem útoků - „related key“ útok

- kryptoanalytik musí mít k dispozici OT zašifrovaném mnoha klíči, mezi kterými je vhodná vazba
- týká se jen AES-256 s 10 rundami a AES-192 s 9 rundami
- princip TMTO (Time Memory TradeOff) - úspora v času znamená zvýšené nároky na pamět
- prostorová složitost útoku na AES-256 je 2^{119}
- nelze prakticky zrealizovat

- 2011 – útoky se složitostí AES-256, $2^{189.7}$ and $2^{254.4}$

- finalisté na AES - základní charaktetistiky + poznat algoritmus podle jeho vnitřní struktury



Obecné charakteristiky finalistů

• všech 5 šifer jsou iterované blokové šifry

• všech 5 finalistů používá whiteningu (bělení)

• promíchání klíče a vstupních/výstupních dat

• 4 finalisté (kromě RC6) používají S-boxy (nelineární substituční funkce)

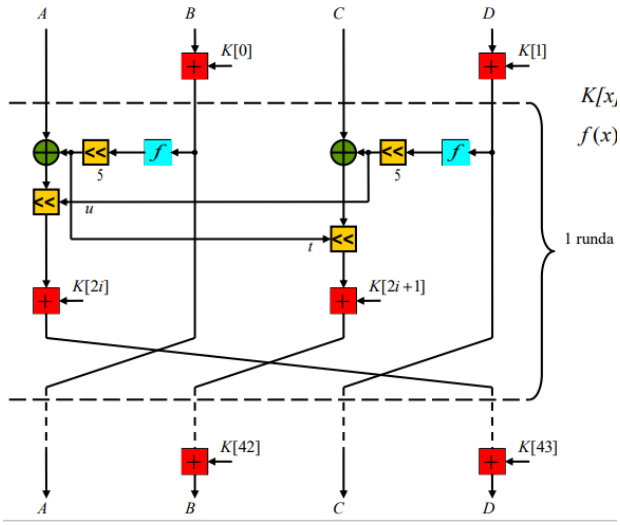
• 3 finalisté (MARS, Twofish, RC6) používají Feistelovo schéma

• 2 finalisté (Rijndael, Serpent) používají substitučně- permutační sítě (zpracovávají paralelně vstupní blok sérií substitucí a lineárních transformací)

Kandidáti na AES – MARS

- E-funkce a S-Box (512 32b čísel)
- Dešifrování probíhá stejně jako šifrování
- **Bezpečnost MARSu**
    - • MARS jediný používá dvě nelineární funkce (S-boxy a datově závislé rotace)
    - • tento fakt spolu s heterogenní strukturou (16 šifrovacích rund a 16 mixovacích rund) zajišťuje větší složitost šifry než u zbývajících kandidátů
    - • toto je ale zároveň i nevýhoda MARSu
    - • nejlepší známý útok na MARS předvedl B. Schneier v roce 2000 kdy se mu pomocí útoku se známým OT podařilo prolomit oslabený MARS (pouze 21 z 32 rund z čehož 16 bylo mixovacích (=výrazně jednodušších))

Kandidáti na AES – RC6 – 32/20/16



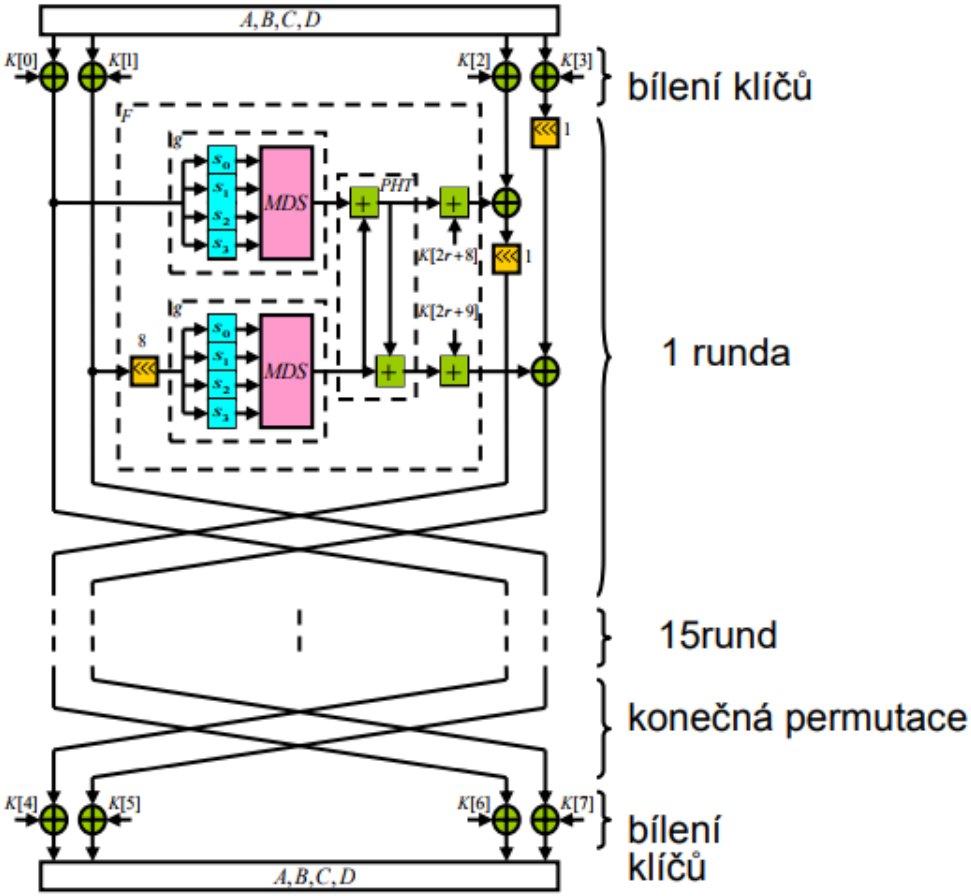
- • RC6 má několik volitelných parametrů:
    - **w** - počet bitů slova
    - **r** - počet rund
    - **b** - počet bajtů klíče, proto se podle nich přesně označuje jako **RC6-w/r/b**.
- Pro kandidaturu na AES w=32b, r=20, b=16B, 24B nebo 32B – blok 128b a klíč 128, 192 nebo 256b

- vychází ze starší šifry RC5
    - RC6 = 2 paralelně propojené šifry RC5
- šifra Feistelova typu
- vstup/výstup: 4x32 bitů
    - v každé rundě:
    - sčítání $mod 2^{32}$
    - násobení 2^{32}
        - tato operace není v RC5, zajistí, že rotace bude záviset na každém bitu slova B resp. D - viz následující slide
    - XOR
    - přičítání klíčů
- nepoužívá S-boxy, ale datově závislé rotace
- existují útoky na zjednodušené verze (15 rund)
- pro deklarovaný počet rund 20 je bezpečný

Bezpečnost RC6

- bez ohledu na jednoduchost designu je RC přiměřeně odolná známým útokům
- není znám žádný útok na 20 rundovou variantu, přestože pro některé autory je počet rund nedostatečný
- pomocí lineární a diferenciální krypotanalýzy je možné prolomit 12 rundovou verzi
- statistické útoky založené na vybraných dvojicích OT-ŠT (choosen plain-ciphertext) prokázaly zranitelnost až do 13 rund

Kandidáti na AES – Twofish



- klasické Feistelovo schéma (jako DES)

- 16 rund
- klíč délky 128 až 256 bitů
- operace podobné jako v Rijndaelu
    - násobení v konečném poli F_2^8
    - sčítání $mod2^{32}$
    - XOR
    - klíčově závislé S-boxy
- **vysoká roveň bezpečnosti, ale velmi složitý návrh**
- podobně jako u Rijndaelu výkon klesá s délkou klíče

Operace v rundě

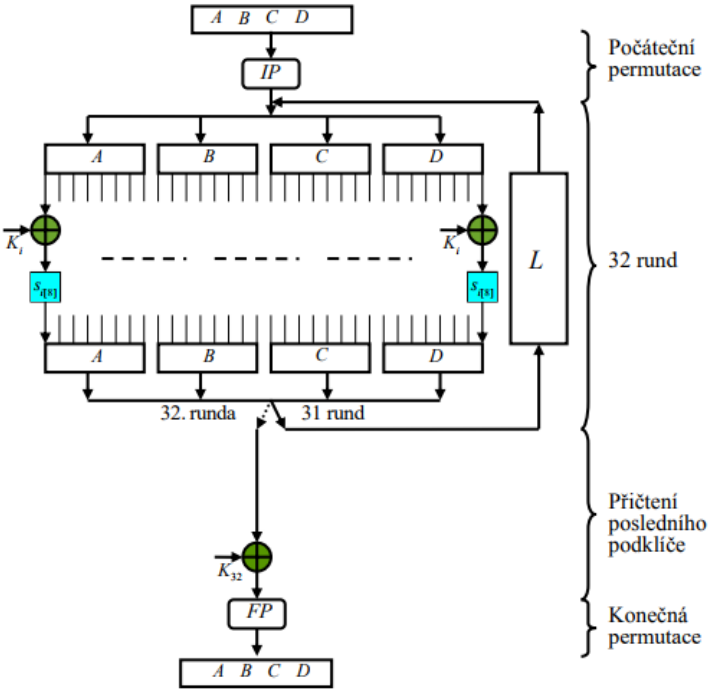
- 4 klíčově závislé S-boxy
- bitové rotace
- PHT - Pseudo-Hadamardova transformace
    - jednoduché míchání dvou vstupů, podle vzorce:

$$a' = a + b\, mod2^{32}, b' = a + 2b\, mod2^{32}$$

    - realizuje difúzi
- polovina klíče je použita na šifrování a polovina modifikuje algoritmus (S-boxy)

- z hlediska bezpečnosti je nejdůležitější funkce g
- vstupní 32bitové slovo je rozděleno na čtyři části
- každá čtvrtina vstupuje do jedné skupiny S-boxů
- každý S-box má 8 bitový vstup a výstup
- čtyři výstupy z S-boxů jsou chápány jako vektor v F_2^8 , který je vynásoben maticí MDS
- následuje přičtení rundových klíčů
- výsledek je interpretován jako 32bitové slovo

Kandidáti na AES – Serpent



- navržen pro co nejvyšší bezpečnost
- odolný vůči všem dnes známým útokům

- 32 rund (velmi bezpečná, ale pomalá)
- **není** to šifra **Feistelova** typu
- substitučně-lineární transformační síť
- jako Rijndael
- délka bloku 128 bitů (vstup/výstup 4x32 bitů)
- klíč může mát libovolnou délku do 256 bitů
- velmi konzervativní návrh
- nepoužívá
    - datově závislé rotace, ani
    - násobení mod n, ani
    - sčítání mod n
- používá „tradiční“ operace
    - XOR
    - S-boxy
- vhodné pro čipové karty
- z počátečního klíče se spočítá 33 rundových klíčů
- pokud je klíč k < 256b, je doplněn jednou „1“ a více „0“ na celkovou délku 256b

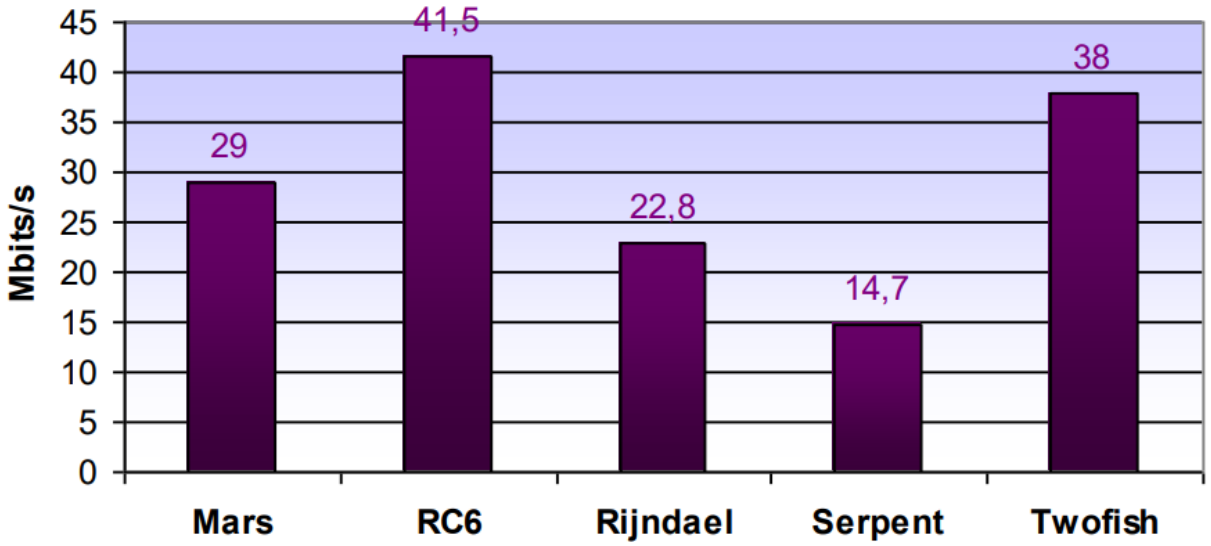
- počáteční permutace (mění pořadí bitů v bloku)
- **32 rund**, každá obsahuje:
    - XOR s rundovým klíčem
    - průchod S-boxem
    - lineární transformace
- konečná permutace, která je inverzí k počáteční

První a poslední krok nemají žádný význam z kryptografického hlediska. Slouží pouze k optimalizaci dat a zvyšují efektivitu výpočtů.

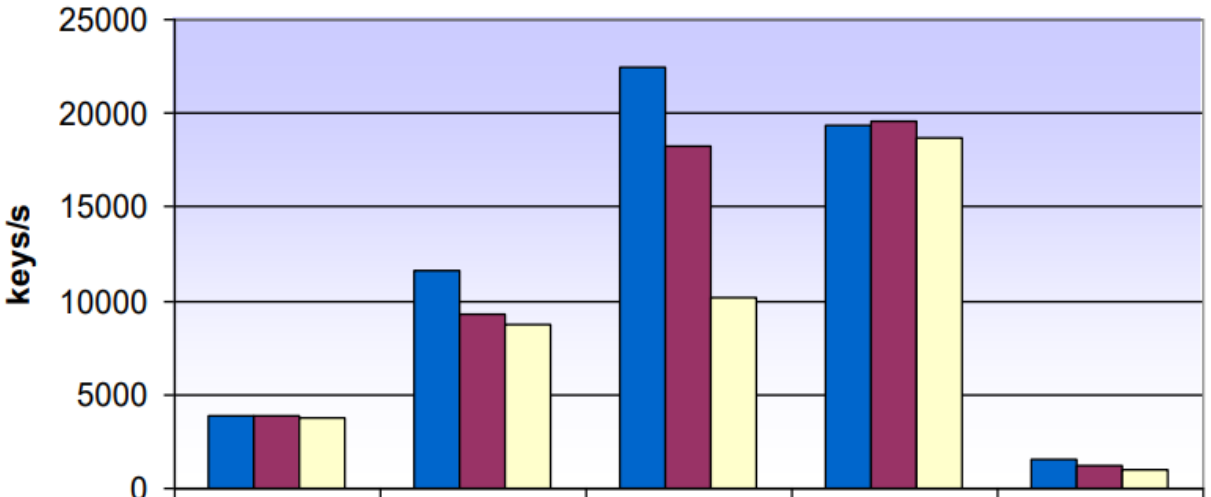
Bezpečnost Serpentu

- Již s 16 rundami je Serpent dostatečně odolný proti všem dnes známým útokům.
- Zvýšení počtu rund na 32 dále zvětšuje celkovou bezpečnost šifry.
- Pomocí diferenciální kryptoanalýzy a útoků se znalostí vybraných OT se povedlo prolomit Serpent s 6 rundami.

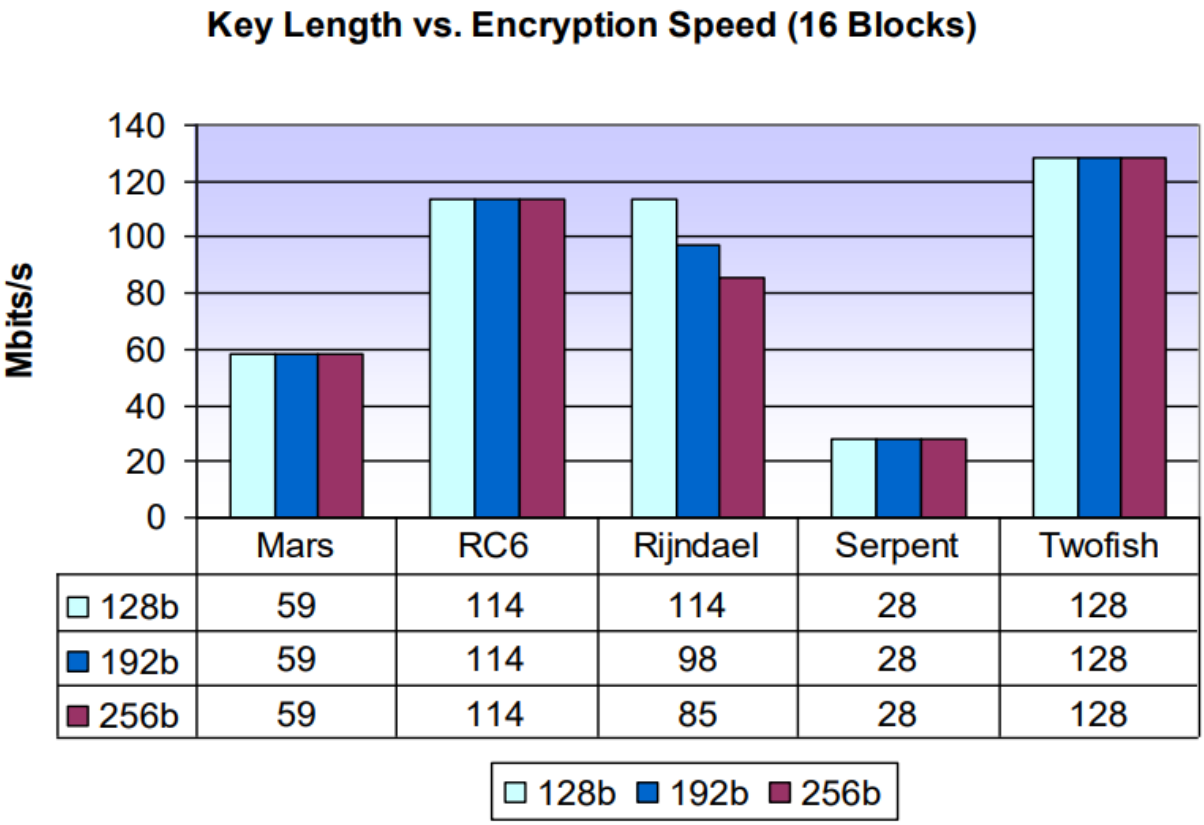
Výkon algoritmů implementovaných pomocí jazyka C na referenční platformě (Pentium 200MHz)



Timing test: Key Expansion



	Mars	RC6	Rijndael	Serpent	Twofish
128b	3878	11666	22434	19339	1558
192b	3896	9276	18258	19530	1237
256b	3789	8780	10197	18706	984



	Mars	RC6	Rijndael	Serpent	Twofish
128b	59	114	114	28	128
192b	59	114	98	28	128
256b	59	114	85	28	128

- generování rundových klíčů - pouze DES,AES

Lineární a diferenciální kryptoanalýza - základní principy

Diferenciální kryptoanalýza

- varianta útoku se znalostí vybraných OT
- jeden z nejdůležitějších pokroků (zveřejněných) v moderní kryptoanalýze
- znám NSA již v 70.letech v době návrhu DESu
- publikován v 1990 - Murphy, Biham & Shamir
- mocný nástroj pro analýzu blokových šifer
- v současnosti se používá k analýze moderních blokových šifer (s různou mírou úspěchu)
- DES je odolný vůči DK (na rozdíl od Luciferu)
    - statistický útok na šifry Feistelova typu
    - předpoklad: máme k dispozici dvojice OT + k němu příslušný ŠT
    - rozdíl mezi $\mathbf{d_P=P_1 \oplus P_2}$, $\mathbf{d_C=C_1 \oplus C_2}$
    - vztah mezi d_C a d_P může odhalit informace o klíči
    - abychom získali dobré diference d_P , je potřeba **mít k dispozici mnoho párů d_C a d_P**
    - D.K. umožní najít **některé** bity klíče, zbytek se získá **hrubou** silou
- Změna problému z „Jaký klíč vygeneruje pár (OT0 , ŠT0)?" na „Jaká množina klíčů může vyvolat změnu ŠT0 na ŠT1 změnou jednoho bitu v OT0?“
- Útok pomocí diferenciální kryptoanalýzy na DES s 8 rundami vyžaduje:
    - 2^{14} = 16,384 vybraných OT, nebo

- 2^{38} známých párů OT-ŠT
- Útok na DES se 16 rundami vyžaduje:
  - 2^{47} vybraných OT, nebo
  - přibližně $2^{55.1}$ známých párů OT-ŠT
- diferenciální kryptoanalýza není příliš efektivní!!!

Lineární kryptoanalýza

- statistická metoda (stejně jako DK)
- analýza vnitřní struktury algoritmu
- náhrada celé šifry (velkou) množinou lineárních rovnic
- DES je lineární až na S-boxy
- Jak aproximovat S-box lineární funkcí ?
  - Těžko – neexistuje dobrá lineární aproximace žádného jednoho konkrétního bitu na výstupu, **ALE**
  - **existují** lineární kombinace výstupů, které lze aproximovat pomocí lineárních kombinací vstupů
- DES není optimalizován proti této technice
  - ale i tak to vyžaduje velké množství dvojic OT…
  - odhad části klíče a zbytek se spočte hrubou silou L

Proudové šifry - synchronní, asynchronní

- výměna prokazatelné bezpečnosti za jednoduchou realizaci (konstrukci)
- proudové šifry jsou inicializovány krátkým klíčem (**seed**)
- klíč je „natažen“ do dlouhého proudu klíče (**keystream**)
- proud klíče se používá stejně jako one-time pad
  - XOR s OT / ŠT
- proudová šifra je generátor proudu klíče
  - obvykle generuje proud klíče po bitech, někdy bajtech

Shanonnova podmínka nepodmíněné bezpečnosti:

- entropie klíče > entropie zprávy
  - $H(\kappa) \geq H(M)$
- toto je omezující podmínka použití proudových šifer
  - nelze ji dodržet
  - pro šifry s $H(\kappa) << H(M)$ – výpočetní složitost
- další problém – distribuce klíčů

Posuvné registry

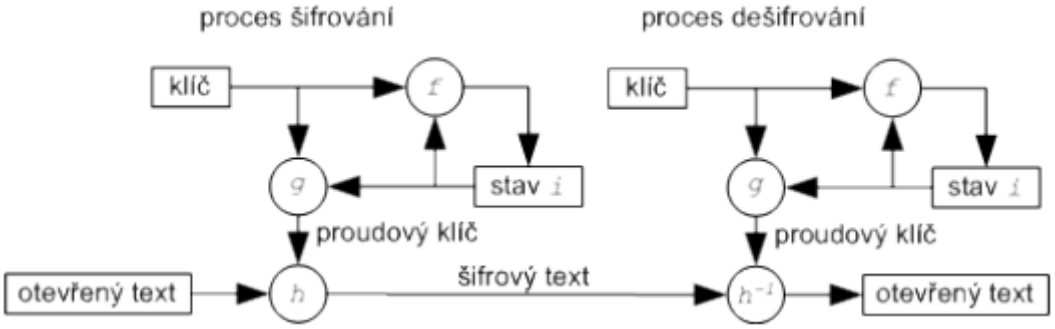
- Klasické proudové šifry byly založeny na posuvných registrech (shift register)
- posuvný registr obsahuje
  - množina stavů každý obsahující jeden bit
  - zpětnovazební funkci
- dnes se využívají i jiné přístupy:
  - LFSR (jednoduchá konstrukce, dobré statistické vlastnosti)
    - Linear Feedback Shift Register
    - jeden ze základních bloků proudových šifer
    - nutno kombinovat s nějakou nelineární části
    - n-bitový čítač vykazující náhodné chování

- Pokud je polynom C(x) primitivní, má výstupní posloupnost LFSR periodu T = 2^L -1.
- Taková výstupní posloupnost má dobré statistické vlastnosti, ale je předvídatelná.

- S-boxy (vnášejí nelinearitu),
- Booleovy funkce (nelinearita a zvýšení lineární složitosti),
- sčítání mod 2n (nelinearity a rušení asociativity)

Synchronní proudová šifra

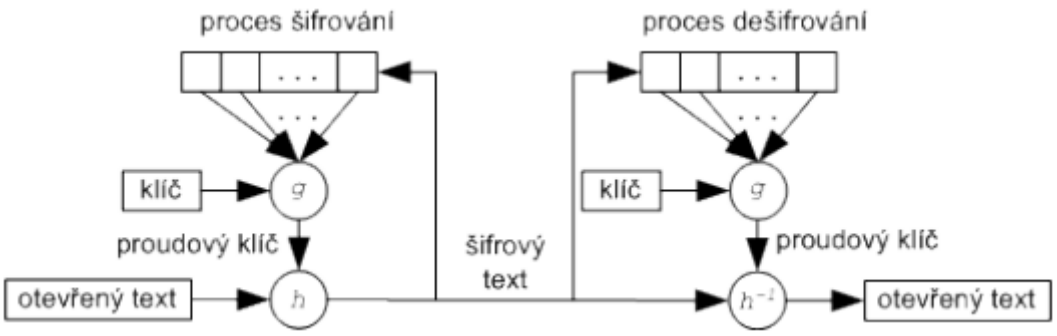
- SSC – Synchronous Stream Cipher
- proud klíče (key stream) je generován nezávisle na OT/ŠT
- SSC šifra vyžaduje, aby odesílatel a příjemce byli synchronizováni při použití proudu klíče a operací na shodné pozici
- ztráta/přidání bitu → dešifrování znemožněno a je nutná resynchronizace obou stran



- resynchronizace spočívá ve vkládání speciálních značek do šifrového textu, které reinicializují ztracenou synchronizaci
- kladnou vlastností SSC je nešíření chyb • změna bitu ŠT neovlivní dešifrovací proces, pouze bit OT
- SSC jsou náchylné na útoky typu mazání, vložení či opakování bitů - důsledkem je ztráta synchronizace
- v praxi je nezbytné tyto situace detekovat a zavést mechanizmy pro zajištění autentizace a integrity dat

Asynchronní proudové šifry

- SSC - Self-synchronnizing Stream Cipher
  - proudové šifry s vlastní synchronizací nebo
  - samosynchronizující se šifry nebo
  - CTAK – Cipher Text Auto Key
- proud klíče závisí na OT/ŠT
- proudový klíč je generován v závislosti na
  - klíči K
  - pevně daném počtu bitů předchozího ŠT



- vlastní synchronizace je možná, i když jsou některé bity smazané či vložené, protože dešifrovací mapování závisí na pevném počtu předchozích zašifrovaných bitů
- po ztrátě synchronizace se automaticky obnoví správná činnost - ztratí pouze pevný počet bitů otevřeného textu.
- pokud stav asynchronního proudu šifry závisí na t předchozích šifrovaných bitech a je-li jeden šifrovaný bit modifikován (smazán nebo vložen) během přenosu, potom může být dešifrování maximálně t následujících bitů šifrového textu nesprávných, než nastane opět správné dešifrování
- omezené šíření chyb
- Příklady:
    - CFB režim blokových šifer

Současné proudové šifry

- vyskytují se v běžných technologiích
    - WiFi – IEEE 802.11
    - Bluetooth – IEEE 802.15
    - GSM – ETSI GSM 09.01
- časté HW implementace

- algoritmy RC4,A5/1

Proudová šifra RC4

- nemá IV → pro každé spojení se používá nový klíč
- šifra nebyla nikdy oficiálně publikována
- 09/1994 anonymní hacker uveřejnil zdrojový kód získaný pomocí technologie reverse engineering
- dostupná verze se označuje ARCFOUR
    - RC4 je ochranná známka RSA Data Security Inc.
- volitelná délka klíče 8...2048b
    - nejčastěji128 bitů
- v jednom kroku algoritmus vygeneruje 1B proudu klíče

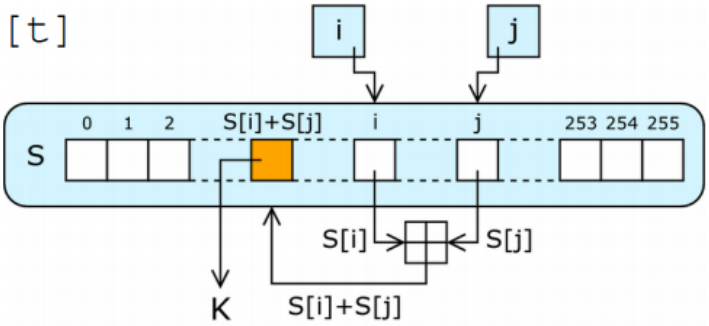
bezpečnost RC4

- pokud se stejný klíč má použít ke generování více proudů, je nutné použít další parametr - nonce (náhodné číslo) k jejich rozlišení
- kryptosystém musí mít mechanismus, jak dlouhodobý klíč a nonce bezpečně kombinovat
- dobrý způsob generování proudových klíčů :
    - hash z klíče a nonce
- v praxi často používaný způsob (špatný):
    - pouhé spojení klíče a nonce
    - vzhledem ke způsobu inicializace RC4 to způsobuje vážné bezpečnostní problémy
    - viz přednáška o zabezpečení bezdrátových sítí (WEP)

- existuje korelace mezi prvním bajtem proudu klíče a prvními třemi bajty klíče
- několik prvních bajtů permutace je lineární kombinací vybraných bajtů klíče
- prvních několik bajtů proudu klíče má velmi NEnáhodný charakter → prolomení WEPu (FMS útok)

V každém kroku se prohodí prvky tabulky a vybere bajt:

```
i = (i + 1) mod 256
j = (j + S[i]) mod 256
swap(S[i], S[j])
t = (S[i] + S[j]) mod 256
keystreamByte = S[t]
```



Proudová šifra A5

- algoritmus utajován
- 1994 – únik informací o obecné struktuře (LFSR)
- 1990 – rekonstrukce algoritmu (reverse engineering)
- A5/1 klíč délky 64 bitů + číslo TDMA rámce (veřejně známé)
    - 10 bitů vždy nastaveno na log.0
    - efektivní délka klíče 54bitů
- 1997 – teoretický útok J. Golić
    - na stejné konferenci, kde popsal útok na RC4 ϑ
- 1999 - A5/2 vyvinuta pro oblast Asie a východní Evropy
    - výrazně slabší → prolomení v řádu ms
    - kryptoanalýza uveřejněná ve stejném měsíci, kdy byla publikována ukázala vážné slabiny
- od 2006 GSMA zákaz podpory A5/2 v mobilních telefonech
- v doporučeních 3GPP je A5/2 dodnes jako volitelný algoritmus

- šifrování na rádiovém rozhraní sítě mezi MS a BTS
    - šifrovací algoritmus A5 je uložen v MS
    - je stejný pro všechny operátory (proč ?)
    - klíč generován pomocí A8 uloženým na SIM kartě
- specifikace A5 nebyla nikdy oficiálně publikována
- vstup do A5:

  relační klíč Kc = A8(RAND, Ki) délky 64 bitů číslo TDMA rámce (22 bitů)

  výstup: 114 bitů
- šifra produkuje vždy 228 bitů proudu klíče

- 114 bitů se používá pro šifrování komunikace od telefonu k základové stanici
- 114 bitů pro šifrování komunikace v opačném směru

- algoritmus A5/1 je založen na kombinaci tří lineárních registrů se zpětnou vazbou (LFSR – X-19b, Y-22b, Z-23b) a nepravidelného taktování
- klíčem je počáteční hodnota registrů
- každá buňka obsahuje jeden bit (srovnej s RC4)
- v každém kroku se každý registr posune nebo zůstane stát
- registr se posune pokud hodnota jeho „hodinového bitu“ (oranžový) souhlasí s většinovou hodnotou všech „hodinových bitů“
- proud klíče vzniká XORem výstupu tří registrů

Varianty A5

- A5/0 – žádné zabezpečení
- A5/1 – původní algoritmus A5 používaný v Evropě, prvních 10b je 0, efektivní délka klíče – 54b
- A5/2 – kryptograficky oslabená varianta pro export (Čína)
- A5/3 – silný šifrovací algoritmus založený na jap. blokové šifře KASUMI

- projekt eStream - obecně, cíle, příklad algoritmů (ty není potřeba detailně znát)

Projekt eSTREAM

- od 2004
- hledání nových proudových algoritmů
  - **rychlejší než AES**
  - **bezpečnější než AES**
- projekt organizovaný evropským konsorciem výzkumných organizací ECRYPT (European Network of Excellence for Cryptology)

3 fáze výběrového řízení

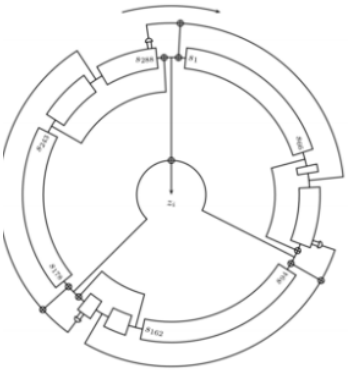
- 1. fáze do 02/2006
  - rámcová analýza všech kandidátů
  - vytvoření dvou profilů
    - **Profil 1**
      - SW implementace
      - vysoká propustnost
      - přijaty pouze algoritmy efektivnější než AES128 -CTR
    - **Profil 2**
      - HW implementace
      - omezené zdroje
      - počet hradel, množství paměti
- 2. fáze od 06/2006 -09/2007
  - podrobné testy kandidátů přijatých z fáze 1
  - pro oba profily byly dodatečně přijaty nové algoritmy
  - kandidáti fáze 2 byly každých šest měsíců reklasifikováni
- 3. fáze
  - celkem osm kandidátů v Profile1 a Profile2
  - 08/2008 - z kandidátů byly vybrány algoritmy:

    Profil 1: HC-128, Salsa 20/12, SOSEMANUK, Rabbit

    Profil 2: Grain v1, MICKEY v2, Trivium, F-FCSR (později vyřazen)

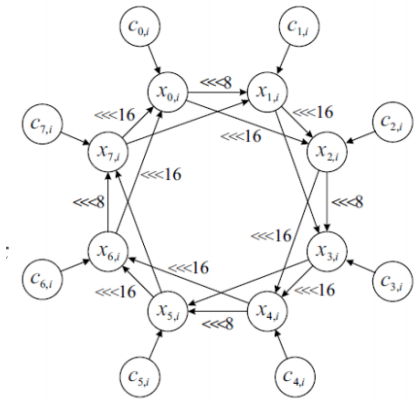
- Každých 6 měsíců by měl být revidován

E-stream Profile 2 - Trivium



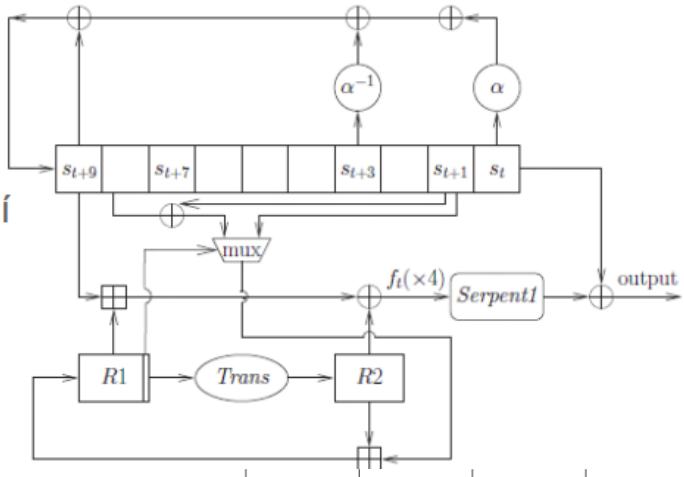
- synchronní proudová šifra
- IV … 80 bitů
- K … 80 bitů
- 3 FSR různé délky
- vnitřní stav celkem 288 bitů
- nejjednodušší návrh ze všech algoritmů v E-stream projektu
- 9/2010 – nejlepší útok na odhalení vnitřního stavu šifry vyžaduje $2^{89.5}$ kroků

E -stream Profile 1 - Rabbit



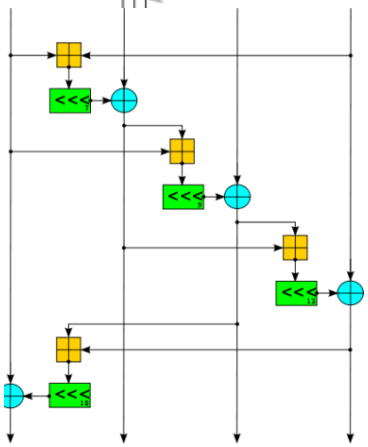
- K …128 bitů
- IV … 64 bitů
- SW implementace
- RFC 4503
- s(K,IV) lze zašifrovat 2^{64} 128bitových bloků
- vnitřní stav …513b
  - osm 32bitových registrů x
  - osm 32bitových čítačů c
  - jeden „carry“ bit

E-stream Profile 1 - SOSEMANUK



- K …128 bitů
- IV …128 bitů
- SOSEMANUK = „snow snake“ v jazyce kmene Cree
  - LFSR
    - deset 32bitových čísel
    - operace v F2 32
  - modifikovaný Serpent
    - bez přičítání klíčů
    - bez lineárních transformací
  - FSM - 64b paměti • realizováno registry R1,R2
  - v každém kroku FSM vezme data z LSFR, aktualizuje R1,R2 a vygeneruje 32b výstupu

E-stream Profile 1 – Salsa20/12



- K…256 bitů
- nonce … 64bitů
- číslo bloku … 64 bitů
- výstup…512 bitů
- Salsa 20/12 používá 12 rund z 20
- jádrem Salsa 20 je hashovací funkce s 64B vstupem/výstupem
- hashovací funkce je použita v CTR režimu, kdy se chová jako proudová šifra

ChaCha20

- odvozena z šifry Salsa
- „The ChaCha20 cipher is designed to provide 256-bit security.“
    - efektivnější než AES (bez AES-NI)
- použití v konstrukcích typu AEAD
    - AEAD - Authenticated Encryption with Aditional Data
    - spolu s algoritmem Poly1305
- varianta MAC (Message Authentication Code)
- alternativa k AES-GCM
- podporována v TLS 1.2
    - v Chrome od verze 32 (cca 11/2013)
    - ve Firefoxu od verze 47 (06/2016)

Režimy činnosti blokových šifer

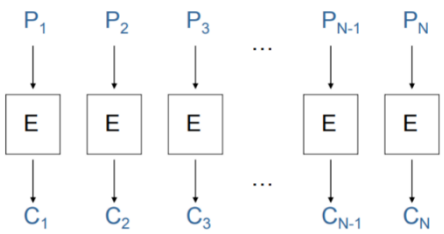
- obecné informace:

- Režim činnosti blokové šifry (block cipher mode of operation), zkráceně režim činnosti, je algoritmus využívající symetrické blokové šifry tak, aby efektivně zajistil některé cíle informační bezpečnosti např. utajení nebo autentizaci.
- Režimy činnosti pro zajištění utajení:
    - FIPS 81 – ECB, CBC, CFB, OFB
    - NIST SP800-38A – CTR, rozšíření CBC o cipher-text stealing (CTS)
- Režimy činnosti pro zajištění autentizace:
    - NIST SP800-38B
    - CMAC, OMAC (One Key CBC-MAX), XCBC
- Kombinované režimy činnosti
    - Zajišťují současně utajení, autentizaci, integritu
    - AE (Authenticated Encryption)
    - AEAD (Authenticated Encryption with Associated Data)
    - GCM, CCM, EAX
- Speciální režimy činnosti
    - Vyvinuto pro situace vyžadující specifické přístupy (šifrování dat na harddisku)
    - LRW, XEX, XTS

- ECB, CBC,OFB,CFB, CTR - nejenom poznat, ale i umět nakreslit

ECB – Electronic Code Book

- Zapisujeme: C=E(P,K)
- OT: P0,P1,…,Pm,… ŠT: C0,C1,…,Cm,…
- Pro každý klíč K existuje jiná kódová kniha!
- **Výhody** ECB:
    - Kazdy blok je šifrován nezávisle.
    - Jednotlivé bloky je možné šifrovat v libovolném pořadí.
    - Možnost paralelizace při šifrování i dešifrování.
    - Šíření chyb - změna bitu = ztrátou celého bloku ostatní bloky nejsou ovlivněny

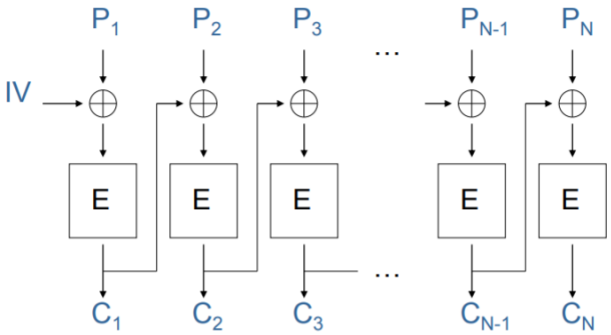


- **Nevýhody** ECB:
    - Nechť Pi =Pj pak Ci =Cj a útočník ví, že Pi =Pj .
    - To dává útočníkovi určitou informaci, přestože nezná Pi nebo Pj .
    - Je možné realizovat útok typu „cut-and-paste”.

- Tabulku šifer je možné začít konstruovat i bez znalosti klíče.
- Ve většině situací mají části zprávy tendenci se opakovat. (začátky zpráv obsahují záhlaví, která mohou být velice podobné, podobně i konce zpráv).

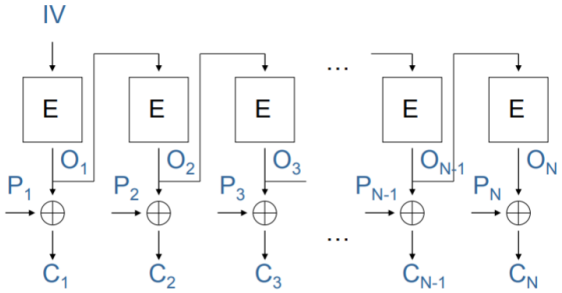
CBC

- bloky jsou „zřetězeny“ dohromady
- na začátku CBC je nutné použít inicializační vektor (IV)
- stejné bloky OT se šifrují na různé bloky ŠT
- cut-and-paste je stále možný, ale je složitější a zůstanou po něm „artefakty“
- pokud je během přenosu blok C1 změněn na G pak P1 ≠ C0⊕D(G,K), P2 ≠ G⊕D(C2,K) ale P3 = C2⊕D(C3,K), P4 = C3⊕D(C4,K) atp.
- dojde tedy k automatické obnově OT !
- v prvním bloku je nutné přidat náhodná data nazývaná inicializační vektor (IV).
- IV zajistí aby dvě zprávy (OT1 a OT2) byly zašifrovány pokaždé jinak i pokud OT1=OT2



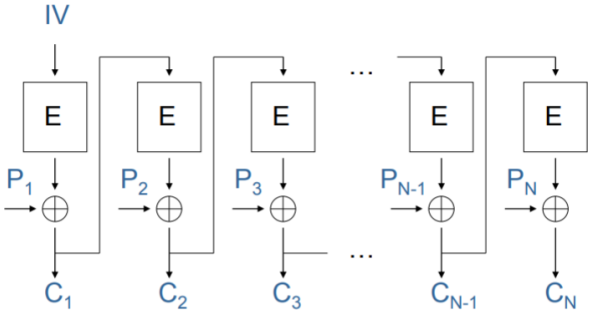
OFB - Output Feedback Mode

- používá pouze šifrovací transformaci -> efektivní HW implementace
- z výstupního bloku není nutné přenášet celý blok do vstupního bloku, stačí pouze část, nějakých n bitů.
- vstup algoritmu = náhodný text
- šifrování a dešifrování neparalelizovatelné
- vhodný pro vysokorychlostní systémy s nepřípustným šířením chyb
- 1-bitová chyba v šifře ovlivní jeden bit po dekódování
- OFB je čistě synchronní proudová šifra, proud klíče není ovlivňován ani otevřeným ani šifrovým textem
- vstupní blok má nejvýše 2 N možných stavů, kde N je délka vstupního bloku v bitech. Po nejvýše 2 N blocích se tak začne proud klíče opakovat



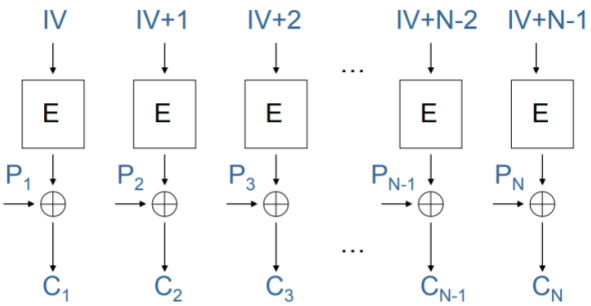
CFB - Ciphertext Feedback Mode

- používá pouze šifrovací transformaci -> efektivní HW implementace
- není nutné přenášet celý blok ŠT do vstupního bloku, stačí pouze část n bitů → označuje se jako n-bitový CFB režim
- vstupní blok má nejvýše 2 N možných stavů, kde N je délka vstupního bloku v bitech. Po nejvýše 2 N blocích se tak začne proud klíče opakovat.
- vstup algoritmu = náhodný text (IV)
- šifrování neparalelizovatelné
- dešifrování paralelizovatelné, v libovolném pořadí
- šifrování znakových terminálů (8-bitový CFB)
- nejčastější varianta 8-bitový CFB, ale lze i jiné (1-bitový, 16 bitový…)
- Šíření chyb:
  - CFB režim je samosynchronizující
  - k obnovení správného OT stačí pouze dva po sobě jdoucí nenarušené bloky ŠT, resp. neporušené příslušné n bitové části ve dvou po sobě jdoucích blocích ŠT



CTR **–** Counter **– čítač**

- používá blokovou šifru jako proudovou, podobný OFB
- v proudu klíče se nemusí používat celý výstupní blok, ale jenom nějaká jeho část
- CTR se používá v případě potřeby náhodného přístupu (mohu šifrovat/dešifrovat libovolný blok bez ohledu na ostatní)
- při šifrování dvou různých zpráv se stejným klíčem se nesmí vygenerovat stejná část proudu klíče



- režimy AEAD pouze základní rozdělení + příklady

Režimy činnosti pro zajištění utajení a autentizace (AEAD)

- vznikají po roce 2000
- ISO/IEC 19772:2009
  - OCB 2.0, CCM, EAX, KeyWrap, GCM, EtM - Encrypt-then-MAC, E&M – Encrypt and MAC, MtE – MAC then Encrypt

Encrypt then MAC (EtM)

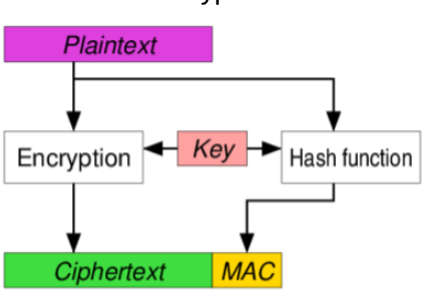
- použito se např. u IPsec
- RFC 7366 – podpora Et
- nejbezpečnější varianta

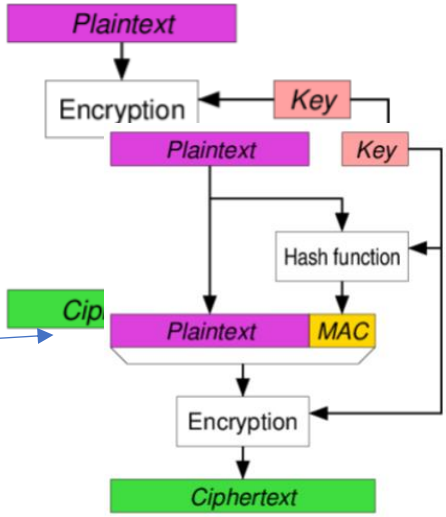
Encrypt and MAC (E&M)

- použito např. u SSH

MAC then Encrypt (MtE)

- použito např. u SSL/TLS jako výchozí metoda





Asymetrické algoritmy

- rozdělení (IFP,DLP,ECDLP)

- symetrické
  - dnes velmi rychlé algoritmy hodící se k šifrování velkých objemů dat
  - malé (relativně) délky klíčů (128-bit, 256-bit)
  - obě dvě strany musí znát stejný tajný klíč
  - problém distribuce/změny klíčů v případě velkého množství komunikujících stran
  - nelze prokázat totožnost autora zašifrovaných dat (data mohla zašifrovat kterákoliv strana vlastnící klíč)
  - blokové (DES, 3DES, IDEA, RC6, Blowfish, CAST, AES - Rijndael….)
  - proudové (Enigma, SEAL, RC4, A5,…)
- asymetrické
  - pomalé (100-1000x)
  - velké délky klíčů (1024b, 2048b,…)
  - různé klíče pro šifrování a dešifrování
  - **klíče spolu vzájemně souvisejí**
  - IFP (RSA, Rabin-Williams, Lucas,…)
  - DLP (DSA, Diffie-Hellman, El-Gamal, …)
  - ECDLP (ECDSA, PSEC,…)

- RSA, DSA (není v přednášce?), DH

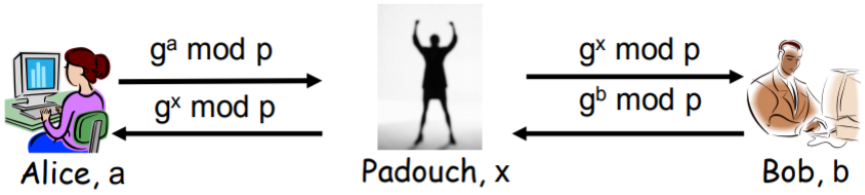
RSA

- první asymetrický šifrovací algoritmus
- nejznámější kryptosystém veřejného klíče
- používán v řadě protokolů (např.: SSL/TLS, IPsec,…)
- umožňuje jak šifrování, tak i podepisování
- založen na umocňování celých čísel modulo prvočíslo
- používá velká prvočísla (více než 100 dekadických míst ~1024 bitů)
- Na stejném zařízení je RSA:
    - při HW implementaci cca 1000x pomalejší nežli DES.
    - při SW implementaci cca 100x pomalejší nežli DES.
- RSA s velkými moduly je neefektivní(se vzrůstající délkou modulu roste bezpečnost pomalu, ale výpočetní nároky rychle.)
    - složitost RSA-1024 je srovnatelná se symetrickou šifrou s klíčem délky 80bitů
    - složitost RSA-15,380 přibližně odpovídá klíči délky 256 bitů (např. AES-256)
    - Čas nutný k podepisování roste s třetí mocninou délky klíče
- Výhoda RSA: Dosud nebyl objeven rychlý algoritmus pro rozklad velkých složených čísel.

DH (Diffie-**Helmannův protokol)**

- zcela první **veřejně publikovaný** kryptosystém veřejného klíče
- nelze použít k šifrování
- nelze použít k podepisování
- lze ho použít „pouze“ pro **výměnu klíčů**
    - používá se k ustanovení sdíleného (symetrického) klíče
- bezpečnost závisí na obtížnosti řešení **diskrétního logaritmu**
- Princip:
    1. Alice vygeneruje tajné náhodné číslo a
    2. Bob vygeneruje tajné náhodné číslo b
    3. Alice odešle Bobovi g^a mod p
    4. Bob pošle Alici g^b mod p
    5. oba si spočítají sdílenou tajnou hodnotu $(g^b \bmod p)^a$ modp $=(g^a \bmod p)^b$ mod p $=g^{ab}$ mod p
    6. toto číslo g^{ab} mod p lze použít jako sdílený tajný klíč

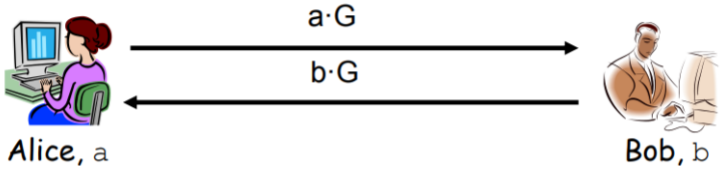
- Systém je prolomen, pokud útočník zjistí hodnotu a nebo b
- Systém je také možné prolomit vyřešením diskrétního logaritmu
- V případě použití protokolu DH je nutné počítat s možností Man in the middle útok.



- ECDH

- algoritmus analogický s klasickým Diffie-Hellmanovým algoritmem
- umožňuje komunikujícím stranám získat sdílenou tajnou informaci použitelnou např. jako klíč pro klasickou symetrickou šifru
- Alice spočítá a ·(b · G)
- Bob spočítá b ·(a · G)

- K = a ·(b · G)= b ·(a · G) je možné použít jako klíč pro symetrickou šifru; K – souřadnice jiného bodu na křivce
- potencionální útočník může zachytit pouze G,aG, bG



- srovnání IFP x DLP x ECDLP

IFP **– Bezpečnost**

- Pollardova ρ, p-1, p+1 metoda
- NFS (Number Field Sieve)
- TWINKLE, TWIRL (zatím jen teoretické návrhy)
  - optická zařízení urychlující (100-1000x) počáteční fázi prohledávání možných řešení. Po skončení se pokračuje s klasickým NFS. Stroj v ceně několika desítek miliónů dolarů umožní prolomení jednoho klíče RSA s délkou modulu 1024 bitů za 1 rok.
- kvantové počítače (zatím jen teoreticky)
  - podařilo se sestrojit KP, který faktorizoval číslo 15

DLP **– Bezpečnost**

- Obdobné metody jako u kryptosystémů na bázi IFP a navíc několik specifických útoků na DLP:
  - Index-Calculus
  - Pollardova ρ a λ metoda pro DL
  - Pohlingův-Hellmanův algoritmus

ECDLP – Bezpečnost

- nejefektivnější známá metoda – Pollardova ρ
- časová složitost $O(\sqrt{\frac{\pi n}{2}})$ kroků (n je typicky 2160 a více !!!)
- výhoda – malá pamětová složitost
- lze paralelizovat – při použití m strojů je složitost $O(\frac{\sqrt{\frac{\pi n}{2}}}{m})$

- doporučené délky klíčů, suite B

Symetrické šifry	ECDLP	IFP/DLP	Bezpečnost
56	112	512	**NEBEZPEČNÉ** - dnes již dokážeme luštit
64	128	768	**Krátkodobá bezpečnost** (do 5-10 letech půjde luštit; jsou publikovány úspěšné pokusy o prolomení)
80	163	1024	
112	231	2048	**Dlouhodobá bezpečnost** - pokud nedojde k novým zásadním objevům nebo konstrukci kvantového počítače
128	283	3072	
192	409	7680	
256	571	15360	

Suite B

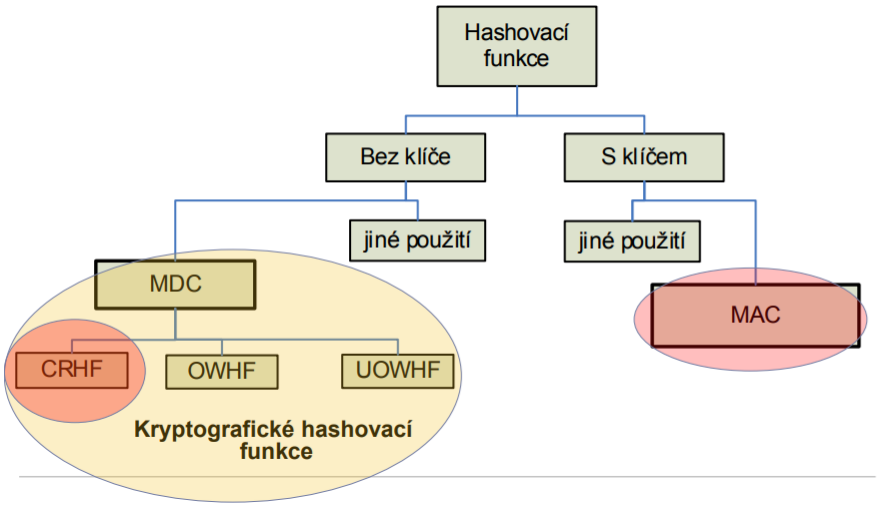
- „doporučení“ pro bezpečné sdílení informací v mezinárodním měřítku z dílny NSA, NIST a NATO
- Suite-B algoritmy jsou určeny pro šifrování klasifikovaných informací* , ale samotné algoritmy nejsou tajné a jsou volně dostupné
- publikován 16.3.2005
- obsahuje tři základní komponenty:
    - asymetrické algoritmy na bázi ECC
    - symetrický algoritmus AES
    - hashovací funkce z rodiny SHA-2
- Algoritmy:
    - Šifrování - AES (FIPS 197)
        - AES-128 až do úrovně SECRET
        - AES-256 až do úrovně TOP SECRET
    - Elektronický podpis (FIPS 186-3)
        - ECDSA s prvočísleným modulem délky 256 bitů až do úrovně SECRET
        - ECDSA s prvočísleným modulem délky 384 bitů až do úrovně TOP SECRET
    - Výměna klíčů (NIST SP 800-56A)
        - ECDH s prvočísleným modulem délky 256 bitů až do úrovně SECRET
        - ECDH s prvočísleným modulem délky 384 bitů až do úrovně až do úrovně TOP SECRET
    - Hashovací funkce (FIPS 180-2)
        - SHA-256 do úrovně SECRET

SHA-384 do úrovně TOP SECRET

- ECC principy, EC over Fp, EC over F2m

Hashovací funkce

- obecné informace (rozdělení, vlastnosti, Merkle-Damgardova struktura)



- Obecně lze hashovací funkce rozdělit na dvě třídy:
    - hashovací funkce bez klíče – mají jediný vstupní parametr – zprávu M
    - hashovací funkce s klíčem – dva nezávislé vstupy – zprávu M a klíč K –
- Hashovací funkce je funkce h, která musí splňovat minimálně tyto požadavky:
    - 1. komprese — funkce H mapuje libovolně konečně velký vstup x, na výstup h(x) pevné délky n

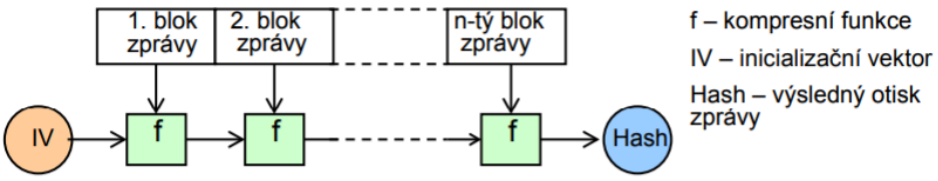
    - 2. jednoduchost výpočtu - pro danou hashovací funkci H a vstup x, je jednoduché spočítat h(x)
    -

MAC - Message Authentication Code
- skupina funkcí parametrizovaných (tajným) klíčem k
- zajišťuje integritu zprávy a zároveň zaručuje i zdroj zprávy (autentizace)
- MAC musí mít následující vlastnosti:
    - 1. komprese — funkce hk mapuje libovolně velký konečný vstup x, na výstup hk(x) pevné délky n
    - 2. jednoduchost výpočtu - pro danou hashovací funkci h, klíč k a vstup x, je jednoduché spočítat h(x). Výsledek se nazývá MAC nebo MAC-hodnota
    - 3. výpočetní odolnost – pro jeden nebo více párů (xi ; hk(xi)) je výpočetně nemožné spočítat jinou dvojici (x; hk(x)) pro x ≠xi (zahrnující i možnost hk(x)=hk(xi) pro některé i) – tzn. z hk(x) není možné získat k (označuje se jako „key non-recovery“)
    -

MDC **–** Modification Detection Code
- někdy též MIC (Message Integrity Check)
- účelem je poskytnout reprezentativní vzorek (hash) zprávy, který budu vyhovovat dalším dodatečným podmínkám
- zajišťuje integritu dat
- OWHF – One-way Hash Function
    - je obtížné najít vstup jež bude mít nějaký předem daný otisk (hash)
- CRHF – Collision Resistant Hash Function
    - je obtížné nalezení dvou různých vstupních textů se stejným otiskem (hashem)
- UOWHF – Universal One-way Hash Function
- OWHF, CRHF, UOWHF – kryptografické hashovací funkce

Základní vlastnosti dobré hashovací funkce
- Odolnost vůči získání předlohy
- Odolnost vůči získání jiné předlohy
- Odolnost vůči nalezení kolize (Collision resistence)
    - Odolnost neznamená, že kolize nebudou existovat (vždy budou, z principu musí!), ale o obtížnost jejich nalezení

Merkle-**Damgårdova struktura hashovací funkce**
- bloky f provádějí kompresní funkci
- toto schéma využívá naprostá většina moderních hashovacích funkcí – MD-5, SHA-1, RIPEMD-160
- vstup musí být doplněn na celistvý násobek délky bloků
- musí být jednoznačně určitelné kolik se doplnilo
    - jinak by jednoduše vznikala řada kolizí - zpráva doplněná na …0110000 by mohla vzniknout ze zprávy …011, …0110, …01100,…
- Merkle-Damgårdovo zesílení (doplnění výplně posledního bloku o délku zprávy – zvyšuje odolnost vůči generování kolizí)



- SHA-1, SHA-2, SHA-3

SHA-1

- oblíbená, ale již kompromitovaná funkce. V únoru 2005 byl zveřejněn objev algoritmu, který umožňuje nalézt kolizi podstatně rychleji než hrubou silou. únor 2017 – Google nalezl kolizi v SHA-1
- Vstup: řetězec proměnné délky (max. 264 -1 bitů)
- Výstup: pevná délka 160 bitů
- Počet rund: 4
- Počet kroků v rundě: 20
- Vstup je zpravován po úsecích délky 512 bitů
- Zpráva je zarovnána tak, aby byla dělitelná 512.

SHA-2
- dosud považována za bezpečnou – není to jedna hashovací funkce, ale více variant souhrnně označovaných jako SHA-2 (SHA-224, SHA-256, SHA-384, SHA-512)
- označují se jako SHA-2 (SHA-224, SHA-256, SHA-384, SHA-512)
- nové vstupní hodnoty (IV) - odpovídají desetinným částem druhých odmocnin prvních osmi prvočísel
- v každém kole se používají odlišné aditivní konstanty
    - odpovídají desetinným částem třetích odmocnin prvních 64 prvočísel
- opět konvence big-endian
- SHA-224 je realizována jako SHA-256 s jinými počátečními hodnotami a příslušně oříznutým výstupem

SHA-3 – standardizována 08/2015, nepoužívá se

- útoky na hashovací funkce - principy

- CMAC,HMAC

CMAC (CBC-MAC)
- bloková šifra v režimu CBC
- IV= 000…000
- tajný klíč KMAC – tento klíč by měl být jiný, než na šifrování zpráv
- ze zprávy M={m1, …, mn} se vygeneruje ŠT={c1, …, cn}
- MAC je tvořen pouze posledním blokem cn
    - v praxi se z posledního bloku obvykle bere jen určitá část o délce potřebné k vytvoření odolného zabezpečovacího kódu
- ostatní bloky ŠT se průběžně zahazují (neodesílají se !)
- MAC zajišťuje autentizaci původu dat
- chrání proti úmyslným modifikacím dat během přenos

HMAC (Keyed-hash Message Authentication Code)
- kryptografický kontrolní součet, kryptografická hašovací funkce, klíčovaná haš,…
- označení podle toho, typu použité hašovací funkce např. HMAC-SHA-1, HMAC-MD5
- zpracovává nejen data, ale i klíč
- výsledek je závislý na klíči a tudíž "nepadělatelný„
- ověřuje integritu dat a zdroj dat o
- dlišuje se od MAC, funkčně podobný, ale používá jiné stavební prvky a je kryptograficky silnější
- HMAC-SHA-1:
    - blok B=64 bajtů,klíč K, H = SHA-1
    - ipad = řetězec B bajtů „0x36“
    - opad = řetězec B bajtů „0x5C“
    - klíč K se doplní nulovými bajty do plného bloku délky B

Obsah - horní část dokumentu

Autentizace

Autentizace (proces identifikace nebo verifikace) uživatele/entity je možné provádět podle toho :

- kdo jsou – identifikace podle globálně jednoznačných parametrů (biometrika – otisky prstů, dlaní, sítnice, DNA apod.)
- co mají – identifikace podle vlastnictví určitých předmětů (klíč, magnetická/čipová karta, USB token apod.)
- co znají – identifikace pomocí přístupových hesel, číselných kombinací, osobních identifikačních čísel – PIN apod.

- rozdělení

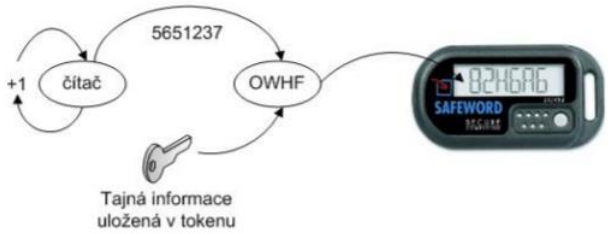
- Biometrika
- Tokeny
- Hesla

- principy jednotlivých technik

Biometrika

- automatizované metody identifikace nebo verifikace osoby na základě měřitelných fyziologických nebo behaviorálních vlastností člověka.“
- identifikace lidí na základě jejich osobních charakteristik
- biometriky se liší různou mírou spolehlivosti, ceny a společenské přijatelnosti
- Proces biometrické autentizace:
    - Registrace uživatele:
        - ověření identity žadatele např. vizuální kontrola fotografie v OP a skutečného vzhledu
        - získání registračního vzorku
        - pro zvýšení přesnosti nejlépe několikrát
        - uložení do databáze
        - zkušební rozpoznání
    - Autentizace:
        - získání aktuálního vzorku
        - porovnání s uloženým registračním vzorkem

Tokeny



- biometrické autentizační metody - principy, rozdělení

- statické charakteristiky = anatomicko-fyziologické charakteristiky jedince
  - otisk prstu, dlaní, chodidel
  - geometrie prstů, ruky
  - oční duhovka, sítnice, zornice
  - tvář – 2D/3D
  - tvar vnějšího ucha
  - rty
  - krev, srdeční pulz
  - snímek vlasů
  - dentální obraz
  - topografie žil zápěstí, cévní řečiště
  - struktura nehtového lůžka
  - chemické složení pachu
  - obsah soli v lidském těle
  - rozměry a váhy lidského těla
  - DNA
- dynamických charakteristiky = behaviorální charakteristiky vyžadují interakci účastníka
  - podpis
  - hlas
  - chůze
  - analýza psaní na klávesnici
  - analýza pohybů myši
  - mimika obličeje a pohyby rtů
  - pravděpodobnostní charakteristiky konverzace

- AAA (Authentication, Authorization, Accounting)

- pro ochranu IP sítí z hlediska přístupu se používá tzv. AAA architektura
- AAA obsahuje položky:
  - Autentizace
  - Autorizace
  - Účtování
- nepřímá autentizace
  - používá se ve větších sítích
  - přesun přihlašovacích údajů z koncových zařízení na jedno místo
  - centrální autentizační server
- v rámci architektury AAA se nejčastěji používají protokoly TACACS+ nebo RADIUS

- PAP,CHAP/MSCHAPv2

PAP – Password Authentication Protocol (RFC 1334)

- slabá autentizace
- heslo se přenáší v OT
- autentizace je možná pouze na počátku spojení
- autentizaci iniciuje klient

Challenge Authentication Protocol (RFC 1994)

- challenge-response (výzva-odpověď)
- neposílá heslo, ale hash z hesla a výzvy response=MD5(heslo,challenge)
- „výzva“ je při každém přihlášení jiná
- obě komunikující strany musejí znát tajné heslo
- autentizaci iniciuje server
- autentizace může být prováděna opakovaně, kdykoliv během spojení

MS-CHAPv2 (RFC 2759)

- podpora od Windows 2000
- umožňuje vzájemnou autentizaci komunikujících stran

- využívá SHA-1

- TACACS+, RADIUS, DIAMETER, Kerberos

TACACS+
- podporuje všechny tři složky architektury AAA
- není kompatibilní s TACACS a XTACACS
- TACACS+ odděluje proces autentizace a autorizace což umožňuje např. použít k autentizaci Kerberos a k autorizaci TACACS+ •
- šifruje celý paket
- na transportní vrstvě používá protokol TCP
- proprietární CISCO protokol 0

RADIUS – Remote Authentication Dial-In User Service
- zahrnuje všechny tři složky architektury AAA
- systém RADIUS má tři složky:
    - protokol
    - autentizační server
    - klienty
- používá transportní protokol UDP
- transakce mezi serverem a klientem se autentizují pomocí sdíleného hesla, které se nikdy nepřenáší v síti
- hesla uživatelů se posílají mezi serverem a klienty zašifrovaná (na rozdíl od TACACS+ se šifrují pouze hesla)
- RADIUS je specifikován v RFC 2865, 2866
- používá se v 802.1x

DIAMETER
- RFC 3588
- následovník RADIUSu
- není přímo zpětně kompatibilní s RADIUSem
- Hlavní rozdíly:
    - používá spolehlivý transportní protokol (TCP nebo SCTP*)
    - může být zabezpečen pomocí IPsec nebo TLS
    - má dočasnou podporu pro RADIUS (tzn. DIAMETER server může vystupovat jako RADIUS)
    - má více adresního prostoru pro AVP(Attribute Value Pairs) a identifikátory (32 bitů místo 8)
    - je to C/S protokol, ale podporuje i „server-initiated“ zprávy
    - dokáže dynamicky objevovat peery (jiné DIAMATER server) (pomocí DNS SRV a NAPTR)
    - má implementováno oznamování chyb
    - má lepší podporu roamingu
    - je snáze rozšiřitelný (dají se definovat nové příkazy a atributy)
    - je zarovnán na 32 bitové hranice
    - má základní podporu pro uživatelské relace a účtování (accounting)

Kerberos
- nepřímý autentizační protokol, C-S model
- vzájemná autentizace
- dnes se používá Kerberos v5
- RFC 4120
- podpora SSO (Single Sign On)

- výchozí autentizační protokol ve Windows (od verze 2000)
- používá vlastní implementaci protokolu
- centralizovaný, síťový autentizační systém, nezajišťuje autorizaci ani účtování
- základem jsou tzv. tickety (ticket-lístek),
    - vzájemná identifikace účastníků komunikace
    - výměna šifrovacích klíčů pro komunikaci s ostatními účastníky
- velikost ticketu - řádově stovky B
- tickety lze vložit do kteréhokoliv jiného síťového protokolu
- všechny zprávy obsahující citlivé informace (hesla) se přenášejí zašifrované

- EAP obecně , EAP-TLS, EAP-PEAPv0

EAP – Extensible Authentication Protocol

- RFC 5247, 7057
- obecný rámec pro autentizaci
- sám o sobě autentizaci nedělá
- řada variant (metod) • EAP-TLS • EAP-TTLS • PEAPv0 • EAP-MD5 • EAP-FAST
- používán v IEEE 802.1x

EAP-TLS (Transport Layer Security)

- nejbezpečnější, nejdražší na implementaci
- vzájemná autentizace pomocí certifikátů a protokolu TLS
- server používá TLS k dokázání vlastnictví digitálního certifikátu a to samé požaduje od klienta
- klient používá svůj certifikát k prokázání své identity a k výměně dat pro generování klíčů
- po úspěšné autentizaci je tunel ukončen, ale klíče odvozené během EAP-TLS se používají k šifrování pomocí AES, TKIP nebo WEP.

PEAPv0/EAP-MSCHAPv2

- 2. nejrozšířenější po EAP-TLS, vyvinul ho Microsoft
- MS neuznává PEAPv1, proto označuje protokol PEAPv0 pouze jako PEAP (tzn. neuvádí v0 a v1)
- známe implementace : AEGIS
    - Windows
    - od firmy Meetinghouse xsupplicant
    - všechny POSIX (Linux, BSD, Unix like OS)
    - projekt Open1x.org

VPN

- obecně

- logická síť v rámci sdílené veřejné infrastruktury
- poskytuje stejný výkon a pravidla jako soukromá LAN
- základní problémy při použití VPN
    - zajištění bezpečnosti
    - zajištění poskytování požadované kvality služeb (QoS) podle požadavků provozu
- tyto požadavky TCP/IP neřeší
- požadavky na bezpečnost se řeší pomocí
    - tunelování
    - šifrování

- autentizace
- řízení přístupu

- GRE

- IPsec - režimy, protokoly (AH,ESP),

- Internet Protocol Security
- komplexní soubor protokolů řešící:
  - šifrování
  - autentizaci
  - integritu
  - tunelování
- zabezpečení síťové vrstvě RM OSI
- doporučená součást IPv6, volitelné rozšíření IPv4
- IPsec pracovní režimy:
  - Transportní režim
    - určen pro spojení host-to-host
    - efektivnější než tunelovací
    - zůstává původní hlavička
  - Tunelovací režim
    - určen pro spojení site-to-site
    - kompletně nová IP hlavička
  - IPsec neobsahuje v záhlaví žádné v pole k určení režimu, režim se určí podle hodnoty pole „next-header“
    - IP … tunelovací režim
    - TCP, UDP, ICMP, cokoliv jiného – transportní režim

Protokoly:

- AH – Authentication header
  - nešifruje data
  - protokol č. 51 (v IP hlavičce)
  - autentizace záhlaví IP paketu
    - pouze neměnná pole
    - nelze chránit všechna pole
  - přidává se za původní IP záhlaví
  - integritu zajišťují algoritmy MD5, SHA-1, Tiger, SHA-2
  - vhodné pro aplikace, kde není nutné utajovat přenášená data, ale stačí je chránit před změnami
  - AH zajišťuje:
    - integritu přenášených IP datagramů
    - autentizaci odesílatele IP datagramů
    - ochrana proti replay útokům
- ESP - Encapsulating Security Payload
  - RFC 2406
  - protokol č. 50 (v IP hlavičce)
  - zajišťuje přenášená data šifrováním
    - záhlaví IP není v transportním režimu chráněno
  - integrita - volitelná
  - algoritmy DES,3DES, IDEA, CAST, RC-5, AES (vše v režimu CBC)
  - blokové šifry → zárovnání paketů !
  - při volbě algoritmu NULL se chová jako AH

- autentizace a dojednání klíčů (ISAKMP, IKEv1,IKEv2)

ISAKMP

- Internet Security Association and Key Management Protocol
- RFC 2408 • používá UDP (port 500)

- nejde o C-S komunikaci, ale výzva/odpověď
- strana, která chce vytvořit nové SA iniciuje komunikaci protokolem ISAKMP
- ISAKMP definuje vlastní SA
- ISAMP slouží k:
    - autentizaci komunikujících stran
    - výměně dat pro šifrovací klíče
    - obecný protokol pro vytváření SA
    - ISAKMP neřeší jak konkrétně se mají autentizované klíče vyměnit (to je práce protokolu IKE)

IKE - Internet Key Protocol
- flexibilní vyjednávací protokol
- vyjednání konkrétní metody autentizace, šifrování, délkách klíčů a dále umožňuje bezpečnou výměnu klíčů.
- RFC 2409
- používá Diffie-Hellmanův algoritmus
- používá se k výměně relačních klíčů (session key)
    - zprávy protokolu IKE jsou zapouzdřeny do paketů protokolu ISAKMP
    - má dvě fáze

IKEv2
- oprava chyb, zjednodušení oproti IEKv1
- snaha zůstat co nejvíc podobný IKEv1
- jediný způsob autentizace (digitální podpis)
- IKE SA + IPsec SA jsou vyměněny ve stejném protokolu pomocí 4 zpráv (~ Fáze 1 u IKEv1)
- zjednodušení dojednávání kryptografických parametrů
    - podpora pro „kryptografické sady“ (suites)
    - možnost oznámit „jakýkoliv z těchto šifrovacích algoritmů“, „jakákoliv z těchto hashovacích funkcí“
- vyšší spolehlivost
    - striktně challenge-response komunikace
    - odesílatel je zodpovědný za opětovné odeslání zprávy, pokud nepřijde odpověď
- dojednání parametrů pomocí „Traffic selector“
    - IKEv1: Responder může odpovídat pouze ANO/NE
    - IKEv2: přidána schopnost vyjednávat
    - více SA může mít stejný TS • obnova klíčů:
- může být vyvolána kdykoliv, libovolnou stranou
    - změny IKE-SA zahrnuje i všechny vnořené child-SA

- funkce SPD, SAD, SPI

SPD (Security Policy Database)
- eviduje IP spojení, které se chránit pomocí IPsec
- podobné routovací tabulce
- pro každý paket se podle zdrojové/cílové IP adresy, protokolu a směru (in/out) určuje, jestli se má :
    - 1) použít IPsec
    - 2) paket doručit bez použití IPsec
    - 3) paket zahodit

SAD - Security Association Database
- eviduje, jak se mají daty, na která má být aplikováno rozšíření IPSec chráněna
- jaké algoritmy a klíče je nutné pro konkrétní IP tok použít
- SAD obsahuje také přiřazení konkrétního SPI , dobu expirace klíčů, dobu expirace SA, minimální velikost MTU po cestě k cíli…

SPD tedy eviduje CO chceme dělat, SAD eviduje JAK to chceme provést.

SPI **–** Security Parameter Index

- SPI = ukazatel do SAD, kde jsou pro konkrétní spoj uvedeny požadavky na šifrování, zajištění integrity
- SPI – součást záhlaví každého paketu (AH, ESP)
- na základě detekce SPI se na příchozí paket aplikuje příslušné pravidlo
- každá strana má ke každému spojení sadu pravidel definující :
    - metodu zabezpečení (AH,ESP,obě)
    - pro každou metodu protokol pro kontrolní součet (MD-5, SHA-1)
    - pro ESP algoritmus (DES, 3DES, AES,…)

- DH grupy, PFS

D-H grupy (Diffie-Hellmann algoritmus pro výměnu klíčů)

- RFC 2409, 3526, 5114
- obvykle se používají grupy 1,2,5
- číslo grupy udává velikosti modulu (mod p)
    - DH-1 768 bitů
    - DH-2 1024 bitů
    - DH-5 1535 bitů
    - DH-14 2048 bitů

- SSL VPN
  - vhodné pro VPN typu extranet
    - umožňuje jednoduše komunikaci typu B2B,B2C
  - VPN brána se jeví jako běžný https server
    - dobře se vyrovnáná s NATem
    - vhodné pro mobilní uživatele
  - VPN (SSL) brána ukončuje spojení a směrem do vnitřní sítě navazuje nové
  - autentizace
    - heslo
    - token + heslo
    - certifikáty X.509
  - Varianty
    - Client-less – stačí HTTPS prohlížeč
    - Thin-client – aplikace (Citrix, SAP), lokální TCP proxy
    - Full-client – virtuální síťový adapter

SSL/TLS

- SSLv3 / TLSv1.0,1.1, 1.2

SSL – Secure Socket Layer

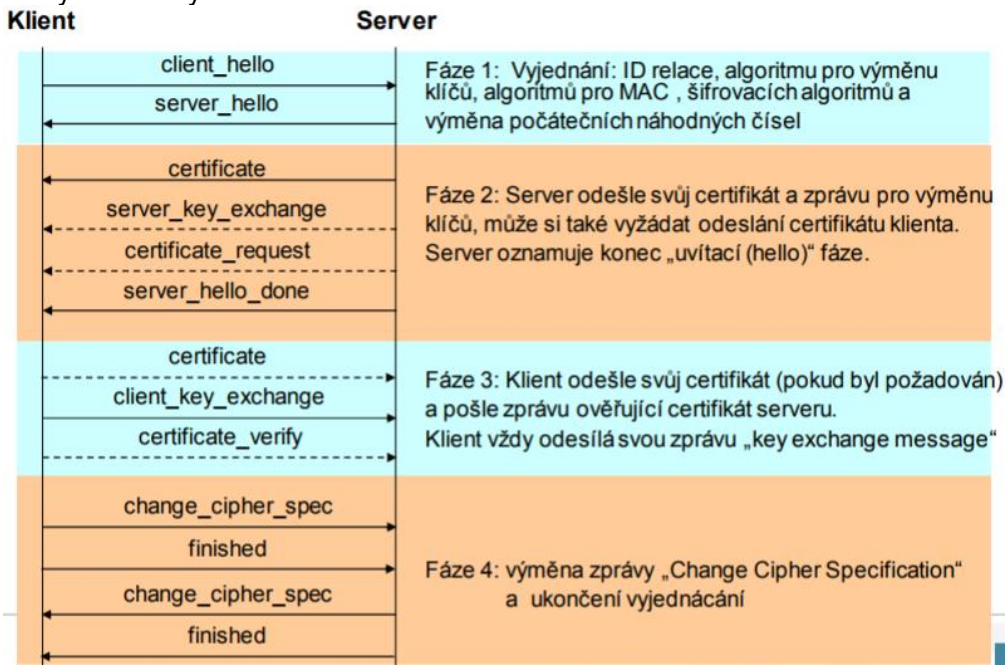
- vyvinut firmou Netscape
- nadstavba TCP/IP
- umístěna mezi aplikační a transportní vrstvou
- zajišťuje bezpečnou komunikaci mezi dvěma uzly
    - utajení přenášených informací
    - integrita přenášených zpráv
    - autentizace komunikujících stran
- dobrá interoperabilita
- snadná rozšiřitelnost o nové algoritmy
- používá spolehlivý transportní protokol (TCP)
- vytvořen s cílem zabezpečit prostředí WWW
- architektura klient/server

    - klient …ten, kdo zahajuje spojení
    - server…ke komu se klient připojuje
- duplexní komunikace
- existuje mnoho implementací protokolu SSL/TLS
    - OpenSSL, CyaSSL, yaSSL

TLS 1.2
- 08/2008
- RFC 5246
- **jiná PRF**
    - **náhrada kombinace hashovacích funkcí MD5/SHA**-1 za jedinou - SHA-256
    - **náhrada kombinovaného MD**-5/SHA-**1 hashe za jediný hash**
- **podpora nových šifrovacích režimů (AES**-GCM)
- **podpora AES přidána přímo do standardu TLS**
- **větší kontrola verzí zpráv HP**
- **sloučení kryptografických sad využívajících AES z externích RFC**

- dojednání klíčů

Cíl výměny: sestavení šifrovaného kanálu pro přenos hodnoty PreMasterSecret ve směru C→S
Lze realizovat více způsoby:
- RSA se šifrovacími klíči
- RSA s podepisovacími klíči
- DSA
- DH výměna s jednorázovými parametry
- DH výměna s permanentními parametry
- anonymní DH výměna



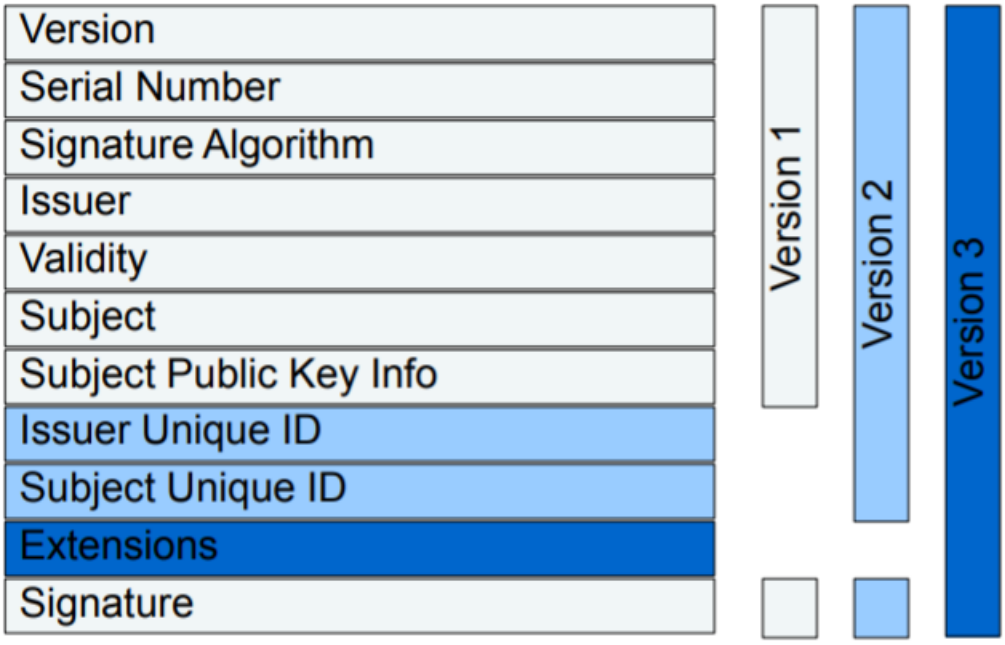
- DTLS

DTLS - Datagram Transport Layer Security
- DTLS 1.0 - 04/2006 (RFC 4347)
- součást OpenSSL od verze 0.9.8
- vychází z TLS, poskytuje obdobné služby
- používá UDP (obecně datagramově orientovaný)

- řazení paketů, ztrátu datagramů, nemožnost přenést datagram větší než …→ neřeší DTLS, ale aplikace
- vhodný pro aplikace citlivé na zpoždění (VoIP) a tunelované aplikace (VPN)
- DTLS 1.2 - 01/2012 (RFC 6347)
- nelze použít kryptografické sady z TLS 1.0
- ochrana handshake protokolu proti DoS útokům
- zpráva HelloVerifyRequest
- výměna cookies (příjemce je schopen přijmout a zpracovat data na udávané IP adrese)

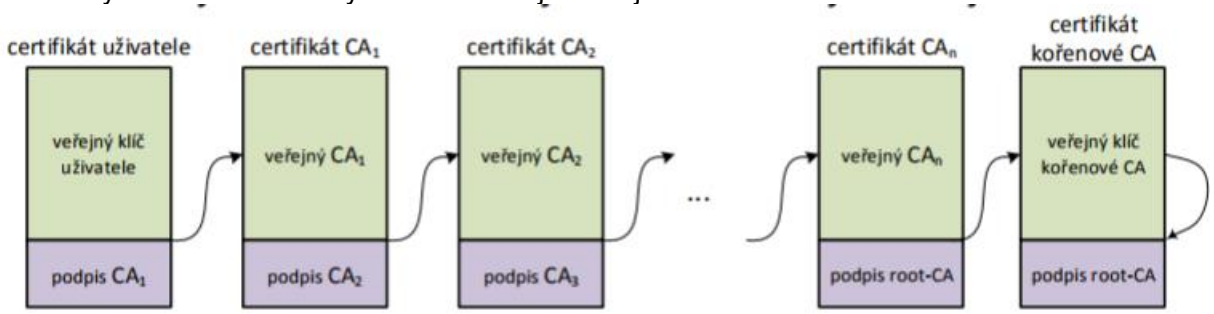
- X.509 certifikát struktura, princip,

- součást série doporučení ITU-T X.500 definujících globální adresářovou službu
- adresářová služba je systém spravující informace o uživatelích
- ve smyslu „telefonní seznam“ ne adresář souborů
- entita je jednoznačně identifikována pomocí význačného jména (DN - distinguished name)
- Struktura:
    - Version •
        - INTEGER (Hodnoty: 0, 1, 2 - dnes se používá verze 3 (hodnota 2))
    - Serial Number
        - INTEGER
        - CA musí zajistit jedinečnost (nesmí se opakovat).
        - Může být dlouhé, ale nesmí přesáhnout 20 bajtů.
        - CA je nemusí vydávat v rostoucí sekvenci
    - Signature Algorithm
        - Identifikátor algoritmu a jejich parametrů použitých CA při podpisu.
        - Musí být stejné jako položce Signature.
    - Validity
        - Specifikace časového intervalu, ve kterém CA ručí za informace na certifikátu.
        - Not Before – začátek platnosti
        - Not After – konec platnosti
    - Issuer a Subject
        - Skládá se z více položek a slouží k jednoznačné identifikaci (DN).
        - Příklad: Common Name, Organization, Organization Unit, Country…
    - Subject Public Key Info
        - Informace o klíči, který je tímto certifikátem svázán s předmětem (subject) certifikátu.
        - Obsahuje:
            - Algorithm – specifikace algoritmu
            - Subject Public Key – samotný klíč
    - Signature
        - Podpis samotného certifikátu od CA
        - Obsahuje:
            - Algorithm – specifikace algoritmu
            - Signature Value – samotný podpis
    - Extensions

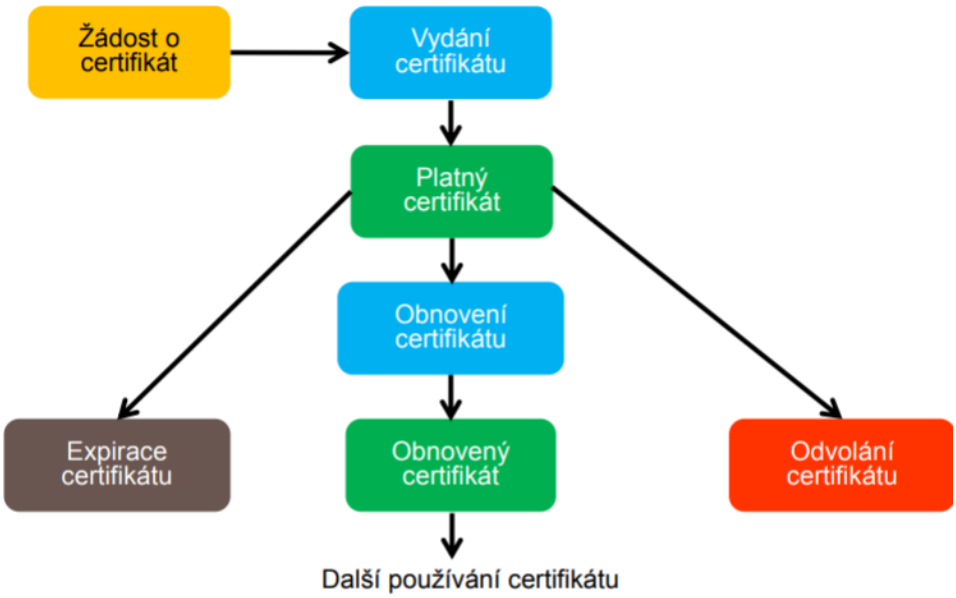


Certifikační autorita

- certifikační cesta, validace certifikační cesty

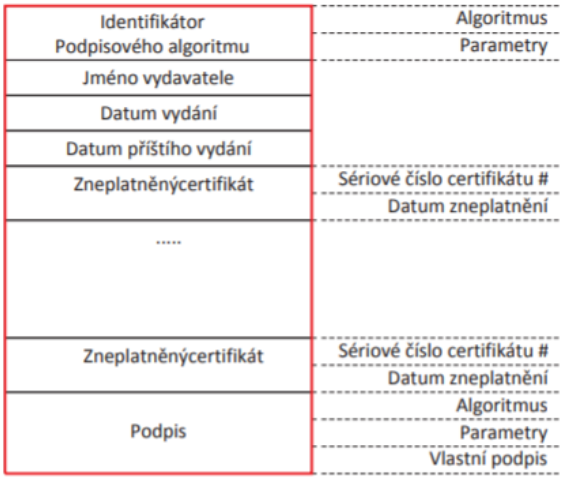
- validace certifikátu X.509.4 (ISO/IEC 9594-8) a RFC3280
- Konstrukce certifikační cesty zahrnuje vytvoření jedné (nebo více cest), které jsou:
  - 1) formálně správně zřetězeny
  - 2) vyhovují i dalším požadavkům (např. maximální přípustné délce cesty, omezením jmen nebo certifikační politiky)
- zřetězení jmen od důvěryhodné CA až k posuzovanému subjektu.
- hodnota atributu Subject Name v jednom certifikátu musí být shodná s hodnotou Issuer Name v následujícím certifikátu v cestě.
- Kořenový certifikát – hodnoty obou atributů jsou stejné
- 

- životní cyklus certifikátu



- kvalifikovaný certifikát

Kvalifikovaný certifikát (QC) – definován v čl. 3 odst. 15) v Nařízení Evropského parlamentu a Rady EU č. 910/2014 (eIDAS) jako certifikát pro elektronický podpis, který je vydán kvalifikovaným poskytovatelem služeb vytvářejících důvěr a splňuje požadavky stanovené v příloze I. Technicky stejný jako „běžný“ certifikát.



- CRL, OCSP

Seznam zneplatněných certifikátů (CRL)

- Umístění sériového čísla certifikátu na seznam zneplatněných certifikátů (Certificate Revocation List)
- Tento seznam je pravidelně zveřejňován a aktualizován
- Odkaz na umístění CRL může být součástí certifikátu
- Seznam je číslovaný, obsahuje datum vydání, nejpozdější datum vydání příštího seznamu a je podepsán CA
- Vydává se jednou za nějaký časový úsek (obvykle 1/den – ale i 1/rok).
- Vydává ho CA nebo jí pověřená autorita pro vydávání CRL.

Odvolání certifikátů – OCSP

- OCSP - Online Certification Status Protocol
- Protokol pro získání stavu certifikátu
- Možné odpovědi:
  - Good
  - Revoked
  - Unknown
- Okamžitá informace o stavu certifikátu
- Provozuje CA nebo jí pověřená autorita pro OCSP
- Alternativa k CRL
- Informace o OCSP serverech může být také přímo součástí certifikátu.
- RFC 2560, klient/server architektura

- Výhody OCSP oproti CRL :
    - Informaci o zneplatnění lze distribuovat prakticky všem okamžitě.
    - Řeší problém velkých PKI, kde je velký objem dat v CRL.
    - Šetří se objem dat přenášených v síti.
- Nevýhody OSCP oproti CRL:
    - Neexistence dlouhodobého důkazu o odvolání
    - Potencionální problém při LTV (Long Term Validation)
    - Není vždy dostupný (u některých CA jen za peníze)

- PKI

PKI – Public Key Infrastructure

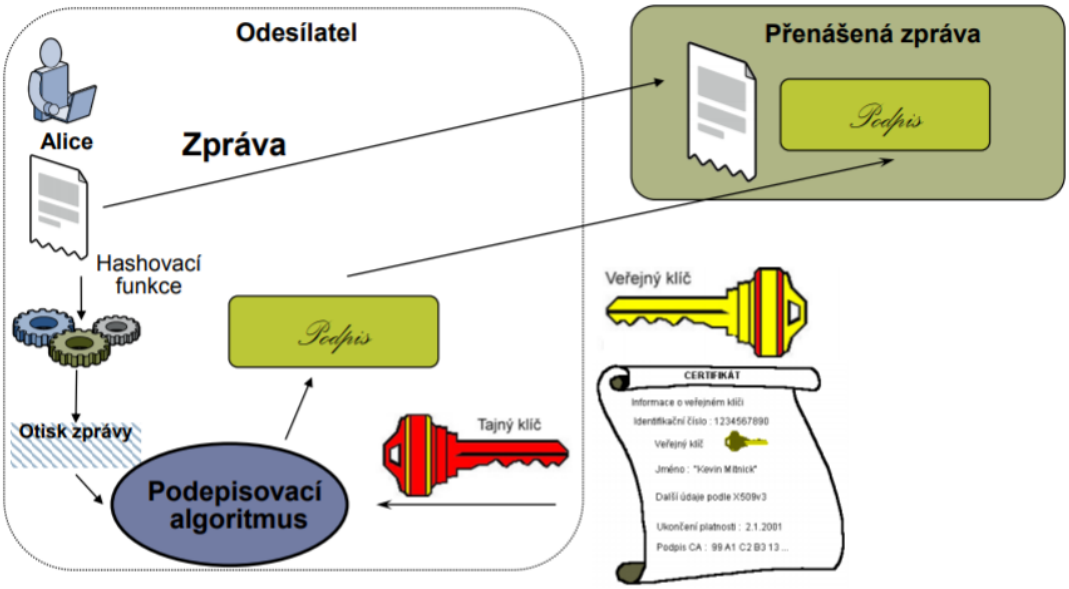
- Česky „Infrastruktura veřejných klíčů“
- PKI je souhrn znalostí, dohod, konvencí, technických postupů, organizačních principů, speciálního hardware a software, aplikací, standardů, norem, legislativy (zákony, prováděcí předpisy), osob a subjektů, které používají nebo se spoléhají na příslušné technologie (certifikáty X.509, elektronický podpis)….
- pojem PKI se často používá právě v souvislosti s elektronickým podpisem
- PKI poskytuje:
    - Autentizaci
    - ověření identity entity (osoba, služba, počítač)
    - pomocí digitálních certifikátů
    - Důvěrnost (utajení)
        - bezpečný přenos informací přes nezabezpečený komunikační kanál (Internet)
        - pomocí šifrovacích algoritmů
    - Integritu
        - pomocí hashovacích funkcí
        - schopnost detekce narušení dat během přenosu
    - Řízení přístupu
        - přístup k důvěrným informacím pouze oprávněným osobám
        - pomocí mechanismu veřejných a soukromých klíčů
    - Nepopiratelnost
- Základní části PKI:
    - Registrační autorita
        - Komunikace s žadatelem
        - Ověřování totožnosti
    - Certifikační autorita
        - Generuje (podepisuje) certifikáty
        - Pod pojem CA se často myslí celý komplex RA+CA+VA
    - Validační autorita
        - Udržuje adresář vydaných certifikátů.
        - Zodpovídá za publikované údaje.
        - Zajišťuje dostupnost certifikátů přes Internet (LDAP)

Elektronický podpis

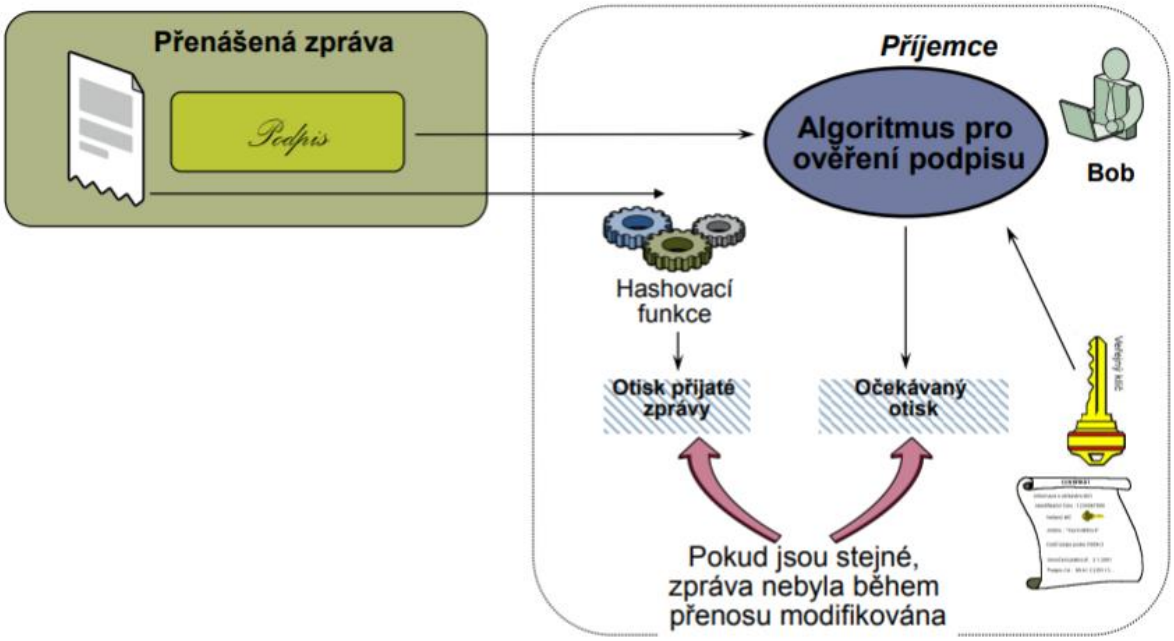
- princip

Obsah - horní část dokumentu

Vytvoření zaručeného EP



Ověření zaručeného EP



Obsah - horní část dokumentu

-EP, zaručený EP, kvalifikovaný EP

- **Elektronický podpis**
  - nejslabší varianta
- **Zaručený elektronický podpis**
  - To, co si většina lidí představí, když se řekne „elektronický podpis“
  - Zaručený elektronický podpis zajišťuje:
    - Integritu přenášených zpráv/dokumentů
    - Identifikace komunikujících stran
      - kdo je na druhé straně
    - Autentizace komunikujících stran
      - ověření předchozího zjištění
    - Nepopiratelnost (neodmítnutelnost)
- **Uznávaný elektronický podpis**
  - české specifikum – do 09/2018
  - zaručený EP založený na kvalifikovaném certifikátu
  - použitelný pro komunikaci s OVM (v ČR!) není podmínka na HW úložiště
- **Kvalifikovaný elektronický podpis**
  - Má stejnou platnost jako vlastnoruční podpis.
  - klíče musí být uloženy v „bezpečném“ prostředku

- elektronická pečeť

- technologicky totéž jako ZEP
- rozdíl je v právní rovině
  - EP používá pouze fyzická osoba
  - eln. pečeť může používat pouze právnická osoba nebo organizační složka státu
- dříve – elektronická značka
- ekvivalent úředního razítka
  - garantuje integritu a původ dokumentu
- kvalifikovaná pečeť
  - založená na kvalifikovaném eln. podpisu
  - ekvivalent kvalifikovaného EP
  - vyžaduje HW pro uložení soukromého klíče – HSM

- časové razítko

- Prokazuje existenci dokumentu v dané podobě a v daném čase
  - neřeší se, kdy dokument vznikl
  - neřeší, kdo dokument vlastnil
- Datová struktura obdobná certifikátu
- Technicky realizováno jako další elektronický podpis od časové autority
- Časová razítka vydává Autorita časových razítek
  - TSA – Time Stamp Authority
- Kvalifikovaná časová razítka
  - ekvivalent kvalifikované pečeti
- Elektronicky podepsaná struktura časového razítka mj. obsahuje:
  - jméno vydavatele
  - jedinečné sériové číslo razítka,
  - kontrolní součet (hash) z dokumentu a čas

HSM - princip, použití

Hardware security module

- Specializované vysoce bezpečné zařízení pro:
  - Generování silných kryptografických klíčů

        - Ochranu privátních klíčů
        - Poskytování šifrovacích služeb
        - Správu a archivaci kryptografických klíčů
    - Typické způsoby nasazení:
        - Sdílené pro více serverů
        - Nasazení v HA architektuře
    - Použití:
        - Zabezpečení klíčů pro kořenovou CA
        - Zabezpečení klíčů pro provádění el. podpisů v systémech
        - Archivace klíčů:
            - elektronická pošta – S/MIME
            - šifrování dat,
            - šifrování FS, disků,
        - Podpora šifrování v databázi
    - Zabezpečení:
        - Detekce narušení fyzické integrity zařízení
            - bezpečnostní samolepky
        - Odolnost vůči vniknutí
            - rozdělení zařízení na přístupnou (možný servis) a nepřístupnou část
            - neprůhledný epoxid, speciální šrouby
            - detekce otevření – senzory světla, teploty
        - Detekce SW narušení
            - testy integrity a automatickou reakcí (smazání dat) v případě detekce
- **elektronická značka**

SSH - princip, vlastnosti

S/MIME, PGP - principy, vlastnosti

Zabezpečení v bezdrátových sítích - pouze v 802.11

- WEP, WPA, WPA2

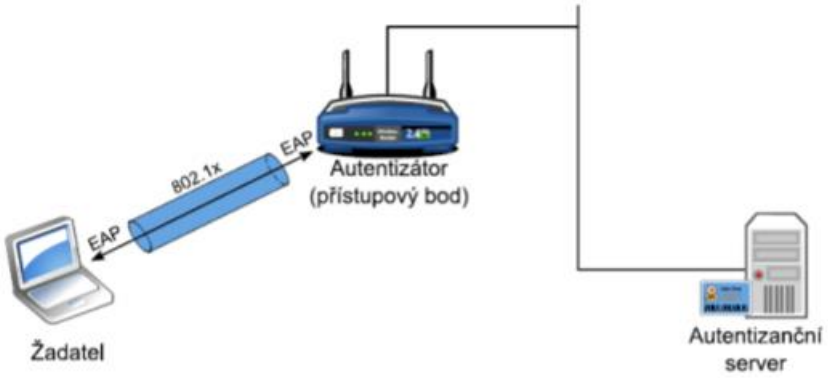
WEP – Wired Equivalent Privacy
- protokol zajišťující nepovinné zabezpečení sítí 802.11
- doslova „soukromí jako v drátových sítích“
- hlavní cíl: utajit přenášená data
- obsah datových rámců je zašifrován algoritmem RC4
- datové rámce jsou chráněny před změnou pomocí CRC-32 (Cyclic Redundancy Check)
- **volitelná autentizace** na základě znalosti sdíleného klíče
- široce podporovaný, ale snadno prolomitelný
- délka klíče 64 bitů (dle standardu, v praxi se používají i délky klíče 128 a 256 bitů)

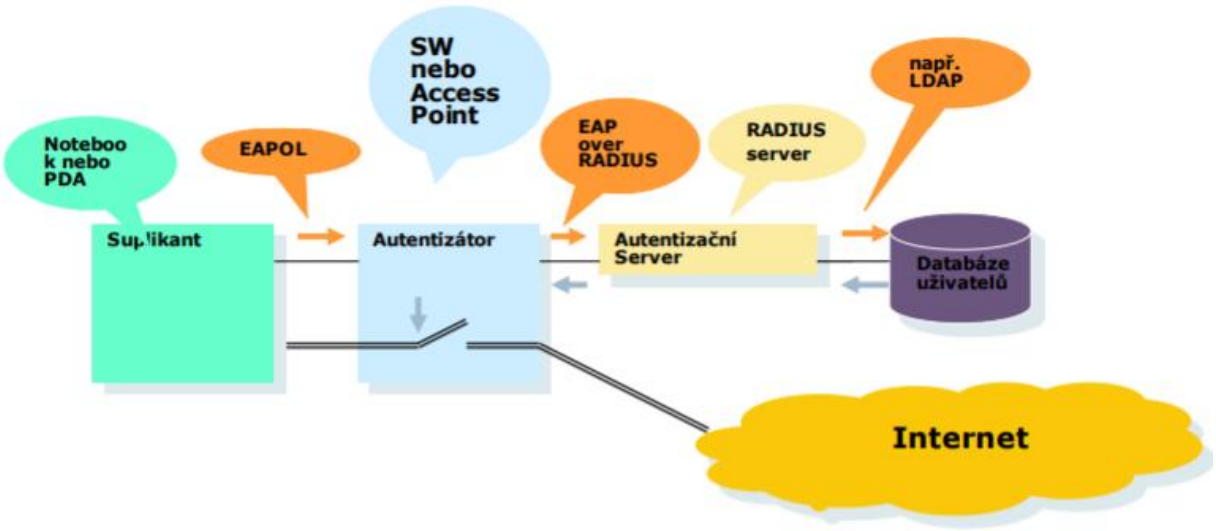
WPA – WiFi Protected Access
- publikován v roce 2002
- dočasné řešení od Wi-Fi Alliance před schválením standardu 802.11i (červen 2004)
- stejný algoritmus jako u WEPu – RC4
- autentizace podle 802.1x založená na protokolu EAP
- WEP nahrazen protokolem TKIP (Temporary Key Integrity Protocol)
    - na rozdíl od WEPu podporuje dynamické klíče
    - klíč se automaticky mění každých 10000 paketů
- k CRC32 přidán další zabezpečující kód MIC (Message Integrity Check, algoritmus Michael)
- kompatibilní se stávajícími zařízeními
    - podporu WPA šlo přidat upgradem firmware

- 802.11i
  - řeší komplexně problematiku zabezpečení v sítích IEEE 802.11
  - zpětná kompatibilita s WPA
    - WPA je podmnožinou WPA2
  - WPA-PSK - alternativní možnost autentizace pro malé sítě •
  - nový šifrovací algoritmus
    - AES v režimu CCMP
  - stávající zařízení nelze upgradovat softwarově
  - Broadcast Key Rotation
    - dříve pouze proprietární technologie Cisca

- autentizace - PSK, 802.1x

- Otevřená síť
- Autentizace pomocí MAC adresy
- Autentizace na VPN bráně
- Autentizace pomocí webového formuláře
- Autentizace pomocí WEPu (sdílený klíč)
- Autentizace pomocí WPA-PSK
  - PSK - Pre-shared Key (předsdílený klíč)
  - alternativa ke správě klíčů pomocí 802.1X
  - vhodné pro menší (domácí) sítě
  - PSK je 256 bitové číslo nebo hesla délky 8 - 63 znaků, která se pak na PSK převádí
  - náchylné na slovníkové útoky
    - nutno zvolit dostatečně dlouhou a složitou passphrase
  - PSK může být různý pro každé zařízení
    - v závislosti na MAC adrese
  - standardně výrobci používají jeden PSK pro celou síť
  - PMK = PBKDF2 (heslo, SSID, délka SSID, 4096, 256), kde PBKDF2 je metoda z PKCS #5 v2.0
- Autentizace pomocí IEEE 802.1X
  - obecný bezpečnostní rámec pro autentizaci v LAN Architekturu IEEE 802.1x tvoří tři základní funkční entity:
    - suplikant (žadatel) připojení k síti,
    - autentizátor zajišťující řízení přístupu,
    - autentizační server provádějící autorizační rozhodnutí.





- útoky

WEP – FMS, KoreK, PTW, Chopchop, padělání ARP, útok fragmentací paketů,
WPA/WPA2 (na PSK) – bruteforce, slovník, zachycení 4way handshake, PMK

Zabezpečení v mobilních sítích - GSM, UMTS

POUZE GSM
SIM obsahuje tajný klíč Ki , který je uložen i v AuC.
Algoritmy, které jsou používány pro šifrování v mobilní síti GSM:

- A3 - autentizační algoritmus
  - algoritmus pro autentizaci MS
  - Vstup: klíč Ki128b , náhodné číslo 128b
  - Výstup: otisk 32 b – SRES
  - takřka vždy realizován pomocí algoritmu COMP128
  - COMP128 byl součástí návrhu specifikace GSM sítě
- A5 - šifrovací algoritmus
- A8 - generátor klíče Kc pro hlasovou komunikaci
  - algoritmus pro generování klíčů pro šifrování rádiového rozhraní
  - většinou pomocí šifry COMP128
  - Vstup: klíč Ki 128b, náhodné číslo RAND
  - Výstup: klíč Kc 64b,
  - Kc je používán dokud MSC nerozhodne o reautentizaci zařízení
- COMP128 není veřejný
  - princip odhalen při používání SIM karet a později unikla kompletní specifikace (1998)
- GSM shrnutí:
  - anonymita **– dobrá**
  - autentizace **– špatná**
    - pouze jednosměrná – možnost existence fake-BTS
    - neexistuje schéma pro výměnu klíčů
    - předsdílený klíč (Ki)
  - **šifrování – špatné**
    - krátký klíč - 54b
    - slabý algoritmus – lze prolomit (rainbow tables)

        - data pro autentizaci včetně klíče KC jdou přenášena v otevřeném tvaru mezi mobilními sítěmi
        - uživatel nepozná jestli je hovor šifrován, nebo ne
    - integrita - **žádná**