Bezpečnost

7. Proudové šifry, blokové šifry, DES, 3DES, AES, operační módy

doc. Ing. Róbert Lórencz, CSc.

České vysoké učení technické v Praze
Fakulta informačních technologií
Katedra počítačových systémů

Příprava studijních programů Informatika pro novou fakultu ČVUT je spolufinancována Evropským sociálním fondem a rozpočtem Hlavního města Prahy v rámci Operačního programu Praha — adaptabilita (OPPA) projektem CZ.2.17/3.1.00/31952 – „Příprava a zavedení nových studijních programů Informatika na ČVUT v Praze“.
Praha & EU: Investujeme do vaší budoucnosti

Obsah přednášky

- Proudové šifry
- Blokové šifry
- DES, 3DES
- AES
- Operační mody

Proudové šifry (12)

Dvojí použití hesla u aditivních šifer

- U proudových šifer, kde každá parciální substituce je posunem otevřeného textu v abecedě A o h_i, postačí k luštění 2 ŠT (c, c'), šifrované tímtéž heslem.
- Mějme $c_i = |m_i + h_i|_q$ a $c_i' = |m_i' + h_i|_q$.
- Odečtením ŠT od sebe eliminujeme heslo a dostáváme $|c_i - c_i'|_q = |m_i - m_i'|_q$
- Z posloupnosti $|m_i - m_i'|_q$ pro $i = 0, 1, \ldots$ pak původní texty m a m' luštíme jako při použití knižní šifry, kde v roli původního otevřeného textu vystupuje m a v roli knižního hesla m'.
- Někdy se používá metoda předpokládaného slova, kdy za m' zkoušíme nějaké slovo postupně od první do poslední pozice, dopočteme m' a sledujeme, zda dává smysl.

Proudové šifry (13)

Použití

- Linkové šifrátory – do komunikačního kanálu přicházejí jednotlivé znaky v pravidelných nebo nepravidelných časových intervalech $\Rightarrow$ v daném okamžiku je nutné tento znak okamžitě přenést.
- Příklad tzv. terminálového spojení, kdy jsou spojeny dva počítače, přičemž to, co uživatel píše na klávesnici na jedné straně, se objevuje na monitoru počítače druhého uživatele.
- Šifrovací zařízení má omezenou paměť na průchozí data.
- Výhodou proudových šifer oproti blokovým je malá "propagace chyby". Vzniklá chyba na komunikačním kanálu v jednom znaku ŠT se projeví u proudových šifer pouze v jednom odpovídajícím znaku otevřeného textu, u blokové šifry má vliv na celý blok znaků.

Proudové šifry (14)

Synchronní a asynchronní proudové šifry

- Pokud proud hesla nezávisí na OT ani ŠT $\Rightarrow$ synchronní proudové šifry $\Rightarrow$ příjemce a odesílatel je přesně synchronizován (jinak výpadek jednoho znaku ŠT naruší veškerý následující OT).
- Šifry eliminující takové chyby se nazývají asynchronní nebo samosynchronizující se šifry. U nich dojde v krátké době k synchronizaci a správné dešifraci zbývajícího OT.
- To se může docílit například tím, že proud hesla je generován pomocí klíče a n předchozích znaků ŠT: $h_i = f(k, c_{i-n}, \ldots, c_{i-1})$.
- K synchronizaci dojde, jakmile se přijme souvislá posloupnost $n+1$ správných znaků ŠT.
- Historická asynchronní šifra – Vigenèrův autokláv.
- Heslo je pouze jedno písmeno klíče, znaky hesla byly tvořeny už přímo předchozím znakem ŠT (sčítání v modulu 26): $c_1 = p_1 + h_1$, kde $h_1 = k$ a $c_i = p_i + h_i$, $i = 2, 3, \ldots$, kde $h_i = c_{i-1}$.

Blokové šifry (1)

Bloková šifra

- Nechť A je abeceda q symbolů, $t \in N$ a $M = C$ je množina všech řetězců délky t nad A. Nechť K je množina klíčů.
- Bloková šifra je šifrovací systém (M, C, K, E, D), kde E a D jsou zobrazení, definující pro každé $k \in K$ transformaci zašifrování E_k a dešifrování D_k tak, že zašifrování bloků OT $m_1, m_2, m_3, \ldots$, kde $m_i \in M$ pro každé $i \in N$, probíhá podle vztahu $c_i = E_k(m_i)$ pro každé $i \in N$ a
- dešifrování probíha podle vztahu $m_i = D_k(c_i)$ pro každé $i \in N$.
- Pro blokovou šifru je podstatné, že všechny bloky OT jsou šifrovány toutéž transformací a všechny bloky ŠT jsou dešifrovány toutéž transformací.
- Za určitých okolností můžeme za blokové šifry považovat šifry substituční a transpoziční.

Blokové šifry (2)

Bloková šifra

- Nejznámější blokové šifry používaly a používají blok o délce 64b: DES (Data Encryption Standard), TripleDES, IDEA, CAST aj.,
- v současné době se přechází na blok 128 bitů, který používá standard AES.
- Blokové šifry využívají principy algoritmů Feistelova typu umožňující postupnou aplikací relativně jednoduchých transformací na bázi nelineárních posuvných registrů vytvořit složitý kryptografický algoritmus.
- Tento přístup je využíván také v jiných oblastech. Např. u zabezpečovacích kódů jsou využívané tzv. zřetězené kódy, které ze 2 relativně jednoduchých standardních zabezpečovacích kódů vytvářejí výkonné zabezpečovací kódy.
- Tento princip je využívaný v symetrických šifrovacích algoritmech, nejznámějším algoritmem tohoto typu je algoritmus DES.

Blokové šifry (3)

Základní principy algoritmů Feistelova typu (1)

Příklad: Mějme 2 permutace

$$f_1 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 2 & 4 \end{pmatrix} \qquad f_2 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix},$$

kde zápis funkce f_1 znamená, že 1. bit se přesune na 3. pozici, 2.bit na 1. pozici, 3. bit na 2. pozici, atd. Dále nechť máme následující způsob šifrování.

Původní 8b zprávu $m \in M$ rozdělíme na 2 4b bloky $m = (m_0\ m_1) \Rightarrow$ vytvoříme novou 8b zprávu $c_1 = (m_1 m_2)$, kde $m_2 = m_0 \oplus f_1(m_1)$ a tento postup ještě jednou zopakujeme, tj. vytvoříme $c_2 = (m_2\ m_3)$, kde $m_3 = m_1 \oplus f_2(m_2)$.

Pro $m = \texttt{1001 1101}$ dostáváme postupně:
$c_1 = (\texttt{1101 1110})$, kde $m_2 = \texttt{1001} \oplus f_1(\texttt{1101}) = \texttt{1001} \oplus \texttt{0111} = \texttt{1110}$
$c_2 = (\texttt{1110 0000})$, kde $m_3 = \texttt{1101} \oplus f_2(\texttt{1110}) = \texttt{1101} \oplus \texttt{1101} = \texttt{0000}$

Blokové šifry (4)

Základní principy algoritmů Feistelova typu (2)

Dešifrování je možné realizovat opačným postupem:
$m_1 = m_3 \oplus f_2(m_2)$ a $m_0 = m_2 \oplus f_1(m_1)$, přičemž dešifrovaná hodnota je $m = \texttt{1001 1101}$.

Pro $c_2 = (m_2\ m_3) = \texttt{1110 0000}$ dostáváme:
$m_1 = (\texttt{0000 1110} \oplus \texttt{f}_2(\texttt{1110}) = \texttt{0000} \oplus \texttt{0101} = \texttt{1101}$.

A pro m_0 dostáváme:
$m_0 = m_2 \oplus f_1(m_1) = \texttt{1110} \oplus f_1(\texttt{1101}) = \texttt{1110} \oplus \texttt{0111} = \texttt{1001}$ a $\Rightarrow$
$m = (m_0\ m_1) = \texttt{1001 1101}$

Z předcházejících příkladů je zřejmé, že funkce f_1 a f_2 nemusí být prosté, protože není potřebné počítat jejich inverzi. Všech možných funkcí $f : V_4 \to V_4$ je $(2^4)^{2^4} = 16^{16} \doteq 1.85 \cdot 10^{19}$.

Blokové šifry (5)

Základní principy algoritmů Feistelova typu (3)

- Ne všechny funkce f jsou pro šifrování vhodné
- Uvedené způsoby šifrování jsou speciálním případem tzv. Feistelových kryptosystémů.

Definice – Feistelův kryptosystém

Nechť množina zpráv M je složená z všech možných $2n$-tic V_{2n} a prostor klíčů tvoří všechny možné h-tice funkci $k = f_1, f_2, \ldots, f_h$, $f_i : V_n \to V_n$ pro každé $i = 1, 2, \ldots, h$ a prostor zašifrovaných textů $C = V_{2n}$. Zobrazení $T_k : K \times V_{2n} \to V_{2n}$, definované rekurentně vztahy

$$\begin{aligned} m_{i+1} &= m_{i-1} + f_i(m_i), \quad \text{pro} \quad i = 1, 2, \ldots, h \\ T_k(m) &= (m_h \; m_{h+1}), \end{aligned}$$

kde $m = (m_0 \; m_1) \in M$, definuje Feistelův kryptosystem.

DES (1)

Algoritmus DES

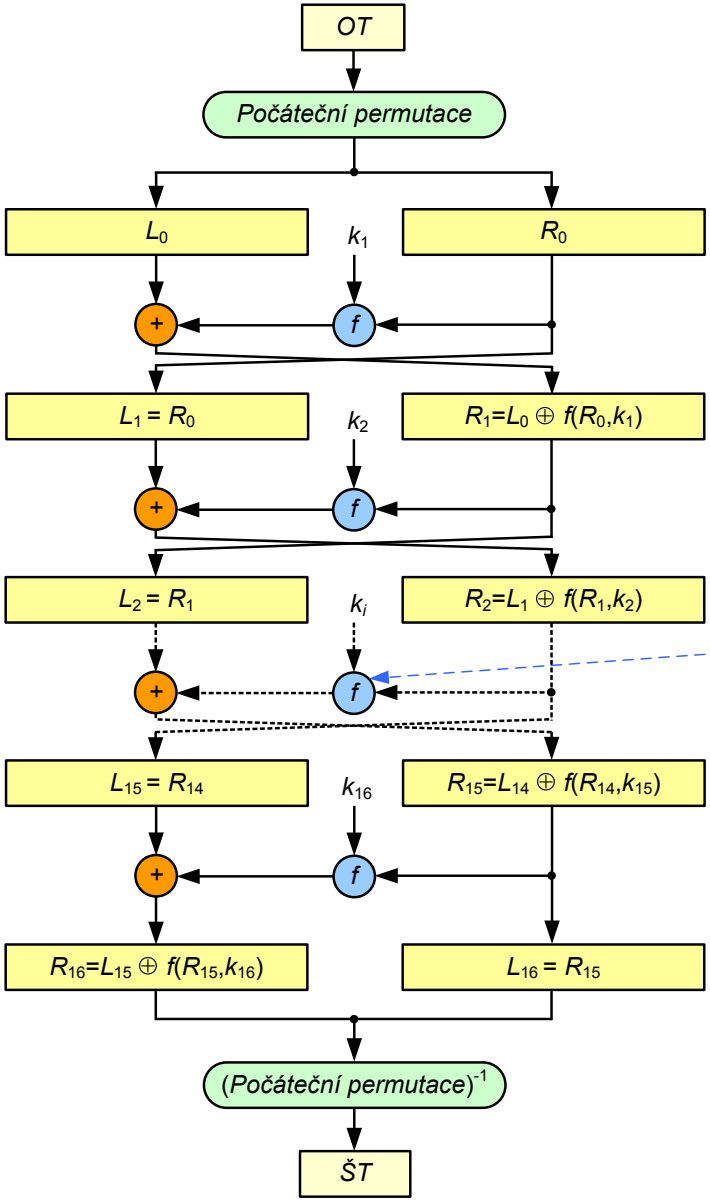
- Veřejná soutěž (1977): šifrovací standard (FIPS 46-3) v USA pro ochranu citlivých, ale neutajovaných dat ve státní správě.
- Součást průmyslových, internetových a bankovních standardů.
- 1977: varování – příliš krátký klíč 56b, který byl do původního návrhu IBM zanesen vlivem americké tajné služby NSA.
- DES – intenzivní výzkum a útoky $\Rightarrow$ objeveny teoretické negativní vlastnosti jako: tzv. slabé a poloslabé klíče, komplementárnost a teoreticky úspěšná lineární a diferenciální kryptoanalýza.
- V praxi jedinou zásadní nevýhodou je pouze krátký klíč.
- 1998: stroj – DES-Cracker, luštící DES hrubou silou.
- DES jako americký standard skončil (jen v "dobíhajících"systé-mech a kvůli kompatibilitě) a místo něj: Triple-DES, (FIPS 46-3).
- Od 26. 5. 2002 – šifrovací standard nové generace AES.

DES (2)

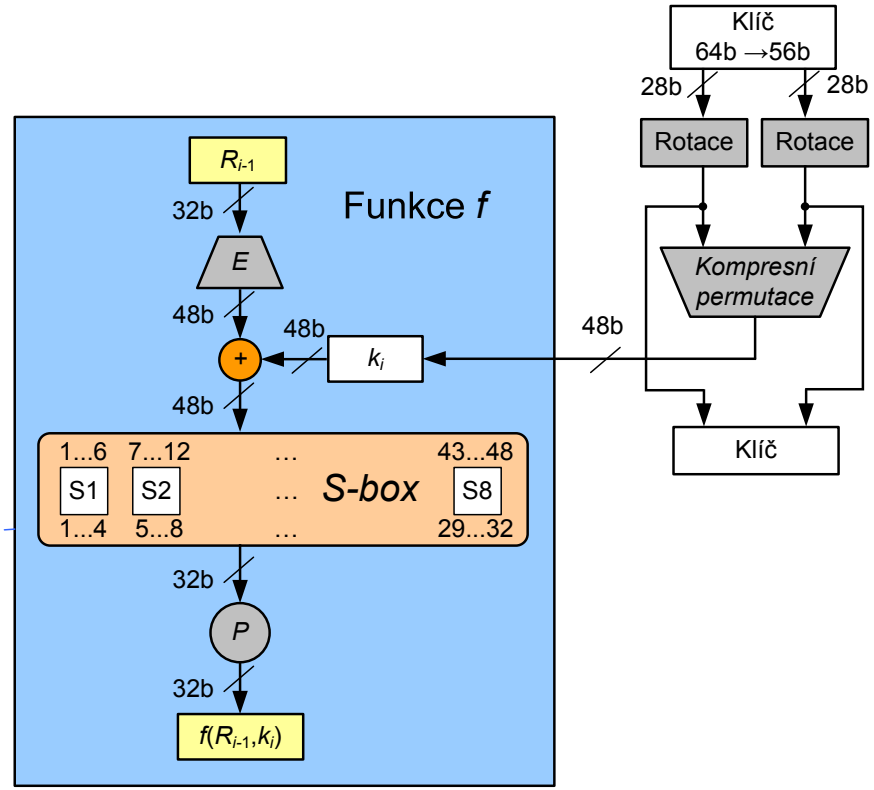
Stavební části DES (1)

- DES je iterovaná šifra typu $E_{k_{16}}(E_{k_{15}}(\ldots(E_{k_1}(m_i)\ldots)))$.
- Používá 16 rund a 64b bloky OT a ŠT. Šifrovací klíč k má délku 56 b (vyjadřuje se ale jako 64b číslo, kde každý 8. bit je bit parity)
- 56b klíč k je v inicializační fázi nebo za chodu algoritmu expandován na 16 rundovních klíčů k_1 až k_{16}, které jsou řetězci 48 bitů, každý z těchto bitů je některým bitem původního klíče k.
- Místo počátečního zašumění OT se používá bezklíčová pevná permutace *Počáteční permutace* a místo závěrečného zašumění permutace k ní inverzní $(\textit{Počáteční permutace})^{-1}$.
- Po *počáteční permutaci* je blok rozdělen na dvě 32b poloviny (L_0, R_0). Každá ze 16 rund $i = 1, 2, \ldots, 16$ transformuje (L_i, R_i) na novou hodnotu $(L_{i+1}, R_{i+1}) = (R_i, L_i \oplus f(R_i, k_{i+1}))$, liší se jen použitím jiného rundovního klíče k_i.
- Ve smyslu definice Feistelova kryptosystému je v tomto případě $h = 16$ a $2n = 64$.

DES (3)



Algoritmus DES



DES (4)

Stavební části DES (2)

- Po 16. rundě dochází ještě k výměně pravé a levé strany: $(L_{16}, R_{16}) = (R_{15}, L_{15} \oplus f(R_{15}, k_{16}))$ a závěrečné permutaci $(\textit{Počáteční permutace})^{-1}$.
- Dešifrování probíhá stejným způsobem jako zašifrování, pouze se obrátí pořadí výběru rundovních klíčů.

Rundovní funkce *f*

- Rundovní funkce se skládá z binárního načtení klíče k_i na vstup. 48b klíč k_i je vytvořen po kompresi ze 2 28b rotovaných částí původního klíče k, kde počet bitů rotace je závislý na čísle rundy.
- Tento klíč k_i je dál xorován s expandovanou 32b částí R_{i-1}, která je expanzně permutovaná v bloku E z 32b na 48b. Tato operace kromě rozšíření daného 32b slova také permutuje bity tohoto slova tak, aby se dosáhlo lavinového efektu.

DES (5)

Stavební části DES (3)

- Následně je prováděna pevná, nelineární substituce na úrovni 6b znaků do 4b znaků s následnou transpozicí na úrovni bitů. Těmito operacemi se dosahuje dobré difúze i konfúze.
- Použité substituce se nazývají substituční boxy: S-boxy, jsou jediným nelineárním prvkem schématu. Pokud bychom substituce vynechali, mohli bychom vztahy mezi ŠT, OT a klíčem popsat pomocí operace binárního sčítání $\oplus$, tedy lineárními vztahy.
- Tato nelinearita je překážkou jednoduchého řešení rovnic, vyjadřující vztah mezi OT, ŠT a K.
- Následně 32b výsledné slovo z S-boxu je permutováno c bloku P. Tato permutace převádí každý vstupní bit do výstupu, kde žádbý vstupní bit se nepoužije $2\times$.
- Nakonec se výsledek permutace sečte modulo 2 s levou 32b polovinou a začne další runda.

TripleDES (1)

TripleDES

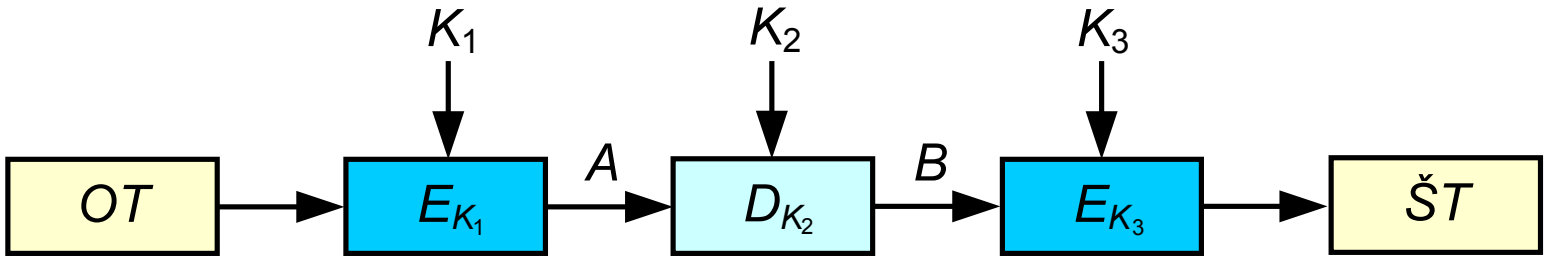
- TripleDES (3DES) prodlužuje klíč originální DES tím, že používá DES jako stavební prvek celkem 3× s 2 nebo 3 různými klíči.
- Nejčastěji se používá varianta EDE této šifry, která je definována ve standardu FIPS PUB 46-3 (v bankovní normě X9.52).
- Vstupní data OT jsou zašifrována podle vztahu $ŠT = E_{K_3}(D_{K_2}(E_{K_1}(OT)))$, kde K_1, K_2 a K_3 jsou buď 3 různé klíče nebo $K3 = K1$. Varianta EDE byla zavedena z důvodu kompatibility → při rovnosti všech klíčů 3DES = DES.
- Klíč 3DES je tedy buď 112 bitů (2 klíče) nebo 168 bitů (3 klíče). 3DES je spolehlivá → klíč je dostatečně dlouhý a teoretickým slabinám (komplementárnost, slabé klíče) se dá předcházet ⇒
- 3DES a AES → platný oficiální standard nahrazující DES.
- 3DES lze, jako jakoukoliv jinou blokovou šifru, použít v různých operačních modech (CBC mod ⇒ 3DES-EDE-CBC).

TripleDES (2)

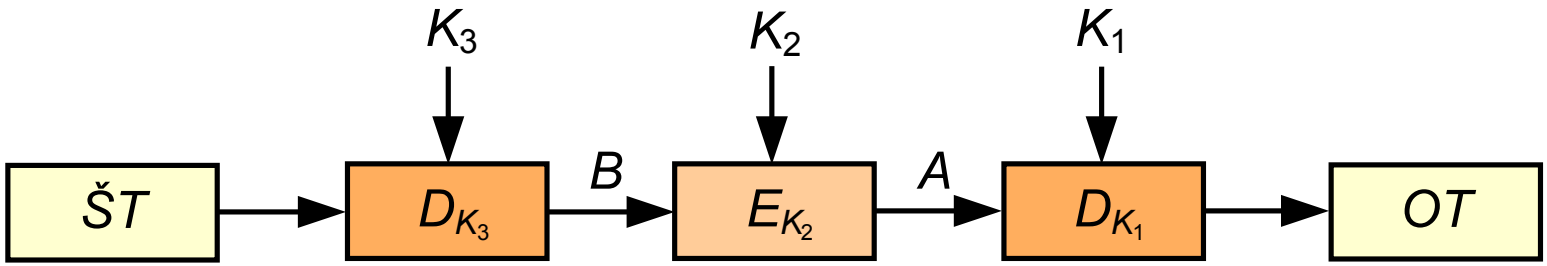
Algoritmus 3DES

Šifrování

3DES-EDE



Dešifrování



$K_1 \neq K_2 \neq K_3 \Rightarrow$ 3 x klíč = 128b $\rightarrow$ 3DES

$K_1 = K_3 \neq K_2 \Rightarrow$ 2 x klíč = 112b $\rightarrow$ 3DES

$K_1 = K_2 = K_3 \Rightarrow$ 1 x klíč = 56b $\rightarrow$ DES

AES (1)

- Po útocích hrubou silou na DES, americký standardizační úřad připravil náhradu - Advanced Encryption Standard (AES).
- 2.1. 1997 výběrové řízení na AES – 15 kandidátů.
- Z 5 finalistů byl vybrán algoritmus Rijndael [rájndol] (autoři J. Daemen a V. Rijmen).
- Jako AES byl přijat s účinností od 26. května 2002 a byl vydán jako standard v oficiální publikaci FIPS PUB 197.
- AES je bloková šifra s délkou bloku 128 bitů, čímž se odlišuje od současných blokových šifer, které měly blok 64 bitový.
- AES podporuje tři délky klíče: 128, 192 a 256 bitů $\Rightarrow$ se částečně mění algoritmus (počet rund je po řadě 10, 12 a 14).
- Větší délka bloku a klíče zabraňují útokům, které byly aplikované na DES. AES nemá slabé klíče, je odolný proti známým útokům a metodám lineární a diferenciální kryptoanalýzy.

AES (2)

- Algoritmus zašifrování i odšifrování se dá výhodně programovat na různých typech procesorů, má malé nároky na paměť i velikost kódu a je vhodný i pro paralelní zpracování.
- AES bude pravděpodobně platným šifrovacím standardem několik desetiletí a bude mít obrovský vliv na počítačovou bezpečnost.
- Označíme-li délku klíče N_k jako počet 32b slov, máme $N_k = 4, 6$ a 8 pro délku klíče 128, 192 a 256 bitů.
- AES je iterativní šifra, počet rund N_r se mění podle délky klíče: $N_r = N_k + 6$, tj. je to 10, 12 nebo 14 rund.
- Tato skutečnost odráží nutnost zajistit konfúzi vzhledem ke klíči. Algoritmus pracuje s prvky Galoisova tělesa $GF(2^8)$ a s polynomy, jejichž koeficienty jsou prvky z $GF(2^8)$. Bajt s bity $(b_7, ..., b_0)$ je proto chápán jako polynom $b_7 x^7 + \cdots + b_1 x^1 + b_0$ a operace "násobení bajtů"odpovídá násobení těchto polynomů modulo $m(x) = x^8 + x^4 + x^3 + x^1 + 1$.

AES (3)

Rundovní klíče

- Rundovní klíče AES využívá $4 + N_r \times 4$ rundovních 32b klíčů, které se definovaným způsobem derivují ze šifrovacího klíče.
- Před zahájením 1. rundy zašifrování se provede úvodní zašumění, kdy se na OT naxorují první 4 rundovní klíče (128b na 128b)$\Rightarrow$
- N_r shodných rund (s výjimkou poslední, kdy se neprovede operace MixColumns), při kterých výstup z každé předchozí rundy slouží jako vstup do rundy následující. Tím dochází k postupnému mnohonásobnému zesložiťování výstupu.

Runda (1)

- Na počátku každé rundy se vždy vstup (16 B) naplní postupně zleva doprava a shora dolů po sloupcích do matice 4x4 B $\mathbf{A} = (a_{ij})$ $i, j = 0, 1, 2, 3$.
- Na každý bajt matice **A** se zvlášť aplikuje substituce, daná pevnou substituční tabulkou SubBytes.

AES (4)

Runda (2)

- Řádky matice **A** cyklicky posunou postupně o 0-3 bajty doleva, operace ShiftRows, 1. řádek o 0, druhý o 1, třetí o 2 a čtvrtý o 3, čímž dochází k transpozici na úrovni bajtů.
- Dále se na každý jednotlivý sloupec matice aplikuje operace MixColumns, která je substitucí 32 bitů na 32 bitů. Tuto substituci lze však popsat lineárními vztahy – všechny výstupní bity jsou nějakou lineární kombinací vstupních bitů. Označíme-li jednotlivé bajty v rámci daného sloupce matice **A** (shora dolů) jako a_0 až a_3, pak výstupem budou jejich nové hodnoty b_0 až b_3, podle vztahů

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} '02' & '03' & '01' & '01' \\ '01' & '02' & '03' & '01' \\ '01' & '01' & '02' & '03' \\ '03' & '01' & '01' & '02' \end{pmatrix} \times \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

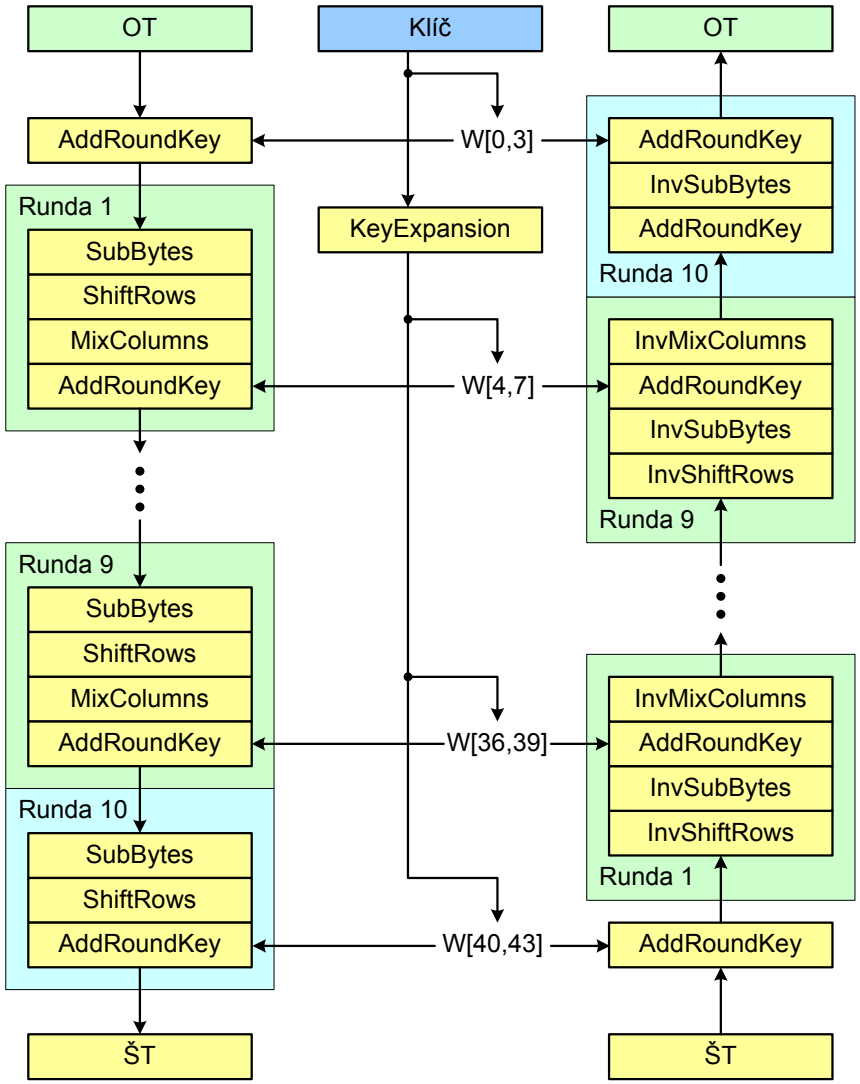
AES (5)

Runda (3)

- Násobení je násobení prvků $GF(2^8)$. Konstantní prvky tohoto pole jsou vyjádřeny hexadecimálně.
- Jako poslední operace rundy se vykoná transformace AddRoundKey, v rámci níž se na jednotlivé sloupce matice **A** zleva doprava naxorují 4 odpovídající rundovní klíče. Tím je jedna runda popsána a začíná další. Po poslední rundě se ŠT jen vyčte z matice **A**.
- Při odšifrování se používají operace inverzní k operacím, použitým při zašifrování, neboť všechny jsou reverzibilní.
- Nelinearity v AES se objevují pouze v substituci SubBytes. V roce 2002 bylo zjištěno, že vzájemné vztahy výstupních (y1,..,y8) a vstupních (x1,..,x8) bitů lze popsat implicitními rovnicemi f(x1,..,x8,y1,..,y8) = 0 pouze druhého řádu.

Algoritmus AES

AES – Struktura šifrování a dešifrování



Operační mody blokových šifer (1)

- Operační mody blokových šifer jsou způsoby použití blokových šifer v daném kryptosystému, kde OT není jen 1 blok blokové šifry, ale obecně posloupnost znaků dané abecedy.
- U moderních blokových šifer chápeme jako znaky bajty, i když se délka bloku N uvádí v bitech. Obvykle $N = 64$ nebo 128. Pomocí operačních modů můžeme získat nové zajímavé vlastnosti a využití blokových šifer. Mody: ECB, CFB, OFB, CBC, CTR a MAC.

Mod ECB (Electronic Codebook)

- Tento operační modus se nazývá elektronická kódová kniha a je základním modem.
- Posloupnost bloků otevřeného textu $OT_1, OT_2, \ldots OT_n$ se šifruje tak, že každý blok je šifrován zvlášť, což lze vyjádřit vztahem $\check{S}T_i = E_K(OT_i)$, $i = 1, 2, \ldots n$.

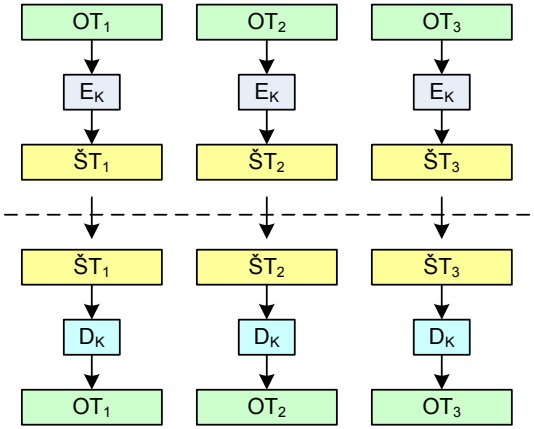
Operační mody blokových šifer (2)

- Nevýhodou takového typu šifrování je, že stejné bloky OT mají vždy stejný šifrový obraz.
- Pokud nalezneme několik shodných bloků $\Rightarrow$ to může v určitém kontextu dokonce rozkrývat i hodnotu otevřeného bloku (například prázdné sektory na disku jsou vyplněny hodnotou 0xFF apod.
- Integrita a modus ECB – Ve zprávě šifrované modem ECB může útočník bloky ŠT vyměňovat, vkládat nebo vyjímat, a tak snadno docilovat pro uživatele nežádoucích změn v OT, zejména, pokud nějakou dvojici (OT, ŠT) už zná.
- Tím je opět ilustrován v praxi často opomíjený fakt, že integrita OT se šifrováním nezajistí.

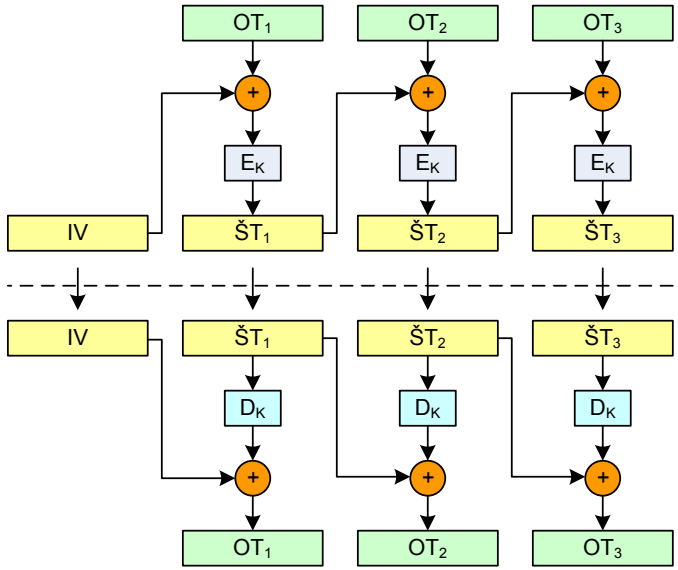
Synchronní proudové šifry mají nedostatečnou vlastnost difúze neboť pracují jen nad jednotlivými znaky abecedy. Moderní blokové šifry naproti tomu dosahují velmi dobrých vlastností jak difúze, tak konfúze.

Operační mody blokových šifer (3)

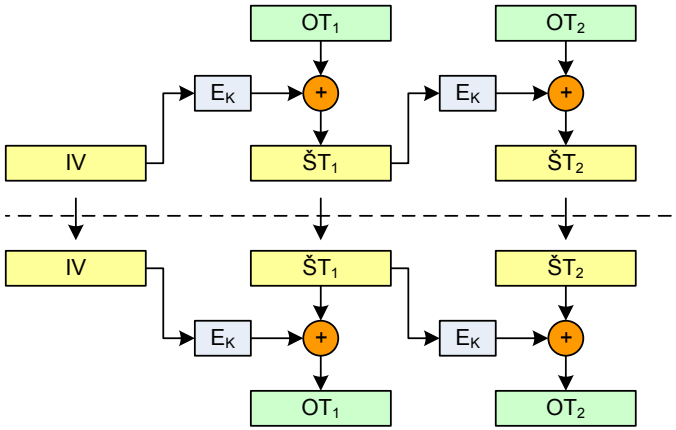
ECB – Electronic Code Book



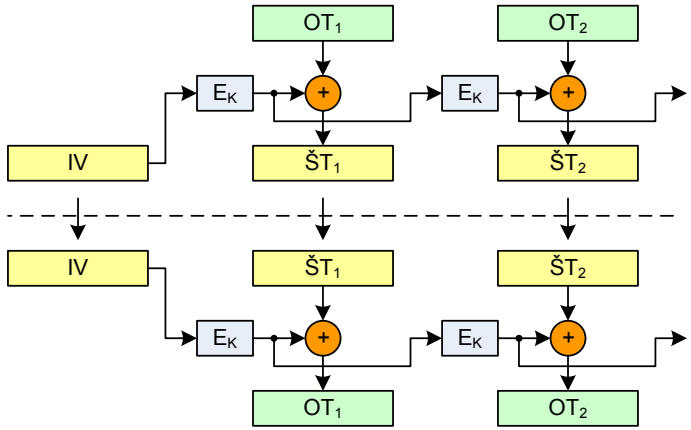
CBC – Cipher Block Chaining



CFB – Cipher FeedBack



OFB – Output FeedBack



Operační mody blokových šifer (4)

Mod CBC (Cipher Block Chaining)

- Pro rozšíření difúzi na více bloků byl definován modus řetězení šifrového textu (CBC). Každý blok OT se v něm nejprve modifikuje předchozím blokem ŠT, a teprve poté se šifruje.
- To zajišťuje, že běžný šifrový blok závisí na celém předchozím OT z důvodu řetězení této závislosti přes předchozí ŠT.
- CBC je nejpoužívanějším operačním modem blokových šifer. Eliminuje některé slabosti modu ECB a zajišťuje difúzi celého předchozího OT do daného bloku ŠT.
- 1. blok OT je modifikován náhodnou hodnotou IV. Šifrování se provádí podle vztahů

$$\check{S}T_0 = IV, \check{S}T_i = E_K(OT_i \oplus \check{S}T_{i-1}), \quad i = 1, 2, \ldots, n$$

a dešifrování podle vztahu

$$\check{S}T_0 = IV, OT_i = \check{S}T_{i-1} \oplus D_K(\check{S}T_i), \quad i = 1, 2, \ldots, n$$

Operační mody blokových šifer (5)

- Náhodný IV způsobí, že budeme-li šifrovat jeden a tentýž, byť i velmi dlouhý, OT 2×, obdržíme naprosto odlišný ŠT.
- Řetězení mírně znesnadňuje útoky v porovnání s ECB.
- Z definice modu CBC však vyplývá vlastnost samosynchronizace. Proces odšifrování je schopen se zotavit a produkovat správný OT už při 2 za sebou jdoucích správných blocích ŠT ($\check{S}T_{i-1}$ a $\check{S}T_i$).

Mody CFB a OFB (Cipher Feedback, Output Feedback)

- Převádí blokovou šifru na proudovou. Inicializační hodnota IV nastavuje konečný automat do náhodné polohy.
- Automat produkuje posloupnost hesla, které se jako u proudových šifer *xoruje* na OT. 1. blok hesla se získá zašifrováním IV.
- Vzniklé heslo (OFB) nebo vzniklý ŠT (CFB) se přivádějí na vstup blokové šifry a jejich zašifrováním je získán další blok hesla.
- OFB má vlastnost čisté (synchronní) proudové šifry – heslo je generováno zcela autonomně bez vlivu OT a ŠT.

Operační mody blokových šifer (6)

- CFB je kombinací vlastností CBC a proudové šifry.
- Předpis pro zašifrování v CFB:

$$\check{S}T_0 = IV, \check{S}T_i = OT_i \oplus E_K(\check{S}T_{i-1}), \quad i = 1, 2, \ldots, n$$

a dešifrování v modu CFB:

$$\check{S}T_0 = IV, OT_i = \check{S}T_i \oplus E_K(\check{S}T_{i-1}), \quad i = 1, 2, \ldots, n$$

- Předpis pro zašifrování v OFB:

$$H = IV = \check{S}T_0, \{\check{S}T_i = OT_i \oplus H, H = E_K(H)\}, \quad i = 1, 2, \ldots, n$$

a dešifrování v modu OFB:

$$H = IV = \check{S}T_0, \{OT_i = \check{S}T_i \oplus H, H = E_K(H)\}, \quad i = 1, 2, \ldots, n$$

- V modech OFB a CFB se bloková šifra používá jen jednosměrně, tj. jen transformace $E_K \Rightarrow$ výhodné při HW realizaci.

Operační mody blokových šifer (7)

- Jako výstup lze použít část bloku hesla/ŠT, např. b bitů $\Rightarrow$ se b bitů hesla (OFB) nebo b bitů vzniklého ŠT (CFB) vede zprava do vstupního registru, přičemž původní obsah vstupního registru se posune doleva o b bitů (b bitů nejvíce vlevo z registru vypadne).
- CFB je samosynchronní, a to podle délky zpětné vazby. Je-li b bitů, pak postačí 2 nenarušené b-bitové bloky ŠT, aby se OT sesynchronizoval.
- OFB $\rightarrow$ synchronní proudová šifra. Heslo generuje konečným automatem, který má maximálně 2^N vnitřních stavů $\Rightarrow$
- se produkce hesla musí opakovat. Délka periody hesla je proto maximálně 2^N, její konkrétní délka je určena hodnotou IV a může se pohybovat náhodně v rozmezí od 1 do 2^N.
- Struktura hesla je značně závislá na tom, zda zpětná vazba je plná nebo nikoli. Pro $b < N$ je střední hodnota délky periody pouze cca $2^{N/2}$, zatímco pro $b = N$ je to 2^{N-1}.

Operační mody blokových šifer (8)

Čítačový modus CTR (Counter Mode)

- Podobný OFB, převádí blokovou šifru na synchronní proudovou. Není problém s neznámou délkou periody hesla (je dána předem).
- IV se načte do vstupního registru (čítače) T. Po jeho zašifrování vzniká první blok hesla. Poté dojde k aktualizaci čítače T, nejčastěji přičtením jedničky a ke generování dalšího bloku hesla.
- Heslo se může využít v plné šíři bloku nebo jen jeho $b < N$ bitů. Způsob aktualizace čítače je definován volně, inkrementovat se může jen například dolních B bitů čítače T.
- V žádných zprávách šifrovaných tímtéž klíčem nesmí dojít k vygenerování stejného bloku hesla vícekrát $\Rightarrow$ obsah čítače nesmí být stejný.
- Jinak dvojí použití hesla $\rightarrow$ rozluštění OT.

Operační mody blokových šifer (9)

- Předpis pro šifrování v modu CTR je:

  $$CTR_i = |IV+i-1|_{2^B}, H_i = E_K(CTR_i), \check{S}T_i = OT_i \oplus H_i, \quad i = 1, 2, \ldots, n$$

  a pro dešifrování:

  $$CTR_i = |IV+i-1|_{2^B}, H_i = E_K(CTR_i), OT_i = \check{S}T_i \oplus H_i, \quad i = 1, 2, \ldots, n$$

- Výstupní blok lze použít celý nebo jen jeho část. Smyslem modu je zaručit maximální periodu hesla, což je zaručeno periodou čítače.
- Výhoda: heslo může být vypočítáno jen na základě pozice otevřeného textu a IV, nezávisle na ničem jiném.
- Tuto vlastnost má i modus OFB, ale k vypočítání hesla v tomto případě pro nějaký blok předchází výpočet předešlých hesel.
- Naproti tomu u modu CTR se vypočte hodnota čítače a provede se jen jedna transformace E_K: $E_K(counter)$.

Operační mody blokových šifer (9)

Metoda solení

- U operačních modů CBC, OFB, CFB i CTR je možné využívat metodu solení IV.
- Komunikujícímu protějšku se předává hodnota IV, ale k šifrování se použije jiná hodnota IV´ ("osolený IV").
- Tato hodnota se na obou stranách vypočítá z IV a klíče K nějakým definovaným způsobem. Např. to může být hašovací hodnota, vypočítaná ze zřetězení obou hodnot.
- Bezpečnostní výhodou je, že skutečně použitá inicializační hodnota IV´ se nikde neobjevuje na komunikačním kanálu.

Mod MAC (Message Authentication Code)

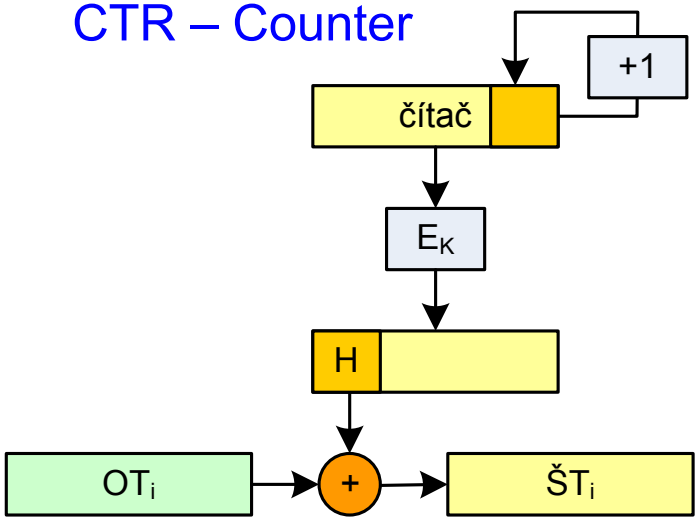
- Proudové a blokové šifry zajišťují důvěrnost, ne integritu zpráv.
- Mody CBC a CFB sice způsobí mírnou propagaci chyby (chyba v jednom bloku ŠT naruší 2 bloky OT), ale v systémech, kde není ve vlastním OT zajištěna nějaká redundance mohou být zpracována chybná data.

Operační mody blokových šifer (10)

- Autentizační kód zprávy (MAC) je dalším modem blokové šifry, který řeší právě zajištění neporušenosti dat. Tento zabezpečovací kód autentizuje původ zprávy a řeší obranu proti náhodným i úmyslným změnám nebo chybám na komunikačním kanálu.
- MAC je krátký kód, který vznikne zpracováním zprávy s tajným klíčem (K1). Klíč by se měl použít jiný, než k šifrování zprávy.
- Výpočet MAC probíhá tak, že se zpráva jakoby šifruje v modu CBC s nulovým IV, přičemž průběžný ŠT se nikam neodesílá.
- MAC je pak tvořen až posledním blokem $\check{S}T_n$, přičemž je možné ještě jedno přídavné šifrování navíc, tj. $MAC = EK_2(\check{S}T_n)$. Z výsledného bloku se obvykle bere jen určitá část (polovina bloku) o délce potřebné k vytvoření odolného zabezpečovacího kódu.
- MAC zajišťuje službu autentizace původu dat. Protože je to symetrická technika, nezaručuje nepopiratelnost.

Operační mody blokových šifer (11)

CTR – Counter



MAC – Message Authentication Code