Symetrická kryptografia

Stanislav Palúch

Fakulta riadenia a informatiky, Žilinská univerzita

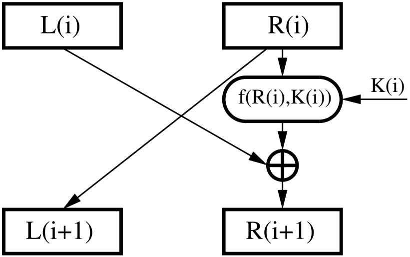
10. novembra 2010

Všeobecný princíp symetrickej kryptografie

1. A a B sa dohodnú na kryptosystéme
2. A a B sa dohodnú na kľúči
3. A (resp. B) šifruje priamy text x ako $y = E_K(x)$
4. B (resp. A) dešifruje zašifrovaný text y ako $x = D_K(y)$

Kryptosystémy Feistelovho typu

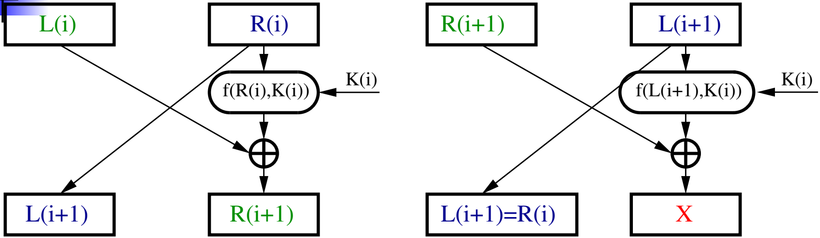
Sú to systémy s blokovou šifrou – šifrujú sa celé bloky priameho textu. Pre kryptosystémy Feistelovho typu musí mať blok párny počet bitov.
Blok sa rozdelí na dve rovnako dlhé časti – ľavú L_i a pravú R_i.



Šifrovanie prebieha po kolách
Jedno kolo urobí:

$$
\begin{aligned}
R_{i+1} &= L_i \oplus f(R_i, K_i) \\
L_{i+1} &= R_i
\end{aligned}
$$

Dešifrovanie

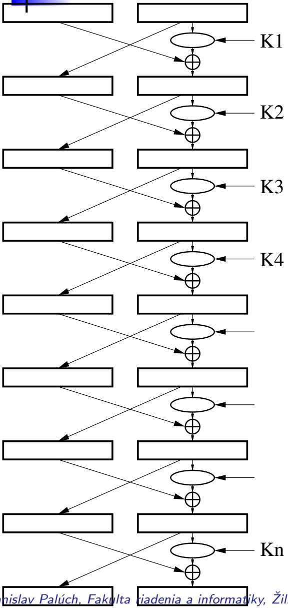


Počítajme X.

$$X = \underbrace{R_{i+1}}_{=L_i \oplus f(R_i,K_i)} \oplus f(\underbrace{L_{i+1}}_{=R_i}, K_i) = L_i \oplus \underbrace{f(R_i, K_i) \oplus f(R_i, K_i)}_{=0} = L_i$$

Dôsledok: Ak kolovému algoritmu vložíme kolový kľúč K_i, na miesto pravej časti L_{i+1} a na miesto ľavej časti R_{i+1}, dostaneme na jeho výstupe na pravej a ľavej časti porade pôvodné L_i a R_i. Ten istý kolový algoritmus (s prehodenou ľavou a pravou stranou a tým istým kolovým kľúčom) teda môžeme použiť ako inverznú funkciu.

Feistelova sieť



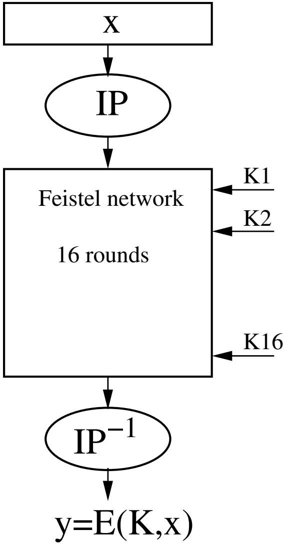
Feistelova sieť je iterované niekoľkonásobné opakovanie kolových algoritmov, každý s iným kolovým kľúčom.

Dešifrovanie sa urobí tou istou sieťou, ktorej sa na vstup vloží zašifrovaný text s poradím kolových kľúčov $K_n, K_{n-1,\dots,K_1}$ a so zameneným poradím pravej a ľavej časti.

Dôležité: Práve popísaný inverzný mechanizmus nezáleží na tvare funkcie $f(R_i, K_i)$.

Na funkcii $f(R_i, K_i)$ však podstatne závisia kryptografické vlastnosti Feistelovej siete.

DES – Data Encryption Standard



- Vyvinutý v IBM, publikovaný 1975
- Bloková šifra – 64-bitový blok
- 56-bitový kľúč
- Feistelova sieť so 16 kolami so vstupnou a výstupnou permutáciou
- IP – vstupná (inicializačná) permutácia
- IP^{-1} – výstupná permutácia

Vstupná a výstupná permutácia nemajú žiaden vplyv na bezpečnosť kryptosystému.

DES – Vstupná a výstupná permutácia

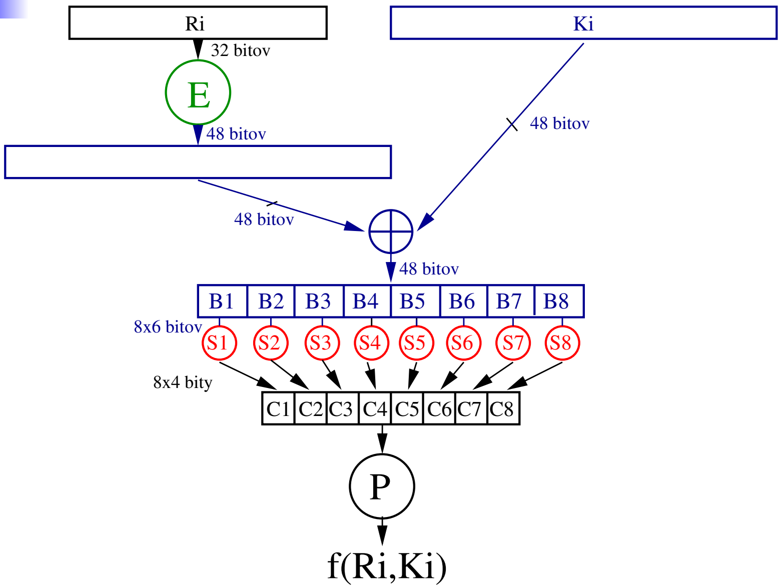
Table 12.1 Initial Permutation

58	50	42	34	26	18	10	2	60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6	64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1	59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5	63	55	47	39	31	23	15	7

Table 12.8 Final Permutation

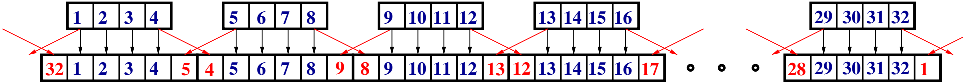
40	8	48	16	56	24	64	32	39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30	37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28	35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26	33	1	41	9	49	17	57	25

DES – Popis funkcie f v kryptosystéme DES

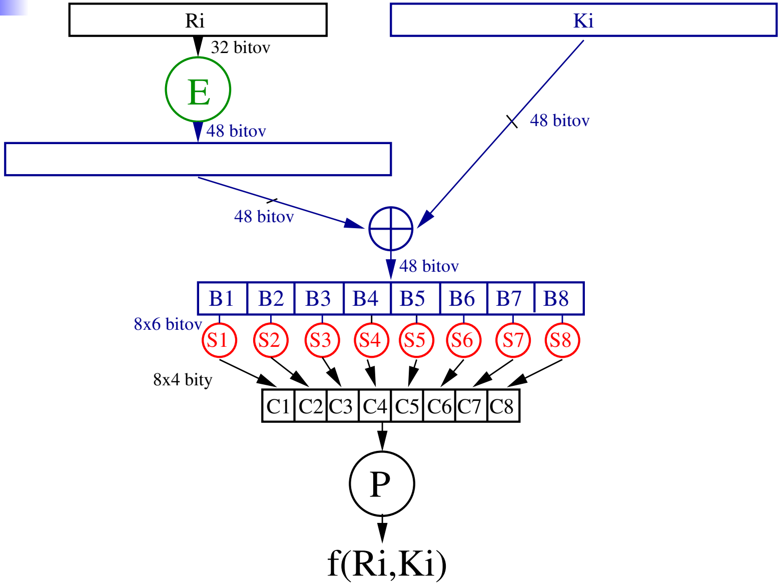


DES – Expanzná operácia

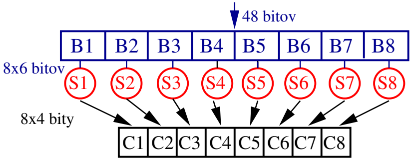
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1



Popis funkcie f v kryptosystéme DES - znovu



DES – Použitie S-boxov



- S-box je tabuľka so štyrmi riadkami a šestnástimi stĺpcami.
- Riadky sú číslované od 0 do 3, stĺpce sú číslované od 0 do 15.
- DES používa 8 S-boxov, bloku B_i je priradený S-box S_i.
- Každé B_i je 6-bitové číslo $b_1 b_2 b_3 b_4 b_5 b_6$ a predstavuje adresu príslušného štvorbitového čísla C_i v S-boxe S_i.

DES – Adresovanie v S-boxe

Adresa sa vypočíta takto:

Nech $B_1 = b_1 b_2 b_3 b_4 b_5 b_6$.

$b_1 b_6$ je číslo riadku, $b_2 b_3 b_4 b_5$ je číslo stĺpca v príslušnom S-boxe.
(Riadky i stĺpce sú číslované od 0 po 3 resp. od 0 po 15.)

S-box 1:

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Príklad:
$B_1 = 101011$. $b_1 b_6 = (11)_2 = 3$, $b_2 b_3 b_4 b_5 = (0101)_2 = 5$.
V S-boxe S_1 je v riadku 3 a stĺpci 5 číslo 9 (pozor, riadky a stĺpce sa číslujú od 0), ktorého binárny rozvoj je 1001. Je teda

$$S_1(B_1) = S_1(101011) = 1001 = C_1.$$

DES – S-boxy 2, 3, 4

S-box 2:

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S-box 3:

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S-box 4:

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

DES – S-boxy 5, 6, 7, 8

S-box 5:

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S-box 6:

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S-box 7:

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

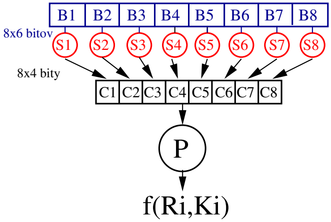
S-box 8:

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

DES – Záverečná permutácia kolovej funkcie

Table 12.7 P-Box Permutation

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	**3**	9
19	13	30	6
22	11	**4**	25

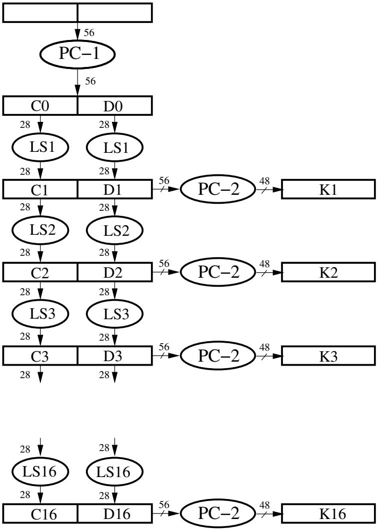


16	**7**	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	**8**	24	14
32	27	3	9
19	13	30	**6**
22	11	4	25

16	7	20	21
29	**12**	28	17
1	15	23	26
5	18	31	**10**
2	8	24	14
32	27	3	**9**
19	13	30	6
22	**11**	4	25

16	7	20	21
29	12	28	17
1	**15**	23	26
5	18	31	10
2	8	24	**14**
32	27	3	9
19	**13**	30	6
22	11	4	25

DES – Generovanie kolových kľúčov



Kľúč pre systém DES je 56-bitový, ale ukladá sa ako 64 bitov s tým, že v každom bajte je 7 bitov kľúča a jeden kontrolný bit doplňujúci bajt na nepárnu paritu. Po odstránení paritných bitov sa získa 56 bitov kľúča, ktorých poradie sa zmení podľa permutácie PC-1.

Potom sa 56 bitov kľúča rozdelí na dve 28-bitové časti C_0, D_0, na každú z nich sa postupne aplikuje ľavý rotačný posun $LS_1, LS_2 \dots, LS_{16}$. Pre $i = 1, 2, 9, 16$ je LS_i posun o jedno miesto, inak o 2 miesta.

Získa sa tak postupnosť $C_1D_1, C_2D_2, \dots, C_{16}D_{16}$ 56 bitových reťazcov, z ktorých operácia PC-2 výberom 48 bitov a ich permutáciou vytvorí postupne kľúče $K_1, K_2, \dots, K_{16}$.

DES – Permutácia PC-1 a zobrazenie PC-2

Permutácia PC-1

57	49	41	33	25	17	9	1	58	50	42	34	26	18
10	2	59	51	43	35	27	19	11	3	60	52	44	36
63	55	47	39	31	23	15	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

Zobrazenie PC-2

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

DES – Pravidlá tvorby S- boxov

Jediná nelinearita šifrovacieho algoritmu DES je v S-boxoch. Na nich závisí odolnosť DESu.

1. Každý riadok je permutáciou čísel 0 – 15.
2. Žiaden S-box nie je lineárnou alebo afinnou funkciou vstupov
3. Zmena jedného vstupného bitu S-boxu spôsobí zmenu aspoň dvoch bitov výstupu
4. Pre každý S-box a pre každé šesťbitové x
   $S(x)$ a $S(x \oplus 001100)$ sa líšia aspoň v dvoch bitoch
5. Pre každý S-box a pre každé šesťbitové x a pre ľubovoľné bity $r, s \in \{0, 1\}$ $S(x) \neq S(x \oplus 11rs00)$.
6. Ak fixujeme hodnotu jedného vstupného bitu, potom počet vstupných hodnôt, pre ktoré je ľubovoľný určený bit rovný 0 (alebo 1), je medzi 13 a 19.

Útoky proti DESu

Útok hrubou silou.
Počet kľúčov 2^{56} sa ukazuje v dnešnej dobe malý. Podarilo sa prelomiť DES distribuovaným výpočtom na Internete.

Diferenciálna kryptoanalýza.
Je to útok typu "chosen plaintext attack". Šifrovaciemu algoritmu s neznámym kľúčom sa dávajú šifrovať dvojice priamych textov P_1, P_2 s určitou diferenciou $P_1 \oplus P_2$ a na základe diferencie príslušných zašifrovaných textov sa usudzuju niektoré vlastnosti kľúča.

Lineárna kryptoanalýza.
Ak pre priamy text $x_1 x_2 \dots x_{64}$, kľúč $k_1 k_2 \dots k_{56}$ a pre príslušný zašifrovaný text $y_1 y_2 \dots y_{64}$ platí

$$\bigoplus_{i=1}^{64} a_i x_i \oplus \bigoplus_{i=1}^{64} b_i y_i = \bigoplus_{i=1}^{56} c_i k_i$$

s pravdepodobnosťou rôznou od $\frac{1}{2}$, dá sa to využiť pri kryptoanalýze.
Pre DES platí

$$x_{17} \oplus y_3 \oplus y_8 \oplus y_{14} \oplus y_{25} = K_{i,26}$$

s pravdepodobnosťou $\dfrac{1}{2} - \dfrac{5}{16} = \dfrac{3}{16}$.
Na základe tohoto faktu bol navrhnutý chosen plaintext attack analyzujúci priemerne 2^{43} známych priamych textov, ktorý odhalil kľúč za 50 dní práce 12 počítačov HP9735 (v roku 1994).

Pokusy o predĺženie kľúča

Namiesto jedného šifrovania kľúčom K_1 zašifrujeme dvakrát – najprv kľúčom K_1 a potom kľúčom K_2. Teda

$$\text{šifrujeme: } y = E_{K_2}\left[E_{K_1}(x)\right] \quad \text{dešifrujeme: } x = D_{K_1}\left[D_{K_2}(y)\right]$$

Ak by boli šifrovacie a dešifrovacie zobrazenia systému DES grupou, t.j. ak by pre K_1, K_2 existovalo K_3 také, že $E_{K_2}\left[E_{K_1}\right] = E_{K_3}$, dvojité šifrovanie by nemalo význam.

Príklady šifier, ktoré sú grupami:

- cézarovská šifra
- všeobecná monoalfabetická šifra
- permutačná šifra
- hillovská šifra

DES však nie je grupou.

Útok typu "Meet-in-the-middle"

Predpokladajme, že poznáme dvojicu x, y priameho textu a textu zašifrovaného dvojicou kľúčov K_1, K_2, t.j. $y = E_{K_2}[E_{K_1}(x)]$.

$D_{K_2}(y) = D_{K_2}\left\{E_{K_2}[E_{K_1}(x)]\right\} = E_{K_1}(x)$

Hľadáme takú dvojicu kľúčov K_1, K_2, pre ktoré je

$$D_{K_2}(y) = E_{K_1}(x).$$

Zostrojíme dve tabuľky – tabuľku 1. závislosti $E_{K_1}(x)$ na K_1 a tabuľku 2. závislosti $D_{K_2}(y)$ na K_2.

Ak nájdeme taký prvok v druhom stĺpci tabuľky 1., ktorý sa rovná niektorému prvku v druhom stĺpci tabuľky 2., našli sme v príslušných prvých stĺpcoch kandidátov na kľúče K_1, K_2.

K_1	$E_{K_1}(x)$
0	
1	
2	
L_1	z
$2^{56}-1$	

K_2	$D_{K_2}(y)$
0	
1	
2	
L_2	z
$2^{56}-1$	

Zložitosť útoku "meet-in-the-middle"

Postup možno zjednodušiť tak, že zostrojíme a zapamätáme si len tabuľku 1. a postupne generujeme $D_{K_2}(y)$ pre $K_2 = 0, 1, \dots$ a hľadáme jeho výskyt v druhom stĺpci tabuľky 1.

Pamäťové nároky: 2^n (2^{56}) riadkov tabuľky 1.

Výpočtové nároky:
2×2^n (2×2^{56}) sifrovaní
$2^n.\log_2 2^n = n.2^n$ (56.2^{56}) krokov na usporiadanie tabuľky 1
a najviac $2^n.\log_2 2^n = n.2^n$ (56.2^{56}) krokov na vyhľadávanie v tabuľke 1.
Spolu: $2.2^n + n.2^n + n.2^n = (2 + 2n)2^n = (1 + n).2^{n+1}$ (57.2^{57}).

Sú známe aj efektívnejšie útoky.

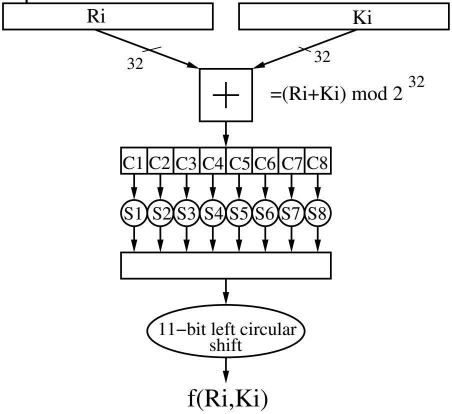
Útok hrubou silou na odhalenie kľúčov K_1, K_2 vyžaduje 2^{2n} (2^{112}) sifrovaní.

Dôsledok: Dvojité šifrovanie neprináša očakávané zosilnenie šifry.

3DES

šifrujeme: $y = E_{K_3}\left\{D_{K_2}\left[E_{K_1}(x)\right]\right\}$ dešifrujeme: $y = D_{K_1}\left\{E_{K_2}\left[D_{K_3}(x)\right]\right\}$

GOST



Sovietsky kryptovací systém používaný v časoch studenej vojny.

Bloková šifra.

64 bitový blok, 256 bitový kľúč.

Feistelova sieť s 32 kolami.

S-boxy sú jednoriadkové tabuľky obsahujúce permutácie čísel $0, 1, \ldots, 15$.

S-boxy kryptosystému GOST

S-box 1:

4	10	9	2	13	8	0	14	6	11	1	12	7	15	5	3

S-box 2:

14	11	4	12	6	13	15	10	2	3	8	1	0	7	5	9

S-box 3:

5	8	1	13	10	3	4	2	14	15	12	7	6	0	9	11

S-box 4:

7	13	10	1	0	8	9	15	14	4	6	12	11	2	5	3

S-box 5:

6	12	7	1	5	15	13	8	4	10	9	14	0	3	11	2

S-box 6:

4	11	10	0	7	2	1	13	3	6	8	4	9	12	15	14

S-boxy kryptosystému GOST

S-box 7:

13 11 4 1 3 15 5 9 0 10 14 7 6 8 2 12

S-box 8:

1 15 13 0 5 7 10 4 9 2 3 14 6 11 8 12

Generovanie kolových kľúčov

Kľúč je 256 bitový. Možno ho rozdeliť na osem 32-bitových kľúčov $K_1, K_2, \ldots, K_8$.

K1	K2	K3	K4	K5	K6	K7	K8

Tieto sa potom použijú v poradí:

$K_1, K_2, \ldots, K_8, K_1, K_2, \ldots, K_8, K_1, K_2, \ldots, K_8, K_8, K_7, \ldots K_1$

IDEA – Špecifikácia

IDEA – International Data Encryption Algorithm (Xueija Lai and James Massey) - 1992.
Je patentovaný, US patent vyprší 7.1.2012.

Bloková šifra – blok 64 bitov
Kľúč 128 bitov.

64-bitový blok sa rozdelí na 4 16-bitové časti x_1, x_2, x_3, x_4, s ktorými sa urobí 8 kôl algoritmu plus záverečné "polovičné kolo."

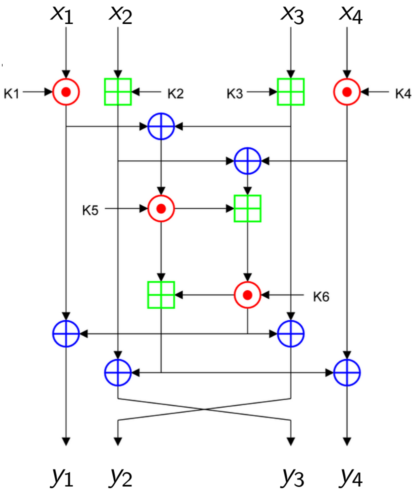
V kolách sa používajú tieto operácie:

$\oplus$ – XOR po bitoch

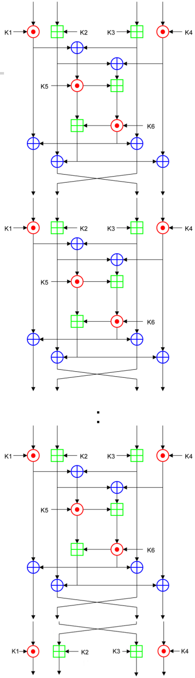
$\boxplus$ – sčítanie mod 2^{16}

$\odot$ – násobenie mod $(2^{16}+1)$ pričom sa 16-bitové slovo pozostávajúce zo samých 0 považuje za reprezentáciu čísla 2^{16}.

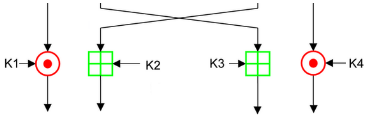
Jedno kolo algoritmu IDEA



IDEA – Generovanie kolových kľúčov



Záverečné polovičné kolo



Generovanie kolových kľúčov

Každé kolo potrebuje 6 kľúčov a záverečné polovičné kolo 4 kľúče, t.j spolu $6 * 8 + 4 = 52$ 16-bitových kľúčov.
Najprv sa 128 bitový kľúč rozdelí na 8 16-bitových kľúčov.
Potom sa na kľúč aplikuje ľavý rotačný posun o 25 bitov a získa sa ďalších 8 kľúčov.
Kľúč sa znovu rotuje o 25 bitov a získa sa ďalších 8 kľúčov. Atď.

Dešifrovanie

Ten istý algoritmus sa použije aj na dešifrovanie s tým, že ako kľúče sa použijú opačné resp. inverzné hodnoty kľúčov zo šifrovania vo vhodnom poradí.

.

Operačné módy blokových šifier

Majme blokovú šifru so šifrovacím zobrazením $y = E_K(x)$
a dešifrovacím zobrazením $x = D_K(y)$.
Máme priamy text vyjadrený ako postupnosť blokov

$$x_1, x_2, \dots, x_n$$

Je niekoľko spôsobov, ako vytvoriť zodpovedajúcu postupnosť blokov zašifrovaného textu

$$y_1, y_2, \dots, y_n$$

s použitím zobrazenia $E_K(x)$ tak, aby sa pomocou dešifrovacieho zobrazenia dala zrekonštruovať pôvodná postupnosť

$$x_1, x_2, \dots, x_n$$

Tieto spôsoby sa nazývajú operačné módy blokových šifier.

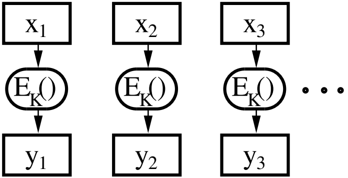
ECB – Electronic Code Book

Najjednoduchším módom je ECB mód, kedy sa šifruje priamy text blok po bloku predpisom

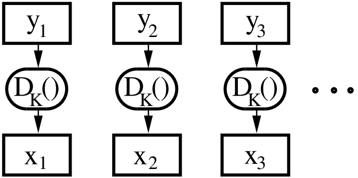
$$y_i = E_K(x_i)$$

a dešifruje predpisom

$$x_i = D_K(y_i)$$



Šifrovanie v ECB móde



Dešifrovanie v ECB móde

Nevýhoda: Rovnaký blok x_i sa zakaždým zašifruje na rovnaký blok y_i, čo môže uľahčiť niektoré útoky.

OFB – Output Feedback Mode

Pri tomto móde sa najprv zvolí náhodný inicializačný blok IV zvaný tiež inicializačný vektor a položí sa $y_0 = IV$. Postupne sa vypočítajú $z_1 = E_K(y_0)$, a rekurentne $z_{i+1} = E_K(z_i)$.

Šifrujeme predpisom

$$y_i = z_i \oplus x_i$$

Zašifrovaná správa je postupnosť $\color{red}{y_0}, y_1, y_2, \ldots, y_n$ (má o jeden blok viacej) a dešifrujeme predpisom

$$x_i = z_i \oplus y_i.$$

Tento mód pripomína prúdovú šifru s prúdom kľúčov $z_1, z_2, \ldots, z_n$, preto je nutné pre každú správu používať iný inicializačný vektor.

Cipher Block Chaining Mode

Šifrujeme predpisom

$$y_i = E_K(x_i \oplus y_{i-1})$$

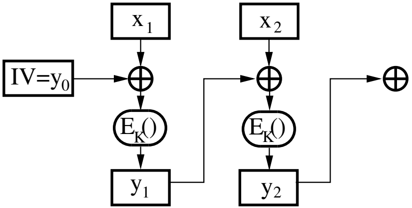
Zašifrovaná správa je postupnosť

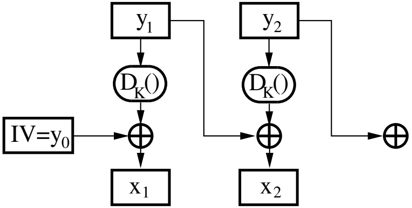
$$\color{red}{y_0}, y_1, y_2, \ldots, y_n$$

(má o jeden blok viacej).

Dešifrujeme predpisom

$$x_i = y_{i-1} \oplus D_K(y_i).$$

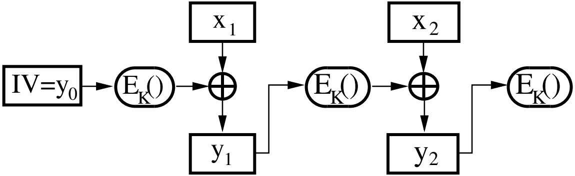




Cipher Feedback Mode

Šifrujeme predpisom

$$y_i = E_K(y_{i-1}) \oplus x_i$$



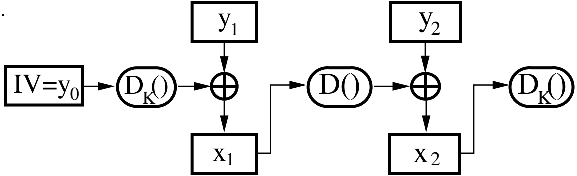
Zašifrovaná správa je postupnosť

$$\color{red}{y_0}\color{black}, y_1, y_2, \ldots, y_n$$

(má o jeden blok viacej).

Dešifrujeme predpisom

$$x_i = y_i \oplus a_K(y_{i-1}).$$



Galoisove pole $GF(2^8)$

Pole $GF(2^8)$ Prvky: polynómy typu

$$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x^1 + b_0$$

s koeficientami v $\mathbb{Z}_2$.
Takýto polynóm modeluje bajt $b_7b_6b_5b_4b_3b_2b_1b_0$. Tak napríklad $\{0\ 1\ 0\ 1\ 0\ 1\ 1\ 1\}$ zodpovedá polynómu $x^6 + x^4 + x^2 + x + 1$.

Sčítanie v $GF(2^8)$ je sčítanie polynómov nad $\mathbb{Z}_2$.

$(x^6 + x^4 + x^2 + x + 1) + (x^7 + x^6 + x^4 + x^2) = (x^7 + x + 1)$
$\{0\ 1\ 0\ 1\ 0\ 1\ 1\ 1\} \oplus \{1\ 1\ 0\ 1\ 0\ 1\ 0\ 0\}$= $\{1\ 0\ 0\ 0\ 0\ 1\ 1\}$
V hexadecimálnom zápise $(57)_H \oplus (D4)_H = (83)_H$.

Sčítaniu bajtov $\oplus$ zodpovedá počítačová bajtová operácia XOR po bitoch.

Násobenie v $GF(2^8)$ sa definuje ako

$$p(x) \otimes q(x) = p(x).q(x) \quad \text{mod } m(x),$$

kde $m(x)$ je ireducibilný polynóm stupňa 8 nad $GF(2^8)$.

AES používa tento ireducibliný polynóm $m(x) = x^8 + x^4 + x^3 + x + 1$.

$$\Big(\underbrace{(x^6 + x^4 + x^2 + x + 1)}_{57_H = \{01010111\}} . \underbrace{(x^7 + x + 1)}_{83_H = \{10000011\}}\Big) \;\text{ mod }\; \underbrace{(x^8 + x^4 + x^3 + x + 1)}_{=m(x)} =$$

$$(x^{13} + x^{11} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + 1) \;\text{ mod } m(x) =$$
$$= \underbrace{(x^7 + x^6 + 1)}_{C1_H = \{11000001\}}$$

V $GF(2^8)$ teda máme $\{01010111\} \otimes \{10000011\} = \{11000001\}$

AES – Násobenie číslom $2 \equiv \{00000010\} \equiv x$

Polynóm x zodpovedá bajtu $\{00000010\}$, t.j číslu $2 = (02)_H$.
Skúmajme, čomu sa rovná $\{00000010\} \otimes b$.

Nech
$b(x) = b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x^1 + b_0$.
Potom
$x.b(x) = b_7x^8 + b_6x^7 + b_5x^6 + b_4x^5 + b_3x^4 + b_2x^3 + b_1x^2 + b_0x$

Ak $b_7 = 0$, $x.b(x) \mod m(x) = x.b(x)$,
kde $m(x) = x^8 + x^4 + x^3 + x + 1$.
Túto operáciu vykoná ľavý posun bajtu b o 1 bit.

Ak $b_7 = 1$,
potom

$$x.b(x) \mod m(x) = x.b(x) \ominus m(x) = x.b(x) \oplus m(x).$$

Túto operáciu vykoná ľavý posun bajtu b a následne bitový XOR s bajtom $\{00011011\}$ (hexadecimálne $(1B)_H$).

AES – Násobenie $\mathbf{a} \otimes \mathbf{b}$

Môžeme teda definovať funkciou

```
xtime(b)
1.  if (b[7] == 1) t=00011011 else t=00000000;
2.  for(i=7 to 1) b[i]=b[i-1];
3.  b = b ⊕ t;
4.  return b;
```

Potom násobenie $\mathbf{a} \otimes \mathbf{b} = \mathbf{c}$ realizujeme nasledovne:

```
1.  c=00000000;
    p = a;
2.  for(i=0 to 7);
      if(b[i] == 1) c = c ⊕ p;
      p=xtime(p);
3.  return c;
```

AES – Výpočet inverzného prvku b^{-1}

$GF(2^8)$ s operáciami $\oplus$, $\otimes$ tvorí pole, v ktorom

- nulový prvok je polynóm 0 – 00000000
- jednotkový prvok je prvok 1 – 00000001 $\equiv 0x^7 + 0x^6 + \cdots + 0x + 1$
- ku každému prvku b existuje opačný prvok – je to samotné b,
- ku každému prvku $b \neq 0$ existuje inverzný prvok b^{-1}.

Inverzný prvok možno vypočítať rozšíreným Euklidovým algoritmom. Pre účely AES však stačí vypočítať tabuľku binárnej operácie $\otimes$ (má rozmer 256×256) a pre každé $b = 1, 2, \ldots, 255$ nájsť to c, pre ktoré je $b \otimes c = 1$, a položiť $b^{-1} = c$.

Ak vytvoríme tabuľku s 256 položkami typu

0	1	2^{-1}	3^{-1}	...	...	255^{-1}

inverzný prvok b^{-1} k prvku b získame ako položku tejto tabuľky na mieste (adrese) b.

AES – Advanced Encryption Standard – História

- 1997 – inicializácia procesu výberu vhodného symetrického kryptografického algoritmu – NIST (National Institute of Standards and Technology - USA)
- Súťaže sa zúčastnilo 15 algoritmov
- 1998 publikovali Vincent Rijmen (1970) a Joan Daemen (1965) (Belgicko) algoritmus Rijndael
- Od roku 2002 bol Rijndael uznaný autoritami NIST, FIPS[1], NSA[2] za nový kryptografický štandard označovaný ako AES
- AES je jediný verejne dostupný šifrovací algoritmus schválený NSA pre najtajnejšie (top secret) informácie

Výhody:
– Výkonnosť v hardvérovej i softvérovej implementácii
– Nízke pamäťové nároky
– Možnosť ochrany pred útokmi parazitnými kanálmi

[1]FIPS – Federal Information Processing Standard)
[2]NSA – National Security Agency

AES - Advanced Encryption Standard – Špecifikácia

- Bloková šifra
- Dĺžka bloku: 128 bitov
- Dĺžka kľúča: voliteľne 128, 192 alebo 256 bitov

128-bitový blok priameho textu berieme ako postupnosť 16 bajtov:

$$a_{00}a_{10}a_{20}a_{30}a_{01}a_{11}a_{21}a_{31}a_{02}a_{12}a_{22}a_{32}a_{03}a_{13}a_{23}a_{33}$$

Tieto sa usporiadajú do tabuľky,
ktora sa vola Stav

a_{00}	a_{01}	a_{02}	a_{03}
a_{10}	a_{11}	a_{12}	a_{13}
a_{20}	a_{21}	a_{22}	a_{23}
a_{30}	a_{31}	a_{32}	a_{33}

Stav

k_{00}	k_{01}	k_{02}	k_{03}
k_{10}	k_{11}	k_{12}	k_{13}
k_{20}	k_{21}	k_{22}	k_{23}
k_{30}	k_{31}	k_{32}	k_{33}

Kolový kľúč

S týmto stavom sa iteračne vykonáva niekoľko kôl operácií, niektoré z nich závisia na kolovom kľúči, reprezentovanom ako matica bajtov

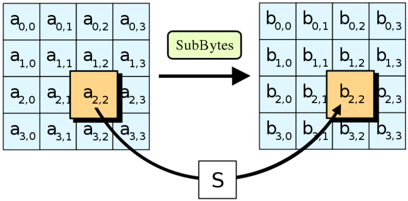
AES - Operácia SubBytes

S každým bajtom a tabuľky Stav sa vykonajú dve operácie:

1. Najprv sa k hodnote a najde v poli $GF(2^8)$ inverzný prvok $x = a^{-1}$, ak $a \neq 0$.

   Ak $a = 0$, položíme $x = 0$.

2. Potom sa vypočíta byte $b = b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7$



$$
\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix}
=
\begin{bmatrix}
1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}
\cdot
\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}
+
\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}
$$

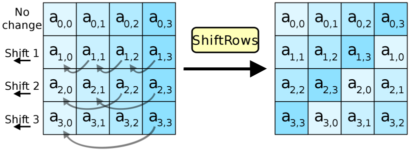
AES – Tabuľka funkcie SubBytes

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

AES - Operácia ShiftRows

Na riadky tabuľky Stav sa aplikujú nasledujúce ľavé rotačné posuny

1. 1. riadok ostáva bez zmeny
2. 2. riadok - posun o 1 bajt - t.j 8 bitov
3. 3. riadok - posun o 2 bajty - t.j 16 bitov
4. 4. riadok - posun o 3 bajty - t.j 24 bitov

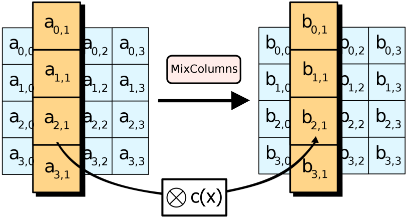


AES - Operácia MixColumns

Pri tejto operácii považujeme maticu Stav za maticu prvkov poľa $GF(2^8)$. S každým jej stĺpcom $\mathbf{a}_i = \begin{bmatrix} a_{0i} & a_{1i} & a_{2i} & a_{3i} \end{bmatrix}^T$ vykonáme

$$\underbrace{\begin{bmatrix} b_{0i} \\ b_{1i} \\ b_{2i} \\ b_{3i} \end{bmatrix}}_{\mathbf{b}_i} = \underbrace{\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}}_{\mathbf{M}} \underset{GF(2^8)}{\bigotimes} \underbrace{\begin{bmatrix} a_{0i} \\ a_{1i} \\ a_{2i} \\ a_{3i} \end{bmatrix}}_{\mathbf{a}_i} \qquad \text{t. j.} \quad \mathbf{b}_i = \mathbf{M} \otimes \mathbf{a}_i$$

V maticovom tvare: $\mathbf{B} = \mathbf{M}.\mathbf{A}$



$$\mathbf{M}^{-1} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix}$$

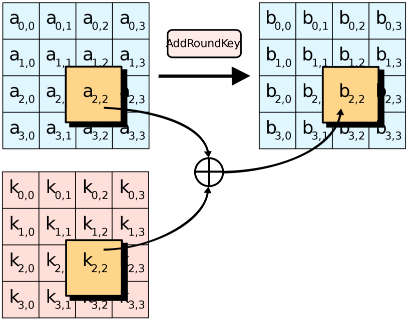
AES – Funkcia AddRoundKey

V tomto kole sa pre každý prvok a_{ij} Stavu vykoná

$$b_{ij} = a_{ij} \oplus k_{ij},$$

kde k_{ij} je prvok matice príslušného kolového kľúča.
V maticovom tvare:

$$\mathbf{B} = \mathbf{A} \oplus \mathbf{K}.$$



AES – šifrovací algoritmus

1. Inicializačné kolo
   - *1.1* AddRoundKey
2. for *Round* $= 1$ to $N_r - 1$
   - *2.1* SubBytes
   - *2.2* ShiftRows
   - *2.3* MixColumns
   - *2.4* AddRoundKey
3. Záverečné kolo (bez MixColumns)
   - *3.1* SubBytes
   - *3.2* ShiftRows
   - *3.3* AddRoundKey

Dĺžka kľúča	128	192	256
Počet kôl N_r	10	12	14

AES – Dešifrovanie

Malo by byť:

1 Inicializačné kolo
   1.1 AddRoundKey
   1.2 InvShiftRows
   1.3 InvSubBytes
2 for $Round = 1$ to $N_r - 1$
   2.1 AddRoundKey
   2.2 InvMixColumns
   2.3 InvShiftRows
   2.4 InvSubBytes
3 Záverečné kolo
   3.3 AddRoundKey

Je:

1 Inicializačné kolo
   1.1 AddRoundKey
2 for $Round = 1$ to $N_r - 1$
   2.1 InvSubBytes
   2.2 InvShiftRows
   2.3 InvMixColumns
   2.4 AddRoundKey
3 Záverečné kolo
   3.1 InvSubBytes
   3.2 InvShiftRows
   3.3 AddRoundKey

Poradie operácií InvShiftRows a InvSubBytes je zameniteľné.

$$AddRoundKey(InvMixcolumns(\mathbf{B})) = \mathbf{K} \oplus \mathbf{M}^{-1}.\mathbf{B}.$$
$$InvMixcolumns(AddRoundKey(\mathbf{B})) = \mathbf{M}^{-1}.(\mathbf{K} \oplus \mathbf{B}) = \mathbf{M}^{-1}\mathbf{K} \oplus \mathbf{M}^{-1}\mathbf{B}.$$

Príklad pre 128 bitový kľúč

$\mathbf{W}_0$	$\mathbf{W}_1$	$\mathbf{W}_2$	$\mathbf{W}_3$	$\mathbf{W}_4$	$\mathbf{W}_5$	$\mathbf{W}_6$	$\mathbf{W}_7$	$\mathbf{W}_8$	$\mathbf{W}_9$	$\mathbf{W}_{10}$	$\mathbf{W}_{11}$
k_{00}	k_{01}	k_{02}	k_{03}								
k_{10}	k_{11}	k_{12}	k_{13}								
k_{20}	k_{21}	k_{22}	k_{23}								
k_{30}	k_{31}	k_{32}	k_{33}								
1. kolový kľúč				2. kolový kľúč				3. kolový kľúč			

$$\mathbf{W}_i = \begin{cases} \mathbf{W}_{i-4} \oplus W_{i-1} & \text{ak } i \text{ nie je delitené 4} \\ \mathbf{W}_{i-4} \oplus SubByte(RotByte(\mathbf{W}_{i-1})) \oplus Rcon(i/4) & \text{ak } i \text{ je delitené 4} \end{cases}$$

$$\begin{aligned} Rcon(i) &= [\{x^{i-1}\}\{00\}\{00\}\{00\}] \\ RotByte[w_1, w_2, w_3, w_4] &= [w_2, w_3, w_4, w_1] \end{aligned}$$

AES – Expanzia kolových kľúčov

```
KeyExpansion(byte key[4*Nk], word w[Nb*(Nr+1)], Nk)
begin
  word temp
  i = 0
  while (i < Nk)
    w[i] = word(key[4*i], key[4*i+1], key[4*i+2], key[4*i+3])
    i = i+1
  end while
  i = Nk
  while (i < Nb * (Nr+1)]
    temp = w[i-1]
    if (i mod Nk = 0)
        temp = SubWord(RotWord(temp)) xor Rcon[i/Nk]
    else if (Nk > 6 and i mod Nk = 4)
        temp = SubWord(temp)
    end if
    w[i] = w[i-Nk] xor temp
    i = i + 1
  end while
end
```

Nb – $=$ 4 – počet stĺpcov matice Stav
Nk – $=$ 4, 6 resp. 8 pre 128-, 192- resp. 256-bitový kľúč
(počet 32-bitových slov kľúča)

Nr – $=$ 10, 12, resp. 16 pre 128-, 192- resp. 256-bitový kľúč – počet kôl