

HILLOVSKÁ ŠIFRA

$x_1 \dots x_n$ — jeden blok
 \parallel
 X_1

$$K = n \times n$$

$$K \cdot X = Y$$

$$K = X^{-1} \cdot Y$$

$$K = \begin{pmatrix} k_{11} & \dots & k_{1n} \\ \vdots & & \vdots \\ k_{n1} & \dots & k_{nn} \end{pmatrix}$$

$$E_K(X) = Y$$

$$D_K(Y) = K^{-1} \cdot Y = X$$

$$K^{-1} = K \cdot E = (K | E) \dots = (E | K^{-1})$$

- slabiny: ak správa obsahuje blavičky, dátum, meno ... dobažeme odhadnúť priamy a šifrovaný text \Rightarrow dobažeme spočítať kľúč. (mail)

VIGENEROVSKÁ ŠIFRA

TEXT — NA — ZASIFROVANIE
 HESLO HESLO HESLO HESLO

$$J_i = H + K \pmod{27}$$

Pomocou IK zistíme, či ide o monoalfabetickú alebo polyalfabetickú šifru.

- ak: $IK = 0,0602\%$ \rightarrow Monoalfabetická šifra

$IK = 0,0384\%$ \rightarrow Polyalfabetická šifra

$$IK = \sum_{i=1}^q p^2(a_i) = \sum_{i=1}^q \frac{n_i(n_i-1)}{n(n-1)}$$

\nearrow z krátkého textu \nearrow z dlhého

q - počet znakov abecedy

n_i - počet výskytu i-tého znaku

$$n = \sum n_i$$

- ak sa jedná o monoalfabetickú šifru, IK je rovnaký, len sa posunujú početnosti.
 pre SR je $IK = 0,0602\%$

- nedá sa použiť frekvencná analýza, pretože jedno písmeno sa môže zmeniť na rôzne.

HESLO HESLO
 A A
 \downarrow \downarrow
 H E

KASISKIHO

- ~~KASISKIHO~~ METÓDA - hľadajú sa rovnaké trojice - sú kódované rovnakou časťou hesla

... RCH ... RCH ...
 \leftarrow k.T \rightarrow

Treba nájsť všetky dĺžky k.T a potom z nich najväčší spoločný deliteľ \Rightarrow pravdepodobná dĺžka hesla

OTP šifra

- spočíva v posune každého znaku správy o náhodne zvolený počet miest v abecede.
- pre binárny tvar : $y_i = x_i \oplus k_i$

vlastnosti na dosiahnutie čo najlepšej šifry:

- dĺžka kľúča = dĺžke správy
- kľúč je dokonale náhodný
- kľúč sa nesmie použiť opakovane

- dešifrovanie : $x_i = y_i \oplus k_i$

Vhodnosť vygenerovaných bitov je možné zistiť pomocou testov: (frekvencný, Two bit, Poker, Runs, Autokorelačný, FIPS 140-1.)

Generátory


1) Lineárny kongruenčný - vyrába hodnoty x_1, x_2, \dots, x_n a to tak, že:

$$x_n = (a \cdot x_{n-1} + b) \bmod c \quad a, b, c - \text{mnohokrát zvolené konštanty tak, aby } x_n \in \langle 0, c-1 \rangle$$

- mal by mať čo najdlhšiu periódu,
- mali by sa využiť všetky n-tice z intervalu.
- dobrý pre simuláciu, nie pre kryptografiu.

2) Kvadratický kongruenčný - $x_n = (a x_{n-1}^2 + b x_{n-1} + c) \bmod d$

- tieto generátory prejdú testami, inak nič moc.

3) RC4 - s_0, s_1, \dots, s_{255} $0 \leq s_i \leq 255$

↓
32-bit-y, permutácia čísel od 0-255

```
rand()
i = (i + 1) mod 256
j = j + s[i] mod 256
swap(s[i], s[j])
t = s[i] + s[j] mod 256
k = s[t]
return k
```

kľúč - zoberieme pole $k[0]$ - zaplníme ho bitmi kľúča. Treba si nastaviť začiatočnú permutáciu. $s[i] = i$ pre $i = 0 \dots 255$

- odporúča sa prvých 1000 b kľúča vypustiť a použiť až ďalšie.
- nemá krátke cykly
- neprejde všetkými permutáciami

RSA

- každý účastník má 1 tajný a 1 veřejný klíč.
- odesílatel šifruje správu veřejným klíčem $y = E_{kv}(x)$
- příjematel dešifruje správu svým privátním (tajným) klíčem $x = D_{kt}(y)$
- algoritmus:

- 1.) Zvolíme 2 tajné, velké prvočísla p a q
 - 2.) $n = p \cdot q$ n - část klíče
 - 3.) Eulerova funkce $\varphi(n) = (p-1) \cdot (q-1)$
 - 4.) najdeme 2 velké čísla $1 < e, d < n$, aby $e \cdot d \equiv 1 \pmod{\varphi(n)}$
kongruence
 - 5.) šifrujeme: $y = E_{kv}(x) = x^e \pmod{n}$
 $x = D_{kt}(y) = y^d \pmod{n}$
- Tajný klíč: $k_t = d, n$
Veřejný klíč: $k_v = e, n$

Bezpečnost algoritmu:

- at znám p a q , ~~řip~~ \rightarrow nabírám celý algoritmus.
 - n nesmí být lidskými ~~silami~~ silami rozložitelné na prvočísla, p a q musí být dostatečně velké.
 - odporůča sa, aby n bolo v dnešnej dobe aspoň 2048 b. dlhé.
 - $\varphi(n)$ musí sa utajiť, ak za $q = \frac{n}{p}$, ostane z Euklida jedna neznáma p .
- Hľadanie prvočísla (najväčšou)
- náhodne vygenerovať ~~p~~ číslo a zistiť, či je prvočíslo - skutočnosť faktorizácia - nie
 - malá Fermatova veta - berieme číslo, kt. sú s veľkou pätou prvočíslo.
ak p je prvočíslo: $a^{p-1} \equiv 1 \pmod{p}$
 - prvočíslo je dost (nekonečne veľa)
 - $\varphi(n)$ - je Eulerova funkcia = počet čísel $\leq n$ nesdeliteľných s n
 - je založený (RSA) na ťažkosti faktorizácie čísel.

TESTY

1) Frekvencný test -

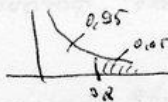
n - počet bitov kľúča

n_1 - počet "1"

n_0 - počet "0"

- či $\text{prst } "1" = \text{prst } "0"$

$$X_1 = \frac{(n_0 - n_1)^2}{n} \approx \chi^2(1)$$



ke $E = 0.95 \Rightarrow \text{prst } "1" = \text{prst } "0"$

2) Two bit test -

n - počet bitov

n_{00} - "00"

n_{01} - "01"

n_{10} - "10"

n_{11} - "11"

- či sú jednotlivé dvojice rovnomerne rozdelené.

$$X_2 = \frac{4}{n-1} (n_{00}^2 + n_{01}^2 + n_{10}^2 + n_{11}^2) - \frac{2}{n} (n_{00}^2 + n_{11}^2) + 1 \approx \chi^2(2)$$

3) Poker test -

$\overbrace{m \ m \ m \ m \ m}^m$ - k - k

$$X_3 = \frac{k \cdot m}{k} \cdot \left(\sum_{i=1}^m n_i^2 \right) - k \approx \chi^2(2^m - 1)$$

$k \cdot m = n$

2^m - počet hodnôt, akým môžeme vyjadriť m -bitové číslo

- či sú rovnomerne rozdelené m -tice v prípade bitov (prst m -tic rovnaká)

4) Runs test -

Block dišly $n \dots 0 \underbrace{111 \dots 110}_n \dots n$ jednotiek - max. postupnosť poradia idielich jednotiek

GAP dišly $n \dots 1 \underbrace{000 \dots 001}_n \dots n$ nul - "1" nul

- či postupnosť "0" a "1" sú v súlade s rovnakým výskytom.

$$X_4 = \sum_{i=1}^k \frac{(B_i - e_i)^2}{e_i} + \sum_{i=1}^k \frac{(G_i + e_i)^2}{e_i} \approx \chi^2(2k - 2)$$

5) FIPS 140-1 test - pre postupnosť 10000 bitov.

a) frekvencný test - pre $9654 < n_1 < 10346$ - počet jednotiek by mal byť v tomto rozmedzí

b) poker - pre $n=4$: $1.03 < X_3 < 57.4$

c) runs test B_i, G_i - $i=1,2,\dots,6$ G_6, B_6 - počet GAPs a blokov dišly ako B .

d) long run test - nesmie existovať blok alebo GAP dišly ≥ 34 .

6) Autokorelačný test -

Hashovacie funkcie

- predpis pre výpočet kontrolného súčtu správy (väčšie mn. dát).
- môže slúžiť ku kontrole integrity dát, porovnaniu dvojice správ, indexovaniu, vyhľadávaniu...
- je dôležitou súčasťou kryptografických systémov pre digitálny podpis.
- formálne je to funkcia h , ktorá prevádza vstupnú postupnosť bitov na postupnosť pevnej dĺžky n -bitov.

$$h: \mathcal{D} \rightarrow \mathcal{R}$$

- môže nastať kolízia, ak $h(x) = h(y)$, čo je nežiadúce.
- Vyhnuť sa im nedať, ale dá sa znížiť pravdepodobnosť vzniku kolízií.

Vlastnosti:

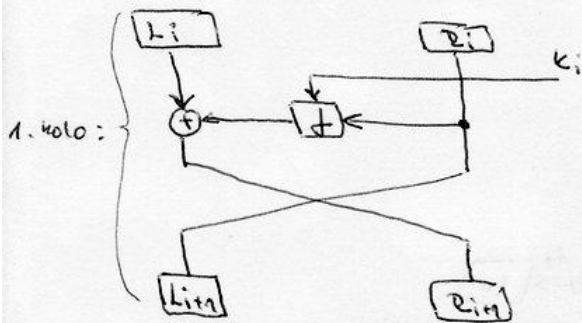
- jednocestnosť - znalosť výstupu nijak nevedie k znalosti vstupného textu
 - pre dané x ľahko spočítať $H(x)$, pre dané $H(y)$ ťažko spočítať y .
- silná bezkolíznosť - ak nie je možné v rozumnom čase nájsť akýkoľvek pár vstupov, aby platilo $H(x_1) = H(x_2)$
- slabá bezkolíznosť - ak nie je možné v rozumnom čase k danému vstupu iný dát, aby platilo $H(x_1) = H(x_2)$ $x_1 \neq x_2$.
- lavínovosť - aj malá zmena vstupu rapídne ovplyvní výstup.
- Rozloženie výstupov - funkcia distribuuje výstupy rovnomerne vo veľkom obore hodnôt, produkuje málo kolízií.

Základné hashovacie funkcie

- MD5 (message-digest) - jeho hlavná vlastnosť je lavínovosť. od 2004 zlomenej
- SHA (Secure Hash Algorithm) - či : SHA1:
od 2005 zlomenej
- SHA2 - rodina 4-hashovacích fcií (SHA-256, 384, 512, 224) - sú súčasťou štandardu FIPS 180-2 - milióni nariadení.

šifry feistelovho typu

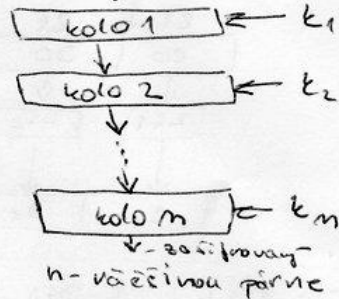
- celý text sa rozdelí na bloky párnej dĺžky (64b, 128b...).
- Potom sa blok po bloku šifruje.
- Jeden blok rozdelíme na 2 bloky (ľavý a pravý, napr. blok=128b \rightarrow 64b + 64b).



$$L_{i+1} = R_i$$

$$R_{i+1} = L_i \oplus f(R_i, k_i)$$

Príamý text



- ten istý mechanizmus používame na šifrovanie aj dešifrovanie.

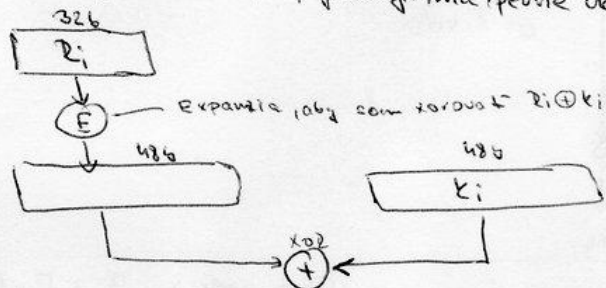
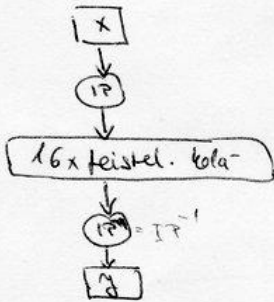
- funkcia f musí byť volená tak, aby systém mal dobré kryptografické vlastnosti.

- existencia inverzie nezávislá od f !

- pri dešifrovaní otočíme poradie kľúčov pretej istej schémy

DES - data encryption standard

- používa 56b. kľúč = málo, 64b. blok, fcn f má pevne danú.



E:

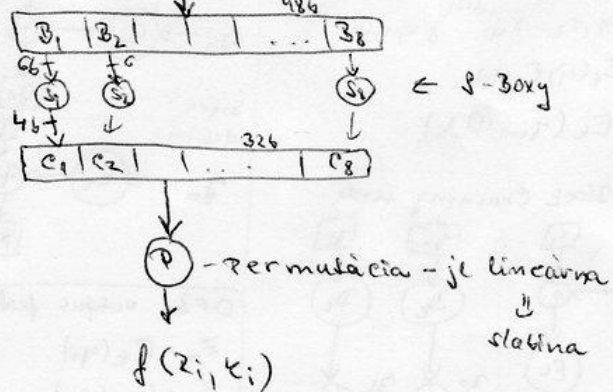
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

S-boxy: 4-nadky, 16 stĺpcov

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

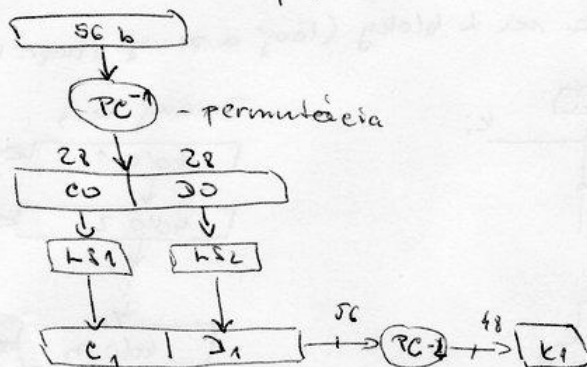
$b_1 b_6$ - bin. číslo viaceru

b_2, b_5 - 16 stĺpcov



klúč pri DES:

- každý posledný bit je kontrolný (dopočítaný na nepárnu paritu)
- 64 b - (56 b. kľúča + 8 kontrolný)
- generovanie:

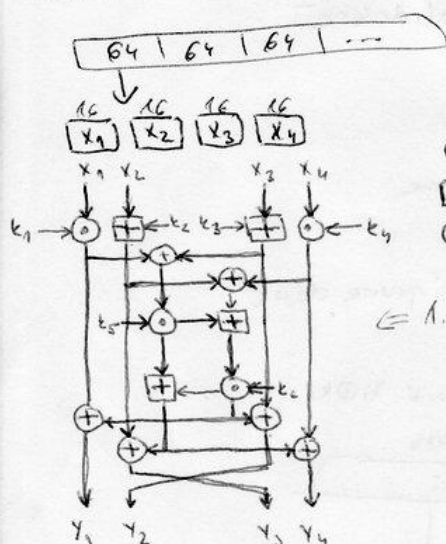


Triple DES: - kľúč 168 b. 3x DES.

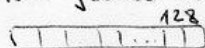
- pomalý

GOST: - kľúč 256 b., blok 64 b., 32 feistel. kol.

IDEA: - kľúč 128 b., 64 b. blok, 8 kol a pd



Na jedno kolo potrebujem 6 kľúčov (6x8+4 = 52 kľúčov)



- ⊙ násobenie modulo $2^{16} + 1$
- ⊕ sčítanie modulo 2^{16}
- ⊗ x02

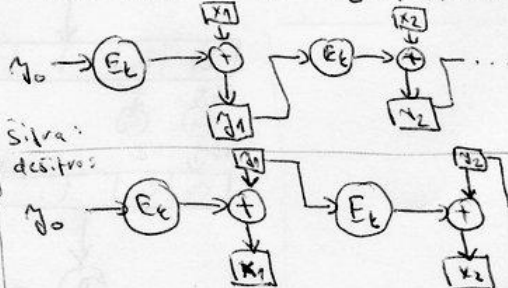
25x rotačný ľavý posun (ten sa robí dokiaľ nemám toľko kľúčov, koľko potrebujem).

≡ 1. kolo

ECB - electronic code Book

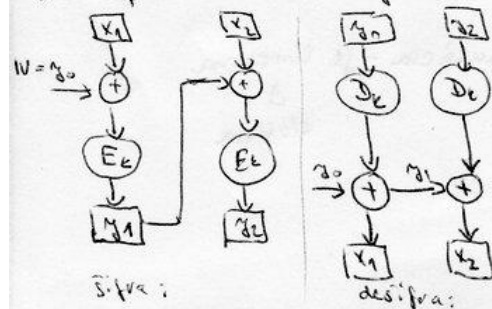
- $\bar{x} = x_1, x_2, \dots, x_n$ { správna } blok
- $\bar{y} = E_k(x_1) E_k(x_2) \dots$
- $y_i = E_k(x_{i-1} \oplus x_i)$

CFB mode: $y_i = E_k(y_{i-1}) \oplus x_i$



Síťva:
desifra:

CB - Cipher Block Chaining mode



Síťva:

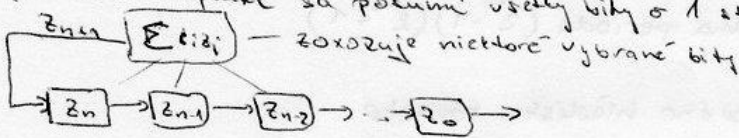
desifra:

OFB: output feedback mode:

- $z_0 = E_k(y_0)$
- $z_1 = E_k(z_0)$
- $z_i = E_k(z_{i-1})$
- $y_i = x_i \oplus z_i$
- $x_i = y_i \oplus z_i$

LF SR - Linear Feedback Shift Register

- SR má n -buniek, do ktorej sa zmestí 1b.
- pri vod. impulze sa posunú všetky bity o 1 stupienok doprava. \rightarrow naplníme najvyšší bit.



- ak dobre nastavíme c_i - LFSR majú dobré vlastnosti.

$$\begin{aligned} c_1 z_1 + c_2 z_2 + \dots + c_n z_n &= z_{n+1} \\ c_1 z_2 + c_2 z_3 + \dots + c_n z_{n+1} &= z_{n+2} \\ &\vdots \\ c_1 z_n + c_2 z_{n+1} + \dots + c_n z_{2n-1} &= z_{2n} \end{aligned}$$

$$\vec{Z}, \vec{C} = \begin{pmatrix} z_{n+1} \\ z_{n+2} \\ \vdots \\ z_{2n} \end{pmatrix} \in \mathbb{Z}_2^n$$

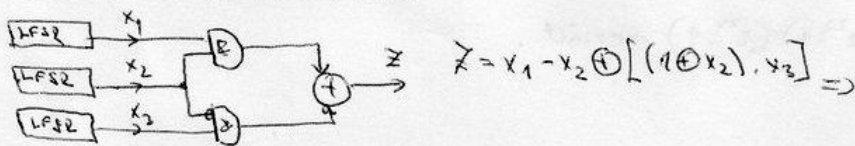
môžeme si zostrojiť polynóm na základe c_i :

$$c_1 x^n + c_2 x^{n-1} + c_3 x^{n-2} + \dots + c_n x + 1 \quad \text{-- connection polynomial.}$$

Aby reg. pracoval dobre, polynóm musí:

- 1) byť primitívny
- 2) ireducibilný (ďalej nerozložiteľný na súčin polynómov) nad \mathbb{Z}_2
- 3) deliť polynóm $x^{2^n-1} + 1$
- 4) nedeliť žiadny taký polynóm $(x^k + 1)$, kt. je deliteľom $2^n + 1$

GEFFEHO GENERÁTOR:



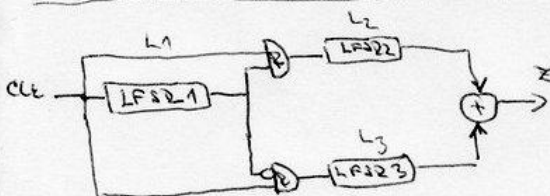
x_1	x_2	x_3	z	$z=x_1$
0	0	0	0	✓
0	0	1	1	✓
0	1	0	0	✓
0	1	1	1	✓
1	0	0	0	✓
1	0	1	1	✓
1	1	0	1	✓
1	1	1	0	✓

$$P(z=x_1) = \frac{5}{8} = \frac{3}{4} \quad \text{-- slabosť}$$

↓
korelačný útok 1. kľúč je počítačové nastavenie generátora

- mápn postupnosť bitov z generátora
- 3 až 4 bitov počat. nastavenia generátora sa budú zhodovať
- pozerám zhody, ak riadne slúpu, možnosť odhalenia

ALternative Step Generator:

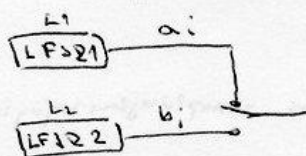


$$(2^{L_1}-1)(2^{L_2}-1)(2^{L_3}-1) \nmid 1) L_1, L_2, L_3 \rightarrow \text{nesúdel. čísla}$$

$$2) \text{ ak } L_1 \approx L_2 \approx L_3 \approx 128$$

1) aj 2) \Rightarrow dobrý generátor (nie je známy útok)

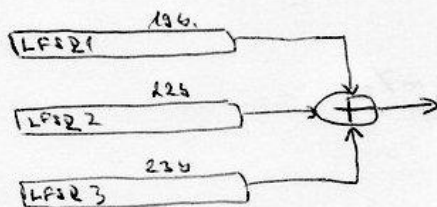
Shrinking Generator:



L_1 a L_2 - sú nesúdeliteľné \rightarrow
 \rightarrow následná perióda $(2^{L_1}-1)(2^{L_2}-1)$

- bity odzial' vyhadzajú z časového hľadiska rovnako.

GSM A5 algoritmus:



klúč je počiatok nastavenie, napr.:

- 2 LFSR1 ... 9-bit $\rightarrow b_1$
- 2 LFSR2 ... 11-bit $\rightarrow b_2$
- 2 LFSR3 ... 11-bit $\rightarrow b_3$

- na základe týchto bitov sa rozhodne, či z generátorov sa posunie

$T()$ - hľa sourceia jeľa - novorí, či. z b_i sa tána najviac vyskytuje

$b_i \oplus T(b_1 b_2 b_3)$ - posuv i-teho registra

- vždy sa posunie aspoň jeden register (inak by to zostalo)

Malo by to mať $(2^9-1)(2^{11}-1)(2^{11}-1)$ periód.

Skúška:

Paluch chce len základy, bolo to v pohode veľa toho nechce; RSA (stacilo mu tých 4-5 bodov, netreba ani ten rozšírený Euklidov algoritmus, ani nič ďalšie), potom bol digitálny podpis, vigenérova šifra, hillovská, afinná, AES k tomu chce len nejaké operácie, aké tam prebiehajú, žiadnu schému a k IDEI tiež staci počet kol, dĺžku bloku a kľuca a že je to založené na Feistelovej schéme - tu treba vedieť, k ostatným obrázky netreba, staci princípy. Žiadne dokazy netreba, stacia úplne základy.

AES – základné veci

AES je bloková šifra, ale NIE JE Feistellovho typu. Blok je 128 bitový (čo znamená 16 bytový)

Kľúč a počet kôl:

Pre 128 bitový kľúč je 10 kôl, pre 192 bitový 12 kôl a pre 256 bitový je 14 kôl.

Priamy text sa rozdelí na bloky a jednotlivé byty v rámci bloku sa usporiadajú do matice. Blok má 128 bitov, teda 16 bytov. Matica bude 4x4. Takto upravený blok do matice nazývame STATE.

Takto môže vyzerat' ukážka.

Pri šifrovaní treba vedieť postup a princíp jednotlivých operácií. Ako získať z kľúča kolové kľúče sa nepýta, lebo by sa šlo moc do hĺbky a on skúša základné veci a princípy.

Šifrovanie:

```
AddRoundKey()  
For i:= 1 to počet kôl - 1 do  
Begin  
    SubByte()  
    ShiftRows()  
    MixColumns()  
    AddRoundKey()  
End;  
SubByte()  
ShiftRows()  
AddRoundKey()
```

AddRoundKey:

Ako som spomínal, každý blok je usporiadaný do štvorcovej matice state 4x4. Operácia AddRoundKey spraví pre každý byte bitový XOR s kľúčom, teda zoxoruje dve matice rovnakej veľkosti po bitoch.

Operácia **SubByte** spraví nasledovné:

Pre každý zo 16 bytov urobí inverzný byte, teda akoby byte na mínus prvú, potom ho vynásobí maticou 8x8 (bitovo) a potom k tomu ešte pripočíta nejaký vektor. Teda možno povedať, že pre každý byte sa spraví akoby afinná šifra.

ShiftRows vykonáva operácie s riadkami matice. Prvý riadok nechá tak, ako je. Druhý posunie rotačne o jeden byte doľava. Tretí posunie o dva byty doľava a posledný o tri byty doľava.

MixColumns vykoná nad každým stĺpcom určitú transformáciu – neviem presne akú, ale je dôležité, že nad každým stĺpcom transformáciu.

Pri **dešifrovaní** sa robia tieto operácie v opačnom poradí a samozrejme inverzne.