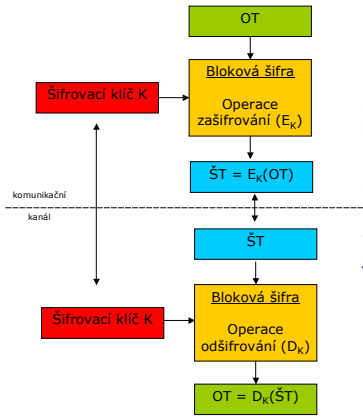
Blokové šifry

RNDr. Vlastimil Klíma
vlastimil.klima@i.cz

Obsah

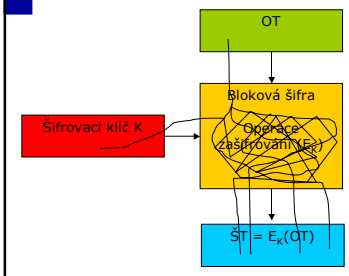
- Blokové šifry
  - principy
  - vybrané blokové šifry (DES, 2DES, 3DES, AES)



zašifrování, odšifrování, - obecně různé operace

- mohou využívat jeden HW

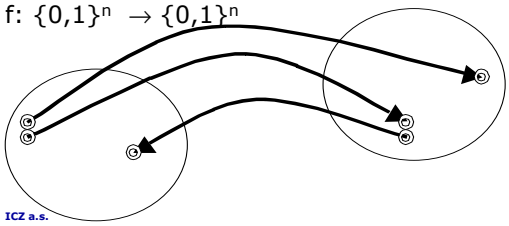
Vlastnosti blokových šifer



- difúze, konfúze, úplnost (vzhledem k OT a ke klíči)
- difúze odráží to, jak jednotlivé bity vstupu (OT,K) pronikají do jednotlivých bitů výstupu (ŠT)
- konfúze je složitost této závislosti
- úplnost je stav, kdy každý bit výstupu závisí na každém bitu vstupu

Blokové šifry jako náhodné permutace

- kvalitní n-bitové blokové šifry se jeví jako náhodné permutace na množině n-bitových bloků
- f: $\{0,1\}^n \rightarrow \{0,1\}^n$



Difúze, konfúze a úplnost u klasických šifer

- substituce, transpozice, aditivní šifry: difúze a konfúze je velmi slabá (vzhledem k OT i ke klíči)

klíč:

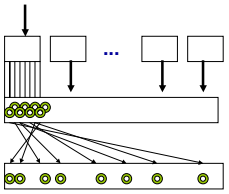
OT	..	D	E	J	M	P	Q	S	T	U	V	..
ŠT	..	T	Y	Z	U	B	W	X	C	V	A	..

zprávy:

OT	S	E	J	D	E	M	E	S	E	U	**Q**	V	P	E	T
ŠT	X	Y	Z	T	Y	U	Y	X	Y	V	**W**	A	B	Y	C
OT	S	E	J	D	E	M	E	S	E	U	**M**	V	P	E	T
ŠT	X	Y	Z	T	Y	U	Y	X	Y	V	**U**	A	B	Y	C

Stavební prvky blokových šifer a jejich vlastnosti

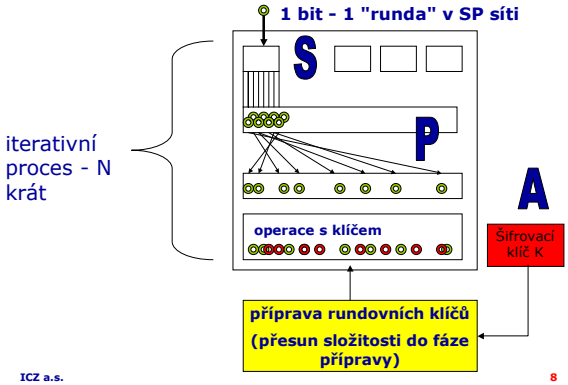
- substituce na úrovni bajtů a permutace na úrovni několikabajtových slov (např. 32b) ... SP síť dosahuje požadovaných vlastností (difúze, konfúze, úplnost)



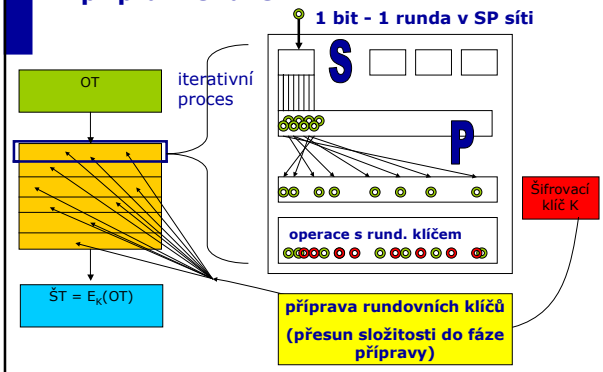
S
P

- při n násobném opakování (S, P) obdržíme náhodnou permutaci na množině $\{0,1\}^{32}$
- smysl SP sítě bez klíče je omezený

SP síť s klíčem



Vliv klíče a otevřeného textu, význam přípravné fáze



Linearita, nelinearita v SP síti a úloha substitučního boxu (S box)

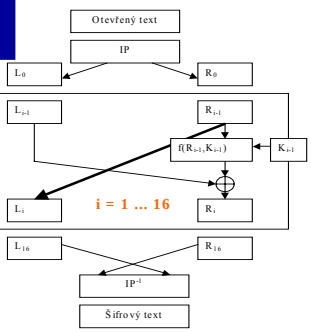
x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8
0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	1
0	0	0	0	0	0	0	1	0	1	1	1	1	1	0	1
0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0
0	0	0	0	0	0	1	1	1	1	1	1	0	1	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

- tabulka, rovnice
- jaký vliv má lineární S box v SP předchozí síti ?
- $y_1 = x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5 \cdot x_6 \cdot x_7 \cdot x_8 \oplus 0 \oplus 0 \oplus x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5 \cdot x_6 \cdot x_7 x_8 \oplus .....$

Blokové šifry: nelinearita

- vliv nelinearit S boxu v SP síti (stupeň výsledného polynomu)
- z teoreticko-informačního hlediska postačuje k luštění několik dvojic (OT, ŠT)
- z praktického hlediska je luštitelnosti bráněno výpočetní složitostí (NP-úplné problémy, soustava B. rovnic)
- geniální objev může zhatit značnou část současné kryptografie
- (nepodmíněná a podmíněná bezpečnost - viz předchozí přednáška, ... *délka klíče kompenzována složitostí algoritmu*)

DES – základní schéma

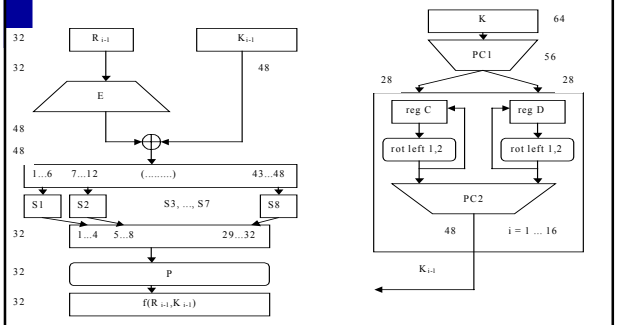


Vstup: OT 64b, klíč K 64b
Výstup: ŠT 64b

1. Příprava klíčů. Vypočti 16 rundovních klíčů K_i z klíče K
2. Počáteční permutace
3. 16 **rund** pro i=1..16 :
   $L_i = R_{i-1}$
   $R_i = L_{i-1}$ xor $f(R_{i-1},K_{i-1})$
4. Vyměň bloky L_{16}, R_{16}
5. Závěrečná permutace

Feistelův princip:
- dešifrování - stejné schéma, opačné řazení rundovních klíčů
- rundovní funkce nemusí být invertibilní

DES - funkce f(R,K) a příprava klíčů



DES

- Pro klíč i otevřený text: úplnost, difúze, konfúze
- Komplementárnost (1976 Stanford)
  - snižuje složitost útoku hrubou silou o jeden bit
- Slabé a poloslabé klíče (1976, Stanford)
  - 4 slabé klíče: $E_K(X) = X$
    - *0101 0101 0101 0101*, FEFE FEFE FEFE FEFE, *1F1F 1F1F 0E0E 0E0E*, E0E0 E0E0 F1F1 F1F1
  - 6 dvojic poloslabých klíčů $(K_1,K_2) : E_{K2}(E_{K1}(X)) = X$
    - 01FE01FE01FE01FE FE01FE01FE01FE01, *1FE01FE00EF10EF1 E01FE01FF10EF10E*, 01E001E001F101F1 E001E001F101F101, *1FFE1FFE0EFE0EFE FE1FFE1FFE0EFE0E*, 011F011F010E010 E1F01 1F010E010E01, *E0FEE0FEF1FEF1FE FEE0FEE0FEF1FEF1*

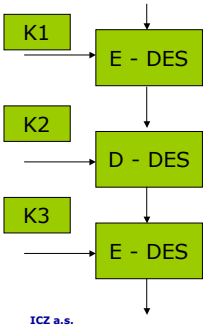
DES

- Teoretické útoky DCA a LCA
  - **DCA** (1990, 1992), Biham a Shamir - CPA, nutno **volit** 2^{47} OT, analyzovat 2^{36} OT, 2^{37} šifrování
  - LCA (1993) Matsui, (1994) LCA s 2^{43} známými **náhodně generovanými** OT a složitostí 2^{43} (12 PC 99MHz, 50 dní)
- Praktický útok – malá délka klíče
  - DES-Cracker, 17.7.1998, HW stroj, v ceně cca 130 000 USD za HW, umožňuje **brute-force attack do 9 dní**
  - Výzva DES Challenge III,19.1.1999 za **22 hod.15 min.**, kombinace Distributed.Net a DES-Crackeru

DoubleDES

- 2DES – Diffie, Hellman 1977
  - místo 2^{112} šifrování a dvou známých párů (OT,ŠT) : KPA, **2 páry** (OT,ŠT), 2^{57} šifrování, 2^{56} dalších operací, 2^{56} jednotek paměti
  - time-memory trade-off obecněji

TripleDES



- Umělé zesílení DES, 1999 FIPS PUB 46-3
- Prodloužení klíče na 56 (+ 56) + 56 bitů
- 3DES_112
- 3DES_168
- používá se všude tam, kde je potřeba schválený a bezpečný algoritmus a nevadí zpomalení

TripleDES

- 3DES-EDE se dvěma klíči – Tuchman, 1978
  - Merkle, 1979, složitost útoku: CPA, **2^{56} párů** (OT,ŠT), 2^{56} šifrování, 2^{56} dalších operací, 2^{56} jednotek paměti
- 3DES-EDE se třemi klíči - DH 1977 a Merkle 1979
  - místo **2^{168} šifrování** a třech známých párů (OT,ŠT):
  - Lucks, 1998: složitost útoku CPA:
    - **2^{16} párů** (OT,ŠT), **2^{106} šifrování**, 2^{112} operací, 2^{72} jednotek paměti
    - **2^{32} párů** (OT,ŠT), **2^{90} šifrování**, 2^{113} operací, 2^{88} jednotek paměti

AES - základní údaje

- soutěž vyhlášena v lednu 1998, z 15 do finále 5 algoritmů: RC6, Twofish, MARS, Serpent, Rijndael
- vítěz Rijndael [:Rejndál:] [:Rájndol:](Belgičané Rijmen, Daemen)
- FIPS PUB 197, pravděpodobně se opět stane nejrozšířenějším algoritmem na světě, **bez licence**
- platí od 26.5.2002 - k ochraně *neutajovaných* informací pro federální orgány USA
- 128bitová šířka bloku
- délky klíčů - 128, 192, 256 bitů (tj. Nk = 4, 6, 8 32bitových slov)

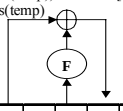
AES - příprava rundovních klíčů

Délka klíče Nk = 4/6/8 32bitových slov => počet rund r = 6+Nk=10/12/14

Expanze: pro i = Nk ... 4*r+3 **W[i] = W[i-Nk] xor F(W[i-1])**

Funkce F: temp=(u_0,u_1,u_2,u_3) -> W=(v_0,v_1,v_2,v_3) je definována takto:
Je-li i mod Nk = 0, pak W = SubBytes(RotBytes(temp)) xor Rcon[i/Nk -1]
Je-li i mod Nk = 4 a Nk = 8, pak W = SubBytes(temp)

V ostatních případech W = temp



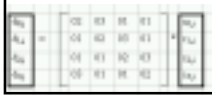
k_0	k_4	k_8	k_{12}							u_0				
k_1	k_5	k_9	k_{13}							u_1				
k_2	k_6	k_{10}	k_{14}							u_2				
k_3	k_7	k_{11}	k_{15}							u_3				

W[0] W[1] .... klíč ..W[Nk-1] expanze ....W[i-Nk] ....W[i-1] W[i] .....

konstanty: Rcon[j] = (x^{i-1}, ´0´,´0´,´0´), kde x^{i-1} a ´0´ jsou prvky GF(2^8)
modul m(x) = $x^8 + x^4 + x^3 + x^1 + 1$, RotBytes ($u_0$,$u_1$,$u_2$,$u_3$) = ($u_1$,$u_2$,$u_3$,$u_0$)

Algoritmus AES

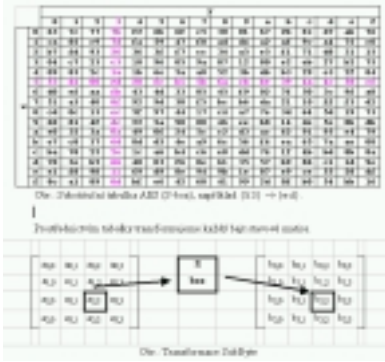
S-box



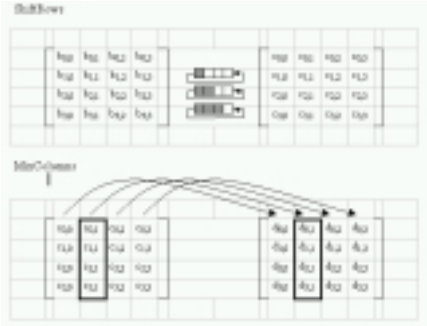
- stavová matice, OT, ŠT
- není Feistelova typu, SP síť
- algoritmus pro zašifrování: *příprava rundovních klíčů*, whitenning, 10/12/14 rund
  - 1 runda:
    - operace SubBytes, ShiftRows, MixColumns, AddRoundKeys
    - poslední runda neobsahuje MixColumn
- bajty reprezentují polynomy v GF(2^8)
- (b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0) <=> b_7x^7 + ... $+b_1x^1 + b_0$
- výpočty modulo m(x) = $x^8 + x^4 + x^3 + x^1 + 1$:
- {57} + {83} = ($x^6 + x^4 + x^2 + x^1 + x^0$) + ($x^7 + x^1 + x^0$) = $x^7 + x^6 + x^4 + x^2$ = {D2}

{57} * {83} = ($x^6 + x^4 + x^2 + x^1 + x^0$) * ($x^7 + x^1 + x^0$) = $x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + x^0$, po dělení m(x) zbytek $x^7 + x^6 + x^0$, tj. výsledek = {C1}

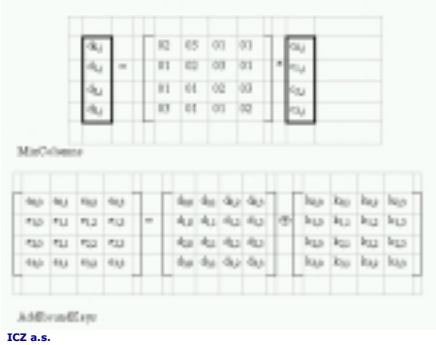
S-box



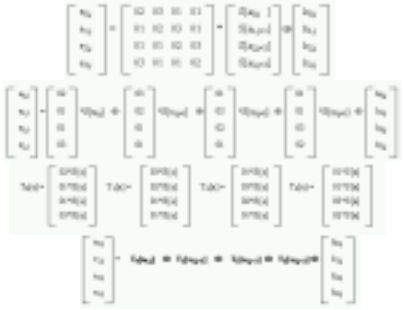
ShiftRows and MixColumns



AddRoundKey



Optimalizace rundovní funkce pro 32bitové procesory



Násobení modulo m(x) = (x⁸ + x⁴ + x³ + x¹ + 1)

- $d_{0,j} = \{02\} * c_{0,j} + \{03\} * c_{1,j} + \{01\} * c_{2,j} + \{01\} * c_{3,j}$
- v = {02}*a =
  $(x^1)*(a_7x^7+a_6x^6+a_5x^5+a_4x^4+a_3x^3+a_2x^2+a_1x^1+a_0x^0)$
  $= (a_7x^8+a_6x^7+a_5x^6+a_4x^5+a_3x^4+a_2x^3+a_1x^2+a_0x^1) =$

je-li $a_7 = 0$, pak v = $a_6x^7+a_5x^6+a_4x^5+a_3x^4+a_2x^3+a_1x^2+a_0x^1$= (a << 1)

je-li $a_7 = 1$, pak

v = $(a_7x^8+a_6x^7+a_5x^6+a_4x^5+a_3x^4+a_2x^3+a_1x^2+a_0x^1) + a_7*m(x)$

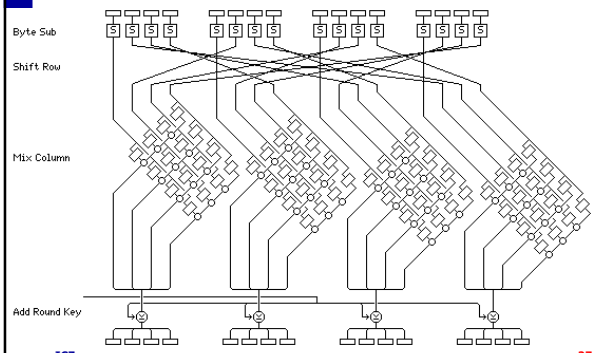
= $(a_7x^8+a_6x^7+a_5x^6+a_4x^5+a_3x^4+a_2x^3+a_1x^2+a_0x^1) +$
$a_7x^8 \quad +a_7x^4+a_7x^3 \quad +a_7x^1 + a_7 =$

= ($a_6x^7+a_5x^6+a_4x^5+(a_3+a_7)x^4+(a_2+a_7)x^3+a_1x^2+(a_0+a_7)x^1+a_7$)

$\{a_7,a_6,a_5,a_4,a_3,a_2,a_1,a_0\} \rightarrow$
$\{a_6,a_5,a_4,(a_3+a_7),(a_2+a_7),a_1,(a_0+a_7),a_7\}$

= (a << 1) + a_7{1B} 1B = 0001 1011

AES: schéma rundovní funkce a nelinearita



Substituční box

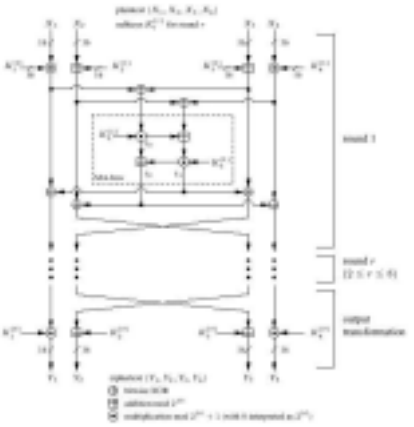
- S(x) = L(Inv(x))
- Inv: x --> x^{-1} v GF(2^8)
  - tj. x*Inv(x) = 1 mod m(x)
  - nelineární operace
- L: b --> b´= M*b + c

Substituční box

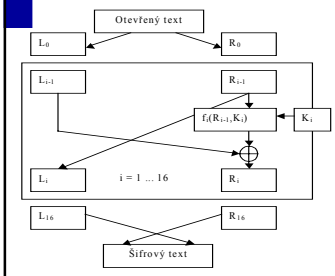
- $y_1 = x_1´*x_2´*x_3´*x_4´*x_5´*x_6´*x_7´*x_8´ \oplus 0 \oplus 0 \oplus x_1´*x_2´*x_3´*x_4´*x_5´*x_6´*x_7*x_8 \oplus$ .....
  - Fuller and Millan, 2002: On Linear Redundancy in the AES S-Box:
  - $y_j = y_1(D_{1j}x) + c$, c =0,1, D_{1j} jsou konstantní binární matice 8x8, j = 2...8
  - Courtois and Pieprzyk, 2002 : Cryptanalysis of Block Ciphers with Overdefined Systems of Equations:
  - mezi x a y existují rovnice 2. řádu: $f(x_1,..,x_8,y_1,..,y_8) = 0$

Vybrané principy konstrukce u dalších blokových šifer

IDEA



CAST



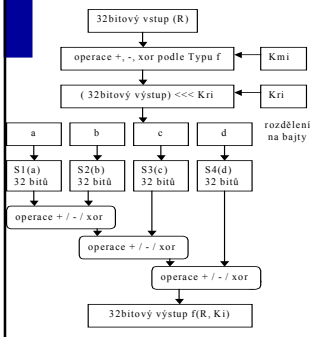
Vstup:
otevřený text m1...m64
klíč K = k1...k128.

Výstup: šifrový text c1...c64

1. Příprava klíčů. Vypočti 16 párů rundovních klíčů Ki = {Kmi, Kri} z klíče K
2. (L0,R0) <-- (m1...m64). Rozděl vstup na 32bitové poloviny L0 = m1...m32 a R0 = m33...m64.
3. Vykonej (12 event.) 16 rund pro i=1..16 : Li = Ri-1;
Ri = Li-1 ⊕ fi(Ri-1,Kmi,Kri), kde
f1,f2,f3,f4,f5 ... jsou různé - typu 1,2,3,1,2 ...
4. c1...c64 <-- (R16,L16). Vyměň bloky L16, R16 a zformuj výstup.

Dešifrování - stejné schéma, opačné řazení rundovních klíčů

CAST - Funkce f(R,K_i)



Typ 1: I = ((Kmi + R) <<< Kri)
I = (a, b, c, d)
f = ((S1[a] ⊕ S2[b]) - S3[c]) +S4[d]

Typ 2: I = ((Kmi ⊕ R) <<< Kri)
I = (a, b, c, d)
f =((S1[a] - S2[b]) + S3[c]) ⊕ S4[d]

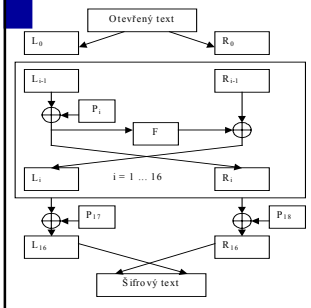
Typ 3: I = ((Kmi - R) <<< Kri)
I = (a, b, c, d)
f = ((S1[a] + S2[b]) ⊕ S3[c]) -S4[d]

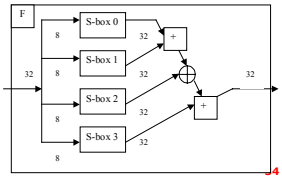
Rundy 1, 4, 7, 10, 13, 16 používají f typu 1
Rundy 2, 5, 8, 11, 14 používají f typu 2
Rundy 3, 6, 9, 12, 15 používají f typu 3

Blowfish

z klíče vypočteny rundovní klíče P_1 až P_{18} a S-boxy ve funkci F

Feistelův princip
Dešifrování - stejné schéma, opačné řazení rundovních klíčů P_{18} až P_1

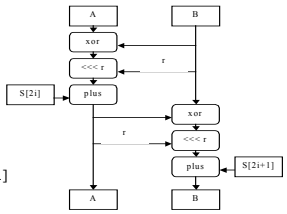




RC5

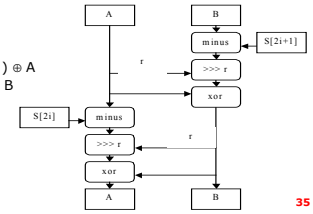
Zašifrování

```
A = A + S[0];  B = B + S[1];
for (i = 1 ; i <= r ; i++)
{
  A = ( (A ⊕ B) <<< B) + S[2*i]
  B = ( (B ⊕ A) <<< A) + S[(2*i)+1]
}
```



Odšifrování

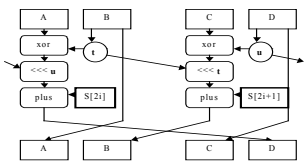
```
for (i = r ; i >= 1 ; i--)
{
   B = (  (B - S[(2*i)+1]) >>> A ) ⊕ A
   A = (  (A - S[2*i]) >>> B )  ⊕  B
}
A = A - S[0];  B = B - S[1];
```



RC6

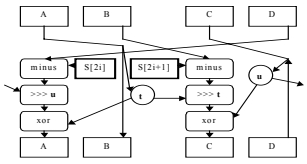
Zašifrování

```
      B = B + S[ 0 ], D = D + S[ 1 ]
for  i  =  1  to  20  do
{   t  =  ( B  x  ( 2B  +  1 ) )  <<<  5
    u  =  ( D  x  ( 2D  +  1 ) )  <<<  5
    A  =  ( ( A ⊕ t )  <<<  u )  +  S[ 2i ]
    C  =  ( ( C  ⊕ u )  <<<  t )  +  S[ 2i + 1]
    (A, B, C, D)  =  (B, C, D, A)
}        A = A + S[ 42 ], C =  C + S[ 43 ]
```



Odšifrování

```
      C =  C - S[ 43 ], A =  A - S[ 42 ]
for  i  =  20  downto  1  do
{   (A, B, C, D)  =  (D, A, B, C)
    u  =  ( D  x  ( 2D  +  1 ) )  <<<  5
    t  =  ( B  x  ( 2B  +  1 ) )  <<<  5
    C  =  ( ( C - S[ 2i + 1 ] ) >>> t ) ⊕ u
    A  =  ( ( A - S[ 2i ] ) >>> u ) ⊕ t
}        D = D - S[ 1 ], B = B - S[ 0 ]
```



Přehled vybraných algoritmů

3DES	64	14, 21 B	ne	FIPS 46-3	1	4,748	
CAST	64	5-16 B	ne	RFC 2144	3,8	18,054	utajované inf.
RC5	var.	0-255 B	ano	RFC 2040	8,3	39,421	
Blowfish	64	4-56 B	ne	lit.,web	3,8	18,051	
IDEA	64	16 B	ano	lit.,web	2,4	11,341	
Skipjack	64	10 B	ne	FIPS 185	1,1	5,326	utajované inf.
RC6	128	16,24,32 B	ano	web	6,9	32,524	
Twofish	128	16,24,32 B	ne	web	5,4	25,667	
MARS	128	16,24,32 B	ano	web	6,3	30,107	
AES	128	16,24,32 B	ano	FIPS 197	6,4	30,325	
RC4	proud	1-256 B	ano	web	13,3	63,039	

MS Visual C++ 6.0 SP4, PC/Celeron 850MHz, Windows 2000,
source: http://www.eskimo.com/~weidai/benchmarks.html

Literatura a další zdroje

Osobní stránka autora
http://cryptography.hyperlink.cz
Archiv článků a prezentací na téma kryptografie a bezpečnost
http://www.decros.cz/bezpecnost/_kryptografie.html
Stránka NIST, normy, dokumenty k AES aj.:
http://csrc.nist.gov/encryption/aes/aes_home.htm
Zdrojové kódy šifer
ftp://ftp.funet.fi/pub/crypt/cryptography/
Bezpečnostní a kryptografický portál
http://www.cs.auckland.ac.nz/~pgut001/links.html