Symetrické šifry

Šifrovací štandard DES

Katedra matematiky a teoretickej informatiky, FEI TU

Symetrické šifry

- používajú na šifrovanie a dešifrovanie **rovnaký tajný kľúč**
- ich **bezpečnosť** je založená na utajení tohoto kľúča
- našli **uplatnenie** v rôznych aplikáciách ochrany údajov a systémov bezpečnosti
- podľa toho, ako realizujú proces šifrovania a dešifrovania ich **delíme** na:
  1. *blokové* (po blokoch)
  2. *prúdové* (po jednotlivých symboloch)
- veľa algoritmov symetrických blokových šifier používa štruktúru, ktorú označujeme ako **Feistelova bloková šifra**

Princípy symetrických blokových šifier

Blokové šifry - praktický problém, ktorý súvisí s veľkosťou zvoleného bloku:

- blokové šifry transformujú otvorený text o dĺžke *n*-bitov na zašifrovaný text o rovnakej dĺžke *n*-bitov
- blok o dĺžke *n*-bitov môže vytvoriť 2^n rôznych blokov otvoreného textu $\Rightarrow$ bloková šifra realizuje transformáciu 2^n blokov otvoreného textu na 2^n blokov zašifrovaného textu o dĺžke *n*-bitov
- ak *n* - malé $\Rightarrow$ šifrovanie je ekvivalentné klasickej substitučnej šifre, ktorú ľahko prelomíme štatistickou analýzou otvoreného textu
- ak *n* - veľmi veľké $\Rightarrow$ pri ľubovoľnej substitúcii sú štatistické charakteristiky otvoreného textu skryté a kryptoanalýza je zložitá, nevýhodou je však značná zložitosť implementácie, ako aj značná dĺžka kľúča

PR.1. Majme text o dĺžke *n=4* bitov, akú dĺžku bude mať kľúč?

Otvorený text	Zašifrovaný text
0000	0111
0001	1110
0010	0000
0011	0100
0100	1001
0101	1101
0110	0101
0111	0001
1000	1100
1001	0010
1010	0110
1011	1111
1100	1010
1101	1011
1110	0011
1111	1000

Zašifrovaný text	Otvorený text
0000	0010
0001	0111
0010	1001
0011	1110
0100	0011
0101	0110
0110	1010
0111	0000
1000	1111
1001	0100
1010	1100
1011	1101
1100	1000
1101	0101
1110	0001
1111	1011

4-bitový vstup vytvára jeden zo 16 možných vstupných blokov, ktorý sa transformuje na jeden zo 16 možných výstupných blokov o dĺžke 4 bity. Potom 2. stĺpec tabuľky predstavuje kľúč uvedenej substitučnej šifry a má dĺžku $4 \times 2^4 = 64$ bitov.

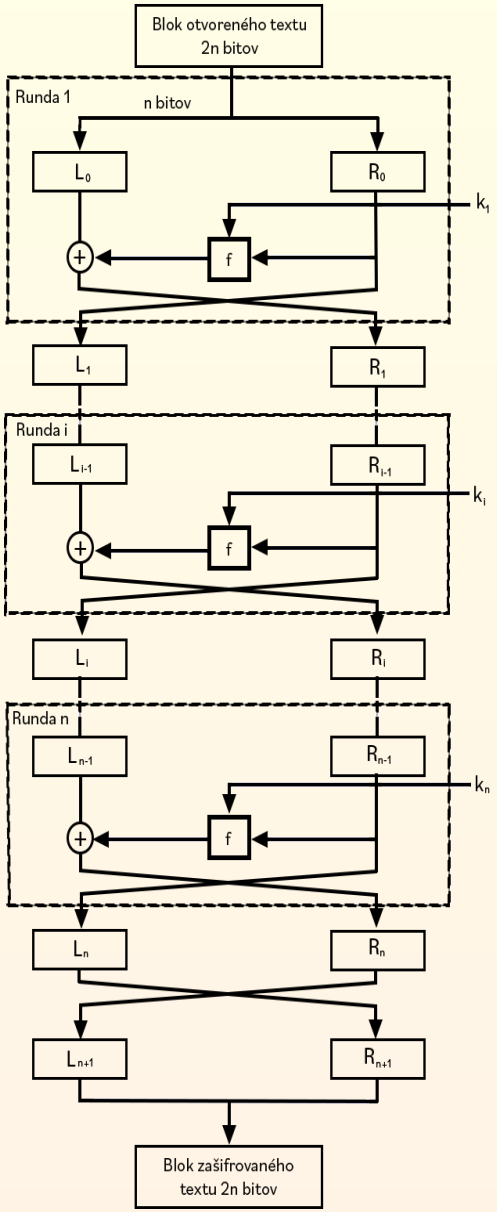
Akým vzťahom je daná dĺžka kľúča pre *n*-bitovú substitučnú blokovú šifru vo všeobecnosti? Určte dĺžku kľúča pre 64-bitovú šifru.

Princípy symetrických blokových šifier

Feistelova bloková šifra:

- nazvaná podľa *Horsta Feistela* – autor šifrovacieho systému LUCIFER, ktorý bol po úprave v r. 1976 prijatý ako štandard DES
- zmierňuje ťažkosti súvisiace s veľkosťou zvoleného bloku
- predstavuje aproximáciu ideálnej blokovej substitučnej šifry pre veľké n
- ide o zloženú šifru, ktorá využíva postupnosť dvoch, resp. viacerých základných šifier tak, že výsledná šifra je z kryptografického hľadiska oveľa silnejšia ako jednotlivé základné šifry
- využíva striedavo substitúcie a permutácie, ktoré realizujú funkcie konfúzie a difúzie

Štruktúra Feistelovej blokovej šifry



Štruktúra Feistelovej blokovej šifry

Konkrétna realizácia blokovej šifry na báze štruktúry Feistelovej šifry je daná voľbou týchto parametrov:

- **veľkosť bloku** - väčší rozmer zvyšuje bezpečnosť, no znižuje rýchlosť šifrovania a dešifrovania
- **dĺžka kľúča** - väčšia dĺžka zvyšuje bezpečnosť, ale znižuje rýchlosť šifrovania a dešifrovania
- **počet rúnd** - znamená počet opakovaní rovnakej postupnosti operácií; volí sa ako kompromis medzi rýchlosťou a stupňom bezpečnosti šifry
- **algoritmus generovania kľúčov rúnd**
- **zložitosť operácie v runde** - väčšia zložitosť sťažuje kryptoanalýzu šifry

Dešifrovanie Feistelovej blokovej šifry

- **proces dešifrovania** je štruktúrou podobný procesu šifrovania
- **algoritmus dešifrovania** je totožný s algoritmom šifrovania, no vstupom tu je zašifrovaný text a výstupom otvorený text
- **kľúče rúnd** sa ale používajú **v opačnom poradí**, t.j. ako prvý sa použije k_n,druhý k_{n-1} atď. a ako posledný kľúč k_1
- táto vlastnosť je **silnou stránkou Feistelovej blokovej šifry**, lebo na šifrovanie a dešifrovanie *nie sú* potrebné dva odlišné algoritmy

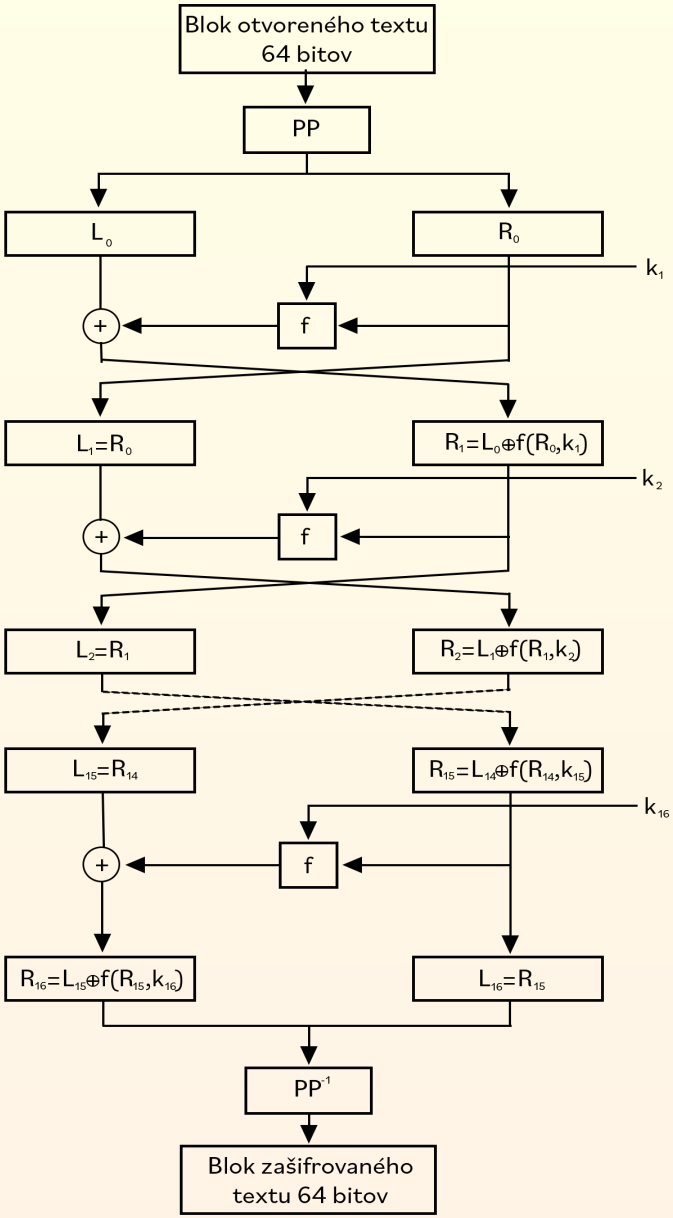
Šifrovací štandard DES

- vyvinul sa z algoritmu, na ktorom pracovala firma IBM a ktorý dostal označenie **LUCIFER** - Feistelova bloková šifra, ktorá pracuje so *64-bitovými blokmi dát* a *kľúčom dĺžky 128 bitov*
- šifrovací štandard **DES (Data Encryption Standard)** používa *64-bitové bloky dát* a *kľúč dĺžky 56 bitov*
- tento štandard bol prijatý v roku **1977** americkými inštitúciami pre bezpečnosť NBS, neskoršie **NIST** (National Institute of Standard and Technology)
- pôvodne bol DES určený pre **aplikácie vo finančníctve**
- v roku **1999** bola prijatá nová verzia DES-u, označená ako **3DES** - využíva trojnásobné šifrovanie pomocou DES a používa pritom dva, resp. tri rôzne kľúče

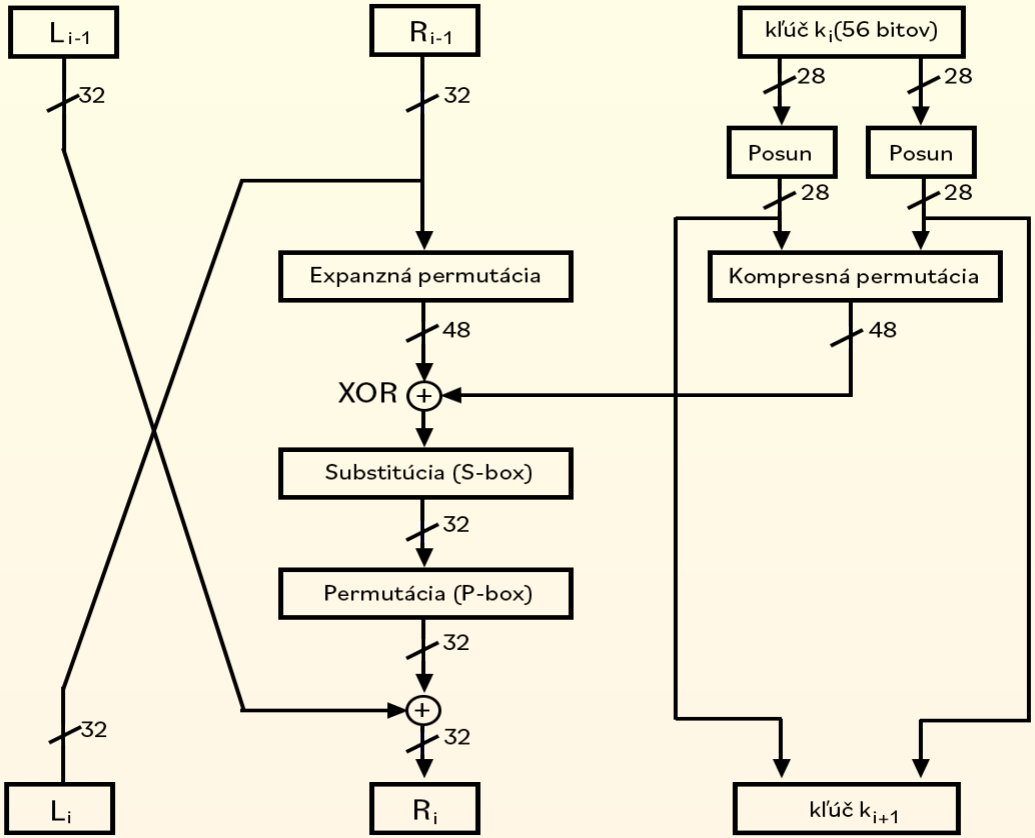
Opis algoritmu DES

- ide o blokovú šifru založenú na **Feistelovej štruktúre**
- šifruje otvorený text po **blokoch o dĺžke 64-bitov** - výstupom je teda 64-bitový blok zašifrovaného textu
- **dĺžka kľúča je 56 bitov** - zadáva sa ako 64-bitové číslo, z ktorého sa vynechaním každého 8 bitu vytvorí skutočný kľúč
- **bezpečnosť DESu** spočíva výlučne v utajení kľúča
- proces šifrovania zahŕňa **16 rúnd**, v ktorých sa používajú rôzne kľúče vytvorené zo základného kľúča

Štruktúra algoritmu DES



Opis *i*-tej rundy



$$L_i = R_{i-1}$$

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

Opis *i*-tej rundy

Počiatočná permutácia PP - realizuje sa pred prvou rundou; transformuje vstupný 64-bitový blok textu podľa tabuľky (t.j. presúva 58. bit otvoreného textu do pozície 1, 50. bit do pozície 2, atď.):

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

Konečná permutácia PP^{-1} - inverzná permutácia k PP (aby bolo možné DES použiť na šifrovanie aj dešifrovanie); funkciu konečnej permutácie vyjadruje tabuľka:

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

Počiatočná permutácia nemá na bezpečnosť šifry žiadny vplyv - jej použitie bolo dané technickými prostriedkami v implementácii DES (napr. zbernice procesov), pričom značne komplikovala softvérové implementácie

Opis *i*-tej rundy

Transformácia kľúča - zahŕňa viaceré operácie:

1. najprv sa vykoná úprava 64-bitového kľúča na 56-bitový kľúč vynechaním každého 8.bitu; zároveň sa vykoná permutácia kľúča podľa tabuľky:

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

2. potom sa kľúč rozdelí na dve 28-bitové polovice, ktoré sa posúvajú rotáciou o jeden alebo 2 bity doľava v závislosti na runde:

Runda	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Počet	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

(tým sa zabezpečí, že každý kľúč je vytvorený inou množinou bitov kľúča)

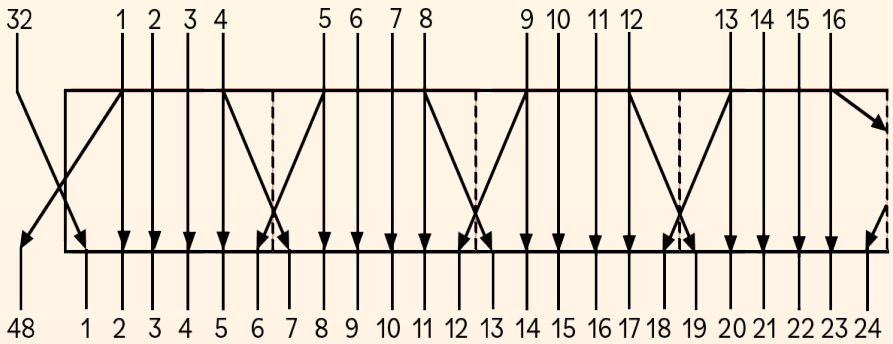
Opis *i*-tej rundy

3. po rotácii sa z 56 bitov vyberie 48 - túto operáciu označujeme ako **kompresná permutácia**, lebo okrem redukcie počtu bitov vykonáva funkciu permutácie:

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

Expanzná permutácia:

- rozširuje prvú polovicu dát R_{i-1} z 32 bitov na 48 + vykonáva funkciu permutácie (rozšírenie počtu bitov je potrebné pre operáciu XOR, ktorá sa vykonáva s výstupom tejto expanznej permutácie a kompresnej permutácie kľúča):



Opis *i*-tej rundy

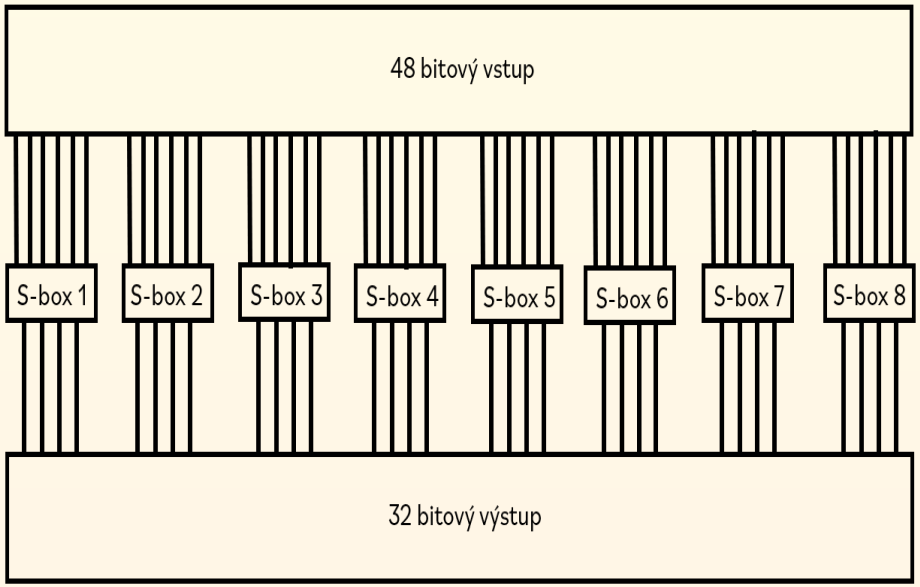
- zároveň pri rozšírení počtu bitov dochádza k **rýchlejšiemu rozptýleniu závislosti výstupných bitov na vstupných bitoch** pri následnej substitúcii
- uvedený jav, ktorý pri malej zmene vstupu vyvoláva veľkú zmenu výstupu, sa označuje ako **lavínový efekt**
- možno ho použiť na **hodnotenie kvality kryptografického algoritmu**, pričom kritériom je zmena počtu bitov zašifrovaného textu pri zmene otvoreného textu, resp. kľúča v jednom bite:
  $\Rightarrow$ čím je zmena počtu bitov zašifrovaného textu väčšia, tým je kvalita kryptografického algoritmu lepšia
  $\Rightarrow$ naopak, pri malej zmene zašifrovaného textu sa redukuje priestor zašifrovaného textu, resp. kľúča na kryptoanalýzu, teda znižuje sa bezpečnosť

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

Opis *i*-tej rundy

Substitúcia:

- realizuje sa v 8 substitučných boxoch (tzv. S-boxy) a tieto realizujú transformáciu 48-bitového vstupu na 32-bitový výstup:



- každý S-box má 6-bitový vstup a 4-bitový výstup a jeho funkcia je vyjadrená tabuľkou so 4 riadkami a 16 stĺpcami (viď skriptá str.79, S-boxy 1 až 8)

Opis *i*-tej rundy

PR.2. Určte výstup S-boxu 6 v algoritme DES, na ktorého vstupe je 010011.

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 6 (010011) = S–box 6 $(b_1b_2b_3b_4b_5b_6)$ =?
Bity b_1b_6 majú hodnotu 01, čo zodpovedá 2.riadku tabuľky.
Bity $b_2b_3b_4b_5$ majú hodnotu 1001, čo zodpovedá 10. stĺpcu tabuľky S–boxu 6.
Na tomto mieste je v tabuľke hodnota 1, čo v 4-bitovom vyjadrení zodpovedá hodnote 0001.
S–box 6 (010011) =0001, t.j. vstupná hodnota 010011 sa po substitúcii v danom S-boxe nahradí hodnotou 0001.

Opis *i*-tej rundy

Permutácia:

- realizuje sa v P-boxe, ktorý má 32-bitový vstup a 32-bitový výstup, t.j. žiaden bit nie je vynechaný, ani sa neopakuje
- označuje sa tiež ako priama permutácia

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

- výsledok permutácie sa sčíta pomocou XOR s ľavou polovicou L_{i-1}, a teda získavame pravú polovicu R_i
- R_{i-1} prepíšeme do L_i, čím získame ľavú polovicu dát, a teda máme dáta pripravené na ďalšiu rundu

Opis *i*-tej rundy

Dešifrovanie:

- používa rovnakú štruktúru rúnd
- jediný rozdiel je, že kľúče sa používajú v opačnom poradí, t.j. $k_{16}, k_{15}, \ldots, k_1$
- algoritmus generovania kľúčov pre príslušné rundy používa posun *doprava* a počet bitov posunutia v rundách je *v obrátenom poradí*

Bezpečnosť algoritmu DES

Bezpečnosť DESu - bola dlho predmetom úvah, ktoré sa sústreďovali na dva hlavné **aspekty**:

1. dĺžka kľúča
2. vlastnosti algoritmu

Dĺžka kľúča:

- vyvolávala od začiatku pochybnosti, a to najmä preto, že pôvodne sa počítalo s dĺžkou kľúča 112 bitov
- jeho skrátenie na 56 bitov však prehlásila NSA za vyhovujúce
- táto dĺžka zabezpečuje možnosť voľby 2^{56} rôznych kľúčov
- existujú v nej však, vzhľadom na spôsob generovania kľúčov pre jednotlivé rundy, aj tzv. **slabé, poloslabé a potenciálne slabé kľúče**

Bezpečnosť algoritmu DES

Slabé kľúče:

- nezabezpečujú vytvorenie rôznych kľúčov pre jednotlivé rundy, resp. generujú rovnaký kľúč pre všetky rundy
- príkladom je kľúč tvorený iba nulami, resp. jednotkami
- celkový počet slabých kľúčov je 4

Poloslabé kľúče:

- vygenerujú namiesto 16 kľúčov len dva
- ich počet je 12

Potenciálne slabé kľúče:

- generujú len 4 rôzne kľúče
- ich počet je 48

Uvedená množina 64 kľúčov je oproti celkovej množine (2^{56}) bezvýznamná, a teda nepredstavuje obmedzenie algoritmu DES.

Bezpečnosť algoritmu DES

Vlastnosti algoritmu DES & bezpečnosť:

- iné úvahy sa sústreďovali na **analýzu S–boxov** a **počet rúnd**
- metodika návrhu S–boxov nebola nikdy zverejnená, a to vyvolávalo rôzne **podozrenia**, že boli zámerne upravené tak, aby to umožnilo ľahšiu kryptoanalýzu pre autorov ich návrhu, resp., že úprava S–boxov umožnila vytvoriť tajný vstup do algoritmu - uvedené úvahy sa však nepotvrdili
- po rôznych skúsenostiach o prelomení algoritmu a možnosti skonštruovať stroje na prelomenie algoritmu, bol **DES v roku 1998 vyhlásený za nevyhovujúci z hľadiska bezpečnosti**
- je však potrebné poznamenať, že od jeho zavedenia sa vyvinuli viaceré jeho **modifikácie**, z ktorých najvýznamnejší je **3DES**
- nový prístup k blokovému symetrickému šifrovaniu predstavuje algoritmus **AES**, ktorý bol predmetom verejnej súťaže

Ďakujem za pozornosť