Kryptografie a počítačová bezpečnost

Symetrické algoritmy

Literatura

- https://www.cs.vsb.cz/ochodkova/courses/kpb/cryptography-and-network-security_-principles-and-practice-7th-global-edition.pdf
- Kapitola 4.1 – 4.5
- Kapitola 6
- Kapitola 7.1
- https://csrc.nist.gov/csrc/media/projects/cryptographic-standards-and-guidelines/documents/aes-development/rijndael-ammended.pdf

Moderní algoritmy

- Kryptoanalýza moderních šifer je nesrovnatelně složitější než u historických šifer.
- Moderní kryptografické algoritmy (a další bezpečnostní mechanismy) jsou považovány (oprávněně) za „zbraně“.
- Kryptografie a kryptoanalýza jsou zbožím dvojího užití – pro obranu i pro útok. Jsou to zbraně, i podle našeho zákona.
- Import/export: Wassenaarská dohoda o vývozní kontrole klasických zbraní a **zboží dvojího použití (určení)**, http://www.wassenaar.org, Ministerstvo průmyslu a obchodu ČR
- Proto dříve
  - restrikce - např. symetrický algoritmus jen klíč 56b, vývoz dnes poměrně liberální, výjimky 7 zemí (Cuba, Iran, Iraq, Libya, North Korea, Sudan, Syria)
  - Irák a Libye vyřazeny po roce 2002 a dále se to mění
- Nyní EU – kvantová kryprtografie:
  - https://esipa.cz/sbirka/sbsrv.dll/sb?DR=SB&CP=32022R0001

Symetrická kryptografie (SK)

- Proudové, blokové
- Jedny z nejvíce užívaných kryptografických algoritmů.
- Výhody SK:
  - rychlá hardwarová implementace,
  - relativně „krátké“ klíče,
  - mohou být „skládány“ a poté využity pro vytváření silných šifer
  - mohou být využity pro jako základní prvek pro konstrukci různých kryptografických mechanismů (PRNG, digitální podpis, hašovací funkce, ...).
- Nevýhody SK:
  - klíč musí být udržován v tajnosti oběma stranami,
  - management klíčů je náročný (n*(n-1)/2 klíčů) pro n uživatelů,
  - klíč by měl být měněn (jak často?),
  - digitální podpis založený na SK vyžaduje velký „klíč“ nebo TTP.

Porovnání velikosti klíčů

- V současnosti již platí nová doporučení, nicméně nejsou ve formě jedné tabulky, proto pro ilustraci je uvedena tato.
- Jinak nově viz viz https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar2.pdf

NIST Recommendations (2016) - Page 2

Keys length recommendations

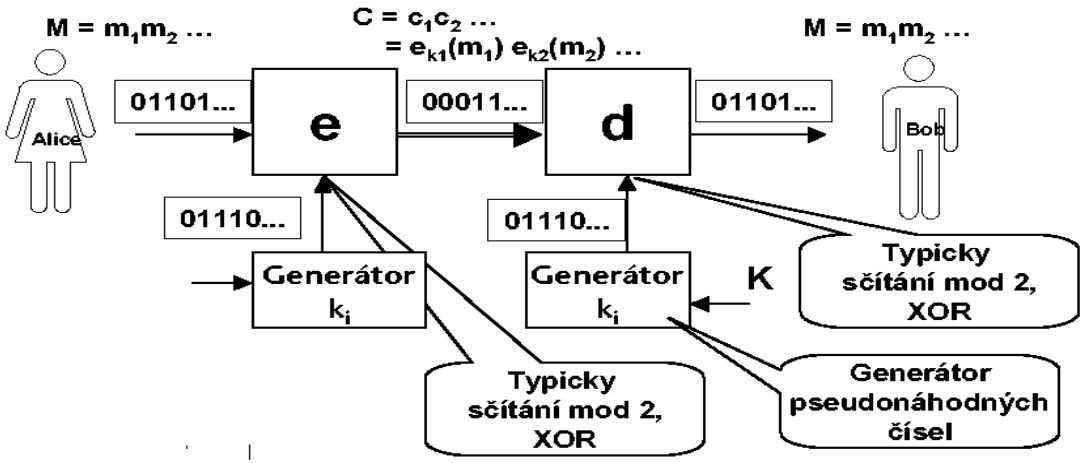
Date	Minimum of Strength	Symmetric Algorithms	Factoring Modulus	Discrete Logarithm Key	Discrete Logarithm Group	Elliptic Curve	Hash (A)	Hash (B)
(Legacy)	80	2TDEA*	1024	160	1024	160	SHA-1**	
2016 - 2030	112	3TDEA	2048	224	2048	224	SHA-224 SHA-512/224 SHA3-224	
2016 - 2030 & beyond	128	AES-128	3072	256	3072	256	SHA-256 SHA-512/256 SHA3-256	SHA-1
2016 - 2030 & beyond	192	AES-192	7680	384	7680	384	SHA-384 SHA3-384	SHA-224 SHA-512/224
2016 - 2030 & beyond	256	AES-256	15360	512	15360	512	SHA-512 SHA3-512	SHA-256 SHA-512/256 SHA-384 SHA-512 SHA3-512

All key sizes are provided in bits. These are the minimal sizes for security.

TDEA (Triple Data Encryption Algorithm) and AES are specified in [10].
Hash (A): Digital signatures and hash-only applications.
Hash (B): HMAC, Key Derivation Functions and Random Number Generation.

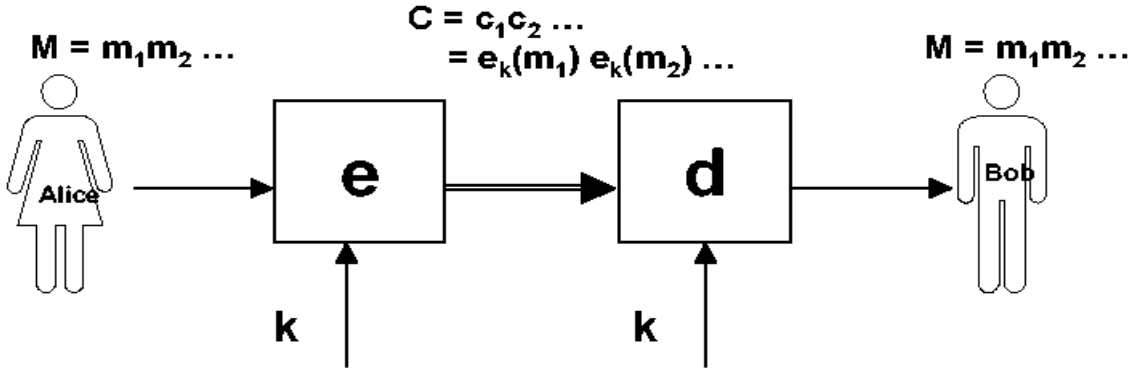
Proudové algoritmy

- Otevřený text se zpracovává bit po bitu nebo byte po bytu.
- Snadná implementace a rychlá šifra.
- Nešíří se chyby - 1 chyba v ŠT způsobí 1 chybu v OT.
- Není ochrana proti manipulaci se šifrou, není zajištěna integrita zprávy.
- Matematicky je lze analyzovat snadněji (nízká úroveň difuse).
- Výhodné pro hardwarovou implementaci.
- Např. RC4, A5, Sober, ….



Blokové algoritmy

- Otevřený text se zpracovává po n-bitových blocích, typicky nyní n≥128.
- ŠT má stejnou délku jako OT
- Délka zprávy musí být podstatně větší než délka bloku, délka bloku musí být dostatečně velká, aby se zabránilo slovníkovému útoku.
- Klíče je třeba často měnit.
- Chyba v jednom bitu OT ovlivní celý blok ŠT



Teoretické základy blokových algoritmů (1)

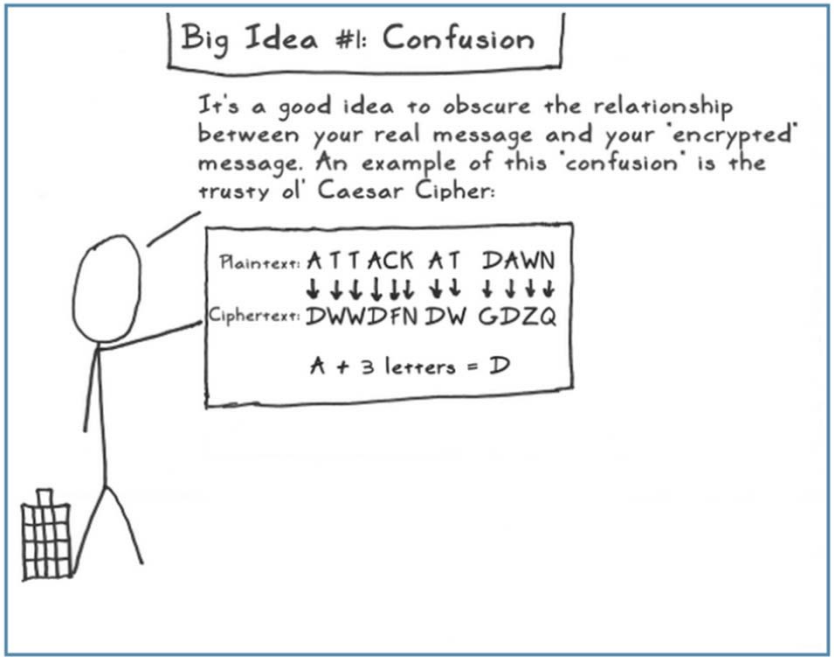
- Koncept moderní kryptografie navrhli C. Shannon a H. Feistel.
  - Claude Shannon: „Communication Theory of Secrecy Systems“, Bell System Technical Journal, Oct 1949,
  - „Prediction and Entropy of printed English“, Bell System Technical Journal, Jan 1951.
- Koncept:
  - entropie, redundance jazyka (redundance ve zprávě je dostačující k jejímu prolomení),
  - teorie o tom kolik informace je třeba pro zlomení šifrového textu -stanovil teoretickou míru bezpečnosti šifry pomocí neurčitosti otevřeného textu, když je dán šifrový text.
- Koncept:
  - Vernamův algoritmus je jediný absolutně bezpečný systém,
  - šíření chyb,
  - výběr klíče
  - Konfuse, difuse (u klasických (historických) šifer je velmi slabá (vzhledem k OT i ke klíči)),
  - Úplnost
  - Lavinový efekt
  - S-P network

Teoretické základy blokových algoritmů (2)

- Shanon:
  - Kvalitní n-bitové blokové šifry se jeví jako náhodné permutace na množině n-bitových bloků, f: $\{0,1\}^n \rightarrow \{0,1\}^n$. Ideální by bylo použít jednu extrémně velkou substituci, což není příliš praktické vzhledem k např. 2^{128} vstupům pro 128-bitový blok.
  - Použijeme tedy menší bloky - **Substitution-permutation (S-P) networks** – moderní forma složených algoritmů.
    - Substituce se realizuje na úrovni bajtů a permutace na úrovni několikabajtových slov (např. 32b) – vzhledem k počtu všech možných substitucí a permutací (snímek 15)
    - SP síť dosahuje požadovaných vlastností (difúze, konfúze, úplnost, lavinový efekt).
    - Při n násobném opakování (S, P) obdržíme náhodnou permutaci na množině např. $\{0,1\}^{32}$
  - Smysl SP sítě bez klíče je omezený.

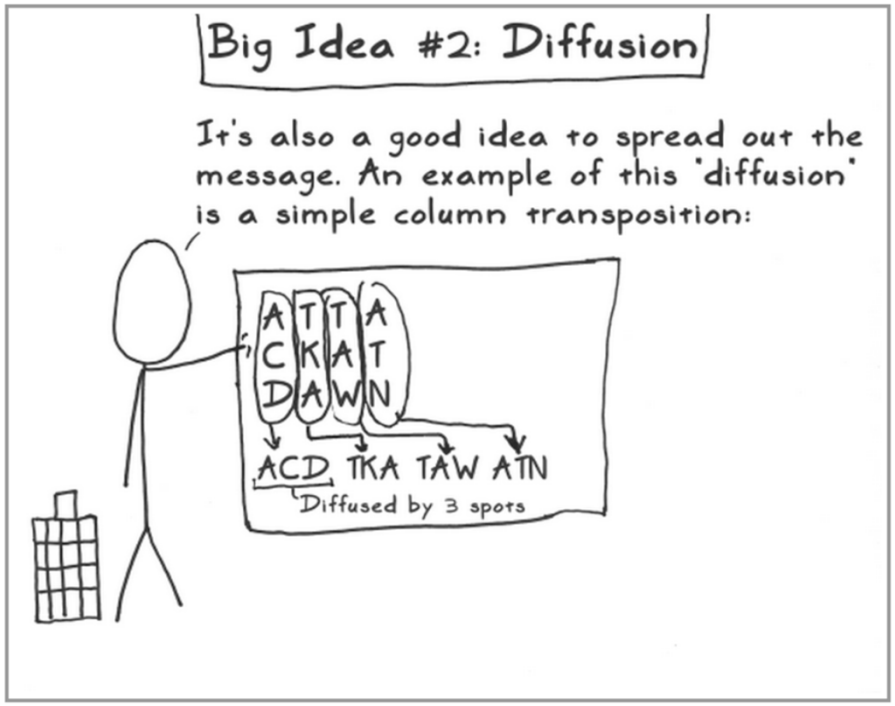
Konfuse

- Konfuse (confusion, záměna, „zmatení“) - komplikuje vztahy mezi statistikou ŠT a klíčem
- realizováno substitucí

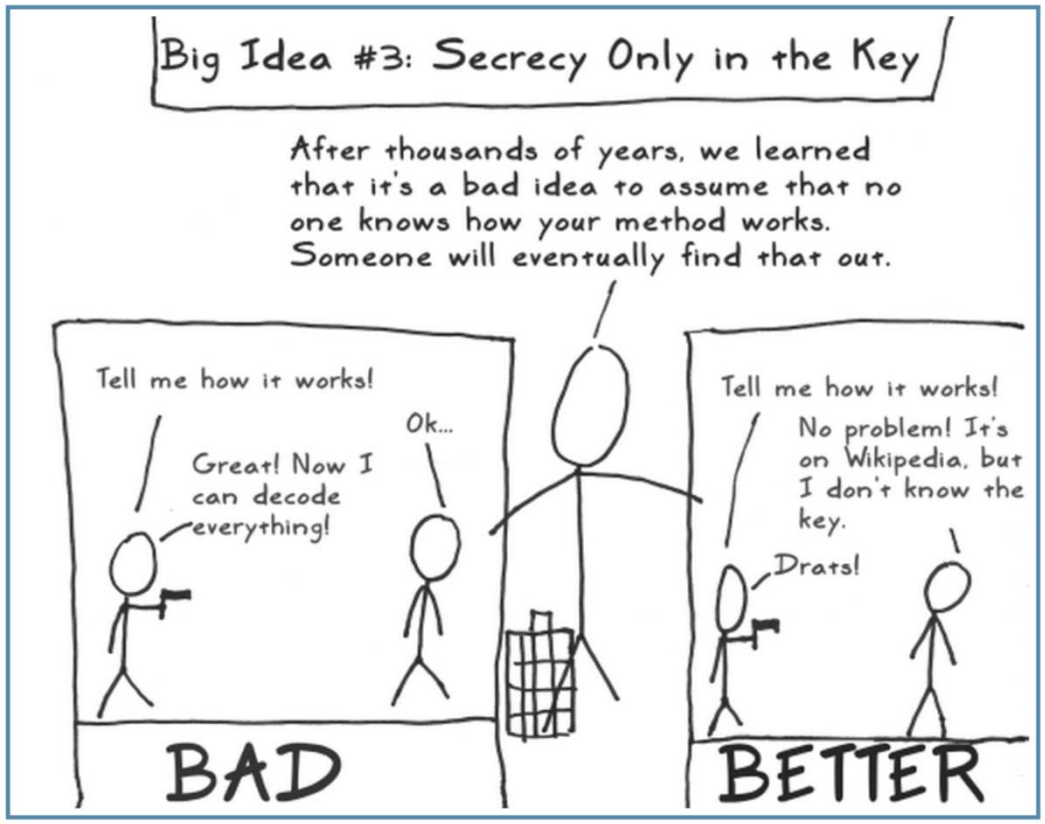


Difuse

- Difuse (diffusion, rozptýlení) - rozptýlení statistiky OT po celé ŠT,
- realizováno permutací



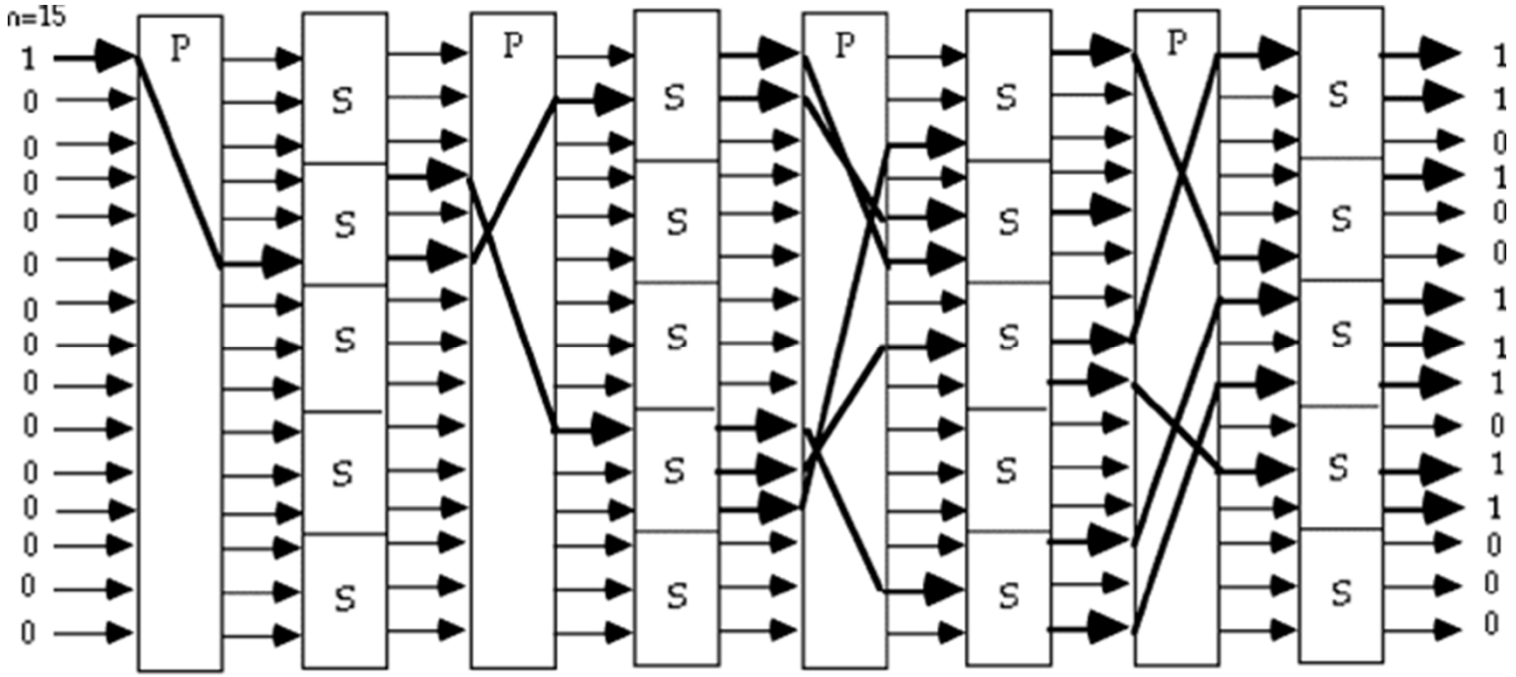
Kerckhoffův princip



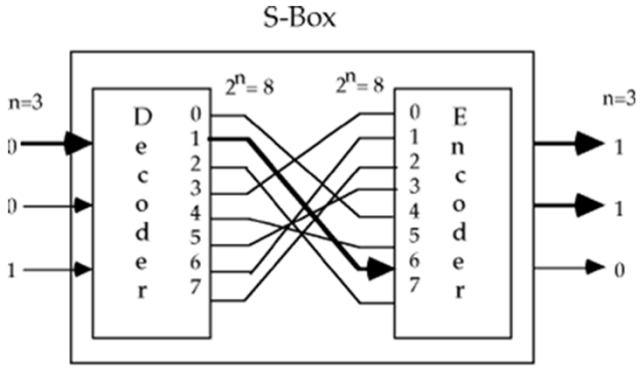
Konstrukce blokových algoritmů

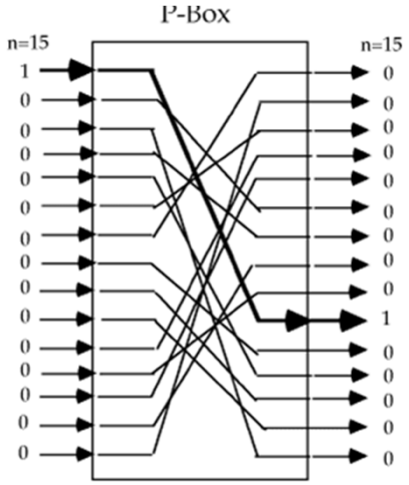
- **Avalanche** (lavinový) efekt – změna jednoho vstupního bitu má vliv na změnu cca poloviny výstupních bitů.
  - Obecně pro funkci f: Pro každý bit i, $0<=i<m$, jestliže 2^m vektorů OT rozdělíme na 2^{m-1} dvojic X a X_i (každý pár se liší jen v bitu i) a jestliže 2^{m-1} provedeme jejich XOR, pak porovnáme-li $V_i = f(X) \oplus f(X_i)$ zjistíme, že asi polovina těchto součtů je 1.
- **Completeness** (úplnost) – každý výstupní bit je funkcí všech vstupních bitů (takže útočník nemůže při útoku použít strategii "rozděl a panuj").
  - Obecně pro funkci f: Pro každý bit j, $0<=j<m$ výstupního (šifrového) vektoru, existuje nejméně jeden pár vektorů OT X a X_i, který se liší jen v bitu i, a pro který $f(X)$ a $f(X_i)$ se liší v bitu j.
- Dobrý návrh (konstrukce) algoritmu má avalanche, completeness, nepředvídaltelnost, nahodilost.
- Špatný návrh - algoritmus má nedostatek nahodilosti a příliš mnoho předvídatelnosti.

Lavinový efekt



Složené algoritmy





- Substituce:
  - Binární slovo je nahrazeno jiným binárním slovem.
  - Pokud použijeme n-bitová slova, klíč je 2^n, roste velmi rychle s rostoucím n (možných $(2^n)!$ substitucí).
  - Tzv. ***S-boxy***
- Permutace:
  - Binární slovo je „přeuspořádáno“ (permutováno), permutace formuje klíč, tedy pro n-bitové slovo máme klíč n bitový.
  - Roste pomaleji, je tak méně bezpeč. než substituce, jen n! možných permutací.
  - Tzv. ***P-boxy.***

Počet S-boxů

An s-box is a bijective function $f : \{0,1\}^b \to \{0,1\}^b$. This reduces the question to "how many of those f exist".

To see this easily, imagine domain and image of this function as two boxes with 2^b elements. How many different sets of arrows can you imagine between those two sets?

For the first arrow, you have 2^b choices. For the second arrow, you have $2^b - 1$ choices. For the third, $2^b - 2$, and so forth to the last arrow, where there's one choice left.

Multiplying those together, you would get $2^b(2^b - 1)(2^b - 2)\ldots 1 = 2^b!$ possible s-boxes, so in your 16-bit case that makes $2^{16}!$ possibilities.

To put this more mathematically, an s-box is a permutation of 2^b elements, and there are $2^b!$ such permutations.

Konstrukce blokových algoritmů

- Mohou být použity ve všech režimech (ECB, CBC, …)
- Softwarová implementace snazší než u proudových šifer.
- Pro šifrování a dešifrování zprávy
  - můžeme definovat inverzní boxy ke každému S & P-boxu, ale tím zdvojnásobíme softwarové/hardwarové požadavky, nebo
  - definujeme snadno invertovatelné (reversibilní) struktury tak, abychom mohli použít stejný kód nebo hardware pro šifrování i dešifrování.
- Je lepší použít otestované a prokázané konstrukce algoritmů.
- První algoritmus Lucifer - 1973

Kryptografie a počítačová bezpečnost

AES

AES - Rijndael (1)

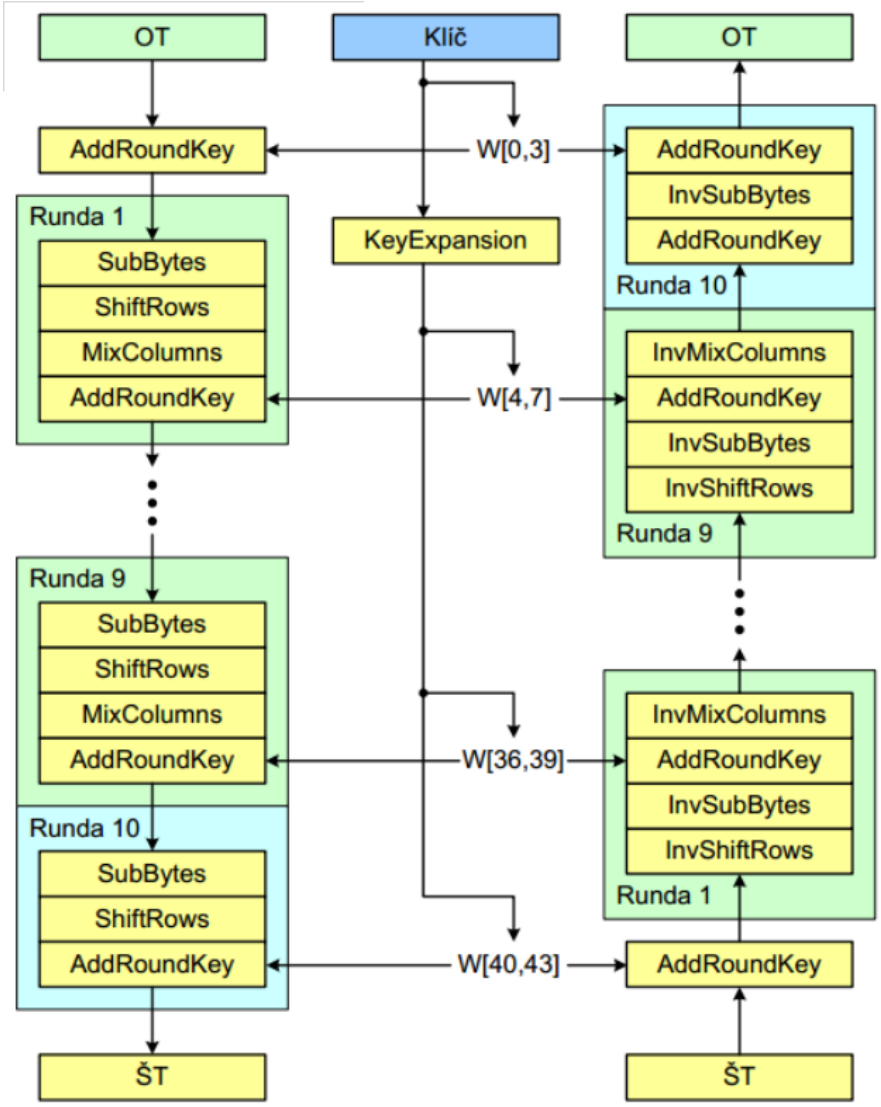
- Soutěž o nový symetrický standard vyhlášena v lednu 1997, z 15 zaslaných algoritmů (k 8/1998) do finále postoupilo 5 algoritmů (9/1999):
  - Kritéria výběru: **bezpečnost, cena** (efficiency / intellectual property), f**lexibilita**
  - algortimy RC6, Twofish, MARS, Serpent, Rijndael
- vítěz Rijndael [:Rejndál:] [:Rájndol:] (autory jsou Belgičané V. Rijmen, J. Daemen)
- AES - NIST standard v FIPS PUB 197, pravděpodobně opět nejrozšířenější symetrický algoritmus, **bez licence.**
- Standard platí od 26.5.2002
- Režimy činnosti – v různých režimech (příště)
- V protokolech např. SSL/TLS, S/MIME a jinde
- Nemá strukturu Feistelovy šifry (jako např. DES, později)

AES - Rijndael (2)

- Pro AES (tedy ve standardu) je
  - délka vstupního a výstupního bloku definována jako 128 bitů, ale algoritmus podporuje i větší bloky.
  - Délka klíče je volitelná - 128, 192, 256 bitů, což je N_k 32-bitových slov, kde N_k= 4, 6, 8.
  - Uvedeným délkám klíče odpovídá N_r = 10, 12, 14 kol (iterací, rund) algoritmu (iterovaná šifra).
- Rijndael je velmi flexibilní. Návrh je přímočarý a za základ jsou použity operace s prvky z GF(2^8) (Galoisovo těleso (někdy později), bajty).
- Příslušné operace s nimi lze provádět buď
  - tabulkově, což je výhodné pro implementaci softwarovou
  - nebo výpočtem přímo – hardwarová implementace.

AES - Rijndael (3)

- Bajtově orientovaný návrh umožňuje optimalizovat programový kód pro různé mikroprocesory
  - od procesorů na čipových kartách až po digitální signálové procesory, na programovatelných hradlových polích, na specializovaných integrovaných obvodech.
  - Obvody typu FPGA (Field Programmable Gate Array) mají z programovatelných obvodů nejobecnější strukturu a obsahují nejvíce logiky (Současné největší obvody FPGA obsahují až 6 a více milionů ekvivalentních hradel (typické dvouvstupové hradlo NAND)).
- Animace algoritmu na https://www.youtube.com/watch?v=gP4PqVGudtg (stejná animace je v Cryptoolu, viz web předmětu. Je tam také animace DESu)
- Jiný dobře vysvětlený princip AESu http://www.moserware.com/2009/09/stick-figure-guide-to-advanced.html


OT
Klíč
OT
AddRoundKey
W[0,3]
AddRoundKey
InvSubBytes
AddRoundKey
Runda 10
Runda 1
SubBytes
ShiftRows
MixColumns
AddRoundKey
KeyExpansion
W[4,7]
InvMixColumns
AddRoundKey
InvSubBytes
InvShiftRows
Runda 9
Runda 9
SubBytes
ShiftRows
MixColumns
AddRoundKey
W[36,39]
InvMixColumns
AddRoundKey
InvSubBytes
InvShiftRows
Runda 1
Runda 10
SubBytes
ShiftRows
AddRoundKey
W[40,43]
AddRoundKey
ŠT
ŠT

Další blokové algoritmy

- Lucifer, FEAL, LOKI, GOST, CAST, Blowfish, IDEA, RC5, SKIPJACK,TWOFISH, SERPENT, MARS, RC6, 3DES…..
- Charakteristika některých symetrických blokových šifer
  - Proměnná délka klíče (Blowfish, RC5, CAST…)
  - Použití více Booleovských operací
  - Proměnlivá funkce F (CAST)
  - Proměnlivý počet rund (RC5)
  - Operace s oběma polovinami zprávy (IDEA, Blowfish)
  - S-boxy závislé na klíči (Blowfish)
  - Rotace (místo S-boxů) závislá
    - na klíči (CAST)
    - na datech (RC5)

Kryptografie a počítačová bezpečnost

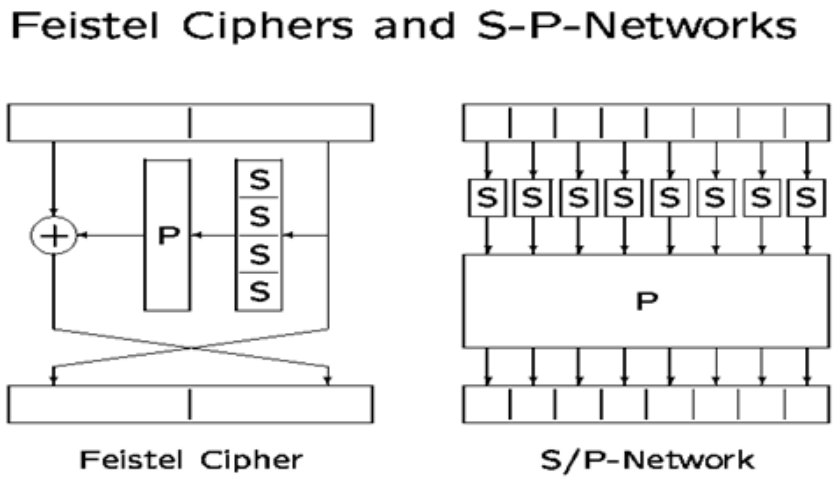
DES

Feistelova síť (1)

- Některé blokové šifry mají tzv. Feistelovu strukturu (např. DES), jiné nikoliv (např. AES)
- Použité pojmy: délka bloku, počet kroků, algoritmus generování podklíče, funkce f, iterovaná šifra
- Feistelova síť:
  - n kroků (iterací, rund), všechny jsou identické.
  - Blok zprávy m_i se dělí na dvě poloviny L_i, R_i
  - Klíč k se dělí na podklíče k_i
  - Funkce f kroku se aplikuje na R_i pomocí k_i:

    $$L_i = R_{i-1},\ R_i = L_{i-1} \oplus f(R_{i-1}, k_i).$$
  - V f se uplatňují S-boxy a P-boxy.
  - Dešifrování je totožný proces jako šifrování, podklíče se používají v opačném pořadí.

Feistelova síť (2)



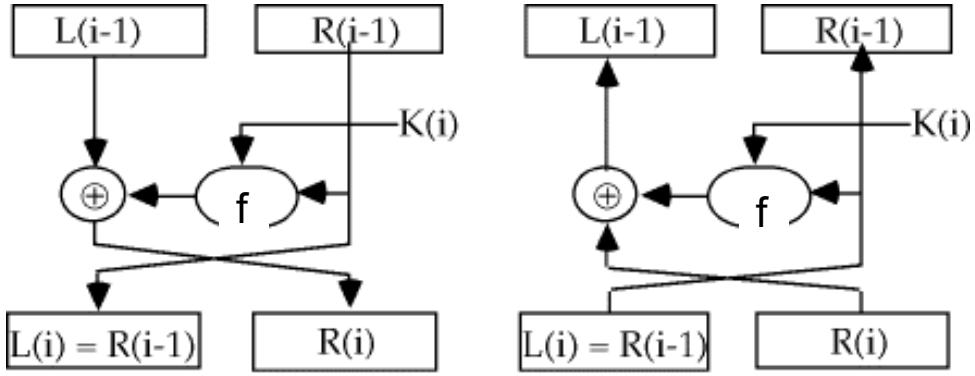


Fig 2.4 - A Round of a Feistel Cipher

DES (1)

- Standard
  - Patent 1975, Od r.1977 do (s výhradami) r.1998 standard FIPS PUB 46-2,
  - také ANSI (American National Standard) X3.92-1981/R1987.
- Režimy činnosti DESu (později):
  - FIPS PUB 81 – ECB, CBC, OFB, CFB,
  - ANSI bankovní standard – ECB, CBC pro šifrování, CBC a CFB pro autentizaci.
- DES založen na proprietárním algoritmu Lucifer (IBM, navržen Feistelem), který měl klíč 128 bitů. Klíč DESu byl zkrácen na 64 (56) bitů (požadavek NSA - implementace na jeden čip).
- 2^{56} klíčů je 72,057,594,037,927,936 klíčů
- Ze 64 bitů klíče je každý osmý paritní, proto 56-bitový klíč
- Iterovaná šifra – 16 iterací (rund)

DES (2)

- „Lavinovitost“ – ano
- Konfuse ano - korelace mezi OT a ŠT, a mezi ŠT a K – vlastnost statisticky testována → neexistuje
- Difuse – každý bit K a OT mají vliv na každý bit ŠT, tento vliv je velmi komplikovaný
  - Na tuto složitost mají největší vliv „nelineární“ S-boxy
  - Každý výstupní bit je nelineární funkcí všech vstupních bitů (XOR, AND).
  - Kdyby byly S- boxy lineární (tj. všechny výstupní bity by byly jen XOR kombinacemi vstupních bitů) → pak by všech 64 bitů bloku ŠT bylo jen lineární kombinací bitů OT a K, což by se dalo vyřešit soustavou lineárních rovnic ($M_1 \oplus M_2 \oplus \ldots \oplus C_1 \oplus C_2 \oplus \ldots = = K_1 \oplus K_2 \oplus \ldots$).

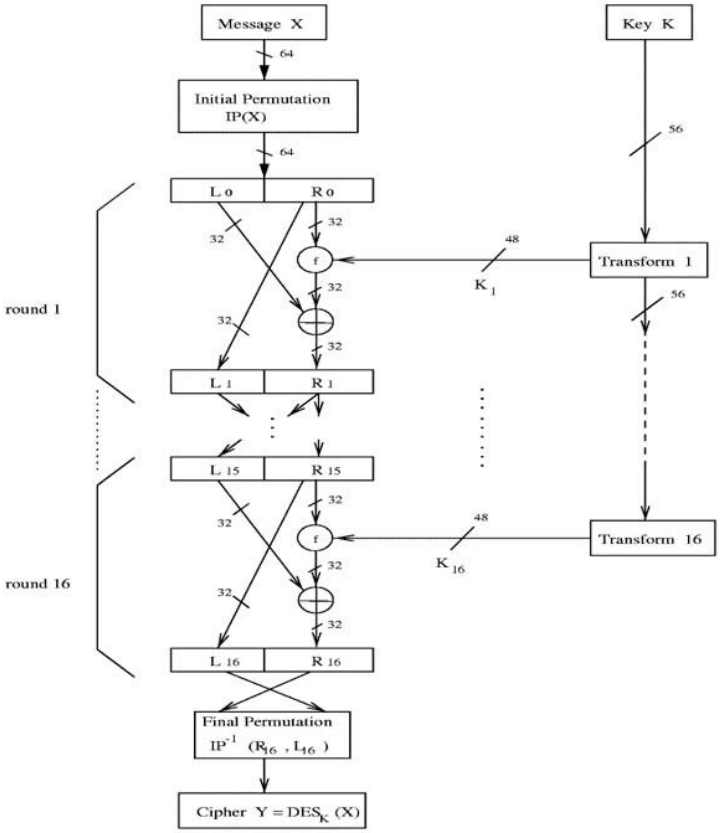
Útoky na DES

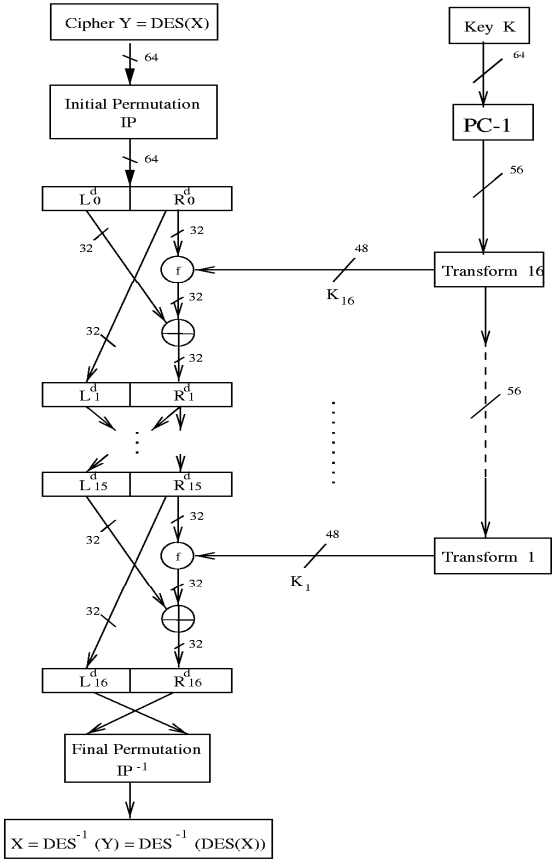
- Praktický útok – malá délka klíče
  - DES cracker (Deep Crack), 17.7.1998, HW stroj,
    - cena útoku 210 000 USD (130 000 za HW)
    - 29 desek se 64 čipy, testování **90 MLD klíčů/sec**.
    - umožňuje **bruteforce attack do 9 dní**
  - Výzva DES Challenge III,19.1.1999 za **22 hod.15 min.**, kombinace Distributed.Net (okolo 100.000 PC) a Deep Crack (http://www.emc.com/emc-plus/rsa-labs/historical/des-challenge-iii.htm)

Deep Crack

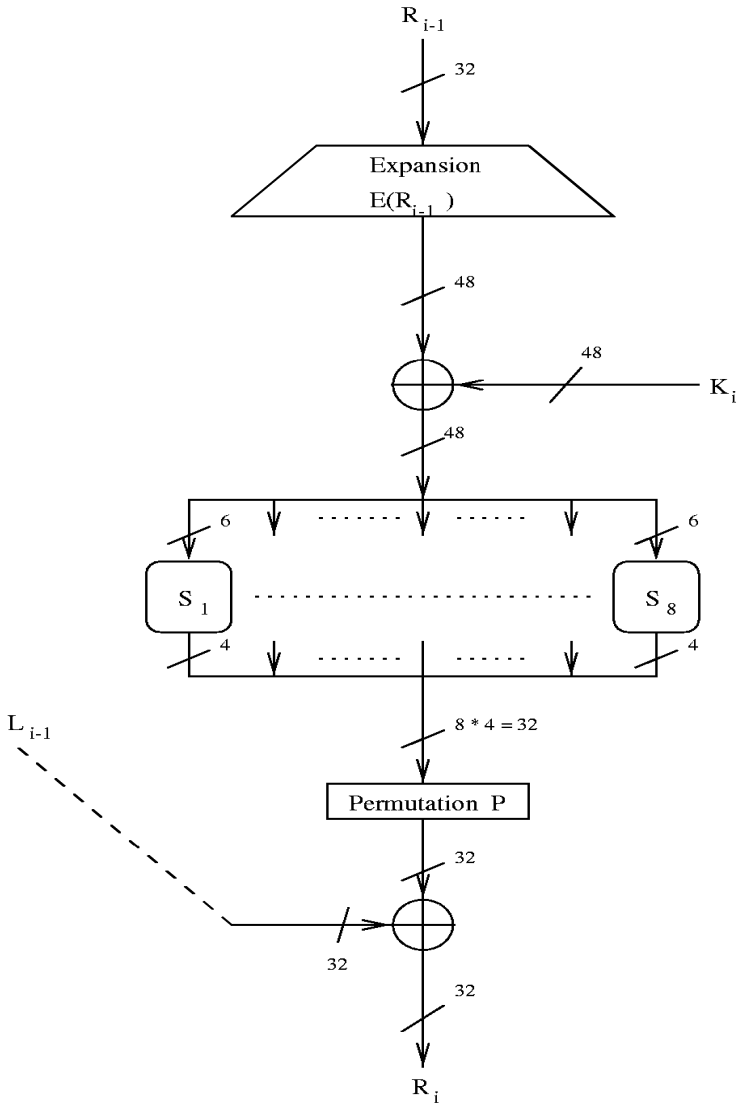
- Praktický útok – malá délka klíče
  - DES cracker (Deep Crack), 17.7.1998, HW stroj,
    - cena útoku 210 000 USD (130 000 za HW)
    - 29 desek se 64 čipy, testování 90 MLD klíčů/sec.
    - umožňuje bruteforce attack do 9 dní
  - Výzva DES Challenge III,19.1.1999 za 22 hod.15 min., kombinace Distributed.Net (okolo 100.000 PC) a Deep Crack (http://www.emc.com/emc-plus/rsa-labs/historical/des-challenge-iii.htm)

DES - šifrování a dešifrování





DES - Funkce f


R i-1
32
Expansion
E(R i-1)
48
48
K i
48
6
6
S 1
S 8
4
4
L i-1
8 * 4 = 32
Permutation P
32
32
32
R i

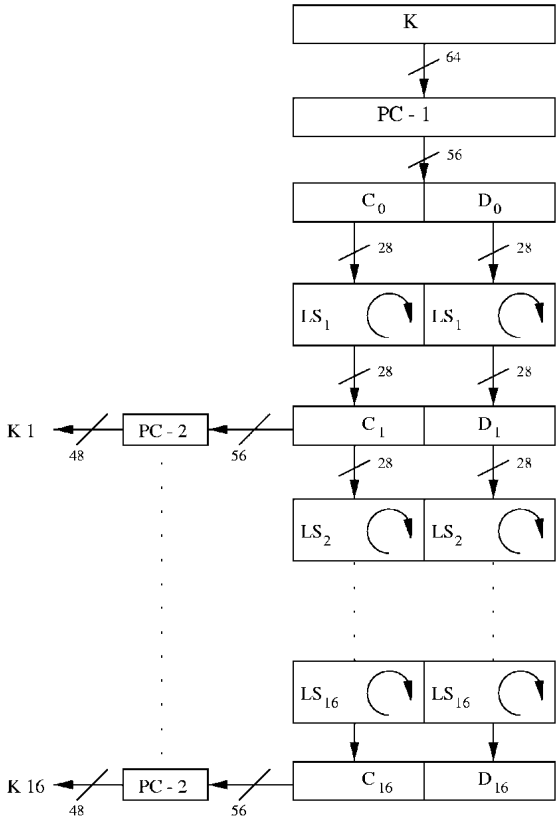
DES - Funkce f: S-box

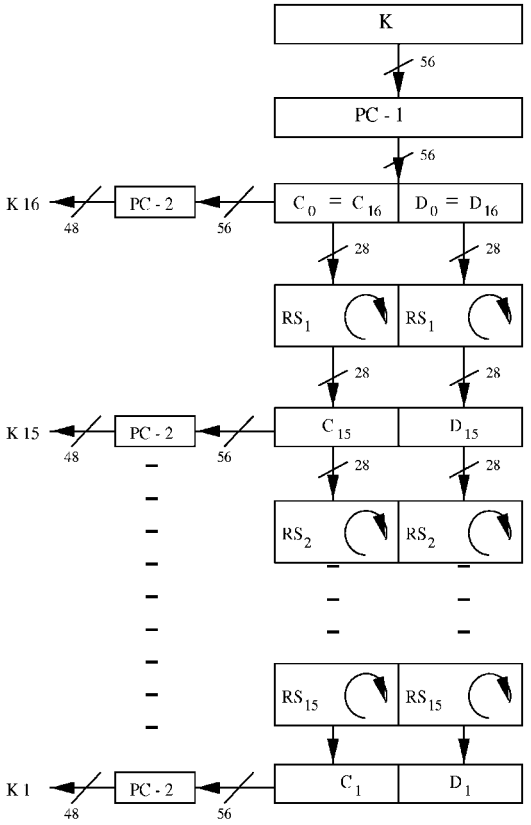
- Vstupem je 6 bitů, výstupem 4 bity např.: $S_1(100101)_2=(37)_{10}$
  - nejlevější a nejpravější bit jsou indexem řádku (indexováno od 0), tj. $(11)_2=(3)_{10}$
  - vnitřní 4 bity jsou indexem sloupce (indexováno od 0), tj. $(0010)_2=(2)_{10}$
  - $S_{ij}=S_{32}=(8)_{10}=(1000)_2$ – výstup
- Nelineární $S(a)\oplus S(b) \neq S(a\oplus b)$.

S1															
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

S-Box 1

DES - schéma generování podklíčů pro šifrování resp. dešifrování





DES - generování podklíčů

- Klíče – podklíče lze vytvářet souběžně se zpracováním otevřeného textu (tzv. on-the-fly) nebo předem.
- Ze 64 bitů klíče se používá 56 (zbývajících 8 se používají jako paritní bity (každý nejnižší bit se považuje za lichou paritu)).
- Bity jsou posouvány cyklicky o jeden nebo dva bity:
- Pro šifrování
  - LS_i=1,2,9,16 se posouvá o jeden bit doleva
  - Pro $LS_i \neq$1,2,9,16 se posouvá o dva bity doleva
- Pro dešifrování RS_i se posouvá cyklicky o 0,1,2,2,2,2,2,2,1,2,2,2,2,2,2,1 bitů doprava

DES - slabé a „poloslabé“klíče

weak key (hexadecimal)	C_0	D_0
0101 0101 0101 0101	$\{0\}^{28}$	$\{0\}^{28}$
FEFE FEFE FEFE FEFE	$\{1\}^{28}$	$\{1\}^{28}$
1F1F 1F1F 0E0E 0E0E	$\{0\}^{28}$	$\{1\}^{28}$
E0E0 E0E0 F1F1 F1F1	$\{1\}^{28}$	$\{0\}^{28}$

Four DES weak keys.

- Krátký klíč (kritika už od 70. let)
- Slabé klíče K: $E_K(M)=M$
- „Poloslabé“ klíče (K1,K2): $E_{K2}(E_{K1}(M))=M$

C_0	D_0	semi-weak key pair (hexadecimal)	C_0	D_0
$\{01\}^{14}$	$\{01\}^{14}$	01FE 01FE 01FE 01FE, FE01 FE01 FE01 FE01	$\{10\}^{14}$	$\{10\}^{14}$
$\{01\}^{14}$	$\{10\}^{14}$	1FE0 1FE0 0EF1 0EF1, E01F E01F F10E F10E	$\{10\}^{14}$	$\{01\}^{14}$
$\{01\}^{14}$	$\{0\}^{28}$	01E0 01E0 01F1 01F1, E001 E001 F101 F101	$\{10\}^{14}$	$\{0\}^{28}$
$\{01\}^{14}$	$\{1\}^{28}$	1FFE 1FFE 0EFE 0EFE, FE1F FE1F FE0E FE0E	$\{10\}^{14}$	$\{1\}^{28}$
$\{0\}^{28}$	$\{01\}^{14}$	011F 011F 010E 010E, 1F01 1F01 0E01 0E01	$\{0\}^{28}$	$\{10\}^{14}$
$\{1\}^{28}$	$\{01\}^{14}$	E0FE E0FE F1FE F1FE, FEE0 FEE0 FEF1 FEF1	$\{1\}^{28}$	$\{10\}^{14}$

Six pairs of DES semi-weak keys (one pair per line).

DES - další vlastnosti

- Komplementárnost:

  $E_K(M)=non\ (E_{non\ K}(non\ M))$

  snižuje složitost útoku hrubou silou o jeden bit.

DES - dvojité šifrování

- Dvojité šifrování: $c = e_{k2}(e_{k1}(m))$, $m = d_{k1}(d_{k2}(c))$.
- ***DoubleDES*** – klíč 2^{112} bitový
- Útok ***meet-in-the-middle*****,** known plaintext attack.
- Princip:
  - Označme $e_{k1}(m) = X = d_{k2}(c)$.
  - Mějme známou dvojici (m1, c1). Nejprve zašifrujme otevřený text m1 všemi 2^{56} hodnotami klíče k1.
  - Získané šifrové texty uložíme do tabulky, setřídíme podle hodnoty X.
  - Potom dešifrujme nám známý šifrový text c1 všemi 2^{56} hodnotami klíče k2.
  - Každou získanou hodnotu ihned hledejme v tabulce.
  - Pokud ji nalezneme, použijme příslušné klíče k1, k2 na jinou dvojici (m2, c2). Pokud získáme korektní c2, našli jsme korektní dvojici klíčů.
- Místo 2^{112} šifrování a dvou známých párů (OT,ŠT) potřebujeme k úspěchu **2 páry** (OT,ŠT), 2^{57} šifrování, dalších 2^{56} operací, 2^{56} jednotek paměti (2^{56} 64-bitových bloků, tj. 10^{17} bajtů paměti), obecně úspěch již po 2^{56} pokusech.

DES - trojité šifrování – 3DES

- Umělé zesílení DES,1999 FIPS PUB 46-3, prodloužení klíče na 56 (+ 56) + 56 bitů.
- Používá se všude tam, kde je potřeba schválený a relativně bezpečný algoritmus a nevadí zpomalení.
- Dva různé klíče ($3DES_{112}$ (také TripleDES)), Tuchman, 1978 (dnes se nedoporučuje se používat $3DES_{112}$):

  $c = e_{k1}(d_{k2}(e_{k1}(m)))$, $m = d_{k1}(e_{k2}(d_{k1}(c)))$,

- Tři různé klíče (Merklova varianta, $3DES_{168}$), DH 1977 a Merkle 1979:

  $c = e_{k3}(d_{k2}(e_{k1}(m)))$, $m = d_{k1}(e_{k2}(d_{k3}(c)))$.
- Útok na $3DES_{168}$
  - Útok hrubou silou je nereálný
  - Nejlepší útok vyžaduje:
    - 2^{32} known plaintexts, 2^{113} steps, 2^{90} single DES encryptions, and 2^{88} memory
    - This is not currently practical and NIST considers it to be appropriate through 2030.