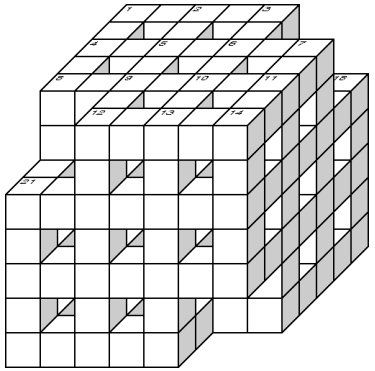
České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra telekomunikační techniky

A7B32KBE 3. přednáška

Moderní blokové šifry I

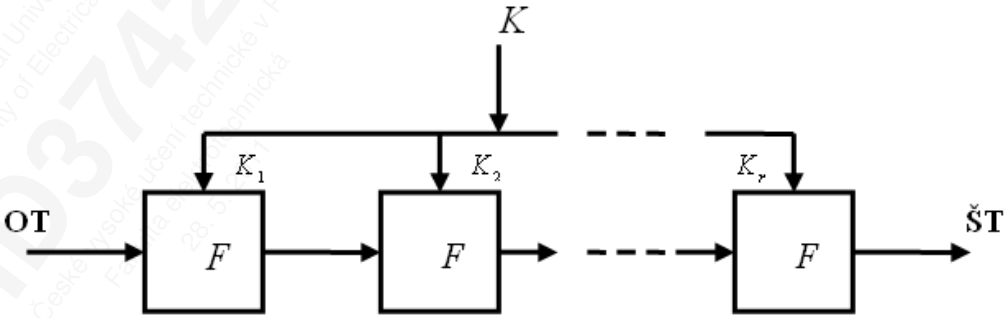


Ing. Tomáš Vaněk, Ph.D. tomas.vanek@fel.cvut.cz

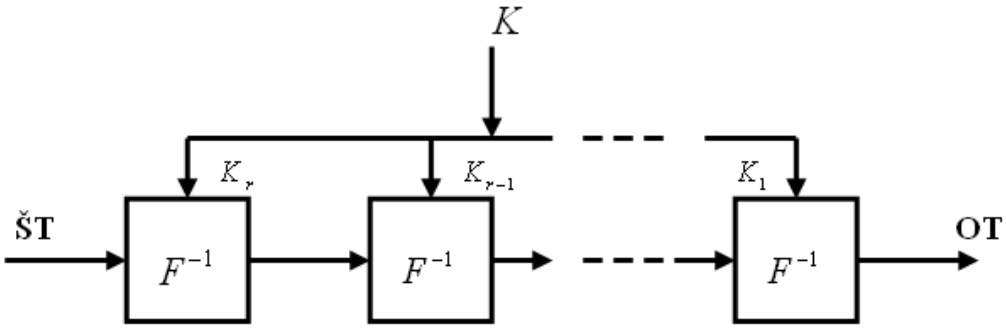
Iterované blokové šifry

- otevřený text i šifrový text mají pevnou délku
- ŠT ze získá z OT pomocí opakované **rundové funkce**
- vstupem do rundové funkce je klíč a výstup z předchozí rundy
- obvykle se implementují softwarově

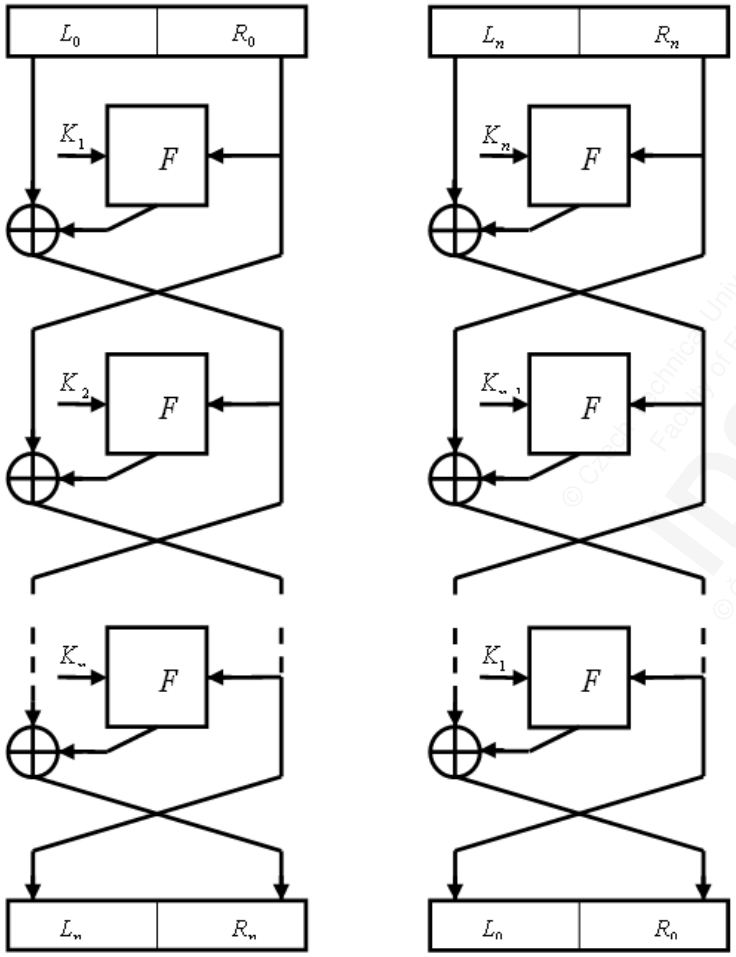
Šifrování pomocí iterované šifry



Dešifrování iterované šifry



Feistelova šifra



- Feistelova šifra představuje určitý typ blokových šifer a nikoliv konkrétní algoritmus
- Hodně moderních symetrických blokových šifer má Feistelovu strukturu.

Feistelova šifra

- Rozdělíme blok otevřeného textu na levou a pravou část:
  OT = (L_0, R_0)
- Pro každou rundu $i=1,2,\ldots,n$, spočítáme

  $L_i = R_{i-1}$

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

  kde f je rundová funkce a K_i je podklíč

  ŠT = (L_n, R_n)
- Dešifrování: ŠT = (L_n, R_n)
- V každé rundě i $i=n, n-1, \ldots, 1$, spočteme

  $R_{i-1} = L_i$

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

  kde f je rundová funkce a K_i je příslušný podklíč

  OT = (L_0, R_0)
- Tento „vzorec“ funguje pro jakoukoliv funkci $f()$
- Bezpečné šifry jsou, ale pouze pro některé funkce $f()$

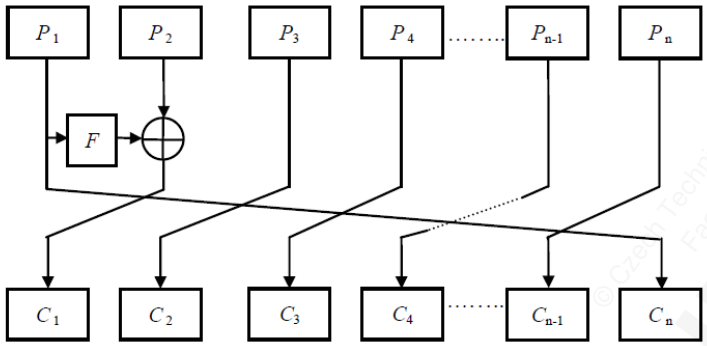
Jak fungují šifry Feistelova typu

- vstupem do algoritmu je OT a klíč K
- blok OT je rozdělen na dvě půlky: L_i a R_i
- tyto dvě půlky projdou *n* koly (rundami) v průběhu kterých se zpracovávají a vzájemně kombinují tak, že na konci vznikne blok ŠT
- í-tá runda má vstupy L_{i-1} a R_{i-1}, které jsou odvozeny z předchozího kola (rundy) a dále jedinečný rundový klíč K_i , který je generován vlastním algoritmem
- všechny rundy mají stejnou strukturu, která zahrnuje operace substituce, transpozice a aplikaci podklíče K_i

EFN - Zobecněné Feistelovy šifry

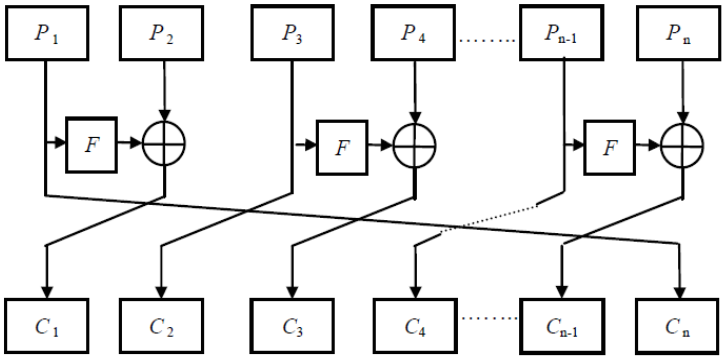
EFN – Extended Feistel Network

- blok OT se nedělí na dvě části, ale *n*



Typ I

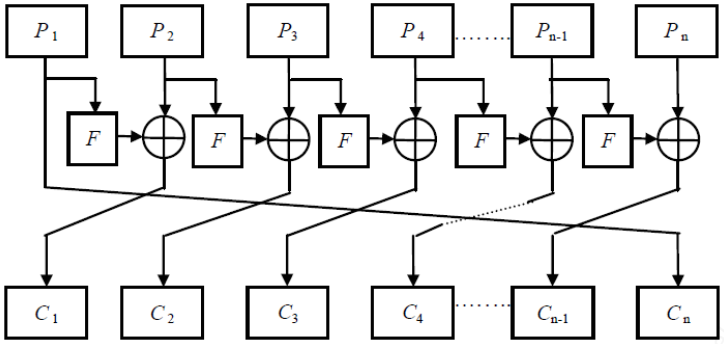
- pouze jedna F funkce
- $C_1,C_2,\ldots,C_{n-1},C_n = P_2 \oplus F(P_1),P_3,\ldots,P_n,P_1$



Typ II

- jedna *F* funkce pro každé dva sousedící bloky
- lepší difúze než Typ I
- $C_1,C_2,\ldots,C_{n-1},C_n = P_2 \oplus F(P1),P_3,P_3 \oplus F(P_4), \ldots P_n, P_{n-1} \oplus F(P_1)$

EFN - Zobecněné Feistelovy šifry



Typ III

- pro každý blok existuje vlastní *F* funkce
- difúze skoro stejně rychlá jako u Typu II
- $C_1,C_2,\ldots,C_{n-1},C_n = P_2 \oplus F(P1), P_3 \oplus F(P_4), \ldots P_n \oplus F(P_{n-1}), P_1$

Koncept EFN použit u několika kanditátů na AES, např:

- CAST-256
- MARS
- RC6

DES – základní informace

- v 60. letech IBM vyvinula šifru Lucifer
  - pod vedeném Horsta Feistela
  - délka bloku 64-bitů
  - délka klíče 128 bitů
- v roce 1973 vyhlásil úřad NBS (National Bureau of Standards) „výběrové řízení“ pro národní šifrovací standard – Data Encryption Standard (DES)
- IBM přihlásila do řízení modifikovaný Lucifer, který byl později schválen a přijat jako DES
- schválen jako norma FIPS-46 (Federal Information Processing Standards) pro šifrovaní **neklasifikovaných informací**
- DES nikdy nebyl určen pro šifrování klasifikovaných informací
- používal se až do roku 2001, kdy byl standardizován jeho nástupce AES

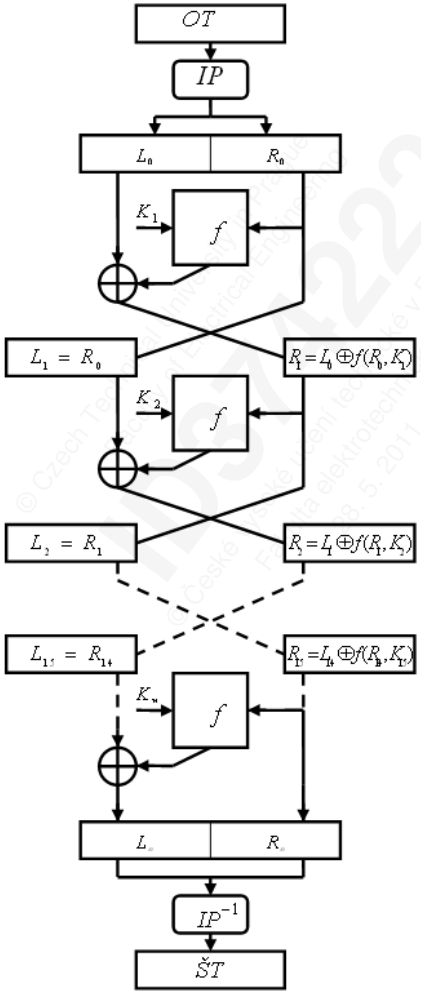
DES – základní informace

- vznik DESu byl „kontroverzní“
  - NSA ho tajně upravila + zkrátila klíč
  - úpravy nebyly dodnes oficiálně popsány
  - hovořilo se o implementaci „zadních vrátek“ do algoritmu – nepotvrzeno
  - spíše šlo o posílení vůči v té době veřejnosti neznámým kryptoanalytickým metodám - diferenciální kryptoanalýze
    - 16.2.2011potvrzeno na konferenci RSA Security Conference technickým ředitelem NSA
    - http://gcn.com/articles/2011/02/16/rsa-11-nsa--no-des-backdoor.aspx?s=gcndaily_170211

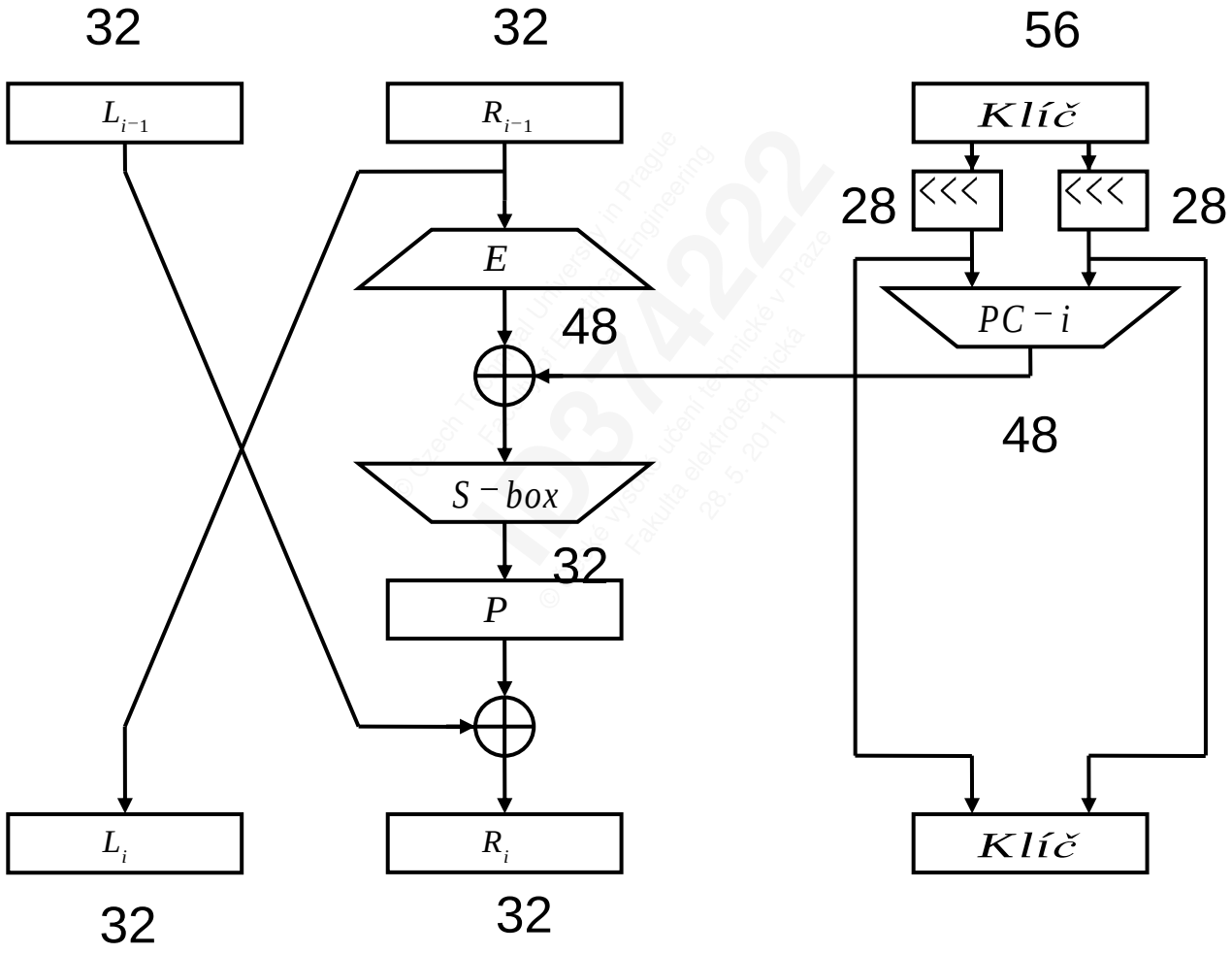
DES v číslech

- šifra Feistela typu
- délka bloku 64 bitů
- délka klíče 56 bitů
- 2^{56} = 72,057,594,037,927,900
- 16 rund
- v každé rundě je použito jiných 48 bitů klíče (podklíč, rundový klíč)
- v každé rundě se s blokem vykonávají stejné jednoduché operace
- bezpečnost DESu závisí na konstrukci “S-boxu”
- každý S-box mapuje 6 vstupních bitů na 4 bity výstupní

DES – celkový náhled



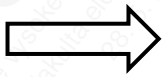
DES - operace v rundě



DES - počáteční permutace (Initial Permutation)

vstupní blok dat (64bitů)

01	02	03	04	05	06	07	08
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64



permutovaný blok dat

58	50	42	34	26	18	10	02
60	52	44	36	28	20	12	04
62	54	46	38	30	22	14	06
64	56	48	40	32	24	16	08
57	49	41	33	25	17	09	01
59	51	43	35	27	19	11	03
61	53	45	37	29	21	13	05
63	55	47	39	31	23	15	07

- účel není příliš jasný, pravděpodobně seskupení odpovídajících si bitů z různých bajtů

DES - Expanzní permutace

- **Vstup - 32 bitů**

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
```

- **Výstup - 48 bitů**

```
31  0  1  2  3  4  3  4  5  6  7  8
 7  8  9 10 11 12 11 12 13 14 15 16
15 16 17 18 19 20 19 20 21 22 23 24
23 24 25 26 27 28 27 28 29 30 31  0
```

DES S-box č.1

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
01	00	15	07	04	14	02	13	01	10	06	12	11	09	05	03	08
10	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
11	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

- 8 různých S-boxů
- 6 vstupů, 4 výstupy – nelineární operace
- první a poslední bit vstupního řetězce určují řádek
- vnitřní čtyři bity určují sloupec
- v průsečíku se nachází výsledek

Příklad: na vstupu je hodnota 100011

DES S-box č.1

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
01	00	15	07	04	14	02	13	01	10	06	12	11	09	05	03	08
10	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
11	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

- 8 různých S-boxů
- 6 vstupů, 4 výstupy – nelineární operace
- první a poslední bit vstupního řetězce určují řádek
- vnitřní čtyři bity určují sloupec
- v průsečíku se nachází výsledek

Příklad: na vstupu je hodnota 100011

zvolíme řádek 11-

DES S-box č.1

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
01	00	15	07	04	14	02	13	01	10	06	12	11	09	05	03	08
10	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
11	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

- 8 různých S-boxů
- 6 vstupů, 4 výstupy – nelineární operace
- první a poslední bit vstupního řetězce určují řádek
- vnitřní čtyři bity určují sloupec
- v průsečíku se nachází výsledek

Příklad: na vstupu je hodnota 100011

zvolíme řádek 11

zvolíme sloupec 0001

DES S-box č.1

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
01	00	15	07	04	14	02	13	01	10	06	12	11	09	05	03	08
10	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
11	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

- 8 různých S-boxů
- 6 vstupů, 4 výstupy – nelineární operace
- první a poslední bit vstupního řetězce určují řádek
- vnitřní čtyři bity určují sloupec
- v průsečíku se nachází výsledek

Příklad: na vstupu je hodnota 100011

zvolíme řádek 11

zvolíme sloupec 0001 průsečík 12 (1100 binárně)

P-box permutace

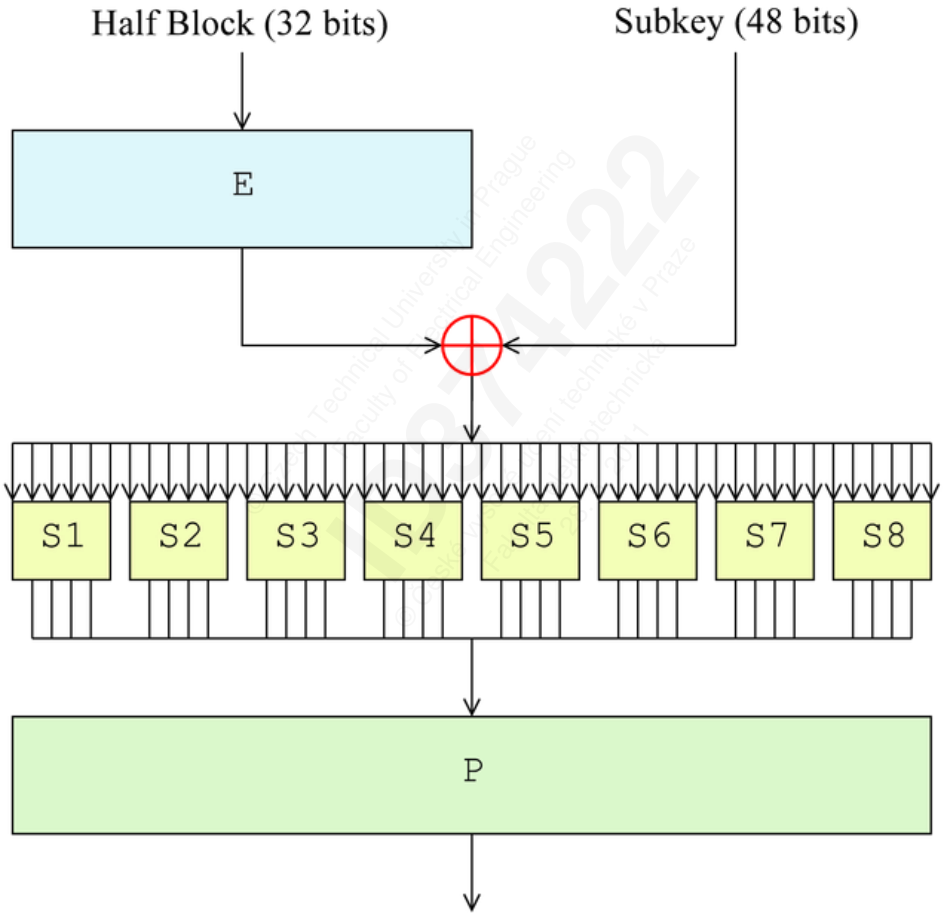
Vstup: 32bitů

01	02	03	04
05	06	07	08
09	10	11	12
13	14	15	16
17	18	19	20
21	22	23	24
25	26	27	28
29	30	31	32

Výstup: 32bitů

16	07	20	21
29	12	28	17
01	15	23	26
05	18	31	10
02	08	24	14
32	27	03	09
19	13	30	06
22	11	04	25

Operace v jedné rundě – jiné vyjádření

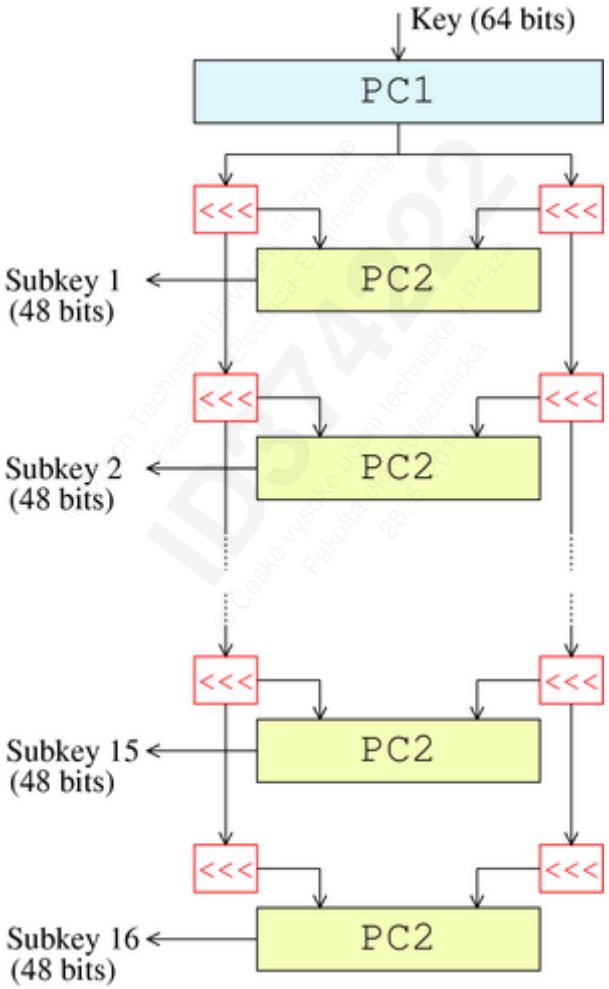


http://en.wikipedia.org/wiki/Data_Encryption_Standard

DES - generování klíčů

- Generování podklíčů, používaných v jednotlivých rundách
- skládá se z:
  - Počáteční permutace klíče (PC1), která rozdělí klíč (56-bitů) na dvě poloviny (2x28-bitů)
  - následuje 16 kroků ve kterých se :
    - z každé poloviny klíče se vybere 24-bitů
    - pomocí transpozice PC2 se zpřehází a aplikují do rundové funkce f
    - každá polovina se nezávisle na druhé zrotuje o jednu nebo dvě pozice podle plánu rotace klíče K (key schedule rotation)

DES - generování klíčů



DES – rundové klíče (podklíče)

- Klíč má 56 bitů 0,1,2,…,55
- Levá polovina klíče, `LK`

```
49 42 35 28 21 14  7
 0 50 43 36 29 22 15
 8  1 51 44 37 30 23
16  9  2 52 45 38 31
```

- Pravé polovina klíče, `RK`

```
55 48 41 34 27 20 13
 6 54 47 40 33 26 19
12  5 53 46 39 32 25
18 11  4 24 17 10  3
```

DES – rundové klíče (podklíče)

- V každé rundě `i=1,2,…,n` se vygenerují
  - `LK =` (`LK` zrotovaný doleva o r_i)
  - `RK =` (`RK` zrotovaný doleva o r_i)

  - Levá polovina podklíče K_i je 24 bitů z `LK`

    13 16 10 23 0 4 2 27 14 5 20 9
    22 18 11 3 25 7 15 6 26 19 12 1
  - Pravá polovina podklíče K_i je 24 bitů z `RK`

    12 23 2 8 18 26 1 11 22 16 4 19
    15 20 10 27 5 24 17 13 21 7 0 3

DES – rundové klíče (podklíče)

- Pro rundy 1, 2, 9 a 16 je posun $r_i = 1$, v ostatních rundách je $r_i = 2$
- Z LK jsou v každé rundě vynechány bity 8, 17, 21, 24
- Z RK jsou v každé rundě vynechány bity 6, 9, 14, 25
- **Kompresní permutace** vybírá 48 bitů podklíče K_i z celkem 56 bitů (28 bitů LK a 28 bitů RK)
- Tento proces se nazývá generování podklíčů (rundových klíčů).

Následující operace nemají žádny význam z hlediska bezpečnosti:

- počáteční permutace
- prohození polovin ŠT po poslední rundě
- konečná permutace (inverze počáteční), která se aplikuje na (R_{16}, L_{16}) a jejímž výstupem je ŠT

Modifikace DESu

- **N-násobný DES** – šifrování s klíčem o délce 56 bitů několikrát za sebou. Typickým zástupcem je TripleDES.
- **DES s nezávislými podklíči** – pro každou rundu je použit jiný naprosto nezávislý podklíč o délce 48-bitů, který není generován z 56-bitového klíče, což při 16 rundách prakticky znamená, že klíč má délku 48*16=768 bitů.
- **DES-X** – varianta DESu od společnosti RSA Data Security, která se stala součástí některých aplikací (šifrování pošty v produktu BSAFE, apod.). Tato varianta používá metodu „bílení“ (whitening), kdy ke klíči je před každou rundou je přičten mod2 (XOR) další 64 bitový klíč.
- **crypt(3)** – varianta DESu implementovaná v UNIXových systémech, používaná pro tvorbu hesel a lehké šifrování.

Modifikace DESu

- **GDES (zobecněný DES)** – zobecněný DES byl vytvořen v roce 1981 pro urychlení a posílení algoritmu klasického DESu. V roce 1990 Eli Biham a A. Shamir prokázali, že GDES je zranitelný pomocí diferenciální kryptoanalýzy a že jakákoliv varianta G-DESu, která je rychlejší než klasický DES je také méně bezpečná (než DES).

- **DES s alternativními S-boxy** – existují řešení umožňují měnit uspořádání či vnitřní řešení S-boxů.

- **RDES** – varianta, která nahrazuje výměnu levých a pravých polovin na konci každé rundy klíčově závislou výměnou, které jsou pevně určeny a závisí výhradně na klíči. Tato metoda má mnoho slabých klíčů a neměla by se používat.

Bezpečnost DESu

Slabé klíče (weak keys)

- slabý klíč: $OT=E_k(OT)$
- existují čtyři slabé klíče:

```
0x0101010101010101, 0xFEFEFEFEFEFEFEFE,
0xE0E0E0E0F1F1F1F1, 0x1F1F1F1F0E0E0E0E
```

Poloslabé klíče (semi-weak keys)

- existuje šest párů poloslabých klíčů (K_1,K_2), takových, že: $OT=E_{K2}(E_{k1}(OT))$
- generují pouze 2 různé rundové podklíče
  - `0x011F011F010E010E` a `0x1F011F010E010E01`
  - `0x01E001E001F101F1` a `0xE001E001F101F101`
  - `0x01FE01FE01FE01FE` a `0xFE01FE01FE01FE01`
  - `0x1FE01FE00EF10EF1` a `0xE01FE01FF10EF10E`
  - `0x1FFE1FFE0EFE0EFE` a `0xFE1FFE1FFE0EFE0E`
  - `0xE0FEE0FEF1FEF1FE` a `0xFEE0FEE0FEF1FEF1`

Bezpečnost DESu

Potenciálně slabé klíč (demi-semi-weak keys)

- existuje 240 potencionálně slabých klíčů
- generují pouze 4 různé rundové podklíče

Celkem existuje 2^{56} (7,21 10^{16}**) klíčů, nepoužitelných je 2^8 (**4+12+240**).**

Bezpečnost DESu

- bezpečnost DESu závisí především na S-boxech
  - vše ostatní v DESu jsou lineární operace
  - lineární = „lehce“ odstranitelné
- S-box – substituční tabulka
- ani po 30 letech nebyla nalezena žádná “zadní vrátka (back-door)”
- nejefektivnější útokem dnes je útok hrubou silou
- existuje řada variant DESu (DESX, crypt(3), GDES, RDES, s^nDES, DES s nezávislými podklíči,…) které měly být rychlejší, bezpečnější, ale většinou je jejich bezpečnost nižší…

- **Jednoznačný závěr:** návrháři DESu (resp. modifikátoři původního Luciferu) věděli co dělají

Lavinový efekt

- vhodná (a vyžadovaná) vlastnost šifrovacích algoritmů
- změna jednoho bitu v bloku vstupních dat nebo v klíči vede ke změně přibližně jedné poloviny výstupních bitů
- ztěžuje kryptoanalýzu
- DES vykazuje silný lavinový efekt

- Tvoří DES grupu ?
  - $\forall(k_1, k_2)\ \exists(k): E_k = E_{k1}\left(E_{k2}(M)\right)$
  - možnost meet-in-the middle se složitostí 2^{28} operací místo 2^{56}
  - opakované šifrování zbytečné
  - 1993 – důkaz, že netvoří ⇒3DES má smysl

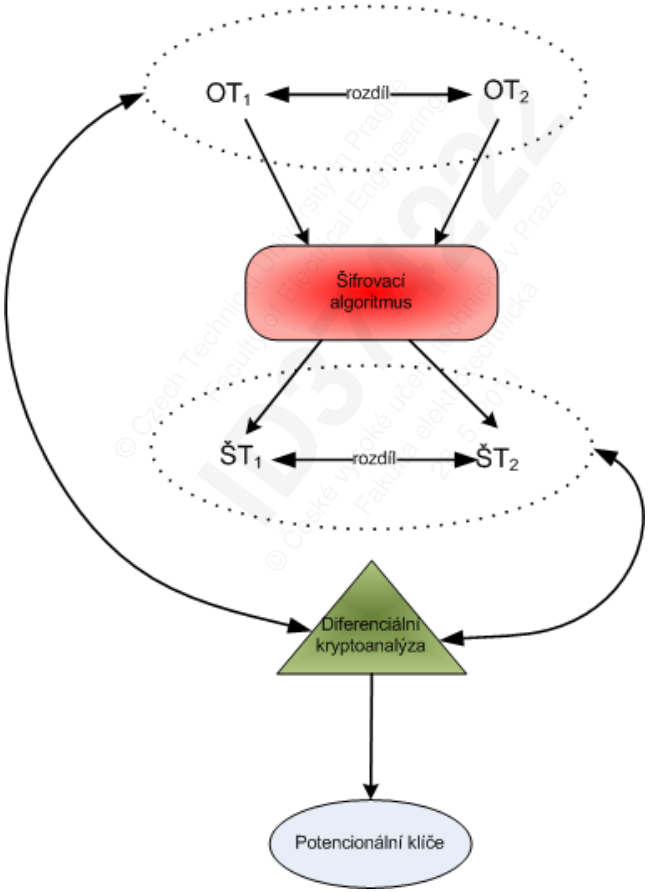
Diferenciání kryptoanalýza

- varianta útoku se znalostí vybraných OT
- jeden z nejdůležitějších pokroků (zveřejněných) v moderní kryptoanalýze
- znám NSA již v 70.letech v době návrhu DESu
- publikován v 1990 - Murphy, Biham & Shamir
- mocný nástroj pro analýzu blokových šifer
- v současnosti se používá k analýze moderních blokových šifer (s různou mírou úspěchu)
- DES je odolný vůči DK (na rozdíl od Luciferu)

Diferenciální kryptoanalýza

- statistický útok na šifry Feistelova typu
- předpoklad: máme k dispozici dvojice OT a k němu příslušný ŠT
- rozdíl mezi $d_P = P_1 \oplus P_2$, $d_C = C_1 \oplus C_2$
- vztah mezi d_C a d_P může odhalit informace o klíči
- abychom získali dobré diference d_P je potřeba mít k dispozici **mnoho párů** d_C a d_P
- D.K. umožní najít **některé** bity klíče, zbytek se získá hrubou silou

Diferenciální kryptoanalýza



Diferenciální kryptoanalýza

- Změna problému z „ Jaký klíč vygeneruje pár (OT_0, $ŠT_0$)?" na „Jaká množina klíčů může vyvolat změnu $ŠT_0$ na $ŠT_1$ změnou jednoho bitu v OT_0?“
- Útok pomocí diferenciální kryptoanalýzy na DES s 8 rundami vyžaduje:
  - 2^{14} = 16,384 vybraných OT, nebo
  - 2^{38} známých párů OT-ŠT
- Útok na DES se 16 rundami vyžaduje:
  - 2^{47} vybraných OT, nebo
  - přibližně $2^{55.1}$ známých párů OT-ŠT
- diferenciální kryptoanalýza není příliš efektivní
- **Návrháři DESu o DK 100% věděli!**

Lineární kryptoanalýza

- statistická metoda (stejně jako DK)
- analýza vnitřní struktury algoritmu
- hledáme místa, kde při vzájemném XORu bitů OT a ŠT získáme bity klíče
- S-boxy nejsou lineární, ale lze je aproximovat lineární funkcí
- k prolomení DESu potřebujeme 2^{47} známých OT
  - prakticky obtížně realizovatelné
- DES není optimalizován proti této technice

Prolomení DES

- Když byl DES standardizován, útok hrubou silou byl technicky nemožný (výkon / cena tehdejších počítačů)
- 28.1.1997 - RSA Security vypisuje soutěž „DES Challenge“
- Úkol: rozluštit zprávu zašifrovanou pomocí DESu
- Odměna: 10.000$
- Proč: důkaz nízké bezpečnosti DESu
- 18.6.2007 – první prolomení DESu hrubou silou
  - zúčastnilo se 78.000 počítačů
  - ve špičce 14.000 během 24 hodin
  - klíč nalezen za 96 dní (prohledána cca ¼ prostoru klíčů)

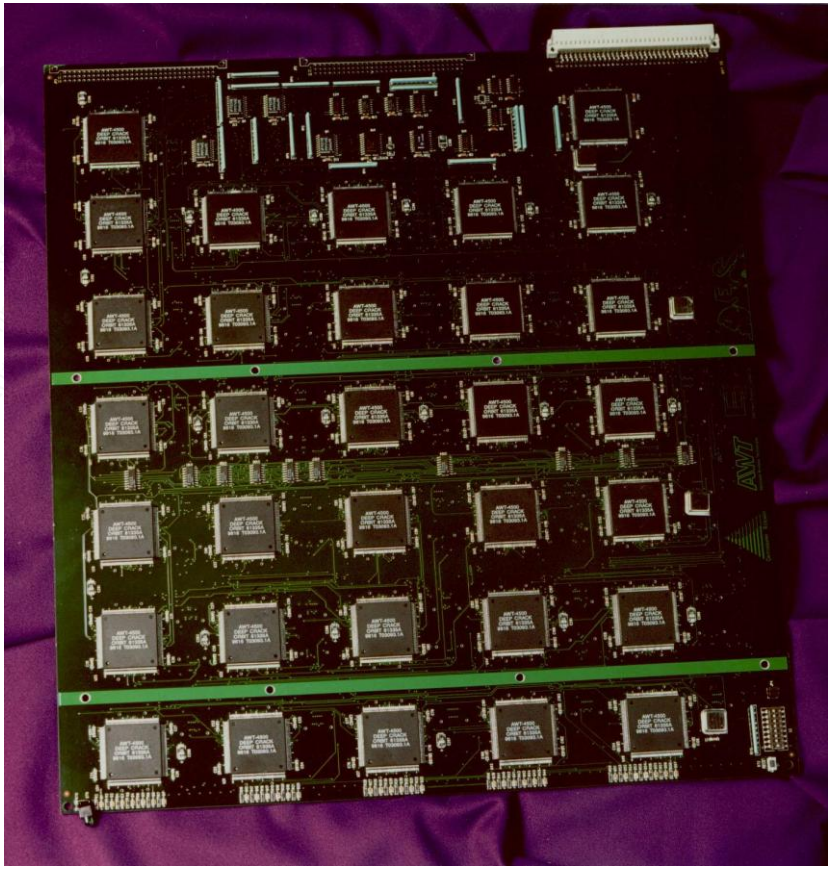
Prolomení DESu

- 1998 - **DES cracker** - projekt EFF (Electronic Frontier Foundation)
- útok hrubou silou na DES
- hrubý výpočetní výkon $9 \cdot 10^9$ klíčů/s
- cena $250.000
  - odměna za prolomení ale, pouze $10.000 !!!
- 29 desek plošných spojů
- na každé desce 64 čipů
- celkem 1856 ASIC čipů
- spočítá 1 DES klíč do 5 dnů

DES Cracker

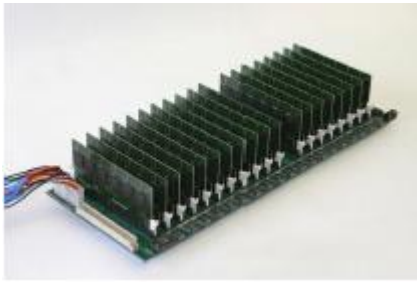
- DES Challenge II byl vyřešen po 56 hodinách pomocí stroje EFF DES Cracker
- OT: "The secret message is: It's time for those 128-, 192-, and 256-bit keys."

- DES Challenge III byl vyřešen 22 hodin a 15 minut kombinace DES Crackeru a distribuovaných výpočtů
- OT: "See you in Rome (second AES Conference, March 22-23, 1999)."



Prolomení DES

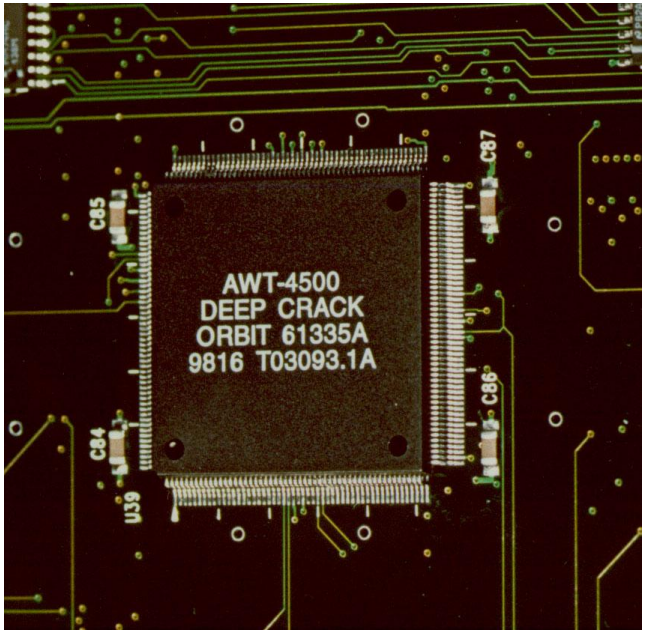
- 2006 – nový stroj založený na FPGA (Field-Programmable Gate Array)
- COPACOBANA (COst-optimized PAralleRl COdeBreaker)
- cena 10.000€
- průměrná doba nalezení klíče – 7 dní
- 2008 - COPACOBANA RIVYERA
  - zdokonalená verze
  - průměrná doba nalezení klíče – 24 hodin



DES cracker

Literatura:

- ***Cracking DES - Secrets of Encryption Research, Wiretap Politics & Chip Design* by the Electronic Frontier Foundation (ISBN 1-56592-520-3).**
- http://www.eff.org/Privacy/Crypto/Crypto_misc/DESCracker/
- http://www.copacobana.org/



Triple DES (3-DES)

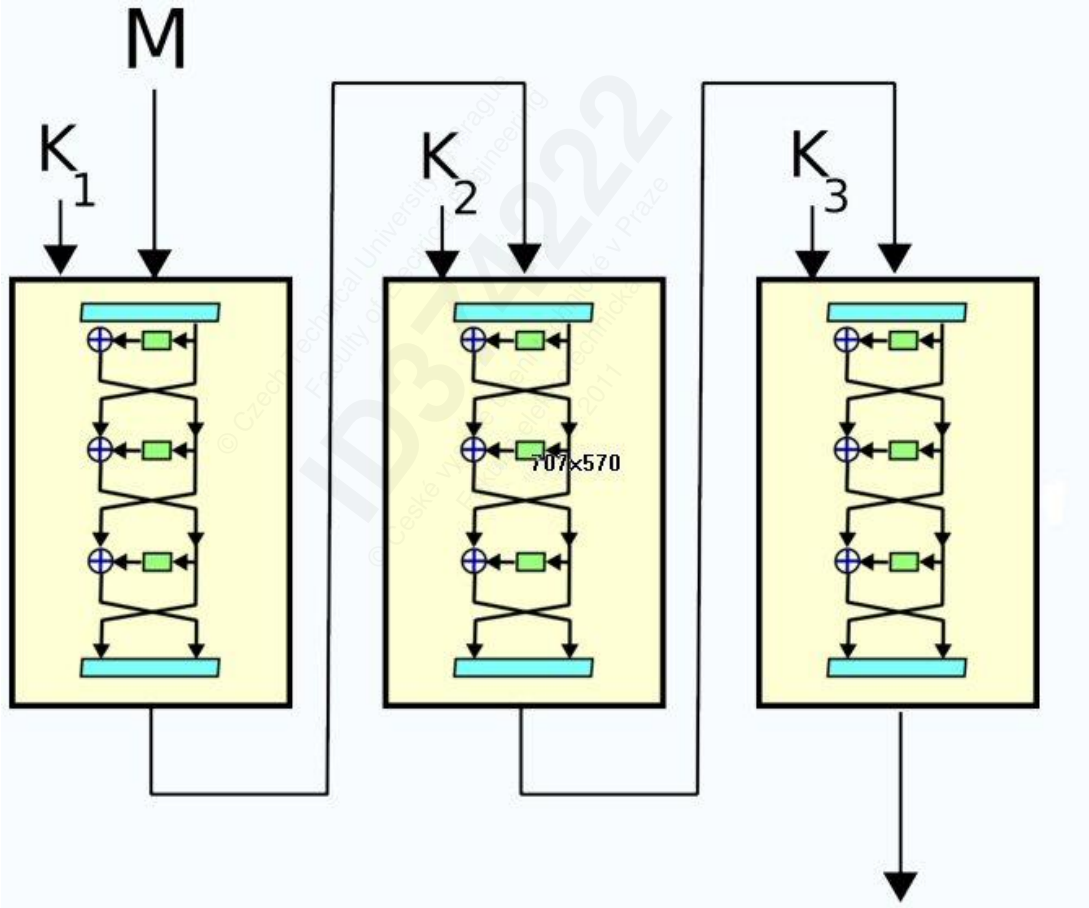
- Klíč o délce 56 bitů přestal v 90. letech postačovat
- DES byl široce rozšířen a nešlo ho rychle nahradit něčím zcela novým

Triple DES

- 3-DES-EDE2 byl standardizován v normách ANSI X9.17 a ISO 8732.

- Délka klíče 3DESu je 168 bitů (3x 56 bitů), ale díky možnosti tzv. meet-in-the-middle útoku (luštění současně z obou stran) je efektivní délka pouze 112 bitů.

Triple DES (3-DES)



Triple DES (3-DES)

3DES může pracovat v následujících režimech:

DES-EEE3	používá tři různé klíče, data jsou 3x šifrována, pokaždé jiným klíčem
DES-EDE3	používá tři různé klíče, data jsou šifrována, dešifrována a opět šifrována (opět pokaždé jiným klíčem).
DES-EEE2	používá pouze dva různé klíče. Data jsou šifrována prvním, poté šifrována druhým a opět zašifrována prvním.
DES-EDE2	používá dva různé klíče. Data jsou zašifrována prvním klíčem, dešifrována druhým a opět zašifrována prvním. Tento režim se používá nejčastěji.

Triple DES (3-DES)

- Nejčastější varianta: **Triple DES-EDE2**
  - $C = E(D(E(P,K_1),K_2),K_1)$
  - $P = D(E(D(C,K_1),K_2),K_1)$

- Proč používat 3DES-EDE se 2 klíči?
  - Zpětná kompatibilita: $E(D(E(P,K),K),K) = E(P,K)$
  - Klíč délky 112 bitů stačí

Triple DES (3-DES)

Proč se nepoužívá varianta $C = E(E(P,K),K)$?

- pořád máme jen 56 bitů klíče → prostor klíčů je stejné velký

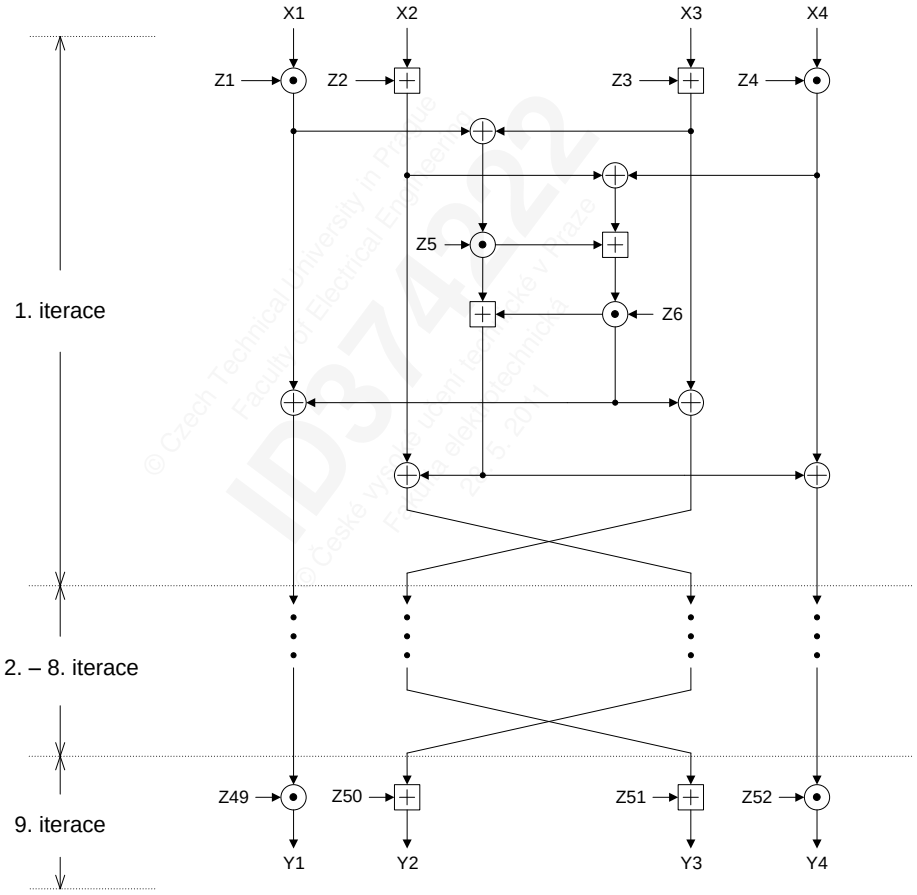
Proč se nepoužívá varianta $C = E(E(P,K_1),K_2)$?

- Existují (spíše teoretické) možnosti útokem se znalostí OT
  - předpočítáme si tabulku $E(P,K_1)$ pro všechny klíče K_1 (výsledná tabulka bude mít 2^{56} položek)
  - poté se pro každý klíč K_2 spočte $D(C,K_2)$ dokud není nalezena shoda
  - když je nalezena shoda získáme $E(P,K_1) = D(C,K_2)$
  - výsledkem hledání je znalost K_1 a K_2

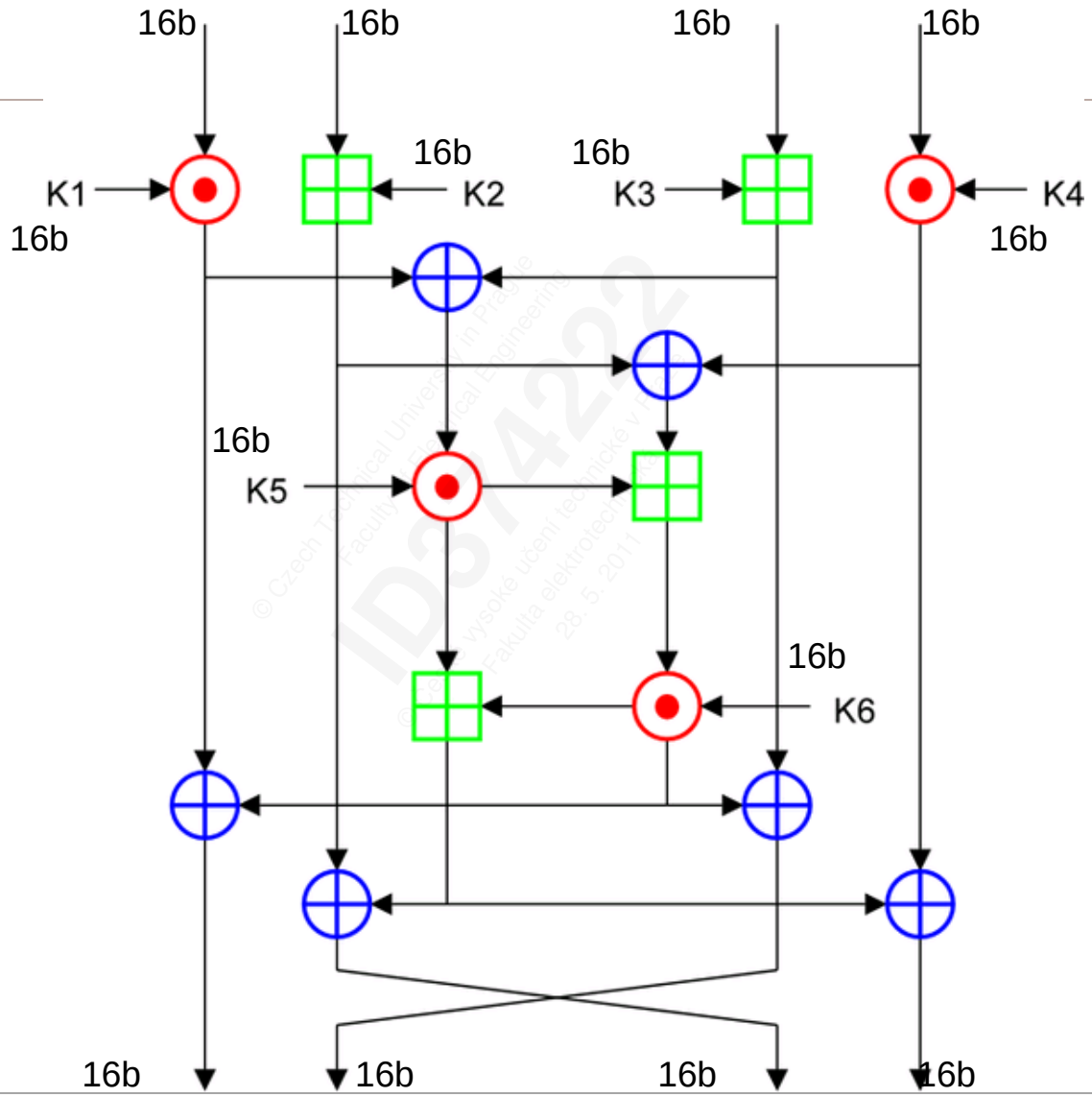
IDEA - International Data Encryption Algorithm

- délka bloku 64 bitů
- délka klíče 128 bitů
- 8,5 rund (vstupní transformace se označuje jako půlrunda)
- publikován v roce 1991
- implementován v rámci protokolu SSL nebo jako součást PGP
- patentován, pro nekomerční použití zdarma
- dvakrát rychlejší než DES, ale výrazně bezpečnější

IDEA - schéma


X1
X2
X3
X4
Z1
Z2
Z3
Z4
Z5
Z6
1. iterace
2. – 8. iterace
9. iterace
Z49
Z50
Z51
Z52
Y1
Y2
Y3
Y4

IDEA


16b
16b
16b
16b
K1
16b
16b
K2
16b
K3
K4
16b
16b
K5
16b
K6
16b
16b
16b
16b

IDEA

- vstupní blok (64 bitů) je rozdělen na čtyři části (4x16bitů)
- IDEA se skládá se z osmi identických transformací a vstupní transformace (poloviční průchod)
- Procesy šifrování a dešifrování jsou podobné
- IDEA odvozuje velkou část své bezpečnosti ze střídání operací z různých grup (modulární sčítání, násobení a XOR)

Operace, které pracují s 16bitovými řetězci, jsou:

- XOR (na obrázku znázorněno modrým $\oplus$),
- sčítání modulo 2^{16} (znázorněno zeleným $\boxplus$),
- násobení modulo $2^{16}+1$, (znázorněno červeným $\odot$).

IDEA – generování klíčů

- v každé rundě je potřeba šest unikátních 16bitových klíčů
- prvních osm rundových klíčů vznikne z původního šifrovacího klíče K rozdělením na osm 16bitových částí
- dalších osm rundových klíčů vznikne z původního šifrovacího klíče K, který je zrotován doleva o 25 a následně rozdělen na osm (opět po 16bitech)
- tento krok se opakuje tak dlouho, dokud není vygenerováno všech 52 rundových klíčů

- po osmi rundách se zbývající čtyři rundové klíče pomocí operace XOR spojí se čtyřmi 16bitovými výstupy a výsledkem je 64bitový blok ŠT

IDEA - bezpečnost

- odolná vůči diferenciální kryptoanalýze
- 2007 – úspěšný útok na omezenou verzi (6 rund)
  - úspěšný = jakýkoliv útok se složitostí < 2^{128} operací
  - složitost útoku $2^{126,8}$ a současné znalosti 2^{64} OT !
- několik slabých klíčů
  - klíče obsahující hodně nul
  - v praxi se klíče často generují náhodně → malá PRST
  - návrh řešení – XOR každého subklíče s 16bitovou konstantou
- patentové problémy
  - patentována, ale pro nekomerční účely zdarma
  - v EU vyprší 16.5.2011
  - v USA vyprší 7.1.2012

Dotazy