* v minulé přednášce jsme začali probírat substituční šifry
* 2 možná vylepšení:
 - delší jednotka pro zakódování, např. místo 1 znaku kódujeme bigramy, celá slova apod.
 - prodloužení hesla

* delší jednotka pro zakódování: slovo nebo fráze - název kódy
 - pro diplomatickou nebo obchodní korespondenci existovaly celé kódové knihy
 - rozluštění kódu jako rozluštění velké monoalfabetické šifry:
  . nutná znalost struktury věty, valence slov
  . pravděpodobně nejčastější symbol pro konec věty, lexémy spojka ”a”, tvary slovesa být atd.

* prodloužení hesla:
 - např. autokláv: klíčem šifrujeme pouze začátek zprávy, dále se heslo neopakuje ale závisí na předchozím otevřeném textu zprávy

Transpoziční šifry
....

Zatímco substituční šifry zachovávají pořadí symbolů a nahrazují je jinými symboly, transpoziční šifry mění pořadí symbolů ale symboly zachovávají.

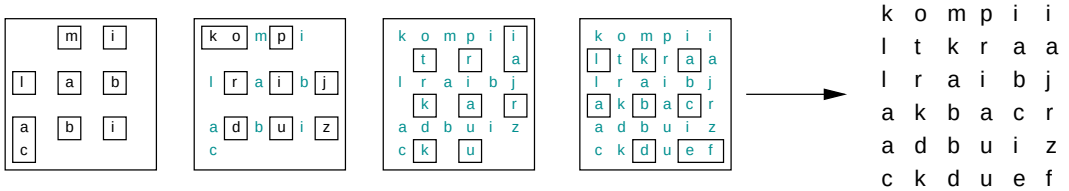
* např. sloupcová transpozice nebo šifrovací mřížka

* sloupcová transpozice:
 - klíčem je fráze; tu napíšeme do hlavičky tabulky a písmena očíslujeme podle pořadí v abecedě
 - do tabulky napíšeme otevřený text

```
  M O R S K Y P E S
  3 4 6 7 2 9 5 1 8    otevřený text: prosimprevedtemiliondolaru
  p r o s i m p r e                   namujsvycarskyucet
  v e d t e m i l i
  o n d o l a r u n     šifrový text: RIPRPOSEMLEVEIDTIMULONRDON
  a m u j s v y c a                   ACSAMYUJAVTURSEKYXC
  r s k y u c e t x
```

* co analytik
 - nejprve musí vědět, že se jedná o transpoziční šifru - dívá se zda frekvence odpovídá otevřenému textu
 - odhad počtu sloupců podle vzdálenosti pravděpodobných digramů a trigramů
  . pokud předpokládáme vyskytuje konkrétní fráze (např. ’miliondolaru’) vyhledáme znaky: RI-R-O--ML---ID-IMULONRDONA--AM-U-A--UR------
  . hledáme největší vzdálenost mezi znaky fráze
 - pak hledáme uspořádání sloupců - vyzkoušet všech k*(k-1) dvojic
  . dvojice která nejlépe odpovídá distribuci digramů považována za správnou
  . hledáme následující a předchozí sloupec - bigramy a trigramy
  . časem rozpoznatelná slova (např. miloin) - můžeme opravit
 - obdobným způsobem všechny transpoziční šifry, které transformují bloky pevné velikosti (šifrovací mřížka) - 1 ”řádek” = 1 zpráva

* šifrovací mřížka, například:



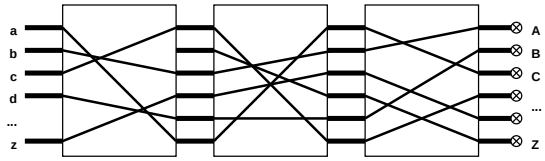
 - analýza obdobným způsobem

* možnost superšifrování - např. výstup transpoziční šifry zašifrujeme

substituční šifrou
* analýza je obtížná, nicméně možná: využívají se specifické vlastnosti konkrétních šifer

Šifrovací stroje
.....

* na elektromechanickém principu, používané za 2. sv. války
* nejznámější německá Enigma:
 - 3 nebo 4 rotory
 - rotor kontakty na obou stranách, každý rotor odlišné mapování mezi kontakty
 - po stisknutí klávesy se rozsvítila odpovídající žárovka apod.
  . posun vstupního rotoru o 1 pozici (opět 'A' - jiný výstup)
  . po dokončení rotace prvního rotoru se posune rotor 2 atd.
 - pro dešifrování opačný chod - připojení klávesnice na výstup a žárovek na vstup



Pro rozluštění byl zapotřebí known plaintext, který ale běžní administrátoři šifrovacích strojů poskytli: posílali si "testovací" zprávy (např. aaaaa) po každé změně klíče.

Pro zjednodušení dalšího výkladu zavedu jména pro komunikující strany:

* Alice chce komunikovat s Bobem tak, aby odposlech/oponent Oskar nerozuměl přenášeným zprávám
* z výkladu asi zřejmé, že všechny uvedené historické šifry dnes rozluštitelné
* vytvořit nerozluštitelnou šifru ve skutečnosti snadné

Jednorázový klíč
.....

* Vernamova šifra, jednorázový klíč (one time pad)
 - vymyslel Gilbert Vernam v r. 1917 v Bell labs při konstrukci šifrujícího dálnopisu
 - téměř perfektní řešení: náhodný klíč je stejně dlouhý jako zpráva
 - pro novou zprávu musí být použit nový klíč (proto "jednorázový")
 - šifrujeme opět Ci=Pi+Ki (mod N), dešifrujeme Pi=Ci-Ki (mod N)
 - dokazatelně bezpečné: při náhodném K jsou všechny P stejně pravděpodobní kandidáti daného C

Přestože dokazatelně bezpečná šifra, má problémy:

* musí být vygenerován dlouhý náhodný klíč - vytváření klíče nejobtížnější část (ručně - házení kostkou; elektricky - zesílení tepelného šumu na rezistoru apod.)
* obě strany musí mít kopii, tj. problém distribuce klíče (klíč si nelze zapamatovat, Alice musí klíč dopravit Bobovi nějakým bezpečným kanálem; pokud ale má bezpečný kanál, nemusí šifrovat)
* množství dat je omezeno velikostí klíče

* nestačil by pseudonáhodný klíč?
 - nestačil: pokud Oskar část otevřeného textu, může určit Ki=Ci-Pi (mod N)
 - zná-li K1, K2, ... může určit generátor pseudonáhodných čísel a jeho parametry
 - např. v mnoha systémech generátor:

$$Y_{n+1} = (a \cdot Y_n + c) \bmod d,$$

parametry a, c, d není možné vybrat libovolně jinak krátká sekvence

* nešlo by klíč použít znovu?
 - nešlo, protože by to byla opět Vigenérova šifra
 - v případě XOR dokonce Ca xor Cb = Pa xor Pb, tj. jako kdyby zpráva
   Pa byla šifrována klíčem Pb (který však má asi 75% redundanci)

* přestože může být někdy praktické (klíč na CD nebo DVD disku),
  problém při velkém množství dat
* podíváme se na moderní šifrovací algoritmy, které dokáží zpracovat
  libovolné množství otevřeného textu

Moderní symetrická kryptografie

* historická kryptografie - jednoduché algoritmy, bezpečnost závisela
  na dlouhém klíči
* moderní kryptografie - snaha aby šifrovací algoritmus odolal
  libovolnému množství vybraného otevřeného textu
* základním principem vytvoření složité šifrovací funkce složením z
  jednodušších, které poskytují vzájemně se doplňující vlastnosti

Produkční šifry

* produkční šifra kombinuje dvě nebo více transformací tak, aby
  výsledná šifra byla bezpečnější než samostatné komponenty
* základní principy např. Německo za 1. světové války
* Shannon, 1949

Nejčastější jednoduché operace:

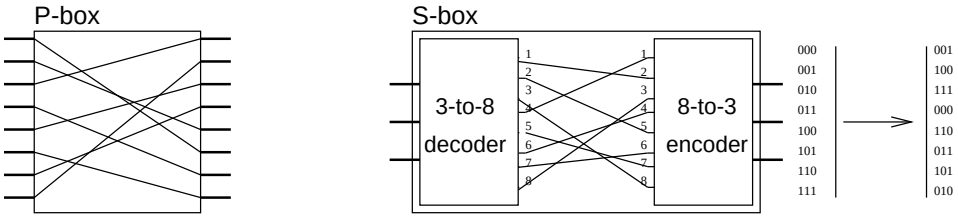
* transpozice, např. permutace bitů
* substituce, např. záměna podle tabulky
* aritmetické operace, např. modulární sčítání nebo násobení

SP-síť

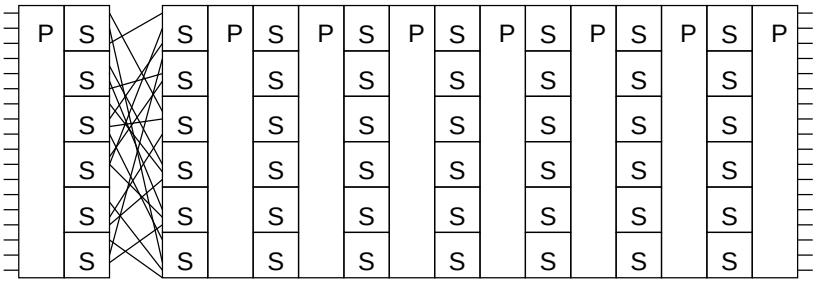
* SP-síť = substitučně-permutační síť
* Feistel, 1974 (patent na SP-síť, 2 typy S boxů, neříká konkrétní uspořádání)

* produkční šifra, složená ze substitucí a permutací

* P-box (P = permutace) provádí transpozici vstupu
* S-box (S = substituce) provádí substituci; v příkladu 3 bitový vstup i výstup
 - 3 bitový vstup aktivuje jeden z osmi výstupů dekodéru
 - aktivuje se odpovídající výstup P-boxu
 - kodér hodnotu zakóduje do 3 bitového výstupu



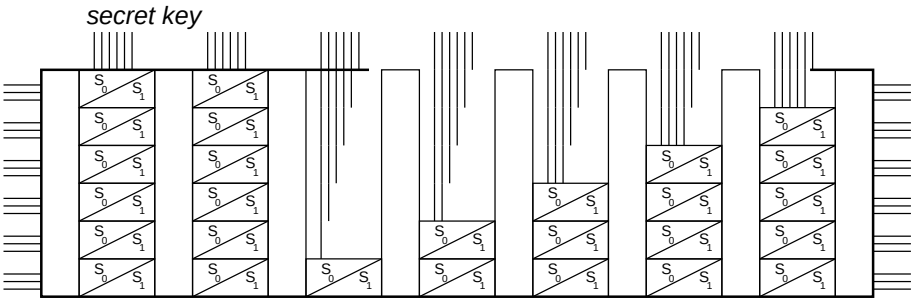
* S-boxy nemohou být příliš velké, zatímco P-boxy mohou
  (pokud n-bitový vstup S-boxu, dekodér n na 2^n)
* síla přístupu je vidět, pokud z S a P boxů sestavíme produkční šifru:



Product Cipher

* pracuje jako jeden velký S-box - ze vstupu vygeneruje příslušný
  výstup
* pro dešifrování potřebujeme SP-síť s obráceným propojením
* změna funkce by byla možná pouze úpravou propojení

* Feistelův návrh - volba ze dvou S-boxů pomocí klíče:



Secret Key Block Cipher

Feistelovská síť
.....

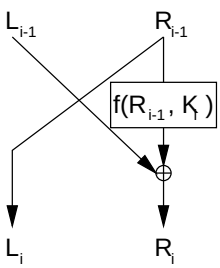
* problém předchozích návrhů - jiný obvod pro šifrování a jiný pro
  dešifrování
* řešení - tzv. feistelovská síť:

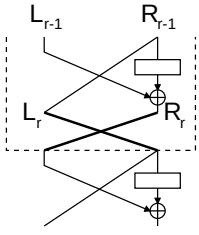
* iterovaná šifra
* otevřený text (n-bitové slovo) se rozdělí na 2 stejně dlouhé části L0 a R0
* v každé iteraci se provede

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

  kde podklíč Ki je odvozen z klíče K
* typicky >=3 iterace, často sudý počet
* po poslední iteraci ještě záměna Ln a Rn

* umožňuje dešifrování, i když fce f() není invertovatelná
* např. poslední iteraci můžeme odčinit jejím opakováním:

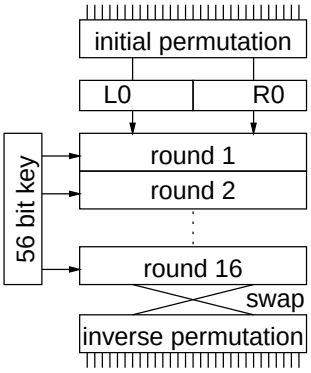


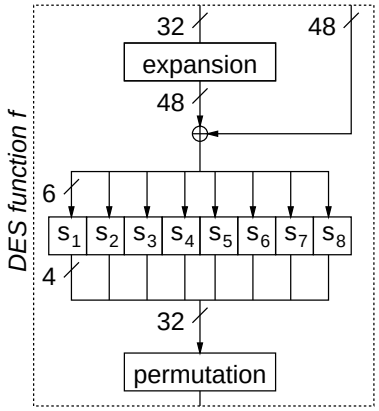


* příslušné podklíče jsou fci f() předkládány v opačném pořadí

DES
---
* začátkem 70 let si NBS (National Bureau of Standards) uvědomila, že
  je zapotřebí šifrovací algoritmus pro potřeby veřejného sektoru
* nakonec se ukázalo, že by mohl být použitelný algoritmus vytvářený IBM
* po modifikacích a ověření kvality byl v roce 1977 schválen jako Data
  Encryption Standard (DES) jako US vládní standard pro šifrování
  neklasifikovaných informací, začal se používat všeobecně (uznán ANSI,ISO)
* původně na 10 let, obnovován každých 5 let, v 1994 naposledy.

* otevřený text šifrován po blocích dlouhých 64 bitů, klíč 56 bitů
* feistelovská síť, 16 iterací, v každé iteraci 48 bitový podklíč





* funkce

$$f(R_{i-1},K_i) = P(S(E(R_{i-1}) \oplus K_i))$$

  kde
 - E pevná expanze 32 na 48 bitů (některé bity dvakrát: r32r1r2...r32r1)
 - S substituce, 8 pevných S-boxů 6 -> 4 bity
 - P pevná 32 bitová permutace
 - Ki je 48 bitový podklíč, vzniklý výběrem a permutací bitů klíče
* před první iterací pevná permutace IP (initial permutation), po
  poslední iteraci záměna L16 a R16, pak inverze IP

* dešifrování - stejný klíč a algoritmus, podklíče po použijí v opačném
  pořadí

* DES byl navržen pro snadnou implementaci v HW, existují čipy pro
  šifrování DESem (AMD 9518 apod.); SW implementace cca 1000x pomalejší

Módy DESu
----

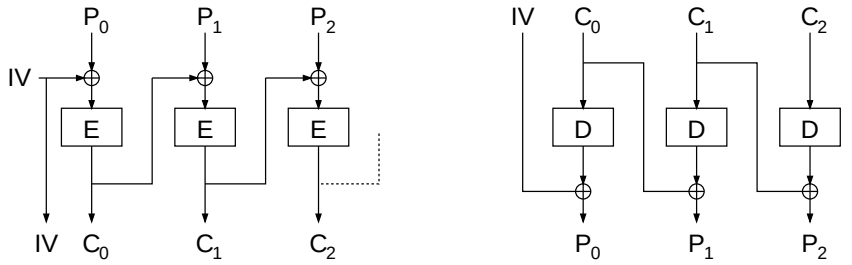
* DES se ve výsledku jeví jako monoalfabetická substituce s 64 bitovými znaky
* ukážeme jakým způsobem by se to dalo využít

* krátká délka bloku, možno vytvořit slovník šifrových textů a
  statisticky určit odpovídající otevřené texty
* představte si, že posíláte bance zašifrovaný soubor s výplatami pro
  své zaměstnance
 - rozdělíme do 64 bitových bloků, každý blok zašifrujeme
Tomas N	ovotny	2	5000 Kc
Pavel Z	loun		7000 Kc
Jiri Bo	ss	3	5000 Kc
 - Zloun ví, že bude mít malou výplatu, může to "napravit" pokud má
   přístup pouze k zašifrovanému souboru?
 - snadno: může vzít blok č. 3 a vložit na místo bloku č. 7
* tomuto způsobu se říká ECB mód (Electronic Codebook) - pro většinu
  aplikací má nevýhody
 - nedoporučuje se pro šifrování zpráv delších než 1 blok

- jeho bezpečnost lze poněkud zvýšit doplněním každého bloku náhodnou výplní

* proto pro šifrování delších zpráv byly navrženy a standardizovány
  další módy (NIST 1980): CBC, CFB a OFB, nově ještě CNT (NBS)

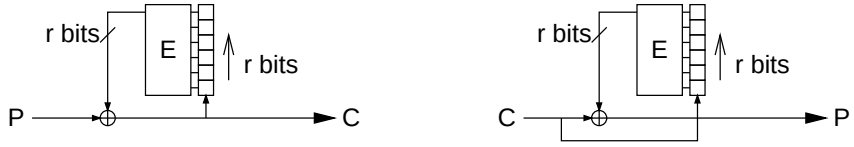
* mód CBC (Cipher-block Chaining)
 - využijeme výstup šifry pro maskování obsahu otevřeného textu:
  . před šifrováním je provedeno Pi xor předchozí C
  . stejný P už nebude dávat stejný C (nebude monoalfabetická substituce)
  . pro první blok je provedeno P xor IV, kde IV je náhodně zvolený
    inicializační vektor



CBC mode

- nevýhoda - je zapotřebí celý 64 bitový blok před tím, než můžeme
  začít šifrovat; někdy ale potřebujeme šifrovat např. po bitech nebo bytech

* mód CFB (Cipher Feedback) - šifruje po r-bitových slovech
 - máme dva čipy v šifrovacím režimu, na vstupu 64 bitový posuvný registr
 - registr na začátku naplníme IV, zašifrujeme
 - provedeme xor výstupu šifry s P0, získáme C0
 - šifrový text vstupuje do posuvného registru



CFB mode

- dešifrování - zavedeme IV do registru, zašifrujeme
- vstupem bude C0, P0 = C0 xor E(IV)
- pokud je nějaký bit Ci poškozen, je výstup poškozen dokud se chybný bit
  vyskytuje v posuvném registru; ve chvíli kdy zmizí opět OK

Poznámka (IP v DESu)

Původně vládla domněnka, že IP v DESu nemá kryptografický význam, ale v
roce 1993 bylo ukázáno, že má význam právě v CFB módu.

[]

❄