**BI-SPOL-7 Symetrické šifry blokové a proudové (AES, 3DES, RC4) základní parametry, operační módy blokových šifer (ECB, CBC, CFB, OFB, CTR, MAC), jejich základní popis a slabiny**

BI-BEZ

#### Symetrické šifry

* k šifrování a dešifrování používají stejný klíč
* nižší výpočetní náročnost než u asynchronních šifer
* Často se využívají jako hlavní prostředek pro šifrovanou komunikaci poté, co se obě strany shodnou na stejném klíči pro šifrování (předává se přes asymetrickou šifru)

## Proudové šifry

* šifrují zvlášť jednotlivé znaky abecedy
* hlavní rozdíl oproti blokovým:
  + bloky u blokových jsou šifrovány stejnou transformací Ek(Dk), kde k je šifrovací klíč
  + proudové šifry nejprve z klíče k vygenerují posloupnost h1, h2, … a každý znak otevřeného textu šifrují jinou transformací Ehi

#### Symetrické proudové šifry

* nechť A je abeceda q symbolů, M = C je množina všech konečných řetězců nad A a K je množina klíčů
* proudová šifra se skládá z transformace (generátoru) G, zobrazení E a zobrazení D
* Pro každý klíč k z K generátor G vytváří posloupnost hesla h1, h2, … přičemž prvky hi reprezentují libovolné substituce Eh1, Eh2, … nad abecedou A
* Zobrazení E a D každému klíči k z K přiřazuje transformace zašifrování Ek a odšifrování Dk
* Zašifrování OT m = m1, m2, … probíhá podle vztahu  
  c1 = Eh1(m1), c2 = Eh2(m2), …
* Dešifrování ŠT c = c1, c2, … probíhá podle vztahu m1 = Dh1(c1), m2 = Dh2(c2), kde Dhi =
* ŠT vzniká tak, že jednotlivé bity proudu hesla jsou postupně slučovány s jednotlivými bity proudu OT binárním sčítáním (používá se XOR)
* Vzhledem k rovnosti binárního sčítání a odčítání je transformace pro zašifrování a odšifrování stejná
* Příjemce i odesílatel musí mít k dispozici stejný klíč (heslo)
* Heslo je buď zcela náhodné nebo vygenerováno deterministicky nějakým náhodným šifrovacím algoritmem na základě šifrovacího klíče
  + U algoritmických proudových šifer se používá inicializační vektor (IV), aby se nemusel klíč měnit tak často – IV je přenášen v otevřené podobě

### RC4

* Jedna z nejpoužívanějších šifer na internetu, používá ji např. SSL
* Dnes považována za prolomenou
* nevyužívá IV, pro každé spojení generuje náhodně nový tajný klíč (pomocí asymetrické metody)
* volitelná délka klíče (typicky 40 nebo 128b)
* šifrovací klíč se používá pouze k vygenerování tajné substituce  
   {0, …,255} → {0, …,255}, tedy substituci bajtu za bajt
* pomocí tabulky S se pak konečným automatem generují jednotlivé bajty hesla h0, h1, …, které se XORují na OT nebo ŠT

**Proudová šifra RC4 – inicializace permutace S**

* Naplníme identickou permutací: Si = i pro i = 0, 1, 2, …, 255
* Pomocí posloupnosti bajtů klíče k (délky n) promícháme permutaci S
* Míchání provádíme postupně tak, že v každém kroku i (i é 0, 1, 2, …, 255) v permutaci S vyměníme hodnoty na pozicích podle následujícího předpisu
* Obsah obrázku text

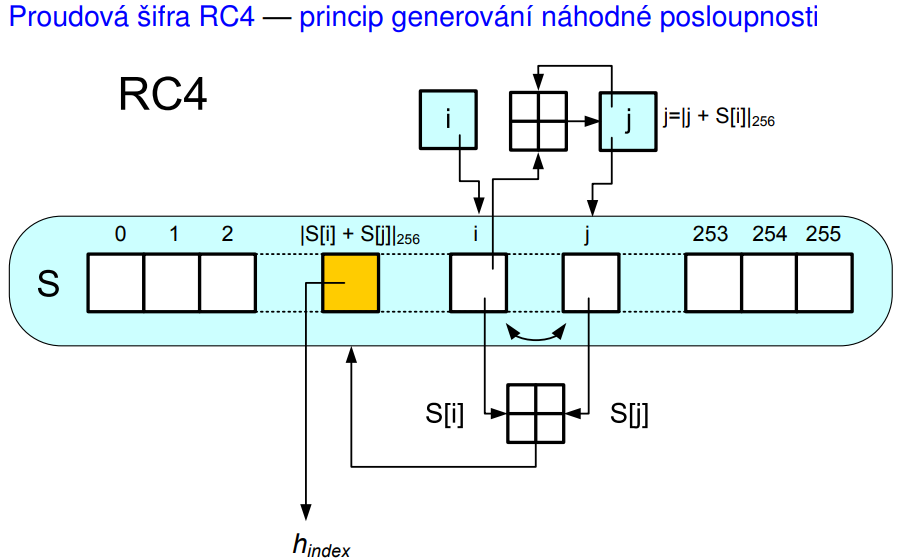
  Popis byl vytvořen automaticky

**Proudová šifra EC4 – princip tvorby hesla RC4**

* Počítání s bajty – redukce modulo 256
* i se systematicky zvyšuje modulo 256
* j je náhodný klíčově závislý index
* hodnota obsahuje heslovou posloupnost generovanou tímto algoritmem
* Obsah obrázku text

  Popis byl vytvořen automaticky

**Proudová šifra RC4 – princip generování náhodné posloupnosti**



* nejprve se podle tajného klíče vytvoří náhodná permutace, poté se začíná šifrovat OT a při každém kroku se mění stav pole (pořád swapuju S(i) a S(j))

**synchronní proudové šifry** = proud hesla nezávisí na OT ani ŠT, výpadek jednoho znaku ŠT naruší veškerý následující OT

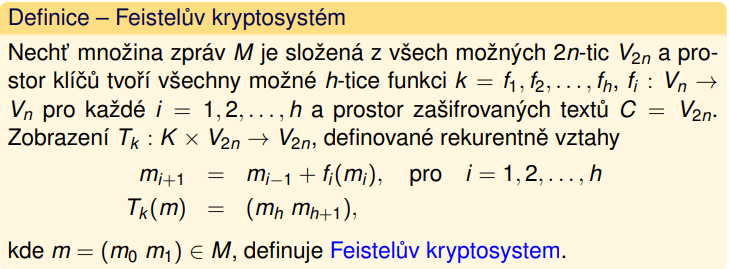
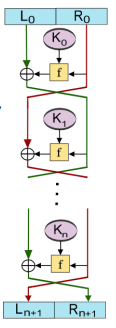
**asynchronní/samosynchronizující proudové šifry** = pokud heslo závisí na OT nebo ŠT, dokáže se samo zotavit po chybě se špatně přijatým ŠT

## Blokové symetrické šifry

* vstup je zpracováván po blocích, poslední je doplněn o padding
* bloková šifra je šifrovací systém (M, C, K, E, D), kde:
  + M: množina všech řetězců délky t
  + C: množina všech řetězců délky t
  + K: množina klíčů
  + E: zobrazení, definující pro každé k z K transformaci zašifrování Ek
  + D: zobrazení, definující pro každé k z K transformaci dešifrování Dk
* Zašifrování bloků OT m1, m2, …, kde mi je z M, probíhá podle vztahu **ci = Ek(mi)**
* Dešifrování probíhá podle vztahu **mi = Dk(ci)**
* Všechny bloky OT jsou **šifrovány toutéž transformací** a všechny bloky ŠT jsou **dešifrovány toutéž transformací**
* Šifry DES a 3DES používají blok o délce 64b
* Šifra AES používá blok o délce 128b → **lepší**
* některé využívají principu algoritmů Feistelova typu – umožňuje postupnou aplikaci relativně jednoduchých transformací vytvořit složitý kryptografický algoritmus
* lze používat v různých operačních módech
* moderní blokové šifry mají dobré vlastnosti difúze (rozptylování) a konfúze (zmatení)

### Feistelova šifra

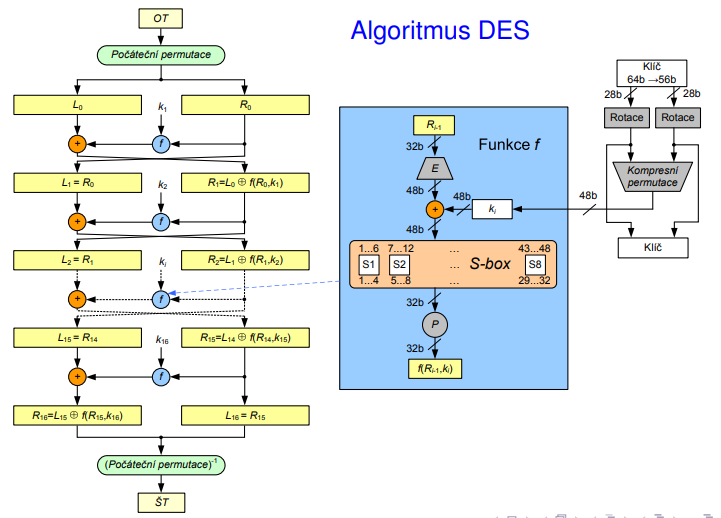
* tento princip používá DES



* Jednoduše: OT se rozdělí na poloviny L0 a R0 a poté následuje několik rund (iterace)m kde se provede:
  + L zkopíruje minulou pravou část: =
  + R XORuje minulou levou část a „upravenou“ (pomocí funkce, např. změnou permutace) pravou: = XOR f(, )
* Ki je subklíč odvozený z hlavního klíče
* jednoduše: vezmu zprávu *m*, tu rozdělím na dvě, proženu jednu část nějakou funkcí (třeba změnou permutace), sečtu (např. naXORuju) se druhou částí a vznikne mi nová část textu. Opakuju v *h* rundách.

### DES

* v praxi je jedinou zásadní nevýhodou krátký klíč
* DES je iterovaná šifra typu ((…((mi)…)))
* **Používá 16 rund a 64b bloky OT a ŠT**
* Šifrovací klíč má délku 56b (vyjadřuje se ale jako 64b číslo, kde každý 8. Bit je bit parity)
* 56b klíč je v inicializační fázi nebo za chodu algoritmu expandován na 16 rundových klíčů k1 až k16, které jsou řetězci 48 bitů, každý z těchto bitů je některým bitem původního klíče k
* Místo počátečního zašumění OT se používá bezklíčová pevná permutace Počáteční permutace a místo závěrečného zašumění permutace k ní inverzní
* Po počáteční permutaci je blok rozdělen na dvě 32b poloviny (L0, R0). Každá ze 16 rund i = 1, 2, …, 16 tansformuje (Li, Ri) na novou hodnotu   
  (, ) = (Ri, Li XOR f(Ri, ,)), liší se jen použitím jiného rundovního klíče ki
* Po 16. rundě dochází ještě k výměně pravé a levé strany: (, ) = (R15, L15 XOR f(R15, ,)) a závěrečné permutaci
* Dešifrování probíhá stejným způsobem jako zašifrování, pouze se obrátí pořadí výběru rundovních klíčů

****

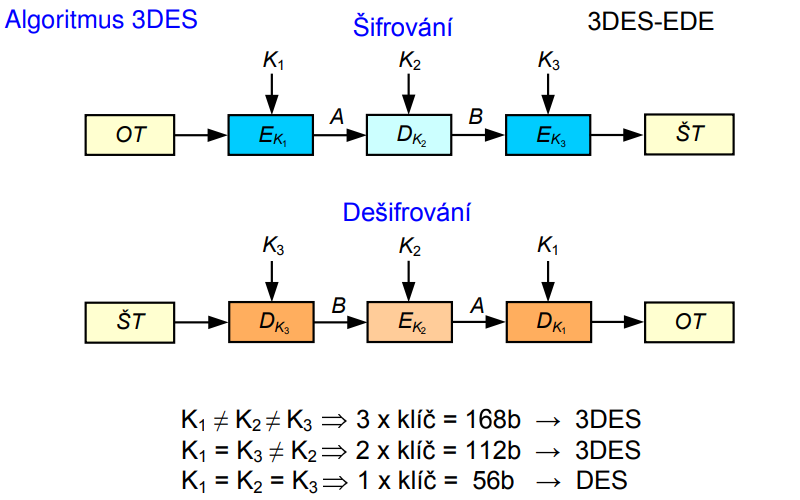
Obsah obrázku text

Popis byl vytvořen automaticky**Obsah obrázku text

Popis byl vytvořen automaticky**

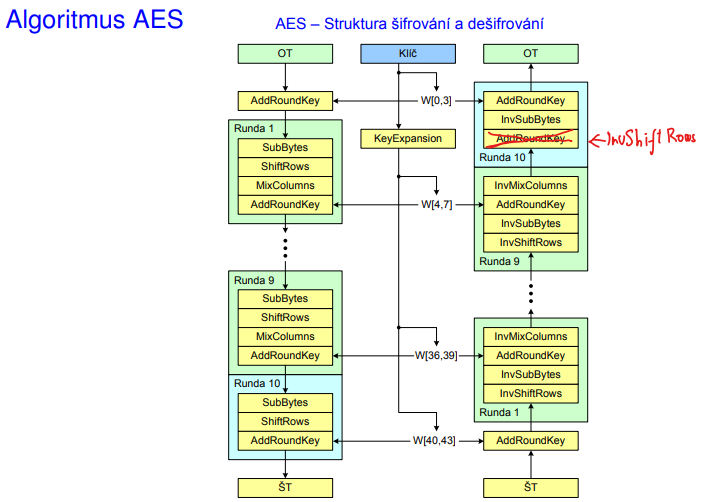
### 3DES

* Prodlužuje klíč originální DES tím, že používá DES jako stavební prvek celkem 3x s 2 nebo 3 různými klíči
* Nejčastěji se používaná varianta EDE
  + ŠT se nejdříve šifruje s pomocí 1. klíče, následně dešifruje druhým a poté šifruje třetím  
    ŠT = ( ((OT)))
  + K1, K2 a K3 jsou buď 3 různé klíče nebo K3 = K1
  + Varianta EDE byla zavedena z důvodu kompatibility
    - Když K1 = K2 = K3 tak 3DES = DES
* Klíč 3DES je velký buď 112b (2 klíče) nebo 168b (3 klíče)



### AES

* délka bloku 128b (delší než DES) – vezme 128b blok OT a zašifruje do 128b ŠT
* 3 délky klíče (128, 192, 256) a s počtem rund (10, 12, 14) – v tomto pořadí
* AES nemá slabé klíče, je odolný proti známým útokům a metodám lineární a diferenciální kryptoanalýzy
* není Feistelovského typu
* vhodný pro paralelní zpracování a má malé nároky na paměť i velikost kódu
* stále platný
* stejný proces pro OT → ŠT a ŠT → OT
* klíč K se rozdělí na klíče k1, k2, … pro jednotlivé rundy
* *SubBytes*: prohazování B v tabulce 4x4 (16B = 128b) – dáno substituční tabulkou
* *ShiftRowns*: posunutí řádků matice o 0 – 3B doleva
* *MixColumns*: prohazuje sloupce – násobíme matice
* *AddRoundKey*: na jednotlivé sloupe XORujeme rundovní klíče

****

* 4 fáze rundy (16B vstup):

*SubBytes* - substituce bytů pevnou tabulkou

*ShiftRows* - posun řádků v matici

*MixColumns* - další substituce, neprovede se v poslední rundě

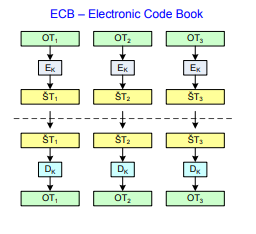
*AddRoundKey -* naxorování s rundovním klíčem

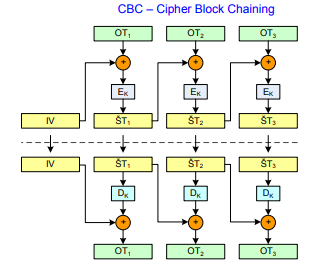
### Operační módy blokových šifer

* Operační módy jsou způsoby použití blokových šifer v daném kryptosystému, kde OT není jen 1 blok blokové šifry, ale obecně posloupnost znaků dané abecedy
* Pomocí operačních módů můžeme získat zajímavé vlastnosti a využití blokových šifer

#### ECB

* Elektronická kódová kniha, základní mód
* Každý blok se šifruje zvlášť – ŠTi = Ek(OTi)
* negativa:
  + stejné bloky OT mají stejný šifrový obraz
  + pokud nalezneme několik shodných bloků, můžeme v určitém kontextu rozkrývat hodnotu OT
  + problém s integritou – útočník může bloky ŠT vyměňovat, vkládat nebo vyjímat a docílit nežádoucích změn, zejména pokud už zná nějakou dvojici (OT, ŠT)





#### CBC (Cipher Block Chaining)

* každý blok OT se v něm nejprve modifikuje předchozím blokem ŠT, a teprve poté se zašifruje
* první blok je modifikován IV
* To zajišťuje, že běžný šifrový blok závisí na celém předchozím OT z důvodu řetězení této závislosti přes předchozí ŠT
* Nejpoužívanější operační mód
* rozšíření difúze, zřetězení mírně znesnadňuje útoky v porovnání s ECB
* Obsah obrázku text

  Popis byl vytvořen automaticky
* Náhodný IV způsobí, že budeme-li šifrovat jeden a tentýž OT 2x, obdržíme naprosto odlišný ŠT
* Je samosynchronní – proces odšifrování je schopen se zotavit a produkovat správný OT už při 2 za sebou jdoucích správných blocích ŠT

#### CFB a OFB (Cipher Feedback, Output Feedback

Obsah obrázku text

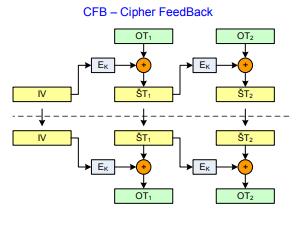
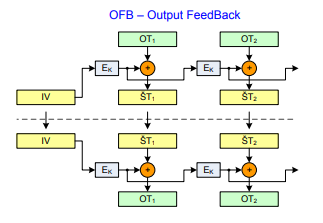
Popis byl vytvořen automaticky

Obsah obrázku text

Popis byl vytvořen automaticky

Obsah obrázku text

Popis byl vytvořen automaticky



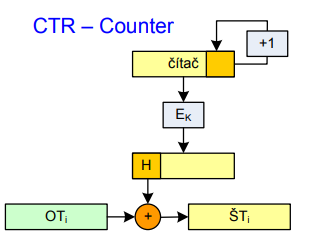
#### CTR

Obsah obrázku text

Popis byl vytvořen automaticky

Obsah obrázku text

Popis byl vytvořen automaticky



Osolený IV – vypočtený z IV a klíče, výhoda je, že se nepřevádí jako otevřený text

#### MAC

* Proudové a blokové šifry zajišťují důvěryhodnost, ale ne integritu
* MAC řeší neporušenost dat
* Autentizuje zprávy

Obsah obrázku text

Popis byl vytvořen automaticky

