Kryptografia

Režimy blokových šifier & Vybrané symetrické šifry

Katedra matematiky a teoretickej informatiky, FEI TU

AES - opakovanie

- Uveďte **základné informácie** o štandarde AES.
- Aké štyri **operácie** sa opakujú v rámci jednej rundy?
- Porovnajte **výhody a nevýhody** štandardu AES.

Režimy blokových šifier

Režimy blokových šifier - spôsoby, akými možno realizovať algoritmy blokových šifier

Rozšírený súbor režimov platný pre všetky symetrické blokové šifry:

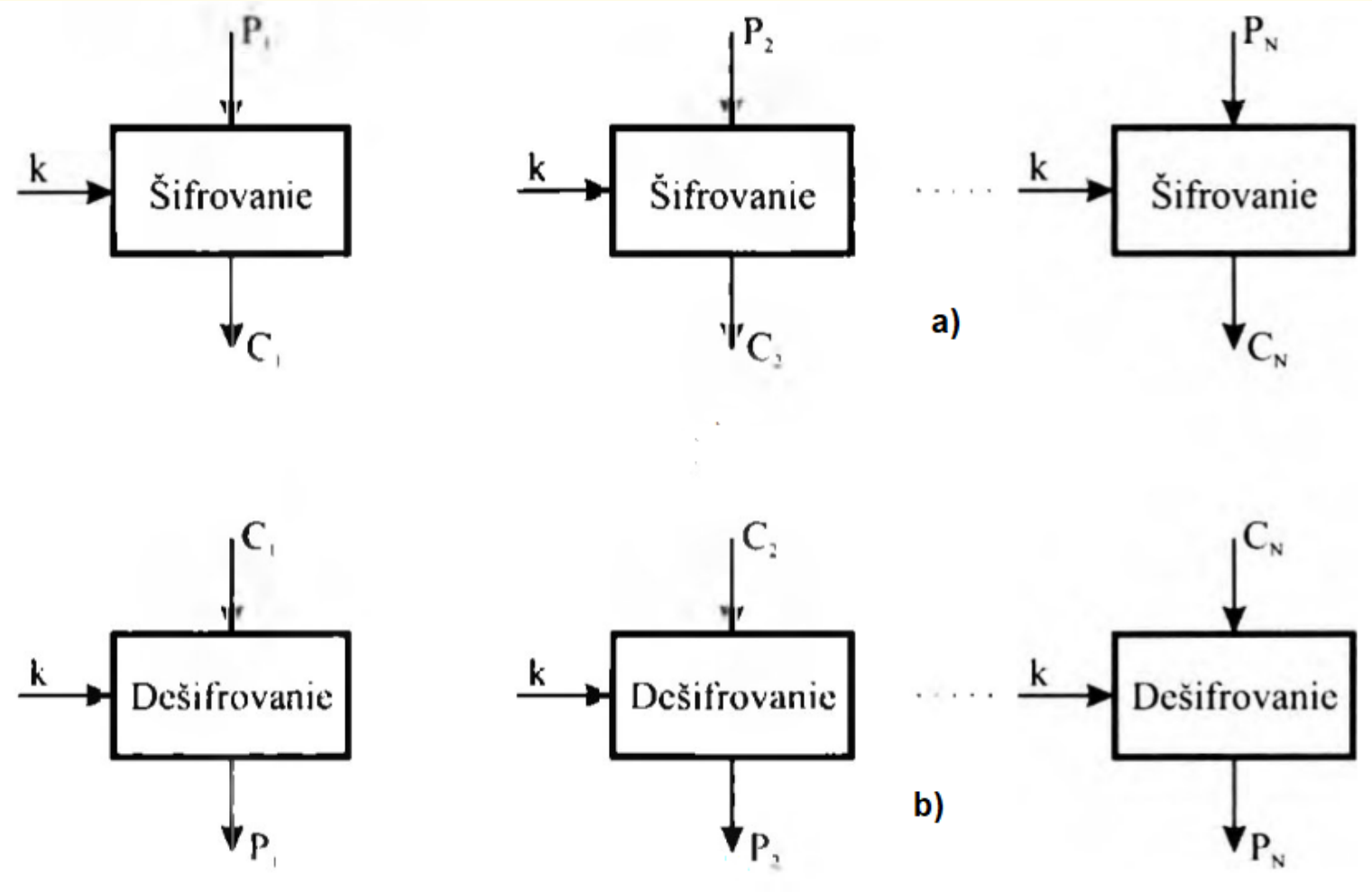
1. elektronická kódova kniha (**ECB** - *Electronic CodeBook*)
2. zreťazenie zašifrovaného textu (**CBC** - *Cipher Block Chaining*)
3. spätná väzba zo zašifrovaného textu (**CFB** - *Cipher FeedBack*)
4. spätná väzba z výstupu (**OFB** - *Output FeedBack*)
5. čítačový režim (**CTR** - *CounTeR mode*)

Elektronická kódová kniha - režim ECB

- **najjednoduchší** režim blokových šifier
- otvorený text sa rozdelí na bloky a každý blok otvoreného textu sa **priamo transformuje** na blok zašifrovaného textu
- jednotlivé bloky sa šifrujú postupne a nezávisle **tým istým kľúčom**
- analogicky prebieha aj **proces dešifrovania**
- je ideálny na **šifrovanie menšieho množstva dát** $\Rightarrow$ možno ho napr. použiť na bezpečný prenos kľúča pre algoritmus DES
- **nevýhoda** - ten istý blok otvoreného textu sa transformuje šifrovaním vždy na rovnaký blok zašifrovaného textu $\Rightarrow$ pri dlhých správach môže uľahčiť kryptoanalýzu
- **výhoda** - keďže každý blok otvoreného textu sa šifruje nezávisle, poškodenie jedného bloku pri jeho prenose neovplyvní ďalšie bloky $\Rightarrow$ režim ECB sa vyznačuje obmedzeným šírením chyby a vysokou odolnosťou voči poruchám

Elektronická kódová kniha

Princíp režimu ECB: a) šifrovanie, b) dešifrovanie:



Zreťazenie zašifrovaného textu - režim CBC

- využíva **spätnú väzbu**, čím sa dosiahne, že výsledok šifrovania aktuálneho bloku otvoreného textu závisí aj od bloku zašifrovaného textu v predošlom kroku $\Rightarrow$ šifrovaním toho istého bloku otvoreného textu možno získať rôzne bloky zašifrovaného textu
- pri **šifrovaní** sa blok otvoreného textu najprv podrobí operácii XOR s predošlým blokom zašifrovaného textu a tento výsledok sa potom zašifruje kľúčom k:

$$\mathbf{C_i} = \mathbf{E_k}(\mathbf{P_i} \oplus \mathbf{C_{i-1}})$$

kde C_i - blok zašifrovaného textu, P_i - blok otvoreného textu, E_k - bloková šifra s tajným kľúčom k, $i = 1, 2, ...$

- proces **dešifrovania** možno vyjadriť v tvare:

$$\mathbf{P_i} = \mathbf{C_{i-1}} \oplus \mathbf{D_k}(\mathbf{C_i})$$

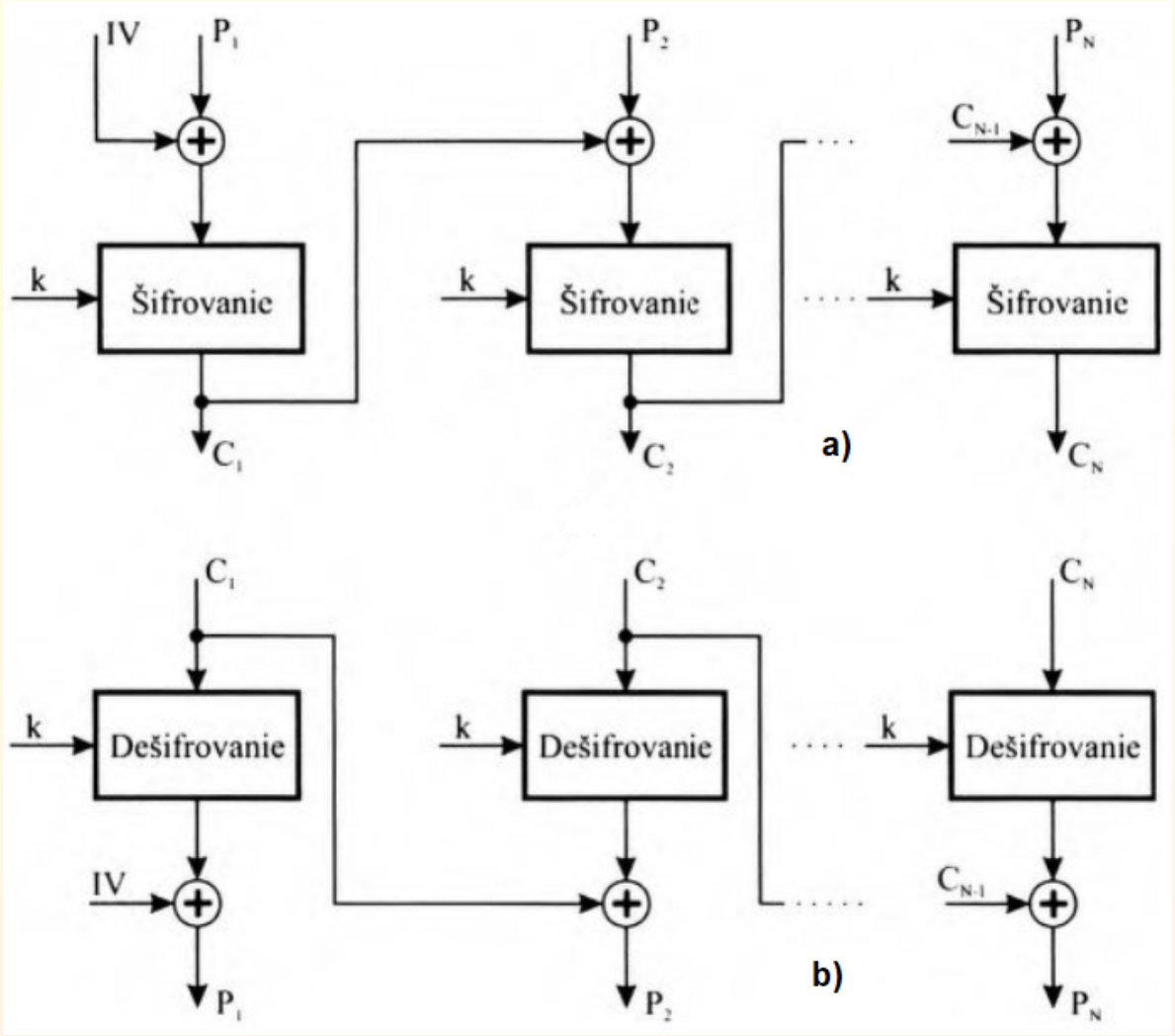
kde D_k - dešifrovanie blokovej šifry s tajným kľúčom k

Zreťazenie zašifrovaného textu

- pri šifrovaní prvého bloku otvoreného textu P_i ešte nie je vytvorený blok zašifrovaného textu $C_0 \Rightarrow$ definuje sa tzv. **inicializačný vektor $\mathbf{IV} = \mathbf{C_0}$**
- režim CBC možno charakterizovať ako **režim so spätnou väzbou** zašifrovaného textu pri šifrovaní a **doprednou väzbou** zašifrovaného textu pri dešifrovaní
- analýza šírenia chýb:
  1. **chyba v jednom bite (jednoduchá chyba) v bloku otvoreného textu** ovplyvní zodpovedajúci blok zašifrovaného textu a všetky nasledujúce bloky zašifrovaného textu $\Rightarrow$ pri dešifrovaní však v dôsledku reverzného efektu sa táto chyba objaví iba v príslušnom dešifrovanom bloku otvoreného textu
  2. **chyba v jednom bloku zašifrovaného textu** (v dôsledku prenosu zašifrovaného textu prenosovým kanálom) $\Rightarrow$ chybne je dešifrovaný zodpovedajúci blok otvoreného textu a ďalší blok dešifrovaného otvoreného bloku textu, t.j. chyba ovplyvní dva susedné bloky otvoreného textu

Zreťazenie zašifrovaného textu

Princíp režimu CBC: a) šifrovanie, b) dešifrovanie:

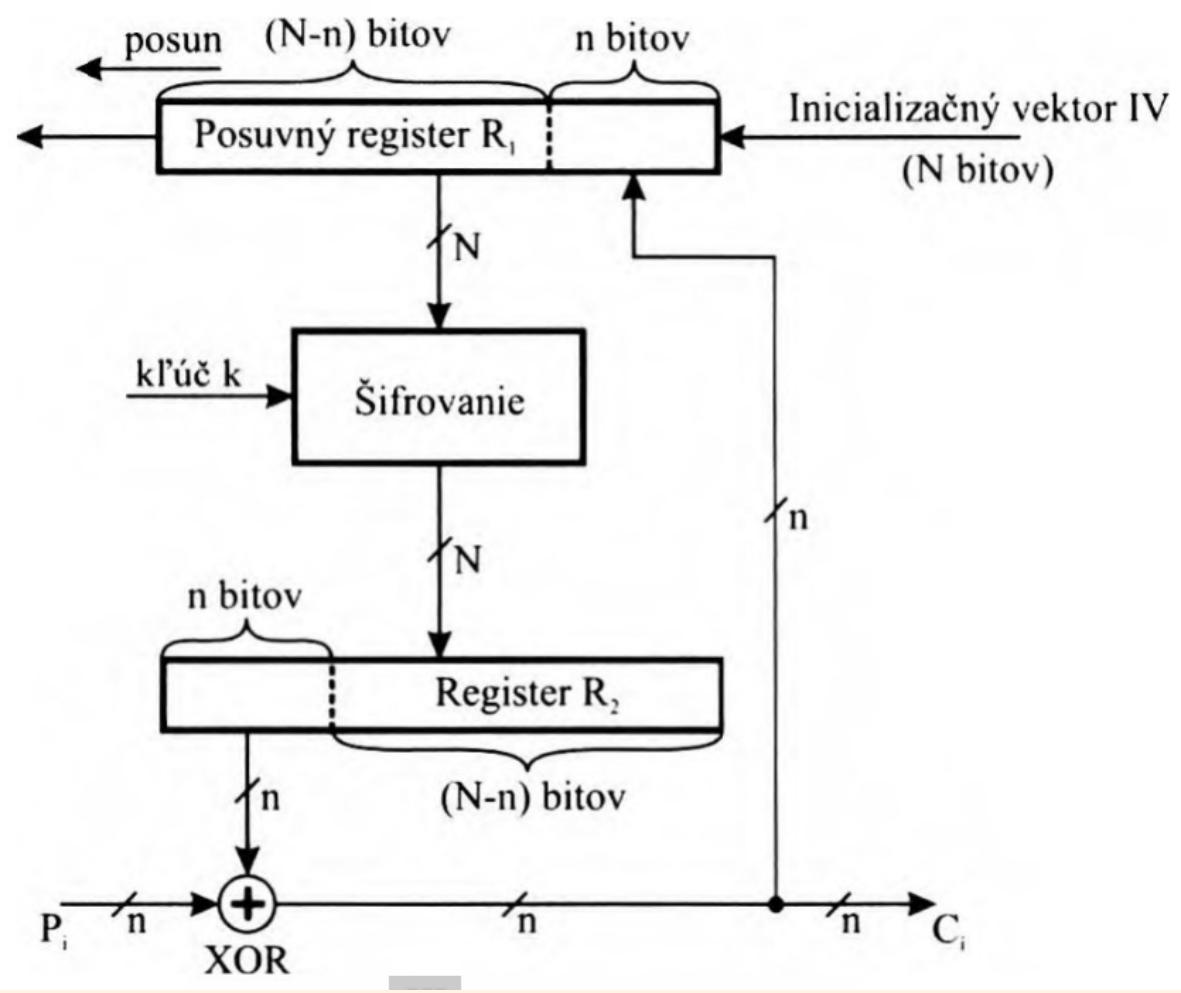


Spätná väzba zo zašifrovaného textu - režim CFB

- je určený na **šifrovanie dát v subblokoch**, ktoré sú menšie než pôvodný blok dát $\Rightarrow$ napr. blok dát o dĺžke 64 bitov sa šifruje po 8 bitoch (8-bitový režim CFB), resp. 8-bitový blok sa šifruje po jednom bite (1-bitový režim CFB)
- šifrovanie po jednom bite umožňuje realizovať **konverzie blokovej šifry na prúdovú šifru**
- **princíp režimu** - pôvodný blok otvoreného textu, ktorý má N bitov sa šifruje po subblokoch o veľkosti n bitov:
  1. v prvom kroku sa cez sériový vstup do posuvného registra R_1 vloží N-bitový inicializačný vektor IV, ktorý sa zašifruje kľúčom k a výsledok sa uloží do registra R_2
  2. z registra R_2 sa potom vyberie n najvyšších bitov, ktoré sa podrobia oprácii **XOR** s n-bitovým subblokom otvoreného textu P_1, čím sa získa n-bitový subblok zašifrovaného textu C_1
  3. potom sa posuvný register R_1 posunie o n miest doľava, do n uvoľnených bitov posuvného registra sa zapíše n-bitový subblok C_1
  4. v ďalších krokoch sa uvedený postup opakuje

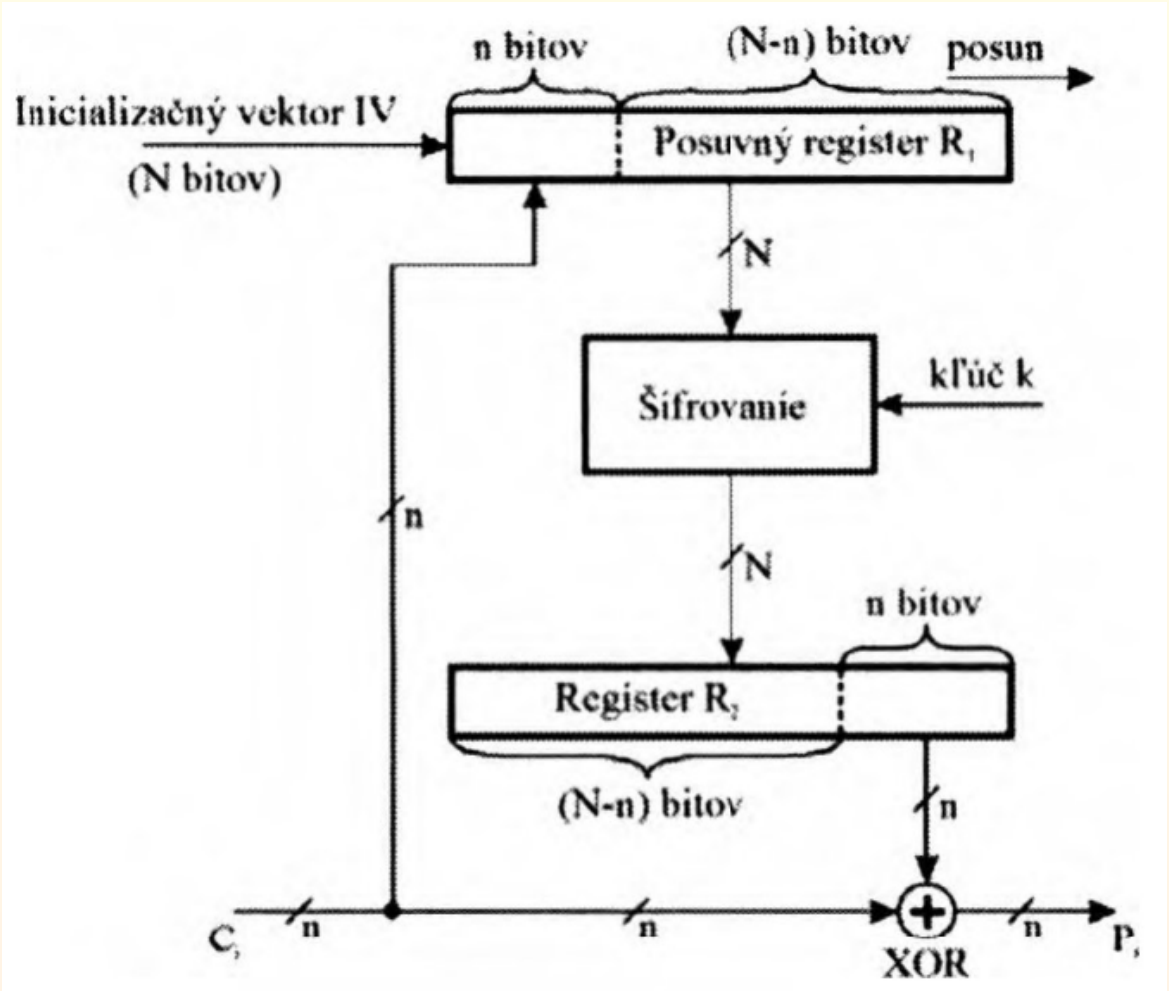
Spätná väzba zo zašifrovaného textu

Princíp režimu CFB - šifrovanie:



Spätná väzba zo zašifrovaného textu

Princíp režimu CFB - dešifrovanie:



Spätná väzba zo zašifrovaného textu

Analýza šírenia chýb:

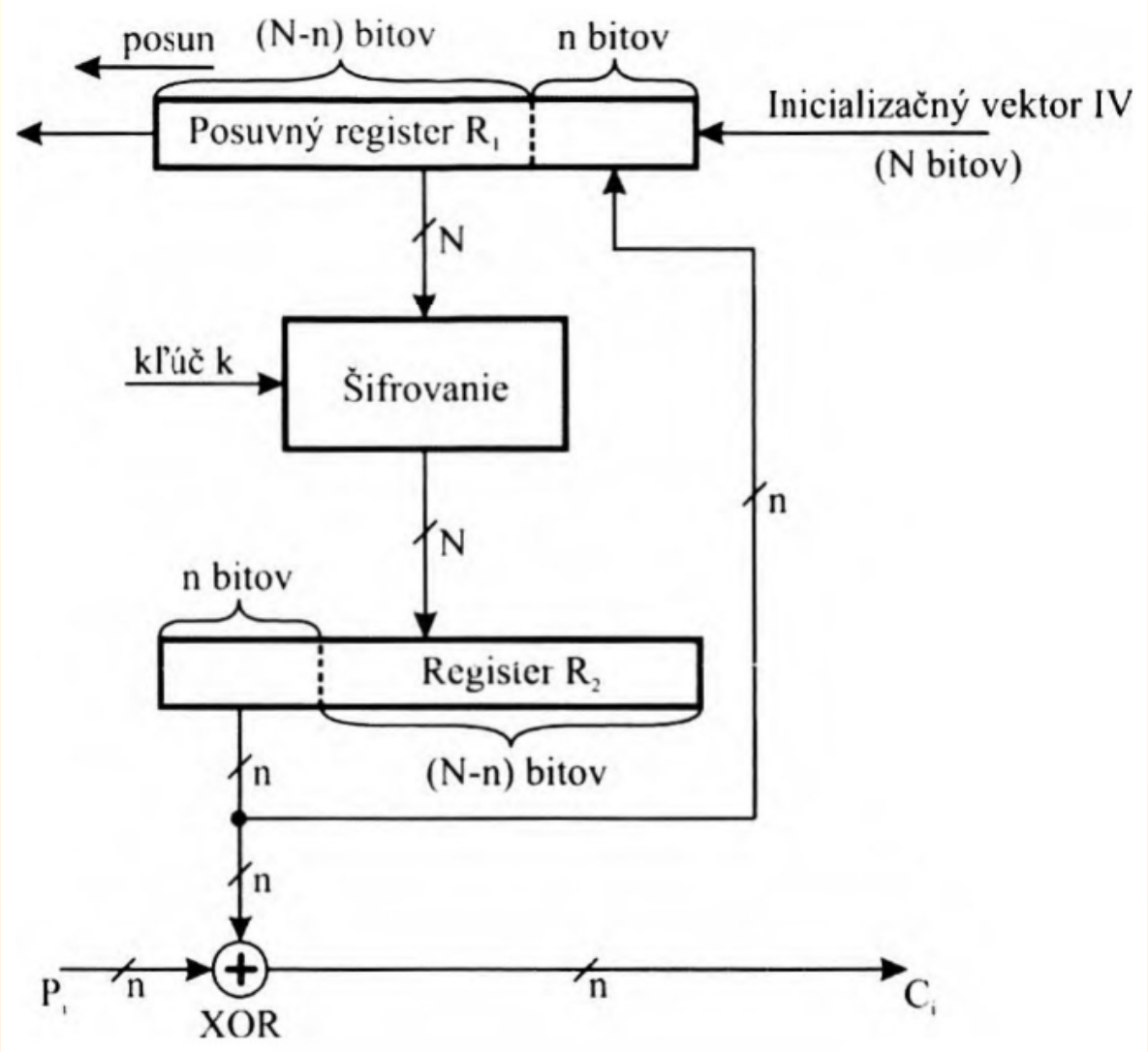
- v režime CFB aj **jednoduchá chyba** v jednom bloku otvoreného textu ovplyvní všetky nasledujúce bloky zašifrovaného textu $\Rightarrow$ **pri dešifrovaní** sa táto chyba premietne reverzne iba do príslušného bitu dešifrovaného bloku otvoreného textu
- ak nastane **chyba v prijatom bloku zašifrovaného textu**, je tento blok chybne dešifrovaný, pričom sa tento chybný blok zašifrovaného textu zapíše aj do registra R_1 a tým ovplyvní aj dešifrovanie ďalších bezchybných blokov zašifrovaného textu
- **počet nesprávne dešifrovaných blokov** závisí od parametrov n a N $\Rightarrow$ ak napr. $N = 64$ bitov a $n = 8$ bitov, potom je chybne dešifrovaných ďalších 8 bajtov, teda celkom je chybne dešifrovaných 9 bajtov

Spätná väzba z výstupu - režim OFB

- je **veľmi podobný režimu CFB - rozdiel** spočíva v tom, že do n nižších bitov posuvného registra R_1 sa subblok n bitov, ktorý sa vyberie z n-vyšších bitov registra R_2, zapisuje ešte **pred** realizáciou operácie **XOR**
- **subblok $\mathbf{C_i}$** sa získa rovnakým spôsobom ako v režime CFB, t.j. operáciou XOR s P_i a subblokom o dĺžke n bitov, ktorý sa vypočíta z pozície n vyšších bitov registra R_2
- podobne ako v režime CFB je potrebné pred šifrovaním naplniť posuvný register R_1 **inicializačným vektorom IV**
- **výhodou** režimu **OFB** v porovnaní s režimom **CFB** je to, že v tomto režime jednoduchá chyba v C_i ovplyvní len jeden bit v dešifrovanom bloku P_i, a teda nedochádza k šíreniu chyby do ďalších dešifrovaných blokov P_i

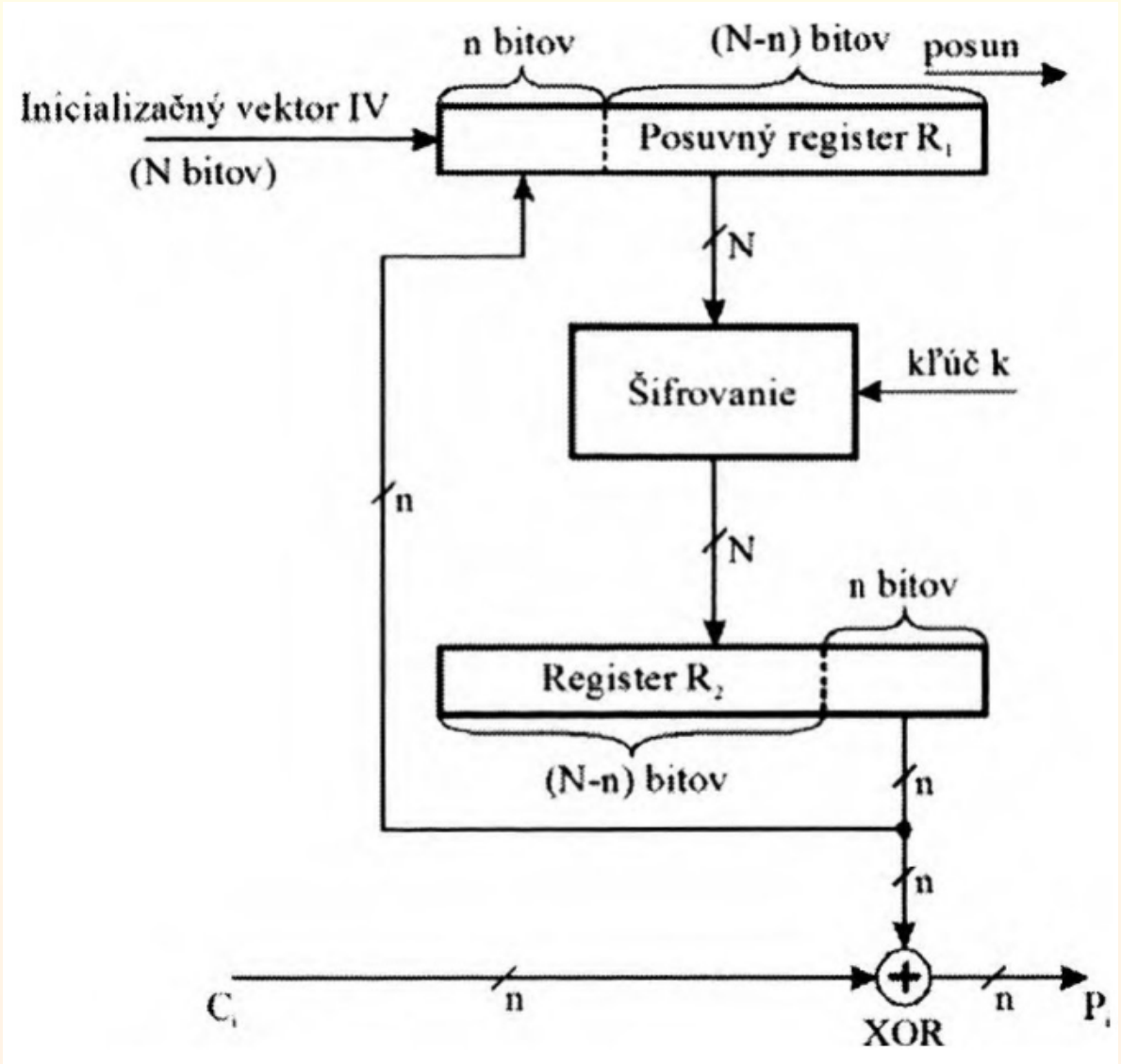
Spätná väzba z výstupu

Princíp režimu OFB - šifrovanie:



Spätná väzba z výstupu

Princíp režimu OFB - dešifrovanie:

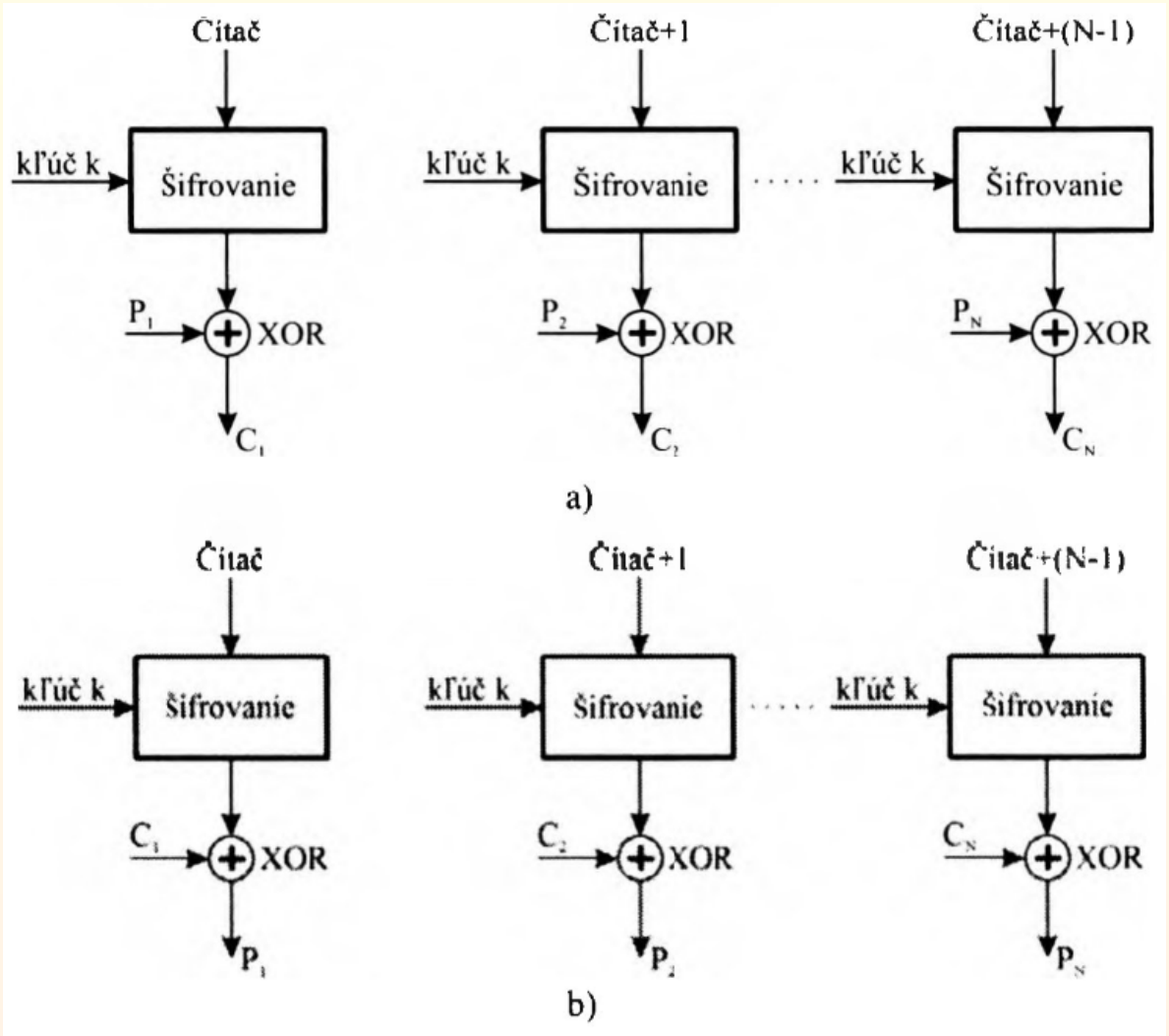


Čítačový režim - režim CTR

- je založený na použití **čítača**, ktorý má rovnaký počet bitov ako je veľkosť bloku otvoreného textu
- čítač sa **nastaví na určitú hodnotu**, ktorá sa potom inkrementuje pre každý šifrovaný blok otvoreného textu
- **výstup čítača** sa podrobí operácii **XOR** s blokom otvoreného textu, čím sa získa blok zašifrovaného textu
- na **dešifrovanie** sa používa rovnaký čítač s rovnakou predvoľbou - jednotlivé bloky otvoreného textu sa získajú operáciou XOR výstupu čítača a príslušného bloku zašifrovaného textu
- **výhody režimu CTR** - jednoduchá hardvérová a softvérová implementácia, pričom bezpečnosť tohto režimu je na úrovni predošlých režimov blokových šifier
- **využíva sa** najmä v technológii ATM, v sieťovej bezpečnosti a IP sieťach

Čítačový režim

Princíp režimu CTR: a) šifrovanie, b) dešifrovanie:



Vybrané symetrické šifry

- symetrické blokové šifry našli **široké použitie v rôznych oblastiach** aplikácií informačných a komunikačných technológií a systémov

- medzi **najpoužívanejšie symetrické blokové šifry** (okrem AES) patria v súčasnosti najmä šifry:
  1. trojnásobný DES (triple DES, resp. 3DES)
  2. Blowfish
  3. RC5

- z **prúdových symetrických šifier** sa najviac používa šifra:
  1. RC4

Trojnásobný DES

- potenciálna zraniteľnosť algoritmu DES metódou totálnych skúšok podnietila záujem o hľadanie alternatívnych prístupov a **zlepšení pôvodného algoritmu**

- jednou z možností zvýšenia bezpečnosti algoritmu DES je **viacnásobné šifrovanie s viacerými kľúčmi**, ktoré je základom modifikácií algoritmu DES

- **najviac používané modifikácie algoritmu DES** sú:
  1. dvojnásobný DES
  2. trojnásobný DES s dvoma kľúčmi
  3. trojnásobný DES s troma kľúčmi

Dvojnásobný DES

- najjednoduchší spôsob viacnásobného šifrovania, ktorý používa **dva šifrovacie procesy E s dvoma rôznymi kľúčmi**
- zašifrovaný text C z otvoreného textu P sa dvojnásobným šifrovaním kľúčmi k_1 a k_2 získa postupom:

$$\mathbf{C = E_{k_2}(E_{k_1}(P))}$$

- proces dešifrovania používa kľúče v obrátenom poradí:

$$\mathbf{P = D_{k_1}(D_{k_2}(C))}$$

- uvedený postup predstavuje zväčšenie dĺžky kľúča na 56 x 2=112 bitov
- v súčasnosti sú známe útoky na tento algoritmus, ktoré sa označujú ako **útoky zo stredu** $\Rightarrow$ výrazne znižujú kryptografickú bezpečnosť dvojnásobného algoritmu DES

Trojnásobný DES (3DES) s dvoma kľúčmi

- používa postupnosť troch krokov, ktoré sa označujú ako **postupnosť EDE** (*Encrypt-Decrypt-Encrypt*)
- proces šifrovania a dešifrovania možno vyjadriť v tvare:

$$\mathbf{C} = \mathbf{E}_{\mathbf{k_1}}(\mathbf{D}_{\mathbf{k_2}}(\mathbf{E}_{\mathbf{k_1}}(\mathbf{P})))$$

$$\mathbf{P} = \mathbf{D}_{\mathbf{k_1}}(\mathbf{E}_{\mathbf{k_2}}(\mathbf{D}_{\mathbf{k_1}}(\mathbf{C})))$$

- použitie 2. kroku v uvedenom algoritme nemá z kryptografického hľadiska podstatný význam, ale **umožňuje dáta šifrované klasickým algoritmom DES dešifrovať algoritmom 3DES**
- efektívna dĺžka kľúča je **112 bitov**
- algoritmus 3DES s dvoma kľúčmi je veľmi rozšírený variant základného algoritmu DES a bol prijatý ako **súčasť štandardu pre manažment kľúčov**

Trojnásobný DES (3DES) s troma kľúčmi

- využíva postupnosť EDE:

$$\mathbf{C} = \mathbf{E}_{\mathbf{k_3}}(\mathbf{D}_{\mathbf{k_2}}(\mathbf{E}_{\mathbf{k_1}}(\mathbf{P})))$$

- kompatibilitu s algoritmom DES možno dosiahnuť, ak

$$\mathbf{k_3} = \mathbf{k_2} = \mathbf{k_1}$$

- efektívna dĺžka kľúča je **168 bitov**

- algoritmus 3DES s troma kľúčmi využívajú **viaceré aplikácie v sieti internet**, napr. PGP a S/MIME

Blowfish

$\Rightarrow$ bloková symetrická šifra, ktorej autorom je **Bruce Schneier** a ktorá transformuje 64-bitový blok otvoreného textu na 64-bitový blok zašifrovaného textu

Charakteristika:

- implementácia na 32-bitových procesoroch, čím sa dosahuje značná rýchlosť šifrovania
- malé nároky na pamäť (menej než 5kB)
- jednoduchá štruktúra, ktorá umožňuje jednoduchú implementáciu
- premenná dĺžka kľúča (do 448 bitov), ktorá umožňuje dosiahnuť kompromis medzi vysokou rýchlosťou a vysokou bezpečnosťou

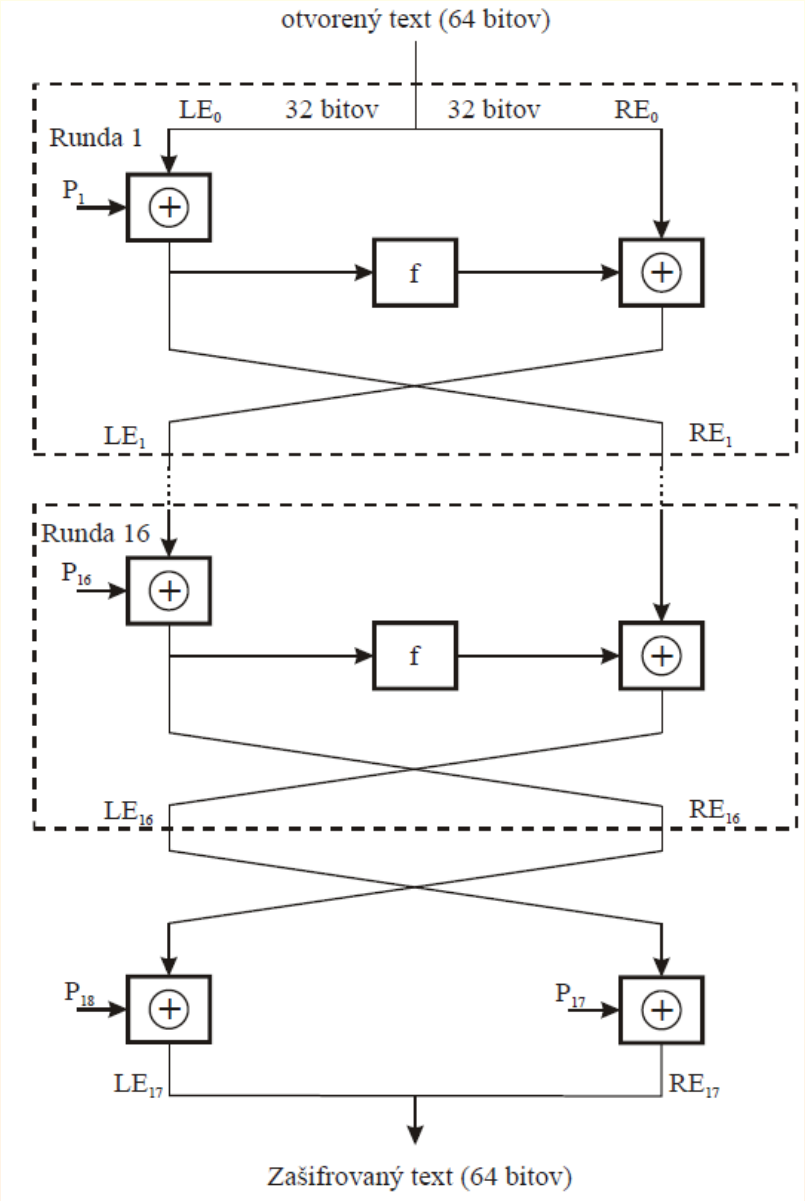
Blowfish

- šifra Blowfish používa **kľúč** v tvare 32-bitových slov a ich počet je od 1 do 14
- uvedený kľúč sa využíva na generovanie kľúčov pre **16 rúnd** a 2 dodatočných kľúčov, takže **celkový počet kľúčov je 18**
- každá runda používa **32-bitový kľúč**
- kľúče pre jednotlivé rundy sú uložené v bloku $\mathbf{P} = \mathbf{P_1}$, $\mathbf{P_2}, \mathbf{P_3}, ...\mathbf{P_{18}}$
- algoritmus šifry Blowfish využíva v každej runde **štyri S-boxy**, ktoré majú 8-bitové vstupy a 32-bitové výstupy a sú závislé od kľúča
- okrem toho algoritmus používa **dve jednoduché operácie**:
  1. sčítanie slov modulo 2^{32}
  2. operáciu XOR

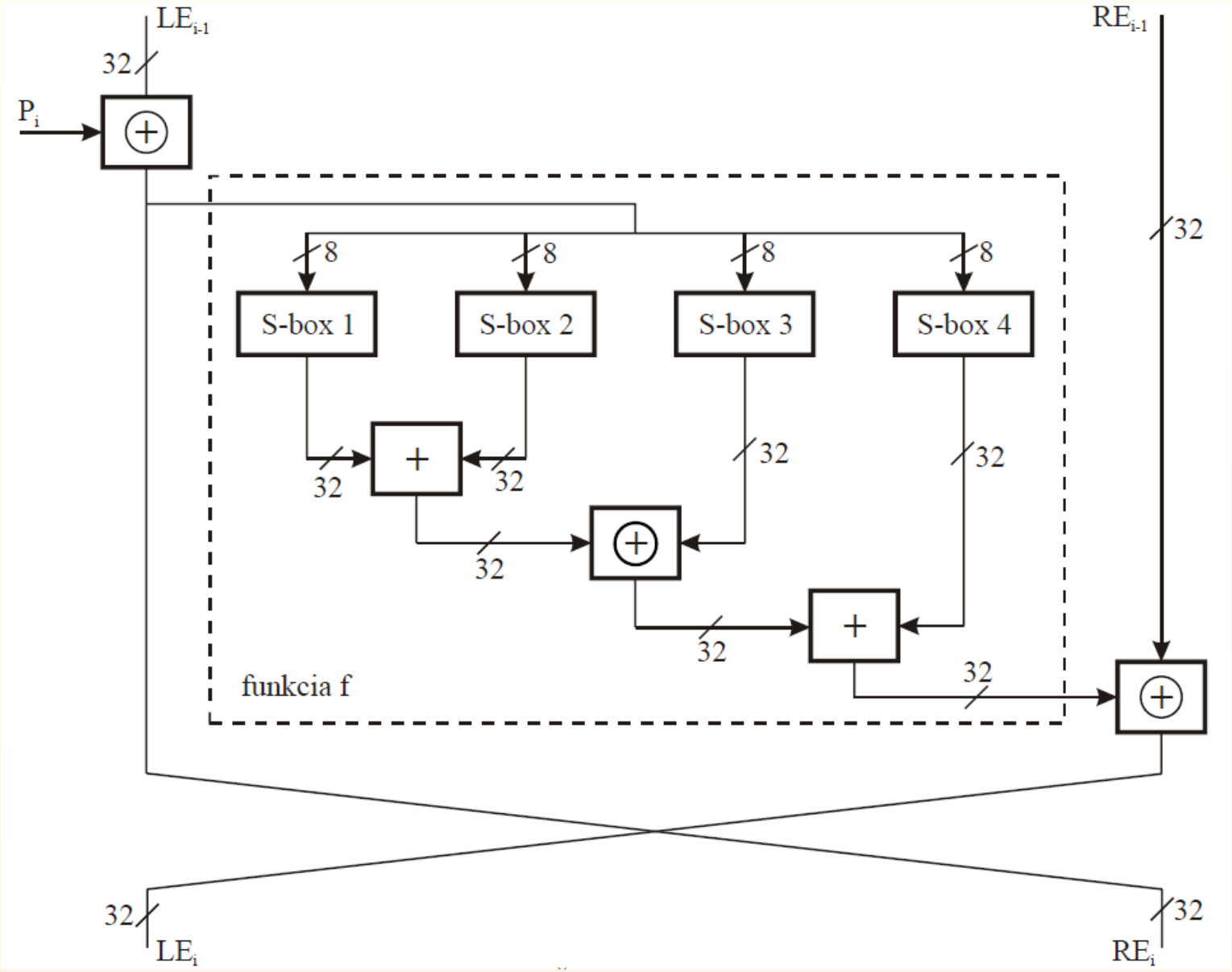
Blowfish

- šifra Blowfish realizuje **operácie s obama polovicami dát** v každej runde, čo je rozdiel oproti klasickej Feistelovej šifre, ktorá realizuje operácie len s jednou polovicou dát $\Rightarrow$ tento prvok zvyšuje kryptografickú bezpečnosť šifry Blowfish,
- substitučné **S-boxy sú závislé od kľúča**, čo zväčšuje lavínový efekt a tým sťažuje kryptoanalýzu,
- šifra pri vhodnej voľbe dĺžky kľúča **nie je** z hľadiska metódy totálnych skúšok **zraniteľná**, pričom dĺžka kľúča môže byť v rozsahu od 32 do 448 bitov,
- výhodou šifry Blowfish je značná **rýchlosť šifrovania**,
- šifru Blowfish **možno voľne používať**, nie je patentovaná.

Štruktúra algoritmu Blowfish



Štruktúra funkcie f



Šifra RC5

- bloková symetrická šifra, ktorej autorom je **Ron Rivest**
- vyznačuje sa **flexibilnosťou parametrov**, ktoré sú označené symbolmi:

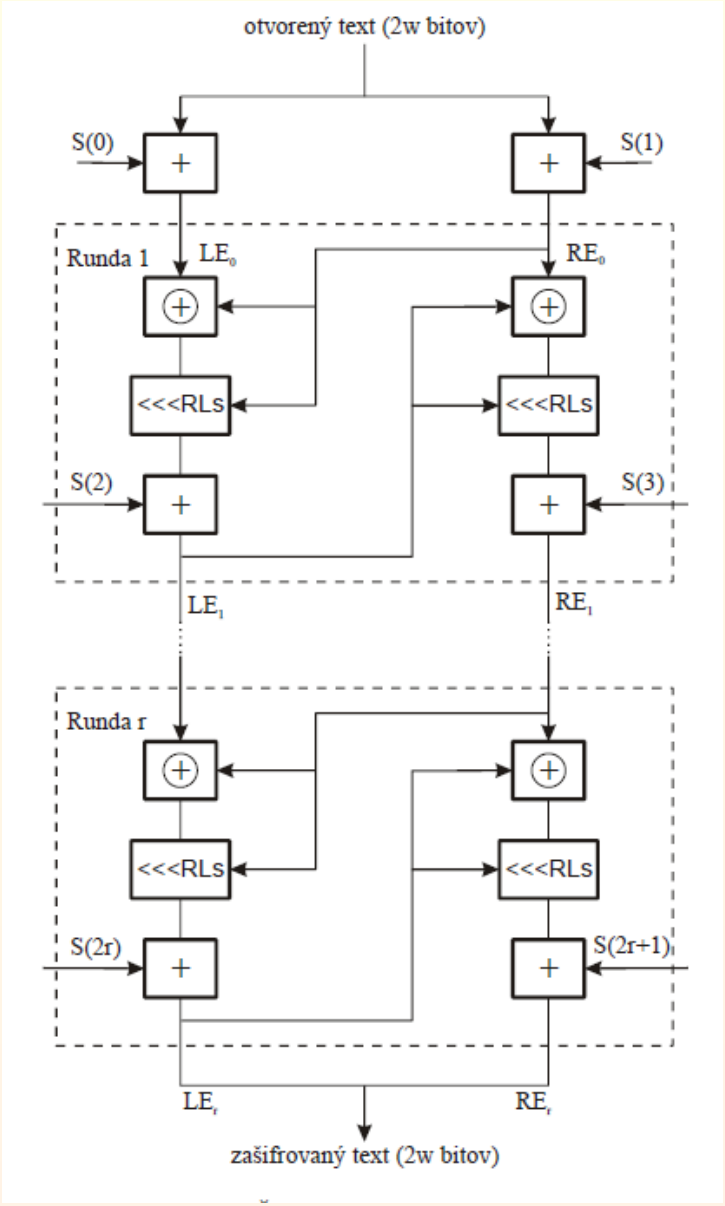
Parameter	Definícia	Rozsah
w	Dĺžka bloku dát v RC5, pričom vstupný blok má dĺžku $2w$	16,32,64
r	Počet rúnd	0,1,...,255
b	Počet bajtov kľúča	0,1,...,255

- jednotlivé verzie šifry RC5 sa označujú ako $\mathbf{RC5-w/r/b}$, napr. $RC5-32/12/16$ je šifra, ktorá používa 32-bitovú dĺžku dát (64-bitový blok otvoreného a zašifrovaného textu), 12 rúnd a dĺžka kľúča je 16 bajtov (16 x 8=128 bitov)
- za nominálnu verziu sa považuje verzia $\mathbf{RC5-32/12/16}$

Šifra RC5

- používa pomerne zložitú množinu operácií na generovanie **subkľúčov** $\Rightarrow$ v každej runde sa využívajú 2 subkľúče, pričom algoritmus využíva ešte 2 kľúče v časti, ktorá nie je súčasťou žiadnej rundy
- pre **počet subkľúčov** $\mathbf{t}$ (každý s dĺžkou w bitov) platí vzťah: $\mathbf{t = 2r + 2}$
- algoritmus šifry RC5 má **subkľúče uložené vo forme bloku**, ktoré má t slov a pozostáva z kľúčov $S(0), S(1), ..., S(t-1)$
- šifra RC5 používa tri jednoduché **operácie**:
  1. *sčítanie*: sčítanie slov, označované symbolom $+$, pričom sa realizuje modulo 2^w
  2. *exclusive-OR*: operácia XOR, ktorá sa označuje symbolom $\oplus$
  3. *rotácia vľavo*: operácia cyklicky posúva slovo o s bitov doľava a označuje sa symbolom $<<< RLs$
- je implementovaná najmä v systéme **RSA Data Security**, ktorý zahrňuje produkty BSAFE, JSAFE a S/MAIL

Štruktúra šifry RC5



Vlastnosti najpoužívanejších blokových šifier

Viaceré **moderné symetrické blokové šifry používajú štruktúru**, ktorá je viac–menej podobná štruktúre DES a Feistelevej blokovej šifry - vyznačujú sa však aj viacerými **odlišnosťami**, ktoré možno zhrnúť takto:

- **variabilná dĺžka kľúča** – umožňuje dosahovať kompromis medzi rýchlosťou šifrovania a stupňom bezpečnosti. Väčšia dĺžka kľúča zaručuje vyššiu odolnosť voči kryptoanalýze metódou totálnych skúšok, ale zvyšuje časové nároky na šifrovanie. Variabilnú dĺžku kľúča používajú šifry Blowfish a RC5.
- **použitie viacerých operácií** – komplikuje kryptoanalýzu, najmä ak použité operácie nespĺňajú distributívny a asociatívny zákon. Všetky uvedené blokové šifry s výnimkou 3DES používajú viaceré operácie.
- **rotácia závislá od dát** – predstavuje alternatívu použitia S–boxov. Táto operácia poskytuje vysoký stupeň konfúzie a difúzie. Uvedenú operáciu používa šifra RC5.

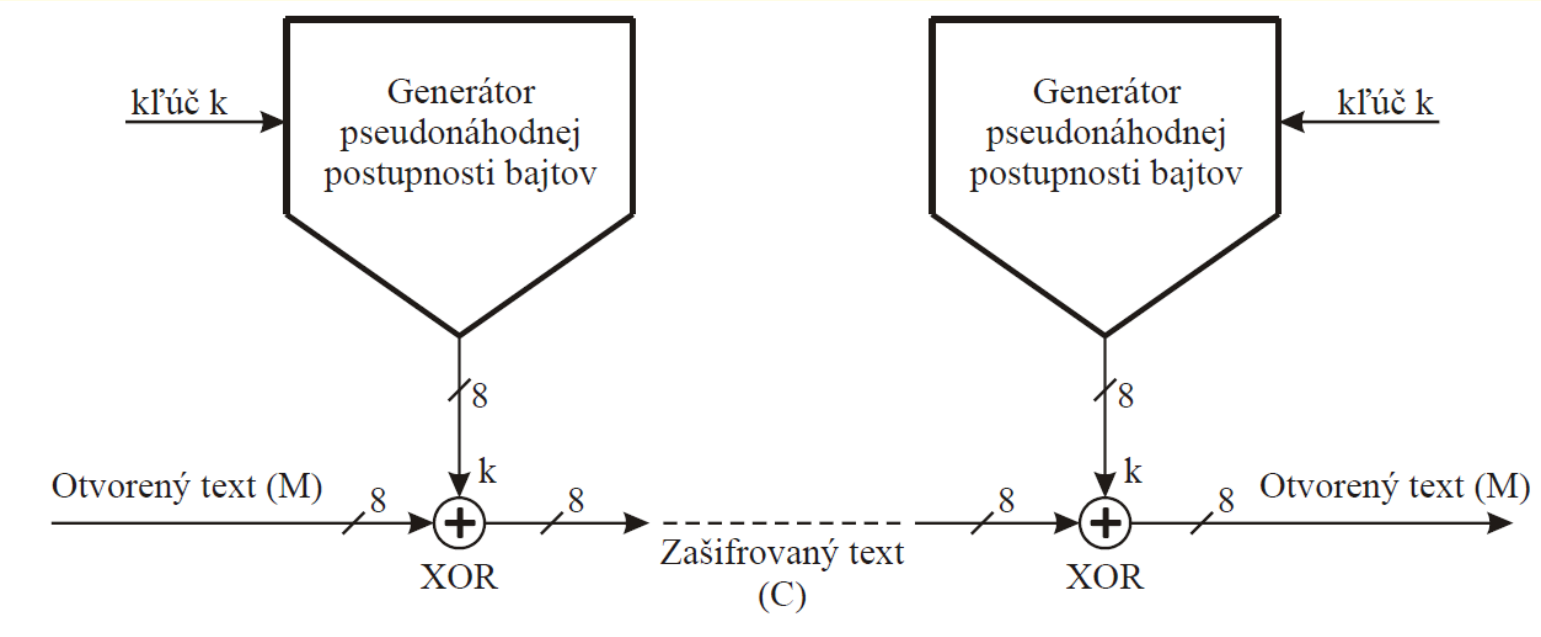
Vlastnosti najpoužívanejších blokových šifier

- **S–boxy závislé od kľúča** – umožňujú modifikovať výstup S–boxov v závislosti od kľúča, čo zvyšuje kryptografickú bezpečnosť vysokým stupňom nelinearity. Tento prístup využíva šifra Blowfish.
- **variabilná dĺžka bloku otvoreného, resp. zašifrovaného textu** – vplýva na stupeň kryptografickej bezpečnosti. Väčšia dĺžka bloku poskytuje vyšší stupeň kryptografickej bezpečnosti. Uvedenú stratégiu využíva šifra RC5.
- **variabilný počet rúnd** – umožňuje dosahovať kompromis medzi stupňom bezpečnosti a rýchlosťou šifrovania. Väčší počet rúnd zvyšuje stupeň kryptografickej bezpečnosti, ale predlžuje dobu šifrovania, resp. dešifrovania. Variabilný počet rúnd používa algoritmus RC5.
- **operácie s oboma polovicami dát v každej runde** – umožňujú dosiahnuť vyšší stupeň kryptografickej bezpečnosti pri minimálnom zvýšení nárokov na dobu šifrovania, resp. dešifrovania. Uvedený postup používajú šifry Blowfish a RC5.

Symetrická prúdová šifra RC4

- jej autorom je **Ron Rivest** - bola navrhnutá v roku 1987 pre kryptografický systém RSA
- šifrovanie otvoreného textu prebieha po bajtoch (8 bitov), teda **operácie sú bajtovo orientované**
- každý bajt otvoreného textu sa podrobí **operácii XOR** s bajtom, ktorý generuje generátor pseudonáhodnej postupnosti bajtov
- **generátor pseudonáhodnej postupnosti bajtov** v šifre RC4 využíva jednoduchú permutáciu bajtov (čísla 0 až 255), ktorá je určená kľúčom k, ktorý má variabilnú dĺžku (1 až 256 bajtov)
- veľkou **výhodou** prúdovej šifry RC4 je značná rýchlosť šifrovania, ktorá vyplýva z podstaty prúdových šifier
- za **bezpečný** sa považuje variant RC4 pri minimálnej dĺžke kľúča 128 bitov

Štruktúra symetrickej prúdovej šifry RC4



- prúdová šifra RC4 je **najpoužívanejšia prúdová šifra** - využíva sa v štandardoch SSL/TLS (*Secure Sockets Layer/Transport Layer Security*), ktoré sú definované pre komunikáciu medzi servermi a WEB prehliadačmi
- je tiež použitá v protokole WEP (*Wired Equivalent Privacy*), ktorý je časťou štandardu IEEE 802.11 pre bezdrôtové siete LAN a je tiež súčasťou protokolu RSA Security

Ďakujem za pozornosť