Simetrični šifarski sistemi

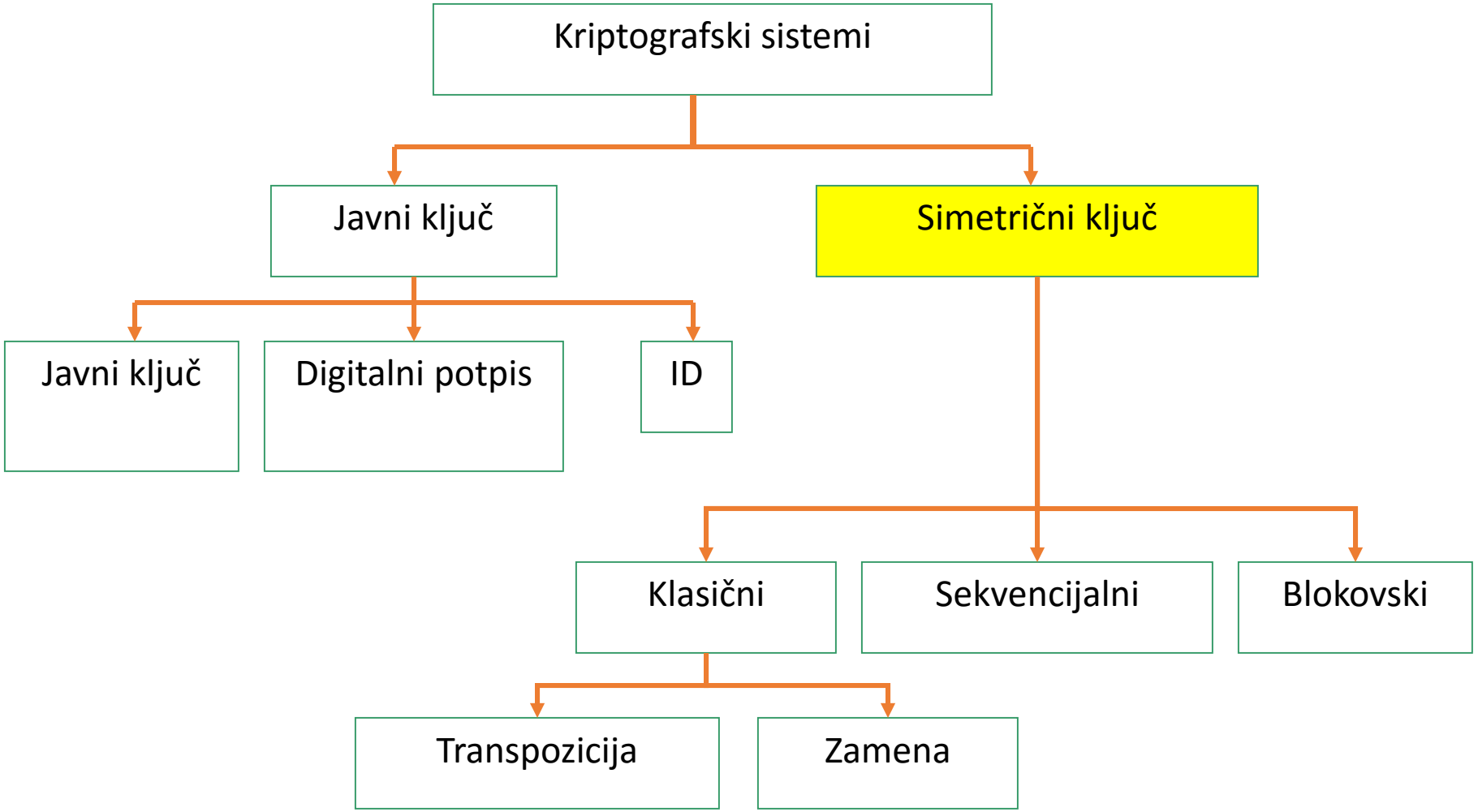
Sadržaj

- Podela šifarskih sistema
- Sekvencijalni algoritmi
  - A5/1
  - RC4
- Blokovski algoritmi
  - Fejstelove mreže
  - DES
  - AES
- Integritet i MAC

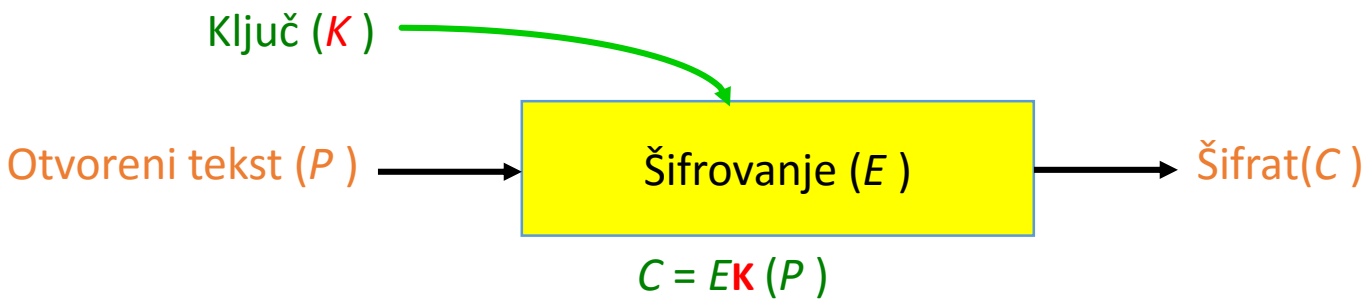
Podela

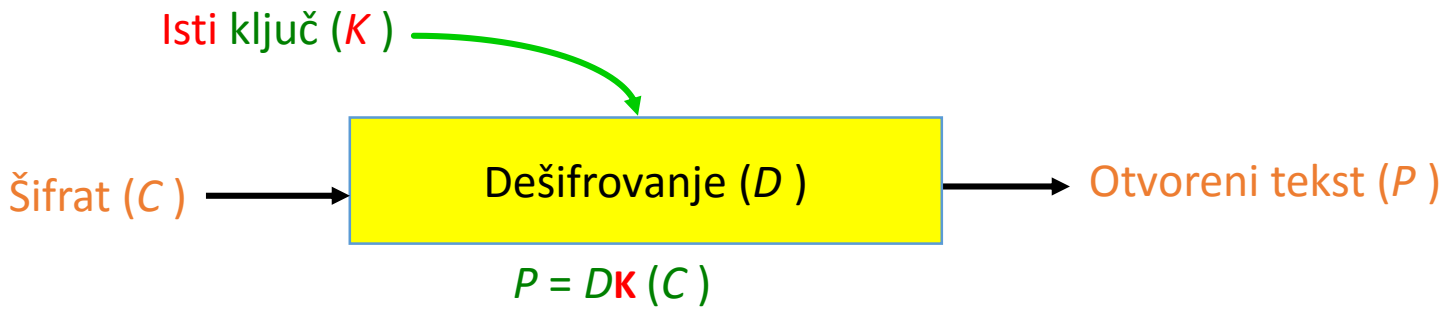
- Osnovna podela kripto sistema.
  - Kriptosistemi sa simetričnim ključem.
    - Koriste isti ključ za šifrovanje i dešifrovanje.
  - Kriptosistemi sa javnim ključem.
    - Dva ključa, za šifrovanje (javni) i za dešifrovanje (privatni).

Podela šifarskih sistema



Simetrični kriptosistemi





$$P = D_K (E_K (P))$$

Uvod

- U prethodnoj lekciji su predstavljeni **klasični kriptografski algoritmi**.
  - Takozvani papir-olovka algortimi.
  - Isti ključ za šifrovanje/dešifrovanje (simetrični ključ).
  - Otvoreni tekst i šifrat se uglavnom predstavljaju slovima.
- Krajem II svetskog rata uvedeni su računari u kriptoanalizu.
  - Odgovor kriptografa: **upotreba računara za šifrovanje i dešifrovanje**.
- Moderni algoritmi podrazumevaju da se otvoreni tekst prvo predstavi preko binarnih brojeva – nizova bitova (**kodovanje**).
  - Način kodovanja (ASCII, ...) nije tajna!
- Algoritmi šifrovanja/dešifrovanja transformišu nizove bitova.

Uvod

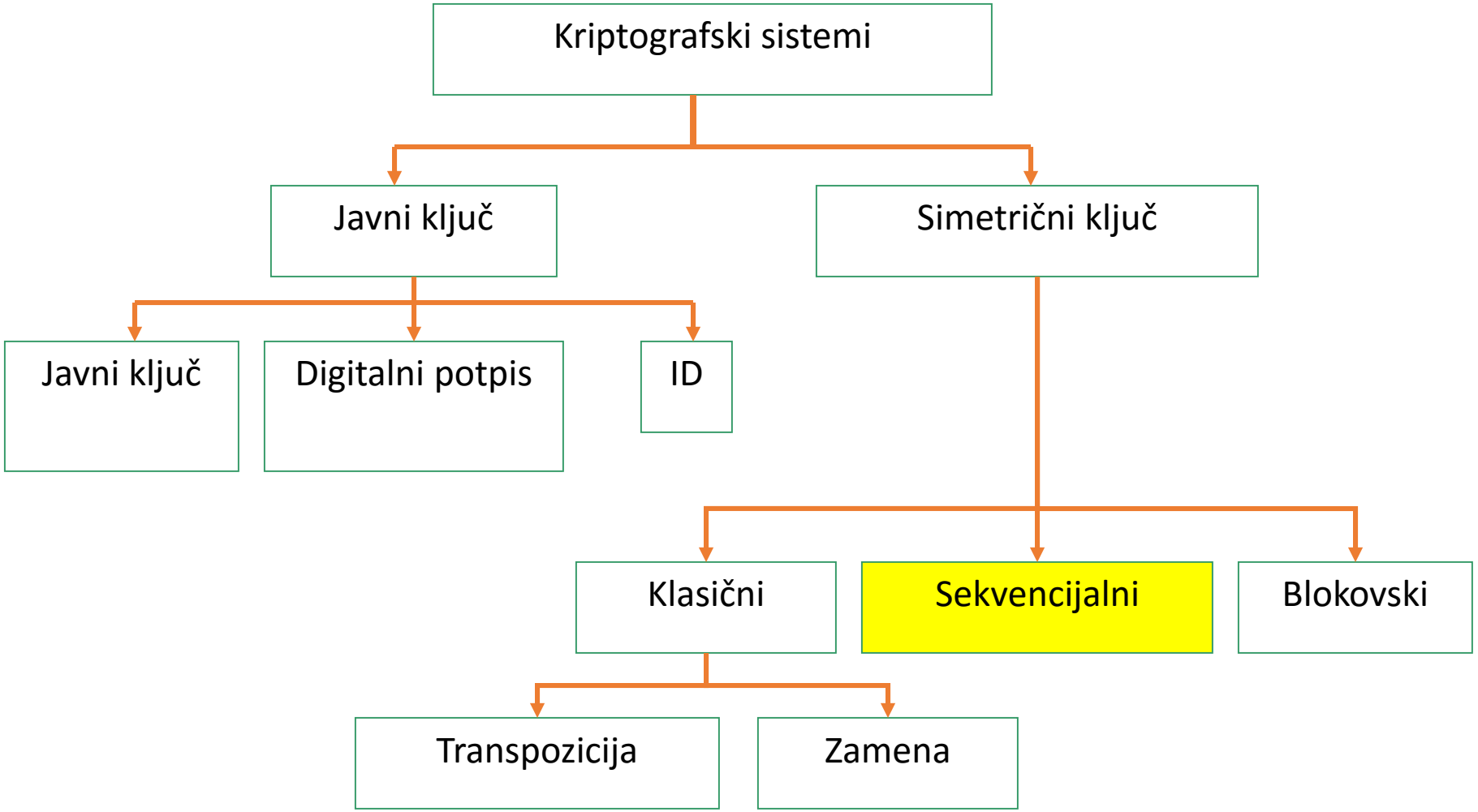
- Šifrovanje pomoću računara je u osnovi **slično klasičnim oblicima šifrovanja**.
  - I dalje se koriste **kombinacije transpozicije i zamene**.
- Osnovne razlike:
  - Moguće je realizovati daleko **složenije šifre** primenom računara od onih koji se mogu realizovati mehanički.
  - **Brzina**: elektronika je neuporedivo brža od mehanike.

Simetrični kriptosistemi – podela

- Algotitmi koji deluju na jedan bit (ponekad bajt)* otvorenog teksta se nazivaju **sekvencijalni (*stream*) algoritmi**.
  - Ključ je **slučajni** niz male dužine!
  - Na osnovu kratkog ključa (i još nekih podataka) obe strane generišu **radni ključ** (*keystream*).
  - Radni ključ se potom koristi kao kod OTP-a.
  - Radni ključ je iste dužine kao poruka.
- Ukoliko se otvoreni tekst podeli na blokove bitova određene dužine (64bita, 128bita...) pa se svaki blok (ne) zavisno šifruje, govorimo o **blokovskim algoritmima**.
  - Zasnovane na **konceptu kodnih knjiga**.
  - Ključ blokovske šifre određuje kodnu knjigu.
  - Svaki ključ određuje novu (različitu) kodnu knjigu.
  - Koriste se koncepti “konfuzije” i “difuzije”.

* Pre pojave računara, algoritmi su kao osnovnu jedinicu otovrenog teksta podrazumevali jedno slovo (znak) koje se može predstaviti sa jednim bajtom.

Podela šifarskih sistema



Sekvencijalni algoritmi

- Koriste ključ *K* dužine n bita na osnovu kog se generiše **radni ključ** (n « dužine poruke).
  - *Sekvencijalna šifra* (*K*) = *S*
  - *S* je radni ključ, $S = s_0 s_1 s_2 ...$
  - *S* je dužine poruke (otvorenog teksta).
- Radni ključ se koristi za šifrovanje poruke (XOR).
  - $c_0 = p_0 \oplus s_0$
  - $c_1 = p_1 \oplus s_1$
  - $c_2 = p_2 \oplus s_2$
  - ...
  - $P = p_0 p_1 p_2 ...$ je otvoreni tekst.
  - $C = c_0 c_1 c_2 ...$ je šifrat.

Sekvencijalni algoritmi

- Dešifrovanje:
  - $p_0 = c_0 \oplus s_0$
  - $p_1 = c_1 \oplus s_1$
  - $p_2 = c_2 \oplus s_2$
  - ...
- Prijemna i predajna strana treba da imaju:
  - Isti **sekvencijalni algoritam** (generisanje radnog ključa).
  - Isti **ključ** *K*.
- Ključ se dostavlja posebnim sigurnim kanalom (kurir, ...).
- Sistem predstavlja adaptaciju OTP-a sa kratkim ključem.
  - Nije dokazivo siguran!
  - Zašto?

Sekvencijalni algoritmi

- Problem sa OTP-om:
  - Ključ *K* je **slučajan**.
    - Nije moguće da se isti ključ generiše na predajnoj i prijemnoj strani.
    - Potreban dodatni kanal za distribuciju *K*.
- Sekvencijalne šifre imaju isti zahtev za distribucuju *K* ali je on **mnogo manje dužine**.
  - Radni ključ se generiše na osonvu K upotrebom generatora radnog ključa.
- Generator radnog ključa mora **zadovoljiti stroge kriptografske zahteve**.
  - Zasnovan na matematičkom pristupu.
  - Zahteva detaljnu statističku analizu da bi se potvrdilo da su osobine dobijenog radnog ključa što bliže slučajnom nizu.

Sekvencijalni algoritmi

- Kako funkcioniše *one-time pad*?
  - Šifrat = otvoreni tekst $\oplus$ Ključ
  - Ključ mora biti slučajan niz bitova **iste dužine kao otvoreni t**ekst.
- Ideja: zameniti slučajan niz sa “pseudo-slučajnim”.
  - Projektovati generator pseudo-slučajnih brojeva (PRNG).
  - PRNG na osnovu slučajnog početnog stanja (manji broj bita) generiše mnogo duži niz pseudo-slučajnih bitova.
    - Npr. 128-bita u 10^{19} pseudo-slučajnih bitova.
- Postoji više rešenja za projektovanje PRNG.
  - Jedano od rešenja je primena pomeračkih registara (linearni pomerački registari sa povratnom spregom).

Sekvencijalni algoritmi

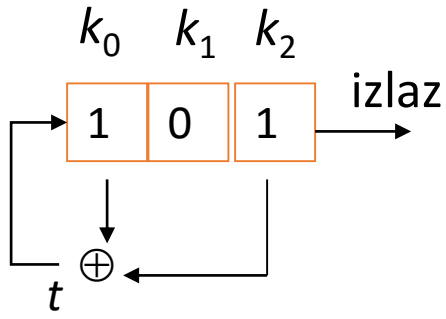
- Linearni pomerački registri se sastoje se od:
  - **Pomeračkog registra** (dužine *n* bita): niz bita koji se u svakom taktu pomera za jedno mesto u desno.
  - **Funkcije povratne sprege**: određuje, na osnovu vrednosti nekih bita u registru, koji bit će se upisati na prvo mesto (može biti linearna).
- **Izlaz** iz pomeračkog registra je jedan bit, najčešće krajnji desno.
- **Početno stanje** i **povratna veza** određuju **izlaz**.
- **Perioda pomeračkog registra** je broj bita izlazne sekvence pre nego što počne da se ponavlja.

Sekvencijalni algoritmi

- Pomerački registar dužine n, može da ima 2^{n-1} različitih početnih stanja (ne mogu sve nule).
- Pravilnim odabirom povratne sprege može se postići perioda izlaznog niza od 2^{n-1} bita.
- Izlazni niz ima osobine **pseudo-slučajnog** niza (**periodičnost**!).
- Nažalost, ima osobine koje nisu prihvatljive:
  - Prvih n bita izlaza su jednaka početnom stanju.
  - Na osnovu poznavanja $2n$ bita izlaza može se rekonstruisati generator (linearnost, kriptoanaliza).

Sekvencijalni algoritmi

- Primerna LPR za generisanje radnog ključa:
  - Ključ *K* određuje početno stanje.
- Primer 1:
  - Ključ dužine n=3 (npr. $k_0k_1k_2$: 101)
  - U svakom koraku (takt): $t=k_0 \oplus k_2$, $k_i=k_{i-1}$, $k_0=t$

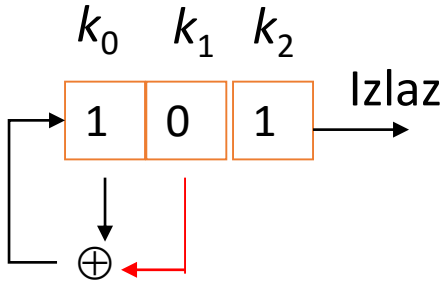


takt	k_0	k_1	k_2	
0	1	0	1	Izlaz *S*
1	0	1	0	
2	0	0	1	
3	1	0	0	
4	1	1	0	
5	1	1	1	
6	0	1	1	
7	1	0	1	

Perioda 2^n-1

- S – radni ključ
- Maksimalne dužine: 2^n-1
- Periodičan

- Primer 2:
  - Ključ dužine n=3 (npt. $k_0k_1k_2$.: 101)
  - U svakom koraku (takt): $t=k_0 \oplus k_1$, $k_i = k_{i-1}$, $k_0 = t$



takt	k_0	k_1	k_2	
0	1	0	1	Izlaz S
1	1	1	0	
2	0	1	1	
3	1	0	1	
4	1	1	0	
5	0	1	1	
6	1	0	1	
7	.	.	.	

Perioda < 2^n-1

- Izbor povratne sprege određuje periodu!

Sekvencijalni algoritmi

- LPR daju **periodični izlaz**.
  - Radni ključ se ponavlja, može se prihvatiti ako je perioda dovoljno velika (?) i koristi se samo unutar jedne periode.
  - Veličina periode zavisi od povratne sprege.
- **Slučajnost**.
  - Statistički gledano ovako generisan radni ključ ima elemente (pseudo) slučajnosti, ali dovoljno je poznavanje 2*n* bita da bi se izračunali naredni biti.
  - **Neprihvatljivo** sa stanovišta sigurnosti.
- Upotrebljivost LPR za generisanje radnog ključa.
  - **Pametnim kombinovanjem više LPR**, uz **uvođenje elemenata nelinearnosti** mogu se prevazići navedeni problemi (računska, ali ne i bezuslovna sigurnost!).

Sekvencijalni algoritmi

- Danas nisu toliko popularne kao blok šifre.
- Razmotrićemo dva predstavnika:
- A5/1
  - Zasnovana na pomeračkim (*shift*) registrima.
  - Primena: GSM mobilna telefonija (telefon-bazna stanica).
  - Algoritam je držan u tajnosti, ali je spletom okolnosti dospeo u javnost.
  - Pokazalo se da ima slabosti!
- RC4
  - Zasnivaju se na primeni promenljivih tabela.

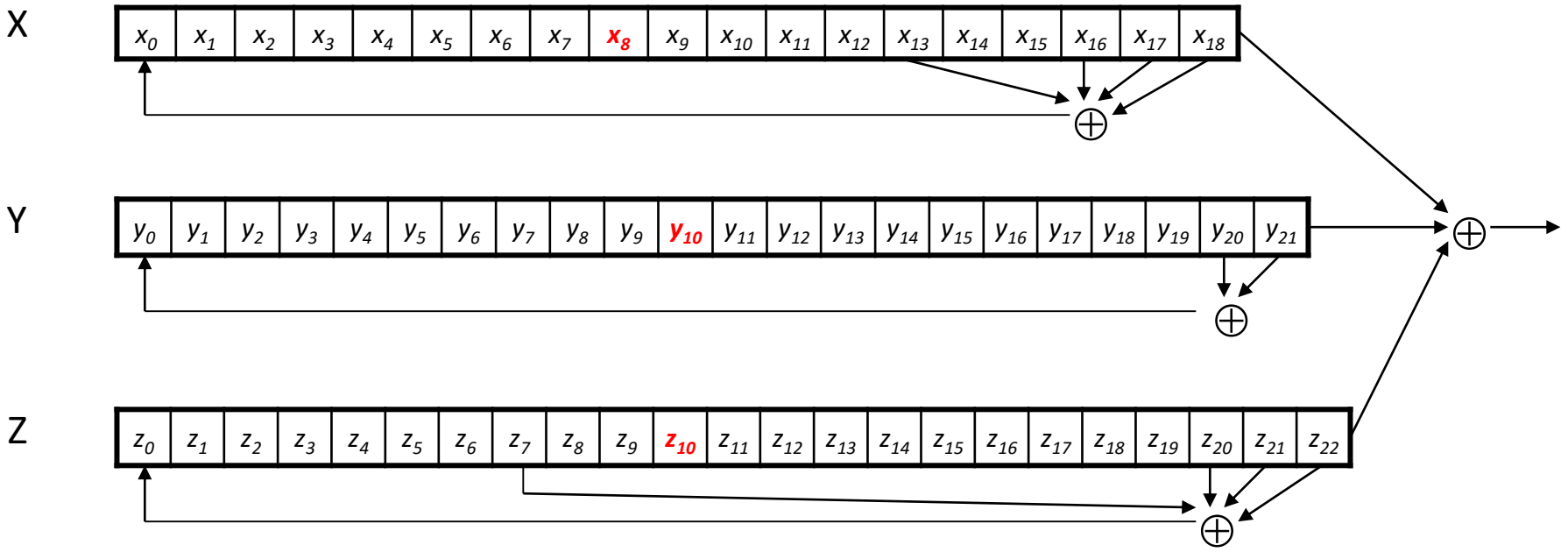
A5/1

- A5/1 sadrži 3 pomeračka registra (X, Y i Z).
  - X: dužine 19 bita (x_0,x_1,x_2, …,x_{18})
  - Y: dužine 22 bita (y_0,y_1,y_2, …,y_{21})
  - Z: dužine 23 bita (z_0,z_1,z_2, …,z_{22})
- Ključ je dužine 64 bita (19+22+23).
  - Koristi se da bi se postavile početne vrednosti sva 3 registra.
  - Sledi generisanje radnog ključa.

A5/1

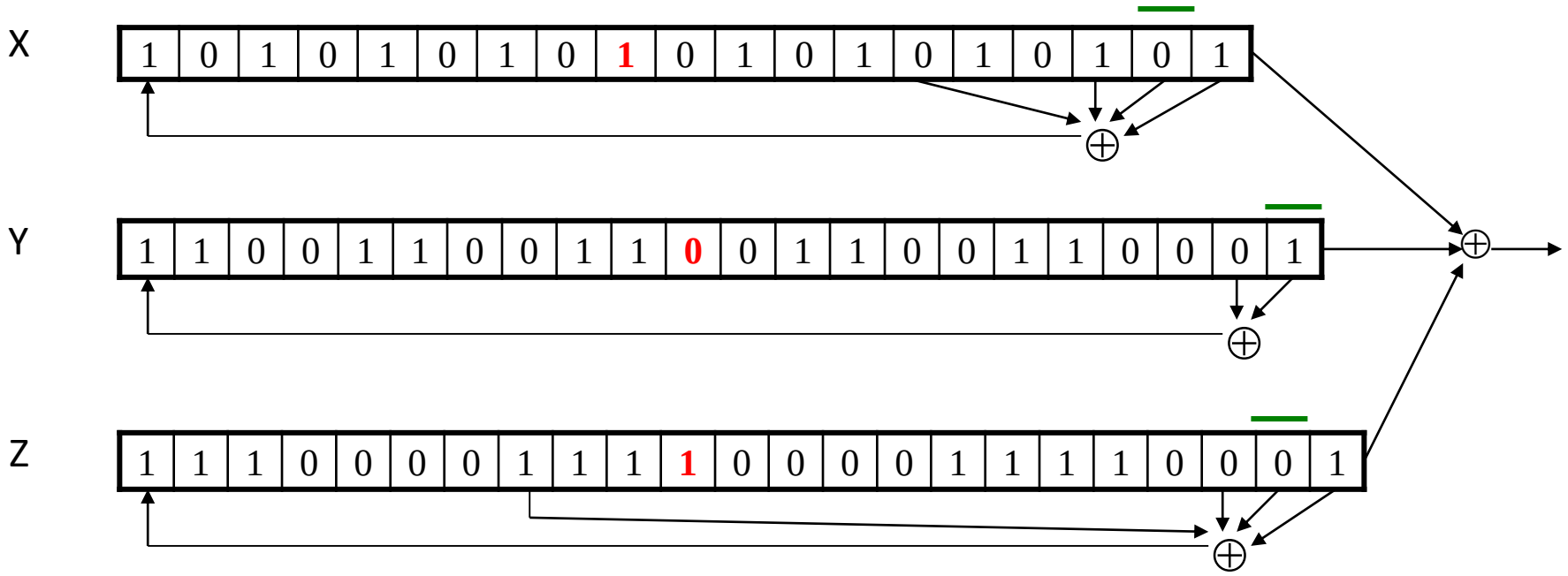
- U svakom koraku: m = maj (x_8, y_{10}, z_{10})
  - Primer: maj (0,1,0) = 0 и maj (1,1,0) = 1
- Ako je $x_8 = m$ tada se računa t i šiftuje X
  - $t = x_{13} \oplus x_{16} \oplus x_{17} \oplus x_{18}$
  - $x_i = x_{i-1}$ za i = 18,17,…,1 i $x_0 = t$
- Ako je $y_{10} = m$ tada se računa t i šiftuje Y
  - $t = y_{20} \oplus y_{21}$
  - $y_i = y_{i-1}$ za i = 21,20,…,1 i $y_0 = t$
- Ako je $z_{10} = m$ tada se računa t i šiftuje Z
  - $t = z_7 \oplus z_{20} \oplus z_{21} \oplus z_{22}$
  - $z_i = z_{i-1}$ za i = 22,21,…,1 i $z_0 = t$
- Bit radnog ključa se računa kao: $x_{18} \oplus y_{21} \oplus z_{22}$

A5/1



- Svaka vrednost (x_i, y_i, z_i) je jedan bit.
- (Unutrašnji) ključ određuje početno stanje registara.
- Šiftovanje registara zavisi od vrednosti (x_8, y_{10}, z_{10}).
- Bit radnog ključa se u svakom taktu dobija XOR-vanjem poslednjih bitova registara (x_{18}, y_{21}, z_{22}).

A5/1



- U ovom primeru m = maj(x_8, y_{10}, z_{10}) = maj(**1**,**0**,**1**) = **1**.
- Registar X se šiftuje, Y se ne šiftuje, Z se šiftuje.
- Bit radnog ključa: $s = 0 \oplus 1 \oplus 0 = 1$.
  Koristi se za šifrovanje/dešifrovanje jednog bita otvorenog teksta/šifrata.

Pomerački registri

- Broj bita (perioda) radnog ključa koji se generiše na osnovu 64-bitnog ključa je veoma veliki.
- Kripto sistemi koji koriste pomeračke registre se najčešće **realizuju hardvereski**.
- **Softverska implementacija** je moguća ali često je **manje efikasna**.
  - Savremeni procesori omogućavaju prihvatljivo rešenje u nekim slučajevima.
- Pomerački registri još uvek imaju primenu u nekim kripto sistemima.

RC4

- Sekvencijalni algoritam.
- Nastao 1987. godine (Ron Rivest za RSA Data Security Inc.)
  - Algoritam je 7 godina držan u tajnosti.
  - 1994 neko je anonimno poslao izvorni kôd na Cypherpunks mejling listu.
- Optimizovan za softversku realizaciju.
- U svakom koraku (takt) generiše **1 bajt** radnog ključa.
  - A5/1 generiše **1 bit** (efikasan u hardveru).
- Zantno jednostavniji algoritam, koristi takozvanu **lookup tabelu**.

RC4

- **Sadržaj tabele** je uvek neka od **permutacija bajtovskih vrednosti**: 0,1,…,255.
- **Početni raspored** u tabeli se određuje na osnovu ključa.
- U svakom koraku RC4:
  - Menja se **raspored elemenata** u tekućoj tabeli
  - Bira se **jedan bajt** iz tabele za radni ključ.
- RC4 algoritam se sastoji iz **2 faze**:
  - Inicijalizacija
  - Generisanje radnog ključa.

RC4 inicijalizacija

- Svaki element tabele S[ ] je jedan bajt (0,1,...,255).
- Ključ key[ ] sadrži N bajtova ključa.

```
for i = 0 to 255
    S[i] = i
    K[i] = key[i  mod  N]
next i
j = 0
for i = 0 to 255
    j = (j + S[i] + K[i])  mod 256
    swap(S[i],S[j])
next i
i = j = 0
```

RC4 generisanje radnog ključa

- Za svaki bajt radnog ključa zameni mesta elementima tabele i selektuj jedan bajt.

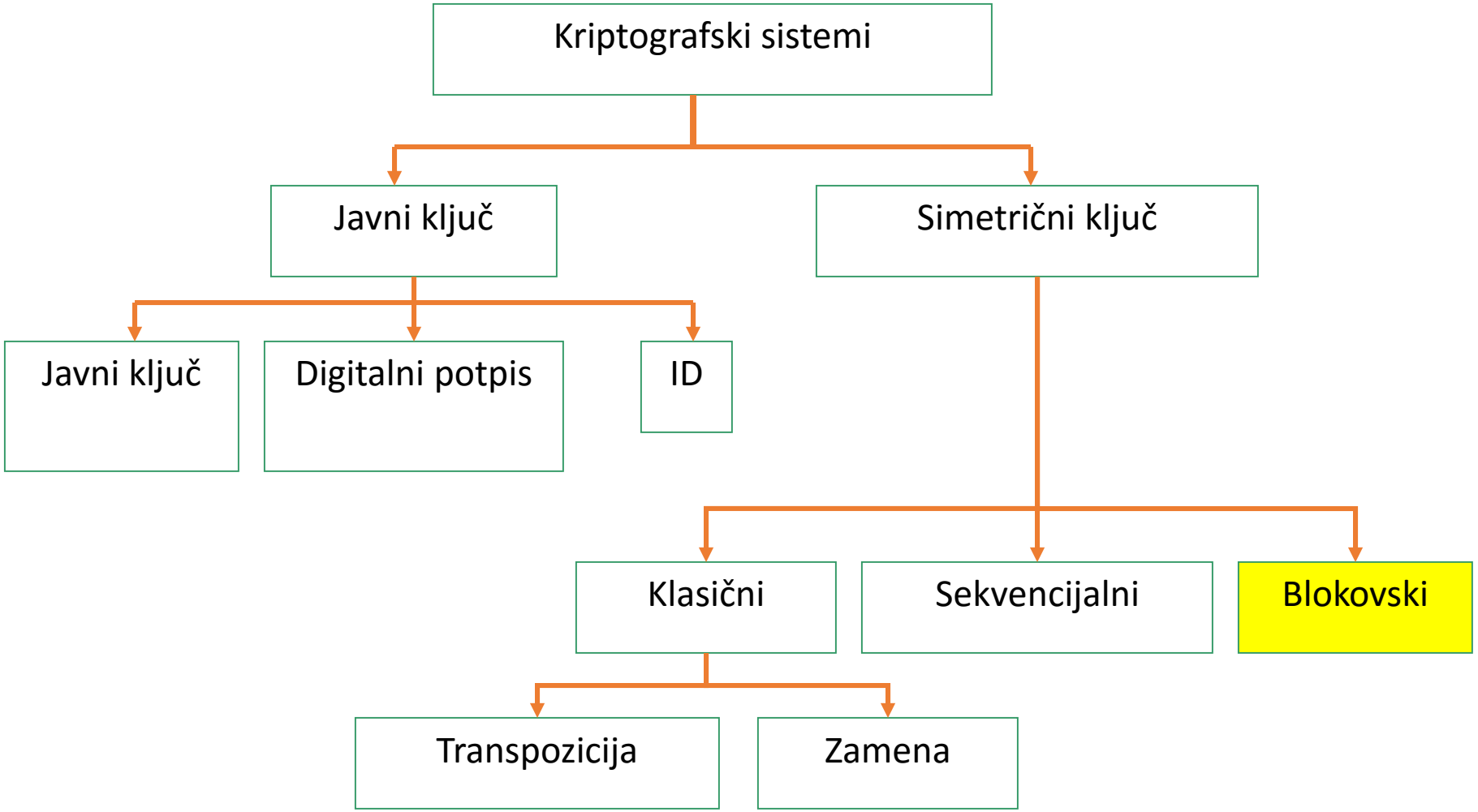
$$
\begin{aligned}
&i = (i + 1) \mod 256 \\
&j = (j + S[i]) \mod 256 \\
&\texttt{swap}(S[i], S[j]) \\
&t = (S[i] + S[j]) \mod 256 \\
&\texttt{keystreamByte} = S[t]
\end{aligned}
$$

- Bajt radnog ključa se koristi kao i kod OTP-a.
- Napomena: prvih 256 bajtova radnog ključa mora da se odbaci.
  - U protivnom napadač može da rekonstruiše (unutrašnji) ključ.

Sekvencijalni algoritmi (rezime)

- Prednosti:
  - Koriste se tamo gde je neophodna brzina.
    - Rad u realnom vremenu.
    - Zaštita govora.
  - Pogodne za primenu u lošim uslovima prenosa.
    - Nema propagacije greške.
  - Efikasno se realizuju hardverski.
- Sekvencijalne šifre su bile dominantne u prethodnom periodu.
  - Današnji procesori su dovoljno brzi tako da se koriste softverska rešenja.
- Budućnost sekvencijalnih šifara?
  - Shamir: “kraj sekvencijalnih šifara”
  - Možda je preteška ocena …

Podela šifarskih sistema



Blokovske šifre

- Otvoreni tekst i šifrat se sastoje od blokova fiksne dužine.
- Šifrovanje se primenjuje na blokove i daje blokove šifrata iste dužine.
- Šifrat se dobija iz bloka otvorenog teksta **višestrukom primenom funkcije** koja se naziva **runda**.
- **Ulazni parametri** runde su:
  - ključ i
  - izlaz iz prethodne runde.
- Najčešće se realizuje softverski.

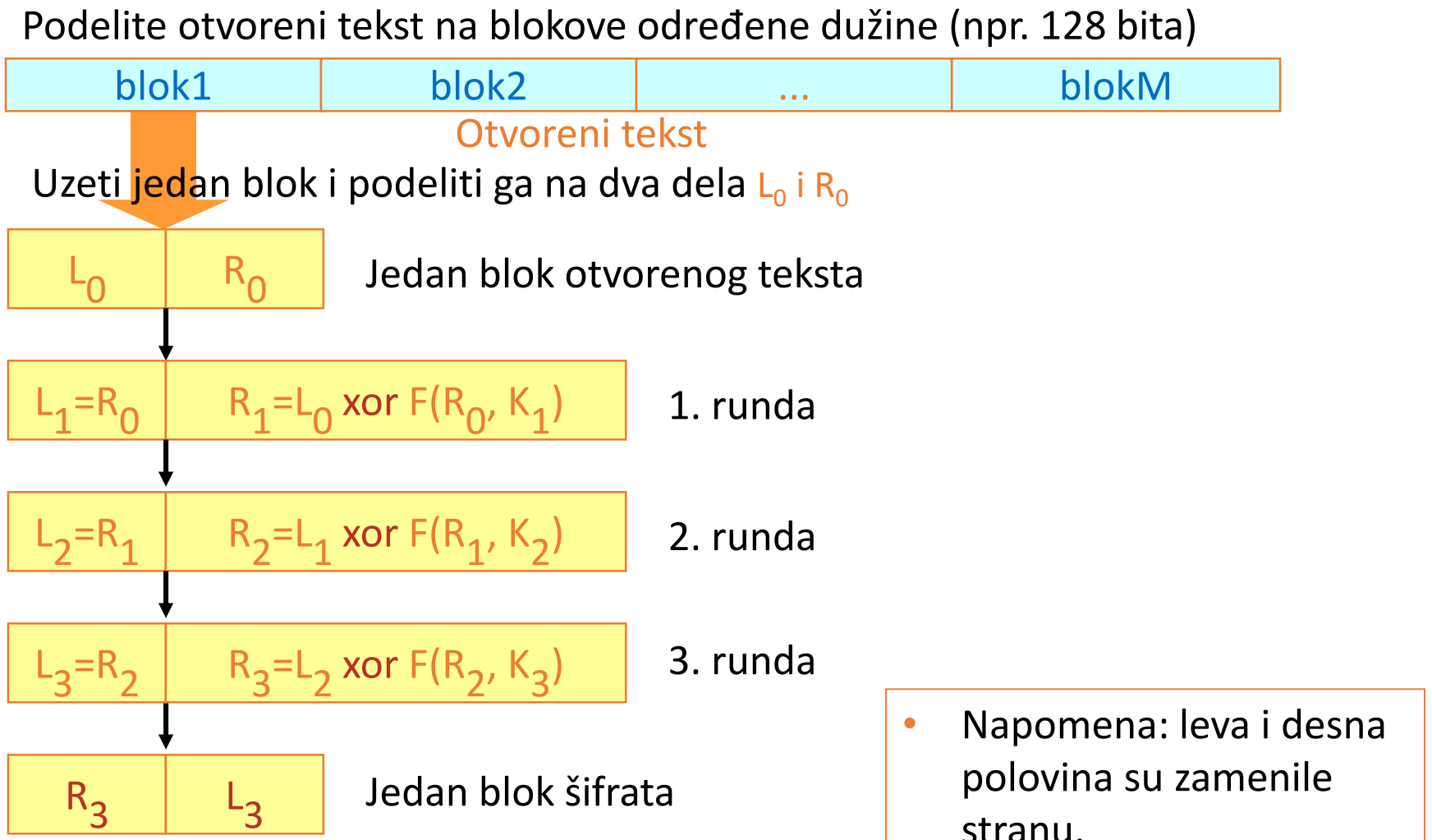
Blokovske šifre

- Zahtevi:
  - **Svojstvo difuzije**.
    - Neka su P_i i P_j blokovi otvorenog teksta a C_i i C_j blokovi šifrata.
    - Poznavanje para P_i i C_i ne sme da omogući da se na osnovu C_j odredi odgovarajuće P_j.
    - Male promene u bloku otvorenog teksta (1 bit) treba da izazovu nepredvidive promene u bloku šifrata.
  - **Svojstvo konfuzije**.
    - Kod napada potpunom pretragom ključa, svi ključevi treba da budu jednako verovatni.
  - **Kompletnost**.
    - Svaki bit šifrata treba da je funkcija svakog bita ključa.

Fejstelova šifra

- Fejstelova (Feistel) šifra predstavlja tip blokovske šifre, a ne posebnu šifru.
- Ideja:
  - Podeliti otvoreni tekst na blokove.
  - Blok podeliti na dva dela: levi (*L*) i desni (*R*).
  - Otvoreni tekst = (L_0, R_0)
  - U svakoj rundi *i* (*i* =1,2,...,n), izračunati
    - $L_i = R_{i-1}$
    - $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$
    - gde je:
      - F **funkcija runde**
      - K_i je **podključ**.
        - Podključ se dobija kombinovanjem bitova ključa K.
  - Šifrat = (L_n,R_n)

Fejstel: primer šifrovanja u 3 runde



- Napomena: leva i desna polovina su zamenile stranu.

Fejstelova šifra

- Dešifrovanje:
  - Šifrat = (L_n,R_n)
  - U svakoj rundi *i* (*i* = n, n–1, …, 1) izračunati
    - $R_{i-1} = L_i$
    - $L_{i-1} = R_i \oplus F(R_{i-1},K_i)$
    - gde je:
      - F **funkcija runde**
      - K_i je podključ.
  - Otvoreni tekst = (L_0,R_0)
- Napomena:
  - Formula “radi” za **bilo koju funkciju** F.
  - Ali, **kriptografski je sigurna samo za neke funkcije** F.

DES – Data Encryption Standard

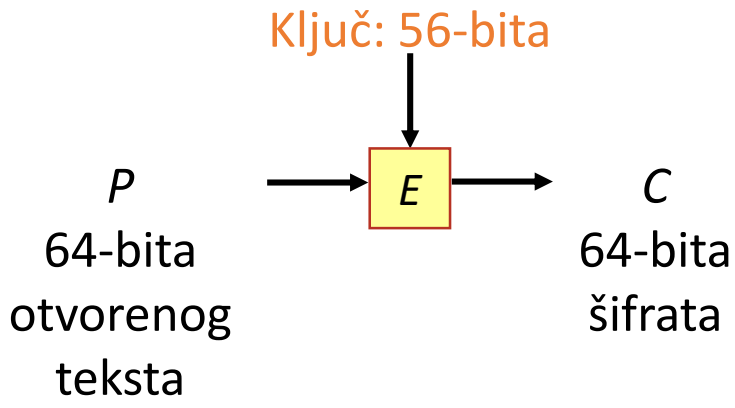
- Životni ciklus.
  - 1973. NBS (“National Bureau of Standards”, SAD, danas NIST) raspisala konkurs za komercijalni kriptografski algoritam.
    - Neuspešno.
  - 1974. Ponovljeni konkurs.
    - NSA urgira da IBM prijavi svoje rešenje Lucifer.
    - NSA modifikuje IBM-ovu šifru.
  - 1975. NBS predstavlja rešenje.
    - Mnogo opravdanih komentara oko modifikacije, “mogućih” zadnjih vrata, skraćenja dužine ključa sa 128 na 56 bita ...
  - 1976. Objavljen standard.
  - 1976 – 1998. Masovna primena u svetu.
  - 1998. Pokazano da je praktično izvodljiv napad potpunom pretragom ključa!

DES – Data Encryption Standard

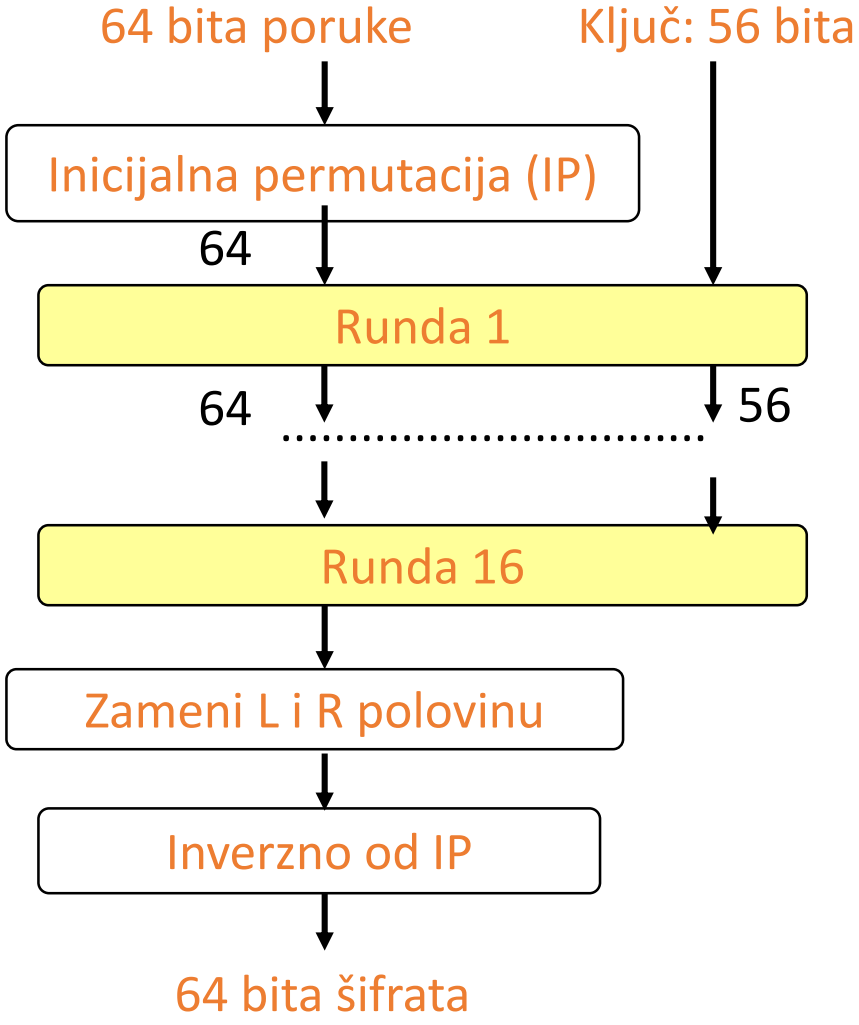
- Cilj dizajna:
  - Šifrat treba da zavisi od otvorenog teksta i ključa na složen način.
    - Uvesti princip **konfuzije**.
  - Svaki bit šifrata treba da je funkcija svih bitova otvorenog teksta i svih bitova ključa.
    - Uvesti princip **difuzije**.
  - **Lavinski efekat**.
    - Male promene ulaza treba da izazovu velike promene izlaza.
    - Kod DES-a promena 1 bita ključa ili 1 bita (od 64 bita) otvorenog teksta menja 50% bita bloka šifrata.

DES – Data Encryption Standard

- DES je Fejstel-bazirana šifra.
- Dužina bloka je 64 bita.
- Ključ je dužine 56 bita.
  - $2^{56} = 7{,}2 \times 10^{16}$ mogućih ključeva.
- 16 rundi.
- U svakoj rundi se koristi podključ dužine 48 bita.

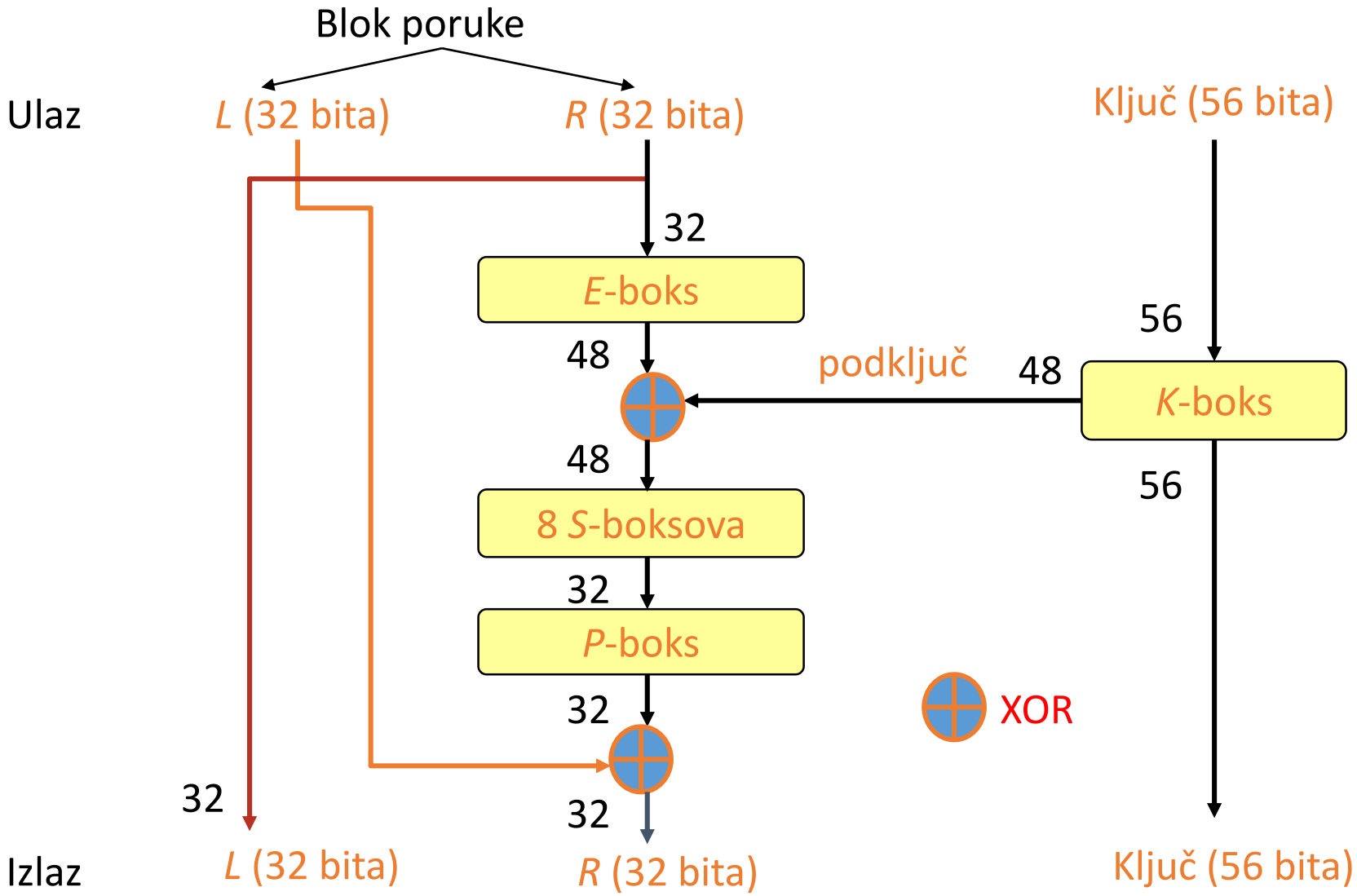


DES: struktura



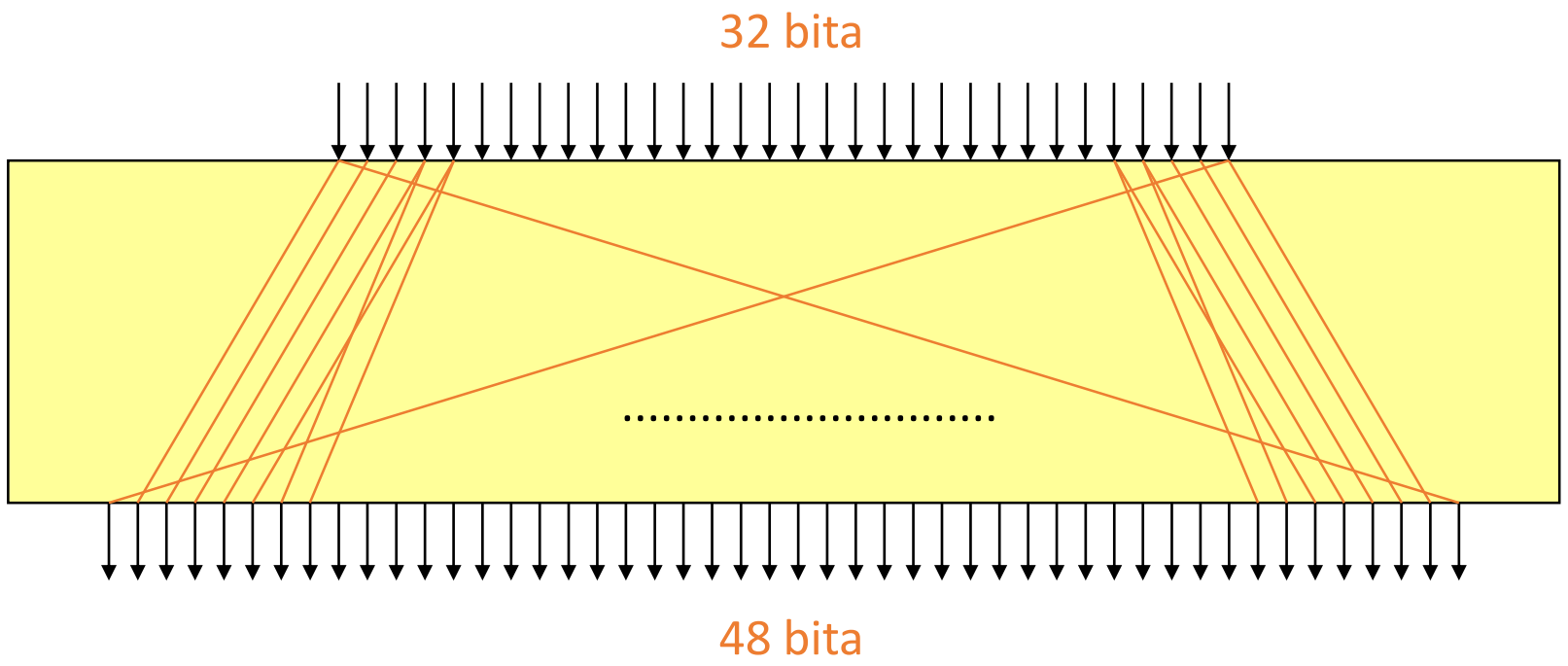
- Šifrovanje.
  - Svaki blok se transformiše u 16 rundi zamena i permutacija (transpozicija).
  - Permutacije unose difuziju podataka, a zamene konfuziju (Shannon).
  - U svakoj rundi se koristi podključ (*subkey*) dužine 48 bita koji se izvodi od ključa.
- Dešifrovanje.
  - Isti proces kao i šifrovanje ali sa podključem u obrnutom redosledu.
- Napomena: inicijalna i inverzna permutacija ne doprinose jačini šifre!

Jedna runda DES-a



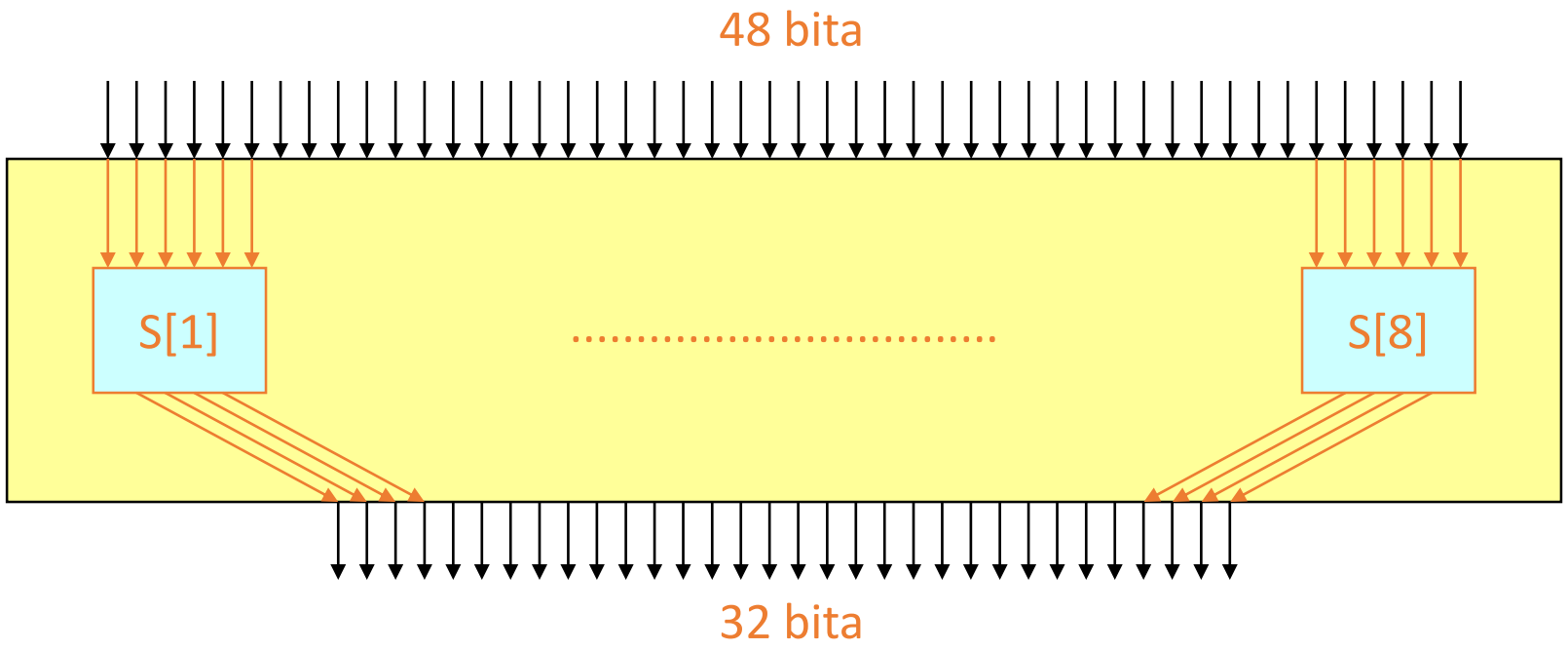
E-boks

- E-boks **proširuje** (*expand*) i **permutuje** ulazne bite (od 32 bita u 48).
- Menja redosled bita, a neke bite ponavlja (potrebno zbog lavinskog efekta).



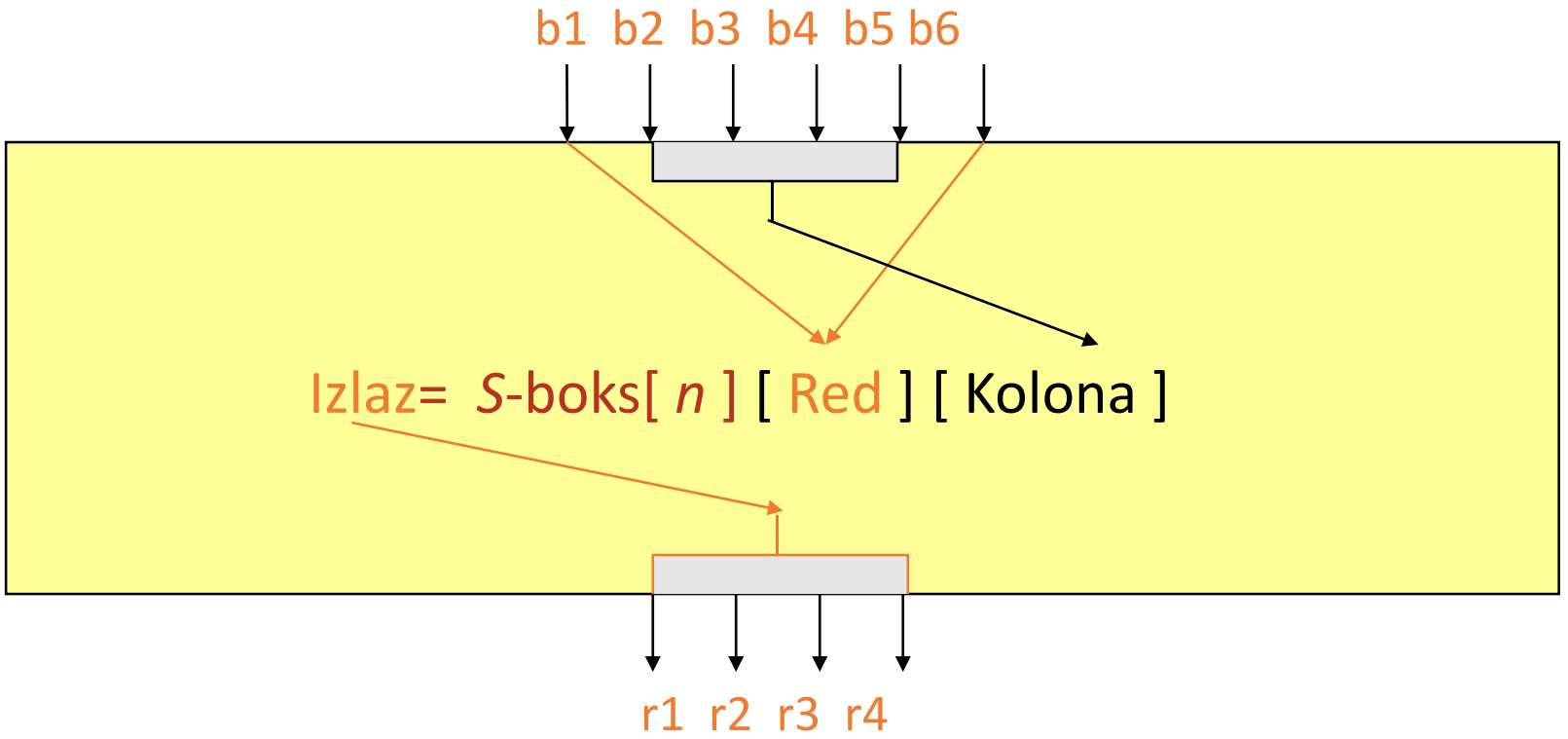
S-boks

- Svaki S-boks od 6 bita ulaza pravi 4 bita izlaza.
- S-boks je osnova sigurnosti DES-a.
  - S-boksovi su nelinearni, teški za analizu.
  - Drugi boksovi su linearni i lakši za analizu.



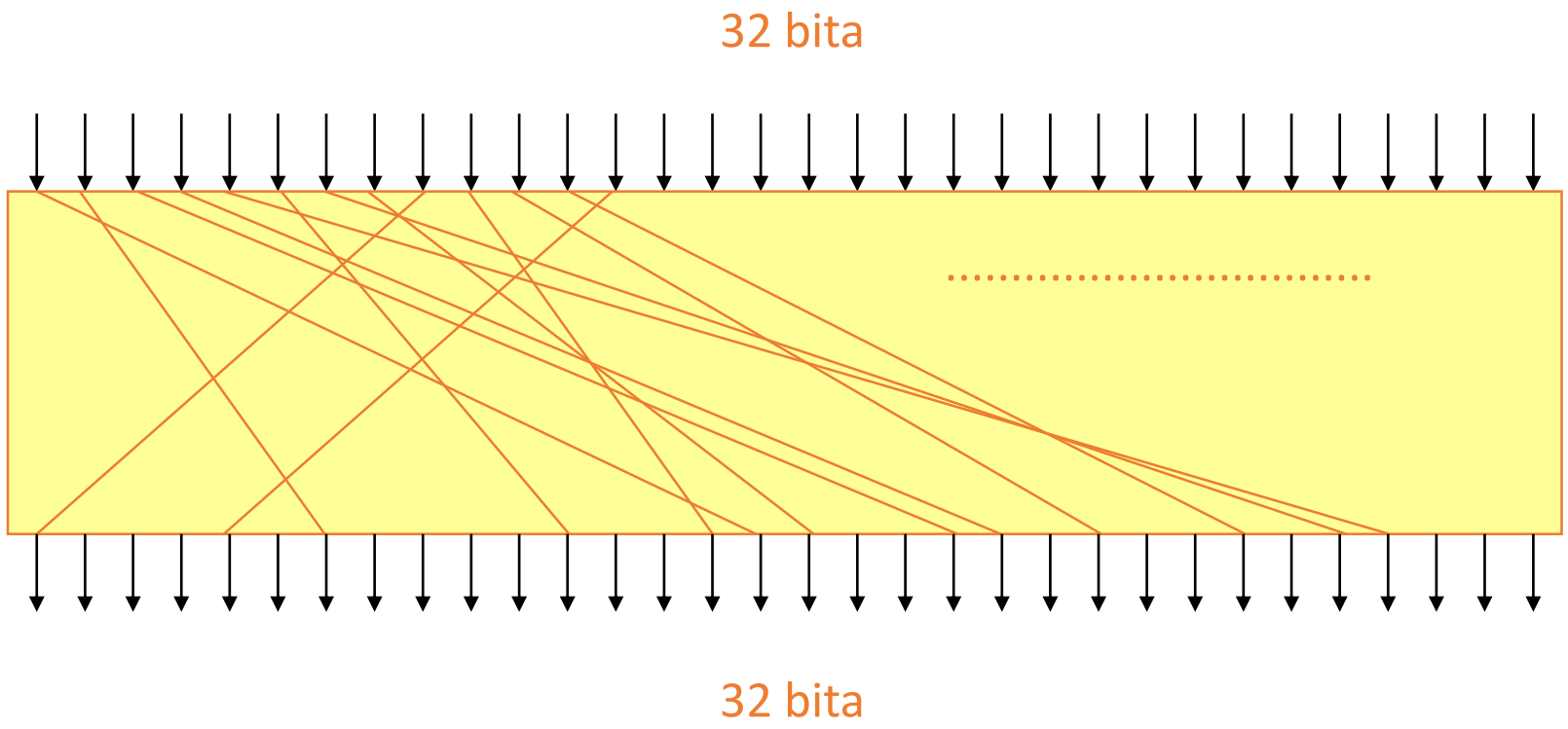
S-boks [n]

- Svaki S-boks ima sopstvenu tabelu zamena.
- Ulaz: spoljašnja 2 bita određuju red tabele, a unutrašnja 4 bita određuju kolonu tabele.
- Izlaz: iz tabele (red, kolona) se čitaju 4 bita.

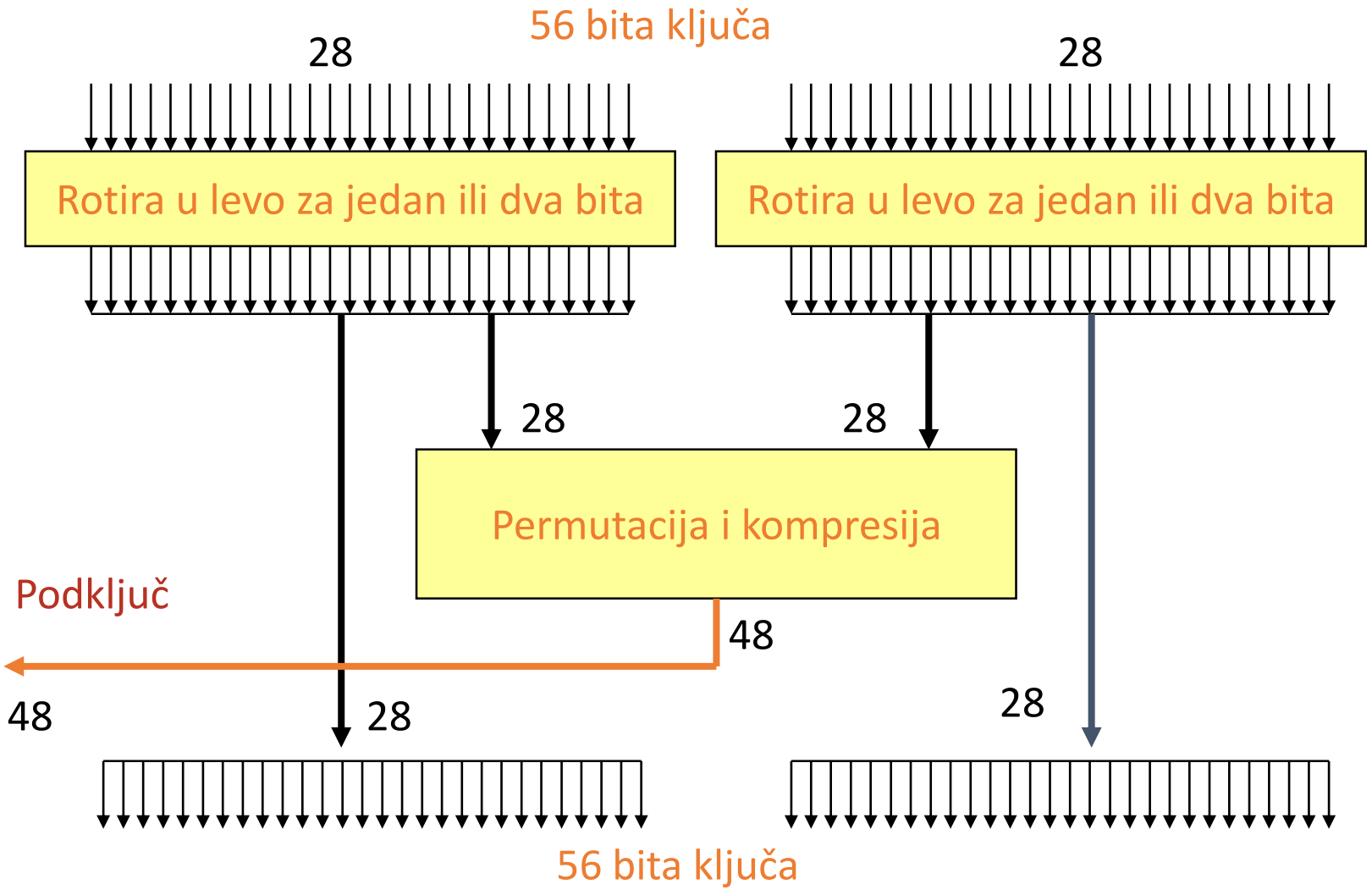


P-boks

- P-box predstavlja permutaciju 32 bita.



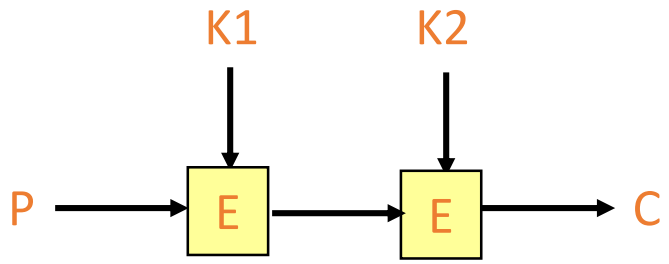
K-boks

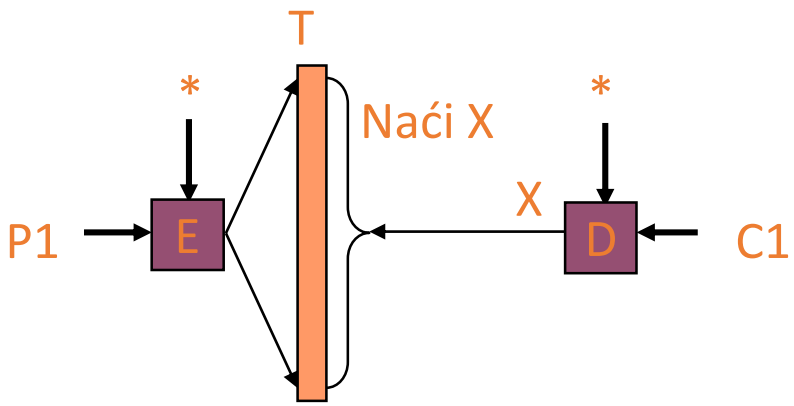

56 bita ključa
28
28
Rotira u levo za jedan ili dva bita
Rotira u levo za jedan ili dva bita
28
28
Permutacija i kompresija
Podključ
48
48
28
28
56 bita ključa

Sigurnost DES algoritma

- Jačina DES-a počiva na S-boksovima.
- Sve ostalo u DES-u je linearno.
- Preko 30 godina intezivne analize nije otkrilo propuste tipa "*back door*".
- Zaključak:
  - Tvorci DES-a su znali šta rade.
  - Bili su ispred svog vremena (otpornost na diferencijalnu kriptoanalizu!)
- Prostor ključeva je 2^{56} ali je danas napad potpunom pretragom ključeva izvodljiv.
- 1993 Michael Wiener je pokazao da je moguće napraviti hardver kojim se može razbiti DES napadom tipa poznati otvoreni tekst:
  - za 35 sati uz budžet od 100.000 $,
  - za 3,5 sata uz budžet od 1 milion $,
  - za 21 min uz budžet od 10 miliona $.
- Neke institucije su tada verovatno već posedovale takav hardver.

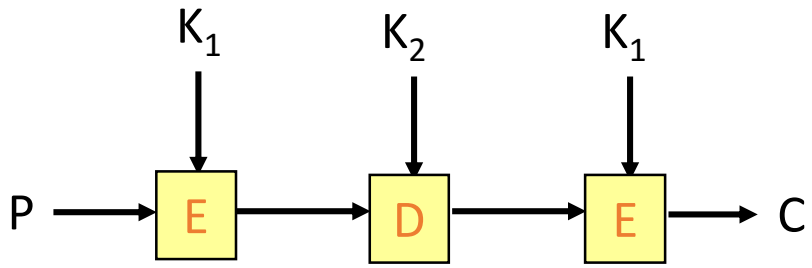
Dvostruki DES





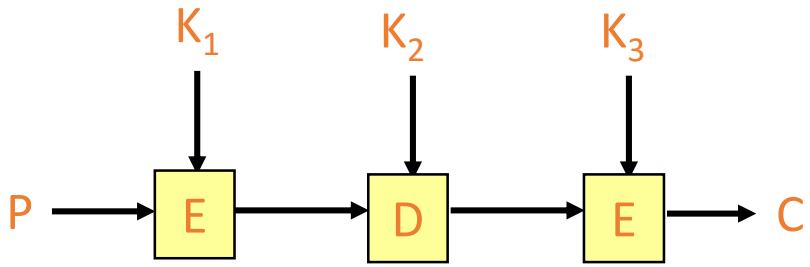
- Šifruje 2 puta sa 2 ključa.
- Ima slabosti.
- Napad: poznati otvoreni tekst
  - Poznat parovi (P1, C1) i (P2, C2).
  - Za svako moguće K1 šifrovati P1.
  - Zapisati rezultate u tebelu T.
  - Pretpostaviti K2, dešifrovati C1 $\rightarrow$ X.
  - Ako X postoji u T, proveri K1 i K2 sa novim parom (P2, C2).
  - Ako je OK, ključevi su pronađeni.
- Smanjuje 2^{112} na 2^{56}, ali T je ogromna!

Trostruki DES



$C = E(D(E(P,K_1), K_2), K_1)$

$P = D(E(D(C,K_1), K_2), K_1)$



- 3DES sa 2 ključa (EDE2)
  - Ključevi se koriste nezavisno.
  - Prostor ključa je 2^{112} (56+56).
  - Druga faza je dešifrovanje jer za K2=K1 postoji kompatibilnost sa običnim DES-om.
  - Primena: PEM (Privacy Enhanced Mail), PGP (ranije verzije), ...
- 3DES sa 3 ključa (EDE3)
  - Neki više vole.
  - Dužina ključa 168 bita (56+56+56).

Advanced Encryption Standard – AES

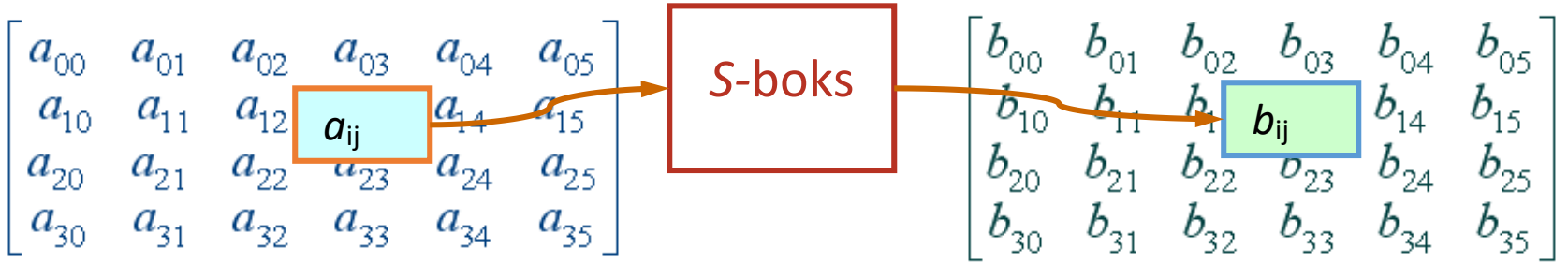
- Zamena za DES.
  - AES takmičenje (krajem 90-tih).
  - NSA je javno uključena u izbor rešenja.
  - Procedura izbora je bila javna.
  - Predložen je veliki broj jakih algoritama.
  - Rijndael algoritam je izabran kao pobednik.
    - Izgovara se slično kao i *Rain Doll*.
- AES je iterativna blokovska šifra (kao i DES).
- AES nije Fejstel šifra (DES jeste).
- AES operacije šifrovanja/dešifrovanja moraju biti invertibilne.

AES osnovni podaci

- Dužina bloka: 128, 192 ili 256 bita.
- Dužina ključa: 128, 192 ili 256 bita.
  - Ne zavisi od dužine bloka!
- Od 10 do 14 rundi (zavisno od dužine ključa).
- U svakoj rundi se koriste 4 funkcije (u 3 “sloja”).
  - ByteSub (nelinearni sloj)
  - ShiftRow (sloj linearnog mešanja)
  - MixColumn (nelinearni sloj)
  - AddRoundKey (dodatni sloj ključa).

AES ByteSub

- Neka je dužina bloka 192 bita.
- AES posmatra podatke kao matricu 4x6 bajtova.
- Funkcija ByteSub je AES-ov „S-boks”.
- Može se posmatrati kao nelinearna (ali invertibilna) kompozicija dve matematičke operacije ili kao tabela.



$$b_{ij} = \mathtt{ByteSub}(a_{ij})$$

AES ByteSub

- Primer: ByteSub (3c) = eb.

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

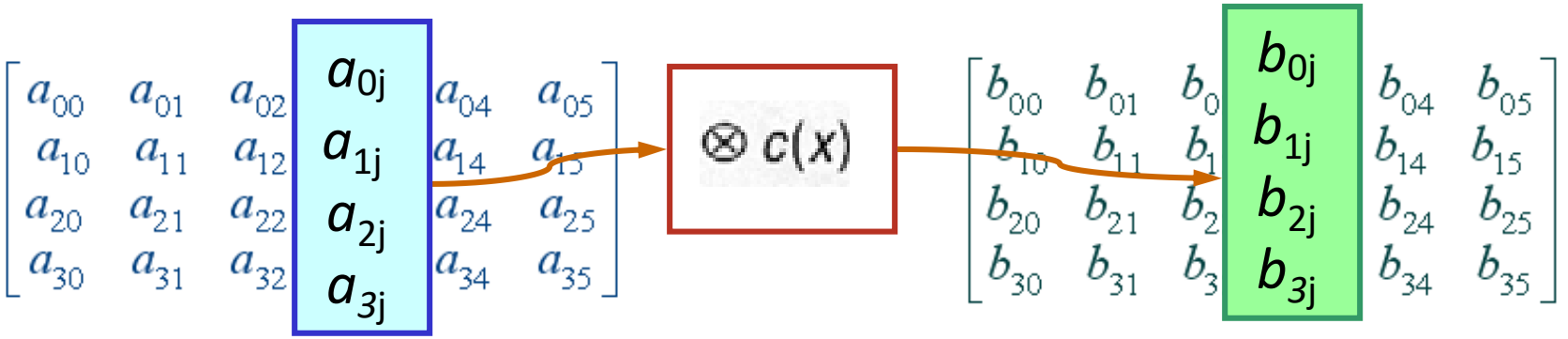
AES ShiftRow

- Ciklično pomeranje poslednja 3 reda matrice za 1, 2 i 3 bajta.

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} & a_{04} & a_{05} \\ a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{20} & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{30} & a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \end{bmatrix} \rightarrow \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} & a_{04} & a_{05} \\ a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{10} \\ a_{22} & a_{23} & a_{24} & a_{25} & a_{20} & a_{21} \\ a_{33} & a_{34} & a_{35} & a_{30} & a_{31} & a_{32} \end{bmatrix}$$

AES MixColumn

- Nelinearna, invertibilna operacija koja se primenjuje na svaku kolonu
- Operacija pomeranja i XOR.
- Implementira se kao tabela.



AES AddRoundKey

- Dodavanje ključa runde.
- XOR podključa sa blokom.
- *RoundKey* (podključ) je određen sa *key schedule* algoritmom.

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} & a_{04} & a_{05} \\ a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{20} & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{30} & a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \end{bmatrix} \oplus \begin{bmatrix} k_{00} & k_{01} & k_{02} & k_{03} & k_{04} & k_{05} \\ k_{10} & k_{11} & k_{12} & k_{13} & k_{14} & k_{15} \\ k_{20} & k_{21} & k_{22} & k_{23} & k_{24} & k_{25} \\ k_{30} & k_{31} & k_{32} & k_{33} & k_{34} & k_{35} \end{bmatrix} = \begin{bmatrix} b_{00} & b_{01} & b_{02} & b_{03} & b_{04} & b_{05} \\ b_{10} & b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\ b_{20} & b_{21} & b_{22} & b_{23} & b_{24} & b_{25} \\ b_{30} & b_{31} & b_{32} & b_{33} & b_{34} & b_{35} \end{bmatrix}$$

Blok Podključ

AES dešifrovanje

- Da bi dešifrovanje bilo moguće proces mora biti **invertibilan**.
  - Inverzija funkcije AddRoundKey je jednostavna jer je operacija XOR istovremeno sopstvena inverzija.
  - Funkcija MixColumn je invertibilna (realizuje se takođe kao tabela).
  - Inverzija funkcije ShiftRow je ciklični pomeraj u suprotnom pravcu.
  - ByteSub je invertibilna (implementirana je kao tabela).

Ostale blokovske šifre

- IDEA
- Blowfish
- RC6
- Serpent
- Mars
- Twofish
- …

Režimi rada blokovskih šifara

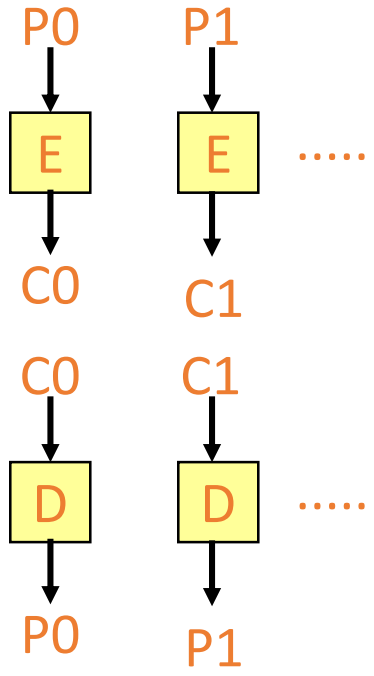
- Otvoreni tekst P može da bude dugačak tako da se mora podeliti na više blokova.
  - Dužina bloka: l.
  - Poruka se deli na na blokove od l bita.
  - Ako je dužina poruke nije umnožak od l, onda se poruka dopunjava.
- Način na koji se primenjuje blokovska šifra na niz uzastopnih blokova se naziva **režim rada**.
- Pitanja:
  - Da li koristiti novi ključ za svaki blok?
    - Veliki broj ključeva!
  - Da li treba šifrovati svaki blok nezavisno?
  - Da li šifrovanje učiniti zavisnim od šifrovanja prethodnih blokova (“ulančati” blokove)?
  - ...

Režimi rada blokovskih šifara

- Od mnogih režima rada ovde će biti predstavljena 3:
  - **Režim elektronske kodne knjige** (*Electronic Codebook – ECB mode*).
    - Svaki blok se šifruje nezavisno.
    - Postoje ozbiljne slabosti.
  - **Režim ulančavanja blokova šifrata** (*Cipher Block Chaining – CBC mode*).
    - Ulančava blokove.
    - Mnogo sigurniji od ECB režima.
  - **Režim brojača** (*Counter mode – CTR mode*)
    - Ponaša se kao sekvencijalna šifra.
    - Primena kod kontrole pristupa.

ECB režim

- $C_n = E(K, P_n)$
- За отворени текст $P_0, P_1, \ldots, P_m, \ldots$
  - Шифровање
    - $C_0 = E(P_0, K)$
    - $C_1 = E(P_1, K)$
    - $C_2 = E(P_2, K)$
    - …
  - Дешифровање
    - $P_0 = D(C_0, K)$
    - $P_1 = D(C_1, K)$
    - $P_2 = D(C_2, K)$
    - …
- Za fiksan ključ K, ovo je elektronska verzija šifre tipa kodna knjiga.
- Za svaki novi ključ dobija se nova kodna knjiga.

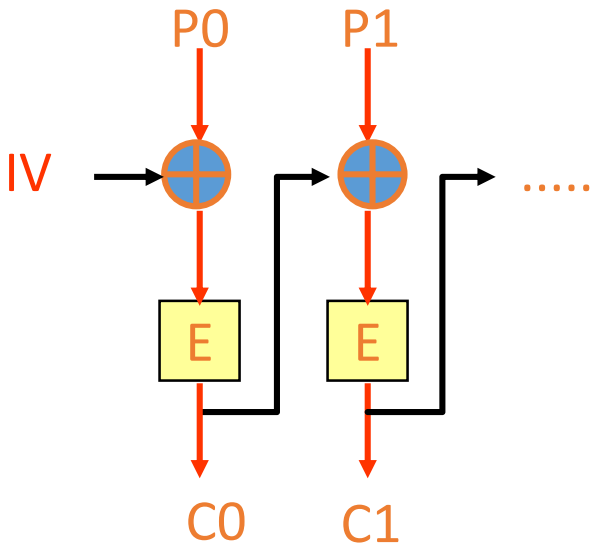


ECB "*cut and paste*" napad

- Neka je otvoreni tekst (Alisa): "Alice digs Bob. Trudy digs Tom."
- Ako je blok od 64 bita i primenjeno 8-bitsko ASCII kodovanje:
  - P_0 = "Alice di", P_1 = "gs Bob. ",
  - P_2 = "Trudy di", P_3 = "gs Tom. "
- Šifrat: C_0, C_1, C_2, C_3
- Trudi seče i kopira: C_0, C_3, C_2, C_1
- Bob dešifruje kao: "Alice digs Tom. Trudy digs Bob."
- Integritet?

CBC režim

- Blokovi se međusobno “ulančavaju”.
- Potreban je inicijalizacioni vektor IV, za inicijalizaciju CBC režima.
- IV treba da je slučajan, nije neophodno da je tajan.
- Šifrovanje:
  - $C_0 = E(IV \oplus P_0, K)$
  - $C_1 = E(C_0 \oplus P_1, K)$
  - $C_2 = E(C_1 \oplus P_2, K)$
  - …
- Dešifrovanje:
  - $P_0 = IV \oplus D(C_0, K)$
  - $P_1 = C_0 \oplus D(C_1, K)$
  - $P_2 = C_1 \oplus D(C_2, K)$
  - …
- Isti blokovi otvorenog teksta u opštem slučaju daju različite blokove šifrata!

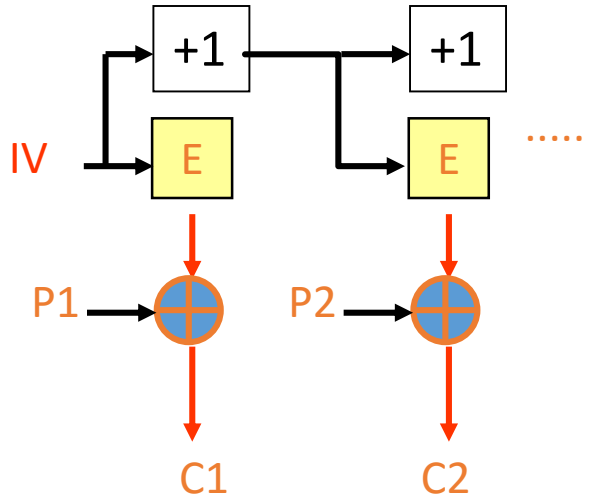


CBC režim

- Identični blokovi otvorenog teksta daju različite blokove šifrata!
- Napad tipa "*cut and paste*" je i dalje moguć ali je daleko složeniji (posledica su greške koje su očigledne nakon dešifrovanja).
- Ako je blok C_1 zamenjen sa lažnim blokom G, onda:
  - $P_1 \neq C_0 \oplus D(G, K)$
  - $P_2 \neq G \oplus D(C_2, K)$
- Ali:
  - $P_3 = C_2 \oplus D(C_3, K)$
  - $P_4 = C_3 \oplus D(C_4, K)$
- Automatski se oslobađa greška!

CTR režim

- CTR je popularan režim rada.
- Koristi se blokovska šifra kao da je u pitanju sekvencijalna šifra.
- IV se šifruje a potom XOR-uje sa blokom otvorenog teksta.
- Za svaki naredni blok otvorenog teksta vrednost IV se inkrementira za 1.
- Šifrovanje:
  - $C_0 = P_0 \oplus E(IV, K)$,
  - $C_1 = P_1 \oplus E(IV+1, K)$
  - $C_2 = P_2 \oplus E(IV+2, K)$
  - …
- Dešifrovanje:
  - $P_0 = C_0 \oplus E(IV, K)$
  - $P_1 = C_1 \oplus E(IV+1, K)$
  - $P_2 = C_2 \oplus E(IV+2, K)$
  - …

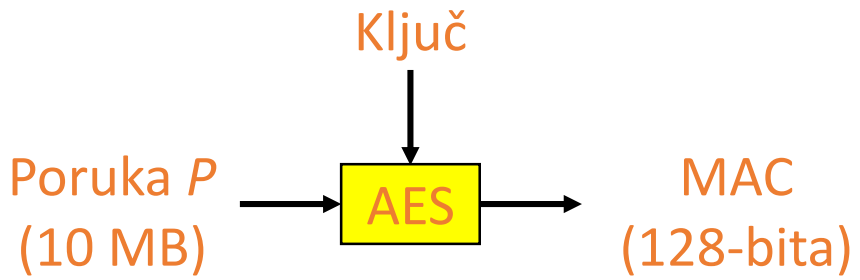


Integritet podataka

- Integritet – **sprečavanje** (ili barem **otkrivanje**) **neovlašćene izmene podataka**.
  - Primer: elektronski prenos novca sa računa A na B.
- Poverljivost je poželjna, integritet je neophodan!
- Šifrovanje može da obezbedi **poverljivost**.
  - Neovlašćenim stranama su podaci nerazumljivi.
- Šifrovanje, u osnovnoj primeni, **ne obezbeđuje integritet podataka**.
- Ako Trudi uspe da izmeni šifrat (bez obzira što ne zna otvoreni tekst) pre nego što stigne do Boba ili ako dođe do greške u prenosu, **narušiće se integritet podataka**.
  - Primer: "*cut and paste*" napad na ECB.

Message Authentication Code (MAC)

- Odnosi se na integritet podataka.
- MAC se računa na osnovu dela CBC.
- Treba primeniti CBC šifrovanje i zapamtiti samo šifrat poslednjeg bloka.



- Napomena:
  - Kod primene mehanizma integriteta, nije neophodno, mada je moguće, da se poruka šalje šifrovano.
  - Najčešće se zahteva primena oba mehanizma (poverljivost i integritet).

Message Authentication Code (MAC)

- Kako se izračunava?
  - Neka postoji N blokova otvorenog teksta:
    - $C_0 = E\,(IV \oplus P_0, K)$,
      $C_1 = E\,(C_0 \oplus P_1, K)$,
    - $C_2 = E\,(C_1 \oplus P_2, K)$,…
    - $C_{N-1} = E\,(C_{N-2} \oplus P_{N-1}, K) =$ ***MAC***
- MAC se šalje zajedno sa otvorenim tekstom.
- Primalac:
  - Mora da zna (tajni) ključ K i javnu vrednost IV.
  - Računa MAC primljenog otvorenog teksta na isti način i proverava da li je dobijeni rezultat isti sa onim koji je dobio kao MAC.

Message Authentication Code (MAC)

- Primer:
  - Pretpostavimo da Alisa računa:
    - $C_0 = E\,(IV \oplus P_0, K)$, $C_1 = E\,(C_0 \oplus P_1, K)$,
    - $C_2 = E\,(C_1 \oplus P_2, K)$, $C_3 = E\,(C_2 \oplus P_3, K) = MAC$
  - Alisa šalje Bobu: IV, P_0, P_1, P_2, P_3 и MAC .
  - Trudi presreće poruku i menja P_1 u neko X .
  - Bob računa:
    - $C_0 = E\,(IV \oplus P_0, K)$, $\boldsymbol{C_1} = E\,(C_0 \oplus X, K)$,
    - $\boldsymbol{C_2} = E\,(\boldsymbol{C_1} \oplus P_2, K)$, $\boldsymbol{C_3} = E\,(\boldsymbol{C_2} \oplus P_3, K) = \boldsymbol{MAC} \neq MAC$
  - Postoji propagacija greške koja daje novi MAC.
  - Trudi ne može da lažira MAC bez poznavanja ključa K!

Poverljivost i integritet

- Šifruje se **jednim ključem**, MAC se računa pomoću **drugog ključa**.
- Zašto se ne koristi isti ključ, bilo bi efikasnije?
  - Poslednji paket (MAC) bi bio ponovljen 2 puta?
  - Ne bi se dobilo na povećanju sigurnosti!
- Upotreba različitih ključeva za šifrovanje i računanje MAC znači dupli posao.
- Objedinjavanje ova dva postupka je predmet istraživanja.
- Računanje MAC na osnovu CBC nije jedini način obezbeđivanje integriteta.

Simetrični kriptosistemi – problemi

- **Skalabilnost**.
  - Za ostavarivanje komunikacije između N osoba potrebno je obezbediti N(N-1)/2 različitih ključeva.
  - Sa porastom N ovo postaje veliki broj.
- **Upravljanje ključevima**.
  - Distribucija ključeva.
  - Čuvanje i pravljenje rezervnih kopija ključeva.
  - Uništavanje ključeva.
  - Zamena ključeva.

Literatura

1. M. Stamp: *Information Security*. John Wiley and Sons.
2. M. Veinović, S. Adamović: Kriptologija 1. Univerzitet Singidunum, Beograd. *

* Može se besplatno preuzeti sa portala: www.singipedia.com

Hvala na pažnji

Pitanja su dobrodošla.