Simetrični algoritmi enkripcije

Blok algoritmi, DES

dr Slavica Tomović

Univerzitet Crne Gore

Enkripcija toka

Tok podataka se enkriptuje bit po bit ili bajt po bajt

Primjeri:

- Autokey Vigenère algoritam
- Vernam algoritam

U idealnom slučaju koristi se *one-time pad* enkripcija, pri čemu je dužina toka ključeva (engl. *keystream*) jednaka dužini toka informacije

Ukoliko je tok ključeva generisan na slučajan način onda je nemoguće razbiti šifru bez tačnog poznavanja toka ključeva

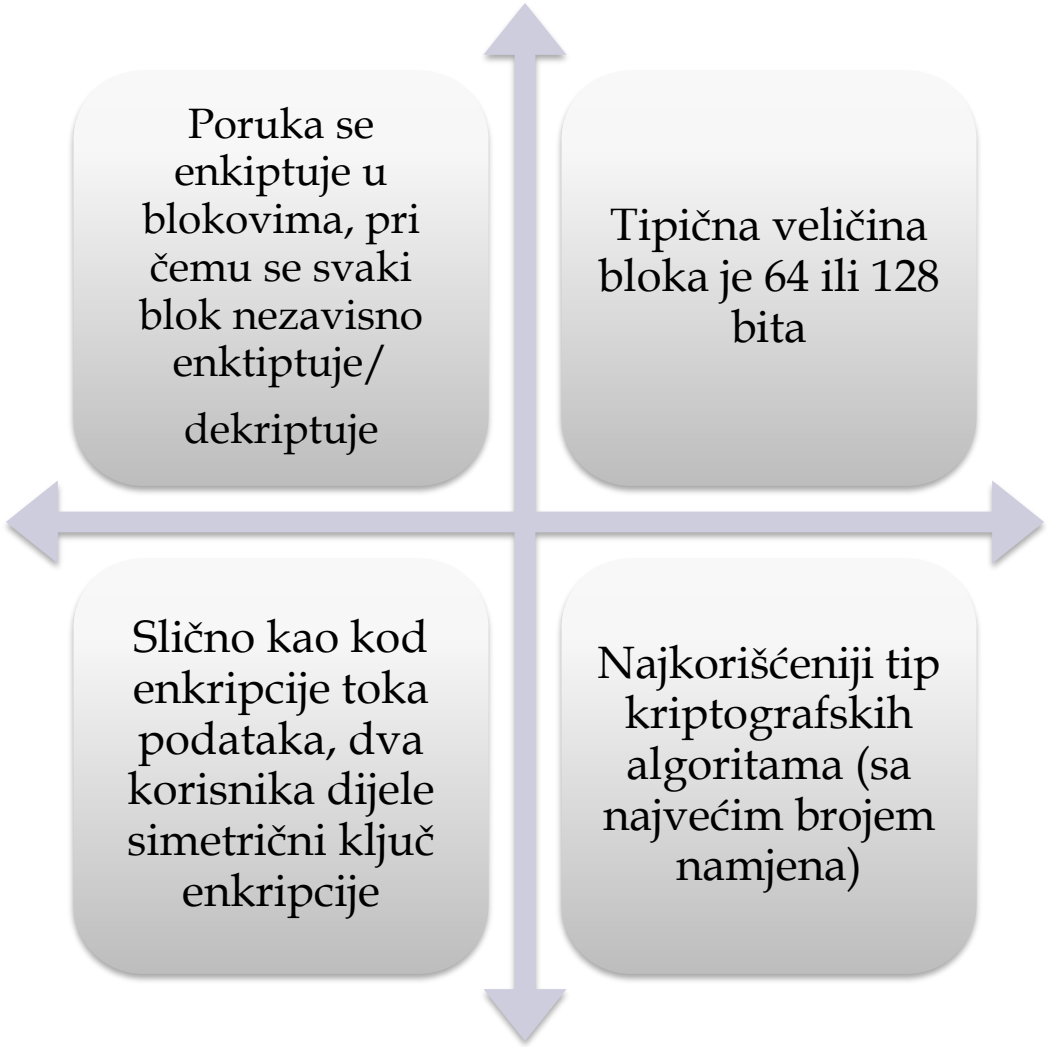
- Tok ključeva se mora učiniti dostupnim i pošiljaocu i primaocu putem sigurnog i nezavisnog kanala komunikacije
- Ovo je veoma problematično realizovati ukoliko je intezitet saobraćaja veliki

Iz praktičnih razloga generator toka ključeva mora biti implementiran kao algoritamska procedura, tako da oba korisnika mogu proizvesti kriptografski tok bita na osnovu generatorskog ključa

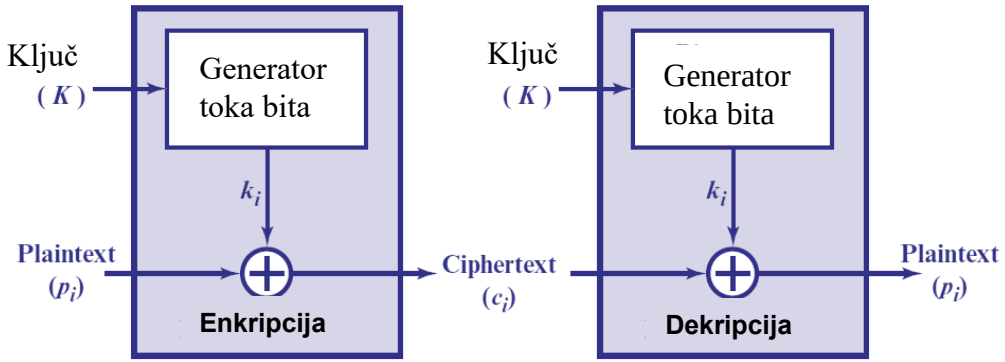
Mora biti računski nepraktično predviđati buduće djelove toka bita na osnovu prethodnih djelova toka

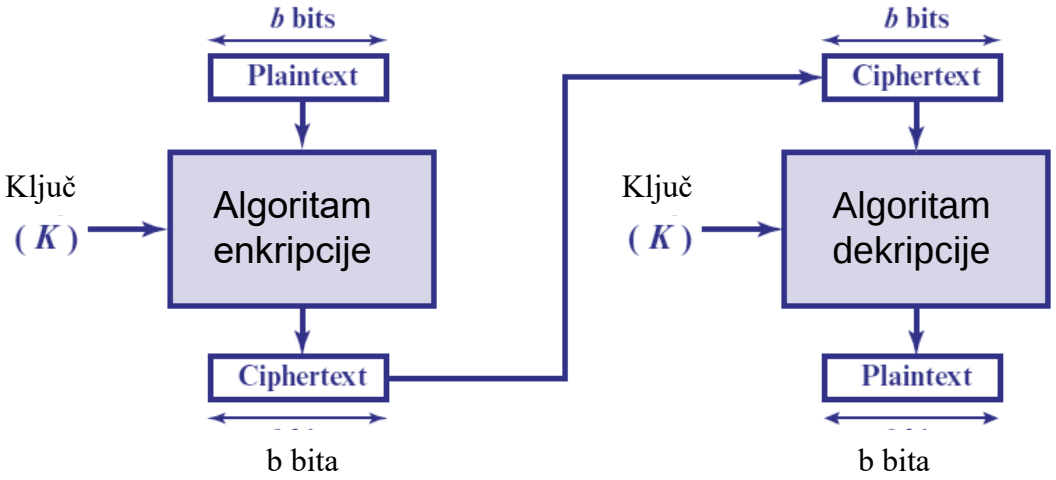
Korisnici moraju dijeliti samo generatoski kluč, na osnovu kojeg oba mogu proizvesti identičan tok ključeva

Blok enkripcija



Enkripcija toka i enkripcija bloka podataka





Blok enkripcija

- *Plaintext* od n bita poruke enkriptuje se u *chiphertext* od n bita
- Blok algoritmi liče na izuzetno velike supstitucione algoritme
  - potrebna tabela sa 2^{64} ulaza za blok od 64 bita
- Svaki od 2^n *plaintext* blokova mora biti mapiran u jedinsveni enkriptovani blok da bi dekripcija bila moguća
  - Ovakav vid enkripcije naziva se reverzibilnim
  - Kod nereverzibilne enkripcije istu kodnu riječ može proizvesti više razlitih *plaintext* blokova

Reverzibilno mapiranje

Plaintext	Ciphertext
00	11
01	10
10	00
11	01

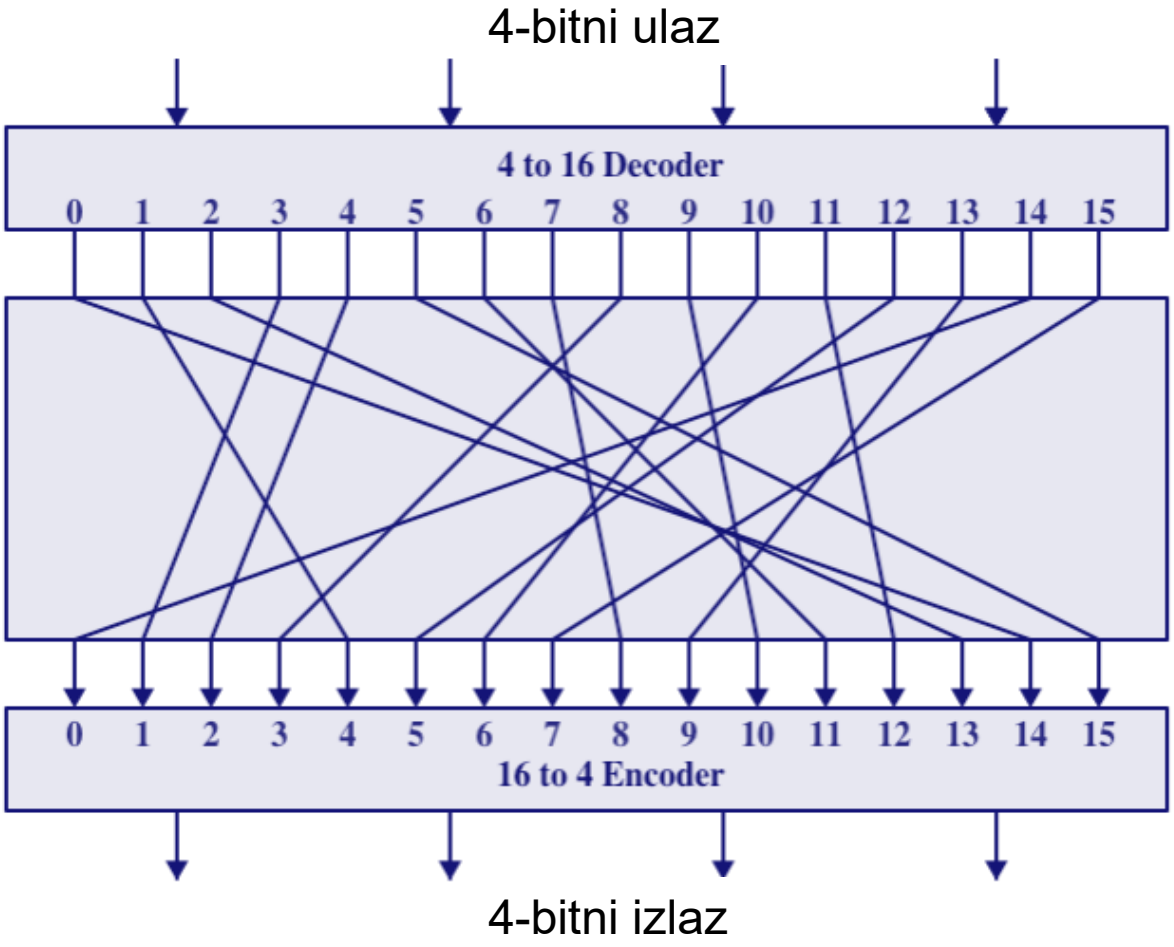
Nereverzibilno mapiranje

Plaintext	Ciphertext
00	11
01	10
10	01
11	01

Principi blok algoritama

- Ukoliko se ograničimo na revezibilnu enkripciju, broj različtih transformcija je 2^n!
- Ukliko se koristi mala veličina bloka, algoritam je ranjiv kao i klasični supstitucioni algoritmi
  - Ovi algoritmi su podložni napadima statističkom analizom
- Ukoliko je veličina bloka dovoljno velika, statističke karakteristike bloka se uspješno maskiraju
- Previše veliki blokovi međutim nisu praktični iz ugla implementacije i performansi
  - Ako uzmemo da izabrano mapiranje predstavlja ključ, za blok od 4 bita imali bismo ključ velikine 4 x 16 = 64 bita
  - Uopšteno, za blok od n bita imali bismo ključ od $n \cdot 2^n$ bita
  - Za blok od 64 bita, što je poželjna veličina zbog napada statističkom analizom, imamo ključ veličine $\approx 10^{21}$ bita.

Supstitucioni blok šifrator za n=4



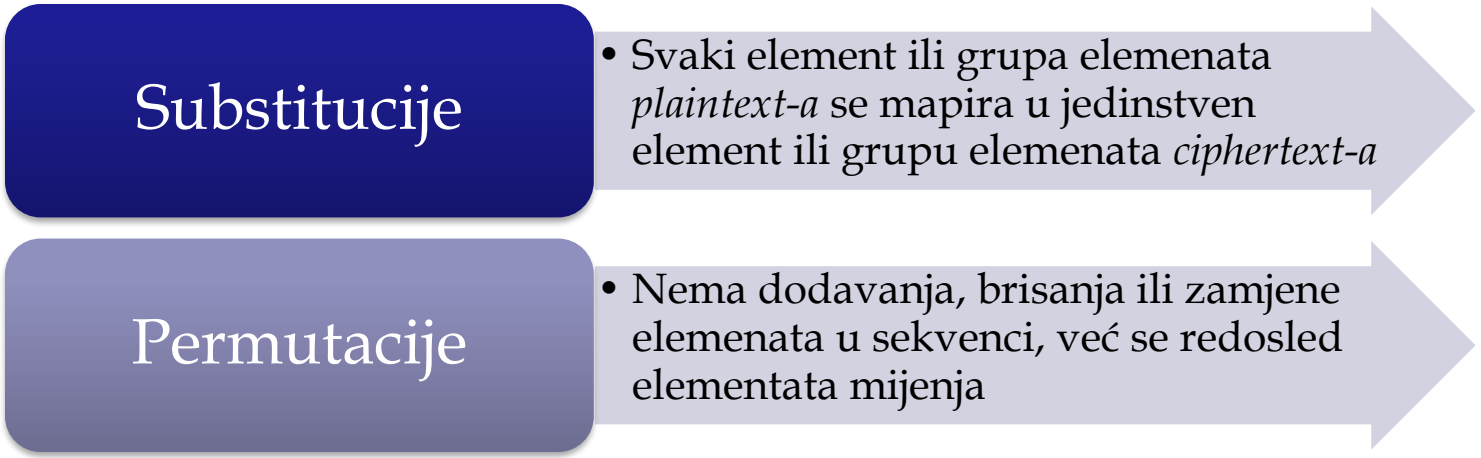
Tabele za enkripciju i dekripciju

Plaintext	Ciphertext
0000	1110
0001	0100
0010	1101
0011	0001
0100	0010
0101	1111
0110	1011
0111	1000
1000	0011
1001	1010
1010	0110
1011	1100
1100	0101
1101	1001
1110	0000
1111	0111

Ciphertext	Plaintext
0000	1110
0001	0011
0010	0100
0011	1000
0100	0001
0101	1100
0110	1010
0111	1111
1000	0111
1001	1101
1010	1001
1011	0110
1100	1011
1101	0010
1110	0000
1111	0101

Feistel algoritam

- Feistel je predložio šifrator koji naizmjenično primjenjuje supstitucije i permutacije.



- Praktična realizacija produkt šifratora koji omogućava *konfuziju* i *difuziju* (ideja Kloda Šenona).
- Većina simetričnih blok algoritama bazirana je na ovoj strukturi.

Konfuzija i difuzija

- Termini koje je uveo Klod Šenon za dva osnovna gradivna bloka svakog kriptografskog sistema
  - Šenon je težio tome da enkriptvana poruka bude lišena bilo kakvih statističkih svojstava originalne poruke.

Difuzija

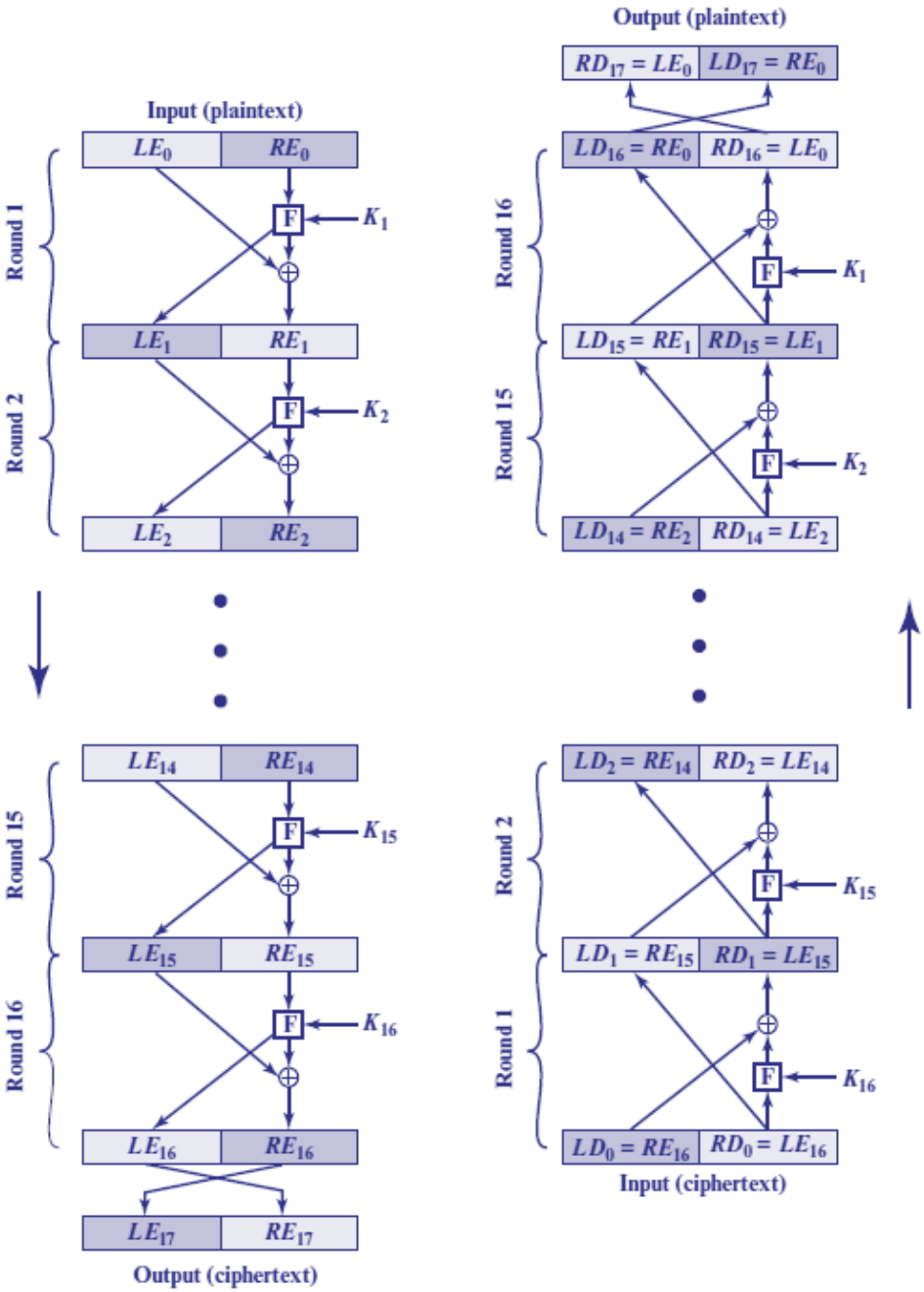
- Neutrališe statističke strukture originalne poruke u enkriptovanoj poruci
- Postiže se tako što svaki bit (ili slovo) originalne poruke utiče na mnogo bita (ili slova) enktiptovane poruke.
- Upotreba nekoliko permutacija za redom, uz primjenu određene funkcije za permutaciju nakon svake permutacije

Konfuzija

- Stvara što komplikovaniju vezu između enkriptovane poruke i ključa korišćenog za enkripciju
- Postiže se korićenjem složenog algoritma supstitucije.

Struktura Feistel algoritma

- Dijeli ulazni blok na dvije polovine:
  - Obrađuje polovine bloka u više iteracija (obično 16).
  - Izvršava algoritam supstitucije za lijevu polovinu podataka na osnovu funkcije iteracije desne polovne i potključa.
  - Na kraju se permutacijom zamijene polovine.



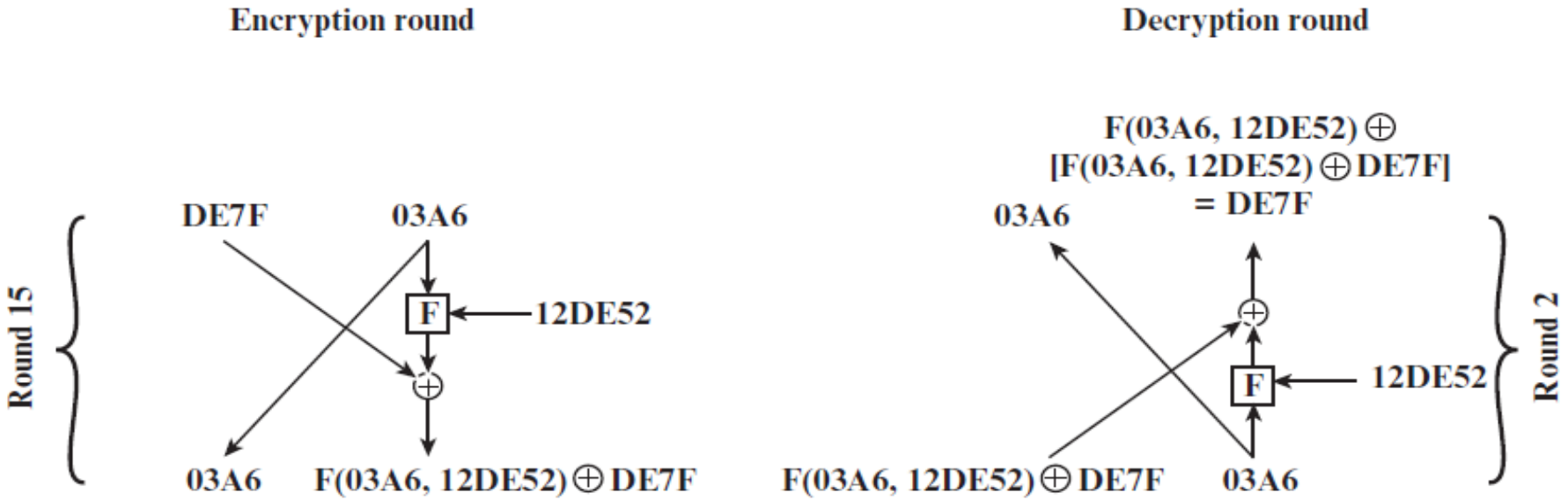
Enkripcija i dekripcija Feistel algoritmom

Principi dizajna Feistel algoritma

- Veličina bloka
  - Sa povećanjem veličine bloka se poboljšava sigurnost ali se usporava algoritam enkripcije/dekripcije
- Veličina ključa
  - Veći ključevi poboljšavaju sigurnost, otežavajući pretraživanje svih ključeva, ali mogu usporiti enkripciju/dekripciju
- Broj iteracija
  - Ključna ideja Feistel šifratora je da jedna iteracija ne nudi dovoljnu zaštitu, ali da se sa većim brojem iteracija značajno poboljšava bezbjednost na račun vremena izvršavanja algoritma
- Algoritam za generisanje potključa
  - Veća kompleknost ovog algoritma otežava kriptoanalizu
- Funkcija iteracije ***F***
  - Veća kompleknost generalno povlači i veću robustnost na kriptoanalizu
- Brzina softvera za enkripciju/dekripciju
  - U velikom broju slučajeva enkripcija se vrši softverski, kao ugrađena funkcija aplikacije. Stoga, brzina izvršavanja algoritma jedan je od velikih izazova implementacije
- Jednostavnost analize
  - Ako se algoritam može sažeto i jasno objasniti lakše je analizirati njegove sigurnosne ranjivosti i samim tim razviti veći nivo pouzdanosti

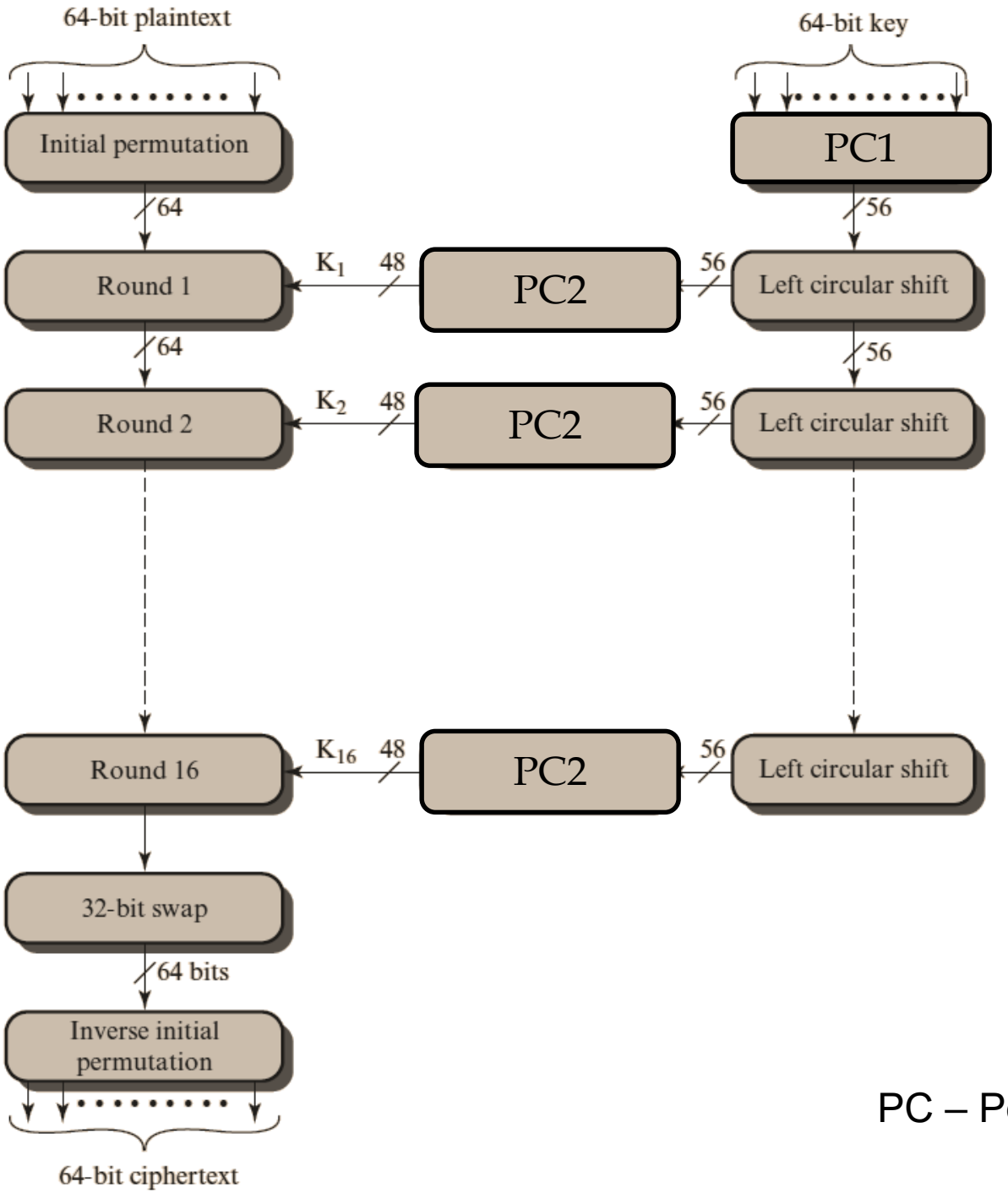
Feistel algoritam dekripcije

- Identičan postupku enkripcije
- Ulazni argumenti u prvoj iteraciji su *ciphertext* i potključ K_n
- Potključevi se koriste po obrnutom redosledu u odnosu na postupak enkripcije



DES

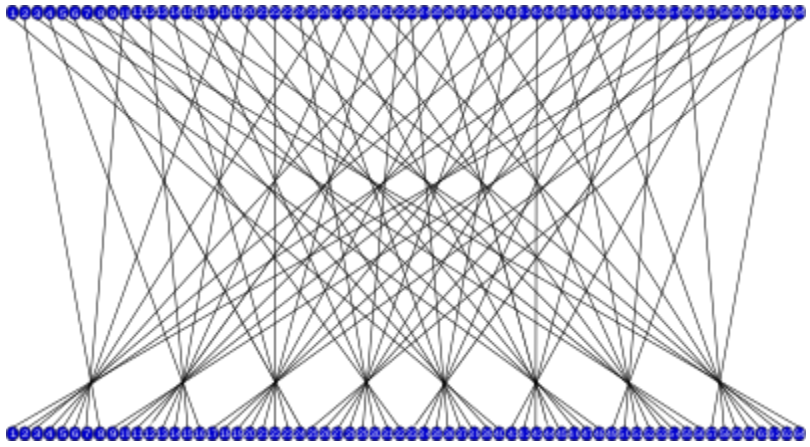
- Razvijen u sklopu istraživanja Američke državne agencije za stanarde (1977) o potrebama raznih državnih tijela za šifrovanjem dokumenata i komunikacija koje se ne smatraju državnom ili vojnom tajnom, ali ne bi smjele biti dostupne široj javnosti.
- Najrasprostranjeniji blok algoritam širom svijeta do uvođenja AES (*Advanced Encryption Standard*) standarda 2001.
- Značajna uloga u finansijskim aplikacijama
- Sami algoritam poznat je pod nazivom *Data Encryption Algorithm* (DEA)
  - Podaci se enkriptujuu u blokovima veličine 64 bita, koristeći ključ veličine 56 bita.
  - Algoritam transformiše 64-bitni ulaz u 64-bitni izlaz kroz seriju transformacija.
  - Isti koraci, sa istim ključem, koriste se za dekripciju.



PC – Permuted choice

Početna DES permutacija

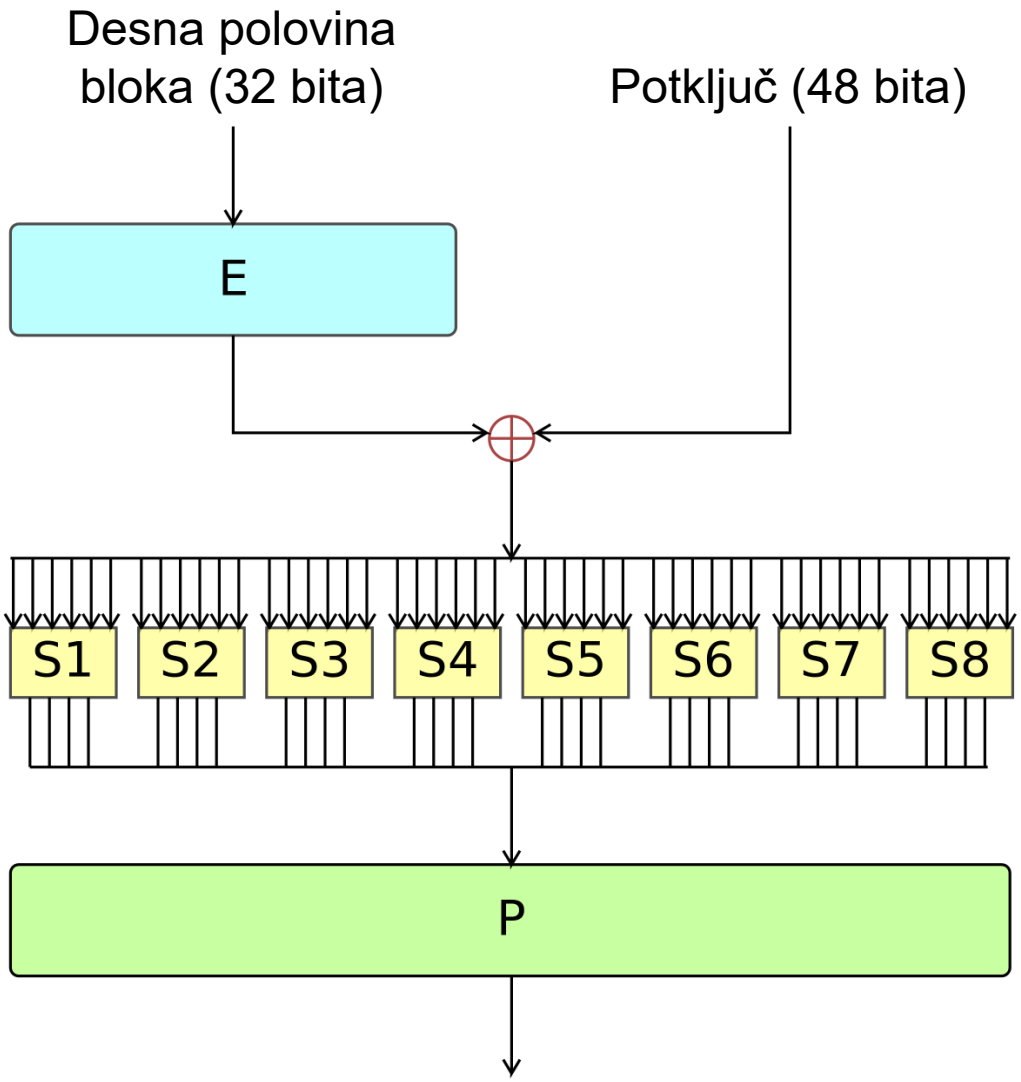
- *Initial Permutation* (IP)
- Prvi korak u algoritmu
- IP preraspodjeljuje ulazni niz bita
- Parni biti idu u lijevu polovinu, neparni u desnu polovinu, prema definisanoj tablici



IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

DES struktura iteracije



DES strukura iteracije

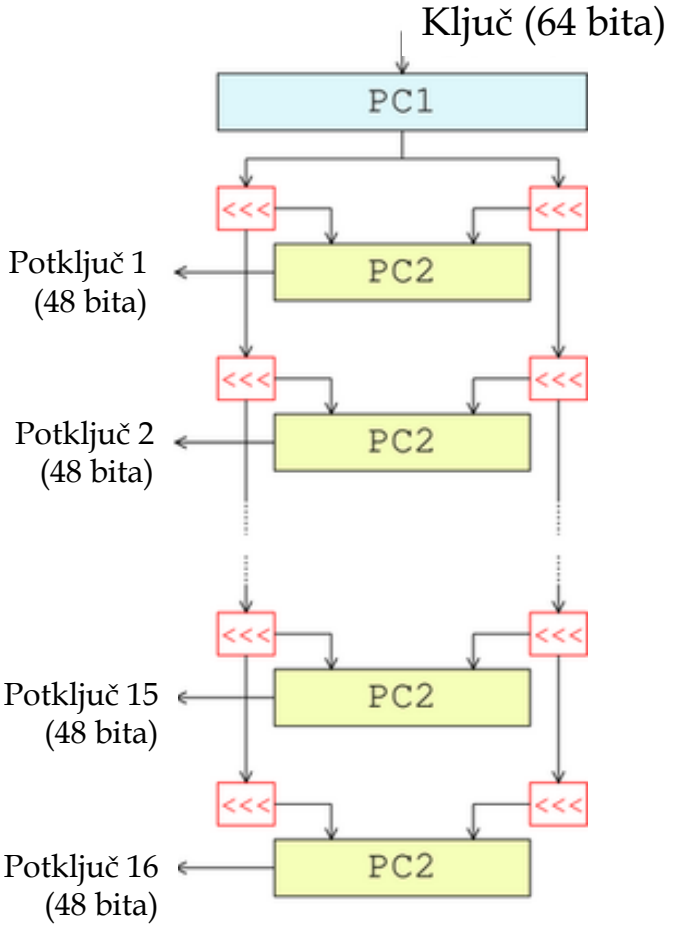
- Koristi dvije 32-bitne polovine (L i R)
- Kao i kod drugih Feistel algoritama važi relacija:
  - $L_i = R_{i-1}$
  - $R_i = L_{i-1}$ $\boldsymbol{XOR}$ $F(R_{i-1}, K_i)$
- Kombinuje 32 bita desne polovine (R) i 48 bita potključa na sledeći način:
  - proširuje R na 48 bita koristeći permutaciju E
  - dodaje ove bite potključu (Supstitucija)
  - prolazi kroz 8 zamjena (S-boxes) da bi se dobio 32-bitni rezultat (S&P)
  - Vrši 32-bitnu permutaciju (P) na kraju

Operacije DES modula

- Perumutacije
  - PC_1 i PC_2 za generisanje potključa, IP, IP^{-1}, E i P
- Supstitucija
  - Koristi se 8 supstitucija (*S-boxes*);
  - Svaki S-box za 6-bitni ulaz daje 4-bitni izlaz
  - Ukoliko posmatraamo 48-bitni ulazni strim bita, prvih 6 bita su ulaz prvog *S-box*-a, drugih 6 bita su ulaz drugog *S-box*-a…
  - Nelinearan korak
- Sabiranje po modulu 2
- 32-bitni registri
  - Koriste se samo za skladištenje podataka; Prilikom generisanja ključa koriste se dva pomjeračka registra koji ciklično pomjeraju

DES generisanje ključeva

- Kreće se od ključa dužine 64 bita, ali svaki osmi bit (*parity check*) se ignoriše, pa zato kažemo da ključ ima 56 bita
- Cilj: na osnovu 56-bitnog ključa generisati potključeve za 16 iteracija
  - PC1 uklanja 8 *parity* bita iz 64-bitnog ulaza
  - 56-bitni izlaz PC1 perumutacije se dijeli u dva pomjeračka registra C i D
  - Sadržaj pomjeračkih registara se u svakoj iteraciji pomjera za jedno ili dva mjesta ulijevo, na osnovu **tabele rotacije ključa**
  - PC2 ignoriše određene ulazne bite, vrši permutaciju i generiše 48-bitni potključ



DES dekripcija

- Prolaze se koraci unazad u odnosu na šifrovanje
- Pomoću Feistel algoritma ponovi se postupak šifrovanja
- Potljučevi se koriste po obrnutom redosledu ($K_{16}, K_{15}, \ldots, K_1$)
- Potključevi se generišu tako što se sadržaj pomjeračkih registara pomjera udesno
- Iza IP se krije konačno rešenje
  - 1. iteracija dekripcije, sa ključem K_{16}, otkriva 16. iteraciju enkripcije
  - ….
  - 16. iteracija dekripcije, sa ključem K_1, otkriva prvu iteraciju enkripcije
  - Finalna permutacija (FP) kod dekripcije otriva inicijalnu permutaciju (IP) enkripcije

Efekat lavine

- Poželjno svojstvo kriptografskog algortima
- Promjena samo jednog bita u ulaznoj poruci ili ključu uzrokuje promjenu otprilike polovine bita na izlazu
- Ovo čini otkrivanje šifre pogađanjem koda praktično nemogućim
- DES predstavlja jaku lavinu

Sigurnost DES-a

- 56-bitni ključevi imaju $2^{56} = 7.2x10^{16}$ vrijednosti
- Brute-force pretraga je teška, ali moguća sa napretkom tehnologije
  - 1997. par mjeseci
  - 1998. za nekoliko dana
  - 1999. za 22 sata

Prosječno vrijeme potrebno za brute-force pretragu

Key Size (bits)	Cipher	Number of Alternative Keys	Time Required at 10^9 Decryptions/s	Time Required at 10^{13} Decryptions/s
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	2^{55} ns = 1.125 years	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	2^{127} ns = 5.3×10^{21} years	5.3×10^{17} years
168	Triple DES	$2^{168} \approx 3.7 \times 10^{50}$	2^{167} ns = 5.8×10^{33} years	5.8×10^{29} years
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	2^{191} ns = 9.8×10^{40} years	9.8×10^{36} years
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	2^{255} ns = 1.8×10^{60} years	1.8×10^{56} years
26 characters (permutation)	Monoalphabetic	$2! = 4 \times 10^{26}$	2×10^{26} ns = 6.3×10^{9} years	6.3×10^{6} years

Kriptoanaliza DES-a

- Mogući napadi:
  - diferencijalna kriptoanaliza
    - Osnovna ideja je poređenje XOR-a od dva otvorena teksta sa XOR-om od odgovarajuća dva šifrata.
  - linearna kriptoanaliza
    - Ideja se sastoji u tome da iako bitovi ključa nisu linearne funkcije otvorenog teksta i šifrata, neki se bitovi ključa mogu dobro aproksimirati linearnom funkcijom.
    - Potreban veliki broj *plaintext-ciphertext* parova.
  - EFF-ov DES Cracker.
- Prva dva napada još uviijek nisu dovela do razbijanja DES-a
  - Njihova je važnost u tome što su primjenjivi na bilo koji simetrični blokovni kriptosistem
  - Kod većine mogućih naslednika DES-a operacije i broj iteracija odabrani su upravo tako da se dobije kriptosistem što otporniji na diferencijalnu i linearnu kriptoanalizu.

Kriptoanaliza DES-a

- Pomoću linearne kriptoanalize Matsui je opisao napad "poznati otvoreni tekst" (eng. *known plaintext*) koji za razbijanje DES-a treba u prosjeku 2^{43} otvorenih tekstova.
  - Ovaj napad je implementiran pomoću 12 radnih stanica i za otkrivanje ključa je trebalo 50 dana.
- Ni diferencijalna ni linearna kriptoanaliza nisu razbile DES, već su to učinili brzi i jeftini čipovi.
  - Pokazalo se da je u praksi lakše napraviti napad "grubom silom" sa 2^{55} DES operacija, nego primijeniti napad linearnom kriptoanalizom koji zahtjeva 2^{43} poznatih parova otvoreni tekst-šifrat.

Kriptoanaliza DES-a

- 1998., koristeći čipove i PC, *Electronic Frontier Foundation* je napravio "DES Cracker".
  - Koštao je $250000, a za njegovu izradu je utrošeno godinu dana.
  - DES Cracker je razbio poruku šifriranu DES-om za 56 sati.
  - Sagrađen je od 1536 čipova koji mogu testirati 88 milijardi ključeva po sekundi.
  - Inače, u samoj konstrukciji ovog stroja nema ništa posebno novo. Sličnih prijedloga bilo je i ranije, ali EFF je prvi to sproveo u djelo i tek nakon izrade DES Crackera moglo se definitivno tvrditi da DES nije siguran kriptosistem.

Sigurnost DES-a

- Vremenski napadi
  - Napad na implementaciju algoritma
  - Koristi se poznavanje implementacije da bi se otkrili neki ili svi biti potključa
  - Informacija o ključu ili originalnoj poruci (*plaintext*-u) se dobija na osnovu posmatranja vremena izvravanja dekripcije za različite ulazne vrijednosti (*chipertext*)
  - Koristi se činjenica da vrijeme enkripcije i dekripcije varira za različite ulaze
  - Za sada djeluje da ovi napadi nemaju puno šansi protiv DES-a ili moćnijih mehanizama simetrične enkripcije kao što su trostruki DES i AES

Principi dizajna blok algoritama

Broj iteracija

Što je veći broj iteracija to je teža kriptoanliza

U opštem slučaju, broj iteracija treba da bude dovoljno veliki tako da je uspješna kriptoanaliza teža od *brute-force* pretrage

Ukoliko bi DES imao 15 ili manje iteracija, diferencijalna kriptoanaliza bi zahtijevala manje vremena od *brute-force* pretrage

Principi dizajna blok algoritama

Dizajn funkcije F

- Samo srce Feistel blok šifratora je funkcija ***F***
- Što je funkcija ***F*** nelinearnija, to je teži bilo koji oblik kriptoanalize
- SAC i BIC kriterijumi jačaju efektivnost funkcije konfuzije

Algoritam bi trebao da ima dobre *osobine lavine*

Strict avalanche criterion (SAC)

Bilo koji bit *j* na izlzu *S-box*-a trebalo bi da se mijenja sa vjerovatnoćom 1/2 kada se bilo koji ulazni bit *i* invertuje

Bit independence criterion (BIC)

Izlazni biti *j* i *k* trebalo bi da se mijenjaju nezavisno kada se bilo ulazni bit *i* invertuje

Principi dizajna blok algoritama
Generisanje ključeva

- Kod svih Feistel šifratora ključ se koristi za generisanje potključa u svakoj iteraciji
- U opštem slučaju, želimo da maksimizujemo kompleknost otkrivanja individualnih potključeva kao i kompleksnost otkrivanja polaznog ključa
- *Elementarni zahtjev*: Raspored ključeva mora biti takav da zadovoljava SAC i BIC kriterijume

Rezime

- Tradicionalna struktura blok šifratora
  - Enkripcija toka
  - Blok enkripcija
  - Motivativacija za primjenu Feistel strukture šifratora
  - Feistel šifrator
- *Data Encryption Standard* (DES)
  - Enkripcija
  - Dekripcija
  - Efekat lavine

- Sigurnost DES-a
  - 56-bitni ključevi
  - Priroda DES algoritma
  - Vremenski napadi
- Principi dizajna blok algoritama
  - Broj iteracija
  - Dizajn funkcije F
  - Algoritam raspoređivanja ključeva