

Bahan kuliah II4020 Kriptografi

Review Beberapa Block Cipher

(Bagian 1: DES dan Triple DES)



Oleh: Rinaldi Munir

Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

2025
Rinaldi Munir/Kriptografi/IF-STI-ITB

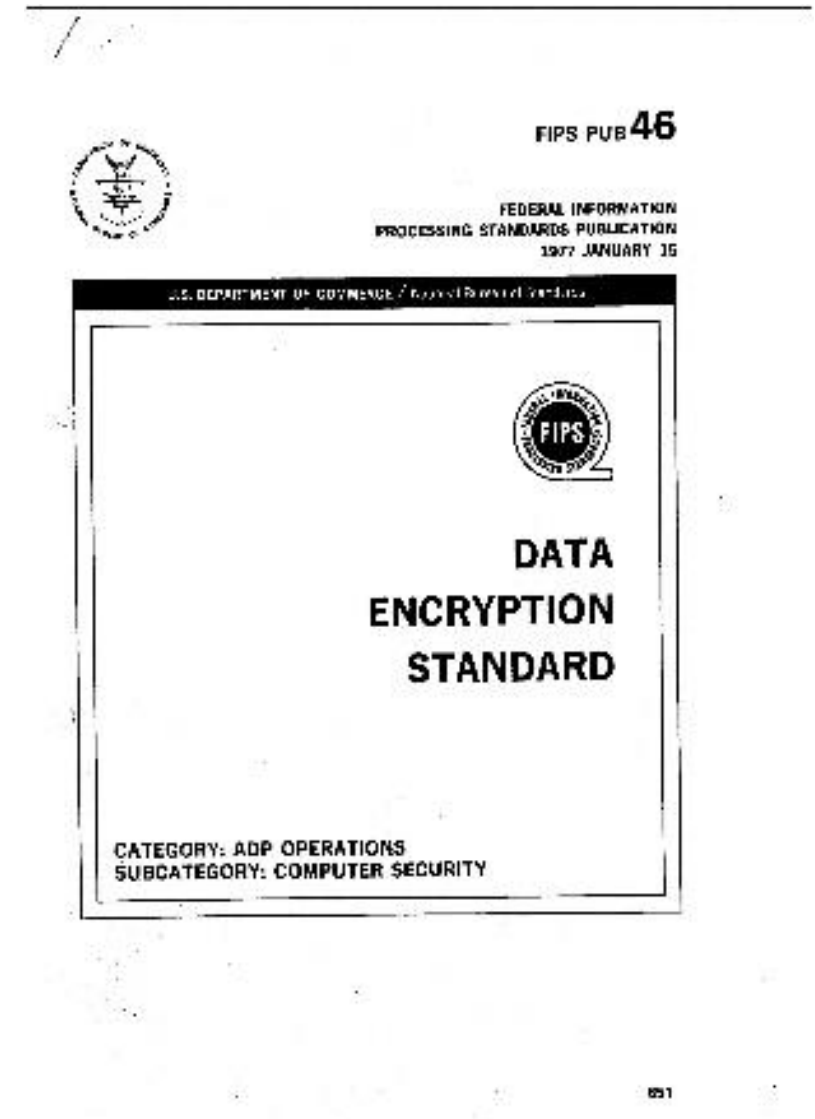
DES Data Encryption Standard

Data Encryption Standard
(DES)

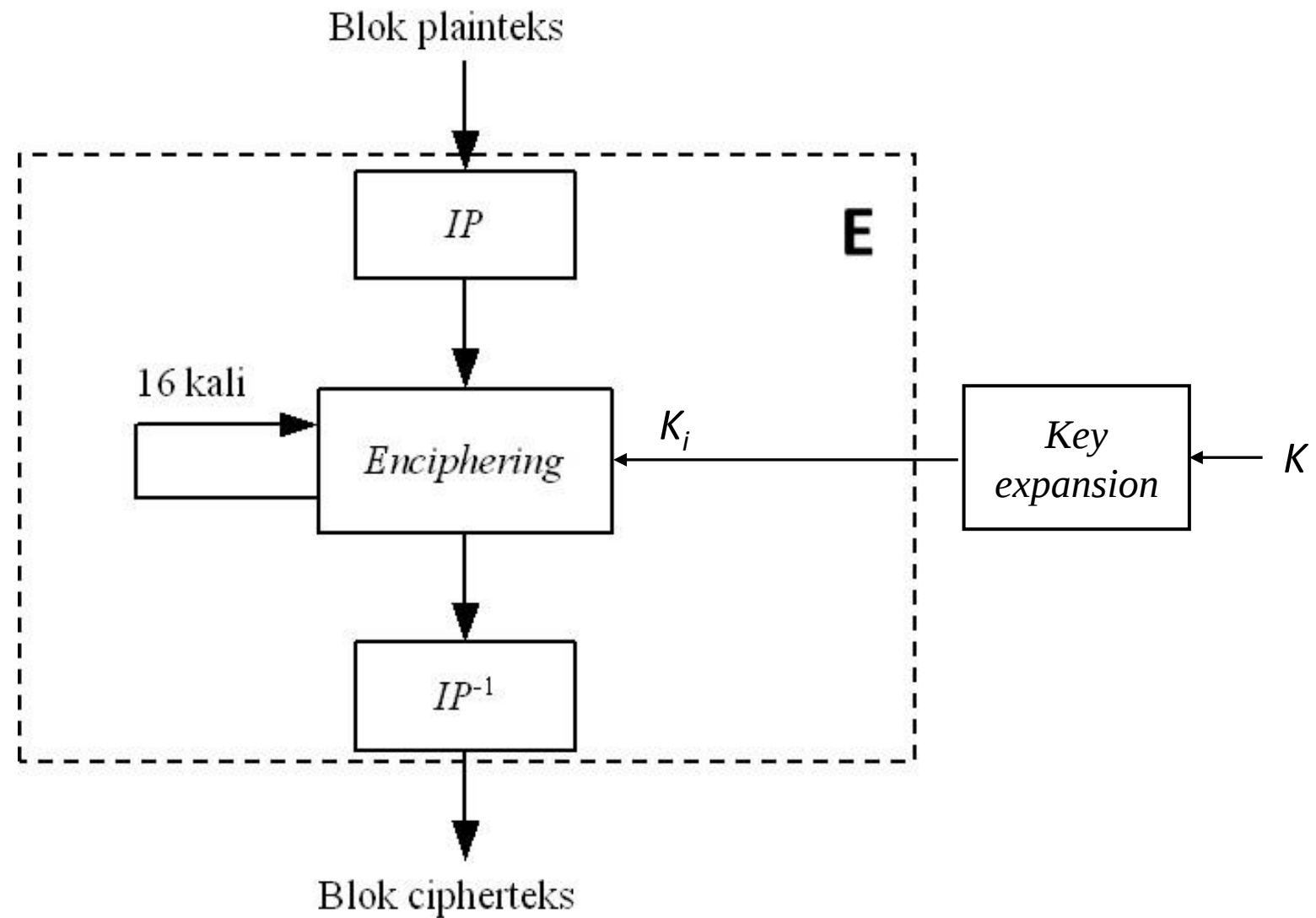
Sejarah DES

- Pada tahun 1972 NIST meminta standard enkripsi *cipher* blok.
- Pada tahun 1974 Horst Feistel di IBM mendesain cipher blok Lucifer (panjang kunci 128bit, Panjang blok 128 bit)
- Lucifer kemudian berkembang menjadi DES dengan beberapa masukan dari NSA (*National Security Agency*):
 - panjang kunci 56 bit, ukuran blok 64 bit
 - 16 putaran.
- Tahun 1976 DES disetujui oleh *National Bureau of Standard (NBS)* setelah penilaian kekuatannya oleh NSA.
- Sejak itu DES diimplementasikan secara luas, baik *software* maupun *hardware*.
- Tahun 1997-1998 DES berhasil dipecahkan dengan *brute force attack*
- Tahun 2001 DES diganti oleh AES

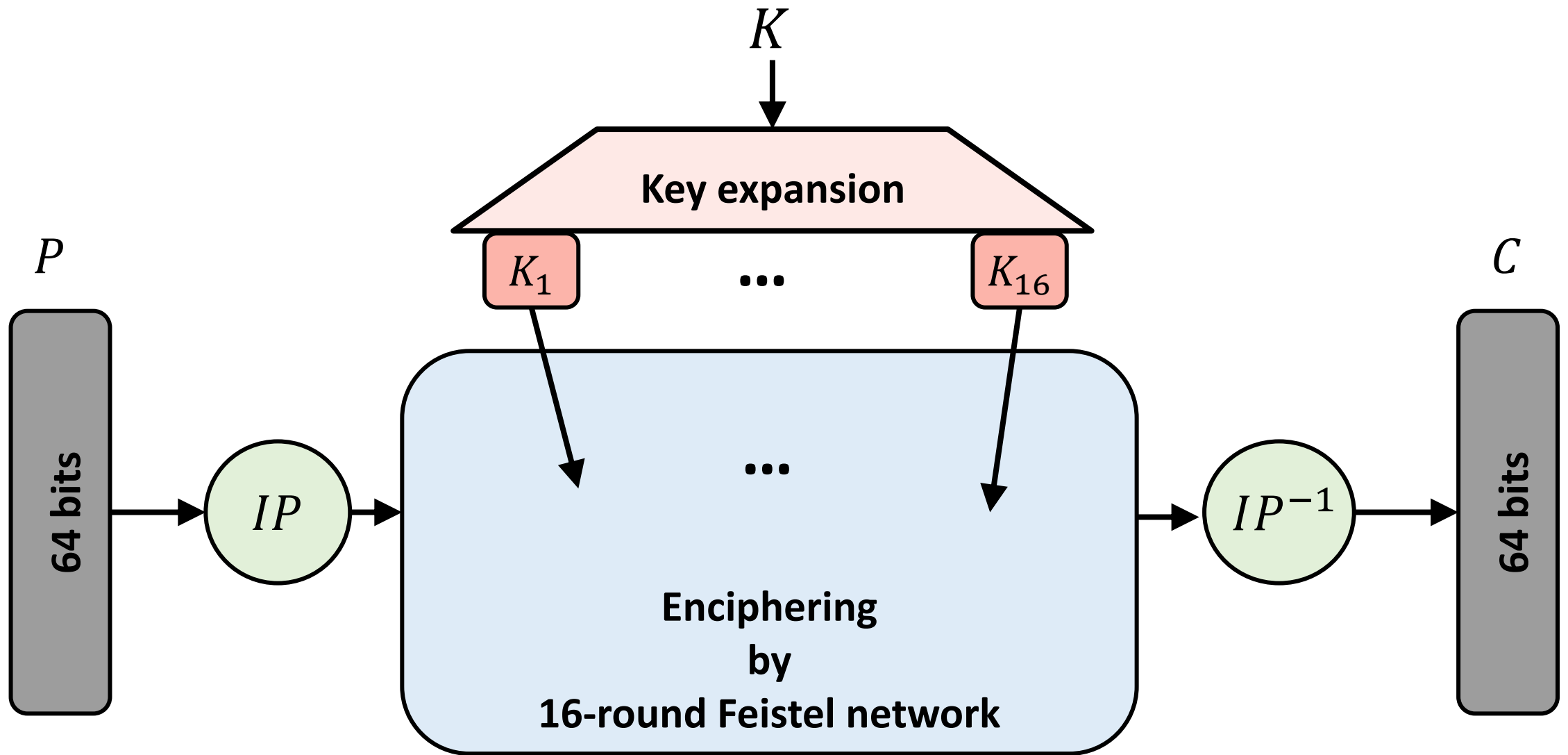
- Catat bahwa DES adalah sebuah standard, sedangkan algoritmanya sendiri bernama DEA (*Data Encryption Algorithm*). Kedua nama ini sering dikacaukan.
- DES termasuk ke dalam algoritma kriptografi kunci-simetri dan tergolong ke dalam *cipher* blok.
- DES beroperasi pada ukuran blok plainteks 64 bit.
- Panjang kunci eksternal = 64 bit (sesuai ukuran blok), tetapi hanya 56 bit yang dipakai (8 bit *parity* tidak digunakan). Bit parity = bit ke-8, 16, 24, dst.

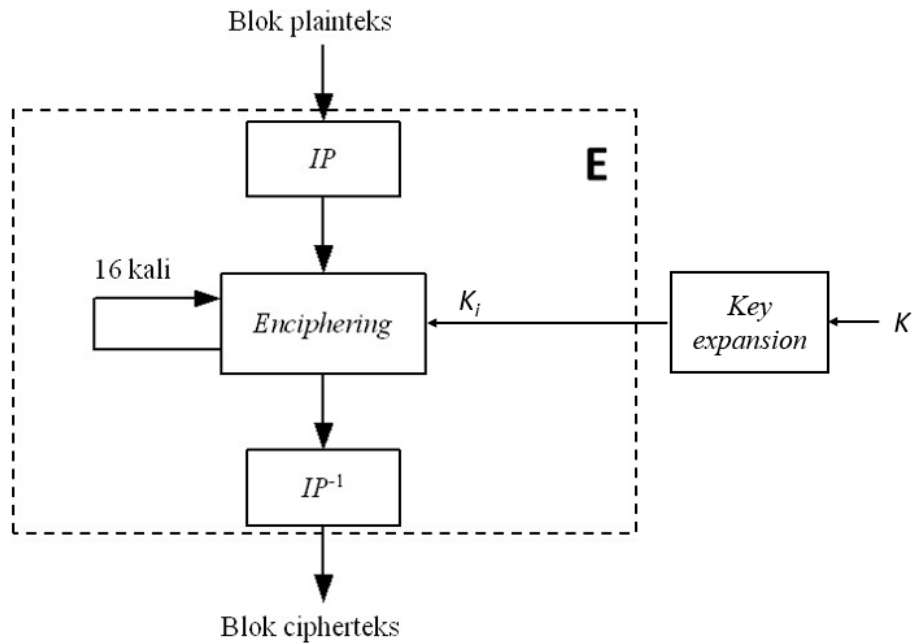


- Setiap blok plainteks dienkripsi dalam 16 putaran *enciphering*.
- Setiap putaran menggunakan kunci internal (kunci putaran) berbeda.
- Kunci internal sepanjang 48-bit dibangkitkan dari kunci eksternal
- Setiap blok plainteks mengalami permutasi awal (IP), 16 putaran *enciphering*, dan inversi permutasi awal (IP^{-1}). (lihat Gambar 1)

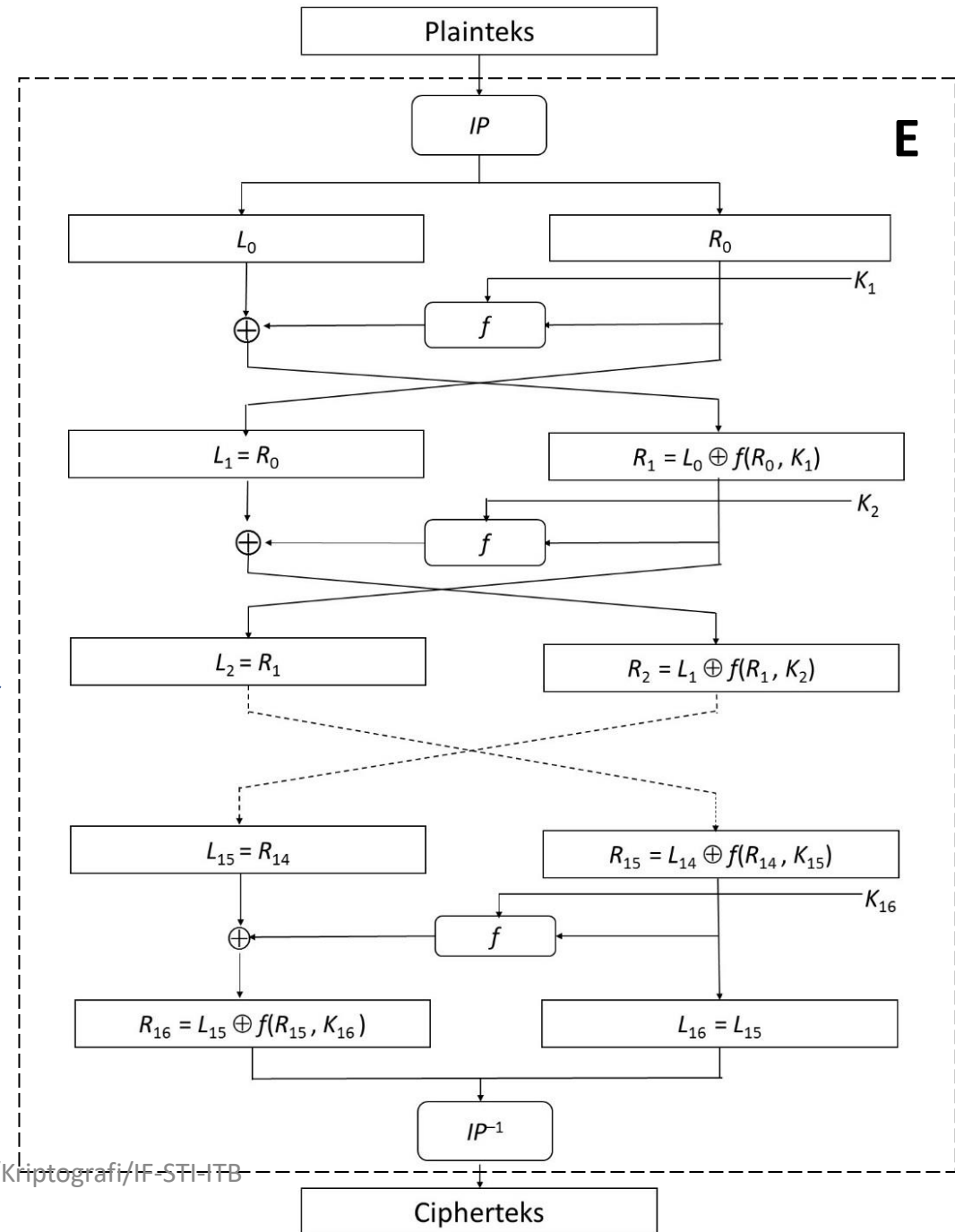


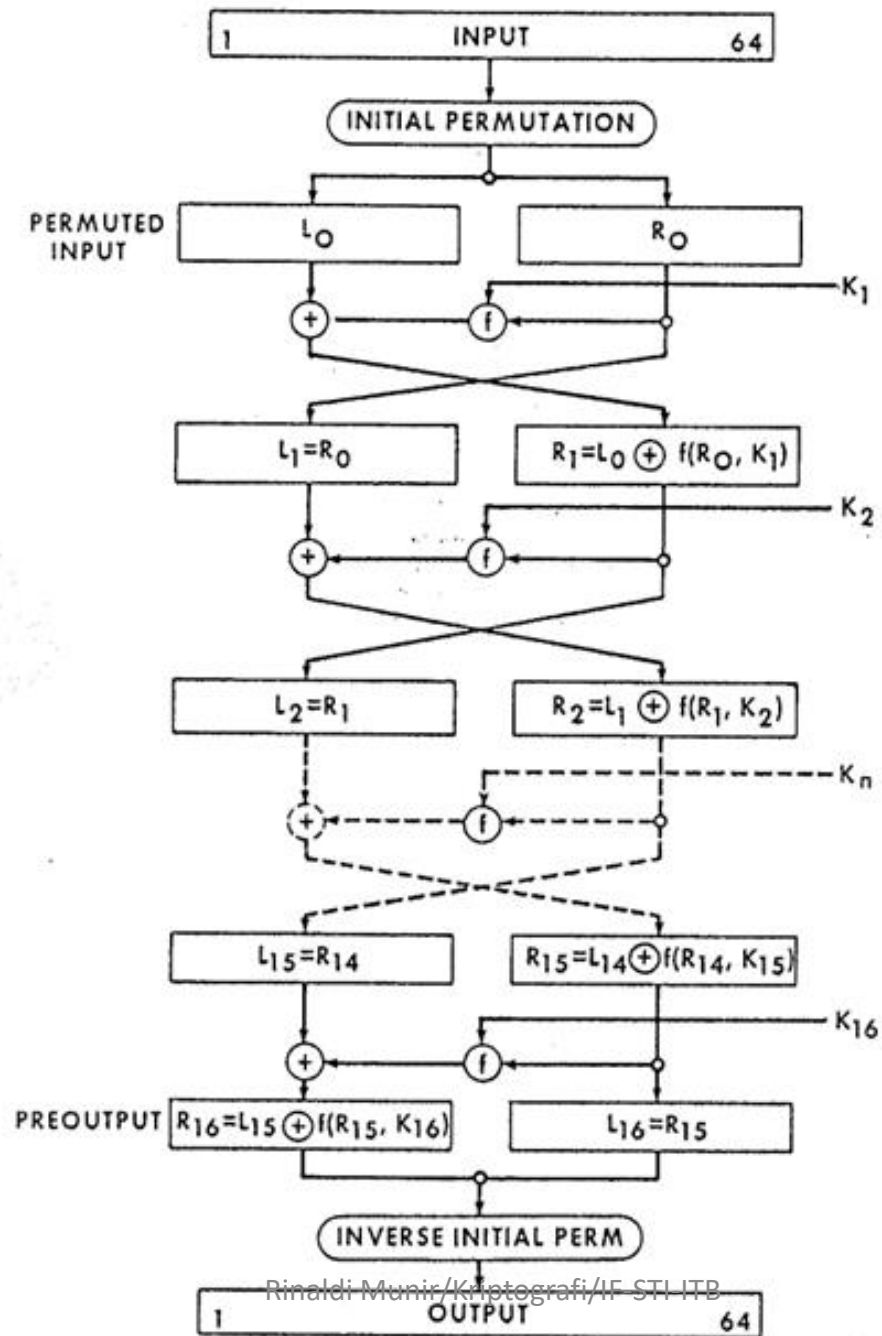
Gambar 1 Skema global algoritma DES





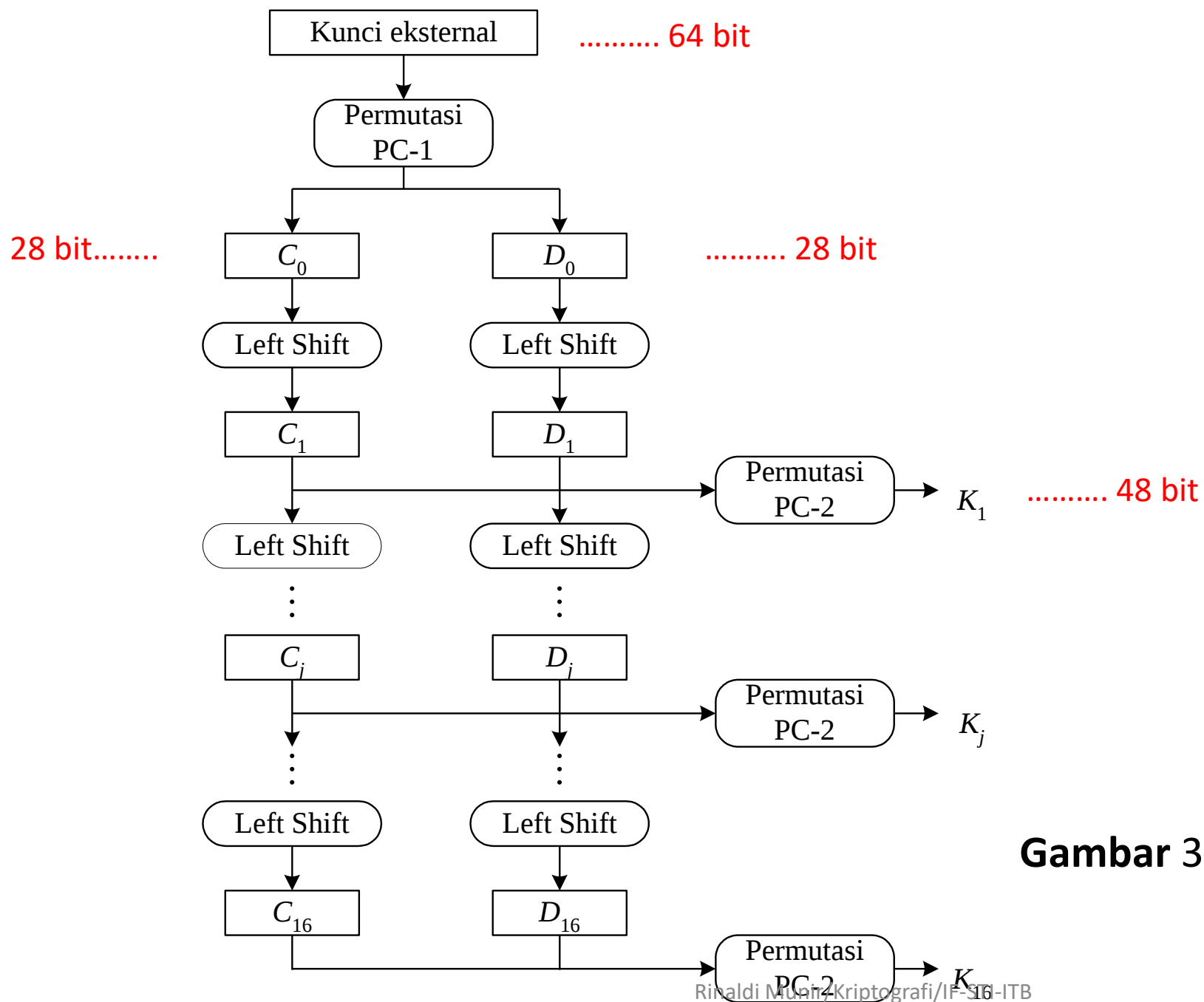
Gambar 2. Algoritma Enkripsi dengan DES



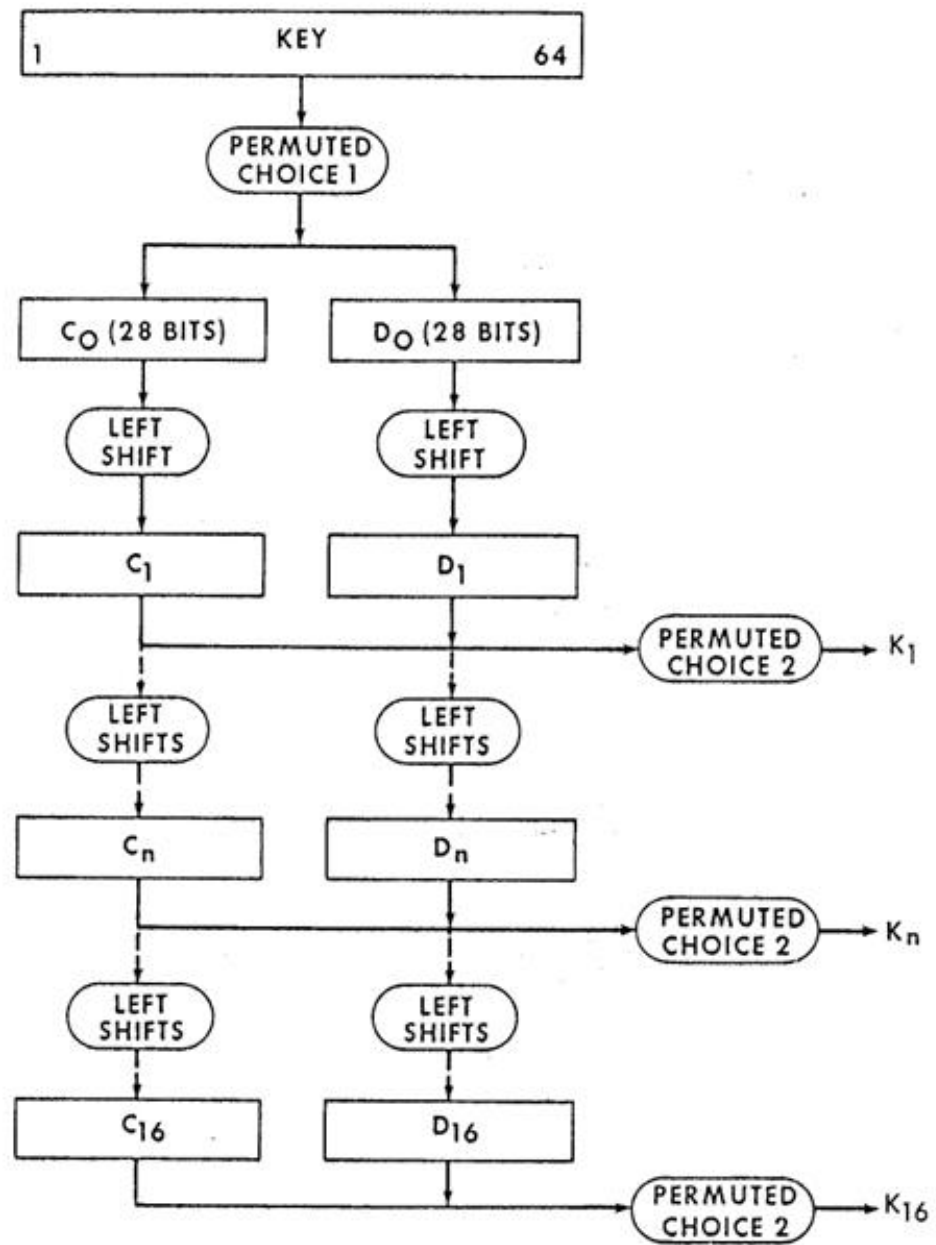


Pembangkitan Kunci Internal (*key expansion*)

- Kunci internal = kunci setiap putaran = *subkey* (*round key*)
- Ada 16 putaran, jadi ada 16 kunci internal: $K_1, K_2, \dots, K_{16}$
- Kunci internal dibangkitkan dari kunci eksternal (64 bit) yang diberikan oleh pengguna. Di dalam dokumen resmi DES pembangkitan kunci internal disebut *key schedule*.
- Panjang kunci internal adalah 48 bit
- Gambar 3 memperlihatkan proses pembangkitan kunci internal.



Gambar 3. Pembangkitan kunci internal



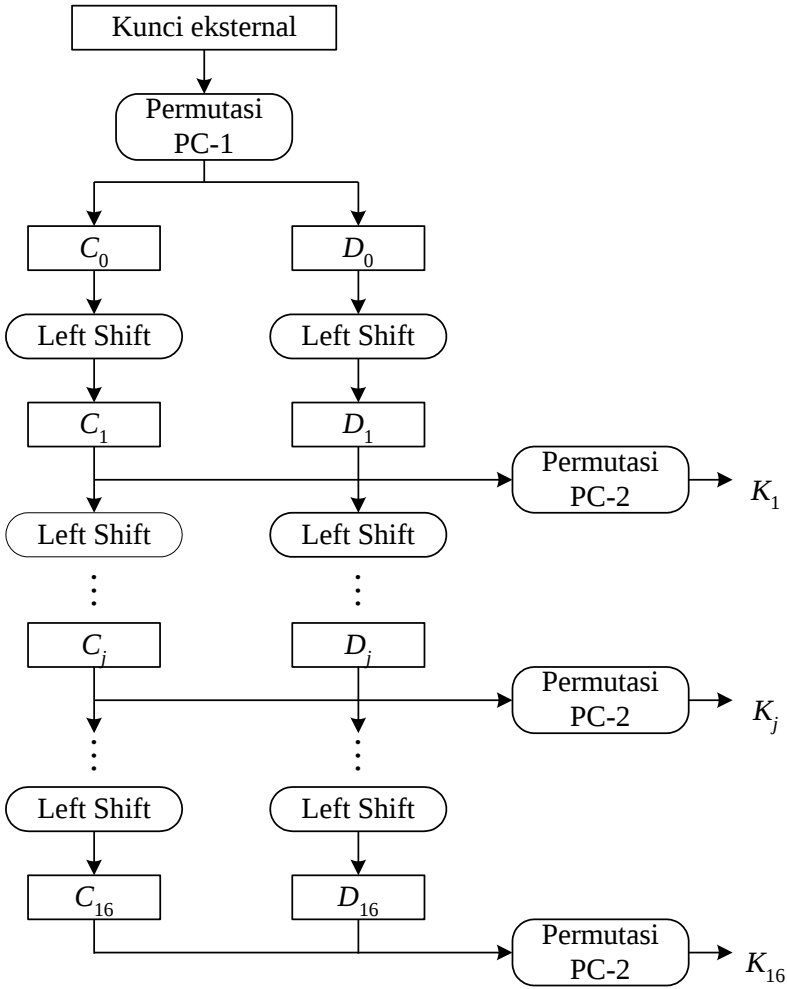
Matriks permutasi kompresi PC-1:

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

C_0 : berisi bit-bit dari K pada posisi
 57, 49, 41, 33, 25, 17, 9, 1, 58, 50, 42, 34, 26, 18
 10, 2, 59, 51, 43, 35, 27, 19, 11, 3, 60, 52, 44, 36

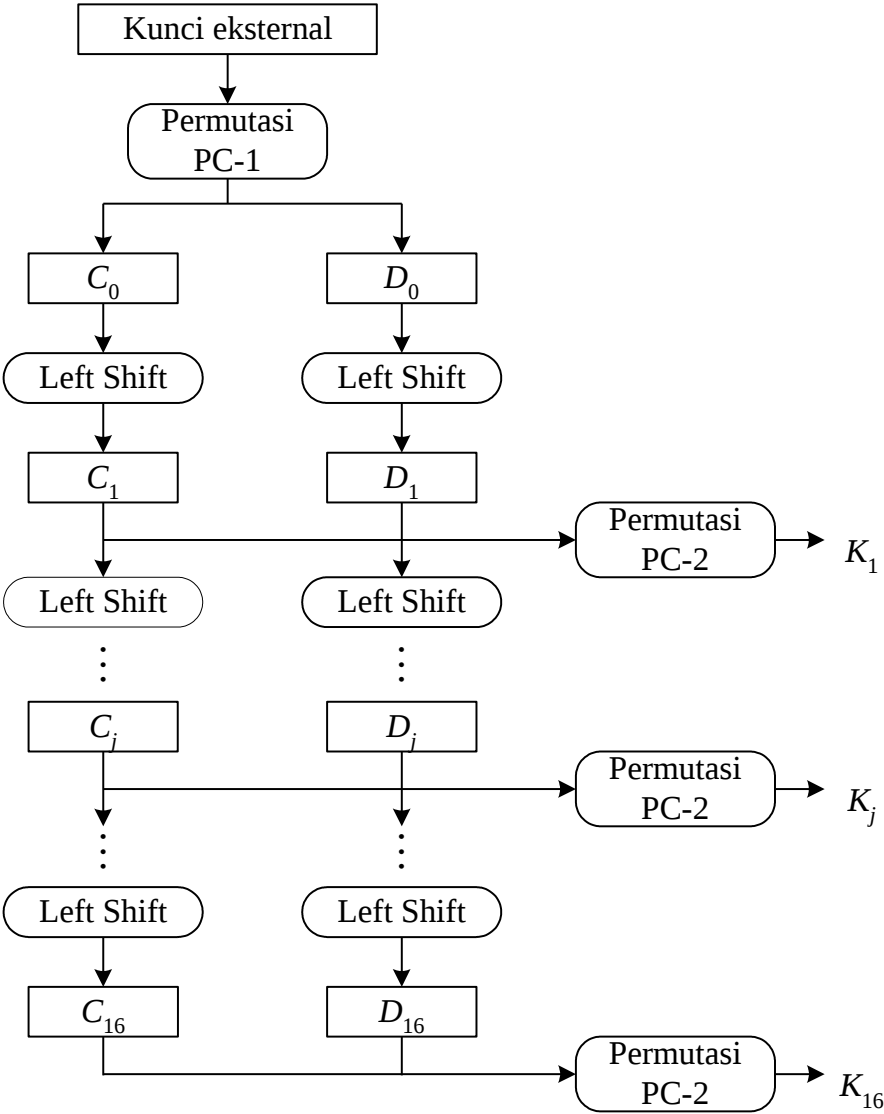
D_0 : berisi bit-bit dari K pada posisi
 63, 55, 47, 39, 31, 23, 15, 7, 62, 54, 46, 38, 30, 22
 14, 6, 61, 53, 45, 37, 29, 21, 13, 5, 28, 20, 12, 4

Perhatikan, bit-bit *parity*, yaitu bit ke-8, 16, 28, ... dibuang, sehingga menjadi 56 bit



Tabel 1. Jumlah pergeseran bit pada setiap putaran

Putaran, i	Jumlah pergeseran bit
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1



Matriks PC-2 berikut:

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

Jadi, K_i merupakan penggabungan bit-bit C_i pada posisi:

14, 17, 11, 24, 1, 5, 3, 28, 15, 6, 21, 10

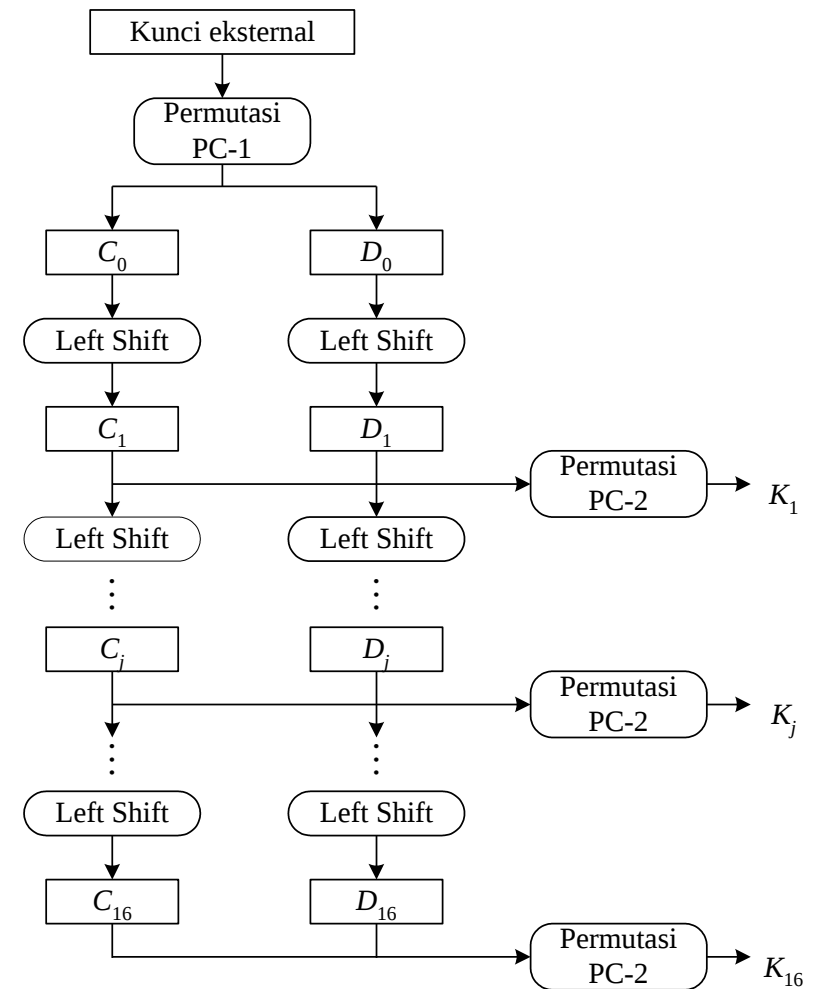
23, 19, 12, 4, 26, 8, 16, 7, 27, 20, 13, 2

dengan bit-bit D_i pada posisi:

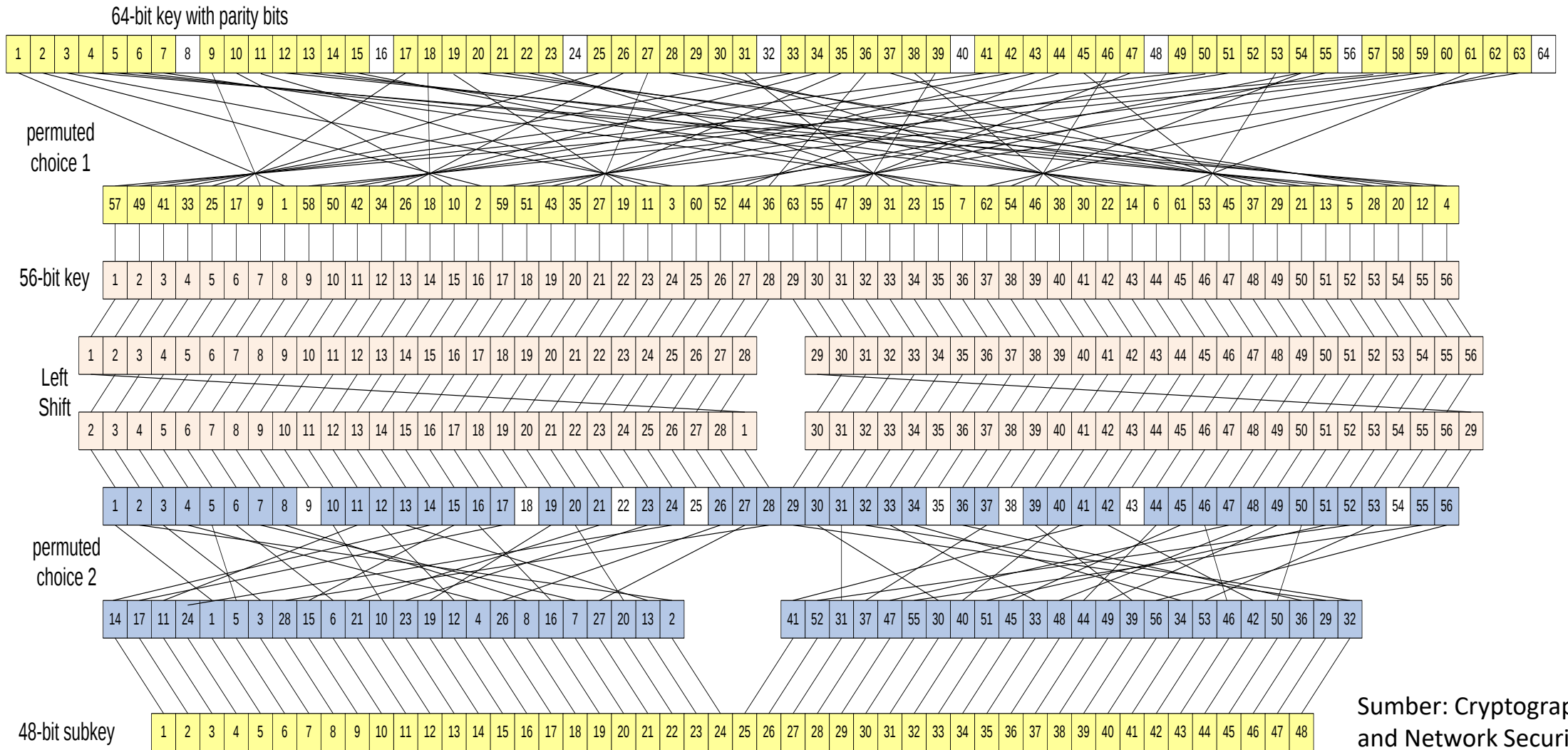
41, 52, 31, 37, 47, 55, 30, 40, 51, 45, 33, 48

44, 49, 39, 56, 34, 53, 46, 42, 50, 36, 29, 32

Setiap kunci internal K_i mempunyai panjang 48 bit.



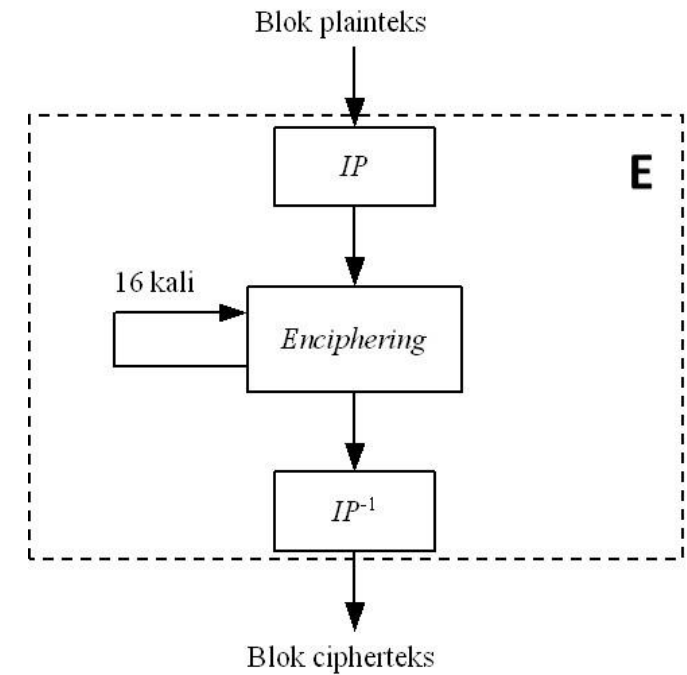
Pembangkitan kunci internal secara lengkap:



Permutasi Awal

- Tujuan: mengacak plainteks sehingga urutan bit-bit di dalamnya berubah.
- Matriks permutasi awal (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

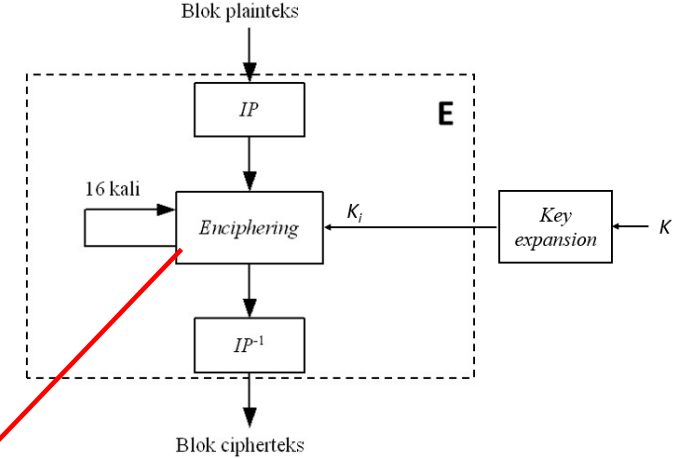
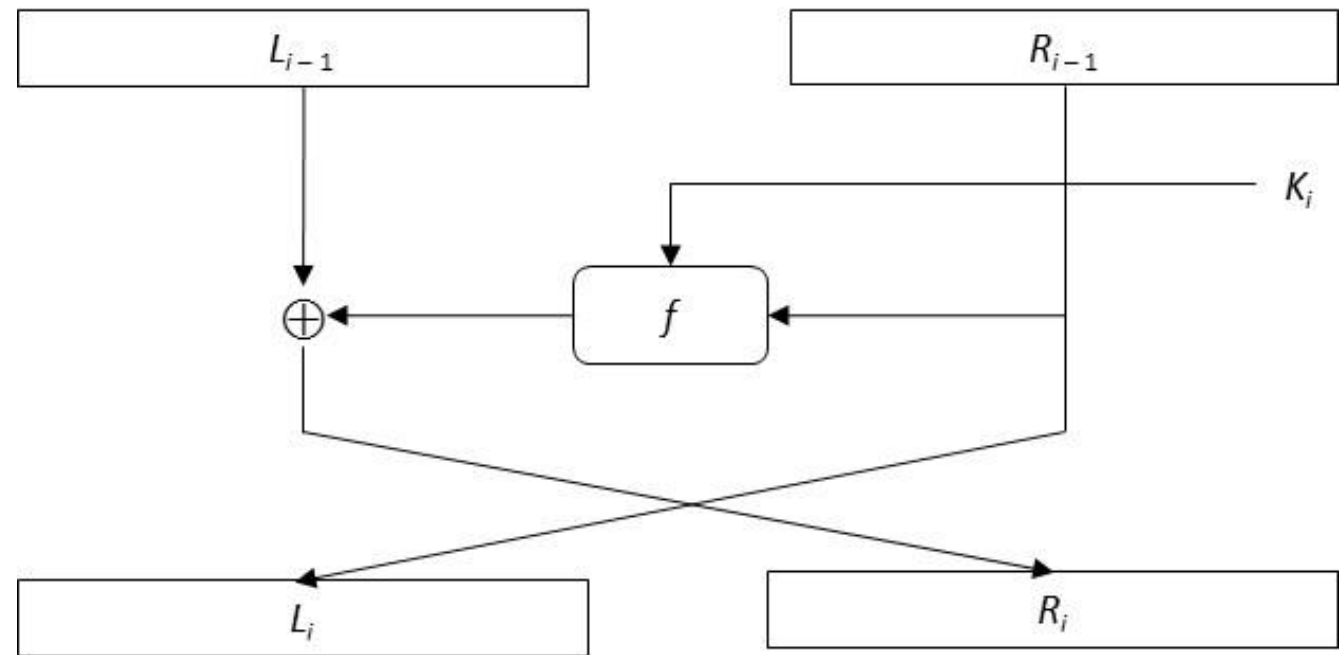


Enciphering

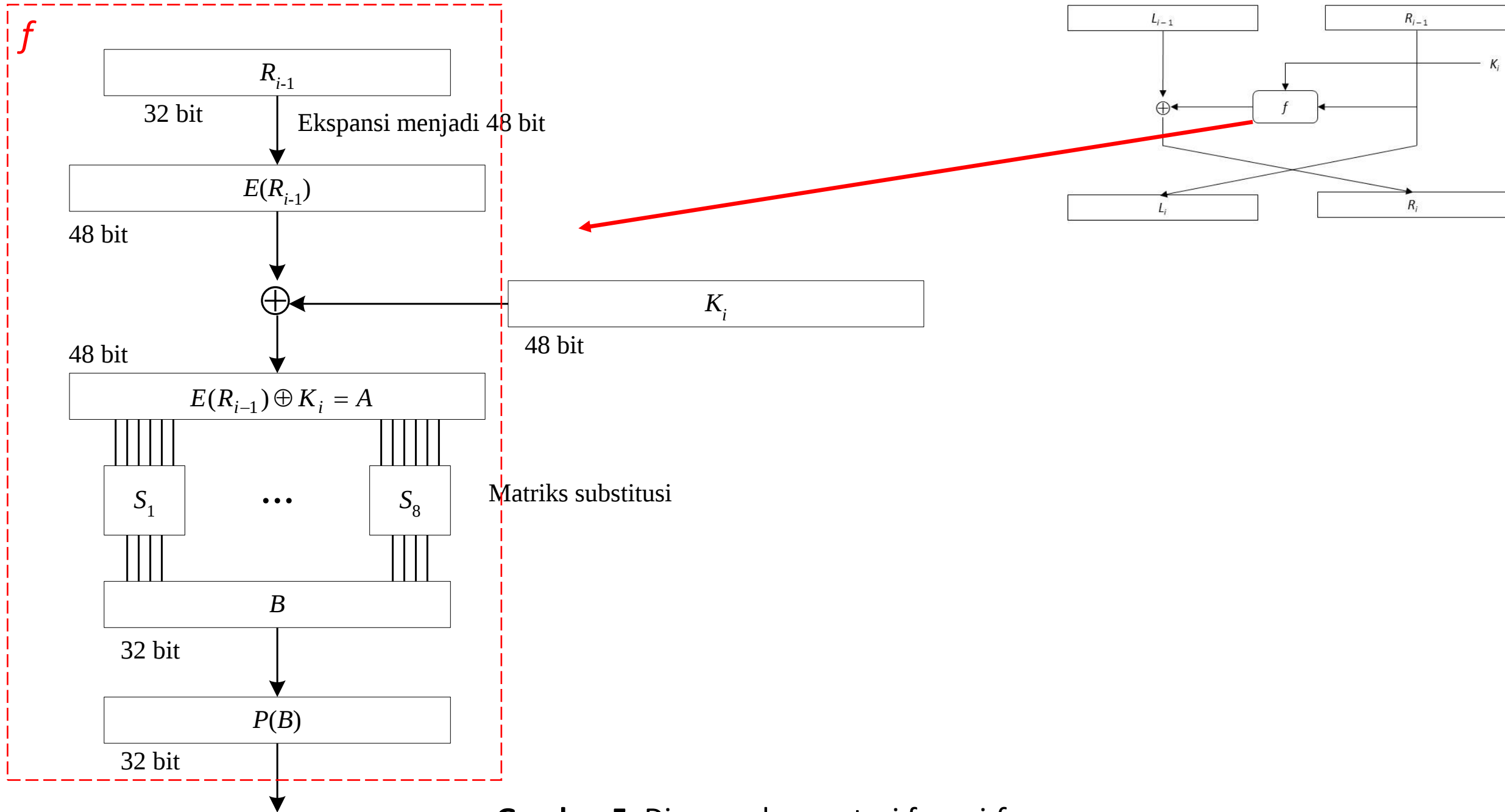
- Setiap blok plainteks mengalami 16 kali putaran *enciphering*
- Setiap putaran *enciphering* merupakan jaringan Feistel:

$$L_i = R_{i-1}$$

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



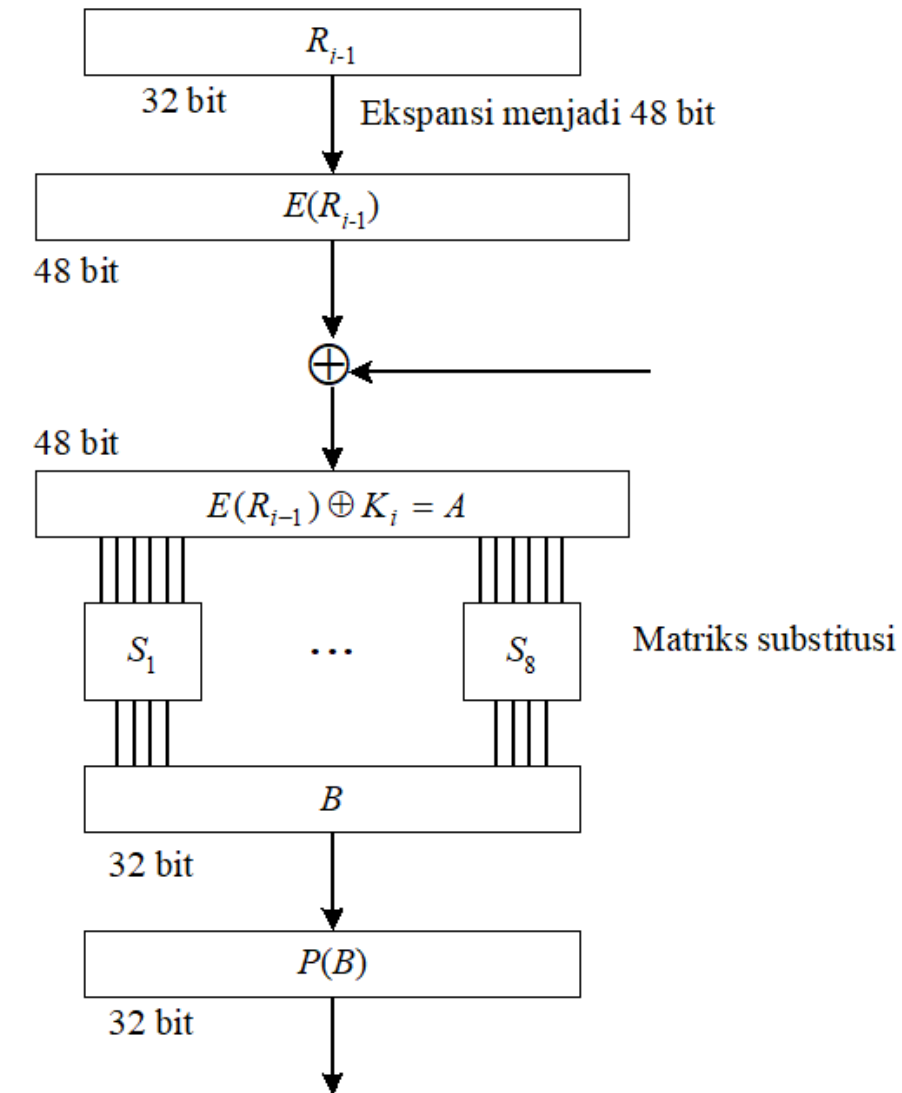
Gambar 4. Satu putaran *enciphering*



Gambar 5. Diagram komputasi fungsi f :

- E adalah fungsi ekspansi yang memperluas blok R_{i-1} 32-bit menjadi blok 48 bit.
- Fungsi ekspansi direalisasikan dengan matriks permutasi ekspansi:

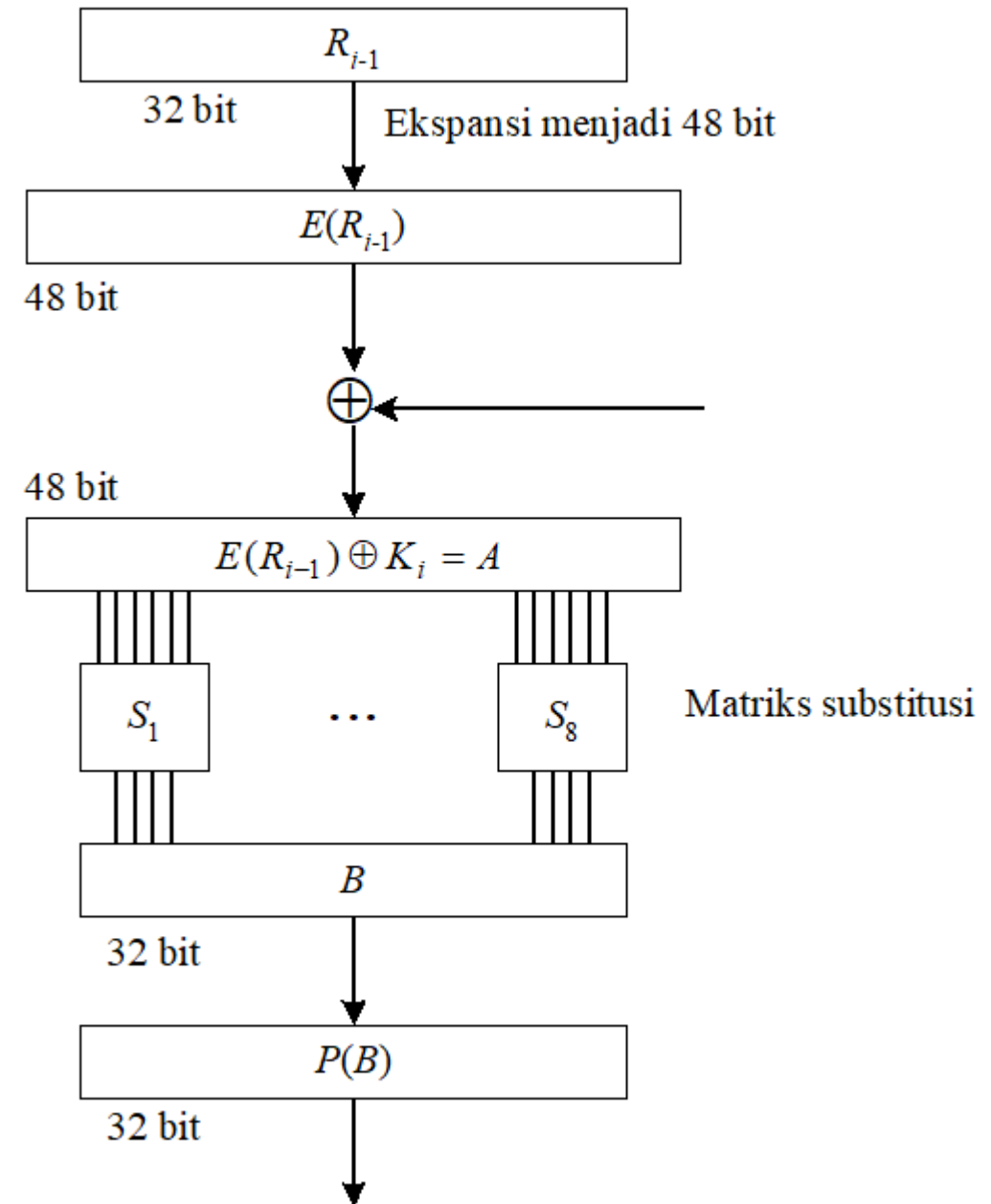
32	1	2	3	4	5	4	5	6	7	8	9
8	9	10	11	12	13	12	13	14	15	16	17
16	17	18	19	20	21	20	21	22	23	24	25
24	25	26	27	28	29	28	29	30	31	32	1



- Hasil ekspansi, yaitu $E(R_{i-1})$ di-XOR-kan dengan K_i menghasilkan blok A berukuran 48-bit:

$$E(R_{i-1}) \oplus K_i = A$$

- Blok A dikelompokkan menjadi 8 kelompok, masing-masing 6 bit, dan menjadi masukan bagi proses substitusi.
- Ada 8 matriks substitusi, masing-masing dinyatakan dengan kotak-S.
- Kotak $-S$ menerima masukan 6 bit dan memberikan luaran 4 bit.



S_1 :

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2 :

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3 :

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4 :

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

Input: 101010

Baris = 10 = 2

Kolom = 0101 = 5

Luaran = 13 = 1101

S_5 :

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	16
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6 :

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7 :

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8 :

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

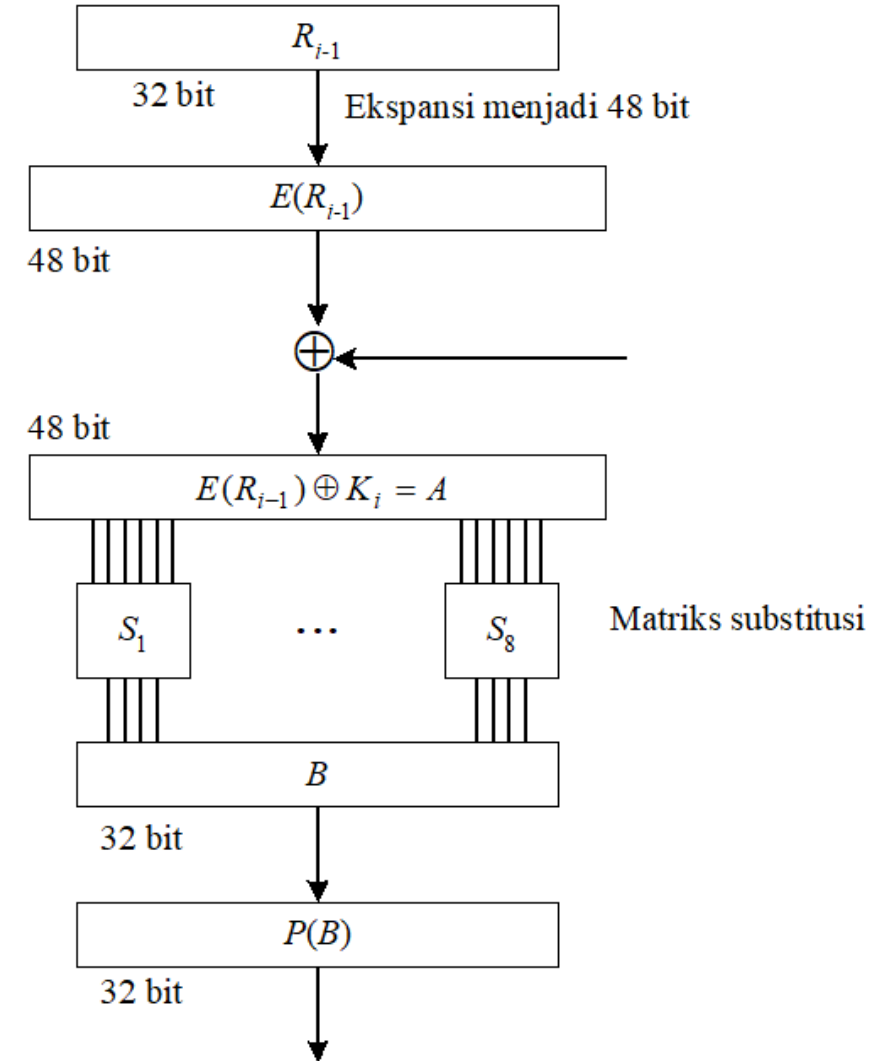
- Setiap baris di dalam S-box adalah permutasi angka-angka 0, 1, 2,..., 15.

- Mengubah satu bit di dalam input S-box mengakibatkan perubahan paling sedikit 2 bit pada output S-box

→ prinsip *confussion*

- Luaran proses substitusi adalah blok B yang panjangnya 32 bit.
- Blok B menjadi masukan untuk proses permutasi.
- Tujuan permutasi adalah untuk mengacak hasil proses substitusi kotak-S.
- Permutasi dilakukan dengan menggunakan matriks permutasi P (P -box) sbb:

16	7	20	21	29	12	28	17	1	15	23	26	5	8	31	10
2	8	24	14	32	27	3	9	19	13	30	6	22	11	4	25

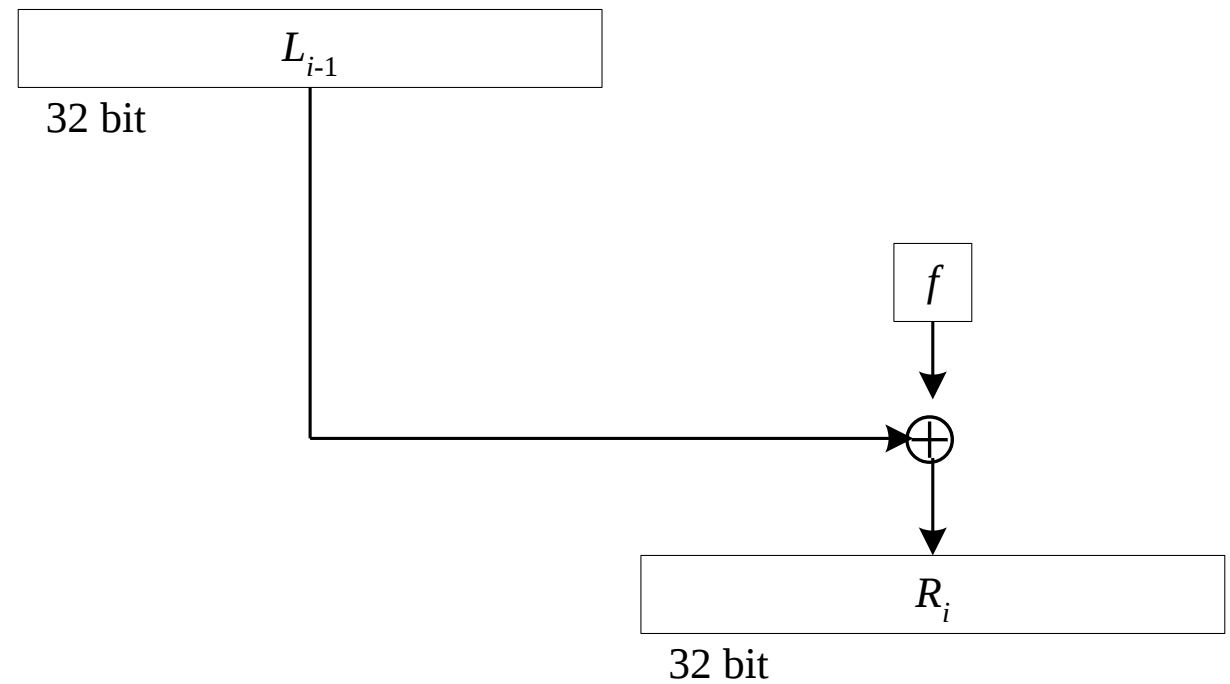


- $P(B)$ merupakan luaran dari fungsi f .
- Bit-bit $P(B)$ di- XOR -kan dengan L_{i-1} menghasilkan R_i :

$$R_i = L_{i-1} \oplus P(B)$$

- Jadi, luaran dari putaran ke- i adalah

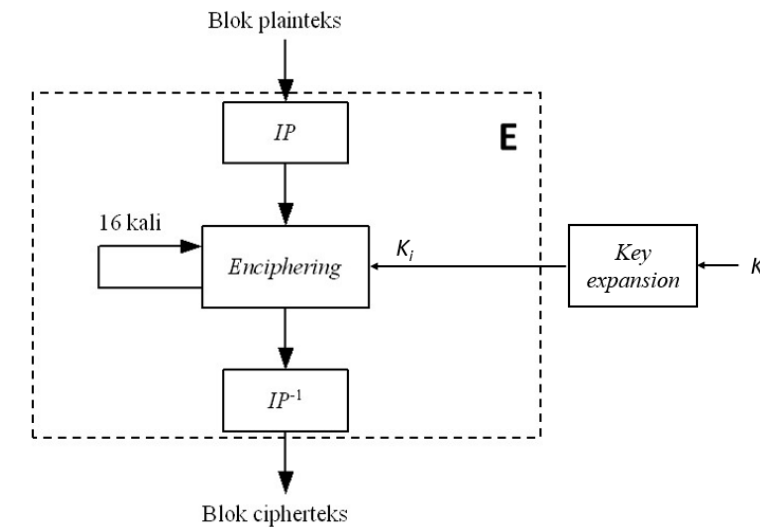
$$(L_i, R_i) = (R_{i-1}, L_{i-1} \oplus P(B))$$



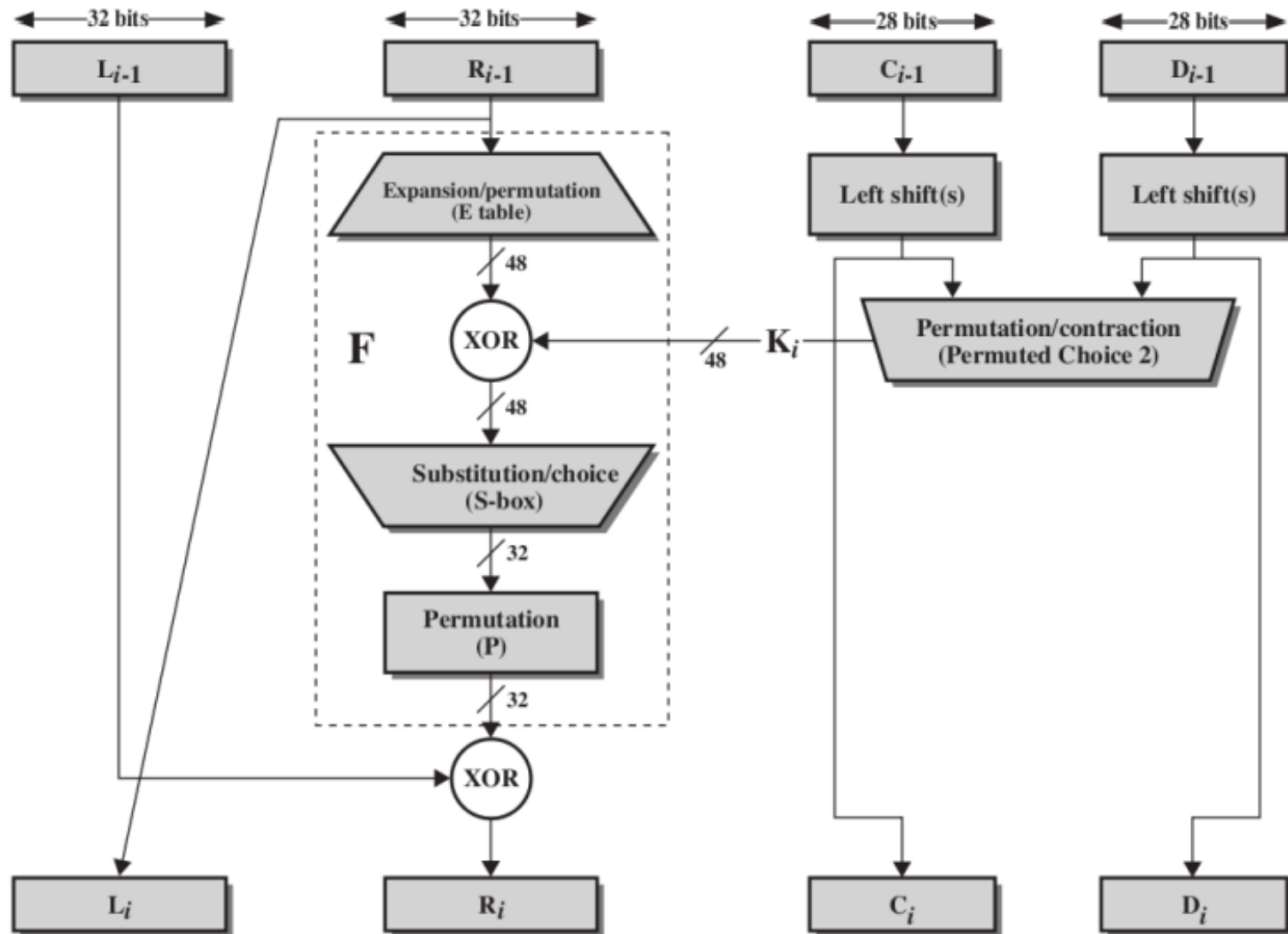
Inversi Permutasi (IP^{-1})

- Permutasi terakhir dilakukan setelah 16 kali putaran terhadap gabungan blok kiri dan blok kanan.
- Permutasi menggunakan matriks permutasi awal balikan (IP^{-1}) sbb:

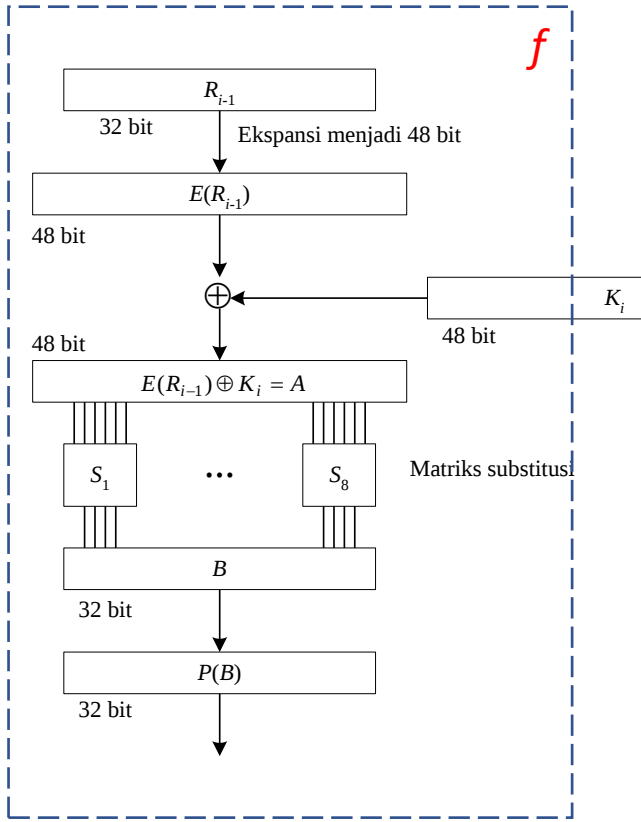
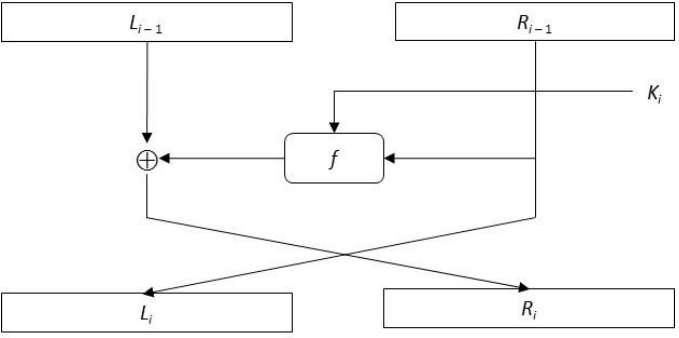
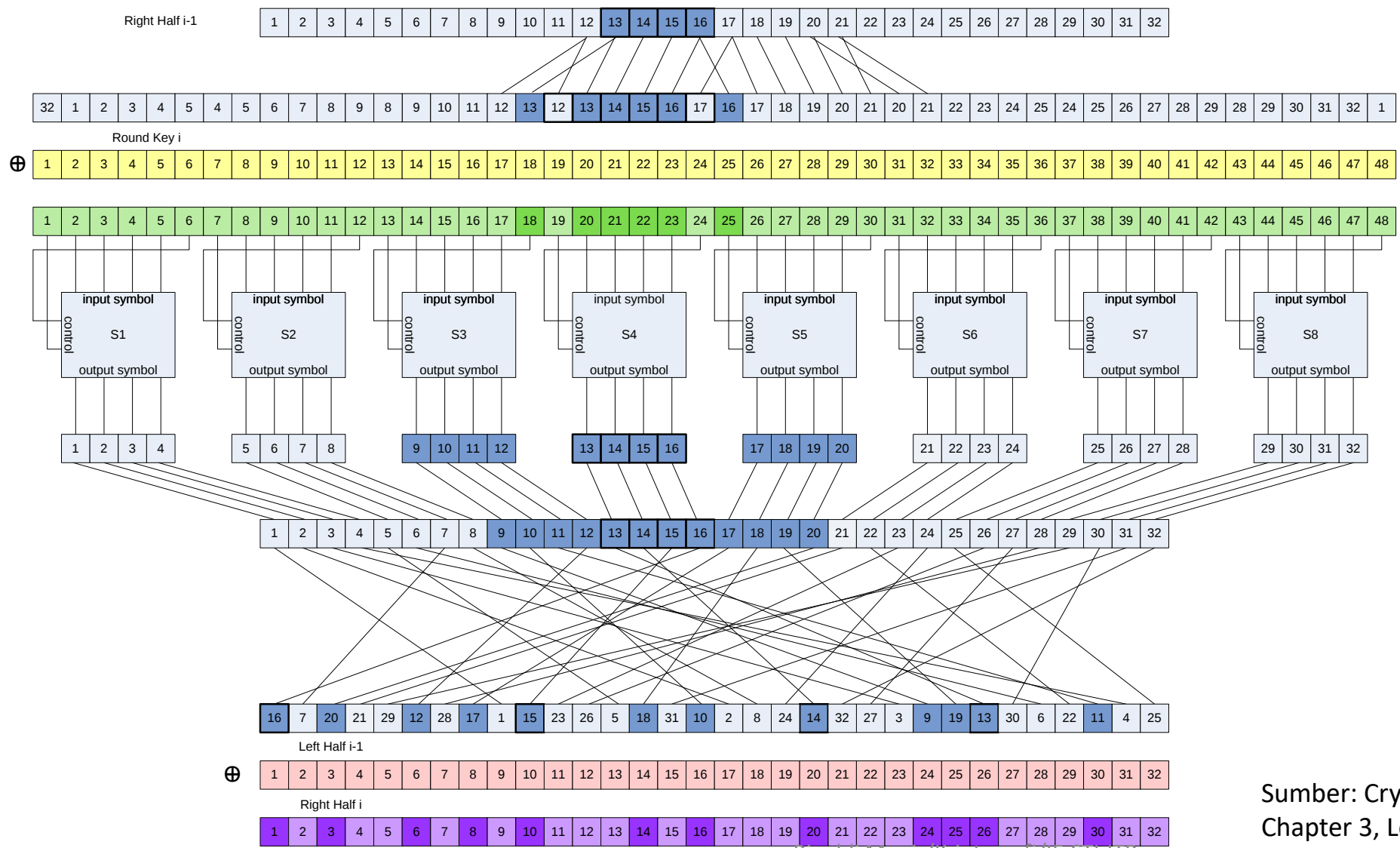
40	8	48	16	56	24	64	32	39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30	37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28	35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26	33	1	41	9	49	17	57	25



Satu putaran di dalam DES:



Satu putaran di dalam jaringan Feistel secara lengkap:



Sumber: Cryptography and Network Security
Chapter 3, Lecture slides by Lawrie Brown
Modified by Richard Newman

Contoh hasil enkripsi DES pada setiap putaran:

Round	K_i	L_i	R_i
IP		5a005a00	3cf03c0f
1	1e030f03080d2930	3cf03c0f	bad22845
2	0a31293432242318	bad22845	99e9b723
3	23072318201d0c1d	99e9b723	0bae3b9e
4	05261d3824311a20	0bae3b9e	42415649
5	3325340136002c25	42415649	18b3fa41
6	123a2d0d04262a1c	18b3fa41	9616fe23
7	021f120b1c130611	9616fe23	67117cf2
8	1c10372a2832002b	67117cf2	c11bfc09
9	04292a380c341f03	c11bfc09	887fbc6c
10	2703212607280403	887fbc6c	600f7e8b
11	2826390c31261504	600f7e8b	f596506e
12	12071c241a0a0f08	f596506e	738538b8
13	300935393c0d100b	738538b8	c6a62c4e
14	311e09231321182a	c6a62c4e	56b0bd75
15	283d3e0227072528	56b0bd75	75e8fd8f
16	2921080b13143025	75e8fd8f	25896490
IP ⁻¹		da02ce3a	89ecac3b

Sumber: Cryptography
and Network Security
Chapter 3, Lecture slides
by Lawrie Brown
Modified by Richard
Newman

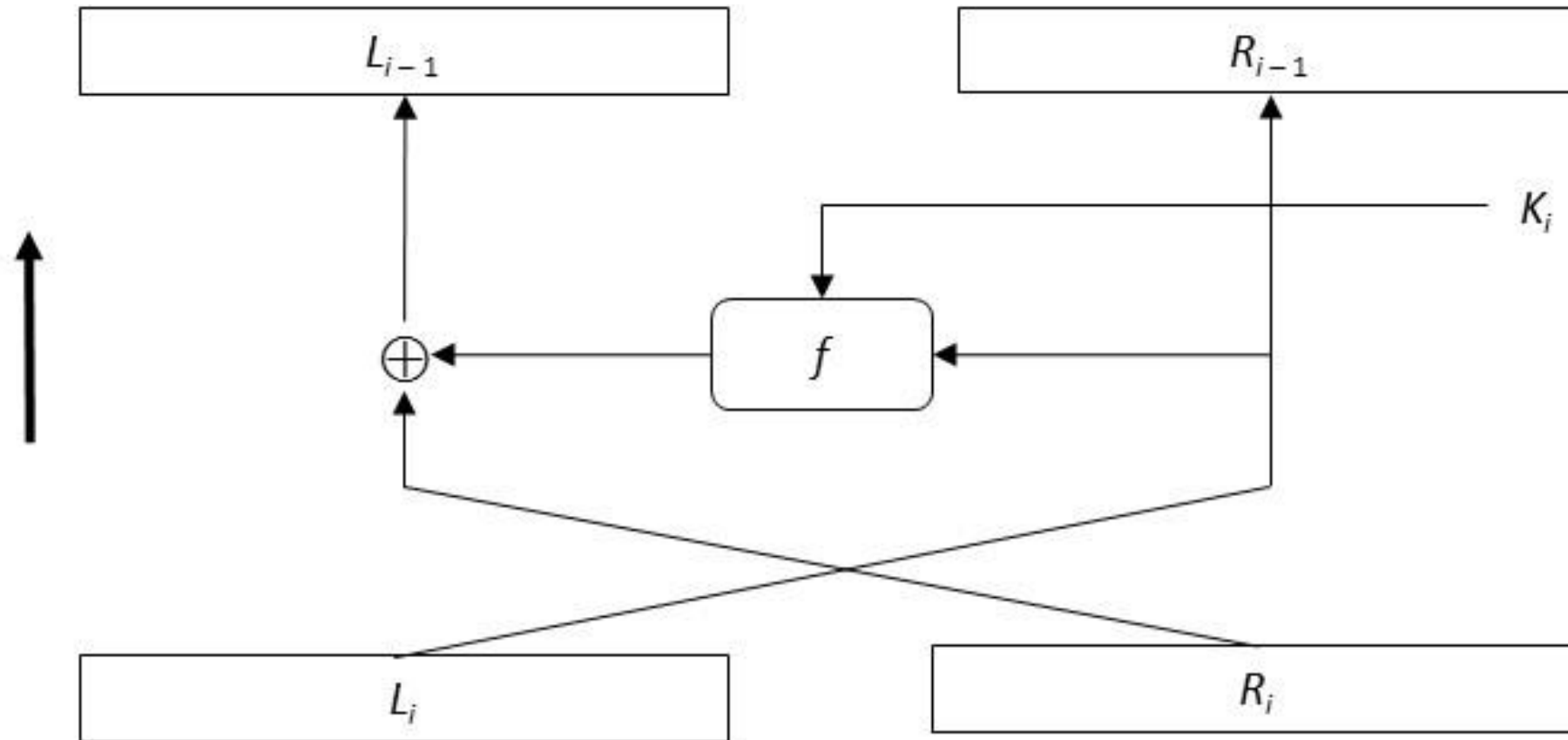
Dekripsi

- Dekripsi terhadap cipherteks merupakan kebalikan dari proses enkripsi.
- DES menggunakan algoritma yang sama untuk proses enkripsi dan dekripsi.
- Pada proses dekripsi urutan kunci yang digunakan adalah $K_{16}, K_{15}, \dots, K_1$.
- Untuk tiap putaran 16, 15, ..., 1, luaran pada setiap putaran *deciphering* adalah

$$R_{i-1} = L_i$$

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

Dekripsi:



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

Efek longsor (*avalanche effect*)

- Efek longsor adalah properti kunci yang diinginkan dari algoritma enkripsi,
- Tujuan: menunjukkan perubahan satu bit pada plainteks atau kunci menghasilkan perubahan besar pada cipherteks
- Efek longsor membuat upaya untuk mencoba menebak kunci menjadi tidak mungkin
- DES menunjukkan efek longsor yang kuat

Efek longsor di dalam DES

(mengubah satu bit plainteks)

original	Round		δ	Round		δ
		02468aceeca86420	1	9	c11bfc09887fbc6c	32
modified		12468aceeca86420			99f911532eed7d94	
	1	3cf03c0fbad22845	1	10	887fbc6c600f7e8b	34
		3cf03c0fbad32845			2eed7d94d0f23094	
	2	bad2284599e9b723	5	11	600f7e8bf596506e	37
		bad3284539a9b7a3			d0f23094455da9c4	
	3	99e9b7230bae3b9e	18	12	f596506e738538b8	31
		39a9b7a3171cb8b3			455da9c47f6e3cf3	
	4	0bae3b9e42415649	34	13	738538b8c6a62c4e	29
		171cb8b3ccaca55e			7f6e3cf34bc1a8d9	
	5	4241564918b3fa41	37	14	c6a62c4e56b0bd75	33
		ccaca55ed16c3653			4bc1a8d91e07d409	
	6	18b3fa419616fe23	33	15	56b0bd7575e8fd8f	31
		d16c3653cf402c68			1e07d4091ce2e6dc	
	7	9616fe2367117cf2	32	16	75e8fd8f25896490	32
		cf402c682b2cefbc			1ce2e6dc365e5f59	
	8	67117cf2c11bfc09	33	IP ⁻¹	da02ce3a89ecac3b	32
		2b2cefbc99f91153			057cde97d7683f2a	

Sumber: Cryptography
and Network Security
Chapter 3, Lecture slides
by Lawrie Brown
Modified by Richard
Newman

Efek longsor di dalam DES

(mengubah satu bit kunci)

Round		δ
	02468aceeca86420 02468aceeca86420	0
1	3cf03c0fbad22845 3cf03c0f9ad628c5	3
2	bad2284599e9b723 9ad628c59939136b	11
3	99e9b7230bae3b9e 9939136b768067b7	25
4	0bae3b9e42415649 768067b75a8807c5	29
5	4241564918b3fa41 5a8807c5488dbe94	26
6	18b3fa419616fe23 488dbe94aba7fe53	26
7	9616fe2367117cf2 aba7fe53177d21e4	27
8	67117cf2c11bfc09 177d21e4548f1de4	32

Round		δ
9	c11bfc09887fbc6c 548f1de471f64dfd	34
10	887fbc6c600f7e8b 71f64dfd4279876c	36
11	600f7e8bf596506e 4279876c399fdc0d	32
12	f596506e738538b8 399fdc0d6d208dbb	28
13	738538b8c6a62c4e 6d208dbbb9bdeaaa	33
14	c6a62c4e56b0bd75 b9bdeeaad2c3a56f	30
15	56b0bd7575e8fd8f d2c3a56f2765c1fb	33
16	75e8fd8f25896490 2765c1fb01263dc4	30
IP-1	da02ce3a89ecac3b ee92b50606b62b0b	30

Mode DES

- DES dapat dioperasikan dengan mode ECB, CBC, OFB, CFB, dan mode *counter*.
- Namun karena kesederhanaannya, mode ECB lebih sering digunakan pada paket komersil, sehingga dikenal dengan nama DES-ECB.

Kunci-kunci lemah DES

- **Kunci lemah DES** adalah kunci K sedemikian sehingga $E_K(E_K(x)) = x$ untuk semua plainteks x , yang artinya enkripsi dan dekripsi menjadi sama.
 - Kunci lemah ini menghasilkan pembangkitan kunci putaran yang sama pada semua putaran.
- DES memiliki 4 kunci lemah (masing-masing panjangnya 56 bit), dinyatakan dalam kode heksadesimal sbb:
0000000000000000
0000000FFFFFFFFF
FFFFFFFF00000000
FFFFFFFFFFFFFFFF
• Kunci lemah seharusnya dihindari.



Implementasi DES

- DES sudah diimplementasikan baik sebagai perangkat lunak maupun perangkat keras.
- Dalam bentuk perangkat keras, DES diimplementasikan di dalam *chip*. Setiap detik *chip* ini dapat mengenkripsikan 16,8 juta blok (atau 1 gigabit per detik).
- Implementasi DES ke dalam perangkat lunak dapat melakukan enkripsi 32.000 blok per detik (dijalankan pada komputer *mainframe* IBM 3090, yaitu komputer tercepat saat itu / tahun 1976).

Keamanan DES

- Keamanan DES ditentukan oleh kunci.
- Panjang kunci eksternal DES hanya 64 bit, tetapi yang dipakai hanya 56 bit.
- Pada rancangan awal, panjang kunci yang diusulkan IBM adalah 128 bit, tetapi atas permintaan NSA, panjang kunci diperkecil menjadi 56 bit.
- Dengan panjang kunci 56 bit akan terdapat 2^{56} atau 72.057.594.037.927.936 kemungkinan kunci.
- Ruang kunci (*key space*) DES sebanyak 2^{56} terlalu kecil, tidak aman karena kunci dapat ditemukan dengan serangan *exhaustive key search (brute force attack)*
- Jika serangan *exhaustive key search (brute force attack)* dengan menggunakan prosesor paralel, andaikan dalam satu detik dapat dikerjakan satu juta serangan. Jadi seluruhnya diperlukan 1142 tahun untuk menemukan kunci yang benar.

- Misalkan kita mengetahui pasangan potongan plaintext (p) dan cipherteks (c) yang berkoresponden.
- Lakukan serangan *brute force* (atau *exhaustive key search attack*) dengan mencoba semua kemungkinan kunci (k):

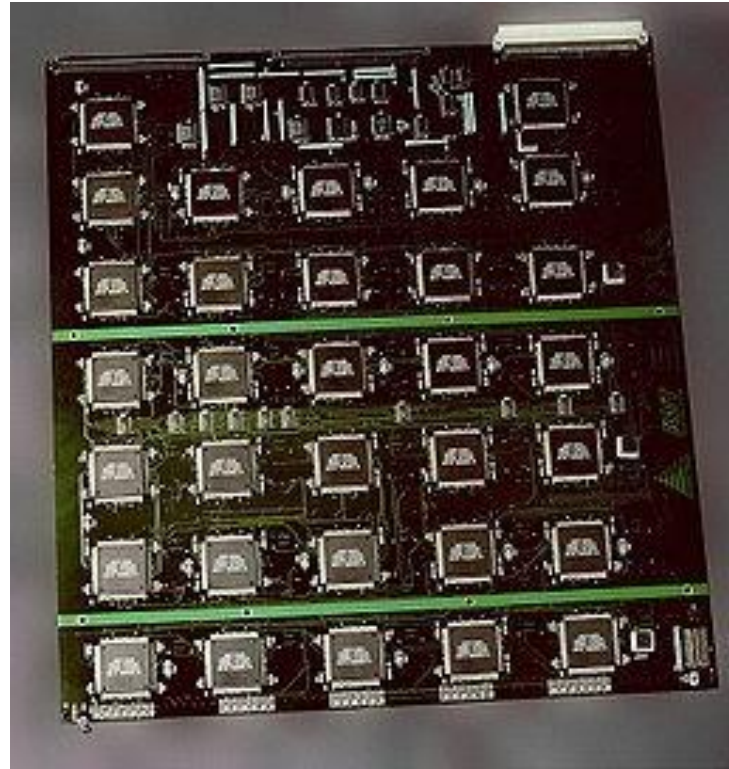
```
for (k=0; k<256; k++) {  
    if (DES(k, p) == c) {  
        printf("Key is %x\n", k);  
        break;  
    }  
}
```

- Kunci diharapkan ditemukan sebelum 2^{55} iterasi.

Dikutip dari Wiki:

- In 1997, [RSA Security](#) sponsored a series of contests, offering a \$10,000 prize to the first team that broke a message encrypted with DES for the contest.
- That contest was won by the [DESCHALL Project](#), led by Rocke Verser, [Matt Curtin](#), and Justin Dolske, using idle cycles of thousands of computers across the Internet.

- Tahun 1998, *Electronic Frontier Foundation (EFE)* merancang dan membuat perangkat keras khusus untuk menemukan kunci DES secara *exhaustive key search* dengan biaya \$250.000 dan diharapkan dapat menemukan kunci selama 5 hari.
- Tahun 1999, kombinasi perangkat keras *EFE* dengan kolaborasi internet yang melibatkan lebih dari 100.000 komputer dapat menemukan kunci DES kurang dari 1 hari.



The [EFF](#)'s US\$250,000 [DES cracking machine](#) contained 1,856 custom chips and could brute force a DES key in a matter of days — the photo shows a DES Cracker circuit board fitted with several Deep Crack chips (Sumber Wikipedia).

- Their motivation was to show that DES was breakable in practice as well as in theory: *"There are many people who will not believe a truth until they can see it with their own eyes. Showing them a physical machine that can crack DES in a few days is the only way to convince some people that they really cannot trust their security to DES."*
- The machine brute-forced a key in a little more than 2 days search.

- Pengisian kotak-S DES masih menjadi misteri. Nilai-nilai awal S-box sudah diubah.
- Sebenarnya delapan putaran sudah cukup untuk membuat cipherteks sebagai fungsi acak dari setiap bit plainteks dan setiap bit kunci.
- Namun dari penelitian dengan melakukan kriptanalisis, DES dengan jumlah putaran yang kurang dari 16 ternyata dapat dipecahkan dengan *known-plaintext attack*.

3DES Triple Data Encryption Standard

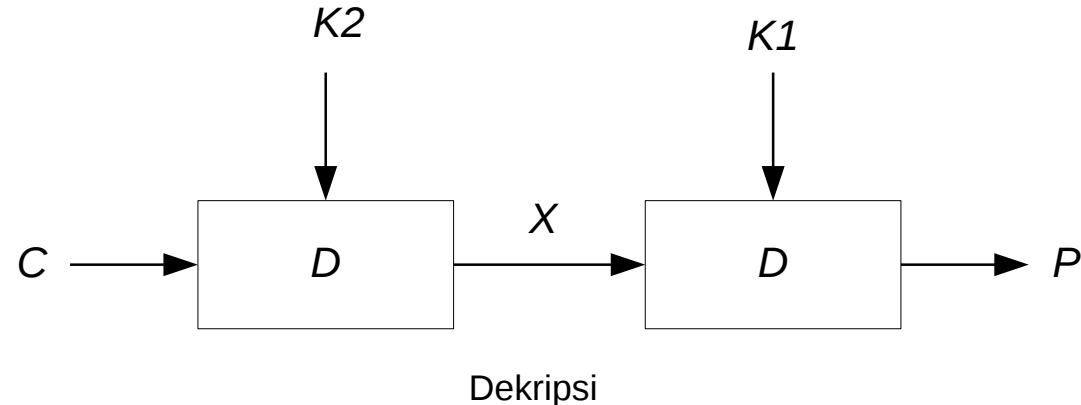
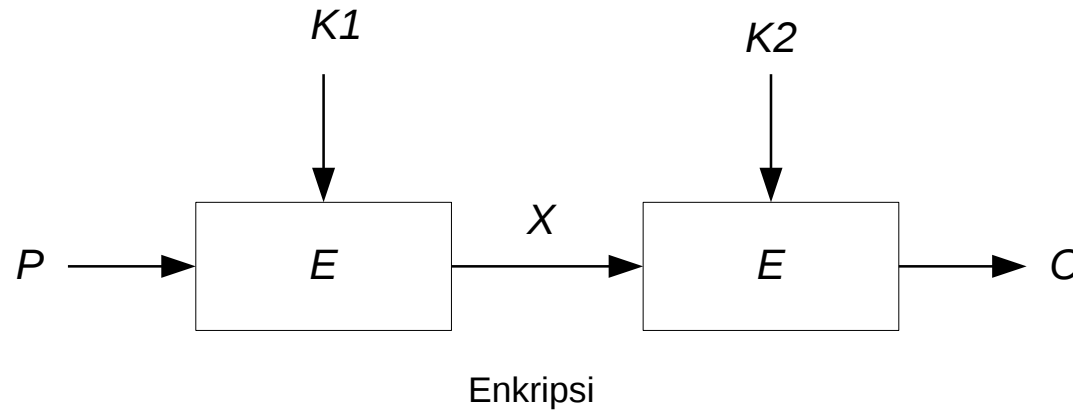
Triple DES (3DES)

DES Berganda

- Karena DES mempunyai potensi kelemahan pada *brute force attack*, maka dibuat varian dari DES.
- Varian DES yang paling luas digunakan adalah DES berganda (*multiple DES*).
- DES berganda adalah enkripsi berkali-kali dengan DES dan menggunakan kunci ganda.
- Tinjau dua macam DES berganda:
 1. *Double DES*
 2. *Triple DES (3DES)*

Double DES

- Menggunakan 2 buah kunci eksternal, K_1 dan K_2 .
- Enkripsi: $C = E_{K_2}(E_{K_1}(P))$
- Dekripsi: $P = D_{K_1}(D_{K_2}(C))$



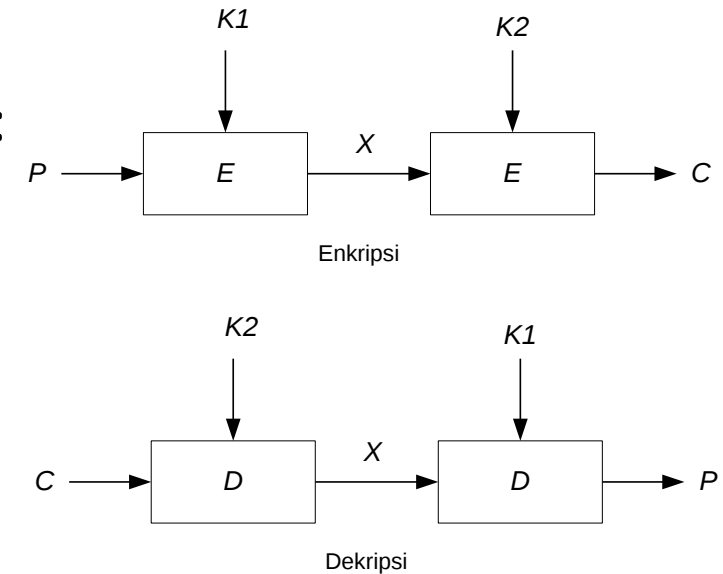
- Kelemahan *Double DES*: serangan *meet-in-the-middle attack*:
- Dari pengamatan,

$$C = E_{K2}(E_{K1}(P))$$

maka

$$X = E_{K1}(P) = D_{K2}(C)$$

- Misalkan kriptanalisis memiliki potongan C dan P yang berkoreponden (*known-plaintext attack*).
- Enkripsi P untuk semua kemungkinan nilai $K1$ (yaitu sebanyak 2^{56} kemungkinan kunci). Hasilnya adalah semua nilai X
- Simpan semua nilai X ini di dalam tabel



- Berikutnya, dekripsi C dengan semua semua kemungkinan nilai $K2$ (yaitu sebanyak 2^{56} kemungkinan kunci).
- Bandingkan semua hasil dekripsi ini dengan elemen di dalam tabel tadi. Jika ada yang sama, maka dua buah kunci, $K1$ dan $K2$, telah ditemukan.
- Tes kedua kunci ini dengan pasangan plainteks-cipherteks lain yang diketahui. Jika kedua kunci tersebut menghasilkan cipherteks atau plainteks yang benar, maka $K1$ dan $K2$ tersebut merupakan kunci yang benar

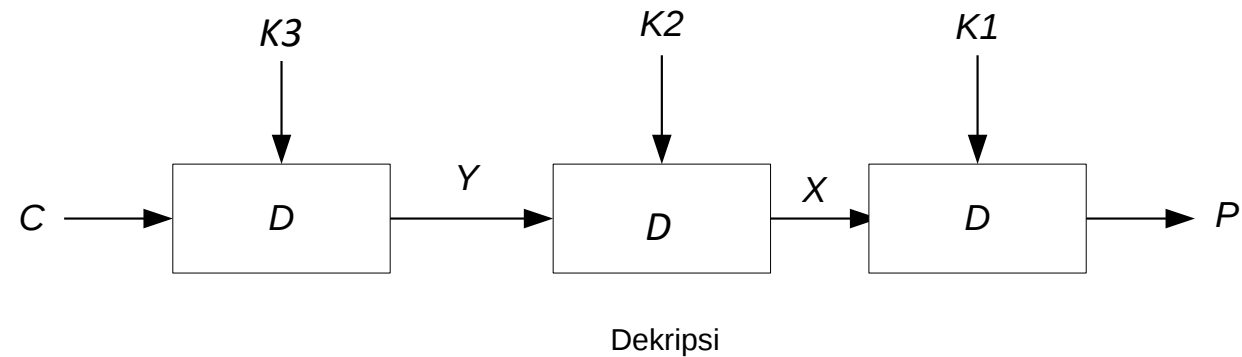
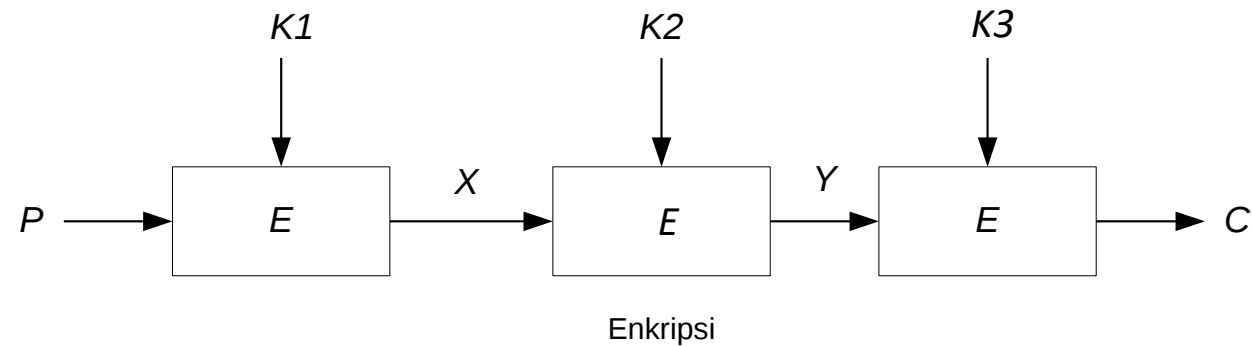
$X = E_{K1}(P)$		$X = D_{K2}(C)$	
K1	X	K2	X
000...00	1010..11	000...00	0110..10
000...01	1101..00	000...01	1000..10
...		...	
...		...	
...		...	
101..10	010..01	101..10	010..01
...		...	
...		...	
...		110..10...	010..01
...		...	
...		...	
111..11	011.. 10	111..11	011.. 10

Triple DES (TDES)

- Menggunakan DES tiga kali
- Bertujuan untuk mencegah *meet-in-the-middle attack*.
- Bentuk umum Triple DES (mode EEE):

Enkripsi: $C = E_{K3}(E_{K2}(E_{K1}(P)))$

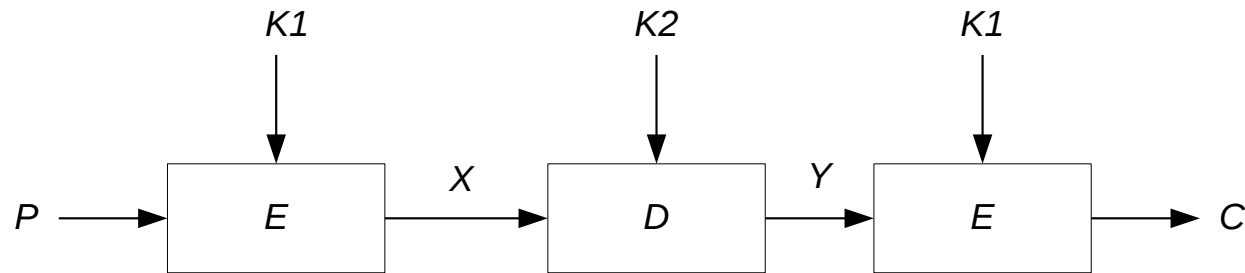
Dekripsi: $P = D_{K1}(D_{K2}(D_{K3}(C)))$



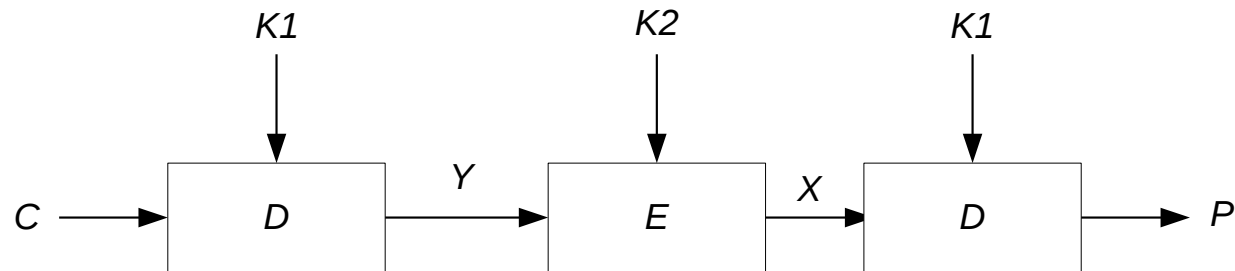
- Untuk menyederhanakan TDES, maka langkah di tengah diganti dengan D (mode EDE).
- Ada dua versi TDES dengan mode EDE:
 - Menggunakan 2 kunci
 - Menggunakan 3 kunci
- Kenapa EDE? Supaya kompatibel dengan single DES:

Triple DES

Triple DES dengan 2 kunci ($56 + 56 = 112$ bit)



Enkripsi



Dekripsi

Triple DES dengan 3 kunci ($56 + 56 + 56 = 168$)

