DATA ENCRYPTION STANDARD

KULIAH KRIPTOGRAFI DAN KEAMANAN JARINGAN

DATA ENCRYPTION STANDARD

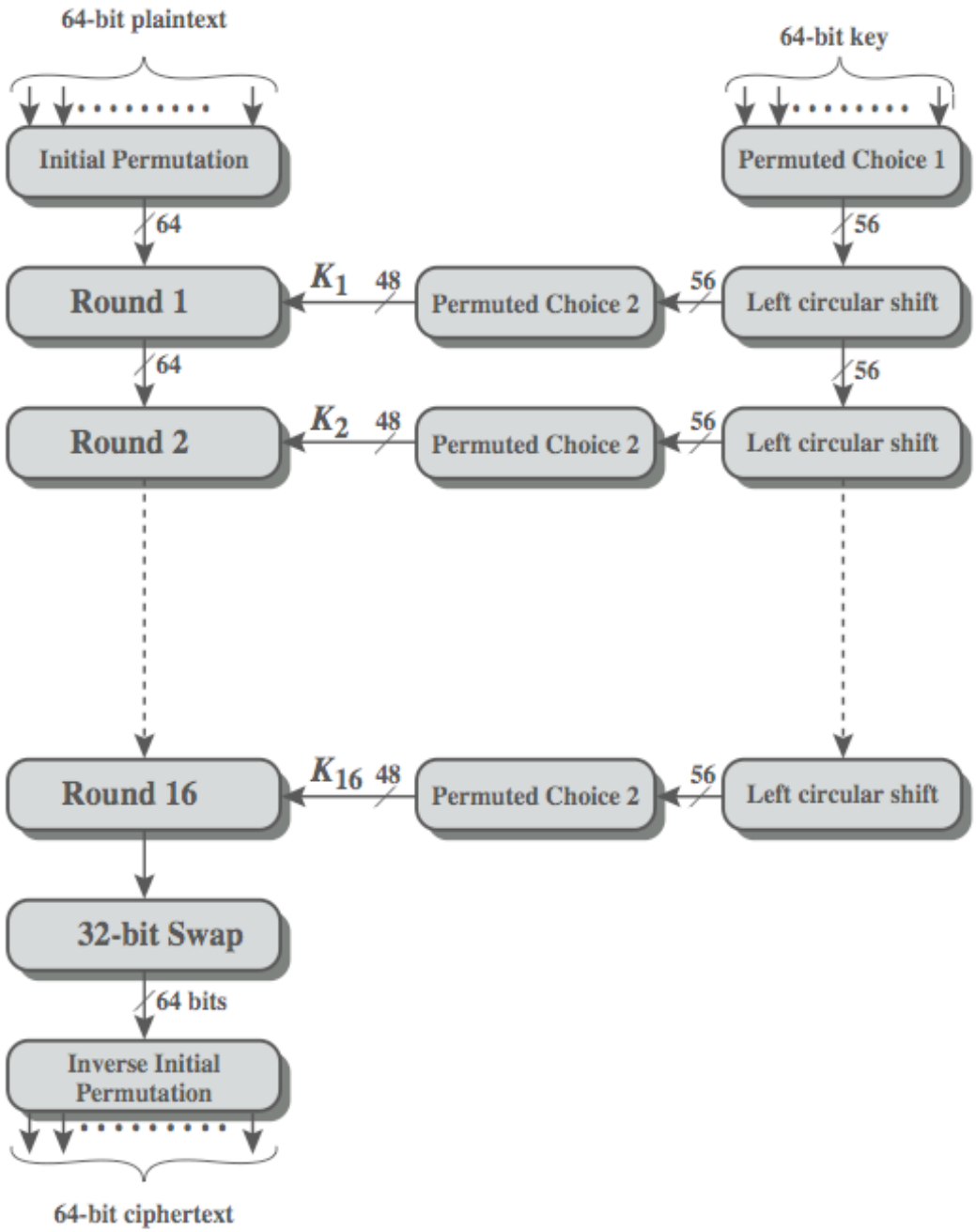
- **Diperkenalkan pertama pada tahun 1977.**
- **Merupakan skema enkripsi yang paling banyak dipakai.**
- **Data dienkripsi dalam blok 64 bit menggunakan kunci 56 bit.**
- **Algoritma mentansformasi input menjadi output dengan sejumlah langkah.**
- **Dekripsi dilakukan dengan kunci yang sama dan langkah yang sama.**
- **Banyak digunakan pada aplikasi-aplikasi finansial.**

SEJARAH

- **IBM membangun Lucifer cipher**
  - Tim diketuai oleh Feistel pada akhir 60-an.
  - Menggunakan 64-bit data blocks dengan 128-bit key
- **Kemudian dibangun kembali menjadi cipher komersial dengan perbaikan dari masukan oleh NSA (National Security Agency) dan lainnya.**
- **Tahun 1973 NBS (National Bureau Standard) meminta proposal untuk standar cipher nasional.**
- **IBM membuat revisi dari Lucifer yang akhirnya diterima sebagai DES.**
- **Pada tahun 1994, NIST merekomendasikan bahwa DES hanya digunakan untuk aplikasi selain yang berkaitan dengan 'classified information'.**
- **Tahun 1999, menyarankan penggunaan 3DES sebagai pengganti DES.**

DES

- **Tiga fase pemrosesan plaintext:**
  - Initial permutation
  - 16 tahap pemrosesan dengan fungsi sama yang meliputi permutasi dan substitusi. Hasil proses tahap terakhir dibalik.
  - Hasilnya dipermutasi yang merupakan invers dari permutasi awal.
- **Hampir sama dengan Feistel cipher, kecuali adanya permutasi.**



- **Tabel-tabel permutasi sudah didefinisikan:**
  - Tabel permutasi awal (IP) → permutasi pada awal proses
  - Tabel invers dari permutasi awal (IP^{-1}) → permutasi setelah semua tahap selesai
  - Permutasi ekspansi (E) → untuk mengekspansi 32 bit menjadi 48 bit (menjadi input pada fungsi F)
  - Fungsi permutasi (P) → melakukan permutasi pada fungsi F
  - Tabel-tabel S-boxes → mengubah 6 bit menjadi 4 bit
  - Tabel-tabel PC-1 dan PC-2 untuk permutasi kunci

TABEL 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

TABEL IP^{-1}

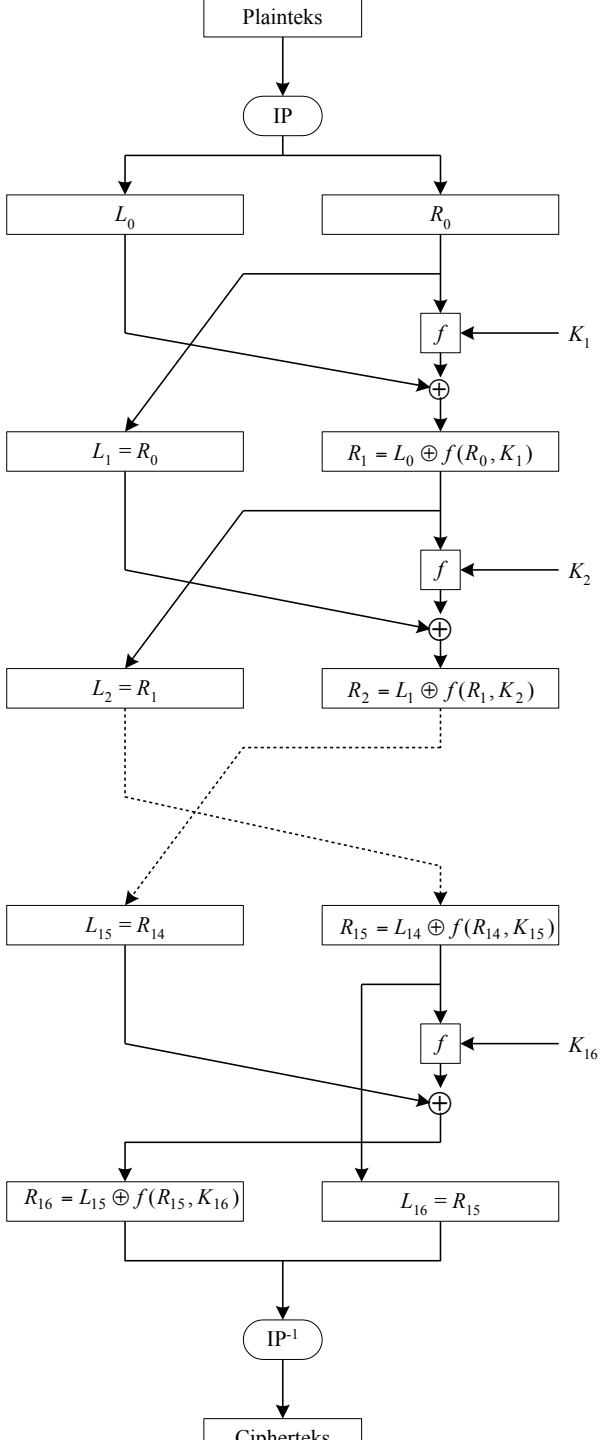
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

DETIL SETIAP ROUND

- **Menggunakan dua 32-bit L & R.**
- **Dapat dideskripsikan seperti pada Feistel cipher :**

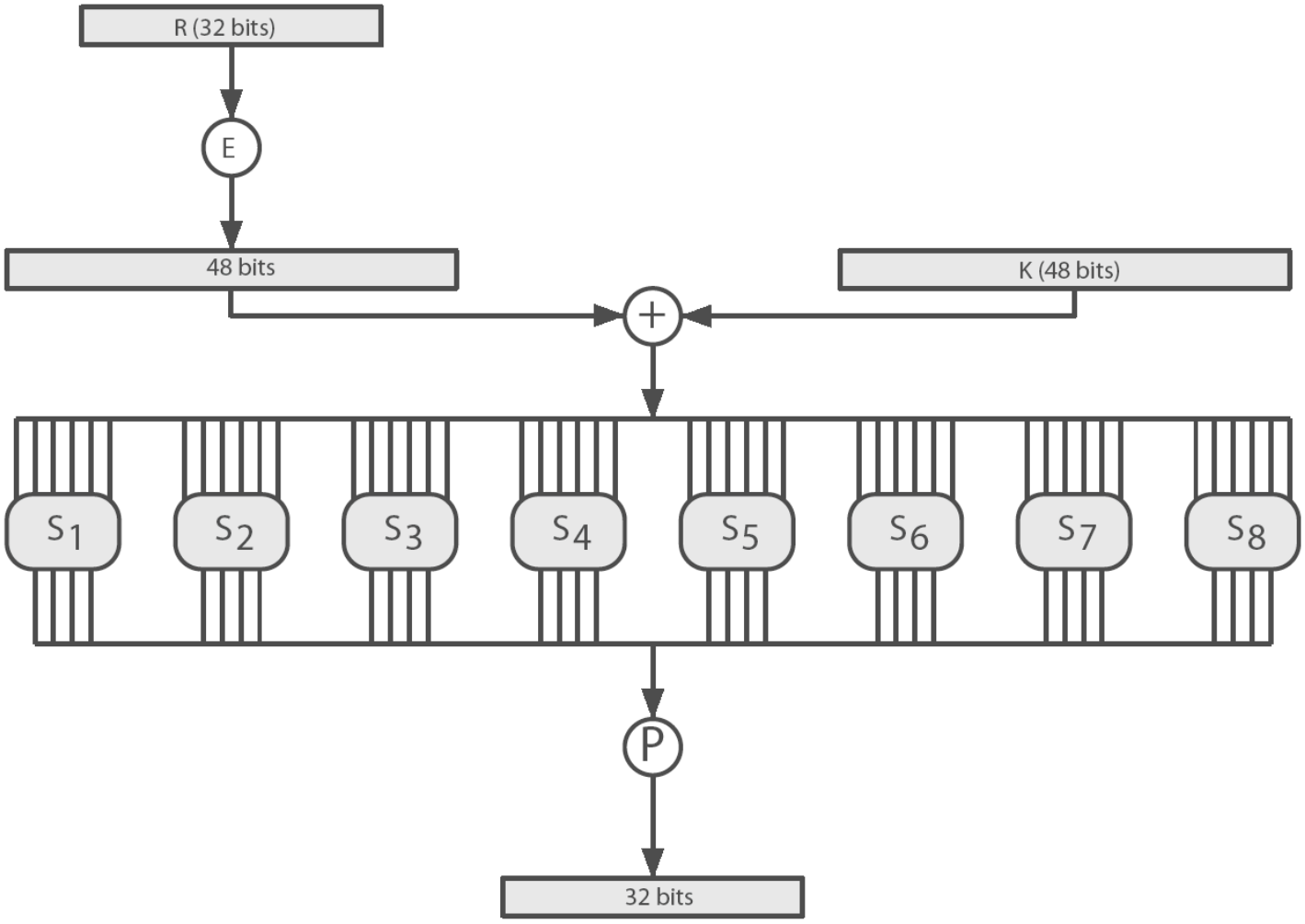
  $L_i = R_{i-1}$

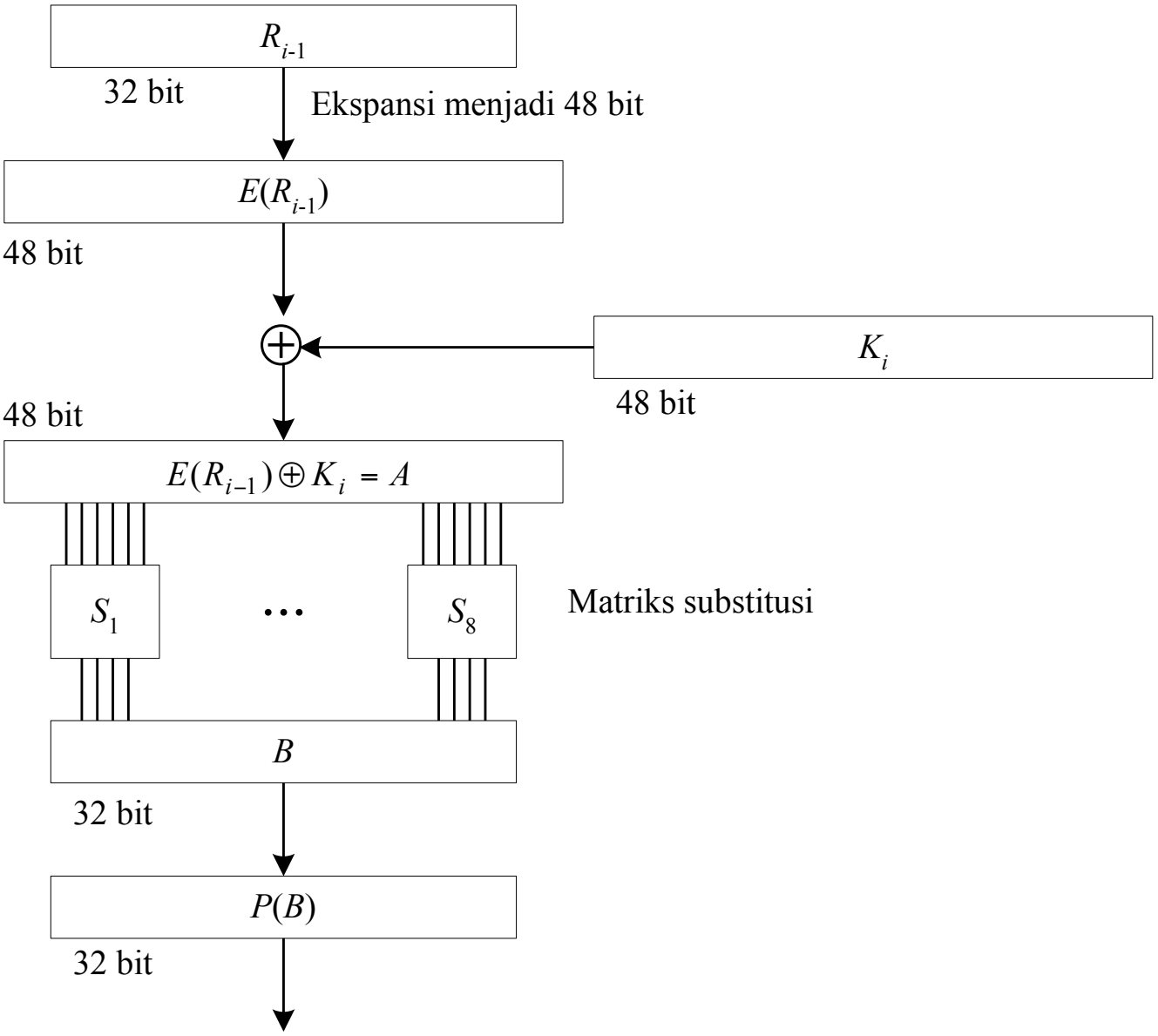
  $R_i = L_{i-1} \oplus \mathrm{F}(R_{i-1}, K_i)$
- **Fungsi F: mengubah input (32 bit R_{i-1} dan 48 bit sub-key) menjadi 32 bit yang dikenai operasi XOR dengan L_{i-1} untuk mendapatkan R_i**



- **Pemrosesan pada kunci:**
  - Untuk setiap tahap, sub-key K_i dihasilkan dengan kombinasi left circular shift dan permutasi.
  - Permutasi sama, tetapi dihasilkan kunci yang berbeda karena adanya left shift.

PENGHITUNGAN F(R,K)




R_{i-1}
32 bit
Ekspansi menjadi 48 bit
$E(R_{i-1})$
48 bit
K_i
48 bit
48 bit
$E(R_{i-1}) \oplus K_i = A$
S_1
...
S_8
Matriks substitusi
B
32 bit
$P(B)$
32 bit

FUNGSI F

- **F mendapatkan input 32-bit R dan 48-bit subkey:**
- **Ekspansi R menjadi 48-bits menggunakan tabel E**
- **Operasikan dengan subkey menggunakan operasi XOR**
- **Masukkan ke 8 S-boxes untuk mendapatkan hasil 32-bit**
  - 48 bit dikelompokkan menjadi 8 buah 6 bit.
  - Tiap kelompok 6 bit menghasilkan 4 bit dengan cara:
    - Gunakan tabel yang bersesuaian (S_1 – S_8)
    - Bit pertama dan terakhir digabung, menjadi 2 bit angka yang menunjukkan baris pada tabel.
    - 4 bit lainnya menunjukkan kolom pada tabel.
    - Angka pada baris dan kolom ybs adalah output dari S-boxes
- **Setelah didapatkan 32 bit, permutasi menggunakan tabel P**

TABEL E (EKSPANSI PERMUTASI)

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

TABEL S-BOX

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

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

TABEL P (FUNGSI PERMUTASI)

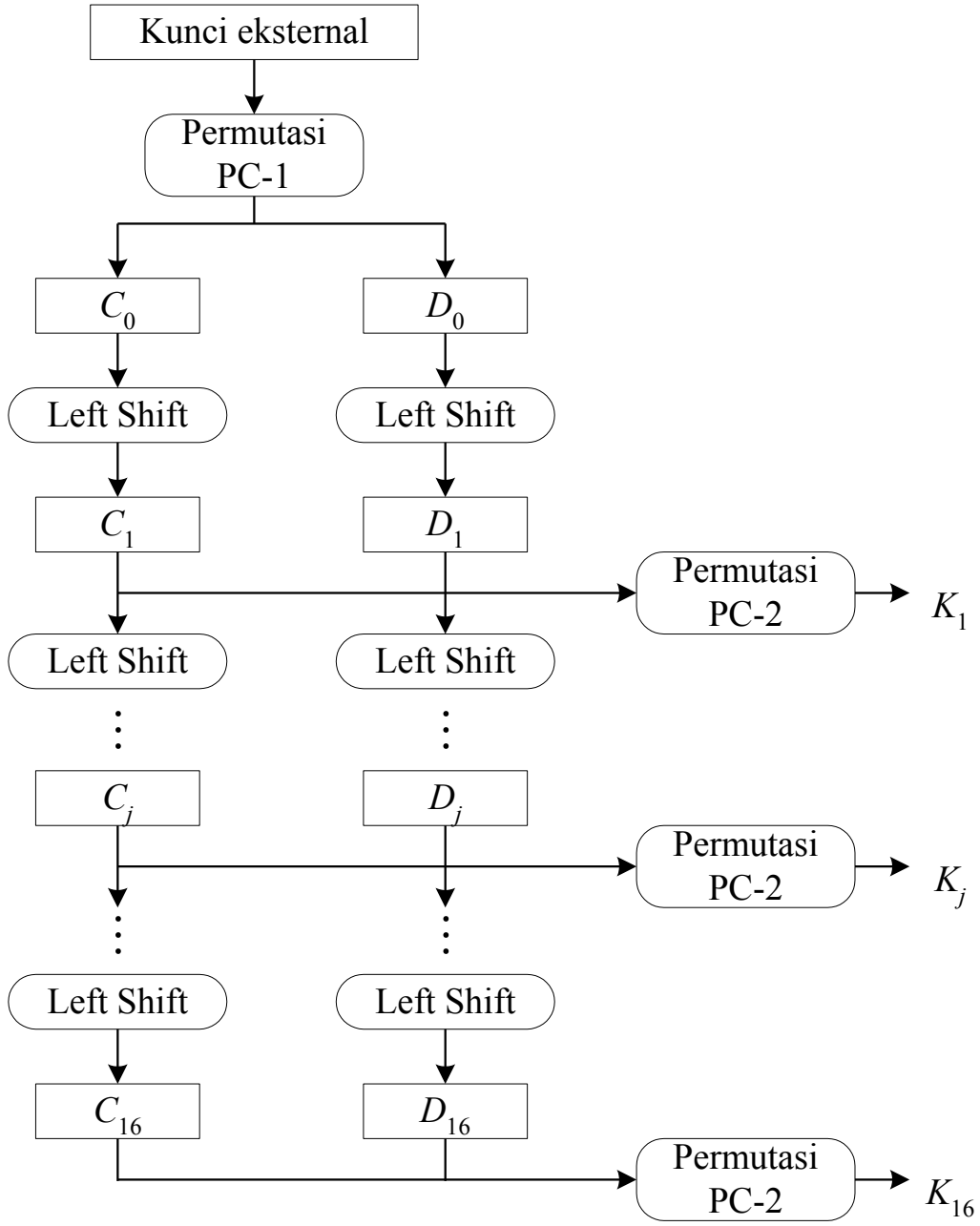
16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

PEMROSESAN KUNCI

- **Proses inisialisasi:**
  - Input: 64 bit
  - Setiap bit ke-8 diabaikan → 56 bits
  - Key dikenai operasi permutasi dengan menggunakan tabel tertentu (tabel PC-1).
- **Proses setiap tahap:**
  - Dibagi menjadi 2 bagian, C_0 dan D_0 masing-masing terdiri dari 28-bit.
  - Masing-masing bagian dikenai operasi circular left shift sebanyak 1 atau 2 bit (setiap round diatur dengan tabel) → menjadi input untuk tahap berikutnya.
  - Kunci dikenai operasi permutasi dengan tabel permutasi yang kedua (tabel PC-2) → menjadi input untuk $F(R_{i-1}, K_i)$

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

ILUSTRASI PEMROSESAN KUNCI



Matriks permutasi kompresi PC-1:

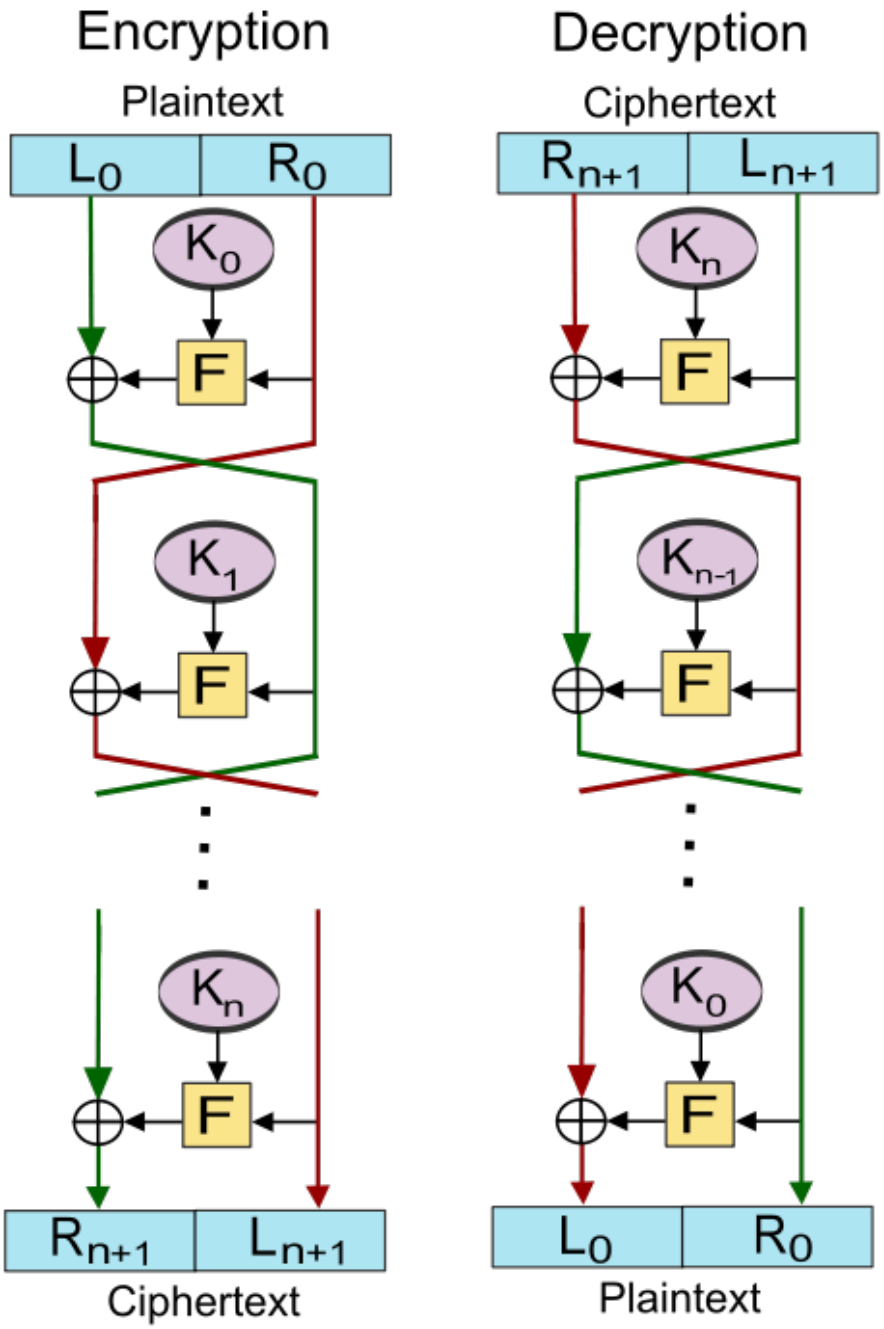
57	49	41	33	25	17	9	1	58	50	42	34	26	18
10	2	59	51	43	35	27	19	11	3	60	52	44	36
63	55	47	39	31	23	15	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

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

DEKRIPSI

- **Menggunakan algoritma yang sama dengan enkripsi.**
- **Penggunaan sub-key dibalik.**



KEUNTUNGAN DES

- **56-bit kunci memiliki $2^{56} = 7.2 \times 10^{16}$ nilai**
- **Brute force search sulit, tetapi mungkin**
  - Tahun 1997 selama beberapa bulan (menggunakan idle cycles dari ribuan komputer di Internet)
  - Tahun 1998 pada dedicated h/w perlu 5 hari (oleh *Electronic Frontier Foundation* (*EFE*))
  - Tahun 1999 perlu kurang dari 1 hari.
- **Tetapi harus dapat mengenali plaintext.**

KRITISI TERHADAP DES

- **Banyak yang mengkritisi keputusan dijadikannya DES sebagai standard.**
- **Alasan utama:**
  - Ukuran kunci (56 bits) terlalu kecil untuk organisasi yang memiliki sumber daya yang mencukupi. Paling tidak, exhaustive keysearch dimungkinkan dengan panjang kunci seperti itu.
  - Kriteria desain untuk tabel yang digunakan pada fungsi f tidak diketahui. Tes statistik menunjukkan bahwa tabel ini tidak benar-benar random.
- **Meskipun demikian, selama 20 tahun pertama sejak menjadi standar, tidak ada publikasi yang menjelaskan cara efektif untuk memecahkan DES.**

LATIHAN

- **Gunakan 1 round DES dengan kunci: 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1$_H$**

  untuk mengenkripsi plaintext:

  a. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0$_H$
  b. 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1$_H$

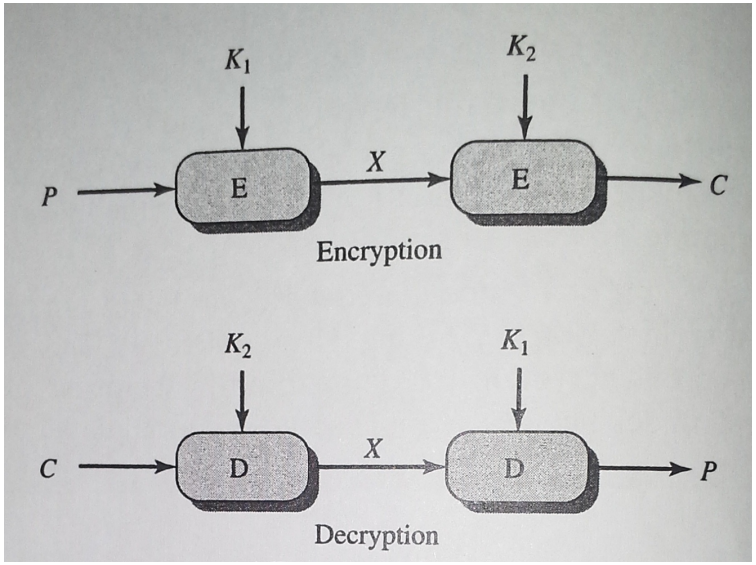
1. **Apakah kunci K1?**
2. **Tunjukkan hasil setelah permutasi awal untuk setiap plaintext.**
3. **Tunjukkan hasil setelah tahap 1 selesai untuk setiap plaintext.**

BAGAIMANA PERKEMBANGAN DES?

- **DES dapat didekripsi:**
  - Pada Juli 1998, the Electronic Frontier Foundation (EFF) mengumumkan bahwa mereka berhasil memecahkan enkripsi DES.
  - Sebuah mesin "pemecah DES" dibuat dengan dana kurang dari $250,000.
  - Waktu yang diperlukan kurang dari 3 hari.
  - EFF mempublikasikan deskripsi detil tentang mesin tersebut.
  - Plaintext harus diketahui → EFF memperkenalkan teknik untuk membedakan plaintext dengan gangguan
- **Untuk menghindari brute force analysis: gunakan kunci yang lebih panjang**
  - Kunci 128 bit memerlukan waktu 10^{18} tahun untuk dipecahkan.

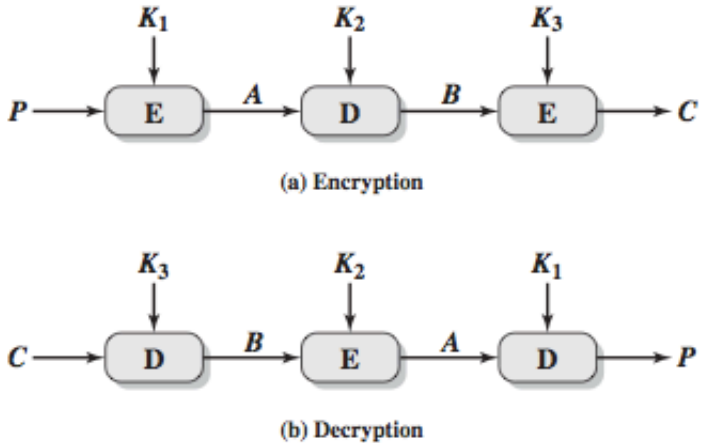
DOUBLE DES

- **Dua tahap enkripsi dan 2 kunci.**
- **Algoritma enkripsi:**
  - $C = E(K_2, E(K_1, P))$
  - P adalah plaintext, K_1 dan K_2 adalah kunci, C adalah chipertext.
- **Algoritma dekripsi:**
  - $P = D(K_1, D(K_2, C))$
- **Panjang kunci: 56 x 2 = 112 bit**



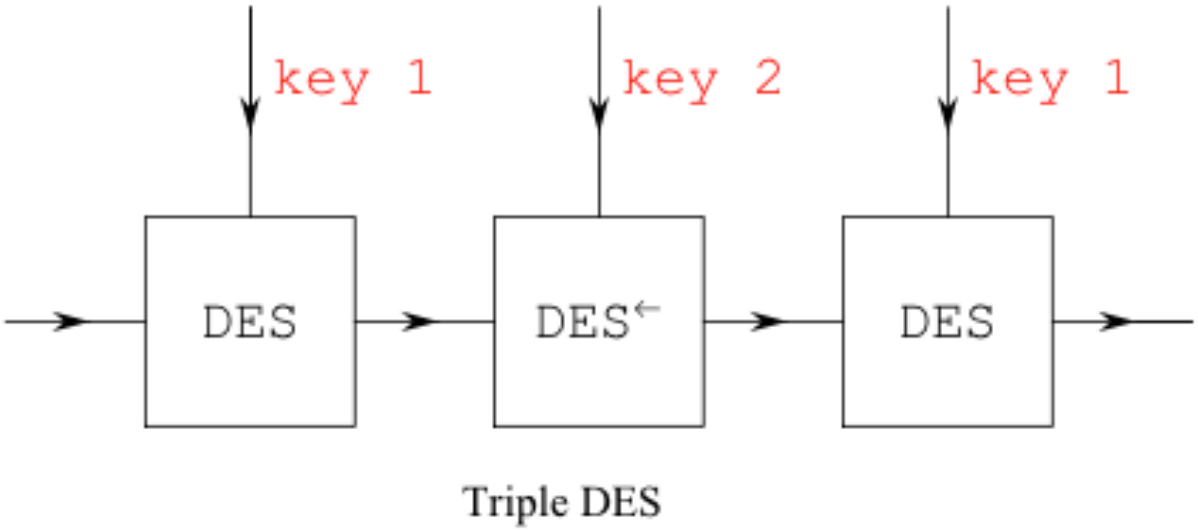
TRIPLE DES

- **Distandarisasi untuk penggunaan pada aplikasi finansial dalam ANSI standard X9.17 pada tahun 1985.**
- **Menggunakan 3 kunci dan 3 eksekusi dari algoritma DES.**
  - Fungsi menggunakan model sekuens encrypt-decrypt-encrypt (EDE) : $C = E(K3, D(K2, E(K1, P)))$
    - C = ciphertext
    - P = plaintext
    - $E[K, X]$ = enkripsi X menggunakan kunci K
    - $D[K, Y]$ = dekripsi Y menggunakan kunci K
- **Dekripsi merupakan operasi yang sama dengan urutan kunci dibalik:**
    - $P = D(K1, E(K2, D(K3, C)))$



3DES AWAL

- **Kunci 3 = kunci 1**



Triple DES

- **Tidak ada peningkatan keamanan kriptografi yang signifikan dari sisi algoritma.**
  - Keuntungannya adalah memungkinkan pengguna 3DES untuk mendekripsi data yang dienkripsi oleh pengguna versi lama DES (single DES):
    - $C = E(K1, D(K1, E(K1, P))) = E[K, P]$
- **3DES menggunakan panjang kunci 168 bit.**
  - Dimungkinkan juga penggunaan $K_1 = K_3$ sehingga panjang kunci 112 bit.

GUIDELINE PENGGUNAAN 3DES

- **Penggunaan 3DES menurut FIPS (Federal Information Processing Standard) :**
  - 3DES merupakan algoritma enkripsi simetris yang disetujui untuk digunakan.
  - Original DES (56 bit key) hanya diijinkan untuk sistem lama, sistem baru harus menggunakan 3DES.
  - Organisasi pemerintah didorong untuk melakukan transisi ke 3DES.
  - 3DES dan AES akan bersama-sama ada sebagai algoritma yang disetujui oleh FIPS, yang memungkinkan transisi gradual ke AES.

KEUNTUNGAN DAN KELEMAHAN 3DES

- **Keuntungan :**
  - Dengan kunci 168-bit, mencegah kelemahan DES dalam brute-force attack.
  - Algoritma enkripsi sama dengan DES.
- **Kelemahan :**
  - Algoritma lambat→ lebih lambat daripada DES.
  - DES dan 3DES menggunakan blok 64-bit. Untuk alasan keamanan dan efisiensi, ukuran blok yang lebih besar lebih disukai.