Bahan kuliah II4021 Kriptografi

Block Cipher

(Bagian 2)

Oleh: Rinaldi Munir

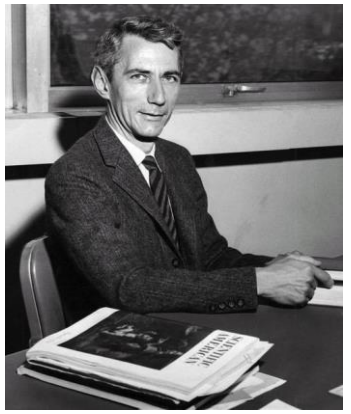
Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2025

Perancangan *Cipher* Blok

Desain sebuah cipher blok seharusnya memperhatikan konsep berikut:

1. Prinsip *Confusion* dan *Diffusion* dari Shannon.
2. Substitusi dan permutasi
3. *Cipher* berulang (*iterated cipher*)
4. Pembangkitan kunci putaran
5. Jaringan Feistel (*Feistel Network*)
6. Ukuran blok dan kunci

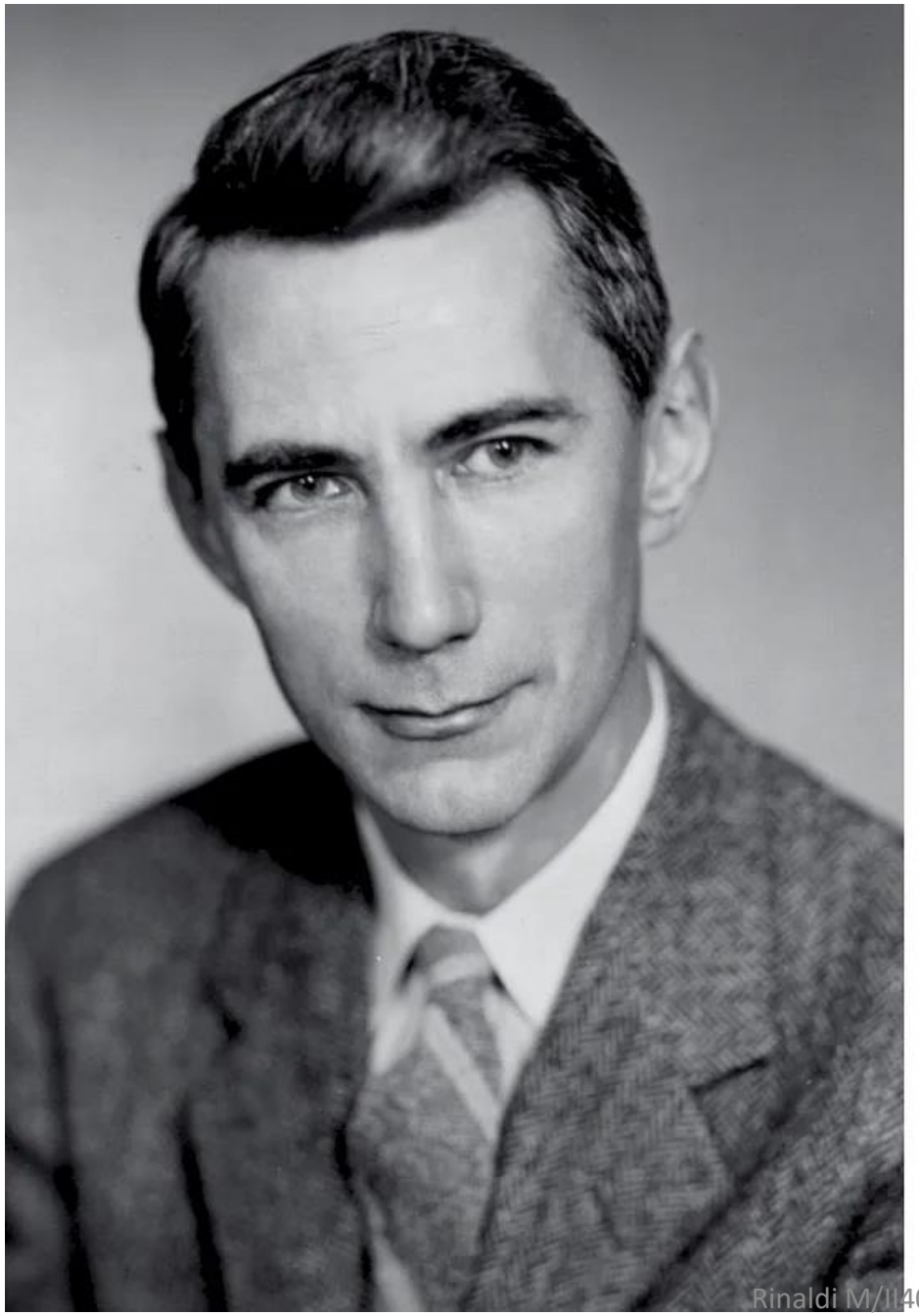
Prinsip *Confusion* dan *Diffusion* dari Shannon.



- Sudah banyak cipher klasik yang telah berhasil dipecahkan karena hubungan statistik antara plainteks dan cipherteks berhasil diekspolitasi untuk mengkriptanalisis cipher.

- Claude Shannon dalam makalah klasiknya tahun 1949, *Communication theory of secrecy systems*, memperkenalkan prinsip *confusion* dan *diffusion* untuk membuat serangan berbasis statistik menjadi lebih sulit dilakukan.

- Dua prinsip tersebut, *confusion* dan *diffusion*, menjadi panduan dalam merancang algoritma kriptografi, baik cipher alir maupun cipher blok.



Communication Theory of Secrecy Systems*

By C. E. SHANNON

1. Introduction and Summary

THE problems of cryptography and secrecy systems furnish an interesting application of communication theory.[1] In this paper a theory of secrecy systems is developed. The approach is on a theoretical level and is intended to complement the treatment found in standard works on cryptography.[2] There, a detailed study is made of the many standard types of codes and ciphers, and of the ways of breaking them. We will be more concerned with the general mathematical structure and properties of secrecy systems.

The treatment is limited in certain ways. First, there are three general types of secrecy system: (1) concealment systems, including such methods as invisible ink, concealing a message in an innocent text, or in a fake covering cryptogram, or other methods in which the existence of the message is concealed from the enemy; (2) privacy systems, for example speech inversion, in which special equipment is required to recover the message; (3) "true" secrecy systems where the meaning of the message is concealed by cipher, code, etc., although its existence is not hidden, and the enemy is assumed to have any special equipment necessary to intercept and record the transmitted signal. We consider only the third type—concealment systems are primarily a psychological problem, and privacy systems a technological one.

Secondly, the treatment is limited to the case of discrete information, where the message to be enciphered consists of a sequence of discrete symbols, each chosen from a finite set. These symbols may be letters in a language, words of a language, amplitude levels of a "quantized" speech or video signal, etc., but the main emphasis and thinking has been concerned with the case of letters.

The paper is divided into three parts. The main results will now be briefly summarized. The first part deals with the basic mathematical structure of secrecy systems. As in communication theory a language is considered to

* The material in this paper appeared originally in a confidential report "A Mathematical Theory of Cryptography" dated Sept. 1, 1945, which has now been declassified.

[1] Shannon, C. E., "A Mathematical Theory of Communication," *Bell System Technical Journal*, July 1948, p. 379; Oct. 1948, p. 623.

[2] See, for example, H. F. Gaines, "Elementary Cryptanalysis," or M. Givierge, "Cours de Cryptographie."

Confusion

- Prinsip ini menyembunyikan hubungan statistik yang ada diantara plainteks, cipherteks, dan kunci.

- Prinsip *confusion* membuat kriptanalis frustasi untuk mencari pola-pola statistik yang muncul di dalam cipherteks.

- *One-Time Pad* adalah contoh algoritma enkripsi yang menerapkan *confusing*.

- *Confusion* dapat direalisasikan dengan menggunakan teknik substitusi yang nirlanjar (*non-linear*), biasanya dengan cara *lookup table*.

Diffusion

- Prinsip ini menyebarkan pengaruh satu bit plainteks atau kunci ke sebanyak mungkin bit-bit cipherteks.

  Contoh: satu huruf plainteks mempengaruhi nilai banyak huruf cipherteks

- Sebagai efek difusi, maka pengubahan kecil pada plainteks sebanyak satu atau dua bit menghasilkan perubahan pada bit-bit cipherteks yang tidak dapat diprediksi.

- Misalkan mengenkripsi plainteks 1111111111111111 menghasilkan cipherteks 0110110000101001. Kemudian enkripsi 1111111011111111, tidak dapat memprediksi apapun tentang cipherteks.

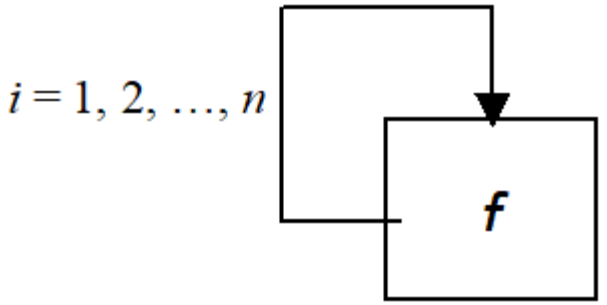
- *Diffusion* dapat direalisasikan dengan menggunakan teknik permutasi atau transposisi secara berulang-ulang. Permutasi adalah mengacak susunan bit atau *byte* di dalam plainteks.

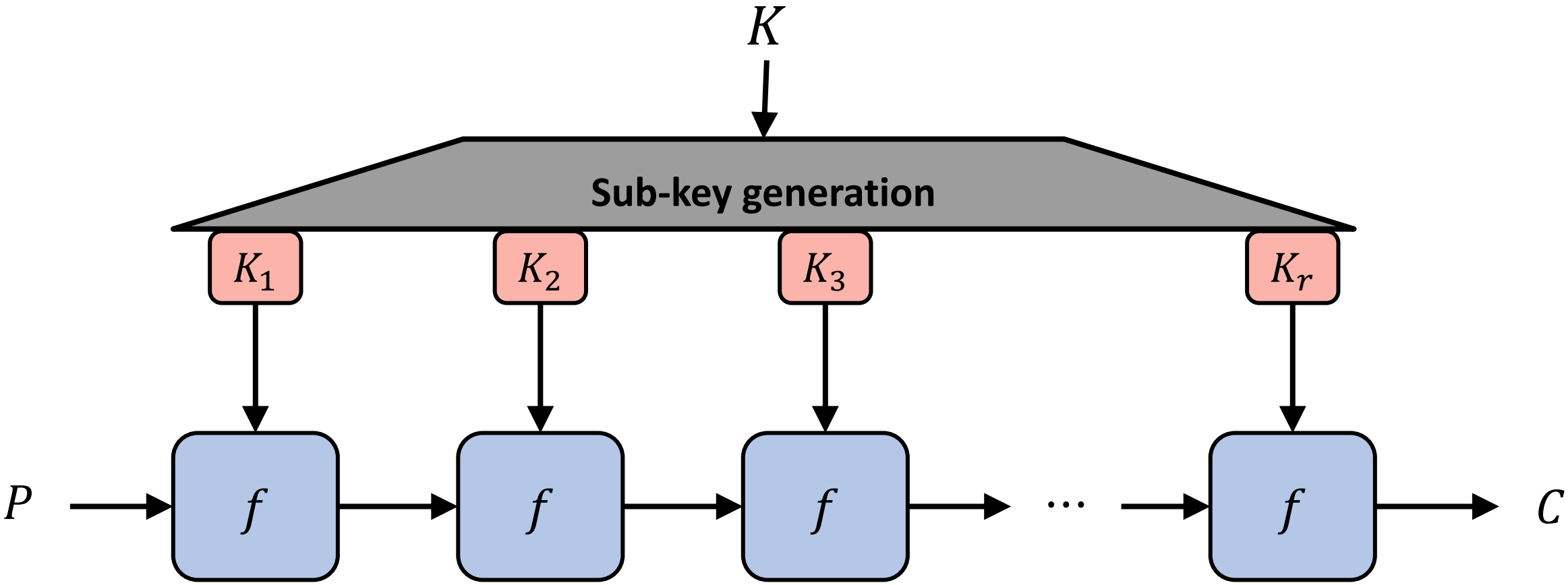
Cipher Berulang (*Iterated Cipher*)

- Untuk menghasilkan *cipher* yang lebih kuat, maka dilakukan *enciphering* sejumlah kali (sejumlah putaran) .

- Caranya adalah dengan melakukan berulang kali suatu fungsi transformasi *f* (disebut juga fungsi putaran atau *round function*) yang mengubah blok plainteks menjadi blokcipherteks.

- Pada setiap putaran digunakan upa-kunci (*subkey*) atau kunci putaran (*round key*) yang dikombinasikan dengan plainteks.





$f(K_i, P)$ dinamakan fungsi putaran, *r* adalah jumlah putaran

DES: $r = 16$

AES-128/192/256: $r = 10/12/14$

8

- *Cipher* berulang dinyatakan sebagai

$$C_i = f(C_{i-1}, K_i)$$

yang dalam hal ini,

i = 1, 2, …, r (r adalah jumlah putaran).

K_i = upa-kunci (*subkey*) pada putaran ke-i

f = fungsi transformasi (round function), di dalamnya terdapat operasi substitusi, permutasi, dan/atau ekspansi, kompresi).

- Plainteks dinyatakan dengan C_0 dan cipherteks dinyatakan dengan C_r.

- Biasanya, plainteks dimanipulasi dulu dengan kunci (umumnya di-XOR-kan dengan kunci eksternal, atau dengan kunci putaran pertama), lalu hasilnya ditransformasikan dengan fungsi *f* berulang kali.

$$C_0 = P \oplus K_0$$
$$C_i = f(C_{i-1}, K_i) \quad , i = 1, 2, \ldots, r-1$$
$$C = C_r = C_{r-1} \oplus K_r$$

- Semakin banyak jumlah putaran, efek diffusi semakin bertambah, namun jumlah putaran yang besar dapat mempuat komputasi menajdi tidak sangkil (efisien).

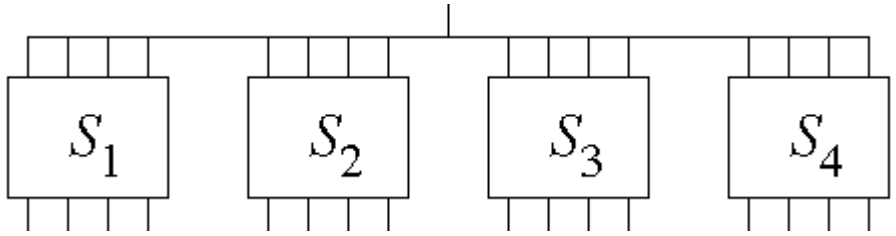
- Jumlah putaran harus dipertimbangkan untuk mempertemukan kriteria keamanan dan efisiensi.

Teknik Substitusi

- Operasi substitusi menerima input sejumlah bit (atau byte) berukuran m_1 lalu menghasilkan sejumlah bit berukuran m_2 $(m_2 \geq m_1)$

- Biasanya operasi substitusi direalisasikan menggunakan sejumlah kotak S (S-box).

- Kotak-*S* adalah matriks yang berisi substitusi sederhana yang memetakan sejumlah bit berukuran m_1 menjadi sejumlah bit berukuran m_2 .



- Kotak-*S* merupakan satu-satunya proses nirlanjar di dalam *cipher*, karena operasinya adalah *look-up table*.

- Masukan dari operasi *look-up table* dijadikan sebagai indeks kotak-*S*, dan luarannya adalah *entry* di dalam kotak-*S*.

- Kotak S yang menerima input m_1 bit dan menghasilkan output m_2 bit dinyatakan sebagai $m_1 \times m_2$ *S-box.*

- Nilai-nilai di dalam S-box dibangkitkan dari suatu perhitungan, jadi bukan berupa bilangan acak.

Contoh: Kotak-*S* di dalam algoritma *DES* adalah 6×4 *S-box* yang berarti memetakan 6 bit masukan menjadi 4 bit keluaran. Salah satu kotak-*S* yang ada di dalam algoritma DES adalah sebagai berikut:

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

Baris diberi nomor dari 0 sampai 3
Kolom diberi nomor dari 0 sampai 15

Masukan untuk proses substitusi adalah 6 bit,

$$b_1b_2b_3b_4b_5b_6$$

Nomor baris dari tabel ditunjukkan oleh *string* bit b_1b_6
(menyatakan 0 sampai 3 desimal)

Nomor kolom ditunjukkan oleh *string* bit $b_2b_3b_4b_5$
(menyatakan 0 sampai 15)

- Misalkan 6-bit masukan adalah 110100

  Nomor baris tabel = 10 (baris 2)

  Nomor kolom tabel = 1010 (kolom 10)

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

  Jadi, substitusi untuk 110100 adalah *entry* pada baris 2 dan kolom 10, yaitu 0100 (atau 4 desimal).

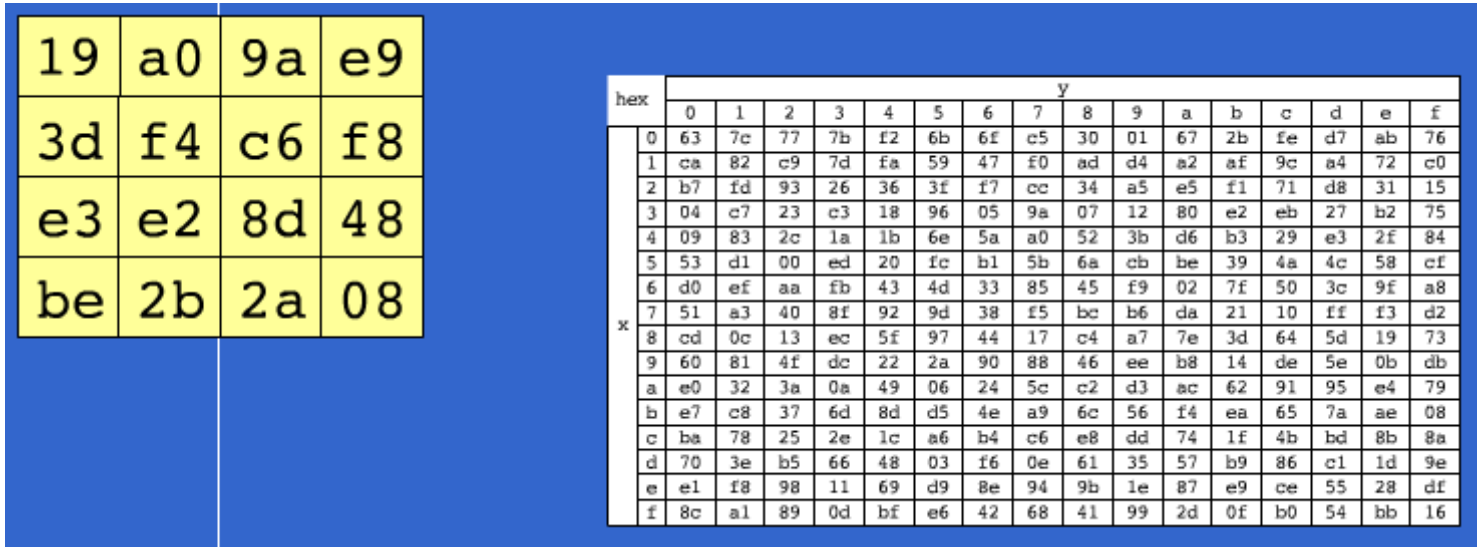
- DES mempunyai 8 buah kotak-S

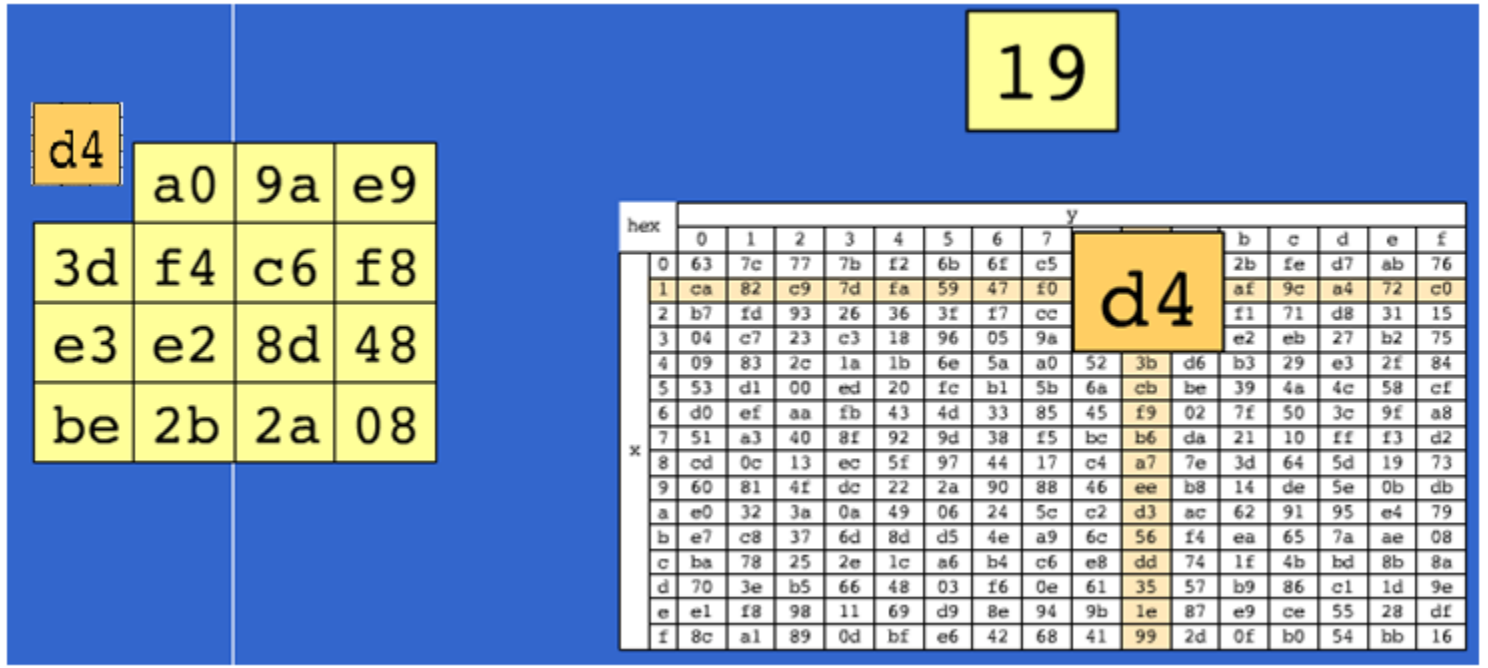
- Di dalam AES kotak S hanya ada satu buah:

	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

Input: 32
Output: 26

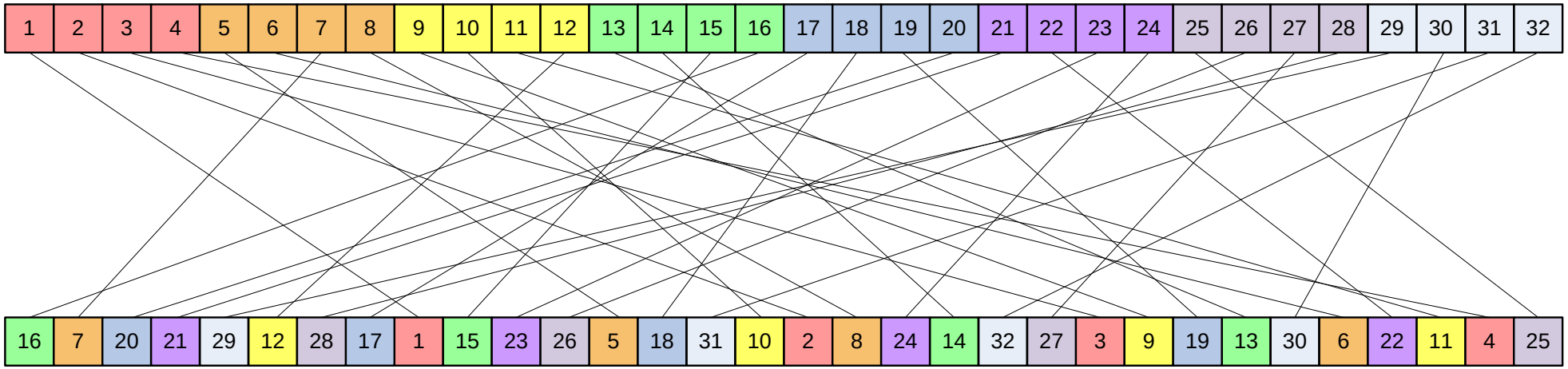
Cara substitusi;
- Cari perpotongan baris 2 dengan kolom 3
- hasilnya adalah 26


19 a0 9a e9
3d f4 c6 f8
e3 e2 8d 48
be 2b 2a 08
hex
y
x
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


19
d4
a0 9a e9
3d f4 c6 f8
e3 e2 8d 48
be 2b 2a 08
hex
y
x
d4
0 1 2 3 4 5 6 7 b c d e f
0 63 7c 77 7b f2 6b 6f c5 2b fe d7 ab 76
1 ca 82 c9 7d fa 59 47 f0 af 9c a4 72 c0
2 b7 fd 93 26 36 3f f7 cc f1 71 d8 31 15
3 04 c7 23 c3 18 96 05 9a 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

Teknik Permutasi

- Permutasi adalah teknik untuk mengacak susunan bit di dalam sebuah blok bit menghasilkan susunan baru.



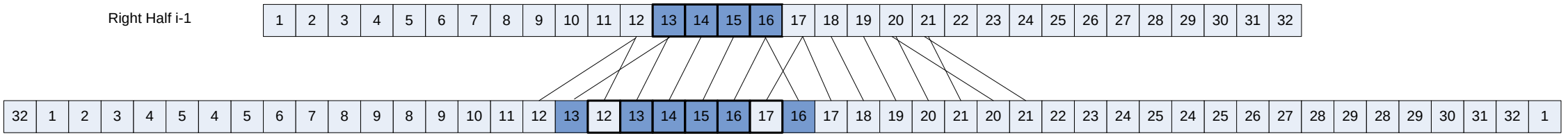
- Pada gambar di atas, posisi bit-bit di dalam blok input 32-bit dipermutasi menjadi susunan baru.

- Di dalam beberapa cipher blok, permutasi direalisasikan dengan menggunakan matriks permutasi yang dinamakan P-box. Entri di dalam P-box menyatakan bit dari posisi sebelumnya.
- Contoh P-box di dalam DES:

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

Artinya, bit ke 58 dari bit-bit input dipindahkan ke posisi pertama, bit ke-50 pada posisi kedua, dst

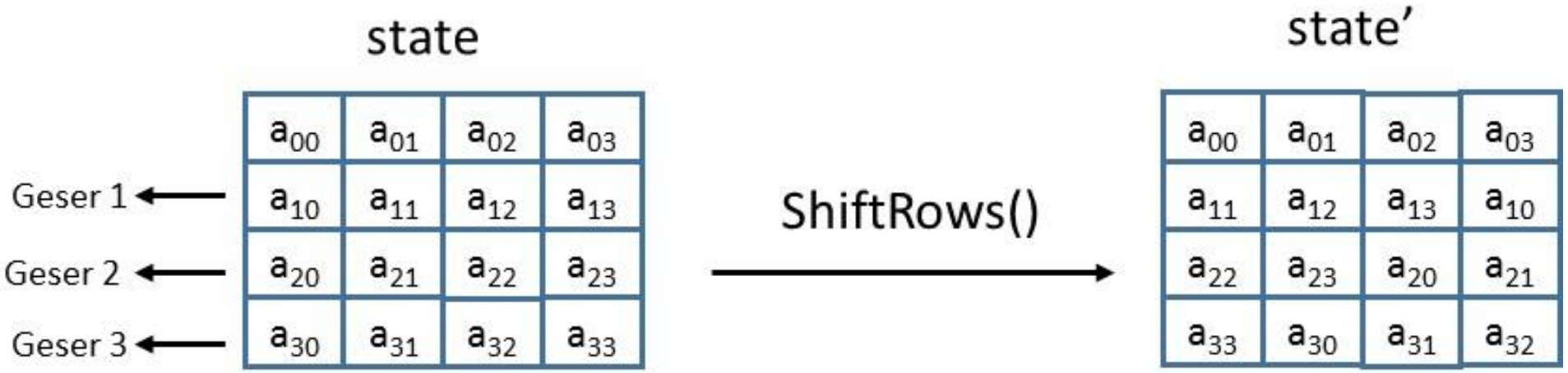
- Beberapa skema permutasi juga melakukan kompresi (mengurangi bit-bit input) atau ekspansi (memperbanyak bit-bit) pada operasi permutasi.



- Sedangkan di dalam *cipher* blok yang lain, permutasi tidak menggunakan P-box, tetapi melakukan pergeseran bit ke kiri atau ke kanan sejauh k bit.

- Pergeseraran bersifat siklik, yaitu bit yang paling ujung muncul pada ujung yag lain

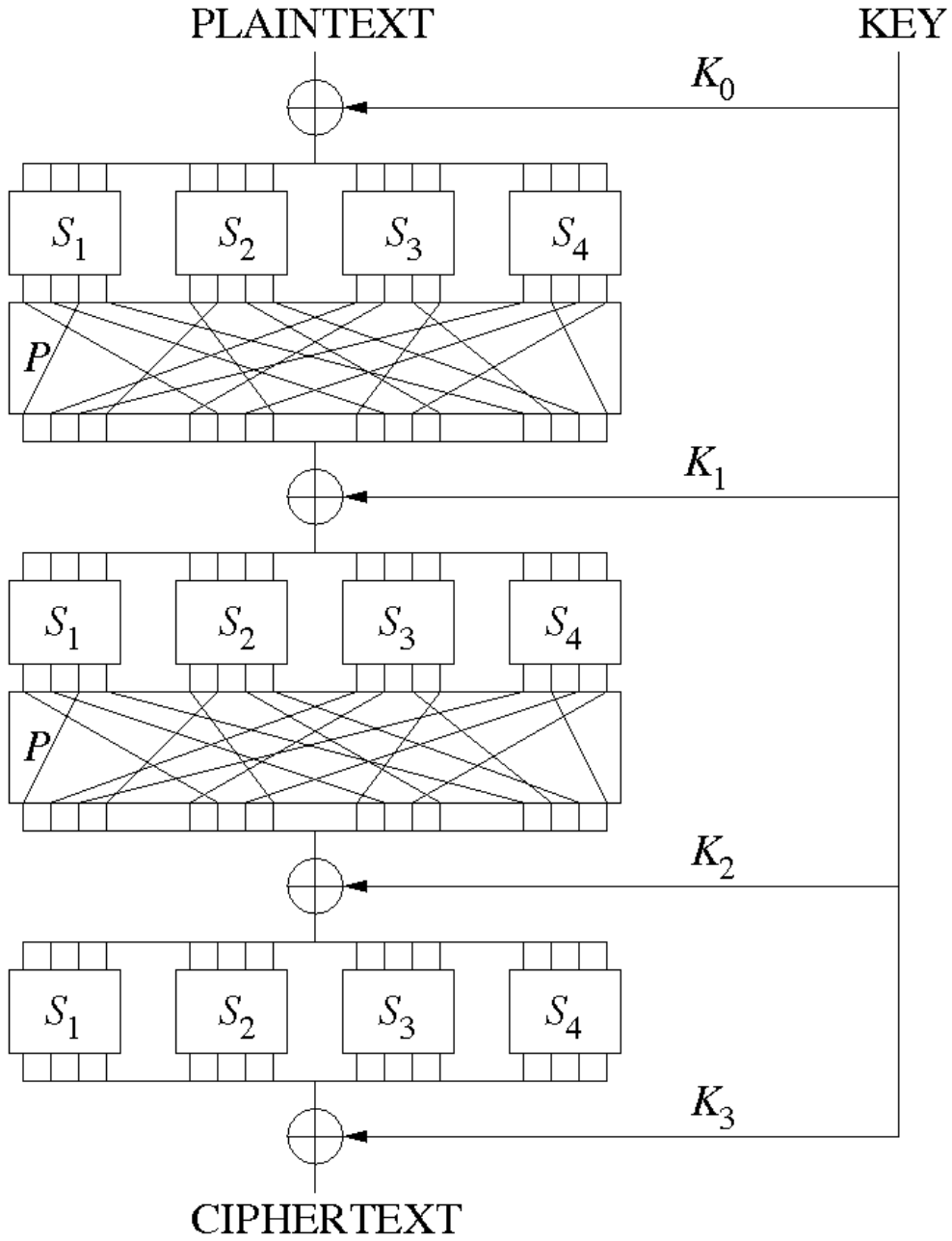
- Contoh pergeseran byte di dalam AES:



- Di dalam beberapa *cipher* blok, operasi substitusi diikuti dengan permutasi, sehingga membentuk jaringan yang dinamakan jaringan substitusi-permutasi.

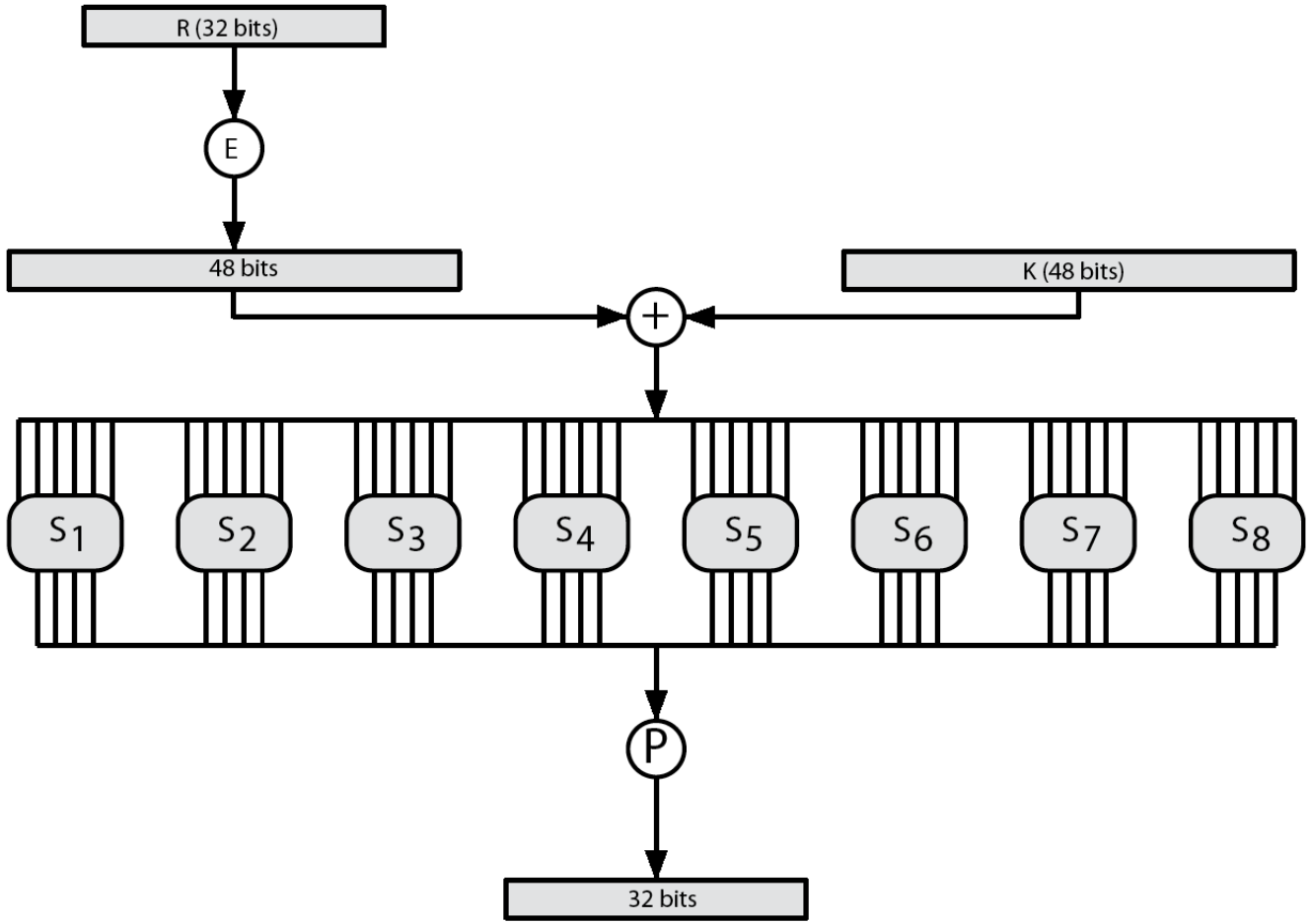
- Operasi substitusi menghasilkan efek *confusion*, sedangkan operasi permutasi menghasilkan efek *diffusion*.

- Operasi permutasi dan substitusi dilakukan pada setiap putaran. Setiap putaran menggunakan kunci putaran yang berbeda-beda.



https://commons.wikimedia.org/w/index.php?curid=6420152

Jaringan substitusi-permutasi di dalam DES:



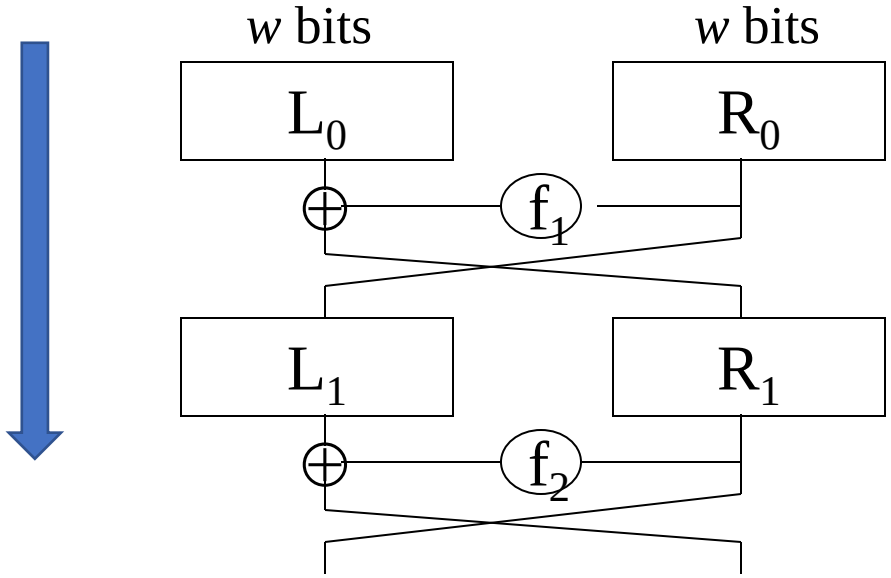
Jaringan Feistel

- Beberapa cipher blok seperti DES, GOST, Feal, Lucifer, RC5, RC6, dan lain-lain menggunakan struktur *enciphering* pada setiap putaran yang dinamakan **jaringan Feistel** (*Feistel Network*).

- Dalam hal ini, blok plainteks dibagi menjadi dua bagian, dan pada masing-masing bagian dilakukan proses transformasi menjadi upa-bagian pada putaran selanjutnya.

- Struktur ini bersifat *reversible*, yaitu untuk melakukan dekripsi maka jaringan Feistel dieksekusi dari “bawah” ke “atas”.

Feistel Network



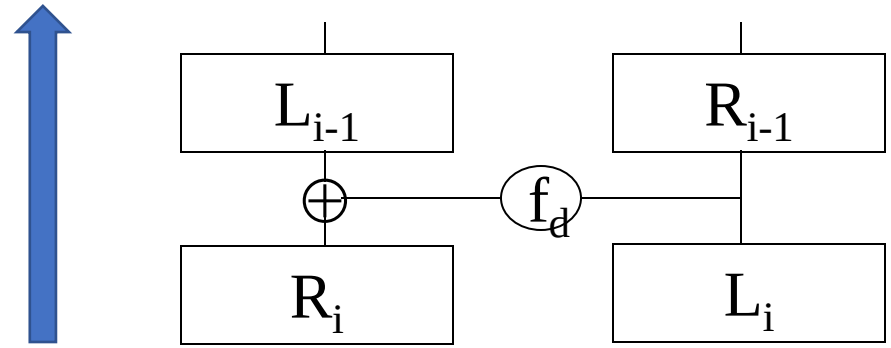
Encryption:

$$L_1 = R_0 \qquad R_1 = L_0 \oplus f_1(R_0)$$

$$L_2 = R_1 \qquad R_2 = L_1 \oplus f_2(R_1)$$

$$\ldots$$

$$L_i = R_{i-1} \qquad R_i = L_{i-1} \oplus f_i(R_{i-1})$$



Decryption:

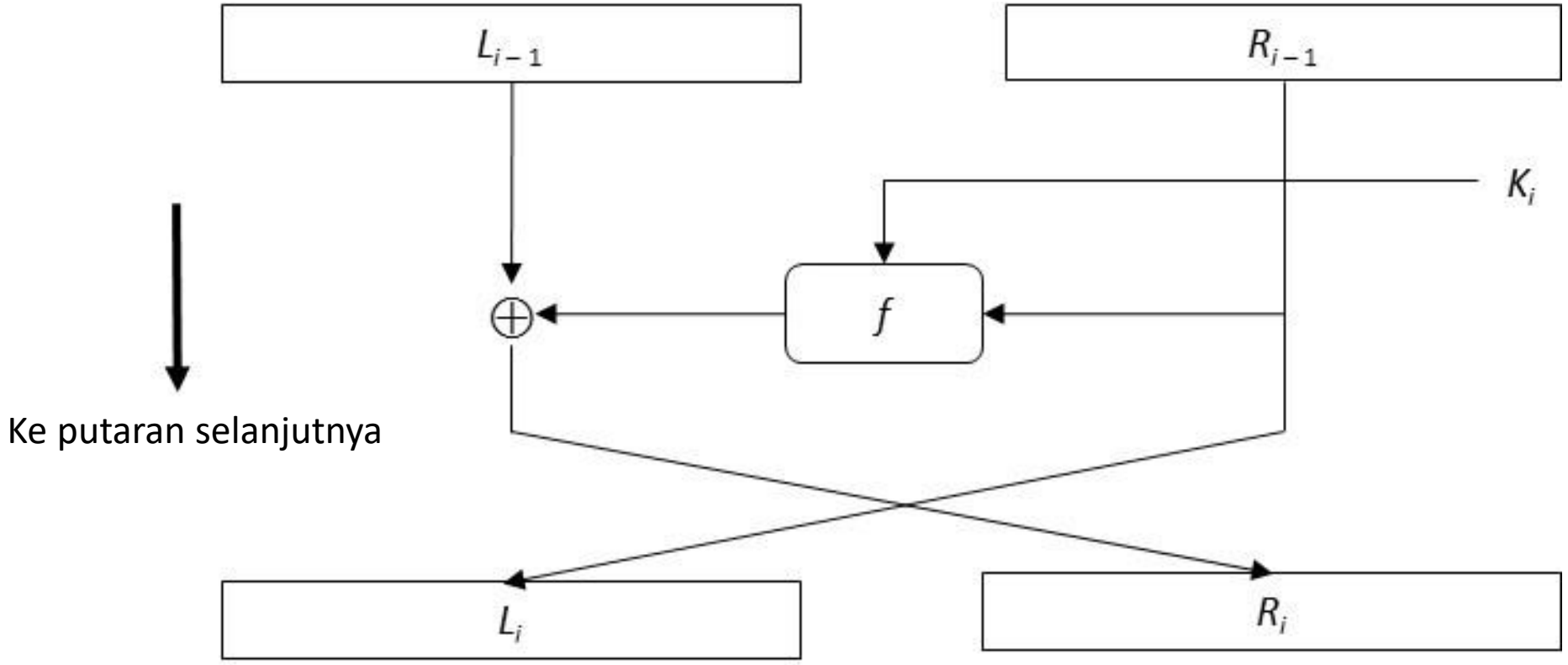
$$R_{i-1} = L_i \qquad L_{i-1} = R_i \oplus f_i(L_i)$$

$$\ldots$$

$$R_0 = L_1; \qquad L_0 = R_1 \oplus f_1(L_1)$$

Sumber: Cryptography CS 555

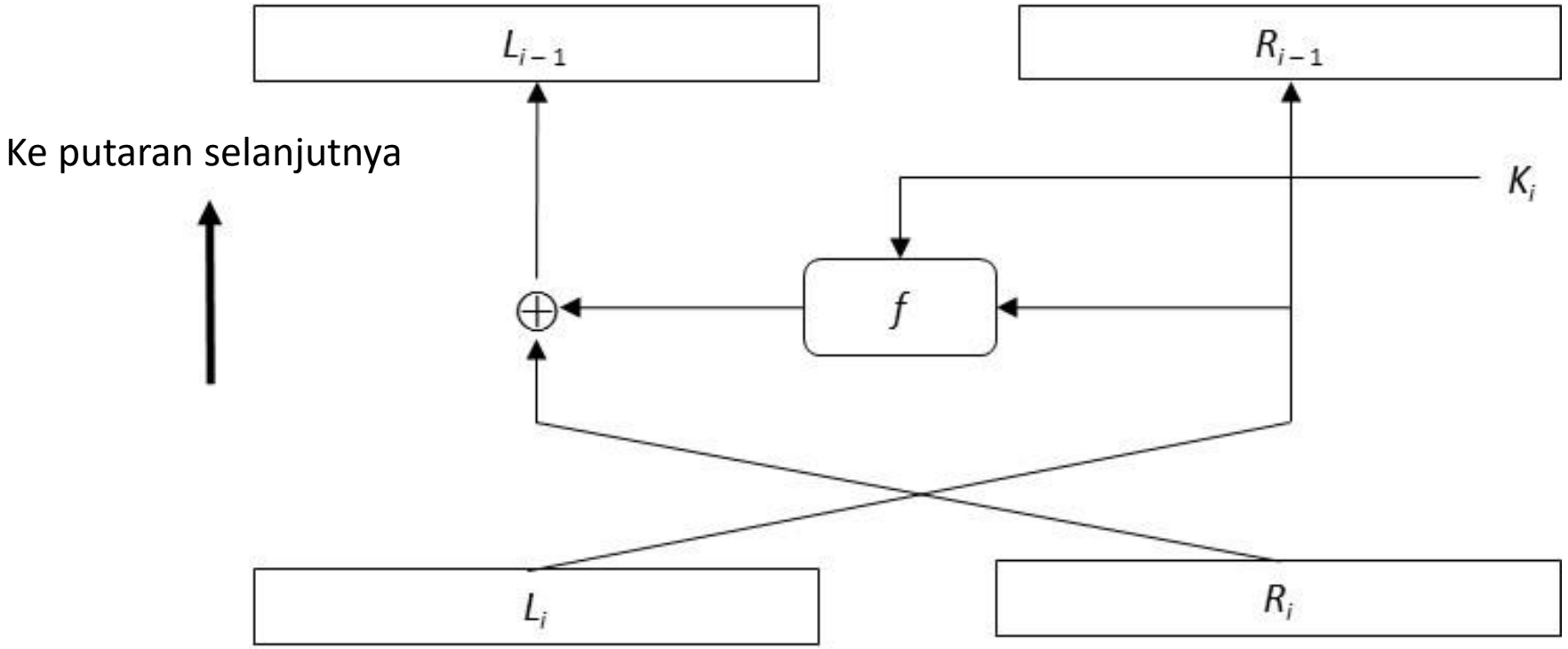
Contoh jaringan Feistel di dalam DES:



Jaringan *Feistel* pada enkripsi putaran ke-i

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

- Untuk melakukan dekripsi, maka urutan komputasi di dalam Feistel dibalik (dari "bawah" ke "atas"), itulah sebabnya dinamakan *reversible*.



Jaringan *Feistel* pada dekripsi putaran ke-i

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

- Sifat *reversible* ini membuat perancang cipher blok tidak perlu membuat algoritma baru untuk mendekripsi cipherteks menjadi plainteks.

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

- Sifat *reversible* tidak bergantung pada fungsi *f* sehingga fungsi *f* dapat dibuat serumit mungkin.

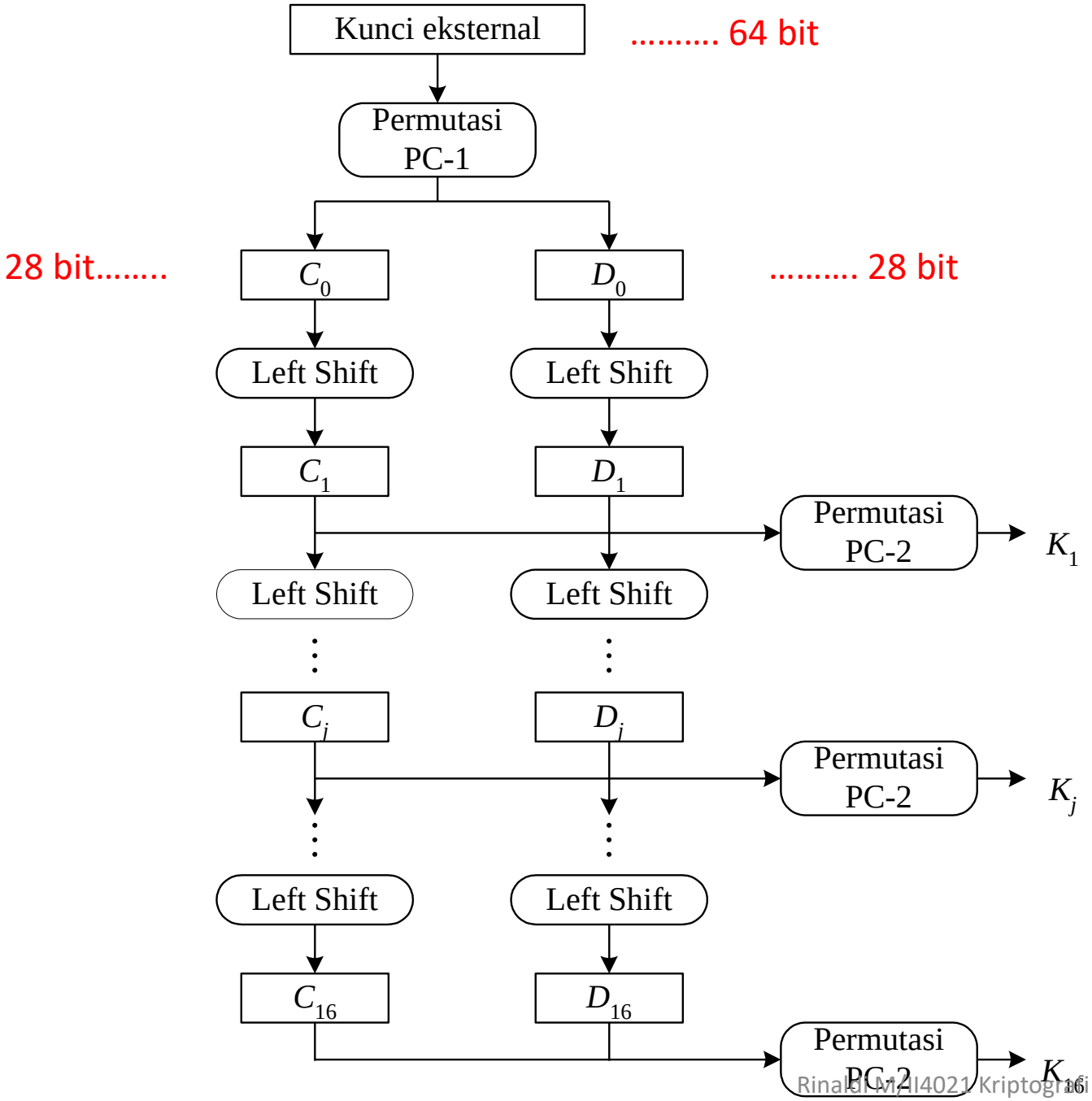
Pembangkitan kunci putaran

- Setiap putaran di dalam *iterated cipher* menggunakan kunci internal (*subkey*) yang dinamakan kunci putaran (*round key*).

- Kunci putaran dibangkitkan dari kunci eksternal yang diberikan oleh *user* melalui proses yang dinamakan *key expansion* atau *key scheduling*.

- Di dalam *key expansion* dilakukan komputasi yang kompleks untuk menghasilkan sejumlah kunci putaran yang berbeda-beda.

- Panjang kunci putaran tidak selalu sama dengan panjang kunci eksternal.



Gambar pembangkitan kunci putaran di dalam DES

Ukuran blok dan kunci

- Ukuran blok pesan (n) yang diproses selama enkripsi/dekripsi selalu tetap.

  Contoh: DES (n = 64), AES (n = 128, 192, 256)
- Makin besar ukuran blok mengarah ke efek *diffusion* yang lebih besar.
- Ukuran kunci (m) bisa sama dengan ukuran blok atau berbeda dengan ukuran blok

  Contoh: DES (m = 56), AES (m = 128, 192, 256)
- Makin besar ukuran kunci mengarah ke efek *confusion* yang lebih besar dan lebih tahan terhadap serangan brute force.