09 - Block Cipher

Oleh: Rinaldi Munir

Cipher Blok (*Block Cipher*)

- Berbeda dengan *cipher* alir, pada *cipher* blok plainteks dibagi menjadi blok-blok bit dengan panjang sama. Ukuran blok yang umum adalah 64 bit, 128 bit, 256 bit, dsb.

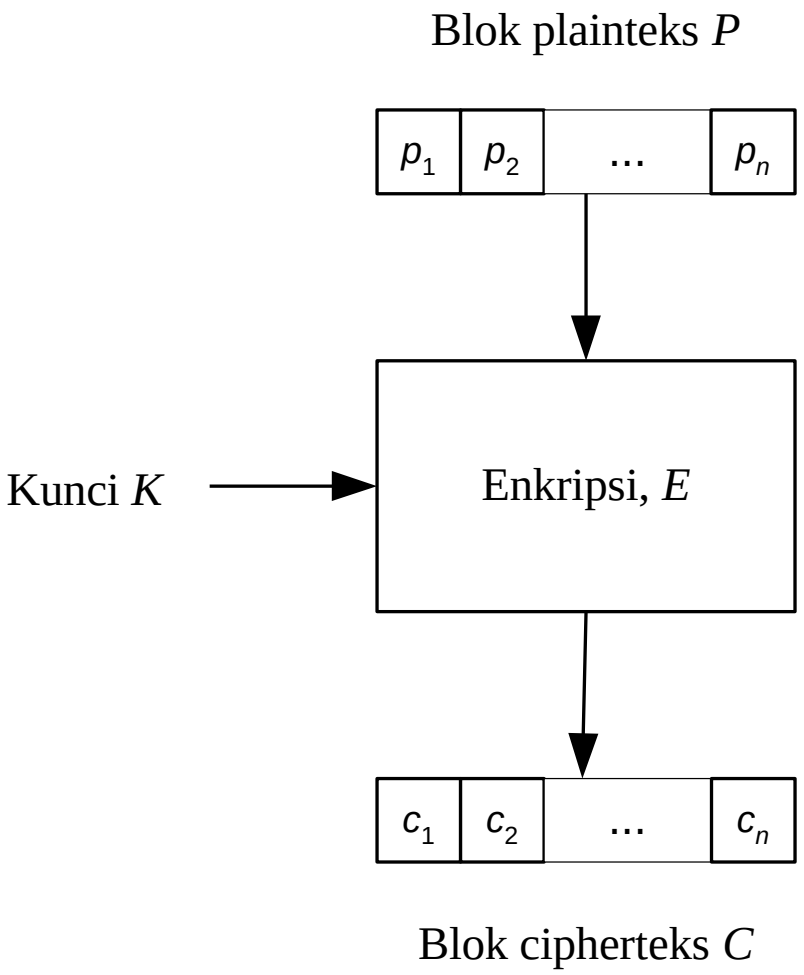
- Panjang blok cipherteks = panjang blok plainteks.

- Enkripsi dilakukan pada setiap blok plainteks dengan menggunakan bit-bit kunci

- Panjang kunci eksternal (yang diberikan oleh pengguna) tidak harus sama dengan panjang blok plainteks. Pada beberapa *cipher* blok, panjang kunci = panjang blok, tetapi pada *cipher* blok yang lain tidak sama.

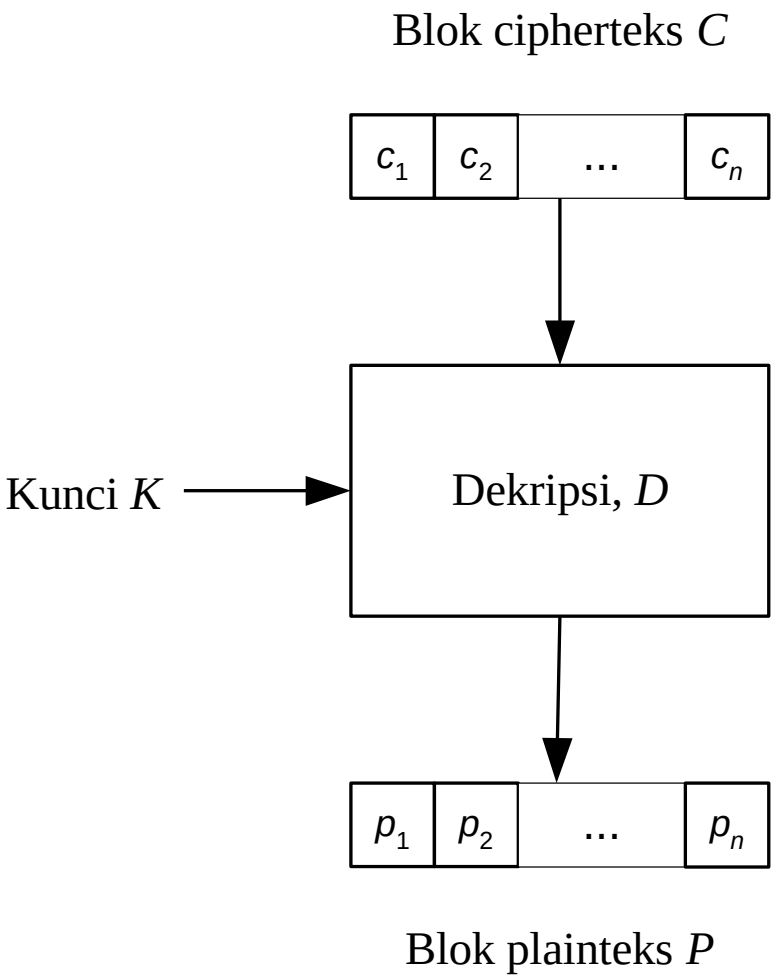
Blok plainteks berukuran n bit:

$P = (p_1, p_2, \ldots, p_n)$, $p_i \in \{0, 1\}$



Blok cipherteks berukuran n bit:

$C = (c_1, c_2, \ldots, c_n)$, $c_i \in \{0, 1\}$



- E dan D adalah fungsi enkripsi dan dekripsi dari suatu *cipher* blok:

  C = E(P) dan P = D(C)

- Contoh fungsi E yang sederhana misalkan algoritmanya adalah sbb:

  1. XOR-kan blok plainteks *P* dengan *K*
  2. lalu geser secara *wrapping* bit-bit dari hasil langkah 1 satu posisi ke kiri

$$C = E_K(P) = (P \oplus K) << 1$$

- D adalah kebalikan (*invers*) dari E. Contoh fungsi D yang sederhana adalah sbb:

  1. Geser secara *wrapping* bit-bit ciphertes C satu posisi ke kanan
  2. Lalu XOR-kan blok hasil Langkah 1 dengan *K*

$$P = D_K(C) = (C >> 1) \oplus K$$

Contoh 1: Misalkan plainteks adalah 110111010001000101000101, dibagi menjadi tiga buah blok masing-masing berukuran 8 bit:

11011101	00010001	01000101
P1	P2	P3

Kunci yang digunakan adalah K = 01010110

Enkripsi blok pertama, 11011101, dengan fungsi E sebagai berikut:

1. P1 ⊕ K = 11011101 ⊕ 01010110 = 10001011
2. 10001011 << 1 = 00010111

Jadi, C1 = 00010111. Cara yang sama dilakukan untuk P2 dan P3.

Dekripsi blok cipherteks pertama, 00010111, dengan fungsi D sebagai berikut:

1. 00010111 >> 1 = 10001011
2. 10001011 ⊕ K = 10001011 ⊕ 01010110 = 11011101 = P1

- Lakukan proses yang sama untuk blok P2, P3, dst.

- Tentu saja algoritma enkripsi E dan D di atas sangat lemah karena mudah dipecahkan.

- *Cipher* blok modern membuat E dan D sangat rumit prosesnya sehingga sangat sukar dipecahkan oleh kriptanalis.

- Fungsi E dan D melibatkan sejumlah teknik seperti teknik substitusi, permutasi, pergeseran (*shifting*), kompresi, ekspansi, dsb.

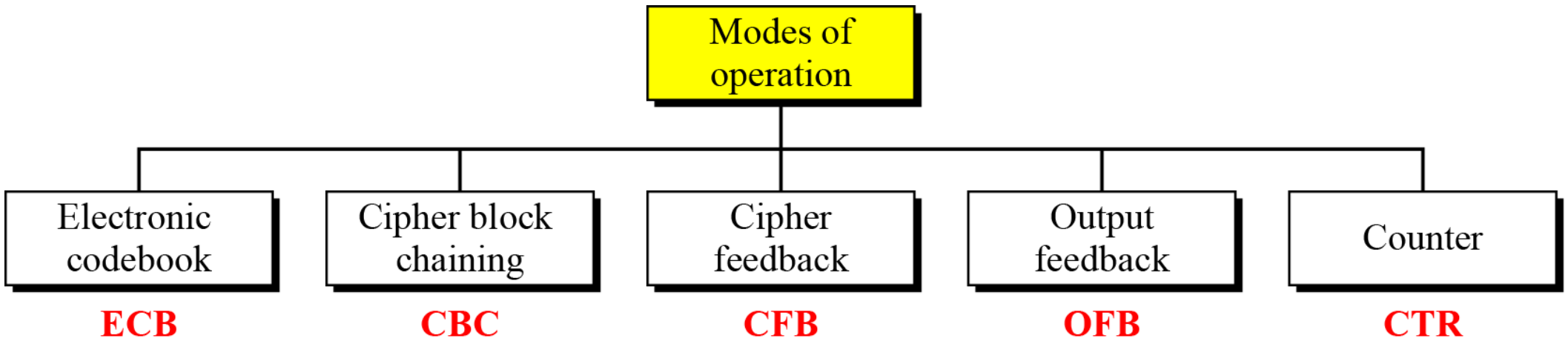
- Setiap blok tidak dienkripsi satu kali, tetapi berulang kali untuk mendapatkan cipherteks yang kuat.

Mode Operasi Cipher Blok

- Mode operasi: berkaitan dengan cara blok dioperasikan sebelum dienkripsi/dekripsi oleh fungsi E dan D.

- Ada 5 mode operasi *cipher* blok:
  1. *Electronic Code Book* (*ECB*)
  2. *Cipher Block Chaining* (*CBC*)
  3. *Cipher Feedback* (*CFB*)
  4. *Output Feedback (OFB)*
  5. *Counter Mode*

- Di dalam kuliah ini hanya dibahas mode ECB, CBC, dan Counter


Modes of operation
Electronic codebook
Cipher block chaining
Cipher feedback
Output feedback
Counter
ECB
CBC
CFB
OFB
CTR

1. Electronic Code Book (ECB)

- Setiap blok plainteks P_i dienkripsi secara individual dan independen dari blok lainnya menjadi blok cipherteks C_i .

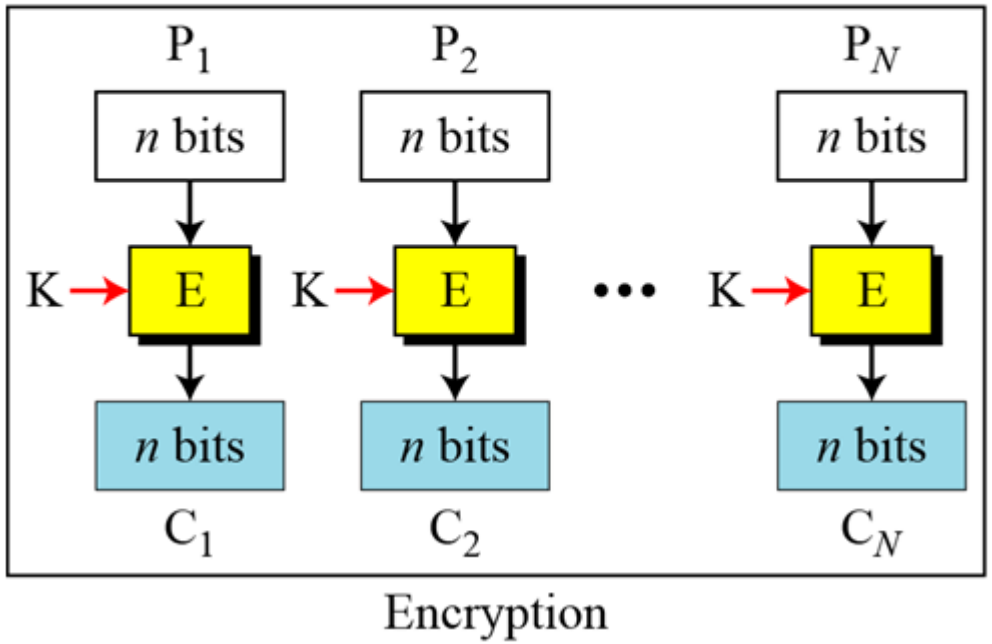
- Enkripsi: $C_i = E_K(P_i)$

  Dekripsi: $P_i = D_K(C_i)$

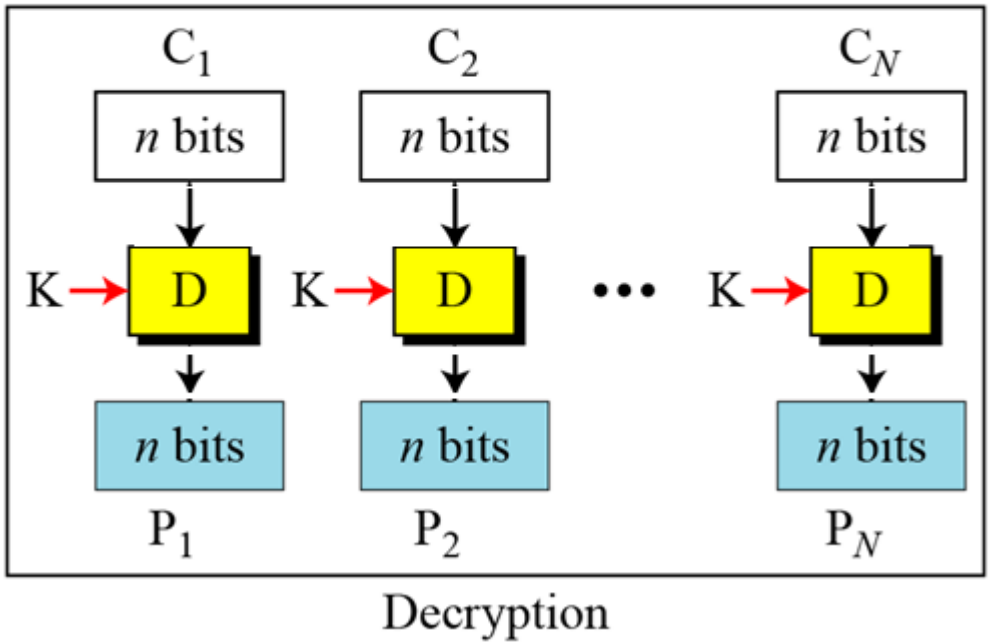
  yang dalam hal ini, P_i dan C_i masing-masing blok plainteks dan cipherteks ke-*i*.

Mode ECB

E: Encryption D: Decryption
P_i: Plaintext block i C_i: Ciphertext block i
K: Secret key



Encryption



Decryption

Cipher Block Chaining(CBC)

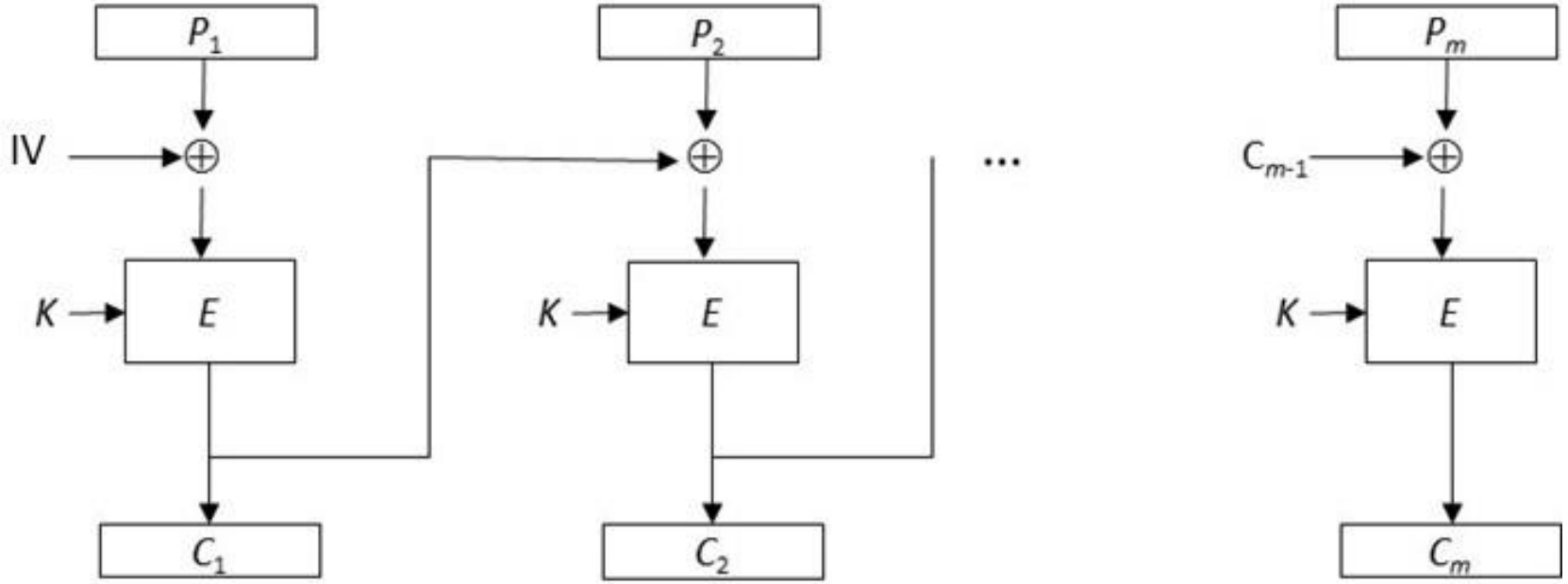
- Tujuan: membuat ketergantungan antar blok (mengatasi kelemahan mode ECB).
- Setiap blok cipherteks bergantung tidak hanya pada blok plainteksnya tetapi juga pada seluruh blok plainteks sebelumnya.
- Hasil enkripsi blok sebelumnya di-umpan-balikkan ke dalam enkripsi blok yang *current*.
- Enkripsi blok pertama memerlukan blok semu (C_0) yang disebut *IV* (*initialization vector*).
- *IV* dapat diberikan oleh pengguna atau dibangkitkan secara acak oleh program.
- Pada dekripsi, blok plainteks diperoleh dengan cara meng-*XOR*-kan *IV* dengan hasil dekripsi terhadap blok cipherteks pertama.

(a) Skema enkripsi mode CBC

Encryption:

$C_0 = IV$

$C_i = E_K (P_i \oplus C_{i-1})$

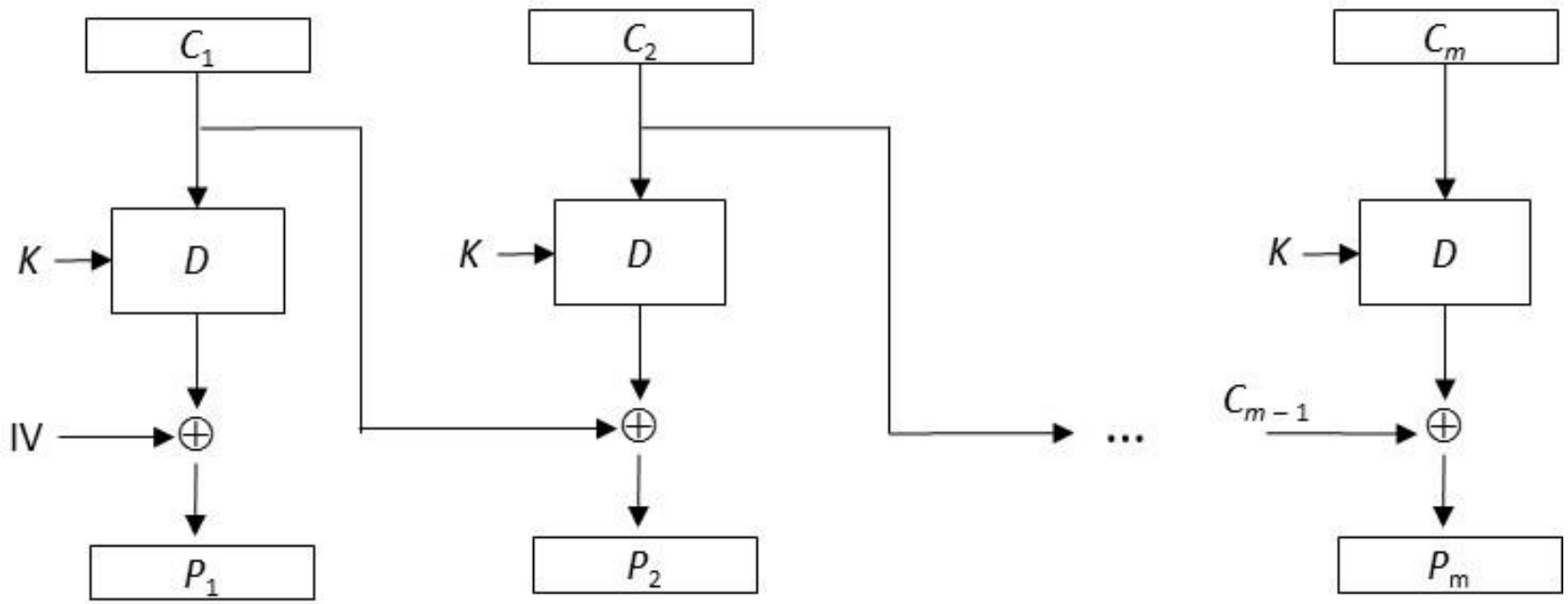


(b) Skema dekripsi mode CBC

Decryption:

$C_0 = IV$

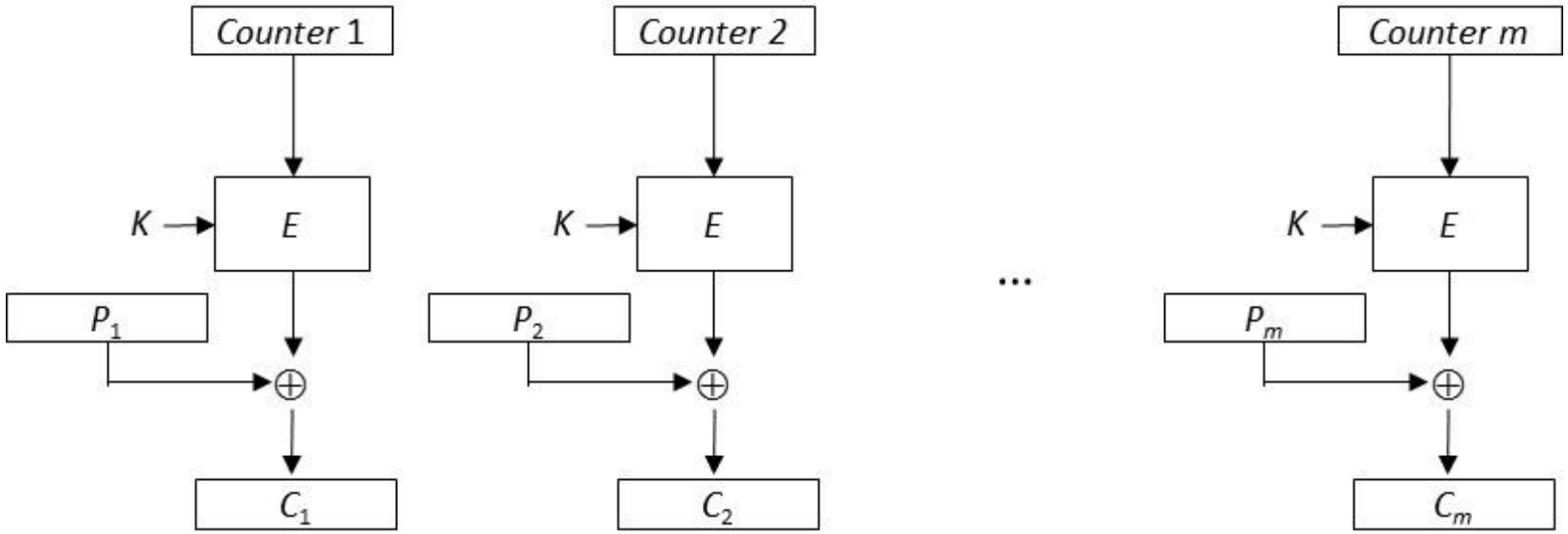
$P_i = D_K (C_i) \oplus C_{i-1}$



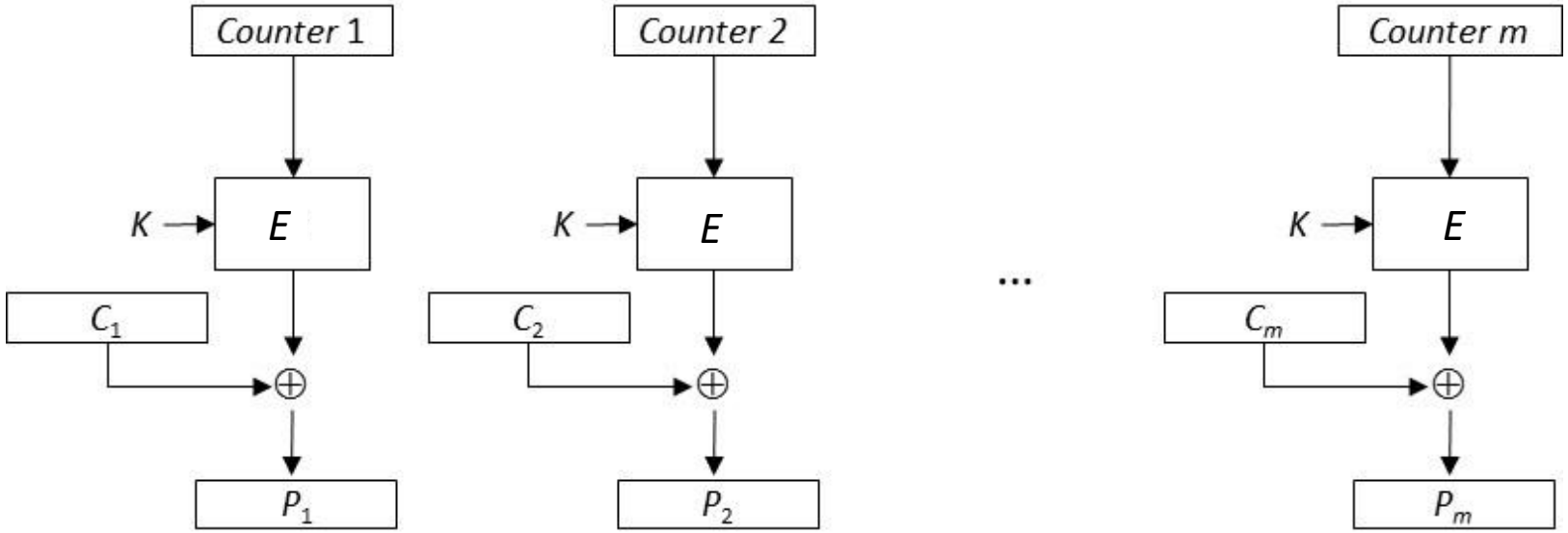
Counter Mode

- Mode *counter* tidak melakukan perantaian (*chaining*) seperti pada *CBC*.
- *Counter* adalah sebuah nilai berupa blok bit yang ukurannya sama dengan ukuran blok plainteks.
- Nilai *counter* harus berbeda dari setiap blok yang dienkripsi. Pada mulanya, untuk enkripsi blok pertama, *counter* diinisialisasi dengan sebuah nilai.
- Selanjutnya, untuk enkripsi blok-blok berikutnya *counter* dinaikkan (*increment*) nilainya satu (counter $\leftarrow$ counter + 1).

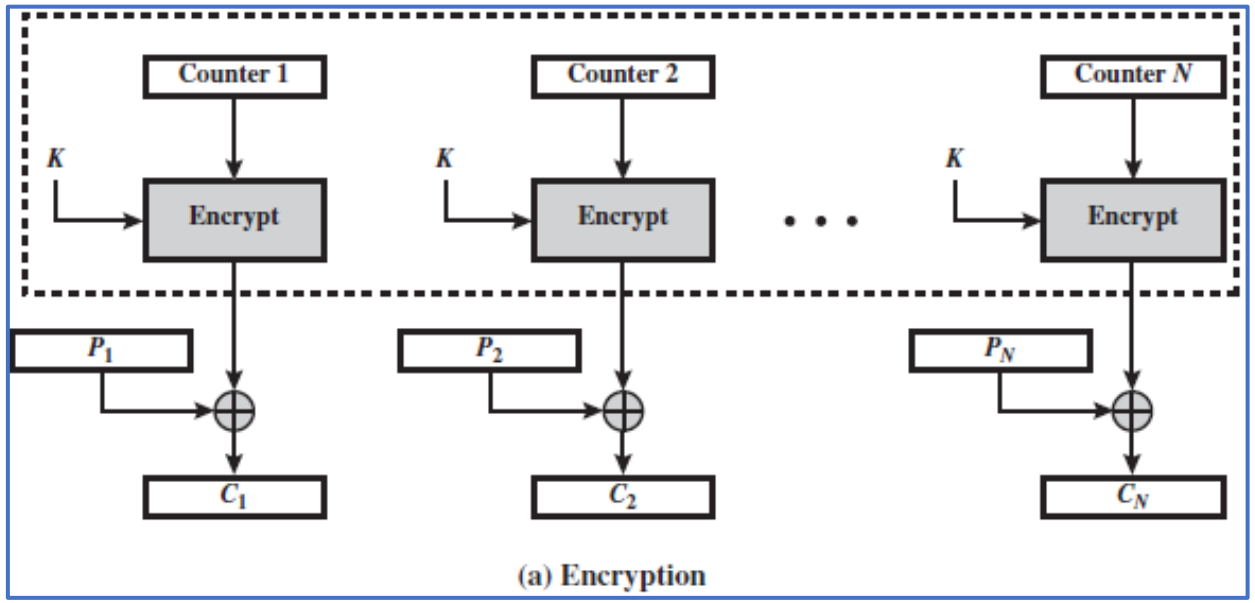
(a) Enkripsi

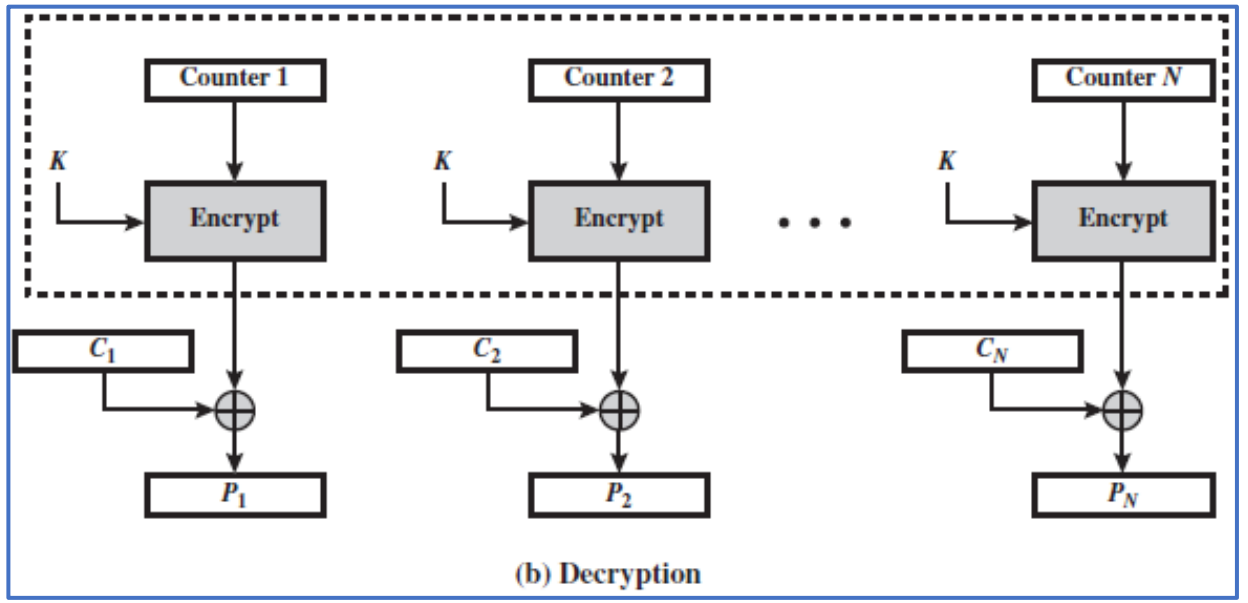


(b) Dekripsi



Mode Counter





CTR	$C_j = P_j \oplus \mathrm{E}(K, T_j) \quad j = 1, \ldots, N-1$ $C_N^* = P_N^* \oplus \mathrm{MSB}_u[\mathrm{E}(K, T_N)]$	$P_j = C_j \oplus \mathrm{E}(K, T_j) \quad j = 1, \ldots, N-1$ $P_N^* = C_N^* \oplus \mathrm{MSB}_u[\mathrm{E}(K, T_N)]$

T_j adalah counter, nilainya bertambah satu setiap kali enkripsi blok

- *Daftar block cipher:*

1. Lucifer
2. DES
3. GOST
4. 3DES (Triple-DES)
5. RC2
6. RC5
7. Blowfish
8. AES
9. IDEA
10. LOKI
11. FEAL
12. CAST-128
13. CRAB
14. SAFER
15. Twofish
16. Serpent
17. RC6
18. Camellia
19. 3-WAY
20. MMB
21. Skipjack
22. TEA
23. XTEA
24. SEED
25. Coconut
26. Cobra
27. MARS
28. BATON
29. CRYPTON
30. LEA, dll

v·t·e	Block ciphers (security summary)
Common algorithms	AES · Blowfish · DES (internal mechanics, Triple DES) · Serpent · Twofish
Less common algorithms	ARIA · Camellia · CAST-128 · GOST · IDEA · LEA · RC2 · RC5 · RC6 · SEED · Skipjack · TEA · XTEA
Other algorithms	3-Way · Akelarre · Anubis · BaseKing · BassOmatic · BATON · BEAR and LION · CAST-256 · Chiasmus · CIKS-1 · CIPHERUNICORN-A · CIPHERUNICORN-E · CLEFIA · CMEA · Cobra · COCONUT98 · Crab · Cryptomeria/C2 · CRYPTON · CS-Cipher · DEAL · DES-X · DFC · E2 · FEAL · FEA-M · FROG · G-DES · Grand Cru · Hasty Pudding cipher · Hierocrypt · ICE · IDEA NXT · Intel Cascade Cipher · Iraqi · Kalyna · KASUMI · KeeLoq · KHAZAD · Khufu and Khafre · KN-Cipher · Kuznyechik · Ladder-DES · LOKI (97, 89/91) · Lucifer · M6 · M8 · MacGuffin · Madryga · MAGENTA · MARS · Mercy · MESH · MISTY1 · MMB · MULTI2 · MultiSwap · New Data Seal · NewDES · Nimbus · NOEKEON · NUSH · PRESENT · Prince · Q · REDOC · Red Pike · S-1 · SAFER · SAVILLE · SC2000 · SHACAL · SHARK · Simon · SM4 · Speck · Spectr-H64 · Square · SXAL/MBAL · Threefish · Treyfer · UES · xmx · XXTEA · Zodiac
Design	Feistel network · Key schedule · Lai–Massey scheme · Product cipher · S-box · P-box · SPN · Confusion and diffusion · Avalanche effect · Block size · Key size · Key whitening (Whitening transformation)
Attack (cryptanalysis)	Brute-force (EFF DES cracker) · MITM (Biclique attack · 3-subset MITM attack) · Linear (Piling-up lemma) · Differential (Impossible · Truncated · Higher-order) · Differential-linear · Distinguishing (Known-key) · Integral/Square · Boomerang · Mod *n* · Related-key · Slide · Rotational · Side-channel (Timing · Power-monitoring · Electromagnetic · Acoustic · Differential-fault) · XSL · Interpolation · Partitioning · Rubber-hose · Black-bag · Davies · Rebound · Weak key · Tau · Chi-square · Time/memory/data tradeoff
Standardization	AES process · CRYPTREC · NESSIE
Utilization	Initialization vector · Mode of operation · Padding
v·t·e	Cryptography [show]

Sumber: https://en.wikipedia.org/wiki/Block_cipher

Data Encryption Standard (DES)

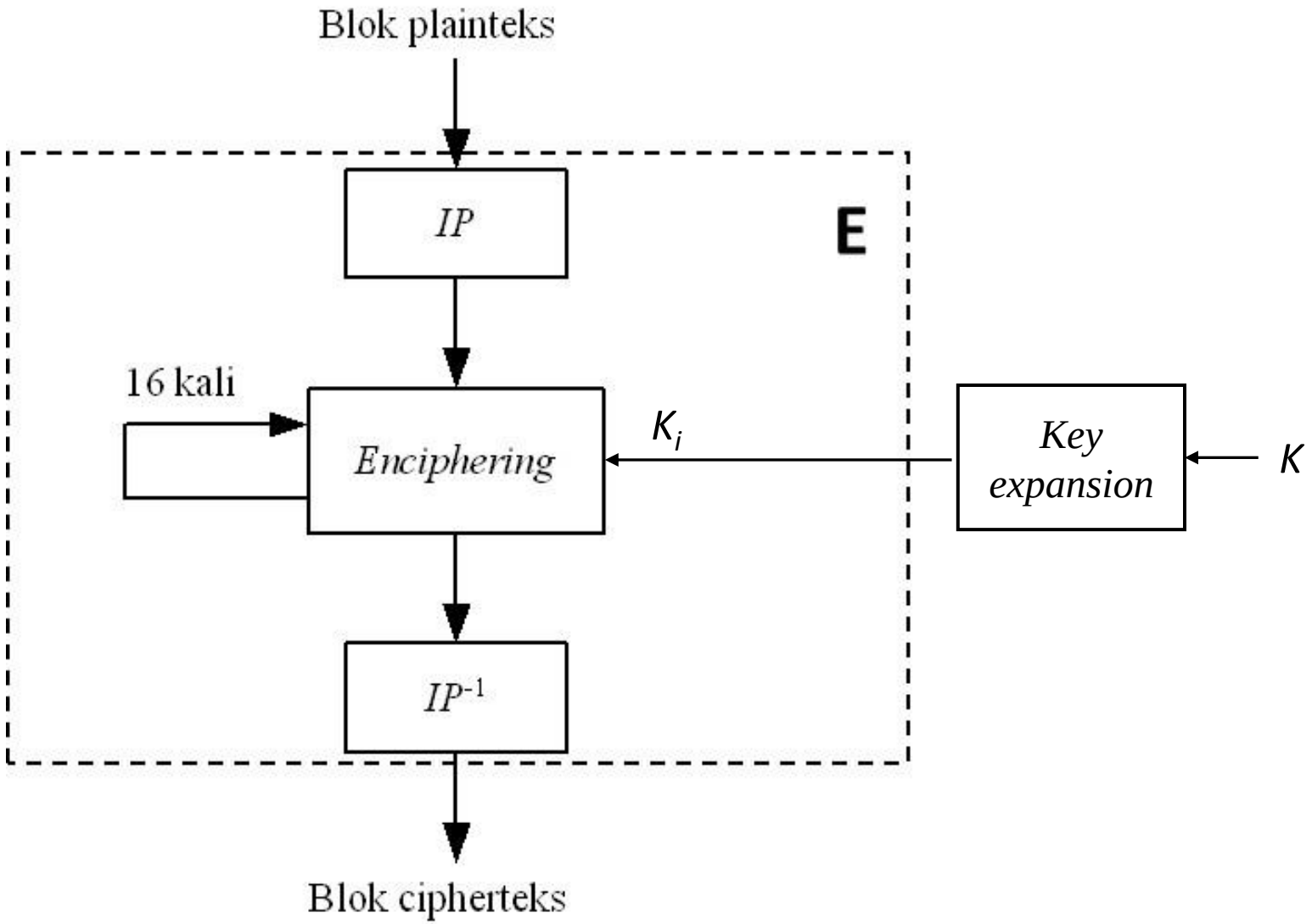
Sejarah DES

- 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

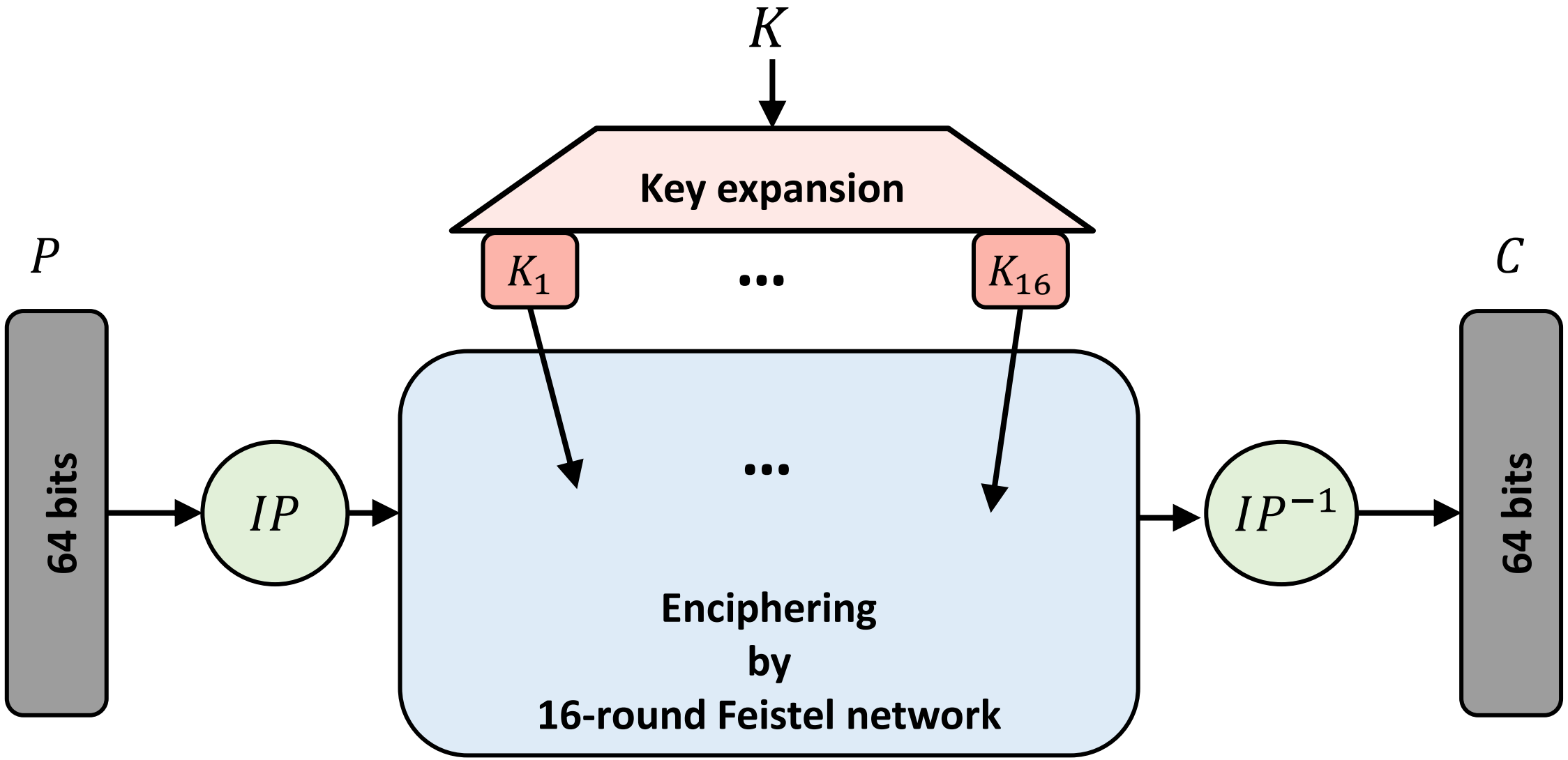
  - 16 putaran.
- Tahun 1976 DES disetujui oleh *National Bureau of Standard* (*NBS*) setelah penilaian kekuatannya oleh *NSA*.
- Sejak itu DES diimplementasikan secara luas, baik *software* maupun *hardware*.
- Tahun 1997-1998 DES berhasil dipecahkan dengan *brute force attack*
- Tahun 2001 DES diganti oleh AES

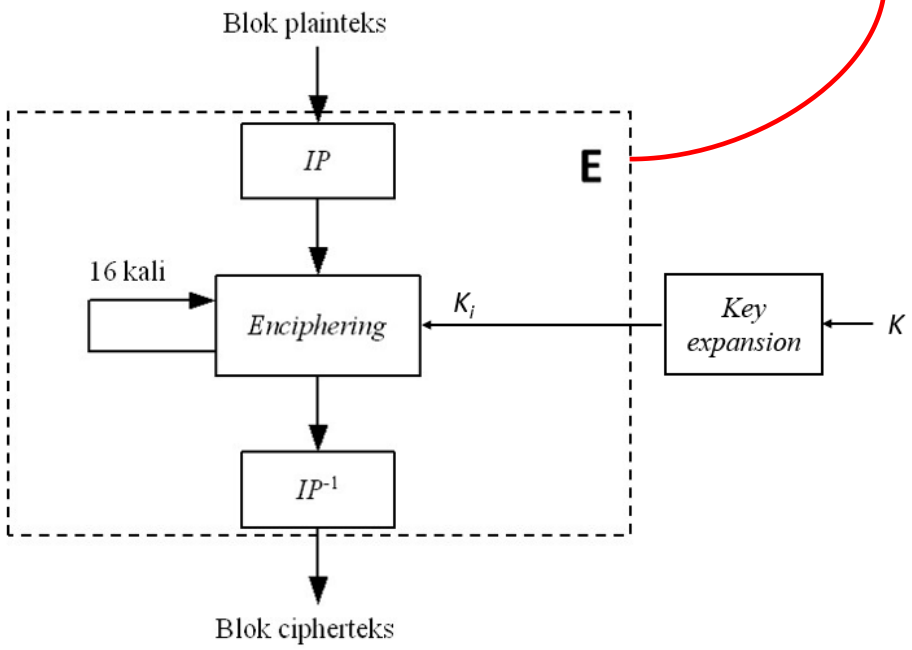
- Setiap blok plainteks dienkripsi dalam 16 putaran *enciphering*.
- Setiap putaran menggunakan kunci internal (kunci putaran) berbeda.
- Kunci internal sepanjang 48-bit dibangkitkan dari kunci eksternal
- Setiap blok plainteks mengalami permutasi awal (*IP*), 16 putaran *enciphering*, dan inversi permutasi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

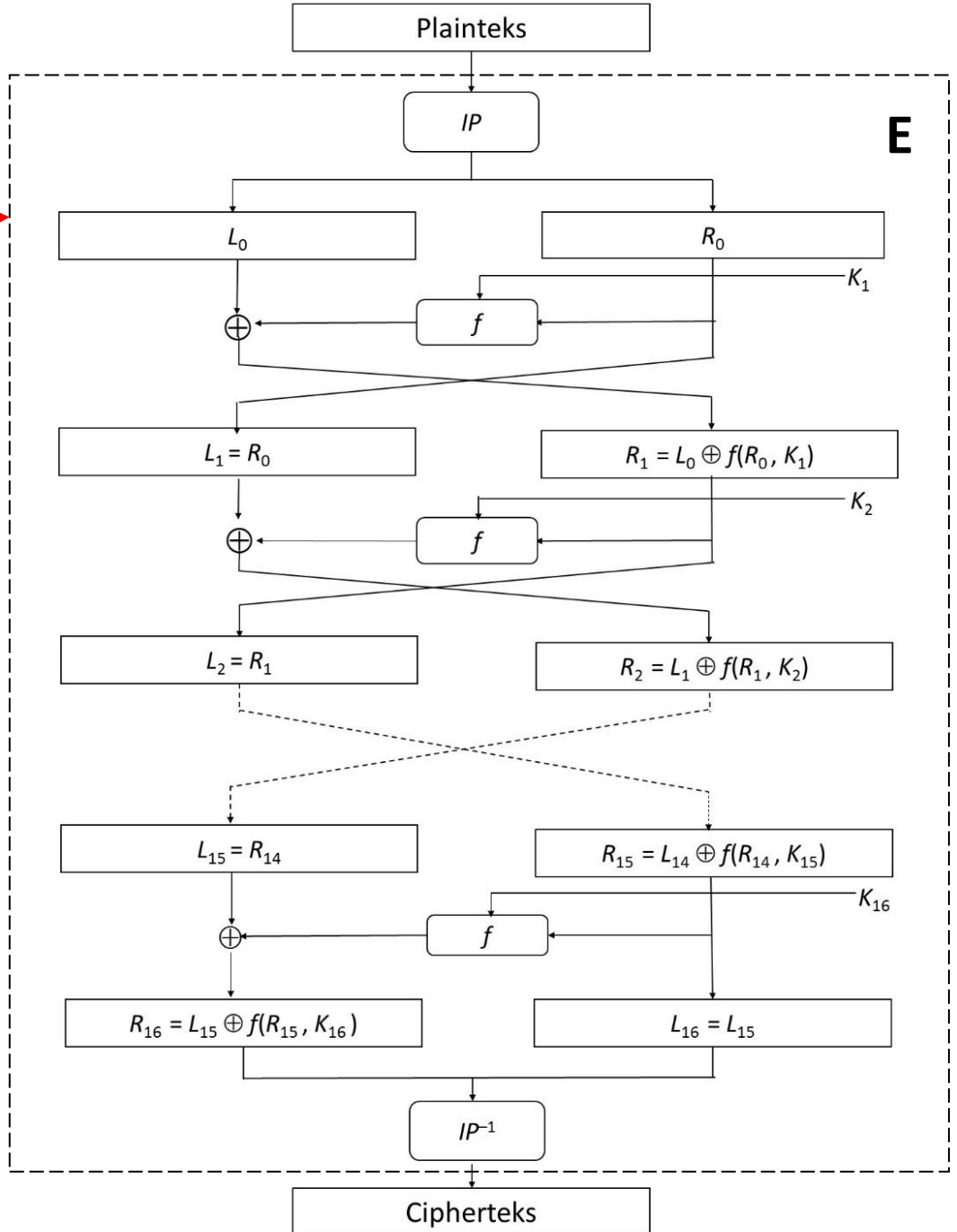


IP = Initial Permutation
IP^{-1} = Balikan (invers) IP

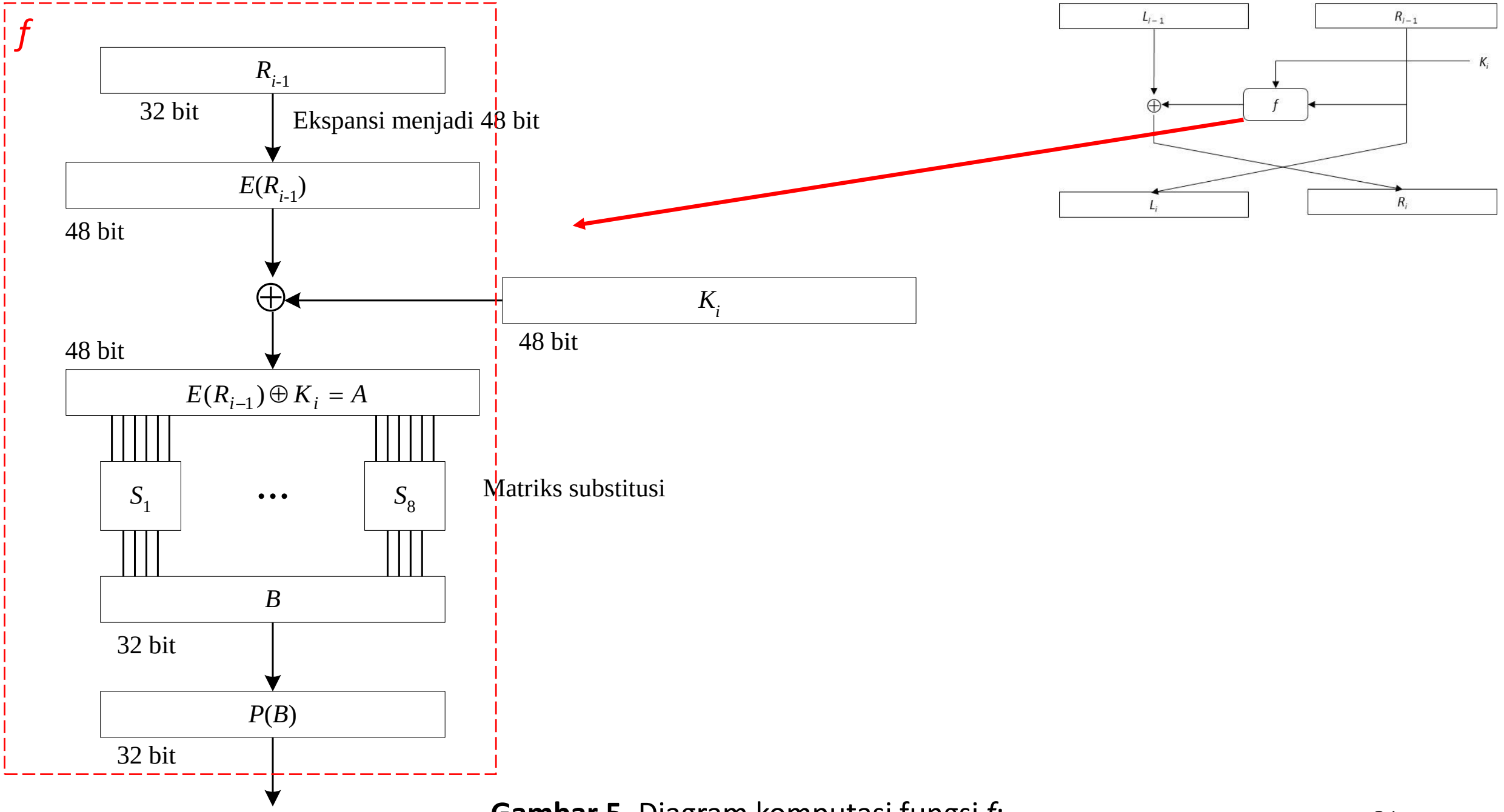
Gambar 1 Skema global algoritma DES


K
Key expansion
K_1
...
K_{16}
P
C
64 bits
IP
...
IP^{-1}
64 bits
Enciphering
by
16-round Feistel network





Gambar 2. Algoritma Enkripsi dengan DES



Gambar 5. Diagram komputasi fungsi *f*:

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

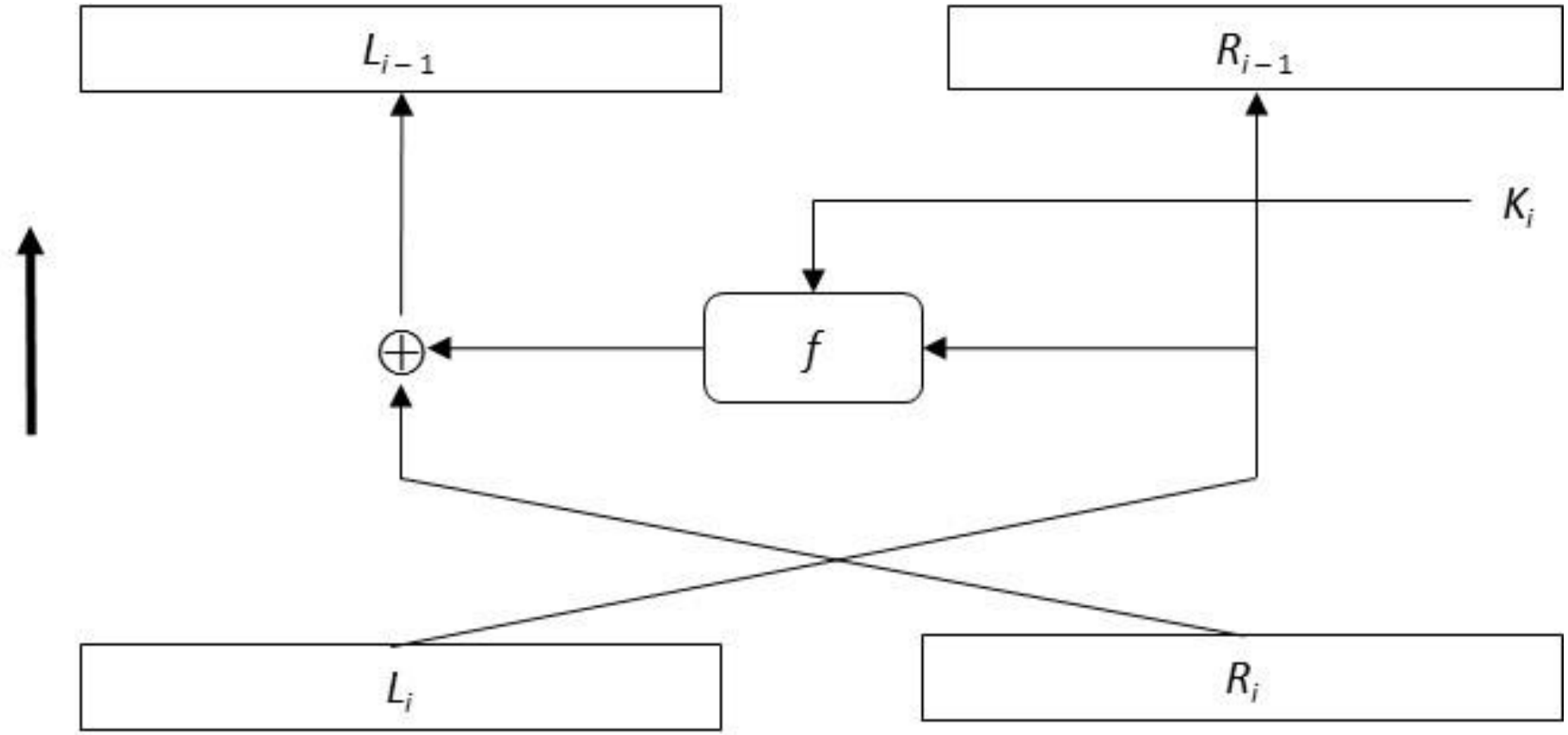
Dekripsi

- 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}, …, 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:



$R_{i-1} = L_i$

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

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.

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).

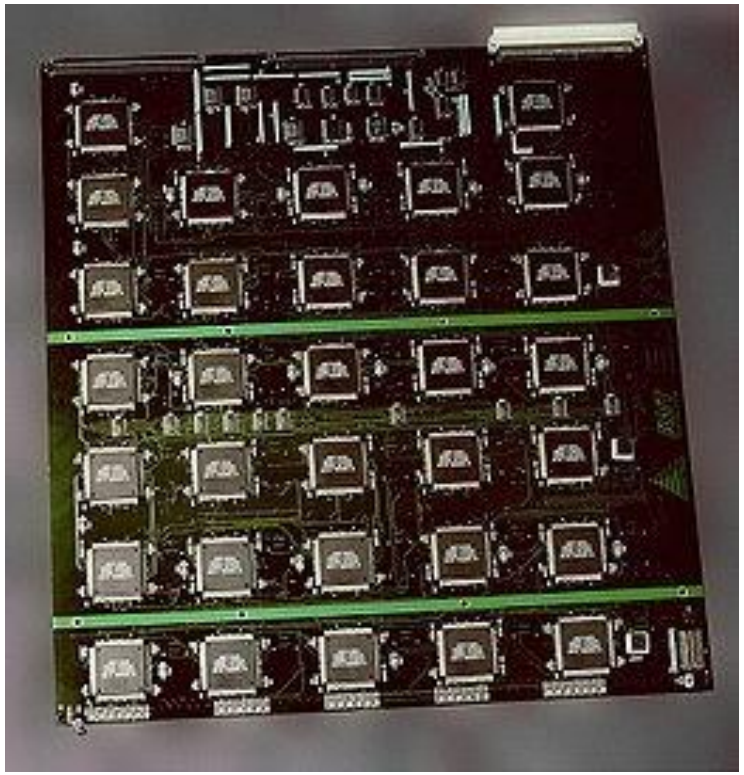
- Implementasi DES ke dalam perangkat lunak dapat melakukan enkripsi 32.000 blok per detik (dijalankan pada komputer *mainframe* IBM 3090, yaitu komputer tercepat saat itu / tahun 1976).

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.

- 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).

Program enkripsi – dekripsi DES dengan Python

```
#import library crypto dan base64
from Crypto.Cipher import DES
from Crypto import Random
import base64
```

```
def enkripsi (pesan):
    print("Plainteks: \n", pesan)
    print("Ketikkan kunci (8 karakter):")
    kunci = input()

    cipher = DES.new (kunci,DES.MODE_ECB)   #defenisikan mode DES ECB
    cipherteks = base64.b64encode (cipher.encrypt(pesan))  #enkripsi pesan

    print("\n")
    print("Cipherteks: \n", cipherteks)     #tampilkan ciphertext
```

```
def dekripsi(pesan):

    print("Cipherteks: \n", pesan)

    print("Ketikkan kunci (8 karakter):")
    kunci = input()

    cipher = DES.new(kunci, DES.MODE_ECB)

    #dekripsi pesan
    plainteks = cipher.decrypt (base64.b64decode(pesan))

    print("\n")

    #tampilkan plainteks
    print ("Pesan setelah didekripsi adalah: \n", plainteks)
```

```
def ProgramEnkripsiDekripsiDES():
    print("-- Enkripsi dan dekripsi pesan dengan DES --")
    print("1: Enkripsi pesan")
    print("2: Dekripsi pesan")

    pilih = input()
    if pilih == '1':
       print("Ketikkan pesan yang akan dienkripsi:")
       pesan = input()
       n = len(pesan)
       if n % 8 != 0:
          pesan = pesan + ' ' * (8 - n % 8)   #padding dengan spasi
       print("Enkripsi pesan...")
       enkripsi(pesan)
    elif pilih == '2':
       print("Ketikkan pesan yang akan didekripsi:")
       pesan = input()
       print("Dekripsi pesan...")
       dekripsi(pesan)
    else:
       print("Pilihan salah")
```

Run program:

Enkripsi

```
ProgramEnkripsiDekripsiDES()
```

```
-- Enkripsi dan dekripsi pesan dengan DES --
1: Enkripsi pesan
2: Dekripsi pesan
1
Ketikkan pesan yang akan dienkripsi:
Hari tanggal 17 Februari 2023, semoga kita sehat selalu gaess...
Enkripsi pesan...
Plainteks:
 Hari tanggal 17 Februari 2023, semoga kita sehat selalu gaess...
Ketikkan kunci (8 karakter):
abcdefgh


Cipherteks:
 b'qjUUi+maltZluPer4W6RArZrbrGJMETx0d/Boq7YNzWv9f65xPT2y06RZyq2Q0sjyF16GLDVEwe2px4YtOx/gA=='
```

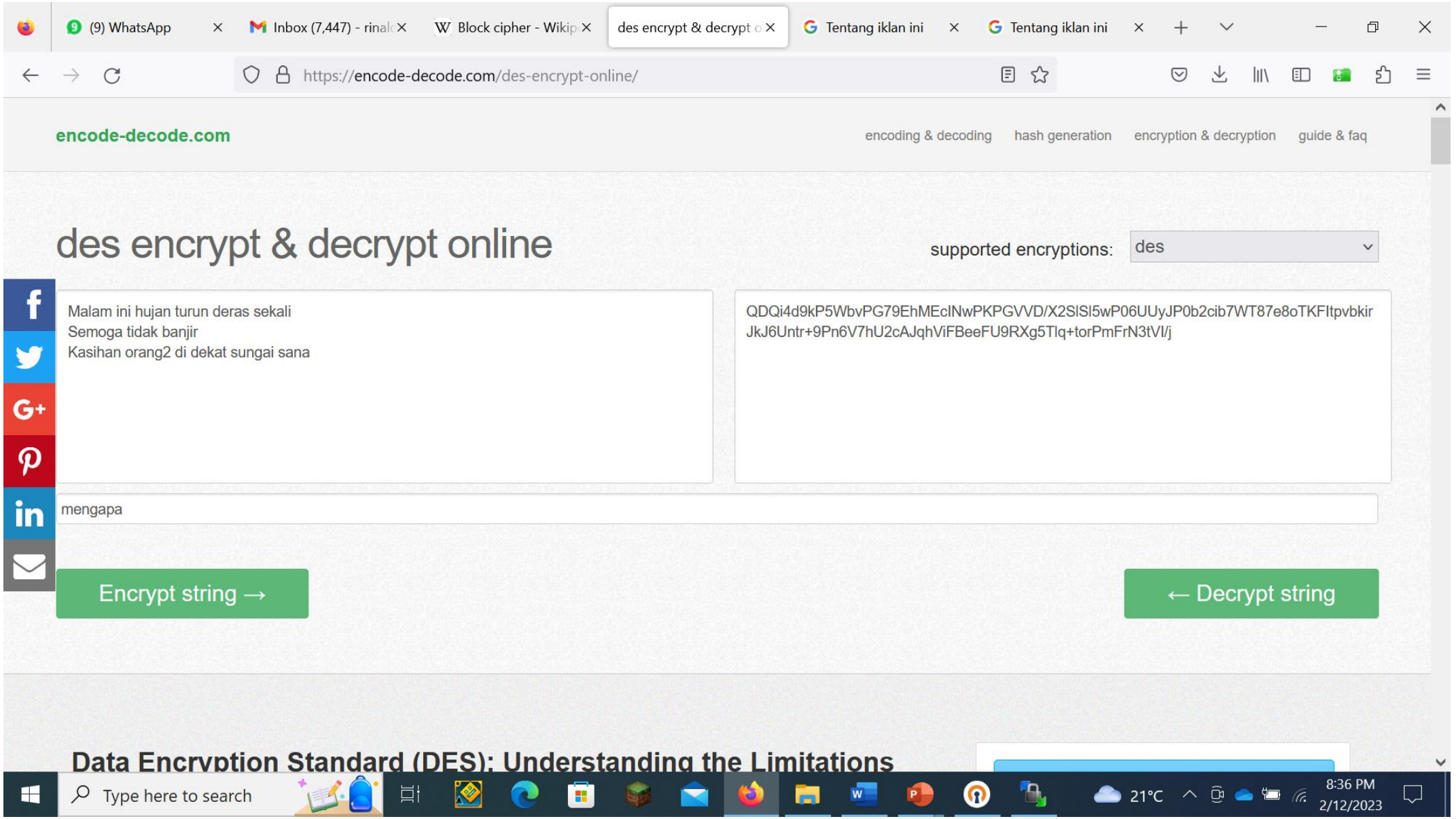
Dekripsi

```
ProgramEnkripsiDekripsiDES()
```

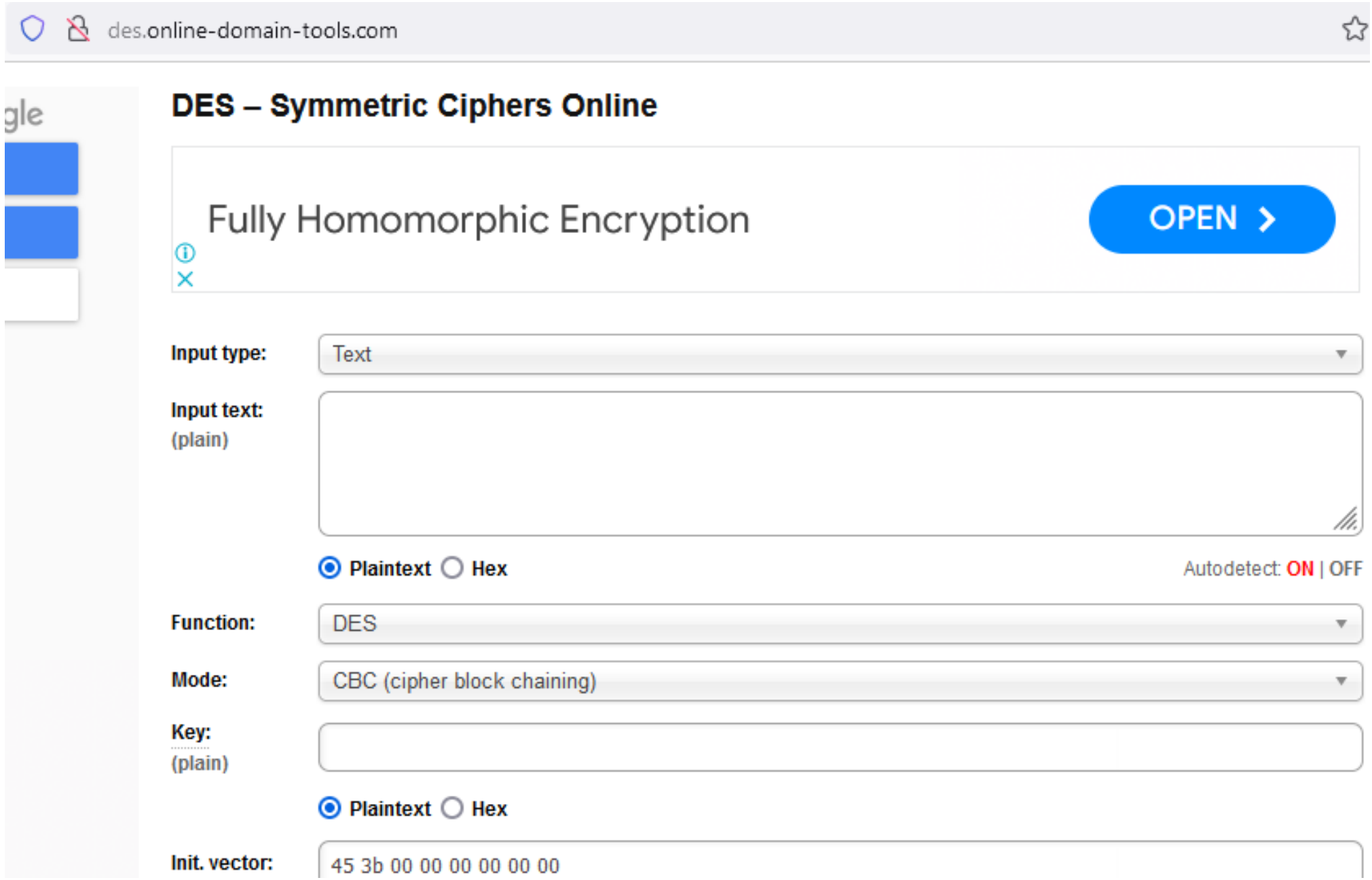
```
-- Enkripsi dan dekripsi pesan dengan DES --
1: Enkripsi pesan
2: Dekripsi pesan
2
Ketikkan pesan yang akan didekripsi:
qjUUi+maltZluPer4W6RArZrbrGJMETx0d/Boq7YNzWv9f65xPT2y06RZyq2Q0sjyF16GLDVEwe2px4YtOx/gA==
Dekripsi pesan...
Cipherteks:
 qjUUi+maltZluPer4W6RArZrbrGJMETx0d/Boq7YNzWv9f65xPT2y06RZyq2Q0sjyF16GLDVEwe2px4YtOx/gA==
Ketikkan kunci (8 karakter):
abcdefgh


Pesan setelah didekripsi adalah:
 b'Hari tanggal 17 Februari 2023, semoga kita sehat selalu gaess...'
```

Demo online DES

https://encode-decode.com/des-encrypt-online/



http://des.online-domain-tools.com/



Latihan 1

1. Pilihlah seorang temanmu sebagai penerima pesan
2. Sepakati sebuah kunci untuk enkripsi dan dekripsi dengan DES dengan temanmu, dengan cara bertemu secara langsung.
3. Ketiklah sebuah pesan pada laman https://encode-decode.com/des-encrypt-online/
4. Enkripsi pesan tersebut dengan DES menggunakan kunci yang telah disepakati pada poin 2. Mode yang digunakan adalah CBC
5. *Copy* cipherteks yang dihasilkan dari laman web poin 3, *paste* ke aplikasi *Line* atau *whatsapp*, kirim ke temanmu.
6. Temanmu sebagai penerima pesan meng-*copy* cipherteks yang diterimanya, lalu pindahkan ke laman web poin3 untuk didekripsi.
7. Ulangi pengiriman pesan dengan cara di atas dari temanmu ke kamu

Latihan 2

1. Pilihlah seorang temanmu sebagai penerima pesan
2. Sepakati sebuah kunci untuk enkripsi dan dekripsi dengan DES dengan temanmu, dengan cara bertemu secara langsung.
3. Pilihlah sebuah file (gambar, music, video, dsb) dengan *browsing* dari laman http://des.online-domain-tools.com/
4. Enkripsi file pesan tersebut dengan DES menggunakan kunci yang telah disepakati pada poin 2. Mode yang digunakan adalah CBC.
5. Kirim file cipherteks yang dihasilkan dari laman web poin 3 via email dan whatsapp/Line sebagai file.
6. Temanmu sebagai penerima pesan mendekripsi file cipherteks yang diterimanya dengan aplikasi web poin 3.
7. Ulangi pengiriman pesan dengan cara di atas dari temanmu ke kamu