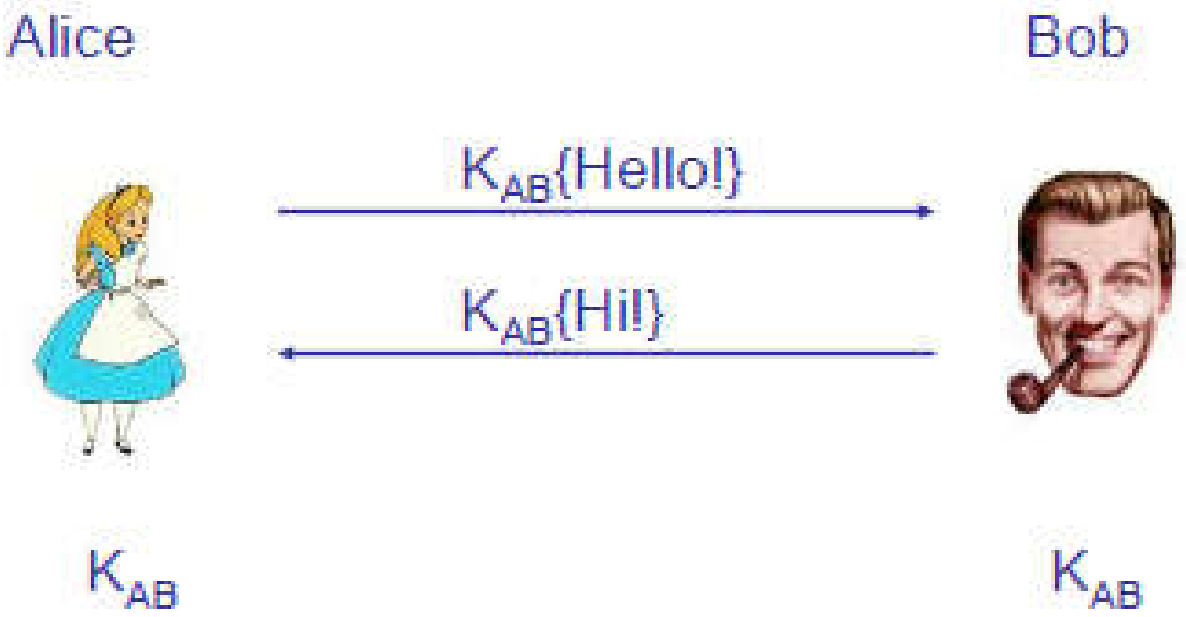
Keamanan Sistem Komputer

DES, AES, RSA

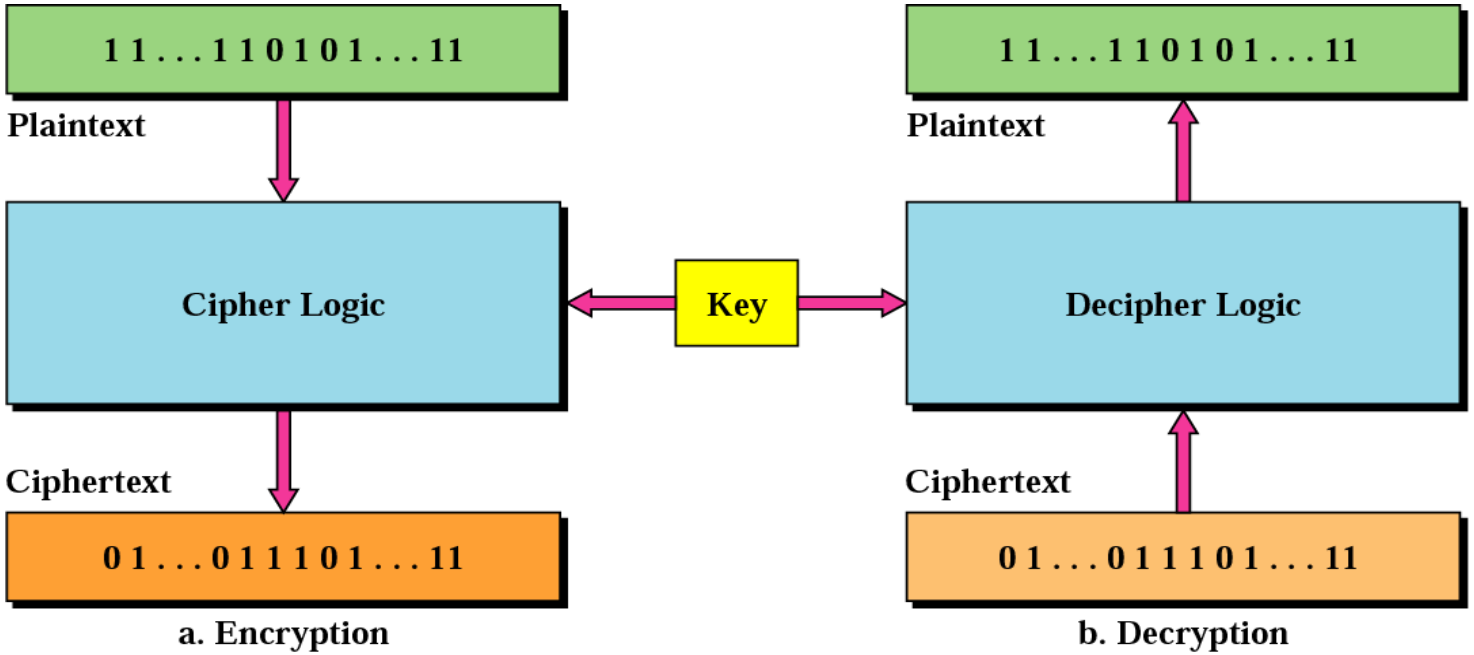
Kunci

- Kunci Simetrik
- Kunci Asimetrik
  - Kunci Publik
  - Kunci Privat

Kanal Aman : Kunci Bersama



Blok Cipher


1 1 . . . 1 1 0 1 0 1 . . . 11
Plaintext
Cipher Logic
Ciphertext
0 1 . . . 0 1 1 1 0 1 . . . 11
a. Encryption
Key
1 1 . . . 1 1 0 1 0 1 . . . 11
Plaintext
Decipher Logic
Ciphertext
0 1 . . . 0 1 1 1 0 1 . . . 11
b. Decryption

Kriptografi Kunci Simetrik

- Pengirim dan penerima menggunakan kunci yang sama
- Kunci harus sangat dirahasiakan (private)
- Disebut juga sebagai kriptografi simetrik atau kriptografi kunci rahasia

- Contoh : DES, Triple-DES, Blowfish, Twofish, AES, Rijndael dsb

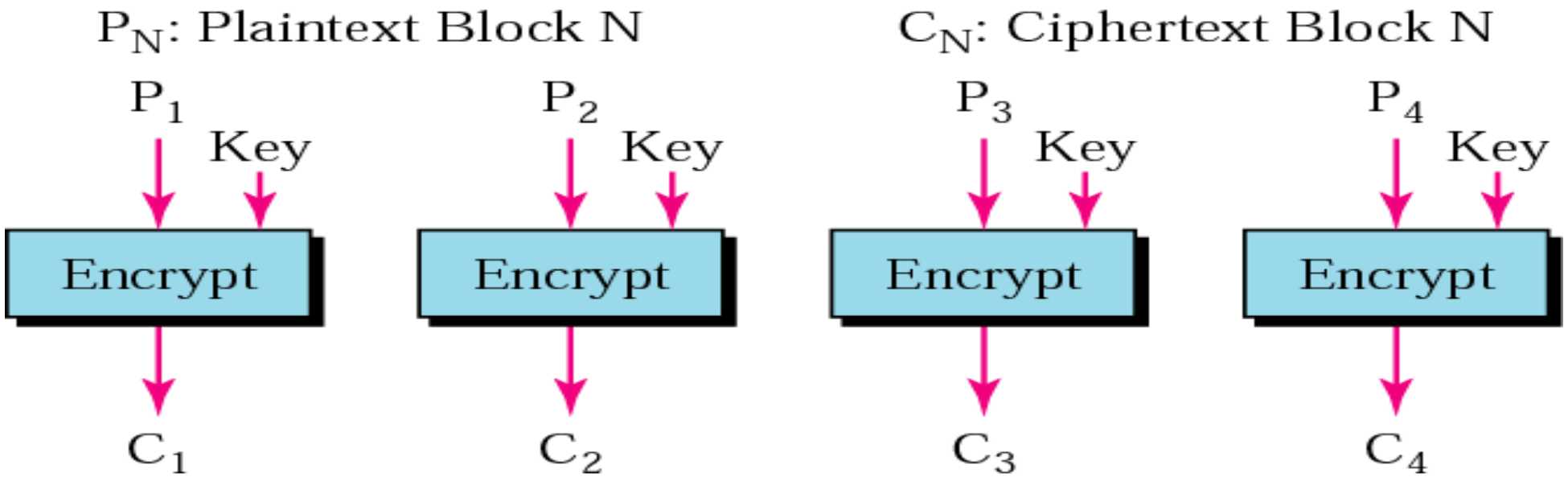
Notasi Kunci

- Algoritma enkripsi
  - E : kunci x plain → cipher
  - Notasi : E = K(m)
- Algoritma dekripsi
  - D : kunci x cipher → plain
- D menginvers E
  - D = K(K(m)) = m
- Gunakan huruf besar K untuk kunci rahasia
- Seringkali algoritma E sama dengan D

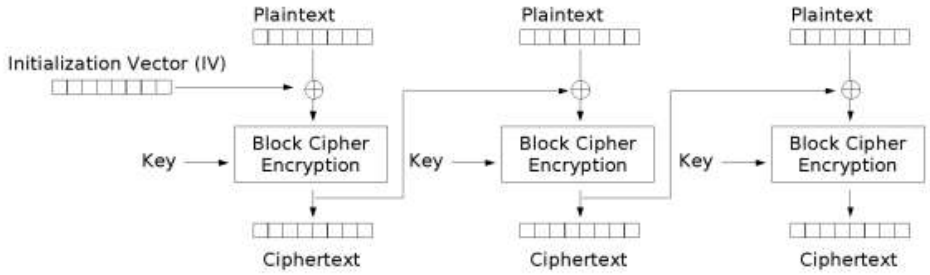
Mode Cipher Blok

- ECB – Electronic Code Book
  - Membagi plaintext ke blok-blok
  - Enkrip setiap blok secara terpisah dengan kunci yang sama
- CBC – Cipher Block Chaining
  - XOR setiap blok dengan ekripsi dari blok sebelumnya
  - Gunakan vektor inisialisasi IV untuk blok pertama
- OFB – Output Feedback
  - Iterasi enkripsi dari IV untuk menghasilkan stream cipher
- CFB – Cipher Feedback
  - Blok output y*i* = input x_i xor enkripsi $K(y_{i-1})$

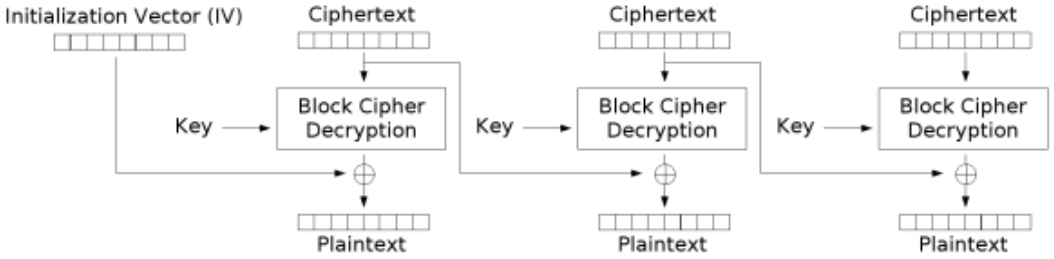
Mode Electronic Code Book (ECB)



Mode Cipher Block Chaining (CBC)



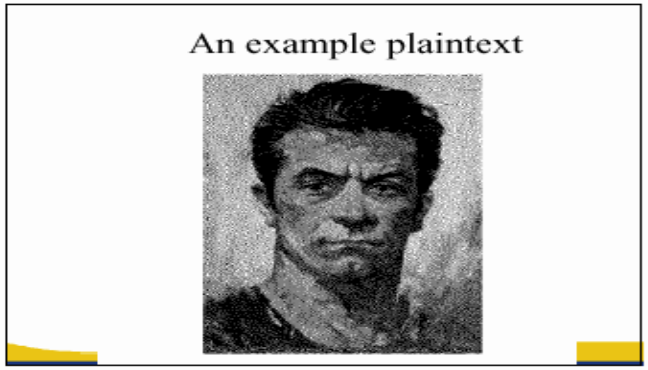
Cipher Block Chaining (CBC) mode encryption

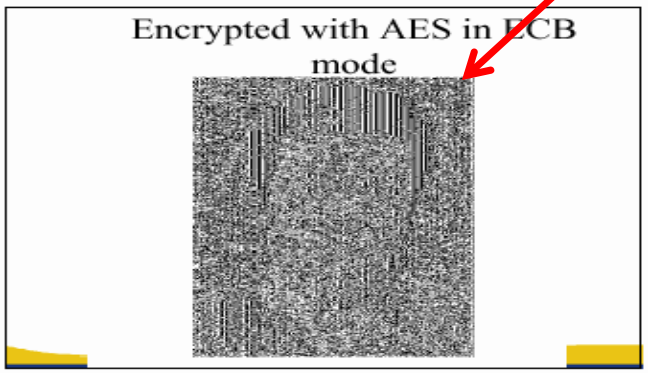


Cipher Block Chaining (CBC) mode decryption

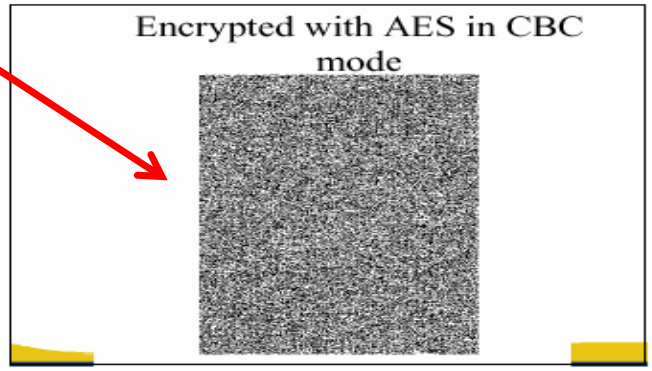
Hasil ECB & CBC

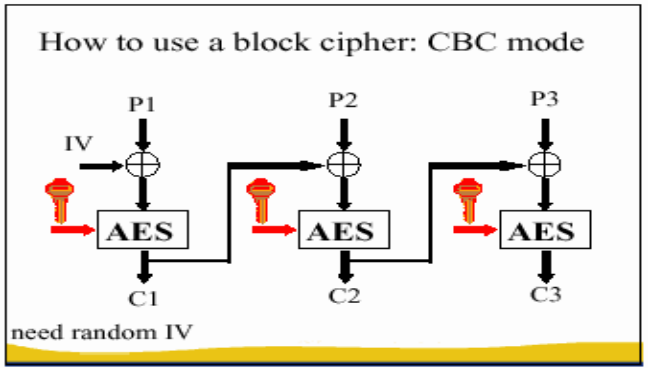
Blok plaintext yang sama menghasilkan ciphertext yang mirip (masih terlihat bentuk kepala)



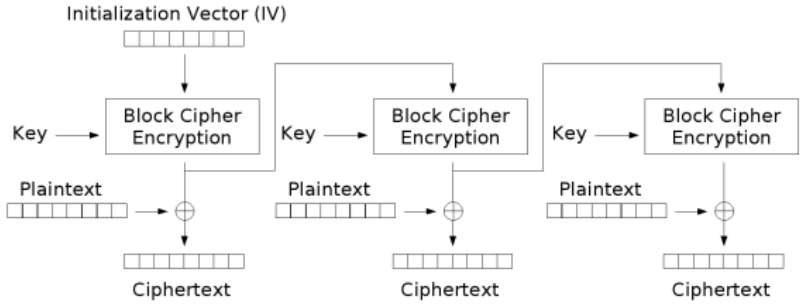


Sudah tidak ada kemiripan / pola di cipertext

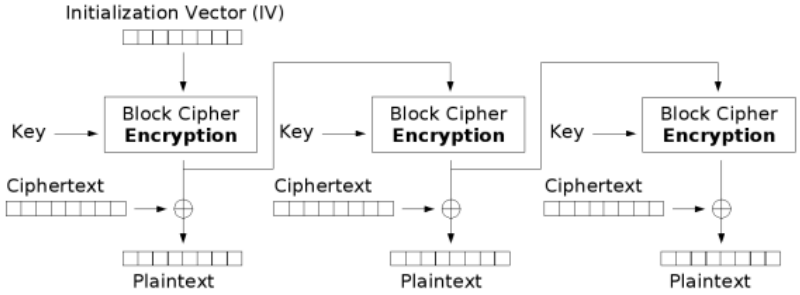




Mode Output Feedback (OFB)

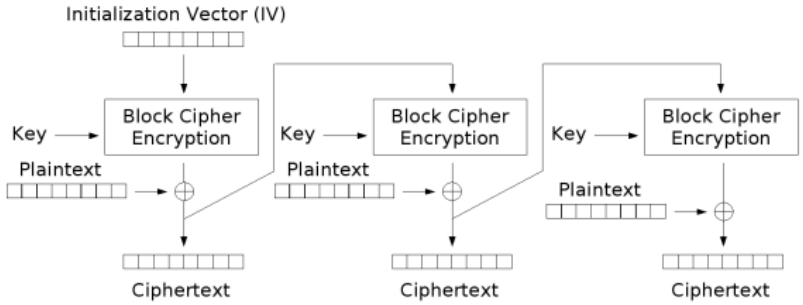


Output Feedback (OFB) mode encryption

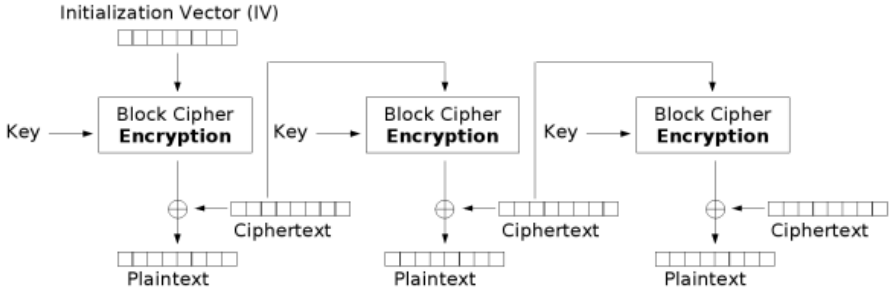


Output Feedback (OFB) mode decryption

Mode Cipher Feedback (CFB)



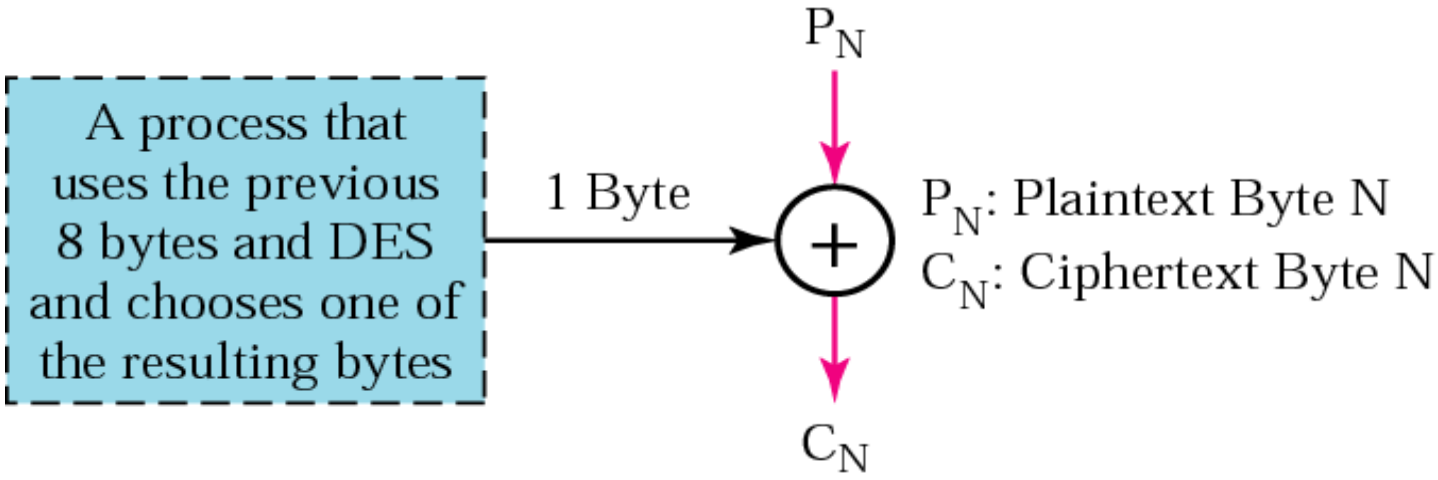
Cipher Feedback (CFB) mode encryption



Cipher Feedback (CFB) mode decryption

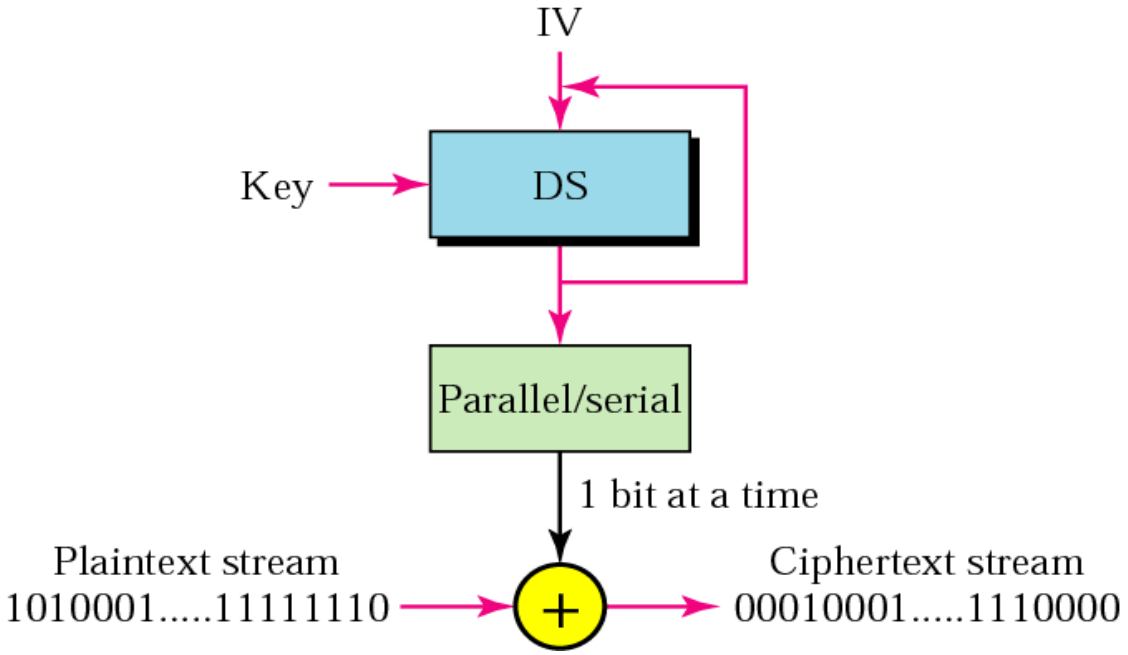
Ciphertext Function Module

- Rotating keys, pada proses ini kunci mengalami perubahan untuk urutan proses
- Komputasi lebih ringan karena yang diubah adalah kunci



Ciphertext Stream Module

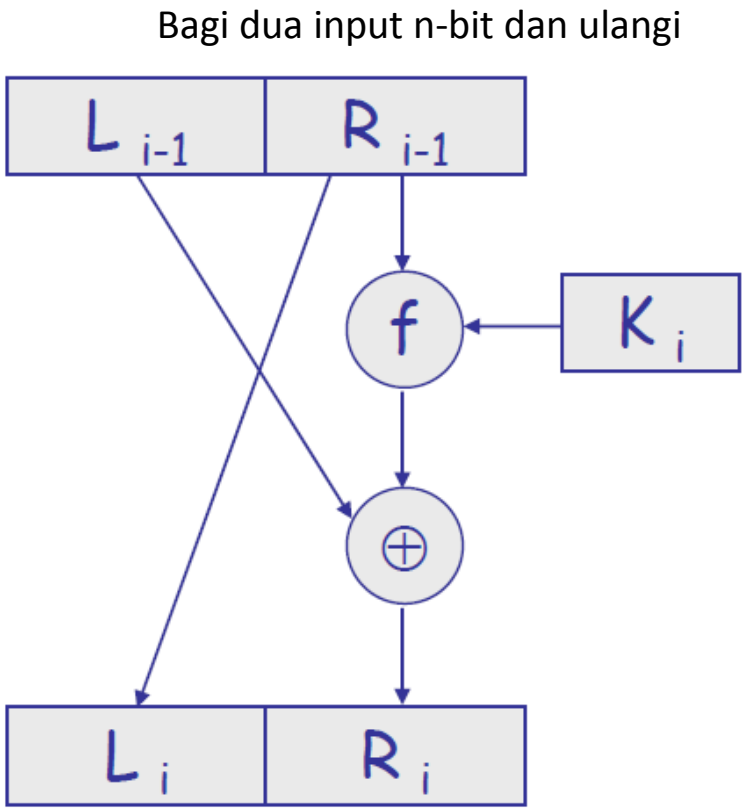
Seperti CFM, hanya digunakan untuk serial stream



Jaringan Feistel

- Banyak (kebanyakan?) chiper blok adalah jaringan Feistel
  - Contoh
    - DES, Lucifer, FREAL, khufu, Khafre, LOKI, GOST, CAST, Blowfish
  - Jaringan feistel adalah bentuk standar untuk
    - Mengiterasi sebuah fungsi f pada bagian dari pesan
    - Menghasilkan transformasi yang bisa dibalik
- AES (Rijndael) berhubungan
  - Juga merupakan cipher blok dengan pengulangan ronde
  - Tapi bukan jaringan Feistel

Jaringan Feistel : Ronde Satu



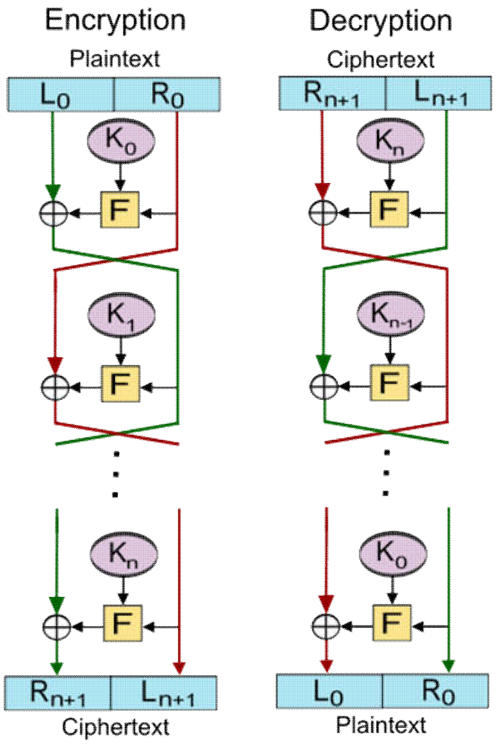
Skema memerlukan

- Fungsi f(R_{i-1},K*i*)
- Komputasi untuk K*i*

Kelebihan

- Kalkulasi sistematik
  - Mudah jika f sebuah tabel
- Bisa dibalik jika Ki diketahui
  - Dapatkan R_{i-1} dari L_i
  - Hitung f(R_{i-1},K_i)
  - Hitung L_{i-1} dengan xor

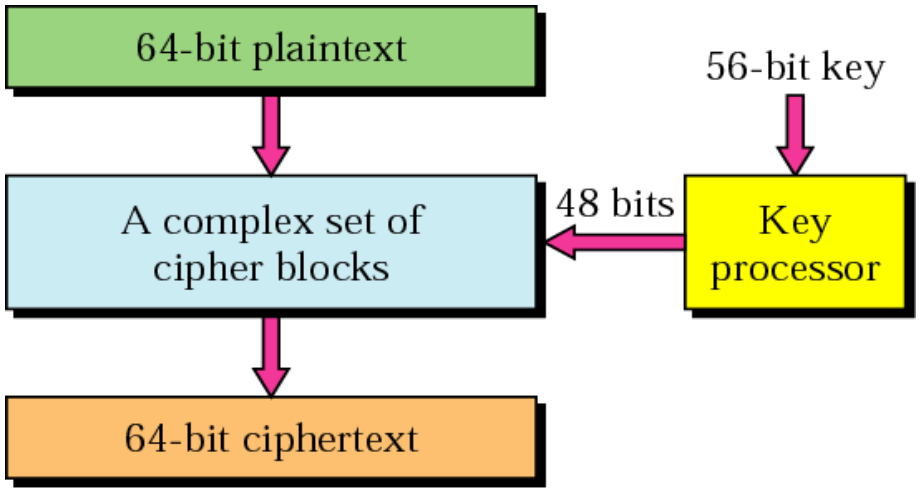
Jaringan Feistel : n Ronde



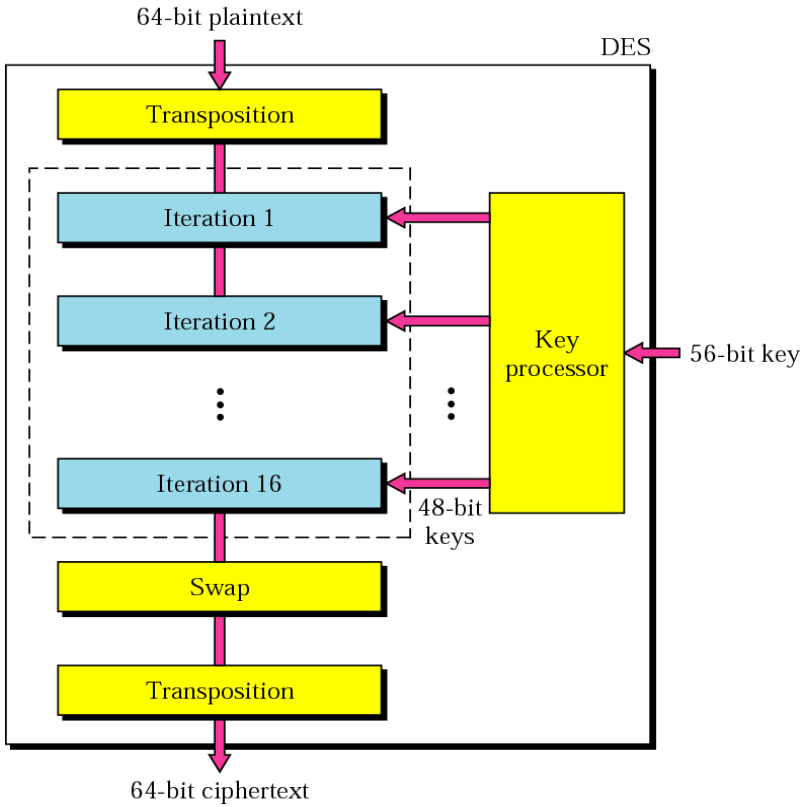
Data Encryption Standard (DES)

- Dirancang oleh IBM (turunan langsung dari Lucifer)
  - Diadopsi menjadi standar pada 1976
  - Keamanan dianalisa oleh NSA
- Panjang kunci 56 bit
  - Dibulatkan ke 64 bit menggunakan 8 pariti bit
- Struktur Feistel
  - Bekerja di blok 64 bit
  - Melakukan permutasi pada bit input
  - Melakukan fungsi subtitusi dengan S-box berkali kali
  - Setiap siklus menggunakan permutasi dan subtitusi untuk mengkombinasikan plaintext dengan kunci
  - Menggunakan permutasi balik untuk menghasilkan output

Arsitektur DES



Arsitektur DES



Pemrosesan Kunci (Key Processor)

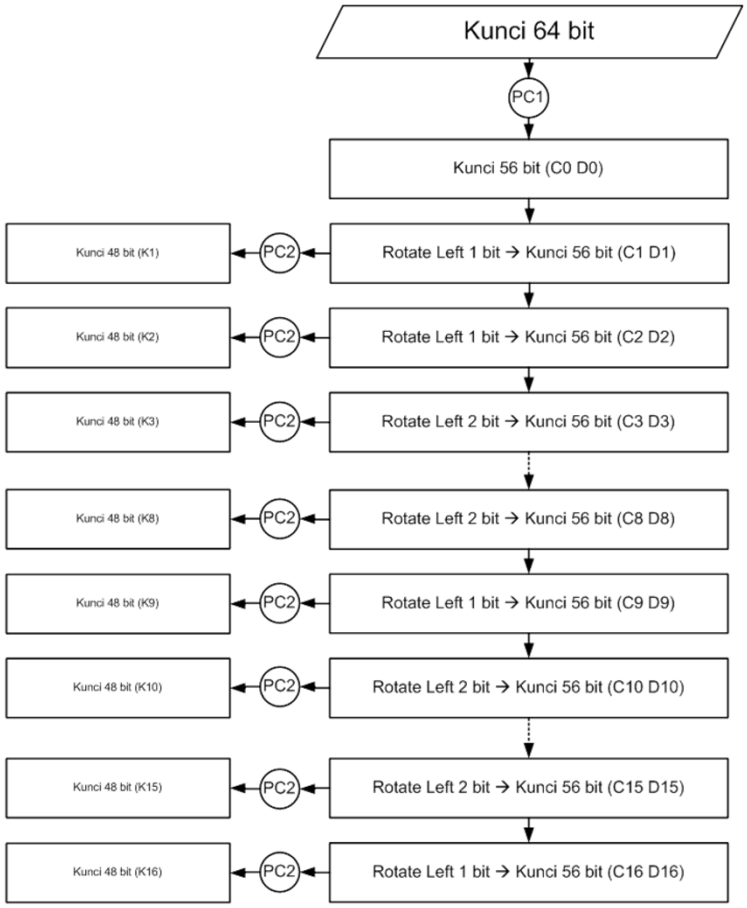
Kunci 64 bit mengalami:

1x Permutasi dengan PC1

dan

16x Permutasi dengan PC2

Iteration Number	Number of Left Shifts	Iteration Number	Number of Left Shifts
1	1	9	1
2	1	10	2
3	2	11	2
4	2	12	2
5	2	13	2
6	2	14	2
7	2	15	2
8	2	16	1



Pola Permutasi Awal (*Initial Permutation - IP*)

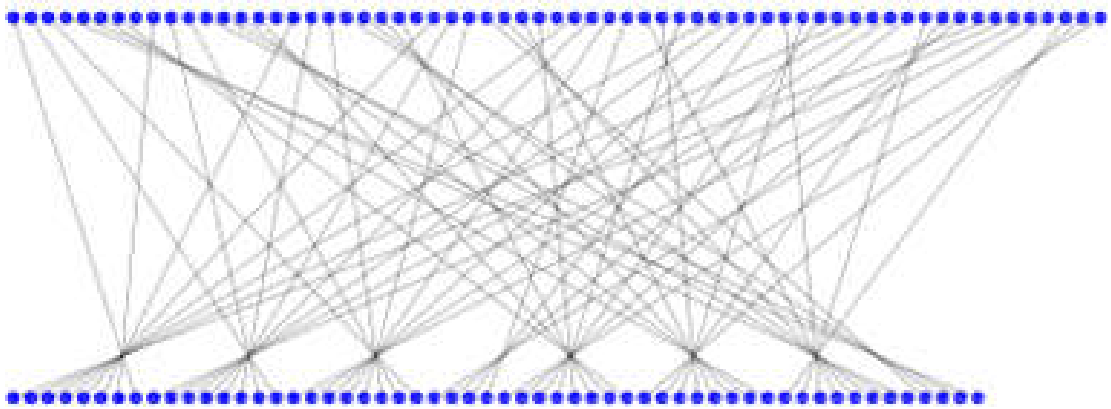
- Plain teks dibagi menjadi blok-blok 64 bit dan setiap blok akan di lakukan permutasi awal sesuai matrik IP ini

Bit	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	58	50	42	34	26	18	10	2
9	60	52	44	36	28	20	12	4
17	62	54	46	38	30	22	14	6
25	64	56	48	40	32	24	16	8
33	57	49	41	33	25	17	9	1
41	59	51	43	35	27	19	11	3
49	61	53	45	37	29	21	13	5
57	63	55	47	39	31	23	15	7

PC-1

Kunci 64 bit diubah menjadi 56 bit dengan menggunakan permutasi PC-1

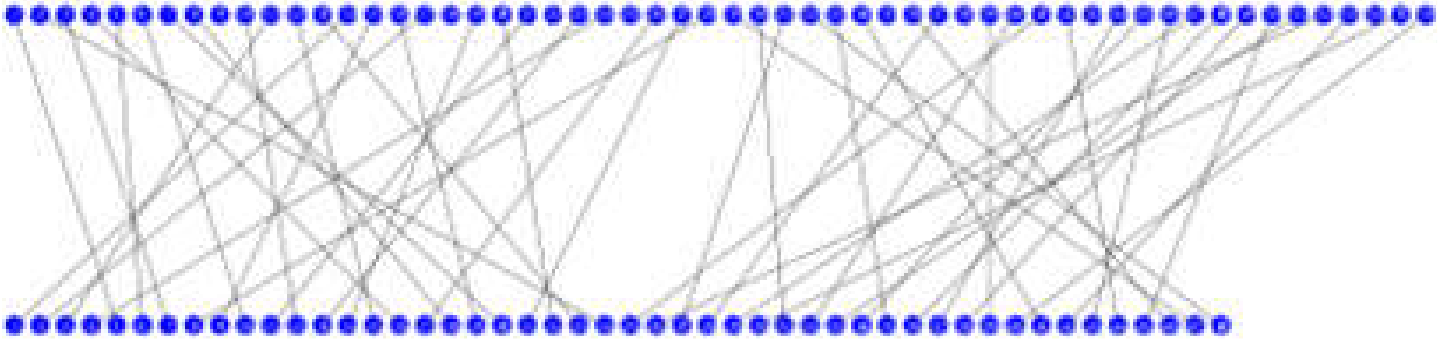
Bit	0	1	2	3	4	5	6
1	57	49	41	33	25	17	9
8	1	58	50	42	34	26	18
15	10	2	59	51	43	35	27
22	19	11	3	60	52	44	36
29	63	55	47	39	31	23	15
36	7	62	54	46	38	30	22
43	14	6	61	53	45	37	29
50	21	13	5	28	20	12	4



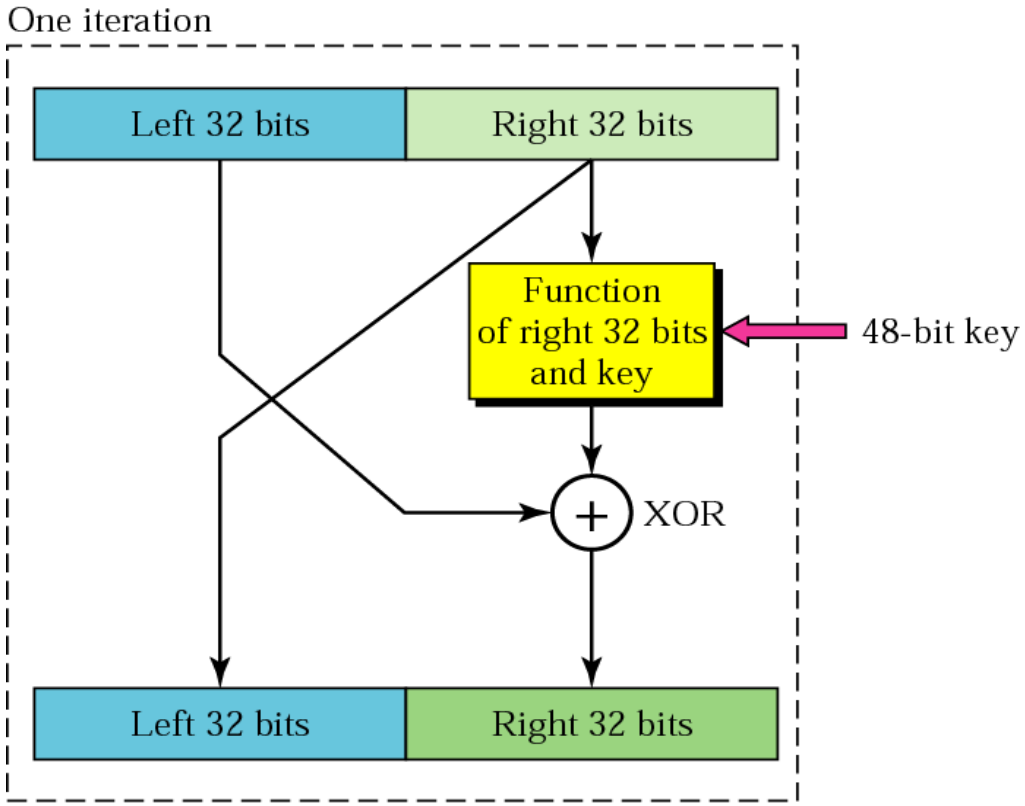
PC-2

Kunci 56 bit diambil 48 bit dengan menggunakan permutasi PC-2

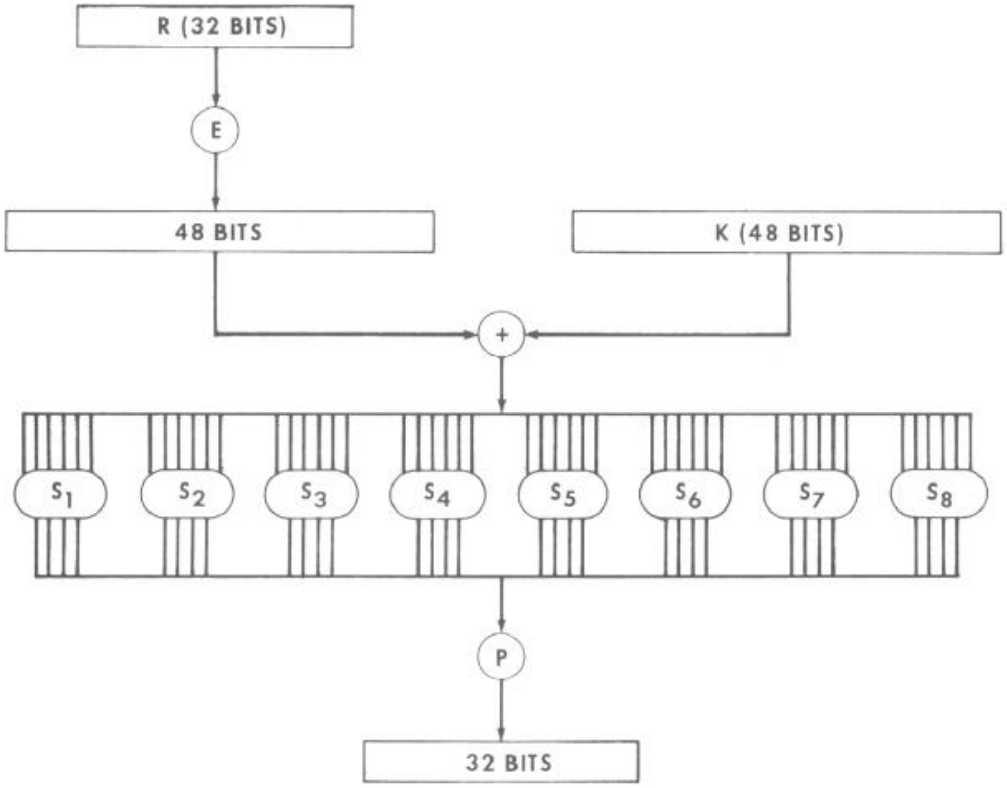
Bit	0	1	2	3	4	5
1	14	17	11	24	1	5
7	3	28	15	6	21	10
13	23	19	12	4	26	8
19	16	7	27	20	13	2
25	41	52	31	37	47	55
31	30	40	51	45	33	48
37	44	49	39	56	34	53
43	46	42	50	36	29	32



Blok Iterasi



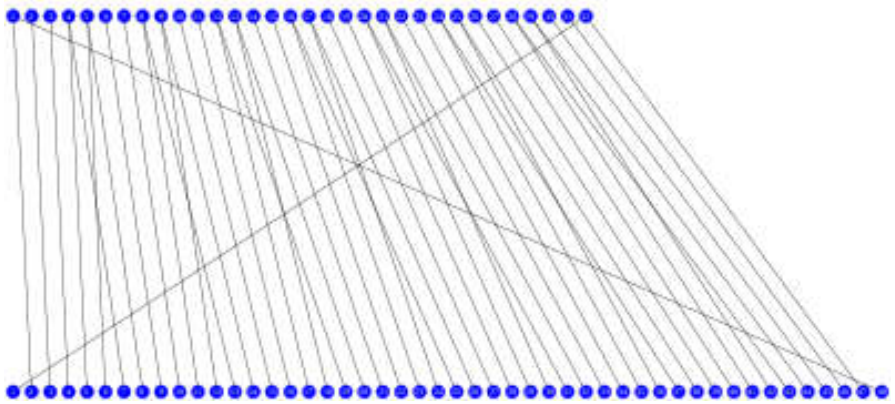
Function of right 32 bits and key



Fungsi E

32 bit data kanan (*R*) di kembangkan menjadi 48 bit dengan fungsi E

Bit	0	1	2	3	4	5
1	32	1	2	3	4	5
7	4	5	6	7	8	9
13	8	9	10	11	12	13
19	12	13	14	15	16	17
25	16	17	18	19	20	21
31	20	21	22	23	24	25
37	24	25	26	27	28	29
43	28	29	30	31	32	1



S-Box DES

- Tabel subtitusi
- Input 6 bit diganti dengan output 4 bit
- Subtitusi yang dilakukan tergantung bit input

- Diimplementasikan sebagai *lookup table*
  - 8 S-Box
  - Setiap S-Box mempunyai tabel 64 inputan
  - Setiap inputan menentukan output 4 bit

Fungsi S

S1

Row No.	Column Number															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Fungsi S

- Hasil operasi data 48 bit dari fungsi E dan kunci K akan dibagi menjadi potongan 6 bit dan direduksi menjadi 4 bit dengan fungsi Sn sehingga akan didapatkan reduksi 48 bit ke 32 bit,
- contoh : potongan pertama 6 bit 011011 akan direduksi dengan fungsi S1 dengan cara diambil MSB 0 dan LSB 1 menjadi 01 menyatakan nomor baris, sisa 1101 dibaca sebagai desimal 13 menjadi nomor kolom, isi baris 1 dan kolom 13 adalah 5 yang dalam biner adalah 0101, sehingga S1(011011) = 0101

Fungsi S (S1 – S8)

S1

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

S2

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

S3

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

S4

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

S5

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	6
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

S6

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

S7

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

S8

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

Dekripsi DES

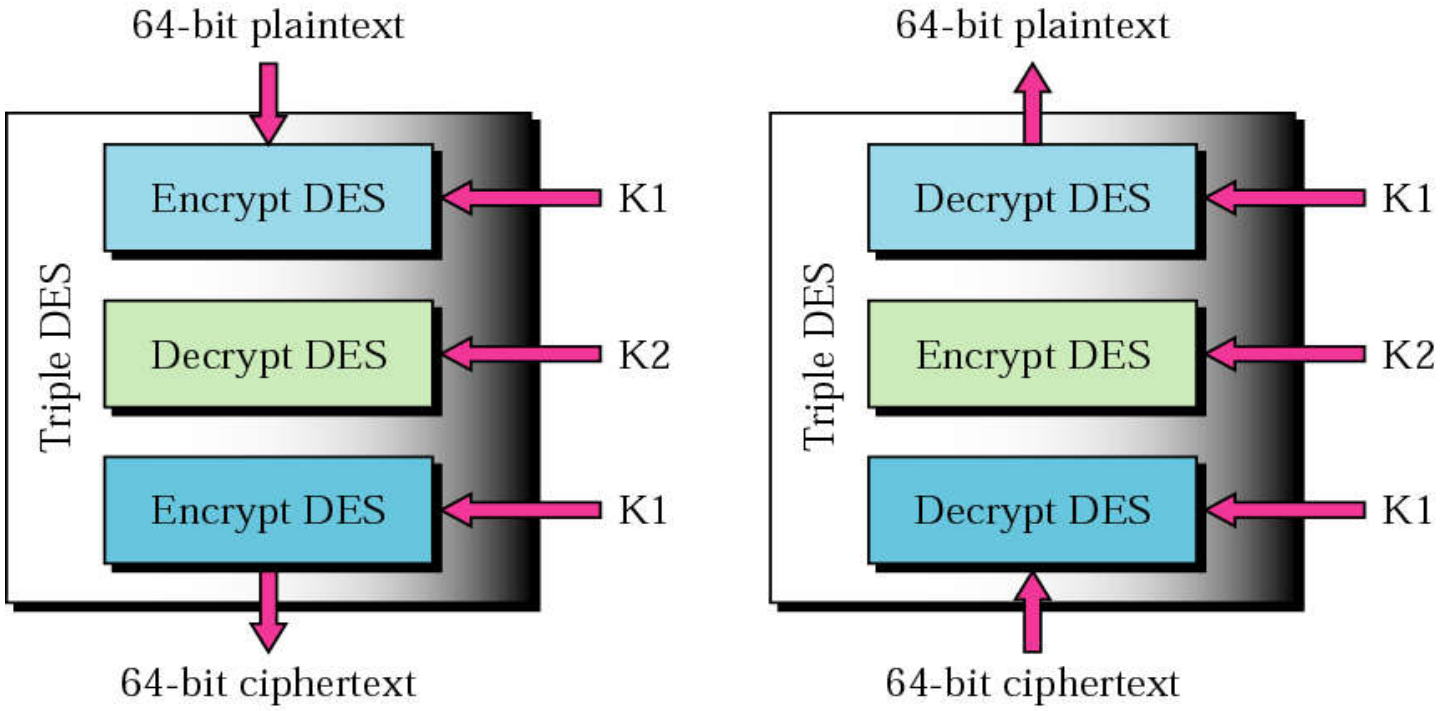
- Menggunakan algoritma yang sama dengan enkripsi hanya saja menggunakan kunci yang dibalik urutan bitnya

- Kunci enkripsi k_1 k_2 … k_{16}
- Kunci dekripsi k_{16} k_{15} … k_1

Kekurangan DES

- Kunci 56 bit terlampau pendek
  - www.distributed.net memecahkan tantangan DES tahun 1999 dibawah 24 jam
  - EFF merancang chip pemecah DES
- Masalah lain
  - Tidak semua kunci aman
- Umum digunakan **Triple-DES** untuk mengatasi masalah ini (Triple disini berarti sangat literal : **diulangi 3 kali**)

Triple DES



a. Encryption triple DES

ciphertext = $E_{K3}(D_{K2}(E_{K1}(\text{plaintext})))$

b. Decryption triple DES

plaintext = $D_{K1}(E_{K2}(D_{K3}(\text{ciphertext})))$

Pemilihan Kunci

- Opsi Kunci:
  - Ketiga kunci berbeda
  - K1 dan K2 berbeda dan K3 = K1
  - Semua kunci sama, K1 = K2 = K3

- Opsi [1] : Paling Kuat, memiliki 3 kunci, masing -masing memiliki panjang 56 bits = 168 bits kunci yang independent
- Opsi [2] : Cukup Kuat, memiliki 2 kunci, masing – masing memiliki panjang 56 bits = 112 bits kunci. Lebih kuat 2x dari DES
- Opsi [3] : Sama seperti DES

AES-Rijndael

- Advanced Encryption Standard a.k.a Rijndael
- Established by U.S National Institute of Standard and Technology in 2001
- AES dasar merupakan subset dari Rijndael cipher
- AES menjadi standard untuk federal government dan disetujui oleh NSA.
- Penemu Rijndael : Joan Daemen & Vincent Rijmen

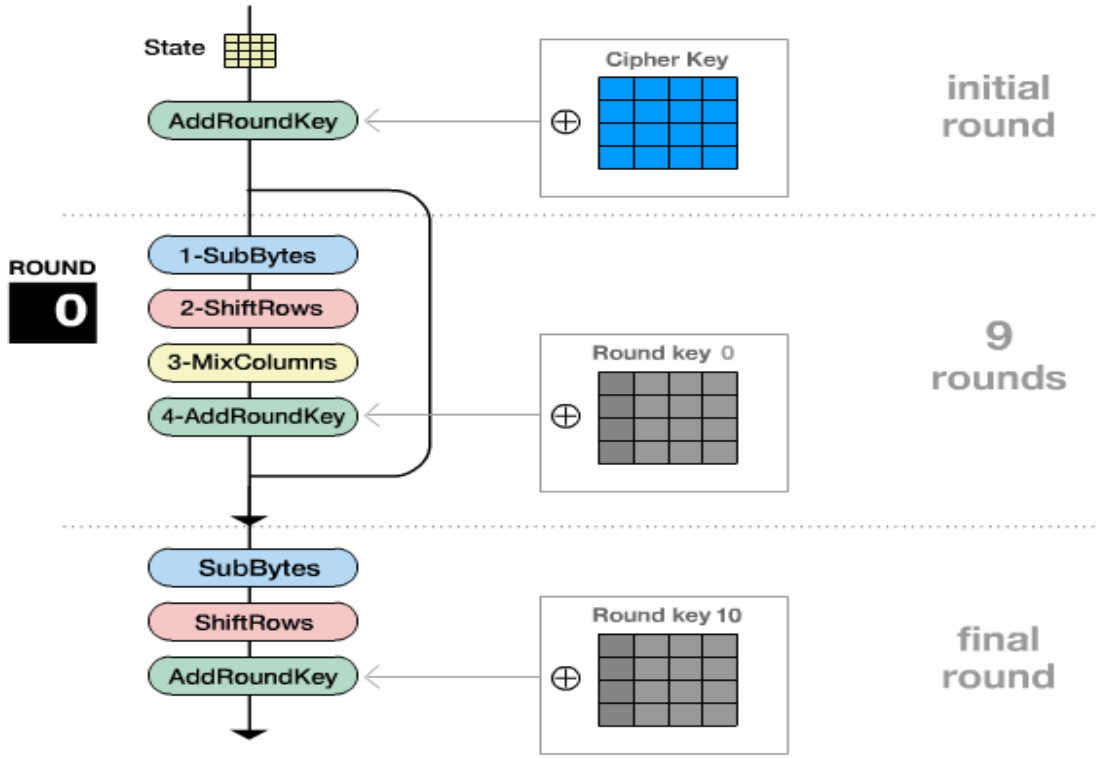
AES-Rijndael

- AES beroperasi menggunakan block berukuran 128 bit, dan menggunakan kunci 128, 192, atau 256 bit
- Perbedaan mencolok antara DES dan AES adalah pada AES tidak menggunakan jaringan feistel
- Sebagai contoh jika terdapat 16 bytes, b_0-b_{15}, byte-byte ini direpresentasikan sebagai matriks

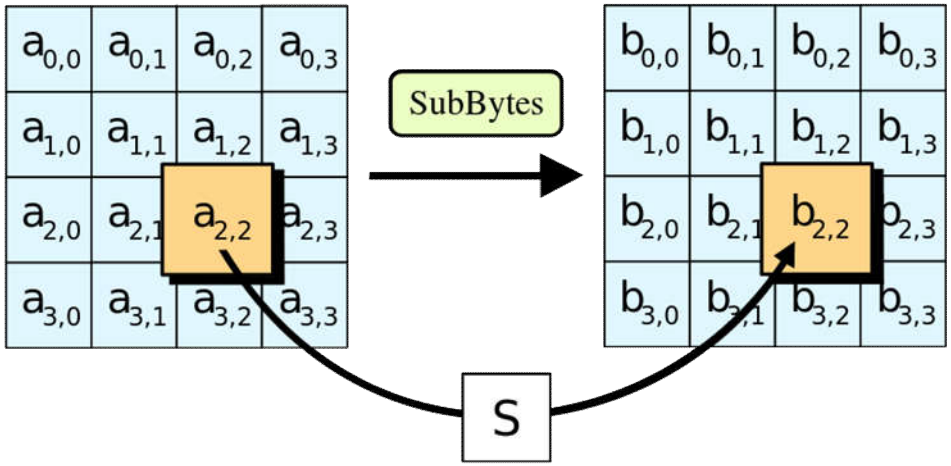
$$\begin{bmatrix} b_0 & b_4 & b_8 & b_{12} \\ b_1 & b_5 & b_9 & b_{13} \\ b_2 & b_6 & b_{10} & b_{14} \\ b_3 & b_7 & b_{11} & b_{15} \end{bmatrix}$$

Nantinya proses enkripsi akan diulang sebanyak :
10 cycles of repetition for 128-bit keys.
12 cycles of repetition for 192-bit keys.
14 cycles of repetition for 256-bit keys.

AES-Rijndael

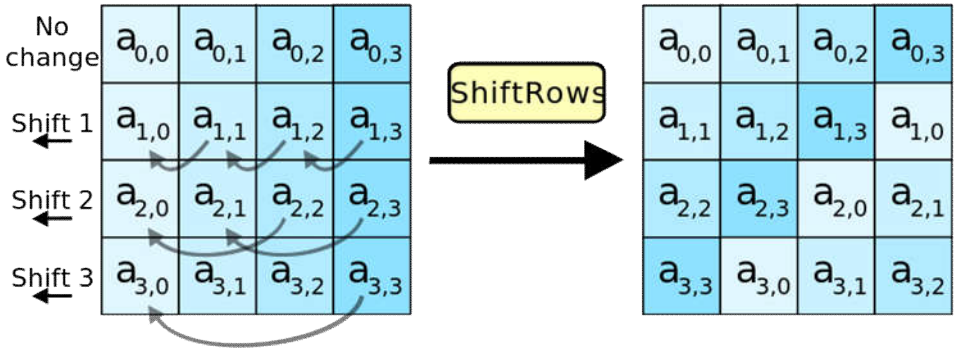


AES-Rijndael (Sub-bytes)



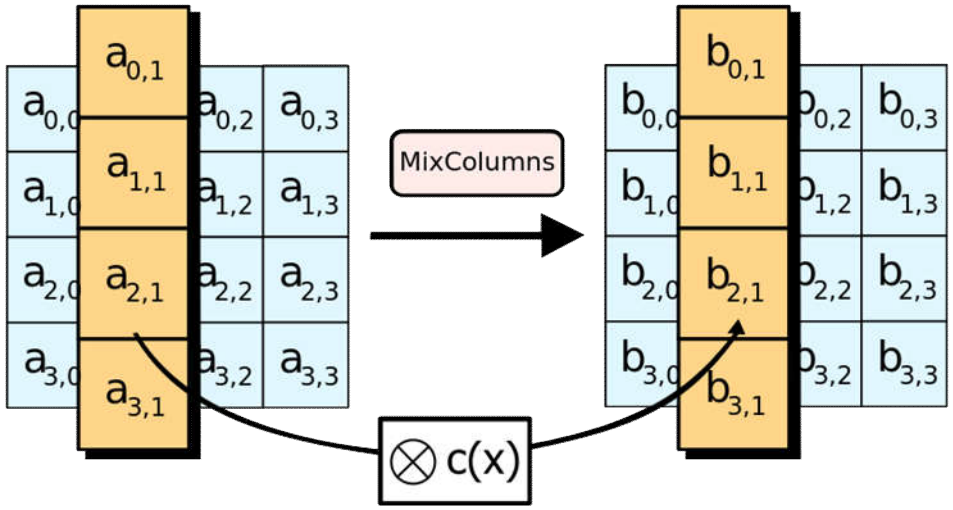
- Setiap $A_{i,j}$ disubtitusikan dengan SubByte $S(A_{i,j})$ dengan menggunakan 8-bit Rijndael S-Box
- $S(A_{i,j})$ tidak sama dengan $A_{i,j}$

AES-Rijndael (Shift Rows)



- Baris pertama tidak dirubah
- Baris kedua digeser kekiri sebanyak 1 kali
- Baris ketiga dan keempat digeser sebanyak 2 dan 3 kali
- Berbeda ketika menggunakan 256 bit blocks, baris kedua, ketiga dan keempat digeser sebanyak 1,3, dan 4 kali

AES-Rijndael (Mix Columns)

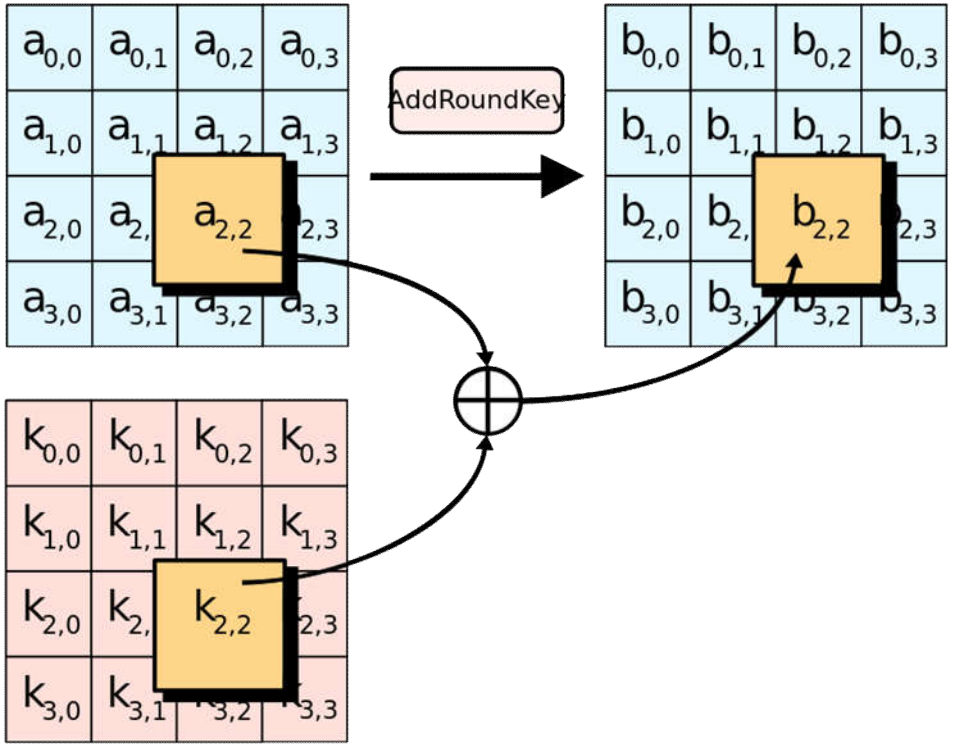


$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

$$b_0 = 2a_0 + 3a_1 + 1a_2 + 1a_3$$
$$b_1 = 1a_0 + 2a_1 + 3a_2 + 1a_3$$
$$b_2 = 1a_0 + 1a_1 + 2a_2 + 3a_3$$
$$b_3 = 3a_0 + 1a_1 + 1a_2 + 2a_3$$

- Setiap kolom akan dikombinasikan dengan menggunakan transformasi linear
- Menggunakan 4 byte sebagai input dan menghasilkan 4 byte sebagai output
- Menggunakan Fixed Matrix

AES-Rijndael (Add Round Keys)



- Subkey dikombinasikan dengan state yang ada
- Dalam setiap round, subkey didapat dari kunci utama dari penjadwalan kunci Rijndael
- Subkey ditambahkan dengan melakukan kombinasi setiap byte pada saat state saat ini dengan byte subkey dengan menggunakan bitwise XOR

Kesimpulan Kunci Simetrik

- Basis komputasi sederhana → S-BOX dan operasi XOR
- Kunci umumnya dilakukan rotasi yang menyebabkan kunci ‘berubah’ untuk setiap iterasi
- Iterasi dilakukan ‘cukup’ banyak → semakin banyak iterasi semakin ‘aman’
- Data ada yang di ‘rotasi’ dan ada yang tidak di ‘rotasi’
- Enkripsi dan dekripsi dilakukan dengan kunci yang sama, yang berbeda umumnya adalah pola rotasi saat enkripsi dan dekripsi

Kriptografi Kunci Asimetrik

- Pengirim mengenkripsi menggunakan kunci publik
- Penerima mendekripsi menggunakan kunci pribadi (private)
- Hanya kunci pribadi yang harus dirahasiakan
  - Kunci publik bisa didistribusi secara bebas
- Contoh terkenal RSA

Kriptografi Kunci Asimetrik

- Permasalahan yang ada pada kunci simetrik:
  - Penerima dan Pengirim harus berbagi kunci (yang rentan sekali dicuri oleh orang lain)
- Dengan adanya kriptografi kunci asimetrik, permasalahan ini tidak ada
  - Satu sisi, dapat melakukan proses enkripsi dengan menggunakan kunci publik
  - Di sisi lain, dapat diterima dan dilakukan proses dekripsi dengan menggunakan kunci pribadi (private)

Notasi Kunci

- Algotitma Enkripsi
  - E : keyPub x plain → cipher
  - Notasi : K(msg) = E(K, msg)
- Algoritma dekripsi
  - D : keyPriv x cipher → plain
  - Notasi : k(msg) = D(k,msg)
- D menginversi E
  - D(k, E(K,msg)) = msg
- Digunakan notasi K untuk kunci publik
- Digunakan notasi k untuk kunci pribadi
- E bisa menggunakan aolgoritma yang sama dengan D

Pro & Kontra untuk Kriptografi Kunci Asimetrik

- Memerlukan komputasi yang lebih intensif dari kirptografi kunci simetrik
  - Algoritma lebih sulit untuk diimplementasikan
  - Memerlukan mesin yang lebih kuat (min. processor yg digunakan utk berhitung)
- Kesulitan ditentukan oleh kompleksitas komputasi
- Memerlukan pasangan satu kunci pribadi dan satu kunci publik
  - Jumlah total pasangan kunci yang aman di komunikasi adalah (n)

Algoritma RSA

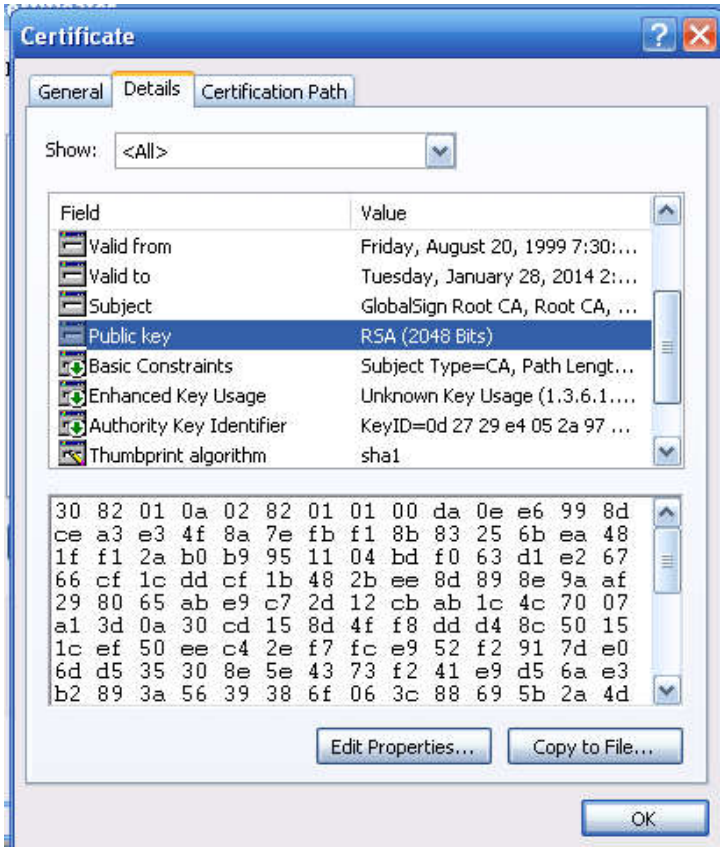
- Ron Rivest, Adi Shamir, Leonard Adleman
  - Diusulkan 1979
  - Memenangkan award Turing 2002

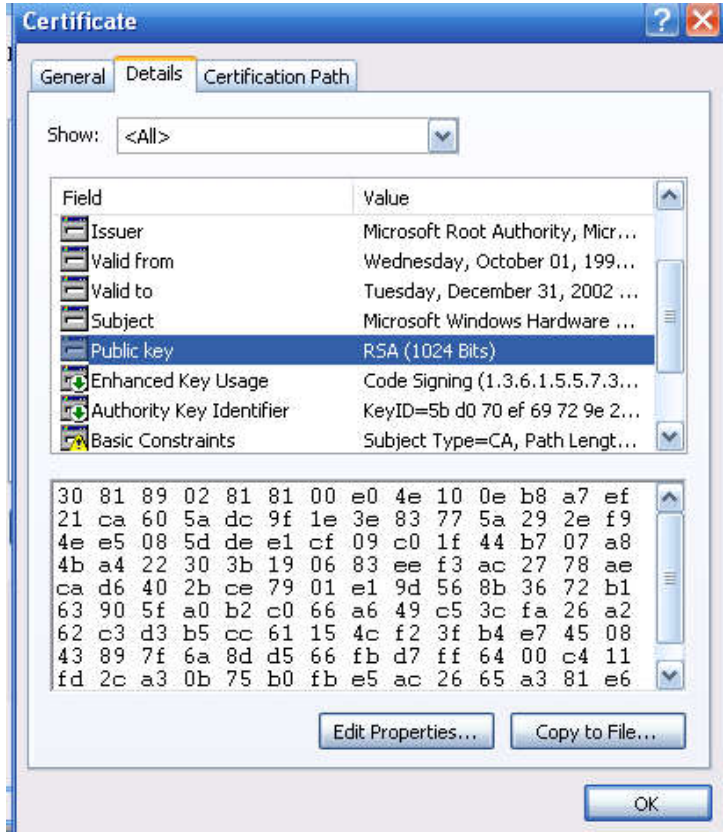
- Implementasi hardware
  - 1000 x lebih pelan dari DES

RSA

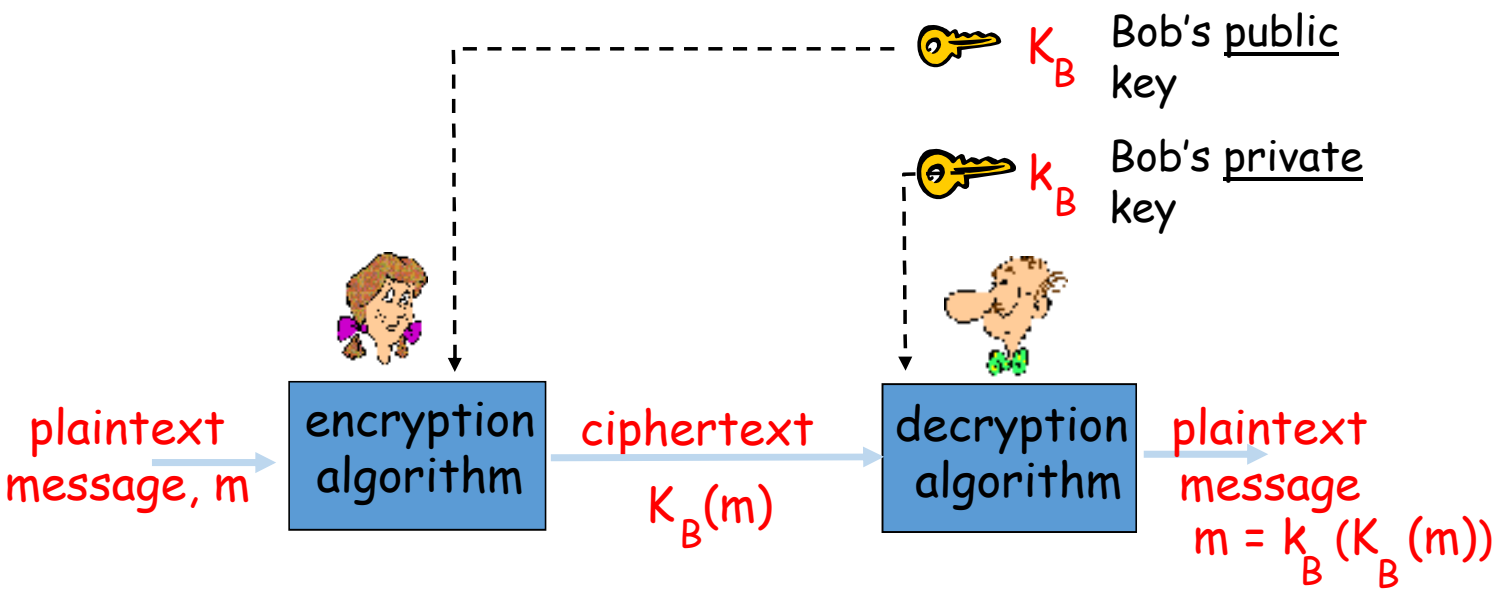
- Pasangan kunci diturunkan dari bilangan prima rahasia
  - Kunci umumnya ≥ 256 bit (512, 1024, 2048)
- Pesan plaintext
  - Diperlakukan sebagai sebuah bilangan biner yang sangat besar
- Untuk memecahkan enkripsi, harus dicari kunci yang merupakan pasangan dari bilangan yang sangat besar tadi
- Dianggap sangat sulit untuk melakukan itu

Penggunaan kunci sangat panjang di RSA





Kriptografi Kunci Asimetrik



RSA : Pemilihan Kunci

1. Pilih dua bilangan prima yang sangat besar *p, q.*
   (misal, masing-masing 1024 bit)

2. Hitung $n = pq$, $z = (p-1)(q-1)$

3. Pilih *e* $(e<n)$ yang tidak punya faktor yang sama dengan z.
   (*e, z* "prima relatif").
   e adalah bilangan prima relatif terhadap z
4. Pilih *d* sehingga *ed-1* bisa dibagi *z*.
   (*ed* mod *z* = *1*).

5. Kunci *Publik* $\underbrace{(n,e).}_{K_B^+}$ Kunci *pribadi* $\underbrace{(n,d).}_{K_B^-}$

RSA: Enkripsi dan Dekripsi

0. Menggunakan (n,e) dan (n,d) yang sudah dihitung tadi

1. Enkripsi pola bit, m, hitung

   $c = m^e \bmod n$ (sisa m^e dibagi n)

2. Dekripsi pola bit diterima, c, hitung

   $m = c^d \bmod n$ (sisa c^d dibagi n)

Contoh RSA

Bob memilih *p=5, q=7*. Maka *n=35, z=24*.

e=5 (sehingga *e, z* prima relatif).
d=29 (sehingga *ed-1* habis dibagi z).

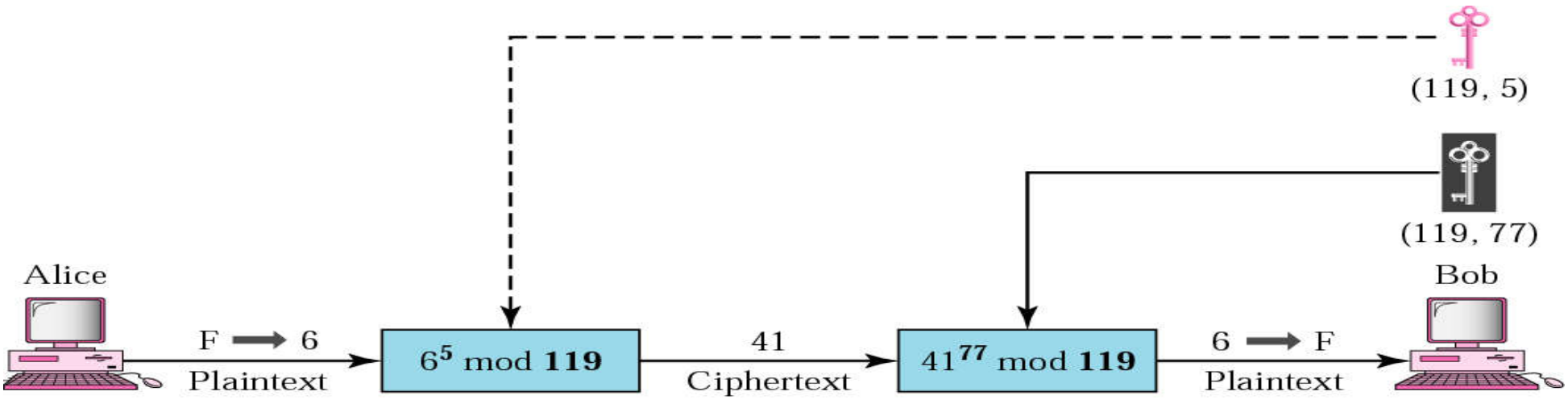
encrypt:

letter	m	m^e	$c = m^e \bmod n$
l	12	248832	17

decrypt:

c	c^d	$m = c^d \bmod n$	letter
17	481968572106750915091411825223071697	12	l

RSA


(119, 5)
(119, 77)
Alice
Bob
F → 6
Plaintext
6^5 mod 119
41
Ciphertext
41^{77} mod 119
6 → F
Plaintext

Cara mengasilkan bilangan prima

- Banyak cara, Rabin-Miller test prima umum digunakan
- Test yang efisien dengan kemungkinan memeriksa sebuah bilangan prima adalah ¾
  - Iterasi test Rabin-Miller t kali
  - Kemungkinan bilangan tersebut bukan prima adalah $(1/4)^t$
- Dalam praktisnya (hanya memerlukan beberapa detik untuk mendapatkan prima 512bit)
  - Hasilkan bilangan random n bit : p
  - Ubah LSB dan MSB ke 1 (untuk memastikan menjadi bilangan ganjil)
  - Periksa apakah prima dengan dibagi 3,5,7 sd <2000
  - Lakukan test Rabin Miller paling tidak 5 kali

Penggunaan lain Kripto Kunci Asimetrik

- Pengirim (A) mengirim $k_A(m)$
- Penerima (B) membuka $K_A(k_A(m))$

- Apa tujuan dari komunikasi ini?
- Untuk melakukan proses ***autentikasi*** yaitu meyakinkan penerima bahwa yang mengirim benar-benar / hanya A

Kritografi Kunci Hybrid

- Memadukan kriptografi simetrik dengan asimetrik dengan tujuan mengambil kelebihan dari kedua sistem : kemudahan komputasi simetrik dan keamanan asimetrik
- Kunci simetrik sebagai kunci komunikasi / kunci sesi dikirim sebagai pesan komunikasi asimetrik

- Pengirim : $K^{+}(K_{AB})$
- Penerima : $K^{-}(K^{+}(K_{AB}))$

- Untuk menambah keamanan komunikasi bisa dibagi menjadi banyak sesi yang masing2 menggunakan kunci simetrik yang berbeda

Homomorphic Encryption

- E=K(m)

- M1 → K(M1)
- M2 → K(M2)

- K(M1) + K(M2)
  → E1 + E2 = E3

M3 = K(E3)