KEAMANAN JARINGAN

- pengantar
- ANCAMAN KEAMANAN
- PERLINDUNGAN JARINGAN
  - FIREWALL
- KRIPTOLOGI
  - Kunci
- ENKRIPSI KONVENSIONAL
- Metode enkripsi
  - Chiper Substitusi
  - Chiper Transposisi
  - Elemen Dasar Transformasi
- ALGORITMA ENKRIPSI DES
  - OUTLINE ALGORITMA DES
  - OUTLINE SATU PUTARAN DES
  - ALGORITMA FUNGSI $f(R_{i-1}K_i)$
  - Contoh S-box
- ALGORITMA DEKRIPSI DES
- TANTANGAN DES
- ALGORITMA RSA
  - Pembangkitan kunci
  - ENKRIPSI RSA
  - deKRIPSI RSA
  - Tantangan rsa
- authentisasi
  - Contoh authentisasi

KEMBALI

pengantar

- Sistem komputer dan jaringannya memiliki kerawanan terhadap serangan sepanjang hidupnya.
- Tidak ada sistem berbasis komputer yang mampu mengamankan dirinya dari semua kemungkinan serangan.
- Upaya yang dapat dilakukan adalah membuat hacker atau cracker kesulitan dalam mengganggu sistem.

KEMBALI

- Sistem keamanan jaringan adalah suatu
  - perlindungan data selama transmisi dilakukan,
  - untuk menjamin keaslian data yang ditransmisikan, dan
  - memelihara kerahasiaan, integritas, dan ketersediaan data.

ANCAMAN KEAMANAN

- PASIF : membuka isi pesan, analisis lalu-lintas
- AKTIF : penyamaran, jawaban (efek yang tidak terotorisasi), modifikasi pesan, penolakan layanan (gangguan, pelumpuhan, pengurangan kinerja)

Pendekatan dalam serangan :

- Pemecahan rahasia : analisis karakteristik rahasia, teks asli, teks rahasia untuk memperoleh kunci.
- Brute force (paksaan) : mencoba berbagai kunci yang mungkin untuk membongkar

KEMBALI

PERLINDUNGAN JARINGAN

Perlindungan dan pemeliharaan data dapat diupayakan melalui berbagai teknik seperti :

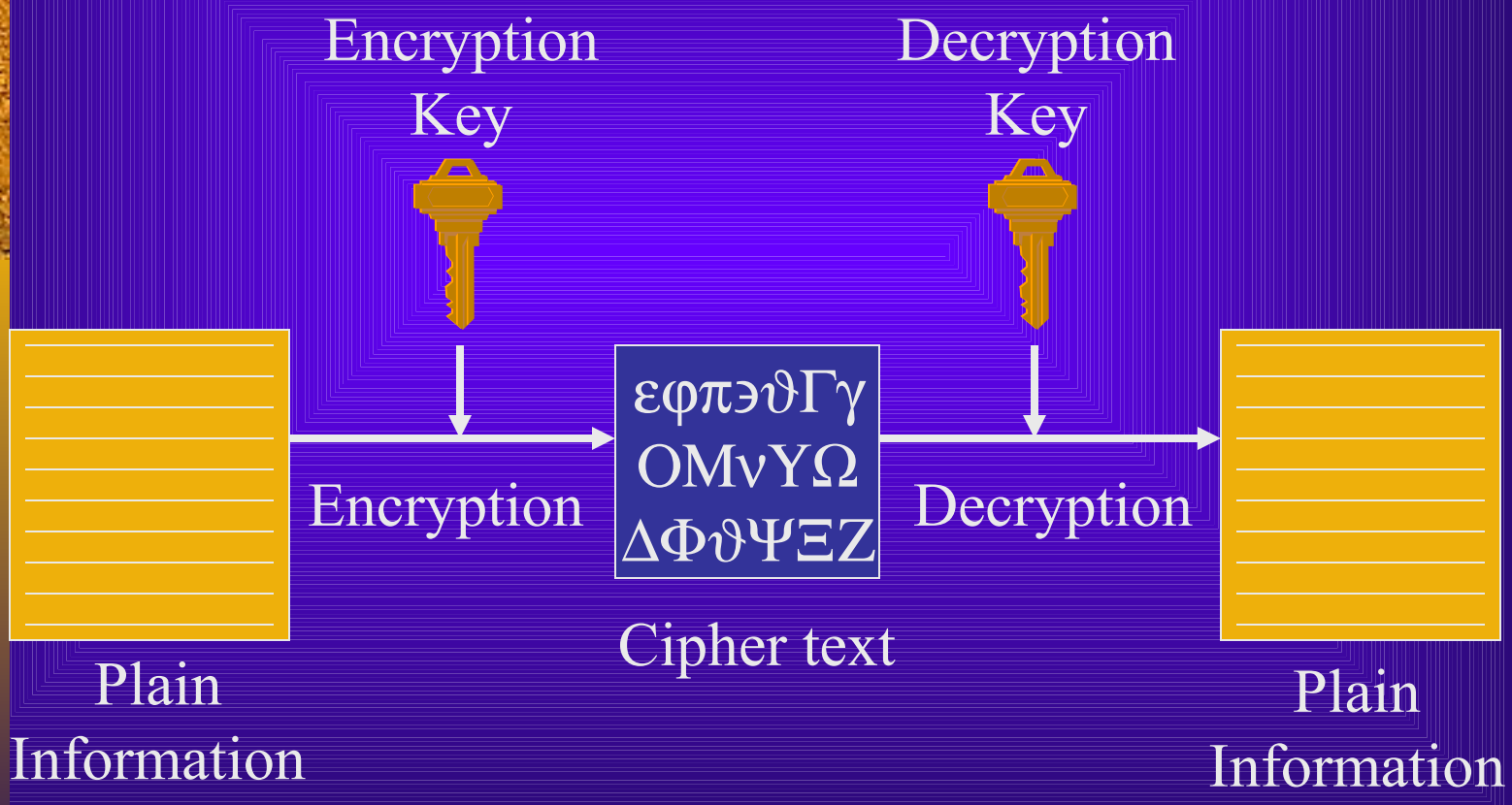
- **LINTASAN PRIBADI** : setiap komputer dihubungkan dengan kabel pribadi yang tidak terhubung dg jaringan manapun.
- **PASSWORD** : setiap pengguna legal (trusted) diberi authoritas untuk masuk ke jaringan dengan kode unik.
- **FIREWALL** : semacam perimeter keamanan jaringan; setiap pesan yang masuk dan keluar harus lewat satu gerbang untuk diperiksa.

  Hal ini dapat diterapkan dengan menggunakan **2 router dan 1 gateway aplikasi**. Router luar untuk memeriksa paket yang masuk, router dalam untuk memeriksa paket yang keluar, dan gateway aplikasi yang akan memeriksa semua pesan masuk dan keluar pada layer aplikasi seperti penyaringan mail, header, ukuran, bahkan teks isi.

  Pemeriksaan didasarkan pada tabel yang disusun admin yang misalnya berisi alamat **IP atau port** yang bisa diterima, ditolak, atau diblokir (port 23 : **Telnet**; port 79: **Finger**; port 119 : **USENET**,dsb)
- **KRIPTOGRAFI**

KEMBALI

Cryptosystem



KRIPTOLOGI

- **KRIPTOGRAFI** (penyandian), lawannya adalah **cryptanalysis** (pemecahan sandi), keduanya secara kolektif disebut **kriptologi**.
- Pesan di**enkripsi** (E) menjadi (**plaintext** = P) melalui transformasi yang diparameterisasi oleh **kunci** (K) menjadi **chipertext** (C) baru kemudian ditransmisikan. Pada sisi penerima chipertext di**dekripsikan** (D) kembali menjadi plaintext [TANE1997].

$$P = D_k(C)$$

$$C = E_k(P)$$

$$D_k(E_k(P)) = P$$

KEMBALI

Kunci

- Kunci umum (public-key) : untuk enkripsi
- Kunci privat (private-key) : untuk dekripsi
- Kunci rahasia (secret-key) : untuk enkripsi dan dekripsi

- Kunci simetris : menggunakan kunci rahasia untuk enkripsi dan dekripsi. Contoh : algoritma DES
- Kunci asimetris : menggunakan kunci privat dan kunci publik. Contoh : algoritma RSA.

KEMBALI

ENKRIPSI KONVENSIONAL

- disebut juga enkripsi simetrik / kunci tunggal, kerahasiaan terletak pada kunci. Kunci dua digit menurunkan 100 kemungkinan (00 sd.99).

- Kunci 56 bit menurunkan 2^{56} kemung-kinan. Bila CPU mampu melakukan pemecahan 1 enkripsi / μs akan diperlukan 1142 tahun untuk membongkarnya. Bila kemampuan CPU meningkat hingga 1 juta enkripsi / μs akan diperlukan waktu 10,01 jam [STAL2000]

KEMBALI

Metode enkripsi

- **Chiper Substitusi (substitusi mono-alfabetis):**

  karakter satu digantikan dengan karakter yang lain, misal a = z; b = y, dsb. Sehingga terdapat 26! kemungkinan. Pembongkaran chiper bisa dilakukan dengan mengidentifikasi karakter yang sering dipakai, dalam bahasa Inggris adalah karakter e, digram (th, in), trigram (the, ing), atau frasa yang berkaitan (account, financial,dsb.)

KEMBALI

Chiper Transposisi

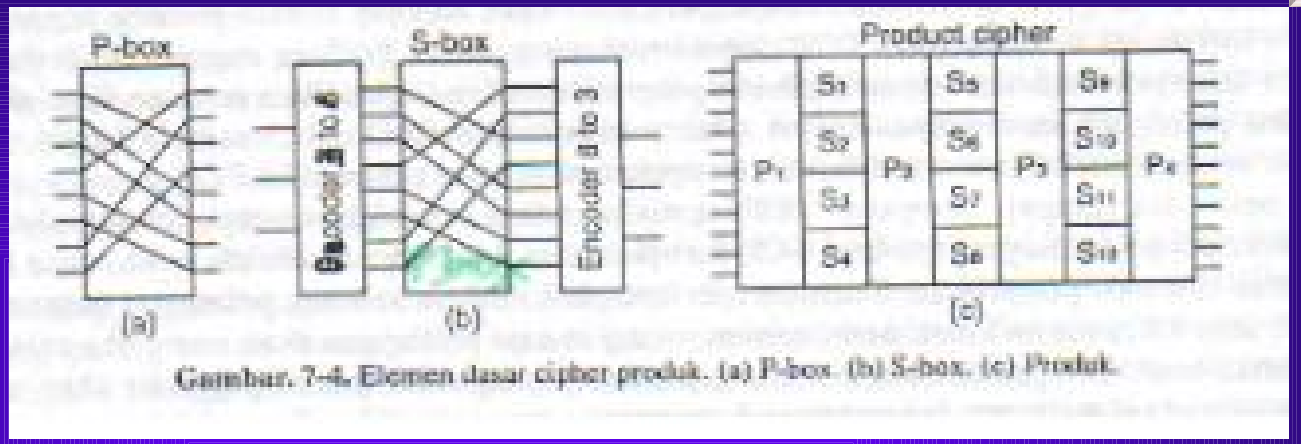
- Transposisi dilakukan berdasarkan kolom, misal :
- kunci : MEGABUCK
- Plaintext : pleasetransferonemilliondollars

M	E	G	A	B	U	C	K
7	4	5	1	2	8	3	6
p	l	e	a	s	e	t	r
a	n	s	f	e	r	o	n
e	m	i	l	l	i	o	n
d	o	l	l	a	r	s	

KEMBALI

- Chipertext :
  AFLLSELATOOSLNMOESILRNNPAEDERIR

Elemen Dasar Transformasi



Gambar. 7-4. Elemen dasar cipher produk. (a) P-box (b) S-box. (c) Produk.

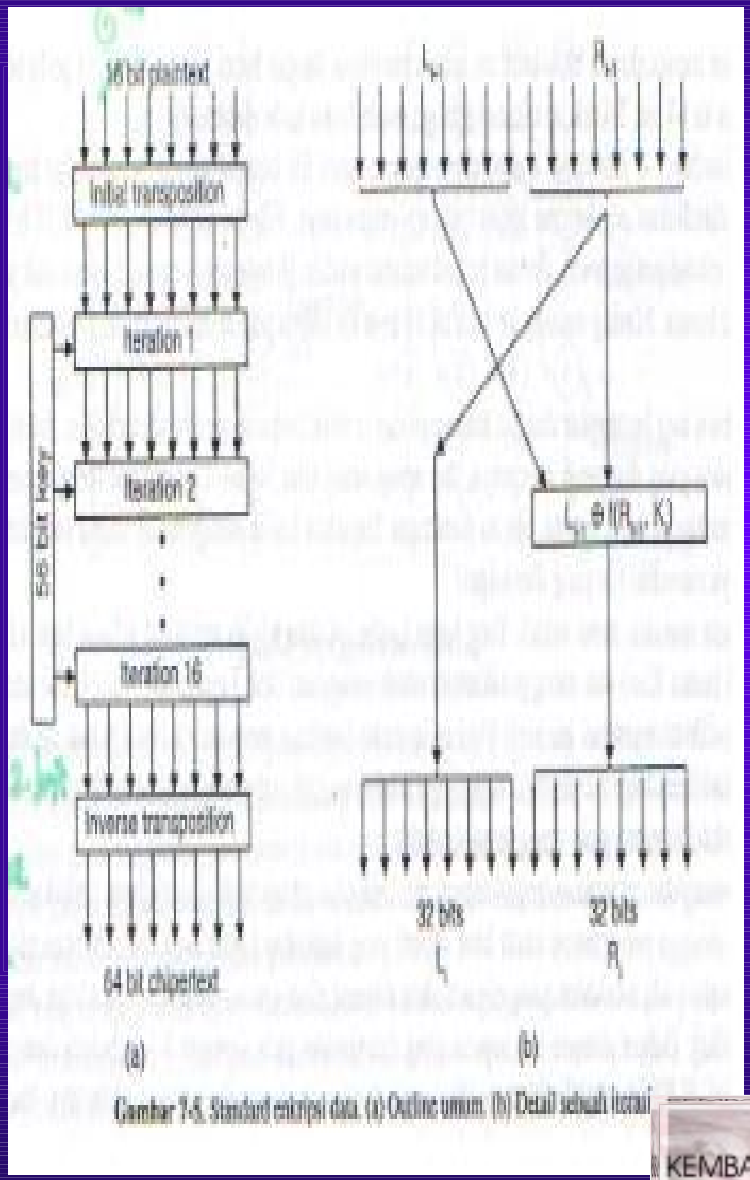
- **Permutasi (P-box)** : digunakan untuk memperoleh berbagai macam transposisi melalui rangkaian tertentu.
- **Substitusi (S-box)** : pada gb.substitusi diperoleh dengan memasukkan 3 bit plaintext ke decoder 3x8, masuk ke P-box, kemudian encoder 8x3 sehingga diperoleh 3 bit chipertext.
- **Chiper Product** : 12 bit plaintext diubah susunannya oleh P-box kemudian dipecah ke dalam 4 kelompok S-box, dst.hinggga dihasilkan 12 bit chipertext.
- **Fungsi bijektif** : pemetaan satu ke satu dari setiap elemen asal (X) ke satu elemen tujuan (Y)
- **Invers** : pembalikan, pemetaan elemen tujuan (Y) ke asal(X).

ALGORITMA ENKRIPSI DES

- Chiper product yang dikembangkan IBM pada tahun 1977 diadopsi oleh National Institute of Standards and Technology (NIST) milik AS sebagai standar nomor 46 dari Federal Information Processing Standard (FIPS PUB 46), yang disebut DES (Data Encryption Standard). Algoritmanya dikenal sebagai Data Encryption Algorithm (DEA).
- DES menggunakan kunci rahasia (simetris); disebut simetris jika untuk setiap asosiasi enkripsi – dekripsi dengan pasangan kunci (e,d) sedemikian sehingga e = d.
- DES mengenkripsi data plaintext yang terbagi dalam blok ukuran 64 bit dengan menggunakan kunci rahasia berukuran 56 bit dan menghasilkan chipertext berukuran 64 bit setelah dilakukan putaran sebanyak 19 kali dengan 16 kunci.

KEMBALI

OUTLINE ALGORITMA DES



Gambar 7-5. Standard enkripsi data. (a) Outline umum. (b) Detail sebuah iterasi

- **Awal Enkripsi** : permutasi blok plaintext pada Initial Permutation IP

  $x_0 = IP(x) = L_0R_0 \quad \{0,1\}^{64} \rightarrow \{0,1\}^{64}$

- **Inti Enkripsi** : L_0R_0 diproses 16 kali putaran menggunakan fungsi f_i sehingga diperoleh L_iR_i dengan berpedoman pada

  $L_i = R_{i-1}$

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

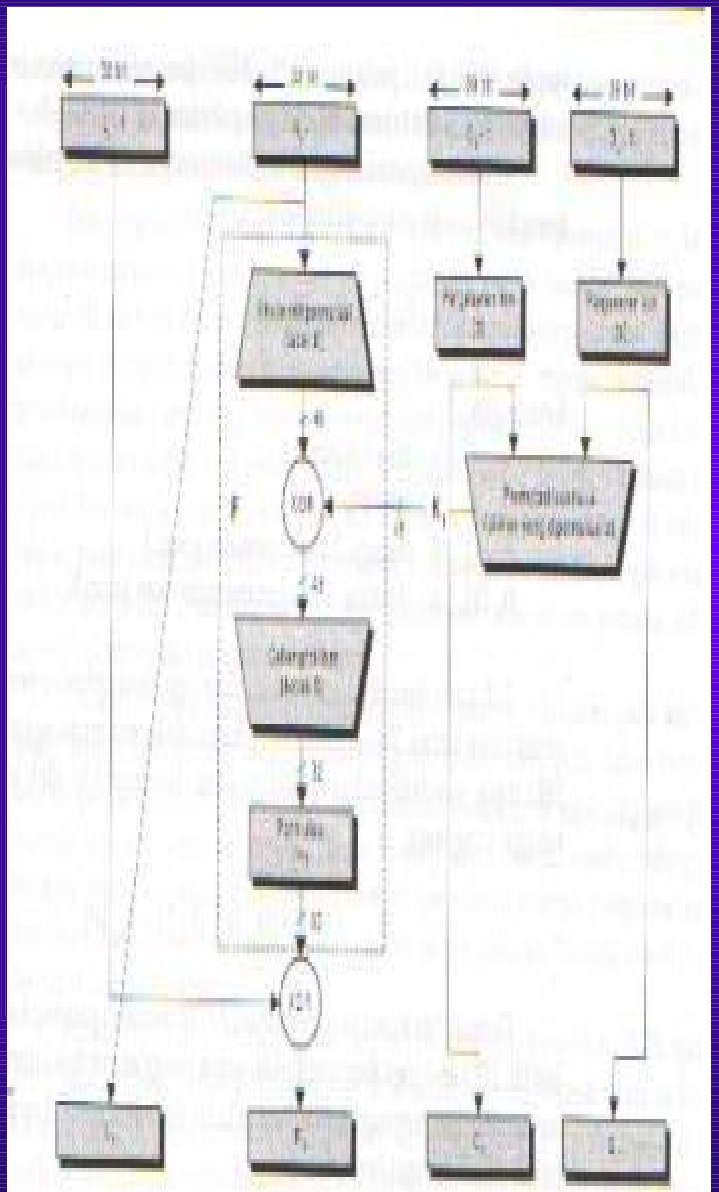
  Ket : $\oplus$ = XOR

  K_i = kunci ke-i (16 bh)

- **Akhir Enkripsi** : putaran terakhir yang berupa swap($L_{16}R_{16}$) sehingga diperoleh $L_{17}R_{17}$, kemudian di-invers permutation IP^{-1} sehing-ga diperoleh chipertext (y).

  $y = IP^{-1}(L_{16}R_{16}) \quad \{0,1\}^{64} \rightarrow \{0,1\}^{64}$

OUTLINE SATU PUTARAN DES



- Blok plaintext berukuran 64 bit, setelah permutasi awal terbagi dua menjadi bag kiri L_{i-1} (32bit) dan bag kanan R_{i-1} (32 bit).
- Kunci master 64 bit dipermutasi ke 56 bit : $\{0,1\}^{64} \rightarrow \{0,1\}^{56}$
- Hasil permutasi dipecah menjadi dua C_{i-1} (28 bit) dan D_{i-1} (28 bit).
- Untuk memperoleh kunci K_i : C_{i-1} dan D_{i-1} masing-masing digeser rotasi sebanyak (i) kali :

  $C_i = Leftshift_i\ (C_{i-1})$

  $D_i = Leftshift_i\ (D_{i-1})$

  dan dilakukan permutasi :

  $K_i = P(C_iD_i)\ \{0,1\}^{56} \rightarrow \{0,1\}^{48}$
- Fungsi $f(R_{i-1}K_i)$ berproses dengan diawali ekspansi R_{i-1} menjadi 48 bit.
- R_{i-1} dan kunci K_i diproses XOR:

  $B = R_{i-1} \oplus K_i$
- kemudian masuk ke S-box dan P-box $\rightarrow \{0,1\}^{32}$ lih. hlm.berikut

KEMBALI

ALGORITMA FUNGSI $f(R_{i-1}K_i)$

Terdiri dari 4 tahap :

1. Ekspansi $R_{i-1} : \{0,1\}^{32} \rightarrow \{0,1\}^{48}$
2. Pemrosesan 48 bit :

   $B = R_{i-1} \oplus K_i$

   pecah B menjadi delapan : $B_1\ B_2$ ….. B_8, masing-masing B_j sepanjang 6 bit.

   Contoh : B_1:100001; B_2:011011; B_3:011110; dst
3. Pemrosesan B_j ke dalam S-box

   $S_j : \{0,1\}^2 \times \{0,1\}^4 \rightarrow \{0,1\}^4$

   B_j terdiri dari b_1, b_2, ….. b_6

   Pemetaan bit b_1, b_2, ….. b_6 ke tabel.

   b_1 dan b_6 menunjukkan letak baris (0 ≤ row ≤3)

   b_2, b_3, b_4, b_5 menunjukkan letak kolom (0 ≤col≤15)

   Hasil berupa $C_j = C_1\ C_2$ ….. C_8, masing-masing C_j sepanja
   4 bit.

KEMBALI

Contoh S-box

	S_j															
No	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	15	11	8	3	10	6	12	5	9	0	7	14	4	13
1	4	14	2	13	1	10	6	12	11	9	5	3	8	0	15	7
2	8	13	6	2	11	15	12	9	7	3	10	5	0	4	1	14
3	2	4	9	1	7	5	11	3	14	10	0	6	13	15	12	8

- B_1:100001 → bit baris 11 = 3, bit kolom 0000 = 0 dari tabel diperoleh nilai 2 → C_1 = 0010
- B_2:011011; → bit baris 01 = 1, bit kolom 1101 = 13 dari tabel diperoleh nilai 0 → C_2 = 0000
- B_3:011110 → bit baris 00 = 0, bit kolom 1111 = 15 dari tabel diperoleh nilai 13 → C_3 = 1101
- B_4 dst.

4. Hasil C_j dipermutasikan : P(C) → $f(R_{i-1}K_i)$

KEMBALI

ALGORITMA DEKRIPSI DES

Input berupa chipertext 64 bit

Kunci 48 bit dipasang terbalik (K_{16}, K_{15}, …K_1)

Output berupa plaintext 64 bit.

Tahapan :

- **Awal dekripsi** : proses permutasi dari invers IP^{-1}

  $$y_0 = IP\,(IP^{-1}(R_{17}L_{17}) = R_{17}L_{17}$$

  kemudian swap($R_{17}L_{17}$), diperoleh $R_{16}L_{16}$

- **Inti Enkripsi** : $L_{16}R_{16}$ diproses 16 kali putaran menggunakan fungsi f_i dan kunci terbalik sehingga diperoleh L_iR_i dengan berpedoman pada

  $$R_i = L_{i-1}$$

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

- **Akhir Enkripsi** : putaran terakhir yang berupa L_0R_0 di-invers permutation IP^{-1} sehingga diperoleh plaintext (x).

  $$x = IP^{-1}(L_0R_0) \quad \{0,1)^{64} \rightarrow \{0,1)^{64}$$

KEMBALI

TANTANGAN DES

- Kecanggihan DES tinggal masalah **kecepatan proses** pembongkaran saja. Terdapat ide tentang **Lotre China** yang menanamkan chip DES murah dengan pembagian sektor kunci ke 1,2 milyard TV radio di China. Bila pemerintah ingin membongkar suatu pesan tinggal mem-broadcast pasangan plain dan chipertext, sehingga dalam waktu 1 menit akan ada yang memenangkan lotre tersebut.
- Kelemahan lain, setiap pasangan blok plaintext - chipertext akan memiliki **pola yang sama** sehingga dengan cara **copy – paste – overwrite** seseorang dapat berharap dia memperoleh keuntungan, misal noreg tabungan di bank. Ini dapat diatasi dengan memasukkan chipertext dalam proses enkripsi berikutnya yang disebut **Chiper Block Chaining :**
- $$C_i = E_k (C_{i-1} \oplus P_i)$$

Langkah dekripsinya :

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

KEMBALI

- DES akhirnya tumbang juga setelah Electronic Frountier Foundation (EFF) pada Juli 1998 mampu membuat **cracker machine** dengan biaya $250.000. Dengan mesin ini DES dapat dibongkar dalam waktu < 3 hari.
- Pengembangan : **Triple DEA** (ANSI X9.17 – 1995 atau FIPS PUB 46-3 1999). TDEA menggunakan tiga kunci (3x56 = 168 bit) dan tiga DEA dengan urutan enkrip – dekrip –enkrip :

$$C = E_{k3}\,(D_{k2}\,(E_{k1}(P)))$$

- Untuk mengurangi overhead K1 = K3

ALGORITMA RSA

- dikembangkan pertama kali tahun 1977 oleh Rivest, Shamir, Adleman (RSA).
- merupakan algoritma **kunci asimetris** dengan menggunakan kunci publik untuk enkripsi / dekripsi.
- menggunakan prinsip-prinsip teori bilangan dari bilangan bulat 0 dan n-1 untuk beberapa n.
- enkripsi / dekripsi berasal dari beberapa bentuk berikut ini :

$$C \equiv M^e \bmod n$$

$$M \equiv C^d \bmod n \equiv (M^e)^d \bmod n \equiv M^{ed} \bmod n$$

Ket : C = blok teks rahasia

M = blok teks asli

- Baik pengirim maupun penerima harus mengetahui nilai n dan e, dan hanya penerima saja yang mengetahui nilai d.

KEMBALI

Pembangkitan kunci

1. Pilih dua bilangan prima p dan q
2. Hitung n = p*q;
3. Hitung z = (p-1)(q-1)
4. Pilih e yang relatif prima terhadap z, artinya pembagi terbesar dari e dan z adalah 1. (1< e < z).
5. Hitung d ≡ e^{-1} mod z, dan d < z
6. Diperoleh **kunci umum** Ku = (e,n) dan **kunci khusus** Kk = (d,n)

Contoh :

8. Pilih p = 7 dan q = 17
9. Hitung n = 7 * 17 = 119
10. Hitung z = 6 * 16 = 96
11. Pilih e = 5, karena prima te
12. Hitung d ≡ 5^{-1} mod 96 → 5c

    d = 77, karena 5*77 = 385 =

6. Diperoleh Ku = (5, 119) ; Kk = (77,119)

a ≡ *b* (mod *n*) if *a* differs from *b* by an exact

multiple of *n*.

That is, *a* ≡ *b* + *Ln*, *L* being an integer

KEMBALI

ENKRIPSI RSA

- Teks asli : M < n
- Teks rahasia : $C \equiv M^e \pmod n$

Contoh : (lih. contoh sblmnya: n=119; e=5; d=77)

- Teks asli misal S (karakter no 19) $\rightarrow$ M = 19
- $C \equiv 19^5 \pmod{119} \equiv 66$

DEKRIPSI RSA

- Teks asli : C
- Teks rahasia : $M \equiv C^d \pmod n$

Contoh :

- C = 66
- $M \equiv 66^{77} \pmod{119} \equiv 19$

KEMBALI

Tantangan rsa

- Pembongkaran enkripsi RSA dapat dilakukan dengan brute force melalui berbagai kunci pribadi (Kk). Bila jumlah bit e dan d diperbesar, memang akan lebih mengamankan namun proses enkripsi /dekripsi akan lamban.

- Algoritma RSA dengan kunci umum 129 digit desimal (428 bit) tahun 1977 pernah diujikan oleh penciptanya dengan prediksi baru akan terbongkar setelah 40 quadrillion tahun. Namun tahun 1994 dibongkar oleh sebuah kelompok kerja setelah 8 bulan melibatkan 1600 komputer . Oleh sebab itu perlu digunakan kunci yang lebih besar, saat ini 1024 bit (300 digit desimal).

KEMBALI

authentisasi

- Authentisasi merupakan teknik untuk meyakinkan bahwa lawan komunikasi adalah entitas yang memang dikehendaki, bukan penyusup.
- Protokol yang bisa digunakan adalah Challenge Response. Kunci yang digunakan adalah kunci rahasia bersama yang disampaikan tidak melalui jaringan komputer.
- Lawan komunikasi di tantang untuk menggunakan kunci rahasia sebagai langkah verifikasi.

KEMBALI

Contoh authentisasi

1. Misal Alice mengirim pesan A ke Bob.
2. Bob tidak mengetahui asal pesan tersebut, kemudian Bob menjawab dengan mengambil bilangan random Rb sebagai plaintext dan mengirimkannya ke Alice.
3. Alice mengenkripsi pesan tersebut dengan kunci rahasia dan mengirimkannya sebagai chipertext K(Rb) ke Bob. Saat Bob menerima pesan ini dia akan mendekripsikannya dengan kunci rahasia sehingga dia tahu bahwa pesan tadi dari Alice, karena kuncinya sama.
4. Namun Alice masih belum yakin bahwa yang menerima pesannya adalah Bob, oleh sebab itu dia mengirim bilangan random Ra ke Bob.
5. Pada saat Bob menjawab dengan mengenkripsi Ra dengan kunci rahasianya menjadi chipertext K(Ra) barulah Alice yakin bahwa lawan komunikasinya adalah Bob.

Langkah tersebut di atas dapat dipersingkat dengan cara Alice mengirimkan pesan A dan Ra yang akan dijawab oleh Bob dengan Rb dan K(Ra). Selanjutnya Alice akan membalasnya dengan K(Rb).

KEMBALI