Jurnal Ilmiah KURSOR
Menuju Solusi Teknologi Informasi
Vol. 3, No. 2, Juli 2007 ISSN 0216 - 0544

PERANAN KRIPTOGRAFI DALAM KEAMANAN DATA PADA JARINGAN KOMPUTER

Sigit Susanto Putro
Sigitida_79@yahoo.com

Jurusan Teknik Informatika
Universitas Trunojoyo

ABSTRAK

Dengan adanya teknologi jaringan komputer memungkinkan pengiriman data jarak jauh yang relatif cepat dan murah. Namun pada pengiriman data menggunakan jaringan komputer baik melalui gelombang radio ataupun melalui media yang lain, sangat memungkinkan pihak lain dapat menyadap dan mengubah data yang dikirimkan. Oleh sebab itu dalam teknologi jaringan komputer telah dan sedang dikembangkan cara-cara untuk menangkal berbagai bentuk serangan semacam itu. Salah satu caranya adalah dengan menggunakan kriptografi yang menggunakan transformasi data sehingga data yang dihasilakan tidak dapat dimengerti oleh pihak ketiga. Ada empat tujuan dasar dari kriptografi yang juga merupakan aspek dari keamanan informasi yaitu untuk kerahasiaan, Integritas, Autentikasi dan non-repudiasi.

Kata kunci *:Kriptografi,Transformasi,Kerahasiaan,Integritas,Autentikasi,Non-Repudiasi*

1. PENDAHULUAN

Secara umum kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan berita (bruce Schneier - *Applied Cryptography)*. Selain itu pengertian kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data (A. Menezes, P. van Oorschot and S. Vanstone - Handbook of Applied Cryptography). Namun tidak semua aspek keamanan informasi ditangani oleh kriptografi.

Ada empat aspek keamanan informasi yang merupakan tujuan dasar dari kriptografi. Kempat aspek tersebut adalah :

- Kerahasiaan data, adalah layanan yang bertujuan memberikan kerahasiaan pesan dan menyimpan data dengan menyembunyikan informasi lewat teknik-teknik enkripsi kepada siapapun kecuali pemegang otoritas atau kata kunci untuk membuka informasi yang telah di enkripsi tersebut.
- Integritas data, adalah layanan untuk memberikan jaminan bahwa pesan tidak akan mengalami perubahan dari saat dibuat sampai dibuka.
- Autentikasi data, adalah layanan untuk identifikasi/pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Layanan ini juga berfungsi untuk menguji identitas seseorang apabila ia akan memasuki sistem tersebut.
- Keabsahan data atau non-repudiation, adalah layanan untuk membuktikan bahwa suatu data ataupun dokumen datang dari seseorang apabila yang bersangkutan menyangkal memiliki data ataupun dokumen tersebut.

Selain itu kriptografi juga bisa diterapkan dalam beberapa hal, diantaranya *Digital IDs,Digital signatures* serta *Secure chanels*.

2. ALGORITMA KRIPTOGRAFI

Algoritma kriptografi adalah suatu algoritma yang berfungsi untuk melakukan konfusi data sehingga data yang dikirimkan tidak dapat diartikan secara langsung tanpa menggunakan algoritma deskripsinya. Selain itu algoritma kriptografi juga bisa melakukan difusi sehingga bisa menghilangkan karakteristik dari data tersebut, sehingga dapat digunakan untuk mengamankan informasi.

Pada implementasinya sebuah algoritma kriptografi harus memperhatikan kualitas layanan dari keseluruhan sistem dimana algoritma tersebut diimplementasikan. Algoritma kriptografi yang handal adalah algoritma kriptografi yang kekuatannya terletak pada kunci, bukan pada kerahasiaan algoritma itu sendiri.

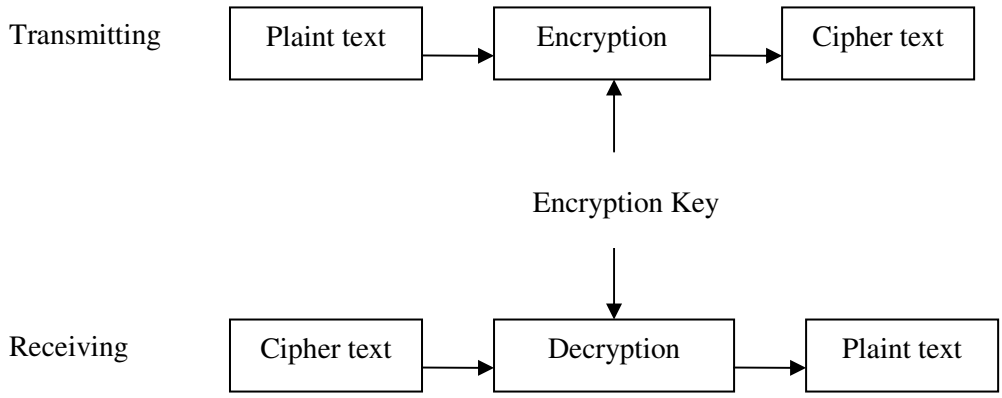
Kriptografi terdiri dari dua proses yaitu proses enkripsi dan dekripsi. Kedua proses tersebut berfungsi untuk mentransformasikan data asli atau lebih dikenal dengan istilah *plaintext* dan data sandi yang dikenal dengan *ciphertext*. Dimana secara matematis dapat diterangkan sebagai berikut. Apabila plaintext kita notasikan dengan P, ciphertext kita notasikan dengan C, enkripsi di notasikan dengan E dan dekripsi dinotasikan dengan D maka akan diperoleh persamaan sebagai berikut:

- E(P) = C (proses enkripsi)
- D(C) = P atau D(E(P)) = P (proses dekripsi)

Secara umum algoritma kriptografi dibedakan menjadi tiga macam yakni algoritma kriptografi kunci rahasia atau bisa juga disebut algoritma kunci simetris, algoritma kriptografi kunci publik atau algoritma kunci asimetris dan algoritma hash.

2.1. Algoritma Kunci Simetris

Disebut algoritma kunci simetris karena dalam algoritma ini untuk melakukan proses enkripsi maupun dekripsi menggunakan kunci yang sama.Berdasarkan jumlah data perproses serta alur pengolahan datanya algoritma ini dibedakan menjadi dua kelas yakni block-cipher dan stream-cepher. Karakteristik dari algoritma ini bisa dilihat seperti gambar 2.1 dibawah ini.

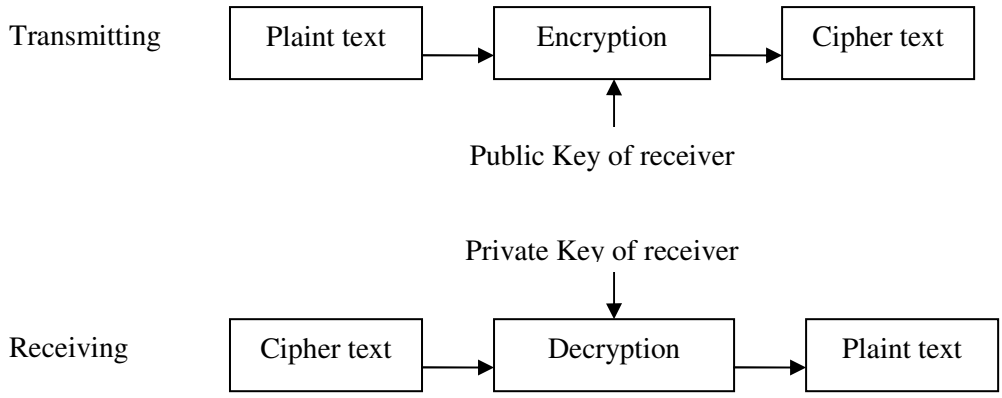


Gambar 2.1. Algoritma kriptografi kunci simetris

Untuk menggunakan algoritma kunci simetris ini pihak pengirim atau transmitter serta pihak penerima atau receiver sebelumnya harus sama-sama tahu kunci enkripsinya. Contoh dari algoritma yang menggunakan kunci simetris adalah DES, Blowfish, Twofish, MARS, IDEA, 3DES serta AES.

2.2. Algoritma Kunci Asimetris

Dalam algoritma kunci asimetris ini untuk proses enkripsi dan dekripsi menggunakan kunci yang berbeda. Algoritma ini juga dikenal dengan algoritma kunci publik karena kunci enkripsinya dibuat untuk diketahui secara umum atau biasa disebut dengan *public – key*. Akan tetapi untuk kunci dekripsinya hanya bisa diketahui oleh yang berhak akan data tersebut, oleh karena itu kunci dekripsinya biasa disebut dengan *private – key*. Karakteristik dari algoritma ini bisa dilihat pada gambar 2.2.



Gambar 2.2. Algoritma kriptografi kunci asimetris

Algoritma ini bisa dianalogikan seperti contoh berikut ini. Apabila Ahmad dan Bejo hendak bertukar informasi, maka yang harus dilakukan adalah:

1. Ahmad dan Bejo masing-masing membuat 2 buah kunci
    - Ahmad membuat dua buah kunci, kunci-publik $K_{publik[Ahmad]}$ dan kunci-privat $K_{privat[Ahmad]}$
    - Bejo membuat dua buah kunci, kunci-publik $K_{publik[Bejo]}$ dan kunci-privat $K_{privat[Bejo]}$
2. Mereka berkomunikasi dengan cara:
    - Ahmad dan Bejo saling bertukar kunci-publik. Bejo mendapatkan $K_{publik[Ahmad]}$ dari Ahmad, dan Ahmad mendapatkan $K_{publik[Bejo]}$ dari Bejo.
    - Ahmad mengenkripsi *plaintex*P ke Bejo dengan fungsi $C = E(P, K_{publik[Bejo]})$
    - Ahmad mengirim ciphertext C ke Bejo
    - Bejo menerima *ciphertext*C dari Ahmad dan membuka *plaintext* dengan fungsi $P = D(C, K_{privat[Bejo]})$

Hal yang sama terjadi apabila Bejo hendak mengirimkan pesan ke Ahmad

1. Bejo mengenkripsi *plaintext*P ke Ahmad dengan fungsi $C = E(P, K_{publik[Ahmad]})$
2. Ahmad menerima *ciphertext* C dari Bejo dan membuka *plaintext* dengan fungsi $P = D(C, K_{privat[Ahmad]})$

Contoh dari algoritma yang menggunakan kunci asimetris adalah Knapsack, RSA serta Diffie-Hellman.

2.3. Algoritma Hash

Algoritma ini mempunyai beberapa sifat keamanan tambahan sehingga dapat dipakai untuk tujuan keamanan data. Umumnya digunakan untuk keperluan autentikasi dan integritas data. Fungsi hash adalah fungsi yang secara efisien mengubah string input dengan panjang berhingga menjadi string output dengan panjang tetap yang disebut nilai hash. Fungsi hash diperlukan dalam bagian konfigurasi sistem, yang berguna untuk memudahkan pengecekan terhadap kelebihan data.

Fungsi hash mempunyai sifat-sifat sebagai berikut:

- *Preimage resistant*: bila diketahui nilai hash *h* maka sulit (secara komputasi tidak layak) untuk mendapatkan *m* dimana *h* = hash(*m*).
- *Second preimage resistant*: bila diketahui input m_1 maka sulit mencari input m_2 (tidak sama dengan m_1) yang menyebabkan hash(m_1) = hash(m_2).
- *Collision-resistant*: sulit mencari dua input berbeda m_1 dan m_2 yang menyebabkan hash(m_1) = hash(m_2)

Contoh dari algoritma hash adalah MD4, MD5, SHA-0, SHA-1, SHA-256, SHA-512.

3. ALGORITMA DES

DES (*Data Encryption Standart*) adalah nama dari sebuah algoritma untuk melakukan proses enkripsi data yang dikeluarkan oleh *Federal Information Processing Standart* (FIPS) 46-1 Amerika Serikat.

DES memiliki blok kunci 64 bit, tetapi yang digunakan dalam proses eksekusi hanya 56 bit. Pada awalnya algoritma ini dirancang untuk implementasi secara hardware. DES disertifikasi ulang setiap 5 tahun oleh NIST Amerika. DES terakhir kali disertifikasi pada tahun 1993, sampai sekarang DES tidak pernah disertifikasi lagi karena disinyalir banyak kelemahannya dan adanya pengembangan algoritma baru yaitu AES (*Advanced Encryption Standart*).

Secara umum algoritma DES terbagi menjadi 3 kelompok, dimana kelompok yang satu dengan yang lainnya saling berinteraksi dan terkait antara satu dengan yang lainnya. Kelompok tersebut adalah Pemrosesan kunci, Enkripsi data 64 bit dan Dekripsi data 64 bit.

3.1 Pemrosesan Kunci

Algoritma dari pemrosesan kunci adalah sebagai berikut:

- User memasukan kunci sebesar 64 bit atau 8 karakter, dimana setiap bit dari kunci tersebut nantinya akan digunakan sebagai bit paritas.
- Setelah itu dilakukan permutasi terhadap kunci tersebut, tetapi sebelumnya perlu diadakan penjadwalan kunci rahasia. Hal ini dilakukan untuk menyusun 16 buah kunci yang akan dimasukkan pada setiap iterasi DES, baik pada proses enkripsi maupun dekripsi.
- Setelah langkah kedua selesai, dilakukan permutasi. Permutasi dilakukan pada kunci 64 bit yang tadi. Pada tahap ini, bit-bit paritas tidak dilibatkan, sehingga bit kunci berkurang menjadi 56 bit. Bit 1 pada kunci ke 56 merupakan bit 57 pada kunci awalnya, bit2 adalah bit 49, dan seterusnya hingga bit 56 adalah bit 4 kunci 64. posisi bit hasil permutasi pada langkah awal ini diberi nama *Permuted Choice* 1 (PC-1). Hasil dari proses ini adalah sebagai berikut.

Permuted Choice 1 (PC-1)

57	49	41	33	25	17	09
01	58	50	42	34	26	18
10	02	59	51	43	35	27
19	11	03	60	52	44	36
63	55	47	39	31	23	15
07	62	54	46	38	30	22
14	06	61	53	45	37	29
21	13	05	28	20	12	04

- Langkah berikutnya PC-1 dibagi menjadi dua bagian, 28 bit pertama disebut C(0) dan 28 bit terakhir disebut D(0).
- Dari C(0) dan D(0) dihitung sub-sub kunci untuk setiap iterasi, dimulai dengan j=1.
- Untuk setiap iterasi, j rotasi ke kri 1 kali atau sebanya 2 kali untu C(j-1) dan D(j-1). Dari hasil rotasi ini didapatkan C(j) dan D(j). Langkah – langkah dari setiap rotasi bisa dilihat pada tabel 3.1.

Iterasi Ke	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Jumlah Step	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Tabel 3.1. Step rotasi yang dilakukan pada setiap iterasi

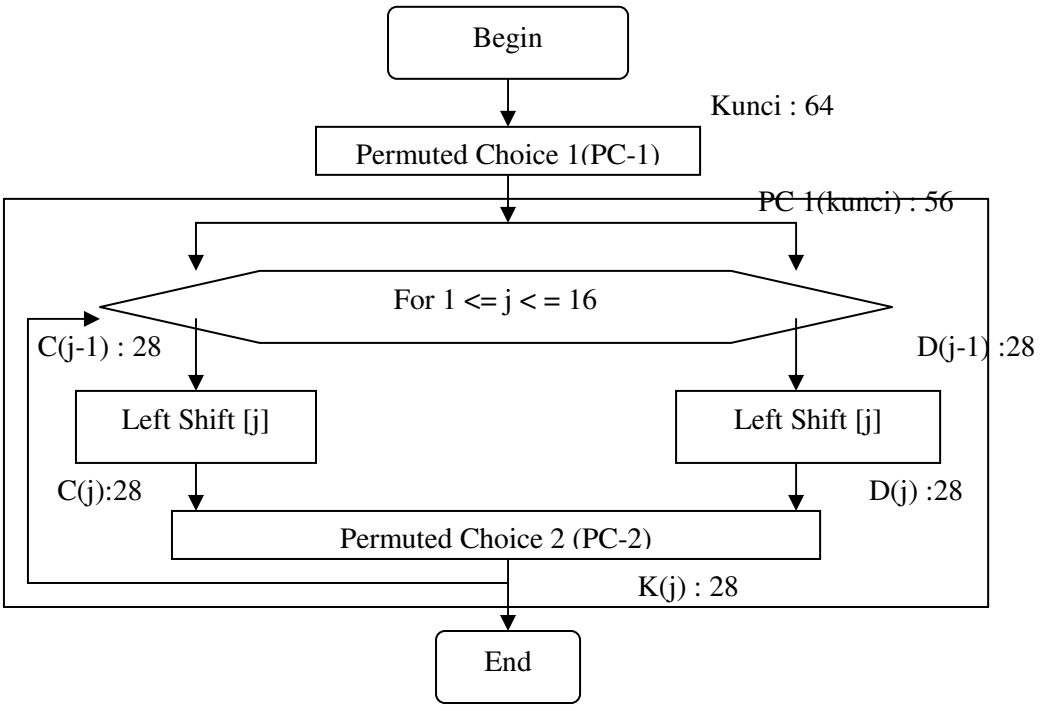
- Untuk setiap hasil C(j) dan D(j), kunci pada iterasi ke j dilakukan dengan cara melakukan permutasi kembali pada C(j) dan D(j). Permutasi itu dikenal dengan *permuted choice* 2(PC-2). Hasilnya adalah sebagai berikut.

Permuted Choice 2 (PC-2)

14 17 11 24 01 05
03 28 15 06 21 10
23 19 12 04 26 08
16 07 27 20 13 02
41 52 31 37 47 55
30 40 51 45 33 48
44 49 39 56 34 53
46 42 50 36 29 32

- Iterasi dilakukan teru menerus hingga 16 unci behasil disusun.

Adapun *flowchart* dari proses diatas dapat dilihat pada gambar 3.1



Gambar 3.1. Flowchart pemrosesan kunci

3.2 Enkripsi Data 64 Bit

Algoritma dari enkripsi data 64 bit adalah sebagai berikut:

- Ambil blok data sebesar 64 bit. Apabila blok data yang diambil kurang dari 64 bit maka lakukan penambahan karakter sebanyak kekurangannya.
- Bentuklah permutasi awal (Initial Permutation atau IP) pada blok data 64 bit tersebut, dengan memperhatikan permutasi sebagai berikut.

  Initial Permutation

  58 50 42 34 26 18 10 02

  60 52 44 36 28 20 12 04

  62 54 46 38 30 22 14 06

  64 56 48 40 32 24 16 08

  57 49 41 33 25 17 09 01

  59 51 43 35 27 19 11 03

  61 53 45 37 29 21 13 05

  63 55 47 39 31 23 15 07

- Kemudian blok data tersebut dibagi menjadi 2 bagian sebesar 32 bit. 32 bit pertama disebut L[0] dan 32 bit kedua disebut R[0].
- Ke 16 sub kunci dioperasikan dengan blok data, dimulai dari j=1 dan terbagi menjadi beberapa cara seperti berikut:

  1. R[j-1] dikembangkan menjadi 48 bit menurut fungsi pemilihan expansi berikut

     Expansion (E)

     32 01 02 03 04 05 04 05 06 07 08 09

     08 09 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 01

  2. Setelah itu E(R[j-1]) di XOR kan dengan K[j].
  3. Kemudian hasilnya dipecah menjadi delapan blok 6 bit. Kelompok bit 1 – 6 disebut B[1], kelompok bit 7 – 12 disebut B[2], dan seterusnya sampai kelompok bit 43 – 48 disebut B[8].
  4. Setelah itu jumlah bit dikurangi dengan penukaran nilai-nilai yang ada dalam tabel S untuk setiap B[j]. Dimulai dengan j=1, setiap nilai dalam tabel S memiliki 4 bit.Langkah-langkah yang ada pada tahap ini adalah sebagai berikut. Ambil bit ke 1 dan ke 6 dari B[j] menjadi nilai 2 bit, misalkan m,yang menunjukan baris dalam tabel S[j]. Kemudian ambil bit ke 2 hingga ke 5 dari B[j] sebagai nilai 4 bit, misalkan n, yang menunjukan kolom dari S[j]. Hasil dari proses ini adalah S[j][m][n] untuk setiap B[j]. Dengan demikian diperlukan 8 kali iterasi. Hasil dari proses ini sering disebut dengan *Subtitution Box*. Sehingga dari proses ini akan menghasilkan 8 Subtitution Box seperti yang di tunjukkan tabel 3.2 sampai tabel 3.9 berikut ini.

	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

Tabel 3.2. Subtitution Box 1 (S[1])

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

Tabel 3.3. Subtitution Box 2 (S[2])

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

Tabel 3.4. Subtitution Box 3 (S[3])

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

Tabel 3.5. Subtitution Box 4 (S[4])

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	2	12	14	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

Tabel 3.6. Subtitution Box 5 (S[5])

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
3	4	3	2	12	9	15	10	11	14	1	7	6	0	8	13	12

Tabel 3.7. Subtitution Box 6 (S[6])

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

Tabel 3.8. Subtitution Box 7 (S[7])

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Tabel 3.9. Subtitution Box 8 (S[8])

5. Permutasi dilakukan kembali pada kombinasi hasil subtitusi diatas S[1][m1][n1] samapai S[8][m8][n8] dengan memperhatikan keterangan berikut ini.

   Permutation P

   16 07 20 21 29 12 28 17

   01 15 23 26 05 18 31 10

   02 08 24 14 32 27 03 09

   19 13 30 06 22 11 04 25

6. Setelah itu hasilnya di XOR kan dengan L[j-1], menjadi R[j] dengan rumus sebagai berikut : R[i]=L[i-1] XOR P(S[1](B[1])...S[8](B[8])

   Dimana B[j] merupakan blok 6 bit hasil kombinasi dari R(R[i-1]) XOR K[i], sehingga fungsi tersebut bisa ditulis sebagai berikut : R[i]=L[i-1] XOR f(R[i-1], K[i])

7. L[i] = R[i-1]

8. Semua proses diatas tersebut diulang hingga K[16]

- Final permutation dilakukan kembali dengan tabel permutasi yang merupakan invers dari permutasi awal,seperti berikut ini.

  Final Permutation

  16 07 20 21 29 12 28 17

  40 08 48 16 56 24 64 32

  39 07 47 15 55 23 63 31

  38 06 46 14 53 22 62 30

  37 05 45 13 53 21 61 29

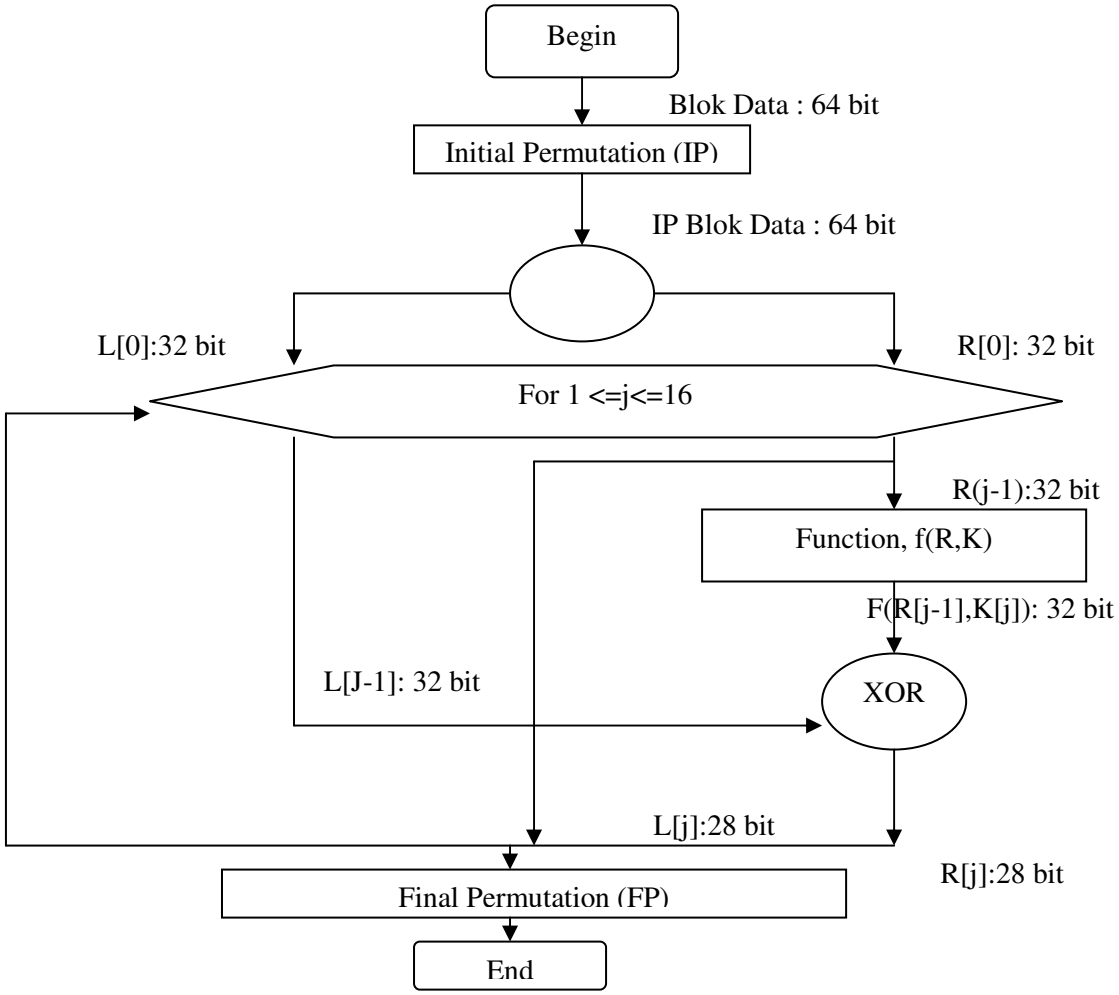
  36 04 44 12 52 20 60 28

  35 03 43 11 51 19 59 27

  34 02 42 10 50 18 58 26

  33 01 41 09 49 17 57 25

Flowchart enkripsi data 64 bit bisa dilihat pada gambar 3.2.



Gambar 3.2. Flowchart Enkripsi data 64 bit

3.3. Dekripsi Data 64 Bit

Algoritma dekripsi data 64 bit adalah sebagai berikut:

- Ambil blok data sebesar 64 bit. Apabila blok data yang diambil kurang dari 64 bit maka lakukan penambahan karakter sebanyak kekurangannya.
- Bentuklah permutasi awal (Initial Permutation atau IP) pada blok data 64 bit tersebut, dengan memperhatikan permutasi sebagai berikut.

  Initial Permutation

  58 50 42 34 26 18 10 02
  60 52 44 36 28 20 12 04
  62 54 46 38 30 22 14 06
  64 56 48 40 32 24 16 08
  57 49 41 33 25 17 09 01
  59 51 43 35 27 19 11 03
  61 53 45 37 29 21 13 05
  63 55 47 39 31 23 15 07

- Kemudian blok data tersebut dibagi menjadi 2 bagian sebesar 32 bit. 32 bit pertama disebut L[0] dan 32 bit kedua disebut R[0].

- Ke 16 sub kunci dioperasikan dengan blok data, dimulai dari j=1 dan terbagi menjadi beberapa cara seperti berikut:

    1. R[j-1] dikembangkan menjadi 48 bit menurut fungsi pemilihan expansi berikut

    Expansion (E)

32	01	02	03	04	05	04	05	06	07	08	09
08	09	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	01

    2. Setelah itu E(R[j-1]) di XOR kan dengan K[j].
    3. Kemudian hasilnya dipecah menjadi delapan blok 6 bit. Kelompok bit 1 – 6 disebut B[1], kelompok bit 7 – 12 disebut B[2], dan seterusnya sampai kelompok bit 43 – 48 disebut B[8].
    4. Setelah itu jumlah bit dikurangi dengan penukaran nilai-nilai yang ada dalam tabel S untuk setiap B[j]. Dimulai dengan j=1, setiap nilai dalam tabel S memiliki 4 bit.Langkah-langkah yang ada pada tahap ini adalah sebagai berikut. Ambil bit ke 1 dan ke 6 dari B[j] menjadi nilai 2 bit, misalkan m,yang menunjukan baris dalam tabel S[j]. Kemudian ambil bit ke 2 hingga ke 5 dari B[j] sebagai nilai 4 bit, misalkan n, yang menunjukan kolom dari S[j]. Hasil dari proses ini adalah S[j][m][n] untuk setiap B[j]. Dengan demikian diperlukan 8 kali iterasi. Hasil dari proses ini sering disebut dengan *Subtitution Box*.
    5. Permutasi dilakukan kembali pada kombinasi hasil subtitusi diatas S[1][m1][n1] samapai S[8][m8][n8] dengan memperhatikan keterangan berikut ini.

    Permutation P

16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
02	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

    6. Setelah itu hasilnya di XOR kan dengan R[i], menjadi L[i-1], fungsi tersebut bisa ditulis sebagai berikut : L[i-1]=R[i] XOR f(L[i], K[i])
    7. L[i] = R[i-1]
    8. Semua proses diatas tersebut diulang hingga K[1]

- *Final permutation* dilakukan kembali dengan tabel permutasi yang merupakan invers dari permutasi awal,seperti berikut ini.

Final Permutation

40	08	48	16	56	24	64	32
39	07	47	15	55	23	63	31
38	06	46	14	54	22	62	30
37	05	45	13	53	21	61	29
36	04	44	12	52	20	60	28
35	03	43	11	51	19	59	27
34	02	42	10	50	18	58	26
33	01	41	09	49	17	57	25

4. KESIMPULAN

Kriptografi adalah salah satu solusi untuk keamanan data pada jaringan komputer. Algoritma dari kriptografi banyak macamnya, antara lain adalah algoritma kriptografi kunci rahasia yang juga sering disebut dengan algoritma kunci simetris,algoritma kriptografi kunci publik atau lebih dikenal dengan algoritma kunci asimetris serta algoritma hash dan semuanya itu bisa dipelajari dengan bebas. Walaupun demikian kita tidak perlu kawatir untuk menggunakannya karena inti dari kriptografi yang handal terletak pada kuncinya bukan pada kerahasiaan dari algoritmanya. Walaupun secara umum algoritma tersebut bisa dipelajari asalkan kunci dari algoritma yang kita gunakan tidak diketahui maka keamanan data tetap terjamin.

Secara umum Algoritma Kriptografi terdiri dari tiga bagian yang saling terkait satu dengan lainnya, ketiga bagian itu adalah penyusunan kunci, Enkripsi dan dekripsi. Kriptografi juga bisa diterapkan pada teknologi mobile, baik itu untuk keamanan sms, mms ataupun pengiriman pesan berupa gambar, suara maupun film. Hal ini dikarenakan kriptografi tidak hanya bisa menyandikan file text saja, namun juga bisa digunakan untuk menyandikan file tipe yang lain seperti gambar, suara, animasi bergerak ataupun film.

5. DAFTAR PUSTAKA

1. Andri Kristanto, KEAMANAN DATA PADA JARINGAN KOMPUTER, GAVA MEDIA, 2003
2. http://id.wikipedia.org/wiki/AES, 15 agustus 2007
3. http://id.wikipedia.org/wiki/Kriptografi, 15 agustus 2007
4. http://id.wikipedia.org/wiki/RSA, 15 agustus 2007
5. William Stallings, Cryptography and Network Security, Principles and Practices, Fourth Edition, 2006