ANALISIS KELEMAHAN ALGORITMA CIPHER BLOK DES DAN KEKUATAN TRIPLE DES SEBAGAI VARIAN PENGGANTI DES

Bemby Bantara Narendra – NIM : 13503105

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if13105@students.if.itb.ac.id

Abstrak

DES atau Data Encryption Standard adalah algoritma cipher blok yang populer hampir selama dua dasawarsa karena menjadi standard algoritma enkripsi kunci simetri. Namun saat ini standard ini telah tergantikan oleh algoritma lain karena DES dianggap tidak aman lagi. Hal ini disebabkan karena pada tahun 1998, ditemukan kunci DES dengan menggunakan pencarian brute force selama 22 jam. Padahal awalnya diasumsikan bahwa pencarian brute force minimal membutuhkan waktu 1142 tahun untuk menemukan kunci yang benar.

Makalah ini akan memaparkan studi literatur mengenai serangan exhaustive key search pada algoritma chiper blok DES yang menyebabkan pemecahan algoritma ini. Studi literatur yang dilakukan meliputi pemecahan publik pertama dari pesan yang dienkripsi dengan DES dengan menggunakan pencarian brute force. Studi literatur juga memperlihatkan bagaimana kekuatan yang diperlukan untuk memecahkan algoritma ini dapat diperoleh oleh individu dan organisasi yang sangat kecil dengan dana yang minim ataupun tanpa dana sama sekali pun.

Selain itu makalah ini juga melakukan analisis terhadap algoritma triple DES sebagai varian dari DES yang lebih kuat dan mampu melindungi informasi dengan baik. Triple DES menggunakan algoritma DES sebagai algoritma utama. Triple DES dikembangkan untuk mengatasi kelemahan ukuran kunci yang digunakan pada proses enkripsi dan dekripsi DES sehingga teknik kriptografi ini lebih tahan terhadap exhaustive key search yang dilakukan oleh kriptanalis. Analisis yang dilakukan dalam makalah ini meliputi analisis algoritma dan analisis keamanan algoritma triple DES.

Kata Kunci : kriptografi, algoritma, Data Encryption Standard (DES), Triple DES, exhaustive key search, brute force

1. Pendahuluan

Kriptografi, secara umum adalah ilmu dan seni untuk menjaga kerahasiaan berita. Ada empat tujuan mendasar dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi yaitu:

1. Kerahasiaan, adalah layanan yang digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka/ mengupas informasi yang telah disandi.

2. Integritas data, adalah berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak- pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya.

3. Autentikasi, adalah berhubungan dengan identifikasi/ pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan melalui kanal harus diautentikasi keaslian, isi datanya, waktu pengiriman, dan lain- lain.

4. Non- repudiasi., atau nirpenyangkalan adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/ terciptanya suatu informasi oleh yang mengirimkan/ membuat.

Untuk mencapai tujuan kriptografi ini diciptakanlah algoritma- algoritma sandi. Algoritma tersebut harus memiliki kekuatan untuk melakukan:

1. konfusi/ pembingungan (confusion), dari teks terang sehingga sulit untuk direkonstruksikan secara langsung tanpa menggunakan algoritma dekripsinya

2. difusi/ peleburan (difusion), dari teks terang sehingga karakteristik dari teks terang tersebut hilang.

sehingga dapat digunakan untuk mengamankan informasi. Algoritma sandi yang handal adalah algoritma sandi yang kekuatannya terletak pada kunci, bukan pada kerahasiaan algoritma itu sendiri.

Secara umum berdasarkan kesamaan kuncinya, algoritma sandi dibedakan menjadi :

1. kunci- simetris/ symetric- key, sering disebut juga algoritma sandi konvensional karena umumnya diterapkan pada algoritma sandi klasik

2. kunci- asimetris/ asymetric- key

Skema algoritma sandi akan disebut kunci-simetris apabila untuk setiap proses enkripsi maupun dekripsi data secara keseluruhan digunakan kunci yang sama. Skema ini berdasarkan jumlah data per proses dan alur pengolahan data di dalamnya dibedakan menjadi dua kelas, yaitu block- cipher dan stream- cipher.

Contoh algoritma yang menggunakan kunci-simetris adalah DES atau Data Encryption Standard.

2. DES (Data Encryption Standard)

2.1 Sejarah DES

Algoritma DES dikembangkan di IBM dibawah kepemimpinan W. L. Tuchman pada tahun 1972. Algoritma ini didasarkan pada algoritma Lucifer yang dibuat oleh Horst Feistel.

Algoritma ini telah disetujui oleh National Bureau of Standard (NBS) setelah penilaian kekuatannya oleh National Security Agency (NSA) Amerika Serikat.

2.2 Tinjauan Umum

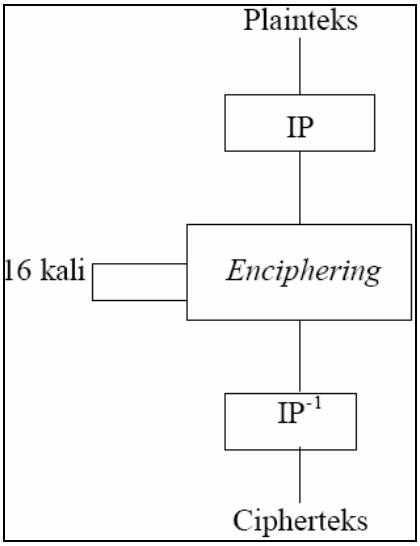
DES termasuk ke dalam sistem kriptografi simetri dan tergolong jenis cipher blok. DES beroperasi pada ukuran blok 64 bit. DES mengenkripsikan 64 bit plainteks menjadi 64 bit cipherteks dengan menggunakan 56 bit kunci internal (internal key) atau upa- kunci (subkey). Kunci internal dibangkitkan dari kunci eksternal (external key) yang panjangnya 64 bit.

Skema global dari algoritma DES adalah sebagai berikut (lihat Gambar 2.1):

1. Blok plainteks dipermutasi dengan matriks permutasi awal (initial permutation atau IP).

2. Hasil permutasi awal kemudian di-enciphering sebanyak 16 kali (16 putaran). Setiap putaran menggunakan kunci internal yang berbeda.

3. Hasil enciphering kemudian dipermutasi dengan matriks permutasi balikan (invers initial permutation atau IP- 1) menjadi blok cipherteks.



Gambar 2.1 Skema Global Algoritma DES

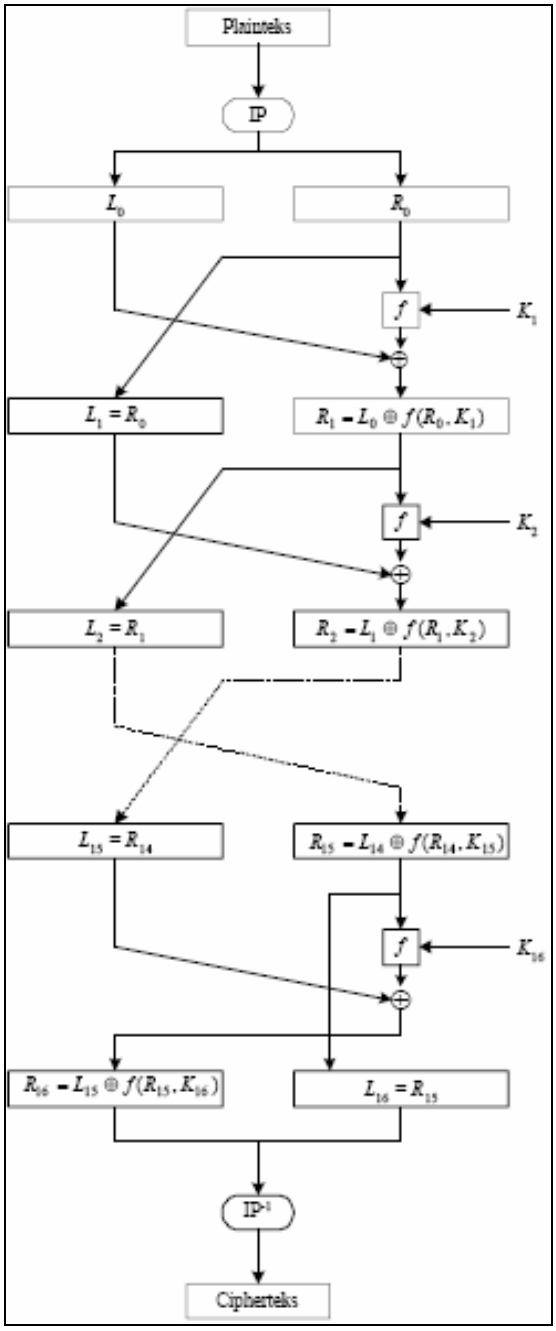
Di dalam proses enciphering, blok plainteks terbagi menjadi dua bagian, kiri (L) dan kanan (R), yang masing- masing panjangnya 32 bit. Kedua bagian ini masuk ke dalam 16 putaran DES.

Pada setiap putaran i, blok R merupakan masukan untuk fungsi transformasi yang disebut f. Pada fungsi f, blok R dikombinasikan dengan kunci internal Ki. Keluaran dari fungsi f di- XOR- kan dengan blok L untuk mendapatkan blok R yang baru. Sedangkan blok L yang baru langsung diambil dari blok R sebelumnya. Ini adalah satu putaran DES.

Secara matematis, satu putaran DES dinyatakan sebagai

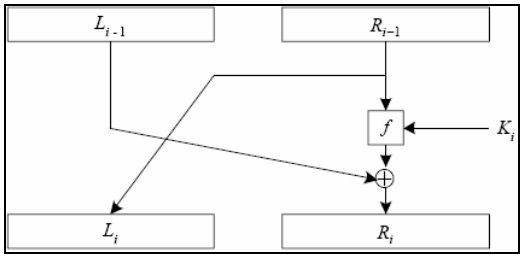
$L_i = R_{i-1}$
$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$

Gambar 2.2 memperlihatkan skema algoritma DES yang lebih rinci.



Gambar 2.2 Algoritma Enkripsi dengan DES

Satu putaran DES merupakan model jaringan Feistel (lihat Gambar 2.3).



Gambar 2.3
Jaringan Feistel untuk satu putaran DES

Perlu dicatat dari Gambar 2.2 bahwa jika (L16, R16) merupakan keluaran dari putaran ke- 16, maka (R16, L16) merupakan pra-cipherteks (pre- ciphertext) dari enciphering ini. Cipherteks yang sebenarnya diperoleh dengan melakukan permutasi awal balikan, IP- 1, terhadap blok pra- cipherteks.

2.3 Permutasi Awal

Sebelum putaran pertama, terhadap blok plainteks dilakukan permutasi awal (initial permutation atau IP). Tujuan permutasi awal adalah mengacak plainteks sehingga urutan bit- bit di dalamnya berubah. Pengacakan dilakukan dengan menggunakan matriks permutasi awal berikut ini:

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

Cara membaca tabel/matriks: dua entry ujung kiri atas (58 dan 50) berarti:

“pindahkan bit ke- 58 ke posisi bit 1”
“pindahkan bit ke- 50 ke posisi bit 2”, dst

2.4 Pembangkitan Kunci Internal

Karena ada 16 putaran, maka dibutuhkan kunci internal sebanyak 16 buah, yaitu K1, K2, …, K16. Kunci- kunci internal ini dapat dibangkitkan sebelum proses enkripsi atau bersamaan dengan proses enkripsi.

Kunci internal dibangkitkan dari kunci eksternal yang diberikan oleh pengguna. Kunci eksternal panjangnya 64 bit atau 8 karakter.

Misalkan kunci eksternal yang tersusun dari 64 bit adalah K. Kunci eksternal ini menjadi masukan untuk permutasi dengan menggunakan matriks permutasi kompresi PC- 1 sebagai berikut:

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

Dalam permutasi ini, tiap bit kedelapan (parity bit) dari delapan byte kunci diabaikan. Hasil permutasinya adalah sepanjang 56 bit, sehingga dapat dikatakan panjang kunci DES adalah 56 bit.

Selanjutnya, 56 bit ini dibagi menjadi 2 bagian, kiri dan kanan, yang masing- masing panjangnya 28 bit, yang masing- masing disimpan di dalam C0 dan D0:

C0: berisi bit- bit dari K pada posisi

57, 49, 41, 33, 25, 17, 9, 1, 58, 50, 42, 34, 26, 18
10, 2, 59, 51, 43, 35, 27, 19, 11, 3, 60, 52, 44, 36

D0: berisi bit- bit dari K pada posisi

63, 55, 47, 39, 31, 23, 15, 7, 62, 54, 46, 38, 30, 22
14, 6, 61, 53, 45, 37, 29, 21, 13, 5, 28, 20, 12, 4

Selanjutnya, kedua bagian digeser ke kiri (left shift) sepanjang satu atau dua bit bergantung pada tiap putaran. Operasi pergeseran bersifat wrapping atau round-shift.

Jumlah pergeseran pada setiap putaran ditunjukkan pada Tabel 1 sbb:

Putaran, i	Jumlah pergeseran bit
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1

10	2
11	2
12	2
13	2
14	2
15	2
16	1

Tabel 1
Jumlah pergeseran bit pada setiap putaran

Misalkan (Ci, Di) menyatakan penggabungan Ci dan Di. (C_{i+1}, D_{i+1}) diperoleh dengan menggeser Ci dan Di satu atau dua bit. Setelah pergeseran bit, (Ci, Di) mengalami permutasi kompresi dengan menggunakan matriks PC- 2 berikut:

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

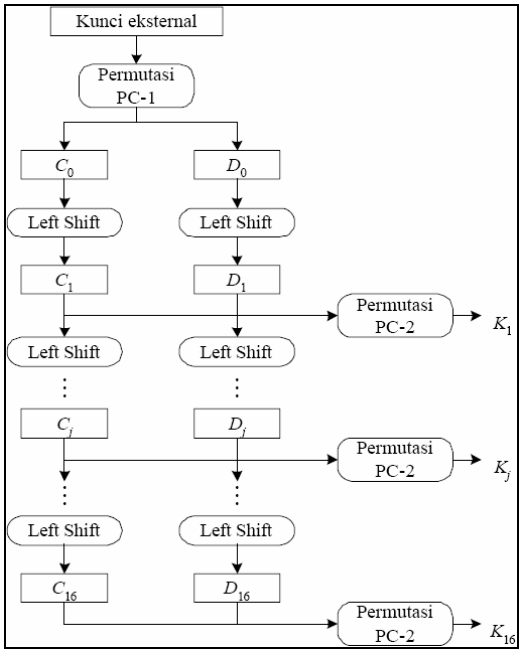
Dengan permutasi ini, kunci internal Ki diturunkan dari (Ci, Di) yang dalam hal ini Ki merupakan penggabungan bit- bit Ci pada posisi:

14,	17,	11,	24,	1,	5,	3,	28,	15,	6,	21,	10
23,	19,	12,	4,	26,	8,	16,	7,	27,	20,	13,	2

dengan bit- bit Di pada posisi:

41,	52,	31,	37,	47,	55,	30,	40,	51,	45,	33,	48
44,	49,	39,	56,	34,	53,	46,	42,	50,	36,	29,	32

Jadi, setiap kunci internal Ki mempunyai panjang 48 bit. Proses pembangkitan kunci-kunci internal ditunjukkan pada Gambar 2.4. Bila jumlah pergeseran bit- bit pada Tabel 1 dijumlahkan semuanya, maka jumlah seluruhnya sama dengan 28, yang sama dengan jumlah bit pada Ci dan Di. Karena itu, setelah putaran ke- 16 akan didapatkan kembali C16 = C0 dan D16 = D0.



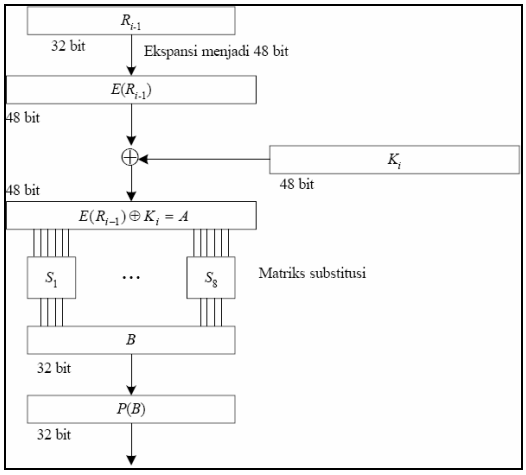
Gambar 2.4 Proses pembangkitan kunci- kunci internal DES

2.5 Enciphering

Proses enciphering terhadap blok plainteks dilakukan setelah permutasi awal (lihat Gambar 2.1). Setiap blok plainteks mengalami 16 kali putaran enciphering (lihat Gambar 2.2). Setiap putaran enciphering merupakan jaringan Feistel yang secara matematis dinyatakan sebagai

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

Diagram komputasi fungsi f diperlihatkan pada Gambar 2.5.



Gambar 2.5. Rincian komputasi fungsi f

E adalah fungsi ekspansi yang memperluas blok Ri – 1 yang panjangnya 32- bit menjadi blok 48 bit. Fungsi ekspansi direalisasikan dengan matriks permutasi ekspansi sbb:

32	1	2	3	4	5	4	5	6	7	8	9
8	9	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	1

Selanjutnya, hasil ekpansi, yaitu E(Ri – 1), yang panjangnya 48 bit di- XOR- kan dengan Ki yang panjangnya 48 bit menghasilkan vektor A yang panjangnya 48- bit:

$$E(R_{i-1}) \oplus K_i = A$$

Vektor A dikelompokkan menjadi 8 kelompok, masing- masing 6 bit, dan menjadi masukan bagi proses substitusi. Proses substitusi dilakukan dengan menggunakan delapan buah kotak- S (S- box), S1 sampai S8.

Setiap kotak- S menerima masukan 6 bit dan menghasilkan keluaran 4 bit. Kelompok 6- bit pertama menggunakan S1, kelompok 6- bit kedua menggunakan S2, dan seterusnya. Kedelapan kotak- S tersebut adalah:

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 16
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

Keluaran proses substitusi adalah vektor B yang panjangnya 48 bit. Vektor B menjadi masukan untuk proses permutasi. Tujuan permutasi adalah untuk mengacak hasil proses substitusi kotak- S. Permutasi dilakukan dengan menggunakan matriks permutasi P (P- box) sbb:

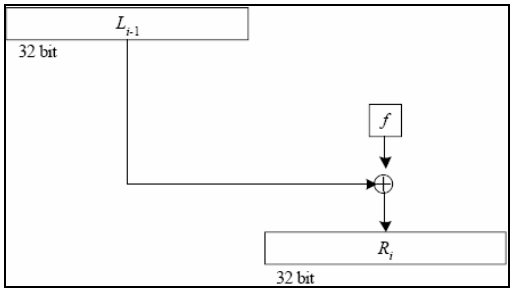
16	7	20	21	29	12	28	17	1	15	23	26	5	8	31	10
2	8	24	14	32	27	3	9	19	13	30	6	22	11	4	25

Bit- bit P(B) merupakan keluaran dari fungsi f. Akhirnya, bit- bit P(B) di- XOR- kan

dengan Li – 1 untuk mendapatkan Ri (lihat Gambar 2.6):

$$Ri = Li - 1 \oplus P(B)$$

Jadi, keluaran dari putaran ke- i adalah (Li, Ri) = (Ri – 1 , Li – 1 ⊕ P(B))



Gambar 2.6 Skema perolehan Ri

2.7 Permutasi Terakhir (Inverse Initial Permutation)

Permutasi terakhir dilakukan setelah 16 kali putaran terhadap gabungan blok kiri dan blok kanan. Proses permutasi menggunakan matriks permutasi awal balikan (inverse initial permutation atau IP- 1) sbb:

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

2.8 Dekripsi

Proses dekripsi terhadap cipherteks merupakan kebalikan dari proses enkripsi. DES menggunakan algoritma yang sama untuk proses enkripsi dan dekripsi. Jika pada proses enkripsi urutan kunci internal yang digunakan adalah K1, K2, …, K16, maka pada proses dekripsi urutan kunci yang digunakan adalah K16, K15, …, K1. Untuk tiap putaran 16, 15, …, 1, keluaran pada setiap putaran deciphering adalah

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

yang dalam hal ini, (R16, L16) adalah blok masukan awal untuk deciphering. Blok (R16, L16) diperoleh dengan mempermutasikan cipherteks dengan matriks permutasi IP- 1. Pra- keluaran dari deciphering adalah adalah (L0, R0). Dengan permutasi awal IP akan didapatkan kembali blok plainteks semula.

Tinjau kembali proses pembangkitan kunci internal pada Gambar 2.4. Selama deciphering, K16 dihasilkan dari (C16, D16) dengan permutasi PC- 2. Tentu saja (C16, D16) tidak dapat diperoleh langsung pada permulaan deciphering. Tetapi karena (C16, D16) = (C0, D0), maka K16 dapat dihasilkan dari (C0, D0) tanpa perlu lagi melakukan pergeseran bit. Catatlah bahwa (C0, D0) yang merupakan bit- bit dari kunci eksternal K yang diberikan pengguna pada waktu dekripsi.

Selanjutnya, K15 dihasilkan dari (C15, D15) yang mana (C15, D15) diperoleh dengan menggeser C16 (yang sama dengan C0) dan D16 (yang sama dengan C0) satu bit ke kanan. Sisanya, K14 sampai K1 dihasilkan dari (C14, D14) sampai (C1, D1). Catatlah bahwa (Ci – 1, Di – 1) diperoleh dengan menggeser Ci dan Di dengan cara yang sama seperti pada Tabel 1, tetapi pergeseran kiri (left shift) diganti menjadi pergeseran kanan (right shift).

2.9 Mode DES

DES dapat dioperasikan dengan mode ECB, CBC, OFB, dan CFB. Namun karena kesederhanaannya, mode ECB lebih sering digunakan pada paket program komersil meskipun sangat rentan terhadap serangan. Mode CBC lebih kompleks daripada EBC namun memberikan tingkat keamanan yang lebih bagus daripada mode EBC. Mode CBC hanya kadang- kadang saja digunakan.

2.10 Implementasi Hardware dan Software DES

DES sudah diimplementasikan dalam bentuk 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 (pada komputer mainframe IBM 3090).

2.11 Keamanan DES

Isu- isu yang menjadi perdebatan kontroversial menyangkut keamanan DES:

1. Panjang kunci
   Panjang kunci eksternal DES hanya 64 bit atau 8 karakter, itupun 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. Alasan pengurangan tidak diumumkan.

   Tetapi, dengan panjang kunci 56 bit akan terdapat 2^{56} atau 72.057.594.037.927.936 kemungkinan kunci. Jika diasumsikan serangan exhaustive key search dengan menggunakan prosesor paralel mencoba setengah dari jumlah kemungkinan kunci itu, maka dalam satu detik dapat dikerjakan satu juta serangan. Jadi seluruhnya diperlukan 1142 tahun untuk menemukan kunci yang benar.

   Tahun 1998, Electronic Frontier Foundation (EFF) merancang dan membuat perangkat keras khusus untuk menemukan kunci DES secara exhaustive search key dengan biaya $250.000 dan diharapkan dapat menemukan kunci selama 5 hari.

   Tahun 1999, kombinasi perangkat keras EFF dengan kolaborasi internet yang melibatkan lebih dari 100.000 komputer dapat menemukan kunci DES kurang dari 1 hari.

2. Jumlah putaran
   Sebenarnya, delapan putaran sudah cukup untuk membuat cipherteks sebagai fungsi acak dari setiap bit plainteks dan setiap bit cipherteks. Jadi, mengapa harus 16 kali putaran?

   Dari penelitian, DES dengan jumlah putaran yang kurang dari 16 ternyata dapat dipecahkan dengan known-plaintext attack lebih mangkus daripada dengan brute force attack.

3. Kotak- S
   Pengisian kotak- S DES masih menjadi misteri tanpa ada alasan mengapa memilih konstanta-konstanta di dalam kotak itu.

2.12 Kunci Lemah dan Kunci Setengah Lemah

DES mempunyai beberapa kunci lemah (weak key). Kunci lemah menyebabkan kunci- kunci internal pada setiap putaran sama (K1 = K2 = … = K16). Akibatnya, enkripsi dua kali berturut- turut terhadap plainteks menghasilkan kembali plainteks semula.

Kunci lemah terjadi bila bit- bit di dalam Ci dan Di semuanya 0 atau 1, atau setengah dari kunci seluruh bitnya 1 dan setengah lagi seluruhnya 0. Kunci eksternal (dalam notasi HEX) yang menyebabkan terjadinya kunci lemah adalah (ingat bahwa setiap bit kedelapan adalah bit paritas).

Kunci lemah (dengan bit paritas)	Kunci sebenarnya
0101 0101 0101 0101	0000000 0000000
1F1F 1F1F 1F1F 1F1F	0000000 FFFFFFF
E0E0 E0E0 F1F1 F11F	FFFFFFF 0000000
FEFE FEFE FEFE FEFE	FFFFFFF FFFFFFF

Selain kunci lemah, DES juga mempunyai sejumlah pasangan kunci setengah- lemah (semiweak key). Pasangan kunci setengah-lemah mengenkripsikan plainteks menjadi cipherteks yang sama. Sehingga, satu kunci dalam pasangan itu dapat mendekripsi pesan yang dienkripsi oleh kunci yang lain di dalam pasangan itu.

Kunci setengah- lemah terjadi bila:

1. Register C dan D berisi bit- bit dengan pola 0101…0101 atau 1010…1010

2. Register yang lain (C atau D) berisi bit- bit dengan pola 0000…0000, 1111…1111, 0101…0101, atau 1010…1010

Ada 6 pasang kunci setengah lemah (dalam notasi HEX):

a. `01FE 01FE 01FE 01FE` dan `FE01 FE01 FE01 FE01`

b. `1FE0 1FE0 0EF1 0EF1` dan `E01F E01F F10E F10E`

c. `01E0 01E0 01F1 01F1` dan `E001 E001 F101 F101`

d. `1FFE 1FFE 0EFE 0EFE` dan `FE1F FE1F FE0E FE0E`

e. `011F 011F 010E 010E` dan `1F01 1F01 0E01 0E01`

f. `E0FE E0FE F1FE F1FE` dan `FEE0 FEE0 FEF1 FEF1`

3. kriptanalisis pada DES

Kriptanalisis adalah pemecahan kode, yaitu pemecahan algoritma enkripsi atau sistem untuk mendapatkan plainteks atau untuk mencoba kekuatan dari algoritma yang digunakan.

3.1 Serangan Brute Force

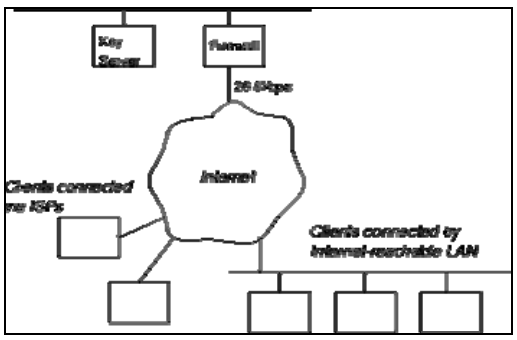
Pada tanggal 28 Januari 1997, laboratorium RSA meluncurkan beberapa tantangan kriptografis. Tujuannya adalah untuk menemukan pesan rahasia yang telah dienkripsi oleh kunci dengan panjang yang berbeda- beda. Salah satu tantangannya menggunakan algoritma DES, yang saat itu telah digunakan secara luas.

RSA menawarkan hadiah sebesar 10.000 US$ untuk tim pertama yang mampu memecahkan pesan rahasia tadi yang telah dienkripsi dengan algoritma DES. Segera setelah dua tantangan yang mudah berhasil dipecahkan, perhatian menjadi tertuju pada tantangan DES.

Dipimpin oleh Rocke Verser, Matt Curtin, dan Justin Dolske, proyek DESCHALL berusaha untuk memecahkan tantangan DES dari RSA dengan bantuan proyek komputer terdistribusi skala besar di internet. Mereka secara sederhana berusaha memecahkan 2^{56} kunci yang mungkin dapat digunakan untuk mengenkripsi pesan rahasia tadi dengan serangan brute force. Serangan brute force seperti ini sangat cocok digunakan pada komputasi paralel atau terdistribusi, karena mereka pada dasarnya terdiri dari masalah independen dalam skala raksasa yaitu mencoba setiap kunci.

Pada akhirnya DESCHALL berhasil memecahkan tantangan DES ini. Hal ini merupakan kali pertama pesan yang dienkripsi menggunakan algoritma DES dipecahkan di publik

Pendekatan DESCHALL berpusat pada server kunci tunggal yang terus memantau blok- blok kunci mana yang sudah dicoba. Client akan kemudian mengontak server via internet untuk meminta pekerjaan dan melaporkan hasilnya. Diilustrasikan dalam gambar 3.1.



Gambar 3.1 Arsitektur DESCHALL

Walaupun bukan jenis serangan yang baru, brute force key search telah menjadi

standard dimana keamanan kriptosistem dinilai. Jika suatu algoritma dipercaya aman maka itu artinya tidak terdapat kemungkinan serangan terbaik yang diketahui saat itu yang mampu menyerang sistem tersebut.

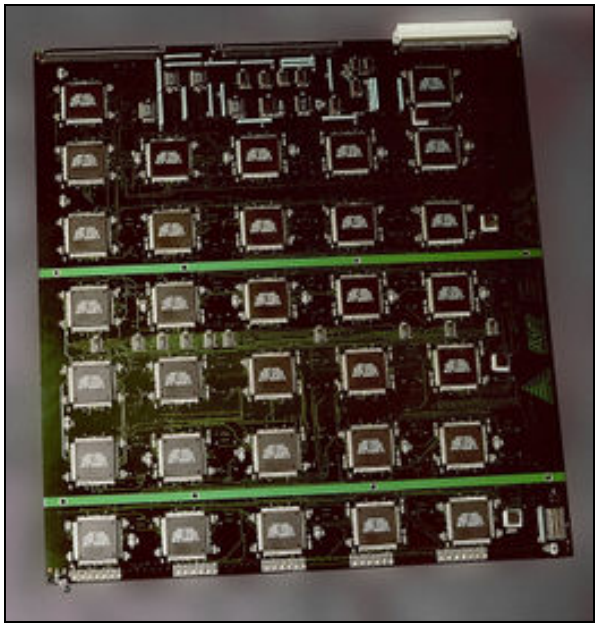
Seringkali, keamanan algoritma bisa disebut aman dengan mengukur biaya yang diperlukan untuk melakukan serangan brute force attack. Jumlah kombinasi kunci- kunci yang ada menentukan mungkin tidaknya serangan brute force berhasil.

Kemungkinan pemecahan DES dengan cepat didemonstrasikan pada tahun 1998 ketika mesin pemecah DES dibuat oleh Electronic Frontier Foundation (EFF), yaitu sebuah grup cyber pembela hak rakyat, dengan biaya kurang lebih 250.000 US$.

Tujuan utama mereka adalah untuk menunjukkan bahwa DES mampu dipecahkan secara prakteknya sesuai dengan teori: *"There are many people who will not believe a truth until they can see it with their own eyes. Showing them a physical machine that can crack DES in a few days is the only way to convince some people that they really cannot trust their security to DES."*

Mesin tadi menghasilkan kunci dengan brute force selama kurang lebih 2 hari, pada saat yang bersamaan dengan pengumuman bahwa DES tidak terpecahkan oleh departemen pengadilan Amerika Serikat.

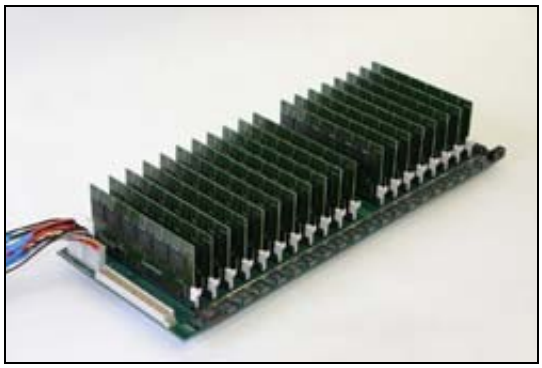
Mesin pemecah DES milik EFF mengandung 1.536 chip buatan khusus sehingga mampu memecahkan kunci DES secara brute force dalam hitungan hari. Gambar 3.2 menunjukkan papan sirkuit pemecah DES.



Gambar 3.2
Papan sirkuit pemecah DES milik EFF

Mesin pemecah DES berikutnya yang diketahui adalah mesin COPACABANA (Cost- Optimized Parallel COde Breaker) yang dibuat dengan biaya 10.000 US$ oleh Universitas Bochum dan Kiel, Jerman.

Tidak seperti mesin milik EFF, COPACABANA mengandung 120 FPGA tipe XILINX Spartan3- 1000, yaitu sirkuit yang dijual umum di pasaran dan murah, yang bekerja secara paralel. Mesin COPACABANA mampu menjalankan exhaustive search kunci pada DES dalam waktu rata- rata 9 hari. Gambar 3.3 menunjukkan ukuran penuh dari mesin ini.



Gambar 3.3
Mesin COPACABANA

Harga COPACABANA berkurang sampai dengan 25 kali dari harga mesin EFF. Ini adalah contoh yang sangat baik dari perkembangan pesat hardware digital.

Dalam dunia akademik, berbagai proposal untuk mesin pemecah DES terus bertambah. Pada tahun 1977, Diffie dan Hellman menawarkan sebuah mesin yang berharga kurang lebih 20.000 US$ yang sanggup menemukan kunci DES dalam waktu satu hari. Pada tahun 1993, Wiener menawarkan proposal mesin pencari kunci seharga 1.000 US$ yang dapat menemukan kunci dalam waktu 7 jam.

3.2 Serangan yang lebih cepat dari brute force

Ada tiga serangan yang diketahui dapat memecahkan enam belas putaran penuh dari DES dengan kompleksitas yang lebih rendah dari pencarian brute force. Namun, ketiga serangan ini hanya sebatas teori dan tidak memungkinkan untuk dilaksanakan.

Tipe serangan semacam ini biasa diistilahkan dengan *certificational weaknesses*. Ketiga serangan itu yaitu:

1. Kriptanalisis differential
   Kriptanalisis differential ditemukan pada akhir tahun 1980- an oleh Eli Biham dan Adi Shamir, walaupun sebenarnya hal ini juga telah diketahui oleh IBM dan NSA namun mereka merahasiakannya.

   Untuk memecahkan 16 putaran penuh, kriptanalisis differential membutuhkan 2^{47} *chosen plaintext (chosen plaintext attack)*. DES didesain untuk kebal terhadap serangan ini.

2. Kriptanalisis linear
   Kriptanalisis linear ditemukan oleh Mitsuru Matsui dan membutuhkan 2^{43} *known plaintext (known plaintext attack).*

   Metode ini diimplementasikan dan merupakan eksperimen kriptanalisis terhadap DES yang pertama yang dilaporkan. Tidak ada bukti bahwa DES kebal terhadap serangan ini. Generalisasi dari kriptanalisis linear yaitu *multiple linear cryptanalysis* diperkenalkan pada tahun 1994 dan diperbaharui oleh Biryukov dan kawan- kawan pada tahun 2004.

   Analisis mereka menyarankan bahwa perkiraan multiple linear dapat digunakan untuk mengurangi jumlah serangan.

3. Serangan Davies
   Sementara kriptanalisis linear dan differential adalah teknik umum yang dapat diterapkan ke banyak skema algoritma, serangan davies terspesialisasi untuk DES saja.

   Pertama kali dikemukakan oleh Davies pada tahun 80- an, dan kemudian dekembangkan oleh Biham dan Biryukov pada tahun 1997.

   Bentuk serangan yang paling hebat membutuhkan 2^{50} *known plaintext (known plaintext attack)*, dengan kompleksitas komputasi sebesar 2^{50} dan mempunyai kemungkinan sukses 51%.

Terlepas dari semua kritikan dan kelemahan-kelemahan dari DES, sampai saat ini belum ditemukan contoh orang- orang yang telah menderita kerugian materi karena terbatasnya keamanan DES.

4. Triple DES

Karena DES mempunyai potensi kelemahan terhadap exhaustive key search attack/ brute force attack, sebagai akibat dari panjang kunci yang relatif pendek (hanya 56 bit) maka dibuat varian dari DES.

Varian dari DES dibuat dengan cara memperbesar ruang kunci tanpa perlu mengubah algoritma DES. Varian DES yang paling luas digunakan adalah DES berganda (multiple DES). DES berganda adalah

enkripsi berkali- kali dengan DES dan menggunakan kunci ganda. DES berganda yang dikenal adalah Double DES dan Triple DES. Double DES mengeksekusi algoritma DES dua kali, sedangkan Triple DES tiga kali.

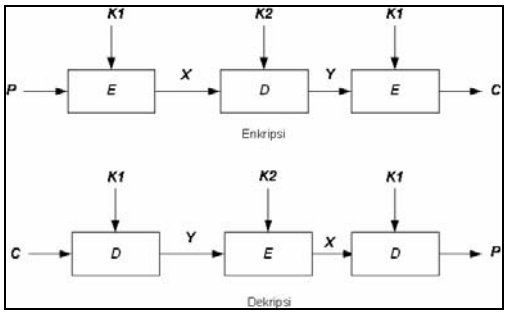
Triple DES atau TDES menggunakan DES tiga kali. Pengunaan tiga langkah ini penting untuk mencegah serangan meet-in-the-middle sebagaimana pada Double DES. Bentuk sederhana Triple DES adalah:

Enkripsi: $C = E_{K3}(E_{K2}(E_{K1}(P)))$
Dekripsi: $P = D_{K1}(D_{K2}(D_{K3}(C)))$

Varian ini umum dikenal sebagai mode EEE (untuk enkripsi) karena pada proses enkripsi semuanya menggunakan enkripsi. Untuk menyederhanakan interoperability antara DES dan TDES, maka langkah di tengah (pada proses enkripsi TDES) diganti dengan dekripsi (mode EDE). Dengan perubahan ini maka dibuat beberapa versi TDES. Versi pertama TDES menggunakan dua buah kunci, K1 dan K2.

Enkripsi: $C = E_{K1}(D_{K2}(E_{K1}(P)))$
Dekripsi: $P = D_{K1}(E_{K2}(D_{K1}(C)))$

Enkripsi DES tunggal dengan kunci K dapat dinyatakan sebagai TDES-EDE dengan K1 = K2 = K. Gambar 4.1 memperlihatkan versi TDES yang menggunakan dua buah kunci. Penggunaan enkripsi pada langkah di tengah tidak mempengaruhi keamanan algoritma.

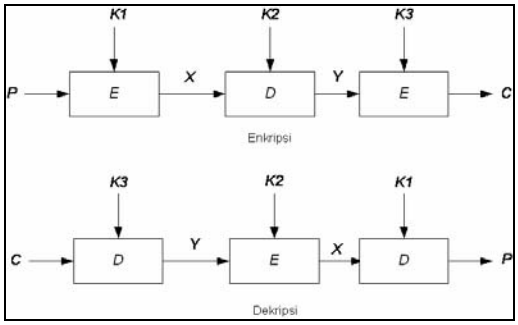


Gambar 4.1 TDES 2 kunci

Versi TDES kedua menggunakan tiga buah kunci, K1, K2, dan K3:

Enkripsi: $C = E_{K3}(D_{K2}(E_{K1}(P)))$
Dekripsi: $P = D_{K1}(E_{K2}(D_{K3}(C)))$

Gambar 4.2 memperlihatkan versi TDES yang menggunakan tiga buah kunci.



Gambar 4.2 TDES 3 kunci

TDES secara perlahan mulai tidak digunakan lagi, sebagian besar digantikan oleh penerus utamanya AES (Advanced Encryption Standard). Satu pengecualian besar yaitu dalam industri pembayaran elektronik dimana masih menggunakan TDES dua kunci dan berlanjut kepada pengembangan dan mengumumkan standard resmi berdasarkannya.

Hal ini menjamin bahwa Triple DES akan tetap aktif dengan baik sebagai standar kriptografi di masa depan.

Dilihat dari sudut pandang desainnya, DES dan tentu saja Triple DES memiliki kekurangan pada performansi yang lambat dalam software.

Dibandingkan menggunakan prosesor-prosesor modern, AES cenderung kurang lebih enam kali lebih cepat dibandingkan DES. Triple DES lebih baik diterapkan pada implementasi hardware, dan memang kecenderungannya bidang penggunaan Triple DES saat ini sebagian besar merupakan implementasi hardware, walaupun AES mampu menyalip Triple DES.

5. Kekuatan Triple DES

Secara umum, Triple DES dengan dua buah kunci mempunyai panjang kunci 2 x 56 = 112 bit, jauh lebih pendek daripada Triple DES dengan tiga buah kunci yang

mempunyai panjang kunci 168 bit (3 x 56 bit).

Namun karena serangan meet-in-the-middle, keamanan efektif untuk TDES dengan tiga kunci hanya sebesar 112 bit. Begitu pula dengan TDES dua kunci, mode ini mudah terkena serangan chosen plaintext ataupun known plaintext sehingga secara resmi memiliki besar keamanan 80 bit.

Untuk seluruh mode operasi Triple DES, tiga kunci kriptografi (K1, K2, dan K3) menjadi sebuah set kunci Triple DES. Setiap set dan masing- masing kuncinya harus memenuhi syarat- syarat berikut ini:

a. Rahasia

b. Digenerate secara random atau pseudo- random

c. Independen terhadap set kunci lainnya

d. Memiliki integritas dimana setiap kunci dalam set tidak pernah diubah secara tidak benar sejak pertama kali kunci itu digenerate, dikirim, atau disimpan oleh sumber yang sah.

e. Digunakan dalam urutan yang benar sesuai mode yang dipakai

f. Dianggap sebagai satu kesatuan dimana satu kunci tidak boleh diubah tanpa mengubah dua kunci lainnya dan tidak bisa dipisahkan dari set kecuali untuk kepentingan desain.

Keamanan Triple DES dipengaruhi oleh jumlah blok yang diproses oleh satu set kunci. Satu set kunci sebaiknya tidak digunakan untuk memproses lebih dari 2^{32} 64 bit blok data.

Pada tahun 2005, serangan terbaik terhadap Triple DES tiga kunci membutuhkan sekitar 2^{32} known plaintext, 2^{113} langkah, 2^{90} enkripsi DES tunggal, dan 2^{88} memori. Hal ini sangat tidak praktis.

Jika penyerang berniat untuk mencari salah satu dari kunci-kunci kriptografinya, ada serangan yang menghemat memori yang akan menemukan satu dari 2^{28} kunci, dengan memiliki chosen plaintext per kuncinya dan sekitar 2^{84} operasi enkripsi.

Serangan ini sangat bisa dilakukan secara paralel dan mungkin bisa diterapkan, dengan menginvestasikan dana milyaran dollar dan tahunan untuk melaksanakan serangan. Apalagi mengingat keadaan yang bisa dimanfaatkan untuk hal ini sangat terbatas.

6. Kesimpulan

Dari hasil studi dan analisis terhadap kelemahan algoritma DES dan kekuatan algoritma Triple DES di atas dapat ditarik beberapa kesimpulan:

1. Algoritma cipher blok DES saat ini sudah tidak bisa dimanfaatkan lagi sebagai algoritma yang aman. Hal ini bisa dilihat dari fakta- fakta perkembangan kriptanalisis terhadap algoritma ini yang dari tahun ke tahun semakin maju. Belum lagi perkembangan teknologi sendiri yang mendukung pencarian exhaustive search yang semakin cepat lagi.

2. Pencarian kunci dengan metode exhaustive search (brute force attack) tidak dapat dianggap sebelah mata. Malahan metode ini telah dijadikan patokan untuk menentukan kekuatan algoritma kriptografi.

3. Alasan utama mengapa algoritma cipher blok DES menjadi tidak aman lagi adalah terutama pada panjang kunci yang dipakai dalam algoritma ini masih terlalu pendek.

4. Atas dasar alasan inilah maka kemudian diciptakan varian- varian dari algoritma cipher blok DES yang kekuatannya terletak pada panjang kunci yang kini berlipat dua dan tiga.

5. Varian dari algoritma cipher blok Triple DES masih dinilai aman karena panjang kunci yang dibutuhkan algoritma ini cukup

panjang untuk menjamin kompleksitas enciphering. Varian DES ini masih menggunakan algoritma yang sama dengan DES karena dengan demikian memudahkan migrasi sistem yang dulunya mempergunakan DES untuk beralih ke Triple DES.

6. Algoritma Triple DES walaupun aman namun pemanfaatannya saat ini semakin berkurang. Hal ini disebabkan oleh waktu performansi algoritma ini dan algoritma DES yang kurang baik ketika diterapkan dalam software. Sehingga saat ini pemanfaatan terbesar algoritma ini terletak pada implementasi hardware.

Daftar Pustaka

[1] CRYPTOGRAPHY RESEARCH (2006). DES DECERTIFICATION Q&A. *http://www.cryptography.com/cnews/des.html*. Tanggal akses: 12 Oktober 2006 pukul 10:00.

[2] CRYPTOGRAPHY RESEARCH (2006). RECORD BREAKING DES KEY SEARCH COMPLETED. *http://www.cryptography.com/cnews/des.html*. Tanggal akses: 12 Oktober 2006 pukul 10:00.

[3] Lucks, Stevan. (1998). Attacking Triple Encryption. Fast Software Encyription. Springer LNCS.

[4] Munir, Rinaldi. (2006). Bahan Kuliah IF5054 Kriptografi. Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung.

[5] NIST FIPS PUB 46-2 (1988). Data Encryption Standard (DES). *http://www.itl.nist.gov/fipspubs/fip46-2.htm* Tanggal akses: 12 Oktober 2006 pukul 10:00.

[6] RSA Laboratories (1997). RSA to Launch "DES Challenge II" at Data Security Conference. *http://www.rsasecurity.com/press_release.asp?doc_id=729&id=1034* Tanggal akses: 12 Oktober 2006 pukul 10:00.

[7] RSA Laboratories (1998). RSA to Launch DES Challenge III Contest at 1999 Data Security Conference. *http://www.rsasecurity.com/press_release.asp?doc_id=627&id=1034.* Tanggal akses: 12 Oktober 2006 pukul 10:00.

[8] RSA Laboratories (2006). What is DES?. *http://www.rsasecurity.com/rsalabs/node.asp?id=2226* Tanggal akses: 12 Oktober 2006 pukul 10:00.

[9] Schneier, Bruce. (1996). Applied Cryptography 2nd. John Wiley & Sons.

[10] The Risks Digest (1996). A New Attack on DES. *http://catless.ncl.ac.uk/Risks/18.54.html#subj1.1* Tanggal akses: 12 Oktober 2006 pukul 10:00.

[11] The Risks Digest (1989). Collition in DES. *http://catless.ncl.ac.uk/Risks/8.21.html#subj3.1* Tanggal akses: 12 Oktober 2006 pukul 10:00.

[12] The Risks Digest (1987). DES and NSA's now code. *http://catless.ncl.ac.uk/Risks/6.01.html#subj4.1* Tanggal akses: 12 Oktober 2006 pukul 10:00.

[13] The Risks Digest (2006). Triple DES Upgrades. *http://catless.ncl.ac.uk/Risks/24.27.html#subj14.1* Tanggal akses: 12 Oktober 2006 pukul 10:00.

[14] Wikipedia. (2006). Data Encryption Standard. *http://en.wikipedia.org/wiki/Data_Encryption_Standard*. Tanggal akses: 12 Oktober 2006 pukul 10:00.

[15] Wikipedia. (2006). Differential Cryptanalysis. *http://en.wikipedia.org/wiki/Differential_cryptanalysis*. Tanggal akses: 12 Oktober 2006 pukul 10:00.

[16] Wikipedia. (2006). Linear Cryptanalysis. *http://en.wikipedia.org/wiki/Linear_cryptanalysis*. Tanggal akses: 12 Oktober 2006 pukul 10:00.

[17] Wikipedia. (2006). Triple DES. *http://en.wikipedia.org/wiki/Triple_DES*. Tanggal akses: 12 Oktober 2006 pukul 10:00.