Odd-Even DES

Alif Raditya Rochman
13511013
Teknik Informatika ITB
Bandung, Indonesia
alifradityar@gmail.com

Muhamad Ihsan
13511049
Teknik Informatika ITB
Bandung, Indonesia
13511049@std.stei.itb.ac.id

***Abstrak*—Setiap tahunnya, dunia kriptografi terus berkembang dengan munculnya algoritma-algoritma baru. Selain itu muncul juga pengembangan dari algoritma yang sudah ada ditambahkan dengan suatu fitur tertentu. Odd-Even DES merupakan algoritma yang mencoba melakukan peningkatan kualitas proses pembangkit kunci internal dengan menggunakan suatu fungsi yang memanfaatkan aspek bilangan ganjil dan genap. Penambahan fungsi ini menimbulkan penurunan performa yang kecil sehingga bisa di applikasikan.**

Kata Kunci— Odd, Even, DES, Cryptography, Key

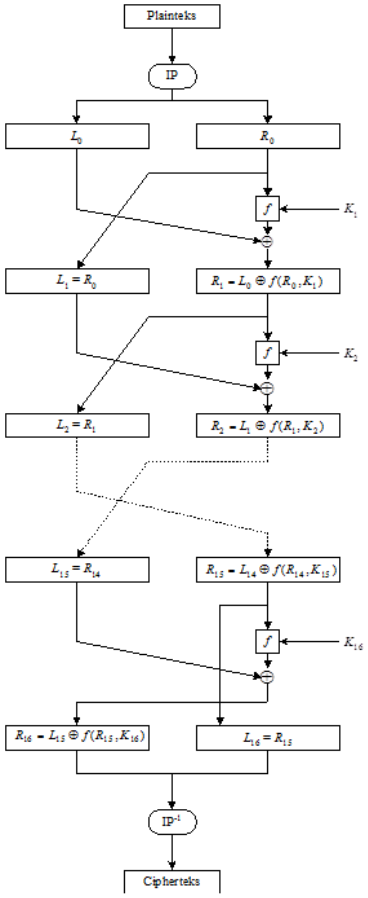
I. PENDAHULUAN

Pada masa-masa kejayaannya, DES merupakan algoritma kunci-simetri yang paling kuat dan paling sering digunakan untuk mengenkripsi data elektronik. Penelitian terkait DES pun membuat pemahaman orang-orang mengenai *cipher* blok dan kriptanalisis DES meningkat. Tetapi sayangnya, DES sekarang dianggap tidak cukup kuat karena panjang keynya yang hanya 56 bit membuatnya menjadi mudah dipecahkan. Konsep dasar dari DES itu sendiri sudah cukup bagus, terbukti dengan penggunaannya yang sempat cukup luas, diharapkan dengan beberapa modifikasi, DES dapat menjadi lebih kuat dan lebih sulit dipecahkan.

II. DASAR TEORI

DES (*Data Encryption Standard*) merupakan standard yang dikembangkan di IBM pada tahun 1972. DES tergolong kedalam kriptografi kunci-simetri dan tergolong juga kedalam jenis *cipher* blok. DES beroperasi pada ukuran blok 64 bit. Panjang sebuah kunci eksternal DES adalah 64 bit, tetapi hanya 56 bit yang dipakai dalam perhitungan.

Setiap blok awalnya dipermutasi dengan menggunakan IP (*Initial Permutation*), kemudian di enkripsi sebanyak 16 putaran, dan terakhir dipermutasi lagi dengan menggunakan *inverse* IP. Hasil akhirnya merupakan cipherteks dari blok plainteks yang dijadikan input DES. Setiap putaran pada tahap enkripsi menggunakan kunci internal yang berbeda-beda. Kunci internal dibangkitkan dari kunci eksternal, dan besarnya 56 bit.



Gambar 1. Skema umum algoritma DES

Matriks yang digunakan untuk IP dan inverse IP masing-masing adalah sebagai berikut :

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

Gambar 2. Matriks IP

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

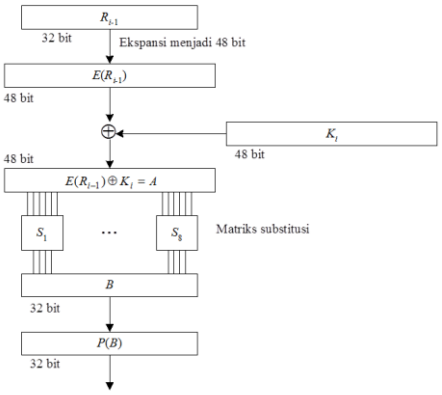
Gambar 3. Matriks *inverse* IP

Sebuah putaran *enciphering* merupakan jaringan Feistel, dimana hasil Li dan Ri masing-masing adalah :

Li = Ri-1

Ri = Li-1 (xor) f(Ri-i, Ki)

Fungsi f akan dijelaskan melalui gambar dibawah ini :



Gambar 4. Skema Fungsi f

Pertama, Ri-1 diekspansi dengan menggunakan matriks ekspansi, dari awalnya berukuran 32 bit menjadi 48 bit. Kemudian hasilnya di-xor-kan dengan kunci internal untuk putaran tersebut (Ki). Hasilnya akan berupa sebuah vektor A berukuran 48 bit. A kemudian dipecah menjadi 8 buah vektor kecil berukuran 6 bit, dan sesuai urutannya, dimasukkan satu per satu ke s-box (*substitution box*) yang akan menerima input 6 bit dan mengeluarkan hasil 4 bit. Hasil substitusi tersebut adalah vektor B yang berukuran 32 bit. B kemudian diacak lagi dengan menggunakan matriks permutasi P. Hasil permutasi terakhir tersebut lah yang menjadi keluaran fungsi f.

Matriks-matriks yang dipakai di dalam fungsi f adalah :

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

Gambar 5. Matriks permutasi ekspansi

S_1:

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

S_2:

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

S_3:

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

S_4:

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

S_5:

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

S_6:

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

S_7:

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

S_8:

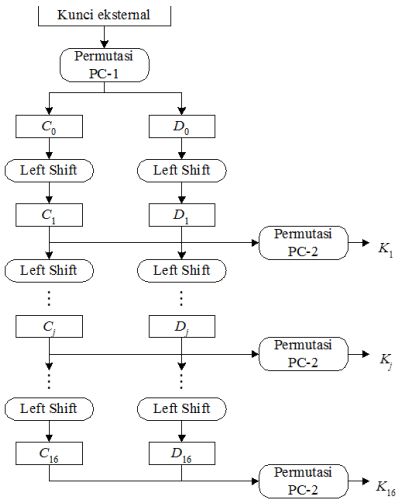
13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Gambar 6. S-boxes untuk DES

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

Gambar 7. Matriks Permutasi P

Pembangkitan kunci eksternal dilakukan agar terbentuk 16 buah kunci internal yang berbeda untuk digunakan di setiap putaran enkripsi. Setiap kunci internal yang dihasilkan akan berukuran 48 bit. Kunci dibangkitkan dengan mengikuti skema berikut ini :



Gambar 8. Pembangkitan kunci eksternal

Matriks-matriks permutasi yang digunakan dalam pembangkitan kunci ada dua, yaitu :

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

Gambar 9. Matriks Permutasi Kompresi PC-1

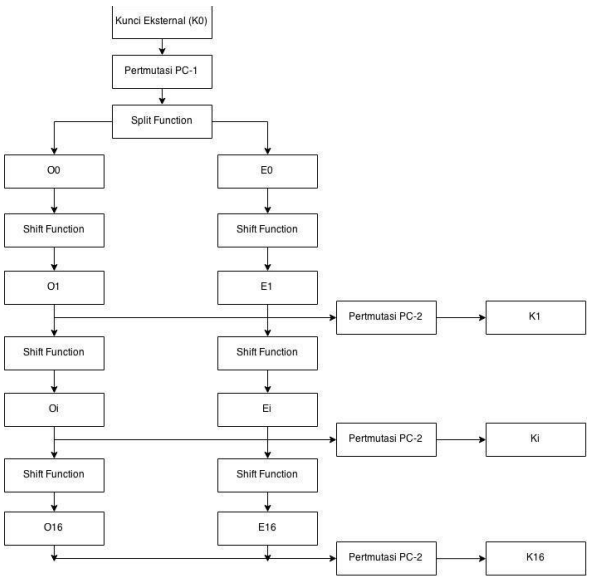
14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

Gambar 10. Matriks PC-2

Untuk dekripsi algoritma DES, prosedurnya sama persis dengan enkripsi, tetapi terletak di urutan penggunaan kuncinya. Pembangkitan kunci eksternal menghasilkan kunci internal K1,K2,...,K16, dan pada enkripsi, kunci K1 digunakan untuk putaran enkripsi pertama, K2 untuk putaran enkripsi kedua, dan seterusnya. Sedangkan pada dekripsi, kunci K16 lah yang digunakan untuk putaran dekripsi pertama, K15 untuk putaran kedua, dan seterusnya.

III. PEMBENTUKAN ODD-EVEN DES

Ide awal dari pembuatan Odd-Even DES berasal dari sebuah mengenai proses pembentukan kunci internal yang digunakan oleh algoritma DES. Pembentukan proses kunci internal memanfaatkan sifat indeks yang memiliki dua jenis tipe : yaitu indeks ganjil dan genap. Suatu kunci eksternal yang berasal dari user akan dipermutasikan dengan matriks yang sama saat menggunakan algoritma DES. Tetapi ditambahkan suatu proses tambahan untuk membagi hasilnya menjadi dua bagian.



Gambar 11. Pembangkitan kunci Odd-Even DES

Pada *split function*, dibuat dua buah serangkaian bit dari K0 yang saling lepas : O0 dan E0. Dengan memanfaatkan kunci eksternal yang diberikan dan permutasi PC-1 yang sudah ada, pembentukan O0 dan E0 dapat ditulis sebagai berikut :

$$O0 = K[P[1]],\ K[P[3]],\ K[P[5]],\ \ldots\ K[P[n-1]]$$

$$E0 = K[P[0]],\ K[P[2]],\ K[P[4]],\ \ldots\ K[P[n]]$$

Pada *shift function*,dibentuk dua buah serangkaian bit Oi dan Ei dengan menggunakan Oi-1 dan Ei.Shift function juga memanfaatkan sifat indeks kelipatan ganjil dan genap. Terdapat dua tahap fungsi ini : swap dan shift. Pada tahap swap, dilakukan pergantian antara dua buah bit dengan indeks yang bertetanggaan pada setiap kelipatan dua. Proses ini dapat ditulis sebagai berikut :

$$O_i[j] = \begin{cases} O_{i-1}[j+1], & j \bmod 2 = 2 \\ O_{i-1}[j-1], & j \bmod 2 = 0 \end{cases}$$

Tahap shift pada shift function sama dengan yang dapat ditemukan pada DES yaitu menggeser indeks ke kiri atau ke indeks yang lebih kecil dan memindahkan bit pada indeks paling kecil ke indeks paling besar. Shift function membuat pergeseran yang tetap mempertahankan sifat indeks ganjil dan genap yang ada. Shift function membuat hasil Oi dan Ei akan mengalami perulangan apabila dilakukan lebih dari 16 kali, sehingga secara umum pembentukan K hanya sampai 16.

Ketika dilakukan permutasi, proses yang dilakukan tidaklah berbeda tetapi akan menghasilkan key internal yang berbeda. Proses ini tidak mempengeruhi proses lainnya, hanya mengubah pembentukan key internalnya saja.

Proses deskripsi yang dibuat disesuaikan dengan proses enkripsi yang dibangun. Karena split function dan shift function merupakan proses yang tidak terlalu kompleks yang ditambahkan kepada proses DES yang normal, tidak ada hal yang cukup spesial yang ditambahkan.

IV. IMPLEMENTASI DAN PERFORMA

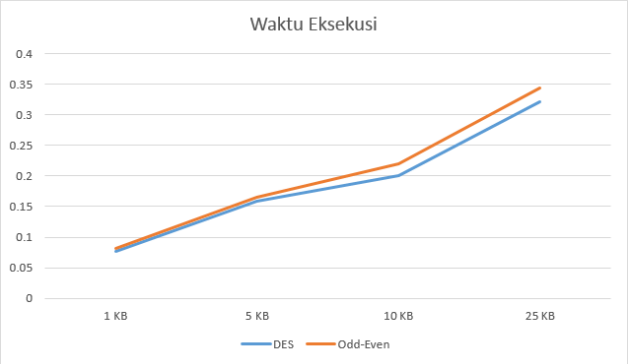
Pengujian dari algoritma ini dilakukan dengan menggunakan bahasa Java dan dijalankan pada sistem desktop dengan Intel Core i3 3217U dan 4GB RAM DDR3. Program dicompile dengan Java Development Kit 1.7. Terdapat 4 data uji yang digunakan yang dikategorikan sebagai ukuran kecil, menengah, cukup besar, dan besar. Dari data uji yang ada, didapatkan hasil seperti dengan tabel berikut.

Tabel 1. Hasil pengujian

No.	Hasil Pengujian		
	Ukuran daa uji	*DES*	*Odd-Even*
1	1 Kb	0.077 s	0.081 s
2	5 Kb	0.159 s	0.165 s
3	10 Kb	0.201 s	0.220 s
4	25 Kb	0.322 s	0.344 s

Berdasarkan tabel pengujian, terlihat bahwa dengan menggunakan Odd-Even DES terdapat peningkatan waktu eksekusi dengan rata-rata sebesar 8%. Hal ini dikarenakan

algoritma Odd-Even tidak mencoba mengoptimasi waktu eksekusi. Tetapi peningkatan tersebut bisa dibilang tidak terlalu signifikan sehingga penggunakan algortima ini memungkinkan pada kasus nyata.



Gambar 12. Grafik perbandingan waktu eksekusi

V. KESIMPULAN

Dengan peningkatan waktu pemrosesan yang cukup kecil daripada DES, algoritma Odd-Even DES dapat digunakan pada enkripsi untuk kasus nyata. Dengan konsep yang sedikit berbeda pada proses pembangkitan kunci internal, Odd-Even DES merupakan suatu varian yang dapat dipertimbangkan ketika mengimplementasikan algoritma kriptografi.

REFERENCES

[1] Coppersmith, Don. (1994). The data encryption standard (DES) and its strength against attacks at the Wayback Machine (archived June 15, 2007). IBM Journal of Research and Development,38(3), 243–250.

[2] Christof Paar, Jan Pelzl, "The Data Encryption Standard (DES) and Alternatives", free online lectures on Chapter 3 of "Understanding Cryptography, A Textbook for Students and Practitioners". Springer, 2009.

[3] Mitsuru Matsui (1994). "The First Experimental Cryptanalysis of the Data Encryption Standard". Lecture Notes in Computer Science 839: 1–11. doi:10.1007/3-540-48658-5_1