ANALISIS SANDI DIFERENSIAL TERHADAP FULL DATA ENCRYPTION STANDARD

Yusuf Kurniawan[1], Adang Suwandi[2], M. Sukrisno Mardiyanto[2], Iping Supriana S.[2]
[1]Jurusan Teknik Informatika Universitas Pasundan
Jalan Setiabudi 193 Bandung
[2]Departemen Teknik Informatika Institut Teknologi Bandung
Jl Ganesha 10 Bandung
E-mail: [1]ysfk2002@yahoo.om, [2]sukrisno@informatika.org, [2]iping@informatika.org

Abstrak

DES (Data Encryption Standard) merupakan block cipher 16 ronde yang memiliki struktur Feistel dan memiliki masukan/keluaran 64 bit, serta memiliki kunci sepanjang 56 bit. Dengan struktur Feistel, algoritma enkripsi memiliki struktur yang sama dengan yang untuk dekripsi. Perbedaannya hanya terletak pada urutan subkey yang dimasukkan. DES memiliki fungsi F yang tidak invertible. Analisis Sandi Diferensial (ASD) adalah teknik untuk mendapatkan kunci algoritma enkripsi modern tanpa harus meminta ijin pemilik kunci. ASD dapat diterapkan terhadap bermacam-macam algoritma enkripsi modern, termasuk Data Encryption Standard (DES). ASD berusaha mendapatkan diferensial masukan yang menghasilkan diferensial keluaran dengan memiliki peluang sebesar mungkin. Untuk algoritma DES, diferensial ini berupa XOR. Pada DES, satu-satunya komponen yang tidak linear adalah kotak subtitusi. Oleh karena itu, ASD dimulai dari Kotak-S ini. Kemudian aliran diferensial ini diperluas ke komponen lainnya hingga mencapai satu ronde. Kemudian, dari satu ronde, diperluas ke ronde-ronde berikutnya hingga mencapai 16 ronde penuh.Dalam makalah ini akan ditunjukkan cara memecahkan kunci DES 56 bit 16 ronde dengan ASD. Memecahkan DES berarti bahwa kita dapat membuktikan bahwa tingkat keamanan full DES 16 ronde lebih rendah daripada yang diklaim pada waktu algoritma diumumkan. Dengan kata lain, DES dapat dipecahkan lebih cepat dari pada brute force attack.

Kata kunci: Full DES, Analisis sandi diferensial, Kotak-S.

1. PENDAHULUAN

Data Encryption Standard (DES) telah digunakan sejak pertengahan tahun 1970-an. Karena hanya menggunakan kunci 56 bit, maka DES dianggap kurang aman dan akhirnya digantikan oleh Advanced Encryption Standard (AES) sejak tahun 2001. Namun karena DES telah banyak ditanamkan ke dalam jutaan perangkat keras dan lunak, maka DES tidak dibuang begitu saja. Namun agar DES dapat digunakan dengan aman, DES yang digunakan adalah Triple DES (TDES), yang mana algoritma DES dijalankan 3 kali terhadap suatu *plaintext*. Enkripsi dapat berupa $(E_{K3}(E_{K2}(E_{K1}(P))))$ bila digunakan tiga kunci 168 bit. Dengan demikian tingkat keamanan TDES cukup aman jika dibandingkan dengan AES.

Feistel *cipher* merupakan struktur *block cipher* yang ditemukan oleh Feistel pada awal tahun 1970-an ketika sedang merancang DES. Struktur ini sangat terkenal seiring dengan ketenaran DES dan karena memiliki struktur enkripsi dan dekripsi yang sama. Ini berarti bahwa tingkat keamanan enkripsi dan dekripsi juga akan sama. Selain itu, cukup satu implementasi untuk enkripsi dan dekripsi.

Cipher Feistel yang memiliki blok berukuran 2n dengan r ronde didefinisikan sebagai berikut[3]:

$$g: GF(2)^n \times GF(2)^n \times GF(2)^m \rightarrow GF(2)^n \times GF(2)^n$$
$$g(X,Y,Z) = (Y, F(Y,Z) + X) \quad (1)$$

di mana F dapat berupa sebarang fungsi yang tidak perlu *invertible* dan mengambil dua argumen n bit dan m bit serta menghasilkan n bit. Lambang + merupakan operasi grup komutatif terhadap blok n bit. Operasi + pada GF(2) adalah XOR.

Bila diketahui *plaintext* $P = (P^L, P^R)$ dan kunci setiap ronde berupa $K_1, K_2, ...K_r$, maka ciphertext $C = (C^L, C^R)$ dapat diperoleh setelah mengoperasikan P dalam r ronde (tahap). Bila $C_0^L = P^L$ dan $C_0^R = P^R$ serta i = 1,2,...r, maka

$$(C_i^L, C_i^R) = (C_{i-1}^R, F(C_{i-1}^R, K_i) + C_{i-1}^L) \quad (2)$$

Kemudian set $C_i = (C_i^L, C_i^R)$ dan $C^L = C_r^R$ serta $C^R = C_r^L$. Kunci ronde $(K_1, K_2, ..., K_r)$ di mana $K_i \in GF(2)^m$ dihitung dengan algoritma penjadwalan kunci dengan masukan kunci induk K.

2. PELUANG DIFFERENTIAL

Peluang diferensial[1] (DP) mengukur korelasi antara beda masukan dengan beda keluaran terhadap pemetaan *boolean*. Bila B: $\{0,1\}^d \rightarrow \{0,1\}^d$ merupakan pemetaan bijektif, dan $\Delta x, \Delta y \in \{0,1\}^d$ merupakan suatu nilai yang tetap, maka jika $X \in \{0,1\}^d$ merupakan variabel acak terdistribusi serbasama, maka peluang diferensial $DP(\Delta x,\Delta y)$ didefinisikan sebagai:

$$DP(\Delta x,\Delta y) \equiv Prob_x \{ B(x) \oplus B(x \oplus \Delta x) = \Delta y\} \quad (3)$$

ISBN: 979-756-061-6

Bila B diberi parameter k, maka kita tulis DP(Δx, Δy; k) dan peluang diferensial yang diharapkan EDP(Δx,Δy) didefinisikan sebagai:

$$EDP(\Delta x,\Delta y) \equiv E_K[\, DP(\Delta x, \Delta y; k)\,] \quad (4)$$

di mana K merupakan variabel acak serba sama yang terdistribusi pada seluruh kemungkinan kunci.

Kita dapat melihat variabel DP(Δx,Δy) (EDP(Δx,Δy)) sebagai isi tabel berukuran 2^d x 2^d, yang pada DES berupa tabel berukuran 2^6 x 2^4.

Isi tabel XOR B(DES): $\{0,1\}^6 \rightarrow \{0,1\}^4$ dapat didefinisikan dengan

$$XOR(\Delta x,\Delta y) \equiv \#\{x \in \{0,1\}^6 : B(x) \oplus B(x \oplus \Delta x) = \Delta y\} \quad (5)$$

untuk $\Delta x,\Delta y \in \{0,1\}^6$. Ini berarti DP(Δx,Δy) =

$$\frac{XOR(\Delta x, \Delta y)}{2^d} \quad (6)$$

DP untuk r ronde $DP^r(\Delta P,\Delta C)$ adalah Pr:

$$Pr=(\Delta C=\delta_r|\Delta P=\delta_o)=\sum_{\delta_1} \sum_{\delta_{r-1}} \prod_{i=1}^{r} \Pr\,(\Delta C_i=\delta_i|\Delta C_{i-1}=\delta_{i-1}) \quad (7)$$

Di mana $\Delta C_0 = \Delta P$

Pada umumnya, suatu diferensial akan memiliki peluang yang lebih besar dibanding karakteristik (seperti pada konsep *linear hull*) Penggabungan karakteristik di antara banyak ronde dengan asumsi bahwa *subkey* saling bebas dan terdistribusi serba sama.

Setiap fungsi boolean f: $\{0,1\}^d \rightarrow \{0,1\}$ dapat dituliskan sebagai polinomial dari bit-bit masukan:

$$f(x_1,x_2, ...x_d) = a_0 + \sum_{1\le i\le d} a_i x_i + \sum_{1\le i\le j\le d} a_{i,j} x_i x_j + ... + a_{1,2,...d} x_1 x_2 ... x_d \quad (8)$$

dengan $a_0, ..., a_{1,2,...,d} \in \{0,1\}$ merupakan koefisien, sedangkan operasi perkalian dan penjumlahan dalam GF(2) berupa operasi AND dan XOR per bit. Derajat fungsi ini merupakan jumlah terbanyak x_i yang berbeda dalam setiap suku yang memiliki koefisien bukan nol. Cipher yang memiliki kotak-S dengan derajat aljabar yang rendah, dapat mudah terserang *higher-order differential cryptanalysis*.

3. ANALISIS SANDI DIFERENSIAL TERHADAP DES LENGKAP

DES lengkap[6] memiliki 16 ronde. Setiap ronde terdiri dari ekspansi permutasi, kotak-S, pencampuran kunci dan permutasi. Sebenarnya setiap ronde dapat dipandang berisi tiga komponen utama, yaitu transformasi linear (ekspansi permutasi dan permutasi), pencampuran kunci dan transformasi tidak linear (kotak-S). Karena satu-satunya komponen tidak linear hanyalah kotak-S, maka Analisis Sandi Diferensial (ASD) dimulai dengan memeriksa kotak-S DES.

Pada [4] dijelaskan bahwa DES dapat dipecahkan dengan ASD. ASD merupakan metode pertama yang dapat memecahkan DES dengan waktu yang lebih singkat dari pada *brute force attack*.

Rasio antara jumlah pasangan benar dan jumlah rata-rata *subkey* yang tidak benar dalam proses perhitungan disebut sebagai *signal to noise ratio* (S/N). Untuk mendapatkan subkey yang benar, dilakukan proses penghitungan dan pasangan benar dalam jumlah yang cukup besar. Jumlah pasangan yang diperlukan tergantung pada peluang karakteristik p, jumlah bit subkey yang dihitung serentak,k, jumlah rata-rata α per pasangan yang dianalisis (tidak termasuk pasangan salah yang dapat dibuang sebelum proses penghitungan), dan fraksi β, pasangan yang dianalisis di antara seluruh pasang yang ada. Penghitung berisi nilai rata-rata $\frac{m.\alpha.\beta}{2^k}$ di mana m adalah jumlah pasangan yang disediakan. Nilai subkey yang benar dihitung sekitar m.p kali dengan pasangan yang tepat, sedangkan pasangan salah akan menghasilkan jumlah perhitungan yang cukup kecil. Rasio *signal to noise* dihitung sebagai:

$$\frac{S}{N} = \frac{m.p}{\frac{m.\alpha.\beta}{2^k}} = \frac{2^k.p}{\alpha.\beta} \quad (9)$$

Konsekuensi dari rumus ini adalah bahwa S/N tidak tergantung dari jumlah pasangan m, yang digunakan dalam *attack*. Biham dan Shamir[4] menunjukkan secara eksperimen bahwa bila S/N jauh di atas 1, maka hanya sedikit pasangan *plaintext* yang diperlukan untuk memperoleh *subkey* DES. Dan bila S/N ≤ 1, maka diperlukan jumlah pasangan yang sangat banyak. Ketika sedang memeriksa ASD pada IDEA yang disederhanakan, Borst[2] memperoleh hasil, bahwa ketika S/N > 1, maka nilai subkey sangat *disarankan* oleh karakteristik, sedangkan ketika S/N < 1 menunjukkan bahwa *subkey* yang tepat sangat tidak *disarankan* oleh karakteristik. Jadi kesimpulannya adalah, bila S/N = 1, maka *subkey* yang benar tidak dapat dibedakan dari *subkey* yang salah.

Tabel 1. memperlihatkan hasil awal ASD pada DES. Dari tabel tersebut diketahui bahwa DES dianalisis secara bertahap. Mula-mula diusahakan untuk memecahkan DES 4 Ronde, kemudian 6 ronde dan seterusnya hingga akhirnya diperoleh 16 ronde lengkap. Untuk mendapatkan kunci DES 4 Ronde hanya perlukan 16 *ciphertext*. DES 6 ronde dapat dipecahkan dengan menggunakan 240 ciphertext dalam waktu beberapa milidetik jika digunakan pentium 4. DES 15 ronde dapat

ISBN: 979-756-061-6

dipecahkan lebih cepat dari pada *brute force attack*, namun DES 16 ronde memerlukan waktu lebih lama daripada *brute force attack*.

Tabel 1. Hasil ASD awal pada DES

Jumlah Ronde	***Kompleksitas***
4	2^4
6	2^8
8	2^{16}
9	2^{26}
10	2^{35}
11	2^{36}
12	2^{43}
13	2^{44}
14	2^{51}
15	2^{52}
16	2^{58}

Sekarang akan dijelaskan cara ASD terhadap DES 16 ronde yang lebih cepat dari pada brute force attack[5]. Pertama-tama dibuat terlebih dahulu tabel distribusi untuk seluruh delapan kotak subtitusi DES, sesuai dengan persamaan (5). Contoh sebagian tabel distribusi ini diperlihatkan pada tabel 2. Isian ke bawah merupakan Δx dan isian ke kanan Δy.

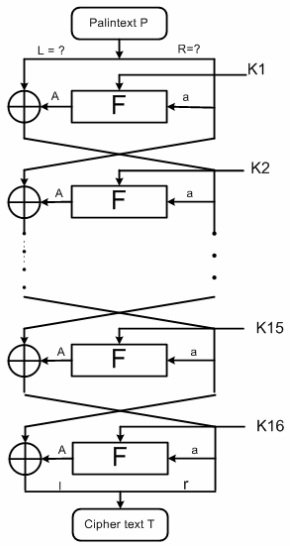
Tabel 2. Contoh Tabel distribusi kotak S1 DES

	0	***1***	***2***	***3***	***4***	***5***	***6***
0	64	0	0	0	0	0	0
1	0	0	0	6	0	2	4
2	0	0	0	8	0	4	4
3	14	4	2	2	10	6	4
4	0	0	0	6	0	10	10
5	4	8	6	2	2	4	4
6	0	4	2	4	8	2	6
7	2	4	10	4	0	4	8
8	0	0	0	12	0	8	8
9	10	2	4	0	2	4	6
10	0	8	6	2	2	8	6
11	2	4	0	10	2	2	4

Dari tabel 2 dapat dilihat bahwa bila beda masukan kotak S1 DES adalah $\Delta x = 0$, makan $\Delta y = 0$ juga dengan peluang p=1. Bila beda masukan = 0011_2, maka beda keluaran = 0000_2 dengan peluang p= 14/16. Sifat ini (mengeksploitasi $\Delta y=0$) akan sering digunakan dalam ASD *full* DES 16 ronde. Ini disebabkan mudahnya analisis terhadap karakteristik yang memiliki sifat ini.

Karakteristik yang sangat penting untuk ASD adalah karakteristik iteratif. Dengan karakteristik ini maka suatu karakteristik digunakan berulang-ulang hingga DES lengkap. Suatu karakteristik $\Omega(\Omega_P, \Omega_\Lambda, \Omega_T)$ disebut karakteristik iteratif bila pertukaran dua bagian dari setengah Ω_P akan menghasilkan Ω_T.

Prinsip pencarian karakteristik terbesar DES adalah mencari karateristik iteratif terbesar sehingga membentuk:



Gambar 1. Karakteristik iteratif 1R

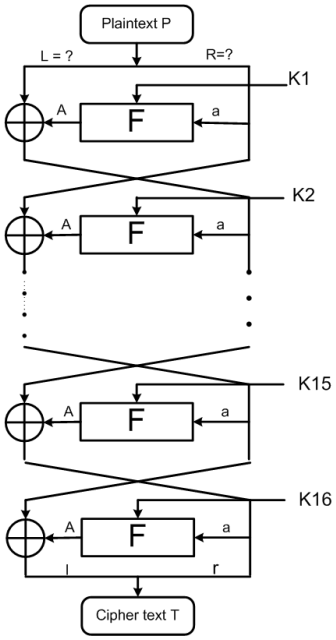
Gambar 1 memperlihatkan karakteristik iteratif 1 ronde a→A yang diulang dari ronde 1 sampai ronde 16. Jadi, nilai L dan R harus dicari agar menghasilkan karakteristik seperti pada gambar tersebut. Gambar 2 memperlihatkan karakteristik iteratif 2 Ronde.

Pada gambar 2, perlu dicari karakteristik L dan R agar dihasilkan a→A pada ronde ganjil, dan b→B pada ronde genap. Demikian pula karakteristik iteratif untuk 3 ronde dan seterusnya memiliki pola yang serupa.

Cara yang paling mudah adalah dengan mencari di mana nilai A atau B nya = 0. Misalkan untuk karakteristik 1 ronde, a=A=0. Dan karakteristik 2 rondenya menjadi a=60 00 00 00_{hex} , A= 00 80 82 00_{hex}, b= 0 dan B = 0.Ini berarti L= 00 80 82 00_{hex} dan R=60 00 00 00_{hex}. Dengan cara yang sama akan diperoleh karakteristik iteratif 3 Ronde yaitu:

L= 00 80 82 00_{hex}	R=60 00 00 00_{hex}
A = 00 80 82 00_{hex}	a = 60 00 00 00_{hex}
B = 0	b = 0
C = 00 80 82 00_{hex}	c = 60 00 00 00_{hex}
l = 00 80 82 00_{hex}	r = 60 00 00 00_{hex}

ISBN: 979-756-061-6



Gambar 2. Karakteristik iteratif 2R

Cara yang paling sederhana untuk mendapatkan karakteristik iteratif adalah dengan memeriksa satu persatu seluruh karakteristik yang mungkin terjadi. Untuk DES 16 ronde, salah satu karakteristik terbaik adalah untuk 2 ronde sebagai berikut:

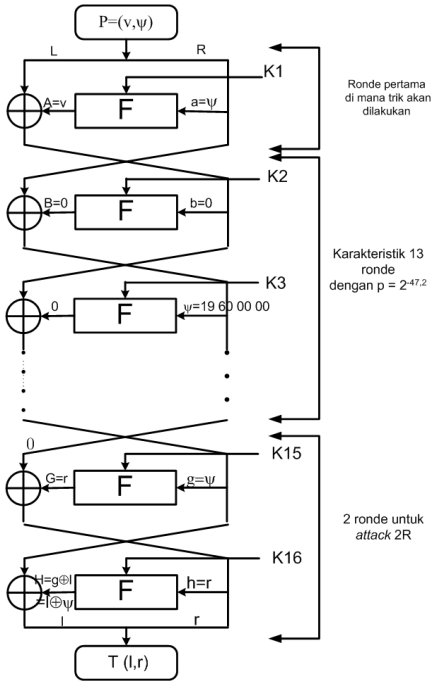
Ψ = 19 60 00 00$_{hex}$	00 00 00 00
A = 0	a = 0
B = 0	b = ψ
0	Ψ

dengan peluang p = 1/234 yang diperoleh dari
(Δx,Δy) pada S1 = (3, 0) p = 14/64
(Δx,Δy) pada S2 = (32$_{hex}$, 0) p = 8/64
(Δx,Δy) pada S3 = (2C$_{hex}$, 0) p = 10/64
(Δx,Δy) pada S lainnya = (0,0) p = 1
Sehingga p karakteristik = $(14/64)(8/64)(10/64)(1)^5 \approx 1/234$
Karena peluang kotak S4-S8 =1 maka terdapat pangkat 5 pada $(1)^5$.

Bila kita gunakan karakteristik untuk DES 16 ronde, maka akan diperoleh $p = (1/234)^7 = 2^{-55,1}$ bila digunakan *attack* 2R (Pangkat 7 muncul karena peluang pada ronde ganjil = 1 dan pada ronde genap=1/234). Attack 2R berarti kita menggunakan (16-**2**)= 14 ronde DES untuk mendapatkan subkey pada ronde ke-16. Artinya diperlukan sedikitnya 2^{55} ciphertext untuk mendapatkan *subkey* DES. Ini setara dengan waktu rata-rata yang diperlukan *brute force attack* yaitu ½ x 2^{56} = 2^{55} yang hanya membutuhkan satu atau dua *ciphertext*. Untuk kondisi ini tentunya ASD tidak akan dipilih.

Karena itu muncullah metode baru untuk mengurangi kompleksitas ASD seperti pada [5]. Konsepnya adalah berusaha tetap menggunakan 6 ronde yang masing-masing memiliki peluang karakteristik = 1/234 sehingga jumlah *plaintext* yang harus dipilih adalah $234^6 \approx 2^{47,222}$. Karakteristik yang digunakan tetap seperti sebelumnya yaitu (ψ,0). Namun karakteristik ini tidak ditempatkan pada ronde pertama, melainkan diletakkan pada ronde kedua. Sehingga ronde yang memiliki p=1/234 adalah ronde 3,5,7,9,11 dan 13 (6 ronde yang digunakan dalam ASD dengan DP = 1/234); sedangkan ronde 14 memiliki p=1. Jadi di sini digunakan attack 2R.

Untuk menghilangkan pengaruh DP pada ronde pertama digunakan trik sebagai berikut:



Gambar 3. Attack pada 16 ronde

Dari ronde ke-2 hingga ronde ke-14, diberikan karakteristik iteratif

$$0 \leftarrow 0$$
$$0 \leftarrow 19\ 60\ 00\ 00$$

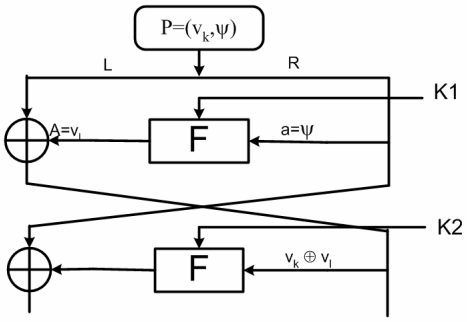
Sehingga dari sini diperoleh $DP = (1/234)^6 \approx 2^{-47,2}$
Pada ronde pertama tidak diberikan karakteristik 0←19 60 00 00 karena akan mengakibatkan DP meningkat menjadi $2^{-55,1}$ seperti hasil yang lalu. Ronde 15 dan 16 tidak ditetapkan nilai DP-nya karena akan dilakukan attack 2R. Sehingga diasumsikan nilai G=r, meskipun g=ψ dapat menghasilkan G=0 dengan DP=1/234.

Langkah berikutnya adalah mencari v pada P(v,ψ) sehingga masukan pada ronde kedua memiliki karakteristik (ψ,0). Pada ASD konvensional, 2N *plaintext* akan membentuk N

ISBN: 979-756-061-6

pasang *plaintext*, dan *attacker* akan memperoleh sekitar p x N pasang *ciphertext* yang memiliki XOR setelah satu ronde berupa (ψ,0). Dan p adalah peluang karakteristik ronde pertama. Ini dapat diperbaiki dengan dengan memilih *plaintext* secara lebih hati-hati.

Misalkan | {v |v←ψ} | = n adalah jumlah kemungkinan beda masukan ψ akan menghasilkan beda keluaran v. Nilai ini diberi nomor v_1, ...v_n. Kita pilih n *plaintext* dalam bentuk dalam bentuk ($P^L \oplus v_j \mid P^R \oplus \psi$) dan n *plaintext* dalam bentuk ($P^L \oplus v_i \mid P^R$). Sehingga pasangan *plaintext* ini memiliki karakteristik (v,ψ). Dari 2n *plaintext* ini bisa dipasangkan n^2 *plaintext* yang memiliki karakteristik dalam bentuk seperti Gambar 4. Namun hanya terdapat n pasang dimana $v_l = v_k$. Ini berarti diperoleh n pasang dari 2n pasang *plaintext*.



Gambar 4. Trik ronde pertama

Untuk DES, n memiliki nilai 2^{12} karena keluaran v hanya diharapkan mengandung bit „1“ pada keluaran kotak S1,S2 dan S3, sebanyak 3x4=12 bit, sedangkan keluaran kotak-S lainnya harus nol.

Masalah yang muncul adalah karena nilai v yang sesungguhnya tidak diketahui, demikian pula 2^{12} pasangan plaintext yang sesuai. Mencoba 2^{24} pasang yang mungkin akan memakan waktu sangat lama. Namun kita dapat memfilter pasangan-pasangan ini untuk mendapatkan pasangan benar. Keluaran kotak-S4 sampai S8 pada ronde 15 harus nol (20 bit). Pasangan *plaintext* yang tidak memiliki nilai nol pada posisi ini akan dianggap pasangan salah. Karena setiap pasang dari 2^{24} pasang yang mungkin akan lolos tes dengan peluang 2^{-20}, maka diharapkan 2^4 =16 pasang akan lolos tes. Dengan memeriksa kotak-S tambahan pada ronde pertama, ke-15 dan ke-16 serta membuang pasangan yang memiliki nilai XOR yang tidak mungkin terjadi pada keluaran kotak-S, sekitar 92,55% pasangan dapat dibuang, dan meninggalkan 16x0,0745 = 1,19 pasang per struktur. Nilai XOR pasangan benar dari masukan kotak-S pada ronde pertama dan ke-15 diketahui dan kita gunakan fraksi tidak nol dari tabel distribusi yang memiliki nilai 14/16, 13/16 dan 15/16 dan kita tidak menggunakan fraksi total XOR keluaran kotak-S yang mungkin (sekitar 80%). Sehingga pasangan yang tetap bertahan adalah $\left(\frac{14}{16}\frac{13}{16}\frac{15}{16}\right)^2 \text{ x } 0{,}8^8 = 0{,}074529$.

Setiap kunci yang disarankan akan dites. Kunci yang disarankan adalah kunci yang menghasilkan XOR pasangan keluaran yang diharapkan pada ronde pertama, ronde 15 dan ronde terakhir. Pada ronde pertama dan ke-15, masukan XOR S4-S8 selalu nol. Dan bila kita ikuti ekspansi kunci DES (*key scheduling*), terlihat bahwa ke-28 bit pada register-kunci kiri digunakan sebagai masukan kotak-S S1,S2 dan S3 pada ronde pertama dan ke-15, dan masukan S1-S4 pada ronde 16. Sedangkan 24 bit register kunci kanan digunakan pada pada ronde 16. Artinya terdapat 28+24 bit = 52 bit kunci yang memasuki kotak-kotak S tersebut. ($2^{-32}/0{,}8^8$) pilihan dari nilai 52 bit tetap bertahan dengan membandingkan XOR keluaran ronde terakhir dengan nilai yang diharapkan, dan membuang XOR keluaran kotak-S yang tidak mungkin, sedangkan $\frac{2^{-12}}{\frac{14}{16}\frac{13}{16}\frac{15}{16}}$ yang tersisa akan tetap bertahan dengan membandingkan XOR keluaran 3 kotak-S pada ronde pertama dengan nilai yang diharapkan. Setiap pasangan yang dianalisis menyarankan sekitar 2^{52} x $\frac{2^{-32}}{0,8^8} x \left(\frac{2^{-12}}{\frac{14}{16}\frac{13}{16}\frac{15}{16}}\right)^2$ = 0,84 nilai untuk 52 bit kunci. Dan masing-masing berkaitan dengan 16 nilai yang mungkin dari kunci 56 bit. Karena itu, setiap struktur menyarankan sekitar 1,19x0,84x16= 16 pilihan kunci. Dengan membuang 2 ronde terakhir, kita dapat memeriksa setiap kunci yang disarankan dengan menjalankan sekitar seperempat enkripsi DES dan meninggalkan sekitar 2^{-12} pilihan kunci. Ini memfilter sekitar 16x ¼ =4 operasi DES ekivalen. Setiap pilihan yang tersisa dari kunci 56 bit diperiksa melalui enkripsi satu plaintext dan membandingkan hasilnya dengan ciphertext yang seharusnya. Jika pemeriksaan ini berhasil, terdapat peluang yang sangat tinggi bahwa kunci tersebut memang kunci yang benar.

4. KESIMPULAN

ASD dapat memecahkan DES lengkap 16 ronde dengan jumlah percobaan enkripsi yang lebih sedikit dari *brute force attack* (mencoba satu per satu seluruh 2^{56} kunci DES). Meskipun hal ini tidak praktis, (karena membutuhkan jumlah plaintext yang sangat besar. yaitu 2^{47} *plaintext* yang dipilih), namun hal ini tetap dianggap sebagai pemecahan DES oleh kalangan akademisi. Sedangkan *brute force attack* hanya membutuhkan dua atau tiga *plaintext* yang

ISBN: 979-756-061-6

diketahui untuk memecahkan kunci DES, namun membutuhkan enkripsi rata-rata 2^{55} kali.

Plaintext yang *dipilih* berarti bahwa kita harus *memilih* pasangan plaintext yang kita inginkan yang kita peroleh dari saluran komunikasi, dan kita harus mengetahui pula pasangan ciphertextnya. Dan ini membutuhkan usaha yang tidak ringan dalam dunia nyata, bila kita harus mendapatkan 2^{47} *plaintext*. Sedangkan untuk *mengetahui* dua atau tiga *plaintext* beserta pasangan ciphertext-nya membutuhkan usaha yang jauh lebih ringan.

Oleh karena itu, meskipun ASD ini dianggap mampu memecahkan DES, DES tetap digunakan secara luas selama bertahun-tahun sejak keberhasilan ASD terhadap DES di tahun 1993. Dan setelah AES ditetapkan sebagai standar pengganti DES pada tahun 2001, Triple DES masih direkomendasikan untuk digunakan.

Daftar Pustaka

[1] Keliher,L. *Linear Cryptanalysis of Substitution-Permutation Networks.* Ph.D Thesis, Queen's University, Canada.2003

[2] J. Borst, L.R. Knudsen, and V. Rijmen. *Two Attacks on Reduced IDEA*. In W. Fumy, editor, Advances in Cryptology, Eurocrypt '97, LNCS 1233, pages 1-13. Springer-Verlag, 1997

[3] Knudsen, L.R. *Block Ciphers- Analysis, Design, and Applications*, Ph.D Thesis, Computer Science department,Aarhus University, 1994,

[4] E. Biham, A. Shamir, *Differential Cryptanalysis of DES-like Cryptosystems*, In Advances in Cryptology : CRYPTO '90, pages 2-21. Springer Verlag.1991

[5] E. Biham and A. Shamir, *Differential cryptanalysis of the full 16-round DES*, Advances in Cryptology-CRYPTO'92, LNCS 740, pp. 487-496, Springer-Verlag,1993.

[6] Man Young Rhee, "*Cryptography and Secure Communications*", textbook, Mc Graw-Hill (1994)