SNATI
Seminar Nasional
Aplikasi Teknologi Informasi 2004
Yogyakarta, 19 Juni 2004

Analisis Sandi Algoritma Enkripsi Subtitusi Permutasi 4 Ronde

Yusuf Kurniawan
Teknik Informatika Universitas Pasundan
Jl. Setiabudi 193 Bandung 40153.
Telp. (022) 2019371; Fax. (022)2019352
e-mail: ysfk2002@yahoo.com

Abstract

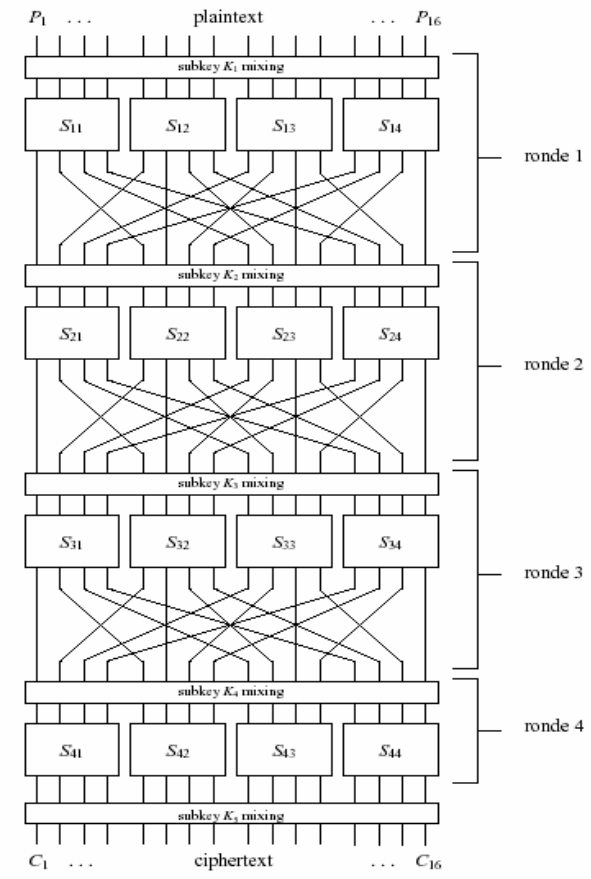
Encryption is one of the most important components in security of application of information technology. We need the good knowledge of encryption attack, to create the strong, rapid, and flexible encryption algorithm. This research shows that security of encryption algorithm does not only depend on the bits number of key, but also depend on the structure of algorithm itself. The more complicated the algorithm is (it is hoped more secure) the slower the algorithm runs, so the algorithm is not comfortable to be used. This research includes cryptanalysis and programming to attack the algorithm of simple encryption, 80-bit key, which can be broken theoretically in 19 years using 1 million computers, with brute force attack. In this research, this algorithm can be broken less than one minute with 10 thousand characters of known plain text with one personal computer, although the algorithm has good statistical property.

Keywords: *encryption, cryptanalysis, brute force attack, statistical property, known plain text*

1. Enkripsi Jaringan Subtitusi Permutasi Sederhana

Pada makalah ini akan diteliti apakah kriteria yang ditetapkan pada sebuah algoritma enkripsi yang berkualitas tinggi sudah dipenuhi algoritma atau tidak. Selanjutnya penelitian akan ditekankan pada bagian kotak subtitusi yang merupakan satu-satunya bagian tidak linear dari algoritma. Dari penelitian ini akan dibuktikan bahwa keamanan algoritma tidak hanya ditentukan oleh banyaknya bit kunci saja, melainkan sangat tergantung pada struktur internal algoritmanya sendiri.

Pada penelitian ini dibahas analisis algoritma enkripsi subtitusi-permutasi (SPN atau *Subtitution Permutation Network*) sederhana dengan kunci 80 bit (gambar 1). Semua analisis sandi modern selalu dimulai dari analisis sandi yang sederhana dan kemudian dilanjutkan pada algoritma yang lebih komplek.



Gambar 1. SPN 4 Ronde

Pertama, memeriksa apakah keluaran algoritma SPN 4 ronde ini memenuhi kriteria statistik algoritma yang ideal : setiap perubahan sebarang 1 bit masukan akan mengubah rata-rata setengah bit-bit keluaran, dan setiap perubahan sebarang 1 bit masukan akan memberikan peluang perubahan setiap bit keluaran sebesar 0,5.

Kedua, menggunakan analisis sandi linear untuk mencari pendekatan linear terhadap fungsi algoritma enkripsi yang tidak linear. Pada pendekatan linear ini, algoritma dianggap cenderung menghasilkan nilai bit "0" atau "1". Dengan kata lain, keluaran algoritma enkripsi dianggap tidak acak, sehingga dimungkinkan untuk mendapatkan kuncinya bila diketahui sejumlah besar pasangan plaintext/ciphertext yang dienkrip dengan kunci yang sama.

2. Analisis Sandi

C Shannon menyarankan agar sistem enkripsi seharusnya mengimplementasikan prinsip *confusion* dan *diffusion*. *Confusion* menghilangkan hubungan antara elemen masukan (plaintext) dan keluaran (ciphertext). Cara termudah adalah dengan subtitusi. Sedangkan *diffusion* menyebarkan pengaruh elemen *plaintext* ke seluruh *ciphertext*. Cara paling mudah adalah dengan permutasi(transposisi). Namun, menurut J Daemen (Daemen1997), permutasi tidak banyak memberikan keamanan terhadap algoritma enkripsi karena memiliki sifat yang linear. Sehingga penelitian kotak-S (Subtitusi) mendapat perhatian yang sangat besar.

Sifat algoritma enkripsi yang penting adalah *Avalanche property* (AVAL) yang merupakan kelanjutan *diffusion*. Sifat ini menyatakan bahwa perubahan satu bit masukan akan mengubah rata-rata setengah dari bit-bit keluaran. Sifat lanjutannya adalah *Strict Avalanche Criterion* (SAC) yang menyatakan bahwa peluang setiap bit keluaran adalah sebesar 0,5 bila terdapat perubahan sebarang 1 bit masukan.

2.1 Prinsip Analisis Sandi Linear

Tujuan Analisis Sandi Linear [Matsui1998] adalah untuk mendapatkan ekspresi linear dari algoritma enkripsi:

$$\mathbf{P[i_1, i_2, \ldots. i_a] \oplus C[j_1, j_2, \ldots j_b] = K[k_1, k_2, \ldots k_c]} \quad \text{..... (1)}$$

di mana $i_1, i_2, \ldots. i_a, j_1, j_2, \ldots j_b$ dan $k_1, k_2, \ldots k_c$ menyatakan posisi bit tertentu. Persamaan (1) akan dipenuhi dengan peluang $p \neq ½$ untuk pasangan *plaintext/ciphertext* acak. *Magnitude* $|p-1/2|$ menunjukkan efektivitas persamaan (1). Bila persamaan ini telah kita temukan, maka dimungkinkan untuk mendapatkan satu bit kunci $K[k_1, k_2, \ldots k_c]$ dengan menggunakan algoritma yang didasarkan pada metode peluang terbesar maksimum:

Algoritma 1:

Langkah 1: Anggap T sebagai jumlah plaintext sedemikian sehingga ruas kiri persamaan (1) sama dengan nol.

Langkah 2:

If T > N/2 (N menyatakan jumlah plaintext)

then terka $K[k_1, k_2, \ldots k_c] = 0$ (jika $p > ½$) atau 1 (jika $p < ½$)

else terka $K[k_1, k_2, \ldots k_c] = 1$ (jika $p > ½$) atau 0 (jika $p < ½$)

Peluang [Matsui1998] bahwa $P[i_1, i_2, \ldots. i_a] = 0$ adalah:

$$½ + \mathbf{2^{n-1}} \prod_{i=1}^{n} (p_i - 1/2) \quad \text{..... (2)}$$

2.2 Kriteria Kotak-S

Kotak-S nxn memetakan (Vergili2001) fungsi $f: \{0,1\}^n \rightarrow \{0,1\}^n$, yang memetakan *string* masukan n-bit, **X**, menjadi string keluaran, **Y**, di mana **Y**=f(**X**).

Bila e_i menyatakan vektor satuan dengan bit i = 1, sedangkan bit lainnya = 0, dan A^{e_i} adalah string beda keluaran, yang disebut juga dengan vektor *avalanche*, di mana hanya bit ke-i pada string masukan yang berubah, maka

$$A^{e_i} = \mathbf{f(X) \oplus f(X \oplus e_i) = [} a_1^{e_i}\ a_2^{e_i} \ldots .a_n^{e_i} \mathbf{]} \quad \text{..... (3)}$$

di mana $a_j^{e_i} \in \{0,1\}$

Kemudian, Kotak-S n x n dinyatakan memenuhi kriteria AVAL jika untuk semua i = 1,2,...,n:

$$\frac{1}{2^n} \sum_{j=1}^{n} W(a_j^{e_i}) = \frac{n}{2} \quad \text{..... (4)}$$

di mana $W(a_j^{e_i}) = \sum_{semua\ X \in \{0,1\}^n} a_j^{e_i}$ adalah total perubahan pada variabel *avalanche* ke-j, $a_j^{e_i}$ dihitung untuk semua alfabet masukan berukuran 2^n (Perhatikan bahwa $0 \leq W(a_j^{e_i}) \leq 2^n$).

Kriteria SAC diusulkan oleh Webster dan Tavares [Webster1986]. Kotak-S dianggap memenuhi SAC, bila

$$\frac{1}{2^n} W(a_j^{e_i}) = \frac{1}{2} \text{ untuk semua i,j} \quad \text{…… (5)}$$

3. Metode Penelitian

Analisis Sandi Linear menggunakan pendekatan *known plaintext attack*, yang artinya adalah, kita harus mendapatkan sejumlah *ciphertext* yang dapat kita tentukan *plaintext*nya.

Di sini akan dicari aproksimasi linear kotak-S (tabel 1) yang diadopsi dari kotak-S DES. Penelitian mengenai permutasi ditinggalkan, karena dianggap sebagai komponen yang linear.

Tabel 1. Masukan keluaran kotak-S

Masukan	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Keluaran	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

Teknik pencarian aproksimasi linear dengan cara mencoba semua kemungkinan yang ada. Karena kotak-S memiliki 4 bit masukan dan 4 bit keluaran, maka jumlah kemungkinan masukan dan keluaran sebesar 16x16. Hasilnya diperlihatkan pada tabel 2.

Misalkan, aproksimasi masukannya 11_{10} (1011_2), maka nilai ini dinyatakan dengan persamaan $X_1 \oplus X_3 \oplus X_4$. Demikian pula jika aproksimasi keluarannya 3_{10}(0011), maka nilai ini dinyatakan dengan $Y_3 \oplus Y_4$. Kemudian dicari aproksimasi pers(1) dengan ruas kanan dianggap nol. Jadi, dalam contoh ini, $X_1 \oplus X_3 \oplus X_4 = Y_3 \oplus Y_4$ digunakan untuk menghitung jumlah pasangan plaintext/ciphertext yang memenuhi persamaan tersebut.

Tabel 2 berisi bias pendekatan linear, atau menunjukkan bahwa bila kita melakukan pendekatan $X_2 = Y_2 \oplus Y_4$ (masukan=4 dan keluaran=5) terhadap kotak-S, akan diperoleh bahwa persamaan tersebut akan benar sebanyak 4 kali dari seluruh kemungkinan 16. Sehingga biasnya sebesar 4-8 = -4 (lihat tabel 2). Bias terbesar adalah +8, namun tidak dapat digunakan untuk melakukan pendekatan linear. Bias berikutnya adalah ±6. Namun jika kita gunakan untuk melakukan pendekatan linear, akan mengakibatkan jumlah kotak-S yang aktif meningkat, sehingga akan semakin sulit untuk mendapatkan kunci. Berikutnya, kita coba gunakan bias ±4 (±4/16):

Tabel 2. Bias pada analisis sandi linear

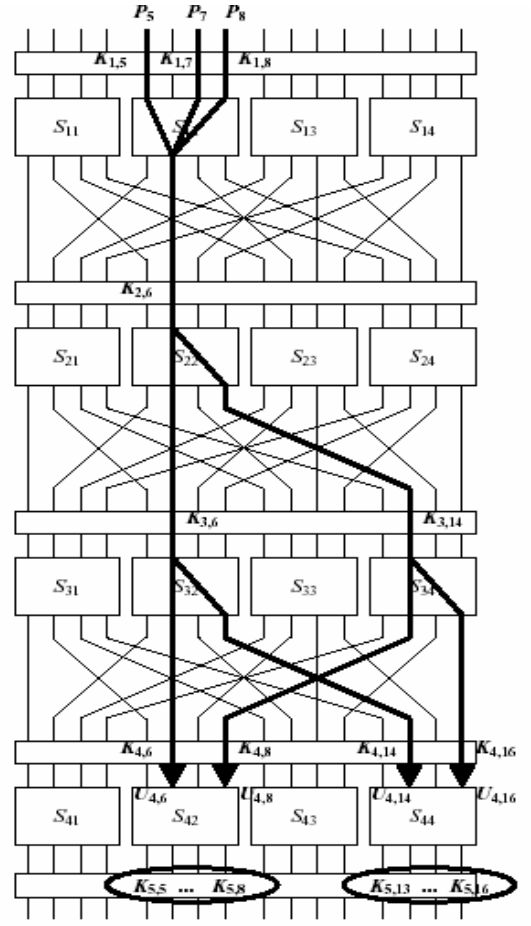
		Jumlah keluaran **0 1 2 3 4 5 6 7 8 9 A B C D E F**
Jumlah masukan	0	+8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
	1	0 0 -2 -2 0 0 -2 +6 +2 +2 0 0 +2 +2 0 0
	2	0 0 -2 -2 0 0 -2 -2 0 0 +2 +2 0 0 -6 +2
	3	0 0 0 0 0 0 0 0 +2 -6 -2 -2 +2 +2 -2 -2
	4	0 +2 0 -2 -2 -4 -2 0 0 -2 0 +2 +2 -4 +2 0
	5	0 -2 -2 0 -2 0 +4 +2 -2 0 -4 +2 0 -2 -2 0
	6	0 +2 -2 +4 +2 0 0 +2 0 -2 +2 +4 -2 0 0 -2
	7	0 -2 0 +2 +2 -4 +2 0 -2 0 +2 0 +4 +2 0 +2
	8	0 0 0 0 0 0 0 0 -2 +2 +2 -2 +2 -2 -2 -6
	9	0 0 -2 -2 0 0 -2 -2 -4 0 -2 +2 0 +4 +2 -2
	A	0 +4 -2 +2 -4 0 +2 -2 +2 +2 0 0 +2 +2 0 0
	B	0 +4 0 -4 +4 0 +4 0 0 0 0 0 0 0 0 0
	C	0 -2 +4 -2 -2 0 +2 0 +2 0 +2 +4 0 +2 0 -2
	D	0 +2 +2 0 -2 +4 0 +2 -4 -2 +2 0 +2 0 0 +2
	E	0 +2 +2 0 -2 -4 0 +2 -2 0 0 -2 -4 +2 -2 0
	F	0 -2 -4 -2 -2 0 +2 0 0 -2 +4 -2 -2 0 +2 0

Tabel 3. Masukan keluaran kotak-S

Masukan	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Keluaran	7	F	6	C	B	9	A	0	D	E	4	5	1	2	8	3

Kotak-S_{12} : $X_1 \oplus X_3 \oplus X_4 = Y_2$ bias +1/4
Kotak-S_{22} : $X_2 = Y_2 \oplus Y_4$ bias -1/4
Kotak-S_{32} : $X_2 = Y_2 \oplus Y_4$ bias -1/4
Kotak-S_{34} : $X_2 = Y_2 \oplus Y_4$ bias -1/4

Pendekatan ini memberi kotak-S yang aktif seperti pada gambar 2. Di sini akan mencari bit-bit kunci $K_{5,5}$ ...$K_{5,8}$ dan $K_{5,13}$ ...$K_{5,16}$ terlebih dahulu, sebelum kunci lainnya. Terlihat bahwa dengan analisis sandi di sini, kita dapat memisahkan hubungan sebagian kunci dengan lainnya, sehingga tidak kita perlukan brute force attack untuk 80 bit kunci secara bersamaan, melainkan cukup brute force attack untuk 16 bit saja.



Gambar 2. Kotak-S aktif

Dari gambar 2 disusun persamaan lengkap:

$U_{4,6} \oplus U_{4,8} \oplus U_{4,14} \oplus U_{4,16} \oplus P_5 \oplus P_7 \oplus P_8 \oplus \sum_k = 0$

di mana

$\sum k = K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \oplus K_{2,6} \oplus K_{3,6} \oplus K_{3,14} \oplus K_{4,6} \oplus K_{4,8} \oplus K_{4,14} \oplus K_{4,16}$

Dari (2) kita peroleh peluang lengkap p =

$\frac{1}{2}+2^3(3/4-1/2)(1/4-1/2)^3 = 15/32$ (yang memiliki bias -1/32)

Dalam penelitian ini, digunakan 20 ribu karakter 8 bit (sebagai plaintext), atau 10 ribu blok (1 blok = 16 bit) masukan. Kemudian dihitung jumlah plaintext/ciphertext yang memenuhi persamaan tersebut. Jika misalkan diperoleh 5337 blok yang memenuhi persamaan tersebut, maka bias yang terjadi adalah 337, sedangkan cipher yang ideal akan memiliki bias=0 yaitu 5000 blok akan memenuhi persamaan dan 5000 sisanya tidak memenuhi. Artinya, peluang kotak-S ideal adalah (5000/10000)=1/2 yang berarti bahwa, keluaran kotak-S memang mengacak masukannya dengan sempurna. Kunci yang bersesuaian dengan bias terbesar dianggap sebagai kunci yang benar.

Tabel 4. Bias pada analisis sandi linear dengan percobaan secara acak terhadap 10000 kotak-S

		Jumlah keluaran **0 1 2 3 4 5 6 7 8 9 A B C D E F**
Jumlah masukan	0	+8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
	1	0 0 0 0 0 0 0 0 0 0 0 0 -4 -4 +4 -4
	2	0 -4 -2 -2 0 0 +2 -2 -2 +2 0 0 +2 +2 0 -4
	3	0 -4 +2 +2 0 0 -2 +2 +2 -2 0 0 +2 +2 +4 0
	4	0 0 0 0 -8 0 0 0 0 0 0 0 0 0 0 0
	5	0 0 0 0 0 0 0 0 +4 +4 -4 +4 0 0 0 0
	6	0 0 -2 +2 0 +4 +2 +2 -2 -2 0 +4 +2 -2 0 0
	7	0 0 +2 -2 0 +4 -2 -2 -2 -2 -4 0 -2 +2 0 0
	8	0 0 -2 +2 0 0 +2 -2 -2 +2 0 0 -2 +2 +4 +4
	9	0 0 -6 -2 0 0 -2 +2 +2 -2 0 0 -2 +2 0 0
	A	0 -4 0 +4 0 0 0 0 0 0 0 0 -4 0 -4 0
	B	0 +4 0 +4 0 0 0 0 0 0 0 0 0 +4 0 -4
	C	0 0 -2 +2 0 0 +2 -2 +2 -2 -4 -4 +2 -2 0 0
	D	0 0 +2 -2 0 0 +6 +2 +2 -2 0 0 -2 +2 0 0
	E	0 0 0 0 0 +4 0 -4 +4 0 +4 0 0 0 0 0
	F	0 0 0 0 0 -4 0 -4 0 -4 0 +4 0 0 0 0

4. Hasil Penelitian

Tabel 2 dan 4 menunjukkan hasil bias yang berbeda bila digunakan kotak-S yang berbeda. Tabel 2 didasarkan pada tabel 1, sedangkan tabel 4 didasarkan kotak-S tabel 3. Kami memilih tabel 4 dari percobaan secara acak terhadap 10000 kotak-S, dan berusaha mendapatkan kotak-S yang memiliki tabel bias terkecil. Namun ternyata, hasilnya serupa dengan kotak-S yang pertama. Artinya, untuk meningkatkan keamanan algoritma enkripsi ini, tidak dapat dilakukan dengan sekedar mencari kotak-S yang memiliki bias terkecil.

Tabel 5. Jumlah bias hasil percobaan

Kunci	Jumlah Bias	Kunci	Jumlah Bias	Kunci	Jumlah Bias	Kunci	Jumlah Bias
95x3	339	9**42B**	174	C5xB3	375	95x3	354
9521	273	9721	174	55C3	368	15A3	353
9623	241	932F	171	6553	363	E5D3	353
9723	220	9722	169	1583	362	75F3	352
9625	208	9621	167	B523	362	4533	352
9B2F	205	952D	161	2513	359	A503	352
952F	199	9220	157	35A3	358	D5D3	351
9421	192	902F	150	35C3	355	25A3	350
9425	178	9029	147	A593	355	7573	350
9423	175	9520	140	8573	354	9573	345

Tabel 5 menunjukkan bahwa bias akan terbesar (di sekitar 339 dalam contoh ini) jika kunci terkaan kita benar (yaitu x5x3). Jadi dengan kunci 5 dan 3 benar, maka sebarang kunci lainnya (x), tidak akan berpengaruh. Artinya, percobaan ini membuktikan bahwa analisis sandi linear, secara praktis terbukti dapat menemukan sebagian kunci, tanpa terpengaruh

kunci lainnya. Untuk mendapatkan 16 bit kunci ini diperlukan waktu kurang dari 1 detik dengan Pentium 4 1,7 GHz. Dengan cara yang sama, maka seluruh kunci 80 bit dapat dihitung dalam waktu kurang dari 1 menit.

Dengan kotak-S seperti pada tabel 1, diperoleh bahwa nilai *avalanche property*-nya sebesar (7,92/16); mendekati nilai yang ideal yaitu ½. Sedangkan Nilai SAC nya juga cukup bagus yaitu sebesar 0,495 dari nilai ideal sebesar 0,5. Namun penelitian ini juga menunjukkan bahwa meskipun, nilai AVAL dan SAC nya cukup bagus, ternyata algoritma ini masih dapat ditembus dalam waktu kurang dari 1 menit. Sementara, untuk menembus sistem enkripsi 80 bit dengan 1 juta komputer yang masing-masing mampu mencoba 1 milyar enkripsi per detik, diperlukan waktu rata-rata sekitar 19 tahun dengan *brute force attack.*

5. Kesimpulan

Dari penelitian ini, dapat diambil kesimpulan sebagai berikut:

a. Jumlah bit kunci bukanlah satu-satunya penjamin keamanan algoritma enkripsi
b. AVAL dan SAC hanyalah penentu awal keamanan algoritma. Jika suatu algoritma lulus tes ini, bukan berarti algoritma telah aman. Namun jika tidak lulus, hampir bisa dipastikan, algoritma tersebut tidak aman
c. Untuk meningkatkan keamanan algoritma, dapat ditingkatkan jumlah ronde, namun peningkatan ini juga mengakibatkan semakin lambatnya implementasi algoritma.
d. Semakin banyak kotak-S yang aktif, semakin aman algoritma

6. Saran

Dari hasil penelitian ini, penulis menyarankan penelitian lanjutan sebagai berikut:

a. Penelitian efektifitas analisis sandi linear dengan jumlah ronde ditingkatkan.
b. Meneliti pencarian persamaan linear secara cepat terhadap algoritma enkripsi.
c. Meneliti teknik untuk meningkatkan jumlah kotak-S yang aktif
d. Meneliti teknik analisis sandi yang lain
e. Mempelajari struktur algoritma yang lain agar tahan terhadap berbagai analisis sandi, dan cepat digunakan untuk berbagai aplikasi.
f. Mencari kotak-S yang tahan terhadap berbagai analisis sandi

Daftar Pustaka

Daemen, J. (1997), *On the Design and Security of Block Ciphers,* Doctoral Dissertation.

Matsui, M. (1998).*Linear Cryptanalysis Method for DES Cipher,* Computer & Information System Lab. Mitsubishi.

Vergili, I. and D. Y. Melek(2001), *Avalanche and Bit Independence Properties for the Ensembles of Randomly Chosen nxn S-Boxes,* EE Department of METU

Webster, A.F. and Tavares, S.E. (1986), *On the design of s-boxes,* Advances in Cryptology:Proc.CRYPTO'85 Springer-Verlag,Berlin,pp.523-534.