Teknik Kriptanalisis Linier

Gede Serikastawan, Teddy Setiawan, dan Denny Ranova

Departemen Teknik Informatika
Institut Teknologi Bandung
Jalan Ganesha 10 Bandung 40132

E-mail : if11063@students.if.itb.ac.id, if11075@students.if.itb.ac.id, if11079@students.if.itb.ac.id

Abstrak

Banyak pihak yang tidak terotorisasi atau dengan kata lain tidak mempunyai kunci rahasia ingin mengetahui isi arsip yang sudah dienkripsi. Untuk selanjutnya pihak ini disebut sebagai kriptanalis. kriptanalis ini melakukan berbagai usaha dan cara yang mungkin untuk mengetahui *plaintext* dari *ciphertext* yang bersesuaian. Di dalam bidang kriptografi, setiap usaha atau cara untuk memecahkan *ciphertext* menjadi *plaintext* yang bersesuaian tanpa mengetahui kunci kunci dikategorikan sebagai serangan terhadap *ciphertext*, dan disebut sebagai kriptanalisis.

Kelebihan kriptanalisis linier adalah bisa diterapkan untuk menyerang sistem kriptografi dengan mode *cipher block*, namun tingkat efisiensi yang berbeda. Contoh penerapannya adalah pada penyerangan sistem kriptografi dengan menggunakan algoritma DES.

***Kata kunci**: kriptanalis, kriptanalisis, plaintext, ciphertext, kunci rahasia, serangan*

1. Pendahuluan

Secara umum kriptanalisis adalah setiap usaha atau percobaan untuk mencoba membaca pesan yang dienkripsi tanpa mengetahui kunci rahasia. Seperti kita ketahui bersama bahwa kriptanalisis ini berkaitan erat dengan serangan terhadap *ciphertext* dan keamanan algoritma kriptografi. Semakin kompleks serangan yang dilakukan pada proses kriptanalisis maka algoritma kriptografi tersebut semakin bagus, demikian juga sebaliknya, semakin kuat algoritma kriptografi yang digunakan maka tugas kriptanalis menjadi semakin berat.

[4)]Kompleksitas serangan dapat diukur dengan beberapa cara :

1. Kompleksitas data (data complexity)
   Jumlah data yang dibutuhkan sebagai masukan untuk serangan. Semakin banyak data yang dibutuhkan untuk melakukan serangan, berarti semakin bagus algoritma kriptografi tersebut.
2. Kompleksitas waktu (time complexity)
   Waktu yang dibutuhkan untuk melakukan serangan. Ini disebut juga faktor kerja (*work factor*). Semakin lama waktu yang dibutuhkan untuk untuk melakukan serangan, berarti semakin bagus algoritma kriptografi tersebut.
3. Kompleksitas ruang memori (space/storage complexity)
   Jumlah memori yang dibutuhkan untuk melakukan serangan. Semakin banyak memori yang yang dibutuhkan untuk untuk melakukan serangan,

berarti semakin bagus algoritma kriptografi tersebut. [4)]

Asumsi yang diambil dalam kriptanalisis ini adalah kriptanalis mengetahui algoritma kriptografi yang digunakan untuk mengenkripsi pesan, sehingga keamanan algoritma terletak sepenuhnya pada kunci. Hal ini berdasarkan prinsip *Kerckhoff* (1883) yang berbunyi : *Semua algoritma kriptografi harus publik; hanya kunci yang rahasia*. Jika keamanan algoritma kriptografi ditentukan dengan menjaga kerahasiaan algoritmanya maka algoritma tersebut disebut dengan algoritma **terbatas** (*restricted*). Pada zaman sekarang algoritma terbatas tidak cocok lagi.

[4)]Sebuah algoritma kriptografi dikatakan aman (*computationally secure*) bila ia memenuhi tiga kriteria berikut:

1. Persamaan matematis yang menggambarkan operasi algoritma kriptografi sangat kompleks sehingga algoritma tidak mungkin dipecahkan secara analitik.
2. Biaya untuk memecahkan cipherteks melampaui nilai informasi yang terkandung di dalam cipherteks tersebut.
3. Waktu yang diperlukan untuk memecahkan cipherteks melampaui lamanya waktu informasi tersebut harus dijaga kerahasiaannya. [4)]

Perkembangan kriptanalisis ini sejalan dengan perkembangan sistem kriptografi, mulai dari yang paling sederhana sampai yang paling kompleks. Begitu sistem kriptografi muncul pertama kali, kriptanalisis sudah ada. Hal ini tentu saja karena keinginan untuk mengetahui isi dari pesan yang terenkripsi tetapi tidak mengetahui kunci rahasia yang bersesuaian.

Selain semata-mata dilakukan untuk tujuan mengetahui isi pesan yang terenkripsi, kriptanalisis juga dilakukan untuk menguji ketangguhan suatu algoritma kriptografi yang baru dipublikasikan.

1.1 Kriptanalisis pada Kriptografi Klasik

1.1.1 Analisis Frekuensi

Salah satu teknik kriptanalisis awal yang sukses adalah menganalisis frekuensi kemunculan suatu huruf dari suatu bahasa tertentu. Sebagai contohnya pada bahasa Inggris, huruf ’E’ adalah yang paling sering muncul. Kemudian berturut–turut adalah huruf 'T', 'A', 'O' dan 'N'. Sedangkan huruf-huruf seperti 'Q', 'X', 'V' dan 'Z' relatif jarang digunakan.

Jika pesan dienkripsi dengan menggunakan teknik penyandian monoalphabetik, teknik kriptanalisis dapat memanfaatkan jumlah kemunculan huruf-huruf sebagai dasar yang kuat untuk menyatakan huruf pengganti dari *ciphertext*. Sebagai contoh, jika dalam suatu *ciphertext* huruf ’P’ adalah huruf yang paling sering muncul, maka dapat diasumsikan bahwa huruf penggantinya didalam *plaintext* adalah huruf ’E’.

1.1.2 Analisis Kontak

Kelakuan lain dari suatu bahasa adalah susunan kemunculan suatu huruf terhadap huruf lainnya. Sebagai contohnya pada bahasa Inggris, huruf ’T’ lebih sering diikuti oleh huruf ’H’ daripada oleh huruf lainnya, sedangkan huruf ’H’ paling sering diikuti oleh huruf ’E’.

1.1.3 Babbage-Kasiski

Teknik penyandian polyalphabetic Vigenère membawa pengaruh yang besar pada kriptografi, karena membuat analisis frekuensi dan analisis kontak tidak dapat diterapkan lagi. Saat itu dianggap teknik penyandian ini sebagai teknik yang tidak dapat dipatahkan.

Metode untuk memecahkan teknik Vigenère ditemukan oleh Charles Babbage dan Friedrich Kasiski. Prinsip utamanya adalah menganalisis frekuensi kemunculan tiap huruf yang ada pada *ciphertext* yang mempunyai huruf kunci yang sama. Sehingga jika dapat mengetahui panjang kuncinya, maka contohnya panjang kuncinya sebanyak 7 huruf, karakter-karakter pertama, kedelapan, kelimabelas, kedua puluh dua, dan seterusnya, karakter-karakter tersebut dapat dipisahkan dan dapat dilakukan analisis frekuensi.

Permasalahannya ketika mencari panjang kuncinya. Caranya adalah dengan menggunakan analisis kontak. Karena pada bahasa Inggris pasangan huruf 'TH' sering muncul dan diekripsi beberapa kali menggunakan kunci yang sama, panjang kuncinya dapat dihitung.

1.1.4 Kontribusi Kerckhoffs

Teknik yang sangat kuat untuk melakukan kripanalisis dapat dilakukan ketika mempunyai beberapa pesan yang dienkripsi dengan menggunakan kunci yang sama. Teknik ini bahkan dapat digunakan untuk memecahkan penyandian *one-time pad* (OTP). Prinsipnya adalah menggunakan analisis frekuensi, yaitu ketika dua pesan dienkripsi dengan kunci yang sama, maka huruf pertama dienkripsi dengan cara yang sama, demikian juga untuk huruf yang kedua dan seterusnya. Dan jika mempunyai pesan dalam jumlah yang banyak, dapat dibuat sebuah monoalphabetik sederhana yang mudah dipecahkan. Teknik ini biasa disebut dengan *superimposition*.

Kerckhoff juga mempunyai kontribusi lain pada bidang kriptanalisis. Dia mengemukakan enam buat point penting dalam penyandian di bidang militer, yaitu :

1. Sistem pasti dapat dipecahkan dalam kenyataanya, walaupun dalam teori tidak mungkin.
2. Kerumitan dari sistem tidak boleh menyusahkan pemakai.
3. Kunci seharusnya mudah untuk diingat tanpa perlu mencatatnya dan mudah untuk diubah-ubah.
4. Kriptogram seharusnya dapat dikirim lewat telegraf.
5. Penyandian dan dokumennya seharusnya dapat dibawa dan dioperasikan oleh satu orang.
6. Sistem seharusnya tidak membutuhkan pengetahuan tentang aturan-aturan yang panjang dan juga tidak mengandung tekanan mental.

Salah satu dari keenam prinsip ini dikenal sebagai 'Kerckhoffs Principle'. Prinsip ini menyatakan bahwa : "pada sistem kriptografi, keamanan sistem tidak boleh bergantung kepada metode penyandiannya". Untuk masa sekarang ini, prinsip ini adalah "keamanan sistem seharusnya terletak pada kerahasiaan kunci".

1.2 Kriptanalisis pada Kriptografi Modern

Asumsi Kerckhoffs telah diterapkan sepenuhnya pada kriptografi modern. Sekarang ini, para ahli di bidang ini selalu menyarankan menggunakan suatu teknik penyandian yang telah dipublikasikan dan

diteliti secara mendalam. Walaupun agak aneh, menyembunyikan algoritma penyandian dapat mengurangi tingkat keamanan dari sistem.

1.2.1 Ciphertext-only attacks

Pada tipe ini, penyerang mengetahui *ciphertext* yang dikirim dan juga mengetahui algoritma yang dipakai, tetapi tidak mengetahui *plaintext* maupun kuncinya. Dengan menganalisis algoritma yang dipakai, dapat dimungkinkan untuk mendapatkan jalan cepat (*shortcut*) untuk mendapatkan kunci dibandingkan mencoba-coba sampai menemukan kunci yang tepat (*brute-force attack*).

Seperti yang sudah diketahui bahwa algoritma ekripsi saat ini menggunakan kunci yang panjang, blok yang panjang, dan putaran yang banyak untuk meningkatkan kompleksitas. Hal ini menyebabkan teknik penyerangan *brute-force* memakan waktu yang banyak.

1.2.2 Known-plaintext attack

Penyerangan akan lebih mudah dilakukan jika dapat mengakses *ciphertext* dan *plaintext*-nya. Ini adalah metode yang umum digunakan untuk menyerang dengan memperhatikan struktur sebelumnya yang berhasil dipecahkan.

Sangatlah masuk akal ketika mengirimkan pesan terenkripsi pada jaringan radio militer dari komandan ke markas pusat menggunakan suatu teknik penyandian tertentu, dan dari komandan ke pemerintah menggunakan teknik penyandian yang lain. Jika salah satu teknik berhasil dipecahkan, penyerang dapat memakainya untuk menyerang teknik lainnya.

1.2.3 Chosen-plaintext attack

Teknik ini mirip dengan *known-plaintext attack*, hanya penyerang memilih *plaintext* terutama yang mempunyai karakteristik matematika tertentu.

1.2.4 Adaptive-plaintext attack

Teknik yang lebih handal daripada *chosen-plaintext attack*. Pemilihan *plaintext* tidak hanya didasarkan pada karakteristik matematikanya saja tetapi juga berdasarkan hasil dari percobaan penyerangan sebelumnya. Seorang penyerang menggunakan tipe *plaintext* yang berbeda untuk mengetahui hasil yang paling baik.

1.2.5 Differential cryptanalysis

Teknik ini diumumkan pada tahun 1990 oleh Bilham dan Shamir. Teknik ini adalah varian dari *chosen-plaintext attack*. Dua buah *plaintext* dipilih dengan perbedaan tertentu dan dienkripsi dengan menggunakan kunci yang sama. Kedua *ciphertext* dijadikan subjek analisis matematika yang dapat diarahkan untuk mendapatkan peluang tiap bit pada kunci. Jika diulang-ulang, dapat dicari dengan statistik apakah bit tertentu pada kunci adalah 1 atau 0.

1.2.6 Linear Cryptanalysis

Teknik ini adalah salah satu metode probabilistik yang menghitung hasil dari XOR antara *plaintext* dengan *ciphertext.* Metode ini termasuk ke dalam *known-plaintext attack*. DES telah terbukti tidak aman terhadap serangan semacam ini.

2. Kriptanalisis Linier (Linear Cryptanalysis)

Kriptanalisis Linier adalah metode yang handal yang diperkenalkan oleh Matsui pada tahun 1993. Metode ini termasuk ke dalam teknik penyerangan *known plaintext*, dimana penyerang (kriptanalis) mempelajari pendekatan-pendekatan linier (*linier approximations*) dari bit-bit *parity* pada *plaintext*, *cipherteks*, dan kunci rahasia. Dengan menemukan pendekatan yang mempunyai kemungkinan yang tinggi dan menghitung bit-bit *parity* dari *known plaintext* dan *ciphertext*, dapat diperkirakan bit *parity* dari kuncinya.

Secara umum kriptanalisis linier mempelajari hubungan linier secara statistik antara bit-bit *plaintext*, *ciphertext* dan kunci rahasia yang digunakan untuk mengenkripsi *plaintext* menjadi *ciphertext* tersebut. Hubungan ini selanjutnya digunakan untuk memprediksi nilai-nilai dari bit-bit kunci rahasia, ketika sejumlah *plaintext* dan *ciphertext* yang bersesuaian diketahui.

Seperti kita ketahui bersama bahwa hampir semua operasi pada algortima DES linier, kecuali S *boxes*, hal ini cukup untuk menurunkan hubungan linier dari S *boxes*. Hubungan ini bisa diturunkan untuk setiap S *box* dengan memilih sebuah *subset* bit-bit masukan dan bit-bit keluaran, untuk selanjutnya menghitung paritas (xor) dari bit-bit ini untuk setiap kemungkinan masukan S *box* dan kemudian menghitung jumlah masukan yang paritas *subset*-nya bernilai 0. Jika S *box* saling berkaitan dalam bit-bit dari *subset*, maka semua masukan harus mempunyai paritas 1. Biasanya sebuah *subset* akan mempunyai banyak masukan dengan paritas 0 dan banyak masukan dengan paritas 1. Jika jumlah masukan dengan paritas 0 semakin mendekati jumlah masukan dengan paritas 1 maka *subset* dikatakan semakin tidak linier. Berarti pada definisi ini, *subset* disebut linier, minimal jika dia mempunyai masukan yang setengahnya memiliki paritas 0 dan setengahnya lagi memiliki paritas 1.

Matsui telah menghitung jumlah dari paritas 0 untuk setiap 64 * 16 = 1024 kemungkinan *subset* bit-bit masukan dan keluaran untuk setiap S *box*. Untuk merepresentasikasikan kelinieran *subset*, Matsui mulai dari menghitung jumlah dari setengah masukan. Dengan cara ini, nilai 0 menunjukkan *subset* yang tidak linier, dan nilai absolut yang tinggi menunjukkan atau mendekati fakta bahwa *subset* tersebut linier. Tabel yang menggambarkan nilai-nilai ini untuk semua kemungkian *subset* dari sebuah S *box* disebut sebagai *linear approximation table*. Tabel 1 di bawah ini adalah adalah *linear approximation table* S5 dari algoritma DES. Kita bisa lihat dari tabel ini, bahwa 30% dari *entry* mempunyai nilai 0.

Nilai absolut yang tertinggi dalam Tabel 1 adalah -20 pada *entry* (10_x, F_x). Karena itu, hanya dalam 12 keluaran dari 64 masukan, paritas 4 bit-bit keluaran sama dengan nilai bit-bit masukan yang kedua. Untuk kemudian hal ini ditetapkan sebagai sebuah keperluan dalam kriteria perancangan algoritma DES. *Entry* khusus ini, yang merupakan *entry* yang paling linier dari semua S *boxes* dari algoritma DES, adalah satu diantara tiga *entries* yang digunakan dalam serangan Matsui.

Solusi Matsui adalah untuk menemukan ekspresi linier secara statistik yang terdiri dari paritas *subset* dari *plaintext*, *ciphertext* dan kunci rahasia, yang diturunkan melalui ekspresi yang sama dari berbagai putaran. Dengan demikian paritas beberapa *set* bit-bit

data adalah fungsi dari paritas beberapa *set* bit-bit data putaran sebelumnya dengan paritas beberapa bit-bit kunci rahasia. Matsui menetapkan bahwa, linierisasi putaran didasarkan pada linierisasi S *boxes*. Jika kita meng-XOR nilai yang sama dengan dua setengah kali data yang ada, kita akan memperoleh paritas yang sama dengan sebelum XOR dilakukan. Karena *subset* bit-bit masukan secara statistik linier dengan subset bit-bit keluaran, paritas data sesudah XOR selalu sama dengan paritas data sebelum di-XOR-kan dengan konstanta kunci rahasia khusus.

Probabilitas bahwa approksimasi dalam sebuah S *box* valid, ditentukan dengan jaraknya dari setengah (½), sebagai contoh probabilitas *entry* dengan nilai -20 adalah :

$$p' = ½ - 20/64 = 12/64.$$

Sebuah entry dengan nilai 0 mempunyai probabilitas ½ dan *entry* ini tidak dipakai saat menyerang sistem kriptografi. Sedangkan semua *entry* lain yang mempunyai nilai tidak 0 (positif/negatif) bisa digunakan untuk menyerang sistem kriptografi. Approksimasi yang diambil mungkin melibatkan lebih dari satu S *box*. Untuk approksimasi yang melibatkan *l* S *boxes*, probabilitas gabungannya adalah :

$$½ + p = ½ + 2^{l-1} * \prod_{i=1}^{l} pi \qquad (1)$$

dengan p_i menyatakan jarak probabilitas approksimasi S *box* yang ke-*i* dan *p* menyatakan probabilitas gabungan.

Jika approksimasi linier dengan probabilitas ½ + *p* diketahui oleh kriptanalis, kriptanalis dapat menyusun sebuah serangan yang membutuhkan pengetahuan terhadap sekitar p^{-2} *plaintext*, *plaintext* ini bisa dipilih secara acak, tetapi semuanya harus dienkrepsi dengan menggunakan kunci rahasia yang sama.

Anggap $A[i]$ adalah bit ke-*i* dari *A* dan $A[i_1, i_2,\ldots, i_k]$ adalah bit parity $A[i_1] \oplus A[i_2] \oplus \ldots \oplus A[i_k]$. Untuk operasi linier yang sederhana seperti XOR dengan kunci atau permutasi dari bit-bit, ekspresi linier sederhana dapat dirumuskan dengan probabilitas=1. Untuk elemen yang tidak linier seperti S-box, pencarian pendekatan linier dengan probabilitas=*p* dilakukan deviasi maksimum $|p-½|$. Pendekatan untuk satu operasi penyandian dapat kemudian digabungkan dengan pendekatan untuk satu ronde/putaran dari sebuah *ciphertext*. Untuk keseluruhannya adalah :

$$P[i_1, i_2,\ldots, i_a] \oplus C[j_1, j_2,\ldots, j_b] = K[k_1, k_2,\ldots, k_c] \qquad (1)$$

(dimana $i_1, i_2,\ldots, i_a, j_1, j_2,\ldots, j_b$, dan $k_1, k_2,\ldots, k_c$ menyatakan lokasi-lokasi bit).

Prinsip dasar kriptanalisis linier adalah bagaimana menemukan satu bit kunci rahasia, yang mana merupakan paritas *subset* bit-bit kunci rahasia. Teknik tambahan untuk mengurangi banyaknya approksimasi yang diperlukan pada putaran adalah dengan menghilangkan putaran yang pertama dan/atau yang terakhir. Kemudian menghitung bit-bit kunci rahasia yang mempengaruhi data pada putaran-putaran yang tidak ada pada approksimasi. Hal ini juga mengurangi jumlah *plaintext* yang dibutuhkan serta meningkatkan jumlah bit-bit kunci rahasia yang ditemukan saat melakukan kriptanalisis.

Input	Output subset															
subset	0_x	1_x	2_x	3_x	4_x	5_x	6_x	7_x	8_x	9_x	A_x	B_x	C_x	D_x	E_x	F_x
0_x	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1_x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2_x	0	4	-2	2	-2	2	-4	0	4	0	2	-2	2	-2	0	-4
3_x	0	0	-2	6	-2	-2	4	-4	0	0	-2	6	-2	-2	4	-4
4_x	0	2	-2	0	0	2	-2	0	0	2	2	4	-4	-2	-2	0
5_x	0	2	2	-4	0	10	-6	-4	0	2	-10	0	4	-2	2	4
6_x	0	-2	-4	-6	-2	-4	2	0	0	-2	0	-2	-6	-8	2	0
7_x	0	2	0	2	-2	8	6	0	-4	6	0	-6	-2	0	-6	-4
8_x	0	0	2	6	0	0	-2	-6	-2	2	4	-12	2	6	-4	4
9_x	0	-4	6	-2	0	-4	-6	-6	6	-2	0	-4	2	-6	-8	-4
A_x	0	4	0	0	-2	-6	2	2	2	2	-2	2	4	-4	-4	0
B_x	0	4	4	4	6	2	-2	-2	-2	-2	-2	2	0	-8	-4	0
C_x	0	2	0	-2	0	2	4	10	-2	4	-2	-8	-2	4	-6	-4
D_x	0	6	0	2	0	-2	4	-10	-2	0	-2	4	-2	8	-6	0
E_x	0	-2	-2	0	-2	4	0	2	-2	0	4	2	-4	6	-2	-4
F_x	0	-2	-2	8	6	4	0	2	2	4	8	-2	8	-6	2	0
10_x	0	2	-2	0	0	-2	-6	-8	0	-2	-2	-4	0	2	10	-20
11_x	0	2	-2	0	4	2	-2	-4	4	2	2	0	-8	-6	2	4
12_x	0	-2	0	-2	2	-4	-2	-8	4	6	4	6	-2	4	-6	0
13_x	0	-6	0	2	-2	4	2	0	4	-6	4	2	-6	4	-2	0
14_x	0	4	-4	0	0	0	0	0	-4	-4	4	4	0	4	-4	0
15_x	0	4	0	-4	-4	4	-8	-8	0	0	-4	4	8	4	0	4
16_x	0	0	6	6	2	-2	4	0	4	0	6	2	2	2	0	0
17_x	0	4	-6	-2	6	-2	-4	4	4	-4	-6	2	-2	2	0	4
18_x	0	6	0	2	4	-10	-4	2	2	0	-2	0	2	4	-2	-4
19_x	0	2	4	-6	0	-2	4	-2	6	8	6	4	10	0	2	-4
$1A_x$	0	2	2	-8	-2	4	0	2	-2	0	4	2	0	-2	-2	0
$1B_x$	0	2	6	-4	-6	0	0	2	6	8	0	-2	-4	-6	-2	0
$1C_x$	0	0	-2	2	4	0	-6	2	-2	6	-4	0	2	-2	0	0
$1D_x$	0	4	-2	6	-8	0	-2	2	10	-2	-8	-8	2	2	0	4
$1E_x$	0	-4	-8	0	-2	-2	-2	2	-2	2	-2	6	4	4	4	0
$1F_x$	0	-4	8	-8	2	-6	-6	-2	-2	2	-2	-2	-8	0	0	-4
20_x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21_x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22_x	0	-4	-2	2	-2	2	-4	8	-4	0	-6	6	2	-2	-16	-12
23_x	0	0	-2	-2	6	-2	-4	4	0	0	-2	-2	-2	6	4	-4
24_x	0	-2	6	4	0	6	-2	4	4	-6	-2	4	0	14	2	0
25_x	0	6	2	0	0	6	2	0	-4	-6	2	-8	0	-2	6	-4
26_x	0	2	4	-2	-2	0	2	-4	4	-2	-4	-2	6	0	-2	0
27_x	0	-10	0	-2	6	4	6	-4	0	6	-12	2	2	0	6	-4
28_x	0	4	-2	-2	0	4	-6	2	2	-6	4	0	6	-2	-4	0
29_x	0	0	2	6	0	0	6	2	2	-2	-8	0	-2	-6	0	0
$2A_x$	0	0	-4	-8	6	6	6	-6	6	2	-2	-2	-8	4	-4	4
$2B_x$	0	8	0	4	6	-2	-6	6	2	6	-2	6	-4	0	4	4
$2C_x$	0	2	4	-6	0	-6	0	6	-2	-4	2	-4	-2	4	6	0
$2D_x$	0	-2	-4	-2	0	-2	-8	2	-2	0	-6	-8	-2	0	-2	4
$2E_x$	0	6	2	-4	6	4	4	-2	-10	-8	0	-2	4	-2	2	0
$2F_x$	0	6	-6	-4	6	-4	4	-2	2	4	4	-6	0	2	-2	-4
30_x	0	2	-2	0	-4	-6	-2	-4	4	2	2	0	0	2	2	4
31_x	0	2	-2	0	0	-2	2	0	0	-2	-2	-4	0	2	2	4
32_x	0	6	0	-2	-2	8	2	4	0	10	0	2	-2	4	2	0
33_x	0	-6	0	10	2	0	-2	-4	0	6	0	-10	2	4	-2	0
34_x	0	0	-12	4	-4	0	4	-8	-4	0	-4	0	-4	-4	0	0
35_x	0	-8	0	0	8	-4	4	0	0	-4	-4	0	4	4	-4	4
36_x	0	4	-2	-6	-2	-2	8	0	4	-4	-2	-2	6	2	-4	0
37_x	0	-8	-6	-6	-6	6	0	4	12	0	2	-2	2	2	4	-4
38_x	0	2	4	-6	0	-2	4	-2	-6	4	-6	0	6	4	-2	0
39_x	0	-2	8	2	-4	6	-4	-6	-2	-4	2	4	-2	0	2	0
$3A_x$	0	6	-10	0	2	4	0	-2	6	-4	0	2	4	-2	-2	-4
$3B_x$	0	-2	-6	-4	-10	0	-8	-2	-10	4	4	-2	0	2	-2	4
$3C_x$	0	-8	-6	-2	0	-4	2	2	-6	2	4	0	10	-2	4	4
$3D_x$	0	4	2	2	4	4	-2	2	-2	10	0	0	2	2	4	0
$3E_x$	0	-4	4	-4	2	2	-2	2	2	-2	-2	-2	4	-4	0	4
$3F_x$	0	-4	-4	-4	14	6	-6	-2	2	-2	6	-2	0	0	-4	0

Tabel 1. *linear approximation table* dari S5

2.1 Teknik Pencarian Matsui untuk Mendapatkan Pendekatan Terbaik

Matsui mengusulkan algoritma pencarian berbasis pada *recursive reasoning*. Dengan mendapatkan nilai probabilitas dari putaran ke-*i* dengan karakteristik terbaik dengan $1 \leq i \leq n-1$, algoritma menghasilkan karakteristik terbaik untuk *n* putaran. Hal ini dilakukan dengan menjalankan pohon pencarian, dimana cabangnya dipotong segera setelah mengetahui dengan yakin bahwa nilai probabilitasnya tidak dapat mencapai suatu nilai perkiraan awal pada pencarian.

Algoritma ini dapat diterapkan pada banyak *cipher* blok tetapi dengan tingkat efisiensi yang berbeda. Waktu pencarian sangat tergantung pada ketepatan pemilihan nilai perkiraan awal. Nilai estimasi yang kecil akan memperbesar waktu pencarian pada pohon pencarian. Sedangkan untuk nilai estimasi yang besar, algoritma ini tidak akan menghasilkan karakteristik apapun. Untuk DES, estimasi yang baik dapat didapatkan dengan mudah, yaitu melakukan pencarian terbatas untuk semua karakteristik yang menyilang pada sebuah S-box pada tiap putaran. Hal ini tidak berhasil dengan baik pada blok *cipher* yang lain. Isi dari S-box juga mempengaruhi efisiensi dari algoritma ini.

2.2 Gabungan Metode

Teknik kriptanalisis linier telah mendapatkan banyak perhatian sejak publikasinya dan telah ada beberapa gabungan metode. Salah satu teknik adalah menggabungkan pendekatan differensial-linier yang diusulkan oleh Langford dan Hellman. Metode yang lain menggunakan perangkingan kunci sebagai pemilihan antara data dengan waktu analisis. Lalu ada teknik criptalanisis denan partisi yang mempelajari hubungan antara partisi dari *plaintext* dengan *ciphertext*. Criptanalisis *chi-square* yang telah berhasil pada beberapa metode penyandian seperti RC6.

3. Kesimpulan

Untuk bisa melakukan tipe serangan dengan kripanalisis linier, dibutuhkan beberapa plaintext sebagai masukan untuk melakukan perhitungan probabilitas approksimasi linier S *boxes*. Dari hasil perhitungan ini akan didapatkan bit-bit paritas kunci rahasia.

Kelebihan kriptanalisis linier adalah bisa diterapkan untuk menyerang sistem kriptografi dengan mode *cipher block*, namun tingkat efisiensi yang berbeda. Contoh penerapannya adalah pada penyerangan sistem kriptografi dengan menggunakan algoritma DES.

4. Daftar Pustaka

[1] Alex Biryukov, *Linear Cryptanalysi*, http://www.esat.kuleuven.ac.be/~abiryuko/Enc/e32.pdf, diakses tanggal 30 Desember 2004 pukul 15:15.

[2] Eli Biham, *On Matsui's Linear Cryptanalysis*, Israel : Haifa 32000, 2004.

[3] *Basic Cryptanalysis*, http://www.bbc.co.uk/dna/h2g2/alabaster/A613135, diakses tanggal 30 Desember 2004 pukul 15:10.

[4] Rinaldi Munir, *Bahan Kuliah : Serangan (Attack) Terhadap Kriptografi*, Bandung : ITB, 2004.