STUDI MENGENAI JARINGAN FEISTEL TAK SEIMBANG DAN CONTOH IMPLEMENTASINYA PADA SKIPJACK CIPHER

Stevie Giovanni – NIM : 13506054

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if116054@students.if.itb.ac.id

Abstrak

Feistel network atau jaringan feistel adalah struktur simetris yang digunakan dalam mengkonstruksi *block cipher*. Model dasar jaringan feistel membagi blok masukan menjadi beberapa bagian, mengenakan fungsi feistel kepada upablok-upablok tersebut, kemudian melakukan operasi XOR antara upablock-upablock tersebut. Kunci enkripsi (*round key*) digunakan pada tiap fungsi feistel yang dikenakan pada upablok. Rangkaian operasi ini dapat dilakukan berulang kali selama beberapa putaran atau *round. Round key* pada setiap putaran dibangkitkan secara *pseudo-random.* Karena sifatnya yang simetris, jaringan feistel memiliki keunggulan yaitu algoritma yang digunakan untuk proses enkripsi dapat digunakan lagi untuk proses dekripsi.

Kelebihan lain dari jaringan feistel yang membuatnya menjadi menarik untuk dibahas adalah karena fungsi feistel yang dikenakan pada upablok dapat dirancang sesulit apapun. Perancang fungsi tidak perlu khawatir akan kehilangan properti *reversible* dari jaringan feistel. Selain itu perancang fungsi juga tidak perlu menentukan fungsi balikan (*f-inverse*) dari fungsi feistel tersebut.

Seperti kita ketahui, makin rumit algoritma enkripsi yang digunakan, makin sulit sebuah *ciphertext* dapat dipecahkan. Jaringan feistel dapat diperumit dengan beberapa cara, misalnya dengan merancang fungsi feistel yang sangat rumit untuk digunakan pada tiap *round*. Cara lainnya adalah dengan memodifikasi model dasar jaringan feistel itu sendiri sehingga menjadi sebuah *extended feistel network.* Umumnya, blok masukan dalam sebuah jaringan feistel dibagi menjadi beberapa upablok dengan ukuran yang sama. Namun untuk memperumit jaringan feistel, kita dapat mambagi blok masukan menjadi beberapa upablok dengan ukuran berbeda. Jaringan feistel seperti ini disebut juga dengan *unbalanced feistel network* atau jaringan feistel tak seimbang.

Dalam makalah ini, penulis akan membahas terlebih dahulu sedikit mengenai *cipher* blok, kemudian secara mendalam akan dibahas mengenai jaringan feistel tak seimbang, kekurangan dan kelebihannya, serta salah satu contoh implementasinya dalam *skipjack cipher*, sebuah algoritma enkripsi yang dikembangkan oleh *U.S. National Security Agency* dengan menggunakan metode jaringan feistel tak seimbang.

Kata kunci: Feistel Network, Extended Feistel Network, Unbalanced Feistel Network, round key, *skipjack cipher*, block *cipher*, *fungsi feistel*.

1. Pendahuluan

Dalam dunia di mana perkembangan teknologi komunikasi berlangsung dengan sangat pesat, kerahasiaan dan keamanan data menjadi salah satu faktor penting yang harus diperhatikan. Data yang mengalir dari satu pihak ke pihak lainnya harus dijaga keamanan dan kerahasiaannya dari serangan-serangan pihak yang tidak berkepentingan. Untuk menjamin hal ini, aliran data umumnya disandikan. Proses penyandian data dari *plaintext* menjadi *ciphertext* disebut enkripsi (*encryption*) atau *enciphering*. Untuk mendapatkan kembali *plaintext* semula, pada *end-point* dilakukan proses dekripsi (*decryption*) atau *deciphering*. Ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya disebut dengan kriptografi.

Seiring dengan perkembangan waktu, makin banyak algoritma-algoritma enkripsi yang bermunculan. Dari algoritma enkripsi yang satu

ke algoritma enkripsi yang lain, terjadi peningkatan pada keamanan data. Berbagai metode digunakan untuk meningkatkan keamanan data ini. Salah satu metodenya adalah dengan menggunakan *feistel network* atau jaringan feistel. Jaringan feistel memiliki keunggulan karena sifatnya yang *reversible*

Salah satu contoh implementasi dari penggunaan jaringan feistel ini adalah pada algoritma enkripsi *skipjack*. *Skipjack cipher* adalah algoritma enkripsi yang dikembangkan U.S. *National Security Agent* (NSA). Algoritma *Skipjack cipher* ini sengaja didesain untuk menggantikan algoritma *Data Encryption Standard* atau DES.

2. *Cipher* Blok

Cipher blok adalah salah satu teknik kriptografi modern yang bekerja dengan cara membagi plainteks menjadi blok-blok dengan panjang yang sama. Setiap blok kemudian dienkripsi untuk menghasilkan blok-blok yang kemudian membentuk cipherteks. *Cipher* blok memiliki empat mode operasi yaitu *Electronic Code Book* (ECB), *Cipher Block Chaining* (CBC), *Cipher Feedback* (CFB), dan *Output Feedback* (OFB).

Pada mode ECB, setiap blok plainteks dienkripsi secara individual dan independen menjadi blok cipherteks. ECB memiliki keuntungan yaitu enkripsi tidak perlu dilakukan secara linear dan terurut. Selain itu Jika satu atau lebih bit pada blok cipherteks mengalami kesalahan, maka kesalahan ini hanya mempengaruhi blok cipherteks yang bersangkutan pada proses dekripsi. Namun ECB memilikim kelemahan karena akan terdapat blok blok cipherteks yang berulang. Selain itu pihak lawan dapat mengelabui penerima pesan dengan menyelipkan atau menghapus blok-blok cipherteks.

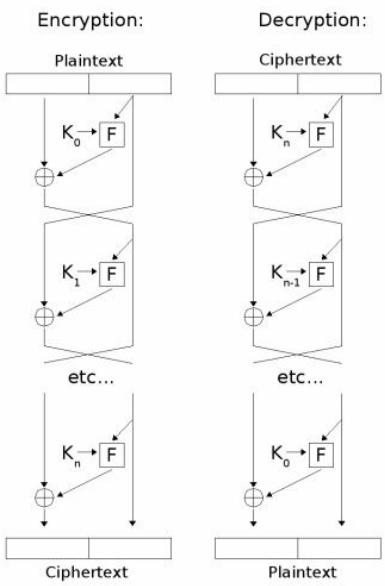
Pada mode CBC, blok plainteks yang sedang dienkripsi, di xor dengan blok cipherteks hasil enkripsi blok plainteks sebelumnya. Keuntungan dari mode CBC, blok cipherteks tidak hanya bergantung pada blok plainteks, tetapi juga terhadap cipherblok sebelumnya. Namun karena ketergantungan ini, CBC memiliki kelemahan, kesalahan satu bit dapat mempengaruhi seluruh blok cipherteks.

Pada mode CFB, data dienkripsi dalam unit yang lebih kecil daripada ukuran blok. Hal ini memberikan keuntungan ketika menerapkan proses enkripsi dalam transmisi data. Walaupun blok plainteks yang diterima belum lengkap, enkripsi tetap dapat dilakukan. Secara umum, CFB p-bit mengenkripsi plainteks sebanyak p bit setiap kalinya, di mana $p \leq n$ (n = ukuran blok). Dengan kata lain, CFB mengenkripsi *cipher* blok seperti pada *cipher* aliran.

Mode OFB memiliki kemiripan dengan mode CFB, kecuali bahwa p-bit dari hasil enkripsi terhadap antrian disalin menjadi elemen posisi paling kanan di antrian tersebut. Dekripsi kemudian dilakukan sebagai kebalikan dari proses enkripsi.

3. Jaringan Feistel

Feistel network atau jaringan feistel adalah adalah struktur simetris yang digunakan dalam mengkonstruksi *block cipher*. Jaringan feistel pertama kali ditemukan oleh Horst Feistel pada tahun 1970. Sejak saat itu jaringan feistel seringkali digunakan dalam banyak algoritma enkripsi, misalnya DES, GOST, Lucifer, Triple DES, dan masih banyak lagi.



Gambar 1 Jaringan Feistel Sederhana

Model dasar jaringan Feistel adalah sebagai berikut :

1. Bagi blok plainteks berukuran n bit menjadi dua bagian, kiri (L) dan kanan (R), yang masing-masing panjangnya n/2.
2. Untuk setiap putaran i = 0,1,2,...,n, lakukan :

$$L_{i+1} = R_i$$
$$R_{i+1} = L_i \oplus \mathrm{F}(R_i, K_i)$$

Dalam hal ini,
n = jumlah putaran
K_i = upa-kunci pada putaran ke-i
F = fungsi feistel

3. Jika blok plainteks adalah (L_0,R_0), maka *Cipher block* adalah (R_{n+1},L_{n+1}).

Untuk menyederhanakan proses di atas, pada satu putaran ke-i jaringan feistel,

$$X_{i+1} = (F_{k_i}(msb_{n/2}(X_i)) \oplus lsb_{n/2}(X_i)) || msb_{n/2}(X_i)$$

Di mana,
X_i adalah blok masukan
X_{i+1} adalah blok keluaran
F_{ki} adalah fungsi feistel dengan upa kunci ke-i
$Msb_{n/2}$ adalah n/2 *most significant bit* dari blok masukkan
$Lsb_{n/2}$ adalah n/2 *least significant bit* dari blok masukkan

Salah satu keunggulan dari penggunaan jaringan feistel dalam algoritma enkripsi adalah fungsi feistel F tidak perlu memiliki atribut *invertible* (memiliki F-*inverse*) dan dapat di buat serumit apapun. Selain itu perbedaan dari proses enkripsi dan dekripsinya hanya terletak pada penggunaan upa-kunci yang dibalik.

3.1 Jaringan Feistel Tak Seimbang

Terdapat banyak variasi yang dapat dilakukan pada model sederhana jaringan feistel. Salah satunya adalah jaringan feistel tak seimbang. Jaringan feistel tak seimbang atau *unbalanced feistel network* (UFN) adalah sebuah jaringan feistel, di mana bagian kiri (L) dan bagian kanan (R) tidak memiliki ukuran yang sama. Pada satu putaran *s-on-t* UFN,

$$X_{i+1} = (F(msb_s(X_i), k_i) \oplus lsb_t(X_i)) || msb_s(X_i)$$

$msb_s(X_i)$ disebut juga *source block* sedangkan lsb_t disebut juga *target block*. Jika s > t, UFN disebut *source heavy*. Jika s < t, UFN disebut *target heavy*.

3.2 Istilah-Istilah Dalam UFN

- UFN homogen dan heterogen
  Sebuah UFN dikatakan homogen jika fungsi feistel F yang digunakan pada tiap putaran sama, dan hanya berbeda pada upa-kunci yang digunakan. UFN dikatakan heterogen jika sungsi feistel F yang digunakan pada tiap putaran berbeda, dan hanya sama pada upa-kunci yang digunakan.
- UFN yang tidak lengkap (*incomplete* UFN)
  Sebuah UFN dikatakan tidak lengkap atau *incomplete* jika s + t < n, yaitu jumlah dari ukuran kedua upablok tidak sama dengan ukuran blok aslinya. Sebaliknya UFN dikatakan lengkap jika s + t = n. Pada *incomplete* UFN, terdapat z = n – s – t bit yang bukan begian dari *target* maupun *source block* yang disebut *null block*.
- UFN yang tidak konsisten (*inconsistent* UFN)
  UFN dikatakan kosisten jika nilai n, s, t, dan z pada setiap putaran sama, dan dikatakan tidak konsisten jika nilai n, s, t, dan z pada setiap putaran berbeda.
- Siklus, rotasi, dan *even* UFN
  Sebuah siklus adalaha jumlah putaran yang diperlukan agar setiap bit pada blok telah menjadi bagian baik dalam *source block* maupun dalam *target block* setidaknya sekali. Didefinisikan dengan,

  $$C = \lceil \frac{n}{\min(s,t)} \rceil$$

  Sebuah rotasi adalah jumlah putaran yang diperlukan yang diperlukan agar setiap bit pada sebuah blok kembali ke posisinya semula. Didefiniskan dengan,

  $$G = \frac{n}{\gcd(s,t)}$$

  Sebuah UFN dikatakan *even* jika C = G.

4. *Skipjack Cipher*

Skipjack cipher adalah sebuah algoritma enkripsi *cipher* blok yang menggunakan mode *electronic code book* atau ECB yang telah dijelaskan sebelumnya. *Skipjack* menggunakan kunci sepanjang 80 bit untuk mengenkripsi blok plainteks berukuran 64 bit. Algoritma *skipjack* mengimplementasikan jaringan feistel tak seimbang sebanyak 32 putaran dalam proses enkripsinya. Algoritma enkripsi *Skipjack* didesain oleh *U.S.* National Security Agent *(NSA)* untuk menggantikan algoritma enkripsi DES.

4.1. Proses Enkripsi *Skipjack Cipher*

Pada algoritma *skipjack*, blok masukan berukuran 64 bit dibagi menjadi empat upablok, W1, W2, W3, dan W4 yang masing-masing berukuran 16 bit sebelum dimasukkan ke dalam putaran feistel. Terdapat dua tipe putaran dalam *skipjack cipher* yang disebut dengan *stepping rule*. Kedua tipe tersebut adalah,

Tipe A :

- Upablok W1 dienkripsi dengan fungsi permutasi G yang adalah empat putaran *feistel cipher* biasa.
- Hasil enkripsinya dan nomor putaran yang bertambah dari satu sampai dengan 32, di xor dengan upablok W4.
- Setiap upablok dirotasi W1 ke W2, W2 ke W3, W3 ke W4, dan W4 ke W1.

Tipe B :

- Upablok W2 di-xor dengan W1 dan nomor putaran.
- W1 dienkripsi dengan fungsi permutasi G.
- Setiap upablok dirotasi W1 ke W2, W2 ke W3, W3 ke W4, dan W4 ke W1.

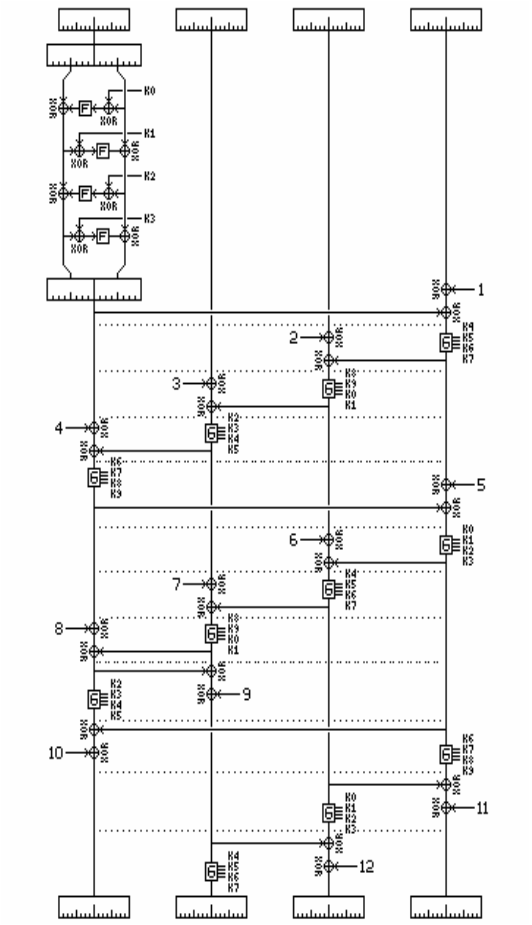
Ke-32 putaran feistel tak seimbang pada algoritma *skipjack* terdiri dari delapan putaran tipe A, delapan putaran tipe B, delapan putaran tipe A, dan delapan putaran tipe B.

Upa-kunci pada setiap putaran digunakan di dalam fungsi feistel pada fungsi permutasi G. Fungsi feistel dalam fungsi permutasi G menerima input delapan bit. Input tersebut kemudian di-xor dengan upa-kunci. Kemudian hasilnya disubstitusi berdasarkan tabel *lookup* yang disebut tabel F.

```
a3 d7 09 83 f8 48 f6 f4 b3 21 15 78 99 b1 af f9
e7 2d 4d 8a ce 4c ca 2e 52 95 d9 1e 4e 38 44 28
0a df 02 a0 17 f1 60 68 12 b7 7a c3 e9 fa 3d 53
96 84 6b ba f2 63 9a 19 7c ae e5 f5 f7 16 6a a2
39 b6 7b 0f c1 93 81 1b ee b4 1a ea d0 91 2f b8
55 b9 da 85 3f 41 bf e0 5a 58 80 5f 66 0b d8 90
35 d5 c0 a7 33 06 65 69 45 00 94 56 6d 98 9b 76
97 fc b2 c2 b0 fe db 20 e1 eb d6 e4 dd 47 4a 1d
42 ed 9e 6e 49 3c cd 43 27 d2 07 d4 de c7 67 18
89 cb 30 1f 8d c6 8f aa c8 74 dc c9 5d 5c 31 a4
70 88 61 2c 9f 0d 2b 87 50 82 54 64 26 7d 03 40
34 4b 1c 73 d1 c4 fd 3b cc fb 7f ab e6 3e 5b a5
ad 04 23 9c 14 51 22 f0 29 79 71 7e ff 8c 0e e2
0c ef bc 72 75 6f 37 a1 ec d3 8e 62 8b 86 10 e8
08 77 11 be 92 4f 24 c5 32 36 9d cf f3 a6 bb ac
5e 6c a9 13 57 25 b5 e3 bd a8 3a 01 05 59 2a 46
```

Gambar 2 Tabel *lookup* F

Untuk mendukung penjelasan mengenai algoritma enkripsi skipjack cipher, berikut disajikan sebuah gambar 12 putaran pertama pada algoritma tersebut.



Gambar 3 12 putaran pertama *skipjack cipher*

Putaran pertama adalah putaran tipe A. Diperlihatkan juga fungsi permutasi G yang terdiri dari empat putaran *feistel cipher* standard. Pada putaran-putaran berikutnya, fungsi permutasi G digambarkan dengan kotak dengan huruf G di dalamnya. Untuk menghindari kompleksitas, fungsi G di geser untuk menggantikan rotasi upablok.

4.2. Proses Dekripsi *Skipjack Cipher*

Proses dekripsi pada algoritma Skipjack dilakukan dengan cara membalik urutan putaran feistel yang terdapat pada proses enkripsinya. Tiap putaran diganti dengan putaran yang ekivalen berikut.

Ekivalen tipe A :

- Upablok W1 di-xor dengan upablok W2 dan nomor putaran.
- Upablok W2 dijadikan input inverse fungsi permutasi G.
- Tiap upablok dirotasi W1 ke W4, W2 ke W1, W3 ke W2, dan W4 ke W3.

Ekivalen tipe B :

- Upablok W2 dijadikan input inverse fungsi permutasi G.
- Upablok W3 di-xor dengan nomor putaran dan hasil hari langkah sebelumnya.
- Tiap upablok dirotasi W1 ke W4, W2 ke W1, W3 ke W2, dan W4 ke W3.

Inverse fungsi permutasi G didapat dengan cara menggunakan keempat upa kunci dengan urutan terbalik dan menukar posisi kiri dan kanan dari upablok masukkan berukuran 16 bit.

4.3. Jaringan Feistel Tak Seimbang Dalam Algoritma *Skipjack Cipher*

Dari penjelasan mengenai algoritma *skipjack cipher*, dapat dilihat bahwa *skipjack cipher* menggunakan modifikasi dari jaringan feistel tak seimbang. Walaupun terlihat bahwa skipjack cipher membagi blok masukan menjadi empat buah upablok W1, W2, W3, dan W4, yang terjadi sebenarnya dalam tiap putaran adalah, algoritma *skipjack* cipher membagi blok masukkan menjadi dua bagian msb_s (W1) dan lsb_t (upablok-upablok lainnya). Msb_s kemudian dikenai fungsi permutasi G.

Pada tipe A, hasil permutasi G kemudian di-xor dengan sebagian dari lsb_t sebelum akhirnya disatukan lagi sehingga membentuk blok keluaran, sedangkan Pada tipe B sebagian dari lsb_t di-xor dengan msb_s yang belum dikenai fungsi permutasi G.

Skipjack cipher menggunakan jaringan feistel tak seimbang yang homogen, lengkap, dan konsisten. Dikatakan homogen karena fungsi feistel yang digunakan pada setiap putaran sama, yaitu fungsi perutasi G. Lengkap karena s + t = n, yaitu jumlah bit seluruh upabloknya sama dengan jumlah bit blok masukannya. *Skipjack cipher* disebut konsisten karena z= 0, s = 16, t = 48, dan n = 64, dan selalu konstan untuk tiapp putaran.

gcd(s,t) = 16 dan min(s,t) = 16. Hal ini membuat nilai C yang menyatakan siklus = 64 / 16 = 4 dan G yang menyatakan rotasi = 64 / 16 = 4. Karena C dan G sama, maka jaringan feistel tak seimbang yang digunakan oleh skipjack cipher mempunyai properti *even.*

5. Analisis Jaringan Feistel Tak Seimbang

5.1. Jaringan Feistel Sederhana

Jika melihat dari definisi umum dan istilah-istilah jaringan feistel tak seimbang, maka jaringan feistel sederhana dapat disamakan dengan jaringan feistel tak seimbang dengan properti homogen, konsisten, lengkap, dan even dengan s = n/2, t = n/2, z = 0, C = 2, dan G = 2.

5.2. Confusion Rate

Jika seseorang mengetahui blok ke i (X_i), tanpa mengetahui upa-kunci ke i(k_i), maka akan sulit untuk mengetahui blok ke i + 1 (X_{i+1}). Hal ini disebut *confusion*. Untuk memastikan algoritma enkripsi aman, maka untuk setiap block $X_{i,}$ di mana tidak diketahui upa-kuncinya, harus terdapat X_{i+t} yang tidak diketahui.

Pada jaringan feistel, sebuah bit dapat menjadi *obscured* hanya jika bit tersebut terdapat pada *target block*. Dengan kata lain, kemungkinan sebuah bit menjadi *obscured* dalam sebuah siklus, atau disebut juga rate of confusion R_c pasti lebih kecil atau sama dengan t / n, di mana t adalah jumlah target bit, dan n adalah ukuran blok. Makin besar nilai R_c maka algoritma enkripsi akan makin kebal terhadap serangan kriptanalisi linear.

5.3. Diffusion Rate

Sekecil apapun perubahan pada X_i harus dapat mempengaruhi seluruh bit pada $X_{i+t,}$ untuk beberapa t yang terdefinisi. Hal ini disebut juga *diffusion.*Dalam feistel *cipher*, satu-satunya cara sebuah bit dapat mempengaruhi bit yang lain adalah ketika bit tersebut berada pada *source block*.

Dari pemahaman ini, kita dapat menghitung rate of diffusion, yaitu kemungkinan terkecil sebuah bit dapat mempengaruhi bit lainnya dalam sebuah siklus. Karena kemungkinan ini hanya bisa dicapai jika bit beraada pada *source block* maka nilai *rate diffusion* R_d pasti lebih kecil atau sama dengan s / n (kemungkinan sebuah bit berada di *source block*). Makin besar R_d, maka

enkripsi akan makin kebal terhadap serangan kriptaanalisis linear.

5.4. Perbandingan Dengan Jaringan Feistel Sederhana

Jaringan feistel sederhana memiliki keterbatasan pada confusion rate dan diffusion ratenya. Karena s dan t sama-sama bernilai n / 2, maka Rd dan Rc hanya bisa mencapai nilai maksimum masing-masing ½. Dengan menggunakan Jaringan Feistel tak seimbang, kita dapat merancang algoritma cipher block yang lebih kuat dengan mengkombinasikn nilai Rc dan Rd dengan struktur jaringan feistel yang sesuai. Dengan merubah nilai s dan t, dengan nilai lainnya tetap konstan, kita sudah bisa mendapatkan algoritma enkripsi dengan kekebalan terhadap kriptanalisis yang berbeda.

6. Analisis Algoritma *Skipjack Cipher*

Eli Biham and Adi Shamir telah berhasil memecahkan 16 dari 32 putaran algoritma *skipjack cipher* dalam sehari sejak algoritma ini dideklasifikasi. Selain itu, Alex Biryukov berhasil memecahkan hingga 31 dari 32 putaran algoritma *skipjack cipher*. Hal ini menyatakan algoritma skipjack cipher masih memiliki banyak kelemahan. Salah satu kelemahannya disebabkan karena Algoritma ini dirancang dengan metode ECB cipher blok yang memungkinkan adanya pola-pola yang berulang pada cipherteks.

Algoritma skipjack cipher memiliki keunggulan karena menerapkan prinsip diffusion dan confusion yang cukup tinggi. Rotasi yang dilakukan terhadap upablok menyebabkan dependency antara upablok yang satu dengan upablok yang lain. Selain itu fungsi feistel dalam permutasi G dapat dirancang serumit apapun untuk meningkatkan confusion.

7. Kesimpulan

- ✓ Keamanan jaringan feistel ditentukan oleh diffusion rate dan confusion rate.
- ✓ Dengan jaringan feistel tak seiimbang dapat dirancang algoritma enkripsi yang lebih kuat terhadap serangan kriptanalisis.
- ✓ Jaringan feistel sederhana memiliki confusion dan difusion rate yang seimbang karena s = t. Hal ini membatasi kekuatan algoritma enkripsi yang menggunakan jaringan feistel sederhana.
- ✓ Algoritma *skipjack cipher* memiliki kelemahan karena menggunakan mode cipher blok ECB.
- ✓ Jaringan feistel tak seimbang yang digunakan pada algoritma skipjack cipher dapat digunakan pada mode cipher blok yang lain.

DAFTAR PUSTAKA

[1] Munir, Rinaldi. Kriptografi, Penerbit Informatika. 2006.

[2] Bruce Schneier and John Kelsey. *Unbalanced Feistel Network and Block-Cipher Design.*

[3] http://www.cs.technion.ac.il/~biham/Reports/SkipJack/note1.html, tanggal akses : 22 Maret 2009.

[4] http://www.cs.georgetown.edu/~denning/crypto/clipper/SKIPJACK.txt, tanggal akses : 1 April 2009.

[5] http://quadibloc.com/crypto/co040303.htm, tanggal akses : 1 April 2009.

[6] http://en.wikipedia.org/wiki/Feistel_cipher#Unbalanced_Feistel_Cipher, tanggal akses : 20 Maret 2009.

[7] http://en.wikipedia.org/wiki/Skipjack_encryption_algorithm, tanggal akses : 20 Maret 2009.