STUDI PERBANDINGAN ALGORITMA SIMETRI *LUCIFER* DAN *BLOWFISH*

Miftah Mizan – NIM : 13507064

Program Studi Informatika, Institut Teknologi Bandung
Jl. Ganesha No. 10, Bandung
E-mail : if17064@students.if.itb.ac.id

Abstrak

Makalah ini membahas tentang studi perbandingan algoritma simetri *Lucifer* dan *Blowfish*. Dalam kriptografi, Lucifer adalah block cipher pertama yang dibuat. Lucifer dikembangkan pertama kali oleh Horst Feistel dan koleganya di IBM. Lucifer sendiri adalah pendahulu langsung dari block cipher yang sekarang banyak digunakan yaitu Data Encryption Standard (DES).Algoritma *Blowfish* diciptakan oleh Bruce Schneier sebagai salah satu alternatif pengganti DES yang dirasa sudah tidak aman lagi. Algoritma ini menggunakan kunci yang panjangnya bisa bervariasi antara 32-bit hingga 448-bit.

Dalam makalah ini juga dilakukan pengujian yang dibatasi hanya dari segi performansi dan keamanan data. Pengujian untuk performansi dilakukan dengan cara menghitung berapa lama proses enkripsi maupun dekripsi dari masing-masing algoritma dieksekusi untuk mengenkripsi arsip yang sama. Untuk pengujian keamanan data, dilakukan modifikasi pada blok-blok cipherteks untuk kemudian didekripsi kembali dan dilihat perubahan dari teks asalnya.

***Kata kunci**: Blowfish Cipher, Lucifer Cipher,* block cipher, Feistel network.

1. Pendahuluan

Sekarang ini informasi merupakan suatu hal penting yang dinilai berharga sebagai aset dalam bidang apapun. Dalam kondisi teknologi yang sangat berkembang pesat hingga sekarang, informasi juga bukan lagi merupakan suatu hal yang statis. Namun telah menjadi hal yang dinamis, dapat kita lihat dari arus informasi yang sangat cepat berubah.

Mengalirnya arus informasi dengan sangat cepat tersebut menimbulkan masalah tersendiri. Confidentiality dan integrity dari informasi yang ada mulai dipertanyakan, tidak hanya itu keamanan dari informasi tersebut juga diragukan. Hal ini dikarenakan informasi itu sendiri mulai disalahgunakan oleh oknum-oknum. Alhasil, informasi yang ada di internet ataupun tempat yang banyak diakses orang sudah tidak memiliki tingkat confidential yang layak.

Banyak sekali cara-cara orang untuk mengubah informasi-informasi ataupun data-data yang mengalir. Mulai dari menyisipkan data semu yang menyebabkan orang tidak dapat menerima pesan sebenarnya, bahkan hingga mencuri data tersebut untuk kepentingan lain.

Seiring dengan munculnya masalah tersebut, mulai banyak ditemukan cara untuk menyembunyikan informasi atau data yang ingin dikirimkan. Sejak zaman dahulu hal tersebut telah dilakukan oleh para kaisar atau penguasa untuk mengirimkan pesan rahasia dengan cara mengubah susunan pesan atau menerjemahkannya ke dalam karakter lain. Salah satu contoh yang banyak digunakan pada zaman itu adalah *Caesar Cipher* dimana setiap huruf pada pesan ditukar dengan huruf lain yang merupakan hasil pergeseran dari huruf pesan.

Seiring berkembangnya teknologi, data atau informasi yang adapun mulai berkembang bentuknya. Tidak lagi hanya berupa kata atau kalimat, namun juga dapat berupa gambar, suara atau video. Sehingga cara untuk menyembunyikan pesan tersebut juga mulai bervariatif. Di sinilah kriptografi mulai berperan.

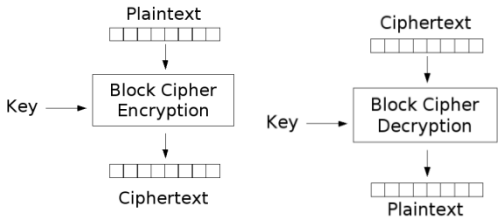
Horst Feistel menemukan cara menyimpan data dengan melakukan enkripsi pada data yang telah terbagi dalam block-block. Beliau saat itu menemukan apa yang disebut Lucifer algorithm yang merupakan block cipher pertama yang menjadi dasar dari block cipher-block cipher selanjutnya. Lucifer juga memiliki banyak variant yang salah satunya menjadi DES, salah satu block cipher algoritma yang populer digunakan.

Setelah beberapa tahun kemudian, ketika DES dirasakan sudah tidak memiliki kapasitas yang memadai, Bruce Schneier menciptakan Blowfish algorithm pada tahun 1993. Cipher block ini juga telah banyak digunakan pada produk-produk enkripsi lain. Blowfish menawarkan encryption rate yang baik pada software dan sampai saat ini belum ada cryptanalysis yang efektif untuk cipher block ini. Pada masa itu Blowfish merupakan salah satu block cipher yang tidak dipatenkan dan bisa diakses orang lain.

Dengan melihat kedua karakteristik yang sangat menarik dari dua algoritma di atas, Lucifer sebagai block cipher pertama dan pendahulu DES serta Blowfish sebagai pengganti DES, kedua algoritma ini memiliki nilai tersendiri untuk dipelajari lebih dalam dan dibandingkan performa yang dapat dilakukan keduanya.

2. Dasar Teori

Block Cipher

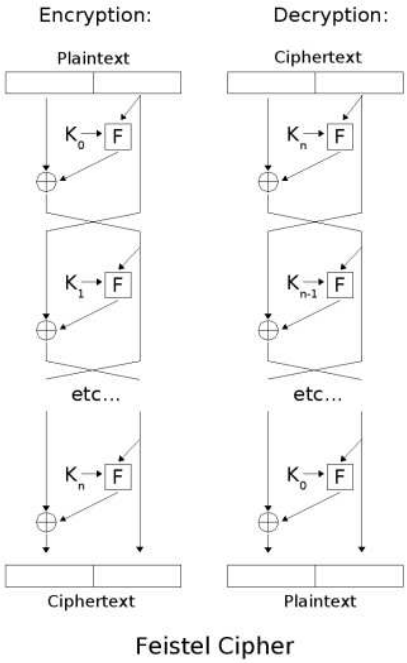


Dalam kriptografi block cipher adalah sebuah cipher dengan kunci simetri yang digunakan pada bit-bit yang merupakan grup dengan panjang tertentu, biasanya disebut block. Sebagai contoh sebuah algoritma enkripsi block cipher dapat menerima masukan berupa 128-bit block dan menghasilkan ciphertext sepanjang 128-bit block juga. Transformasi yang dilakukan dari plaintext menjadi ciphertext tersebut dilakukan dengan menggunakan key atau kunci yang dimasukkan oleh user.

Block cipher berbeda dengan stream cipher, di mana stream ciphers melakukan operasi pada setiap digit di satu waktu dan perubahannya terus terjadi selama enkripsi. Namun perbedaan yang ada tidak selalu terlihat jelas, karena terkadang block cipher juga dapat digunakan pada mode tertentu seperti stream cipher.

Salah satu bentuk block cipher yang sering digunakan adalah Data Encryption Standard (DES) yang kemudian dikembangkan menjadi Advance Encryption Standard (AES).
Untuk melakukan operasi pada pesan yang ukurannya lebih panjang dari ukuran block, maka dibutuhkan penggunaan metode atau mode operasi pada block cipher. Metode atau mode operasi tersebut adalah Electronic Codebook (ECB), Cipher-block Chaining (CBC), Cipher Feedback (CFB), dan Output Feedback (OFB).

Feistel Network



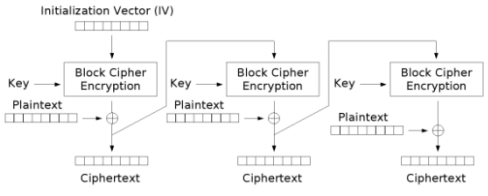
Feistel Cipher

Feistel network atau feistel cipher adalah suatu struktur simetri yang sering digunakan dalam konstruksi suatu block cipher yang pertama kali diciptakan oleh Horst Feistel. Feistel network memiliki fungsi round yang menandakan ada berapa round dalam struktur yang digunakan.

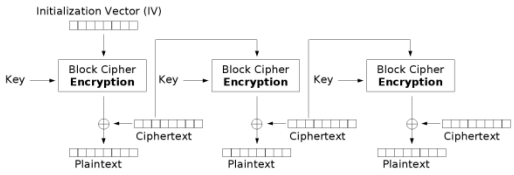
Struktur dari Feistel mempunyai keuntungan karena enkripsi dan dekripsi yang dilakukan identik, yang berbeda hanya urutan key yang digunakan.

Banyak sekali block cipher yang memanfaatkan struktur dari Feistel network ini di antaranya Lucifer dan Blowfish.

Cipher Feedback



Cipher Feedback (CFB) mode encryption



Cipher Feedback (CFB) mode decryption

Pada metode ini, block cipher yang digunakan bertindak layaknya stream cipher dimana akan digenerate sebuah key baru hasil enkripsi dari key yang dimasukkan user dengan ciphertext yang dihasilkan sebelumnya. Kemudian key baru tersebut akan di XOR kan dengan plaintext sehingga menghasilkan sebuah block ciphertext. Untuk membuat setiap pesan unik, sebuah initialization vector harus digunakan di block pertama. Selain itu karena sifat dari XOR, maka dekripsi yang dilakukan tidak jauh berbeda hanya saja kali ciphertext yang bertukar peran dengan plaintext untuk di XOR kan dengan key yang baru. Sementara untuk menggenerate key yang baru dilakukan dengan enkripsi ciphertext dengan key yang dimasukkan user.

3. Lucifer Cipher

Sejarah

Dalam kriptografi, Lucifer adalah block cipher pertama yang dibuat. Lucifer dikembangkan pertama kali oleh Horst Feistel dan koleganya di IBM. Lucifer sendiri adalah pendahulu langsung dari block cipher yang sekarang banyak digunakan yaitu Data Encryption Standard (DES).

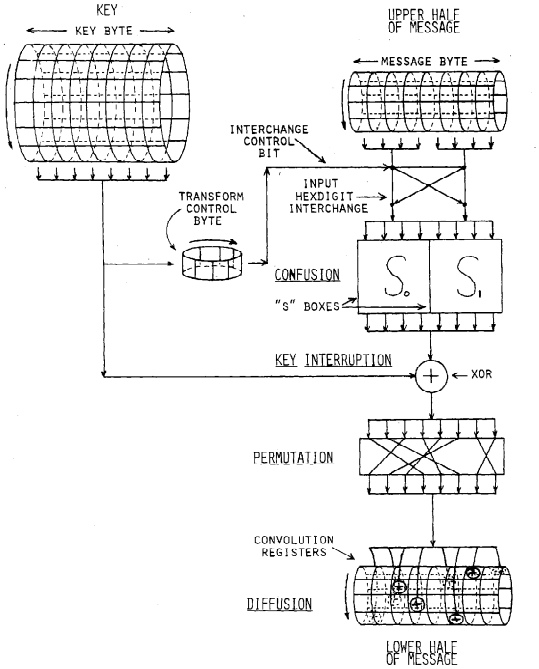
Kemudian Lucifer mulai banyak dikembangkan di dunia kriptografi. Salah satu variant yang dikembangkan menggunakan 48-bit key dan dapat dioperasikan pada 48-bit block. Cipher ini menggunakan substitusi-permutasi dan dua buah S-box 4-bit. Untuk menentukan S-box mana yang digunakan digunakan key yang ada. Variant ini dapat dioperasikan dengan menggunakan 24-bit per waktu ataupun secara sekuensial 8-bit per waktu.

Salah satu variant yang lain menggunakan kunci 64-bit yang beroperasi pada 32-bit block, menggunakan satu addition mod 4 dan sebuah 4-bit S-box. Konstruksi yang dibuat untuk varian ini didesain untuk beroperasi setiap 4-bit dalam satu waktu. Hal ini menyebbabkan dia menjadi salah satu block cipher terkecil yang pernah diimplementasikan.

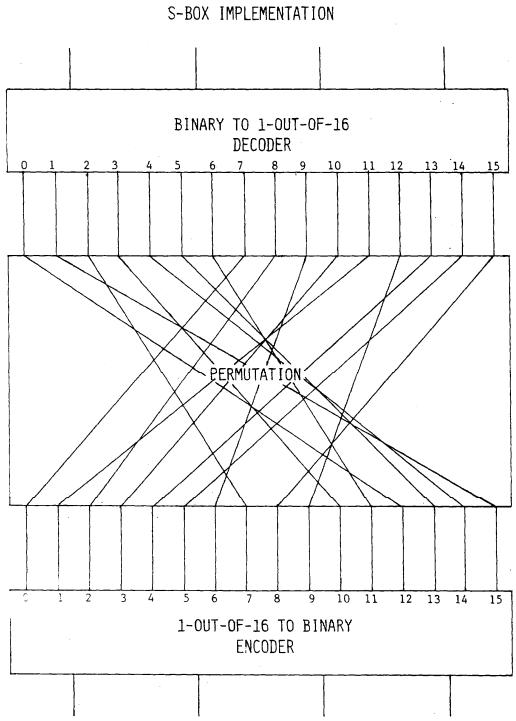
Variant yang lebih kuat menggunakan 128-bit key dan dapat dioperasikan pada 128-bit block. Cipher ini menggunakan substitusi-permutasi dan juga dua buah 4-bit S-box. Key yang ada menentukan S-box mana yang digunakan.

Kemudian yang paling terakhir dikembangkan Lucifer dengan 16-round Feistel network yang juga bekerja di 128-bit block dan menggunakan 128-bit key. Versi ini namun masih bisa dipecahkan dengan 2^{36} plain text dan 2^{36} kompleksitas. IBM mengajukan Lucifer dengan jenis ini sebagai kandidat dari DES. Setelah dilakukan redesign sehingga lebih kuat, versi ini ditetapkan menjadi DES.

Algoritma Lucifer



Lucifer yang menggunakan key dengan panjang 128 bit dan beroperasi pada 128 bit block juga menggunakan subkey pada setiap roundnya. Subkey untuk round pertama dibuat dari byte pertama key yang diulang dua kali kemudian diikuti dengan tujuh byte key berikutnya. Kemudian untuk mengenerate subkey berikutnya, dapat dilakukan dengan memutar key yang ada ke kiri sebanyak tujuh byte.



Untuk algoritma dari Lucifer sendiri atau F-Function dilakukan dengan melakukan XOR antara setengah bagian block sebelah kanan dengan delapan byte terakhir dari subkey yang digunakan pada round tersebut. Kemudian hasil dari XOR tersebut dimasukkan ke S-box dan diproses berdasarkan byte yang ada. Terakhir dilakukan permutasi pada 64 bit block yang dihasilkan.

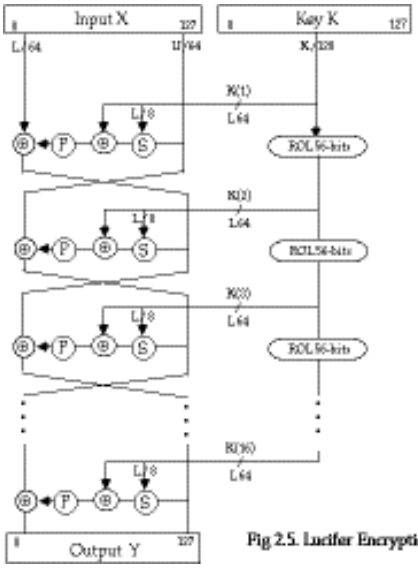


Fig 2.5. Lucifer Encryption Computation

Untuk Feistel network yang dilakukan terdiri dari 16 round. Sehingga awalnya block akan dibagi dua menjadi rightblock dan leftblock. Kemudian leftblock dan subkey pada round itu akan dimasukkan ke F-Function, setelahnya di XOR kan dengan rightblock. Yang hasilnya menjadi setengah bagian block untuk round berikutnya. Setiap akhir round leftblock dan rightblock akan ditukar kecuali pada round terakhir. Setelah itu leftblock dan rightblock dikombinasikan untuk mendapatkan ciphertext yang diinginkan.

4. Blowfish Cipher

Sejarah Blowfish

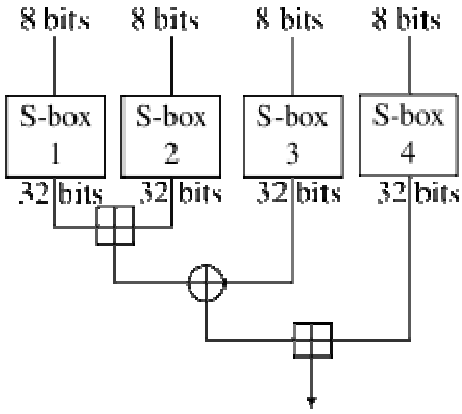
Blowfish adalah sebuah cipher block simetri dengan key, yang didesain oleh Bruce Schneier pada tahun 1993. Cipher block ini juga telah banyak digunakan pada produk-produk enkripsi lain. Blowfish menawarkan encryption rate yang baik pada software dan sampai saat ini belum ada cryptanalysis yang efektif untuk cipher block ini.

Schneier mendesain Blowfish sebagai salah satu algoritma kriptografi yang bisa menggantikan DES yang telah terlalu lama, selain itu Blowfish yang dibuat juga bebas dari masalah ataupun hambatan ketika diasosiasikan dengan algoritma yang lain. Pada saat Blowfish diluncurkan, Blowfish tidak dipatenkan sama sekali. Oleh karena itu bisa berada di domain publik dan bisa digunakan semua orang. Feature penting yang perlu diperhatikan pada desain ini adalah penggunaan S-box dengan key-dependent dan key scheduling yang sangat kompleks.

Algoritma Blowfish

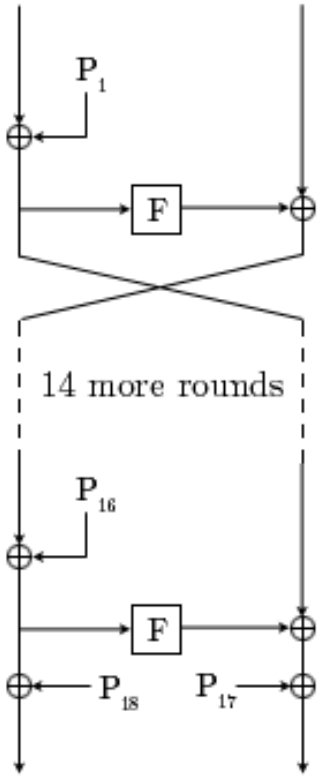
Blowfish adalah sebuah cipher block yang menggunakan variable-length key yang bekerja di 64-bit block. Algoritma yang ada terdiri dari dua bagian yaitu key-expansion dan data encryption.

Bagian key-expansion mengubah key yang ada menjadi beberapa array subkey. Sedangkan bagian data encryption terjadi di 16-round Feistel network yang ada. Setiap round terdiri dari permutasi key-dependent dan juga substitusi key dan data-dependent. Semua operasi yang digunakan adalah XOR dan additon dengan 32-bit.



Blowfish menggunakan subkey dalam jumlah banyak. Subkey ini didapatkan dari perhitungan sebelum dilakukan enkripsi atau dekripsi. Subkey inilah yang digenerate dengan menggunakan algoritma Blowfish (F Function) sebagai berikut :

1. Pertama dinisialisasi P-array yang terdiri dari 18 32-bit block dan juga 4 32-bit S-box.
2. Kemudian P1 di XOR kan dengan 32 bit pertama dari key yang ada, P2 dengan 32 bit yang kedua, dan seterusnya sampai seluruh panjang key (mungkin mencapai P14 atau 448 bit).
3. Enkripsi dilakukan dengan menggunakan subkeys yang digenerate pada step 1 dan 2
4. P1 dan P2 digantikan dengan hasil dari step 3.
5. Enkripsi P3 dengan Blowfish algorithm menggunakan subkey yang telah dimodifikasi.
6. Kemudian P4 dan P5 diganti dengan hasil dari step 5.
7. Proses terus dilanjutkan hingga semua nilai di P array tergantikan, dibantu dengan algoritma di S-box.
8. Bila ditotalkan dibutuhkan 521 iterasi untuk mengenerate semua subkey. Aplikasi diusahakan dapat menyimpan subkey untuk meningkatkan performansi.



Sementara untuk proses enkripsi di Feistel network, Blowfish mempunyai 16 round. Input yang dimasukkan berupa 64-bit block yang kemudian dibagi dua menjadi dua 32-bit block, yang bisa kita sebut leftblock dan rightblock.

Setelah itu, sebanyak 16 kali akan dilakukan :

leftblock = leftblock XOR Pi

berarti leftblock yang baru dihasilkan dari leftblock yang di XOR kan dengan P-array yang ke i. Kemudian

rightblock = F(rightblock) XOR rightblock

berarti rightblock baru dihasilkan dari rightblock yang di XOR kan dengan rightblock yang telah memasuki algoritma Blowfish (F function)

Terakhir dilakukan pertukaran antara leftblock dan rightblock. Setelah round ke 16, leftblock dan right block ditukar kembali. Kemudian dilakukan

rightblock = rightblock XOR P17
leftblock = leftblock XOR P18

Setelah itu leftblock dan rightblock dikombinasikan untuk mendapatkan ciphertext.

Dekripsi yang dilakukan identik dengan proses enkripsi. Hanya saja urutan P-array yang digunakan terbalik. Implementasi dari Blowfish yang membutuhkan kecepatan tinggi harus dipastikan melalui penyimpanan subkey-subkey yang digenerate di cache.

5. Pengujian

Dilakukan beberapa uji coba terhadap kedua algoritma di atas diantaranya dengan :

- Uji Coba 1
  Pada uji kali ini dilakukan enkripsi dan dekripsi pada file text berukuran relatif kecil (2 KB)

- Uji Coba 2
  Pada uji kali ini dilakukan penghapusan beberapa bit pada hasil enkripsi kemudian dilakukan dekripsi untuk melihat perubahannya.

- Uji Coba 3
  Pada uji kali ini dilakukan penambahan beberapa bit pada hasil enkripsi kemudian dilakukan dekripsi untuk melihat perubahannya.

Dari uji coba 1, waktu yang didapat tidak jauh berbeda yaitu sama-sama dibawah 1 detik. Pada uji coba ini tidak dapat terlihat begitu banyak perbedaannya. Mungkin performansi dari kedua algoritma ini baru terlihat ketika digunakan pada file dengan ukuran lebih besar. Namun uji coba tersebut belum dapat dilakukan.
Untuk uji coba yang kedua, bagian block sebelum bit yang dihilangkan tidak mengalami perubahan ketika didekripsi ulang. Namun block setelahnya mengalami perubahan menjadi urutan block baru yang acak.
Pada uji coba ketiga, block sebelum bit yang ditambahkan tidak mengalami perubahan tetapi pada block setelahnya mengalami perubahan menjadi urutan block baru yang acak.

Dari hasil uji coba kedua dan ketiga yang sama pada kedua , dapat dianalisis bahwa hal ini terjadi karena perubahan kunci yang digunakan untuk enkripsi atau dekripsi bit yang seharusnya. Hal tersebut mengakibatkan perubahan untuk bit-bit selanjutnya.

6. Kesimpulan

Dari hasil studi dan perbandingan algoritma Lucifer dan Blowfish, didapatkan kesimpulan berikut :

- Lucifer algorithm merupakan algoritma block cipher pertama yang menggunakan Feistel network, yang proses kerjanya menggunakan bit-rotation, substitution, dan permutation
- Blowfish algorithm merupakan pengganti DES yang juga menggunakan Feistel network, yang memiliki performa tinggi dengan penggunaan subkey yang banyak untuk proses enkripsi.
- Pengurangan ataupun penambahan block cipher semu akan menyebabkan perubahan informasi pada keseluruhan pesan

7. Daftar Pustaka

[1] Munir, Rinaldi. (2009). Bahan Kuliah IF3058 Kriptografi. Program Studi Informatika, Institut Teknologi Bandung.

[2] *http://www.schneier.com/blowfish.html*

[3] *http://www.quadibloc.com*

[4]http://www.kremlinencrypt.com/**algorithms**.htm

[5] http://ww.fuseki.com