FC – Algorima Block Cipher Baru

Muhammad Nassirudin
Sekolah Teknik Elektro dan Informatika (STEI)
Institut Teknologi Bandung, ITB
Bandung, Indonesia
13511044@std.stei.itb.ac.id

Mohamad Rivai Ramandhani
Sekolah Teknik Elektro dan Informatika (STEI)
Institut Teknologi Bandung, ITB
Bandung, Indonesia
13511043@std.stei.itb.ac.id

***Abstrak*—Pertukaran informasi penting perlu disertai dengan suatu metode keamanan untuk menghindari penggunaan yang tidak berwenang. Pada makalah ini diajukan sebuah algoritma kriptografi *block cipher* baru bernama Friendzone Cipher (FC) sebagai salah satu metode keamanan tersebut. Hasil eksperimen dan analisis menunjukkan algoritma yang diajukan mampu melakukan enkripsi dan dekripsi pesan dengan benar serta memiliki tingkat keamanan yang cukup baik.**

Kata kunci—block cipher; kriptografi; pertukaran informasi

I. PENDAHULUAN

Saat ini, pertukaran data dan informasi sangat mudah dilakukan terutama berkat adanya Internet. Berbagai informasi dikirimkan dalam volume yang sangat besar setiap harinya. Sebuah pesan berisi informasi yang penting tentu perlu dijaga dari penggunaan yang tidak berwenang. Perlindungan terhadap pesan-pesan tersebut membuat kebutuhan atas kriptografi semakin meningkat. Pesan perlu dipertukarkan dalam keadaan terenkripsi sehingga keamanannya terjaga.

Pada makalah ini diajukan sebuah rancangan *block cipher* yang disebut Friendzone Cipher. Rancangan yang diajukan terinspirasi dari algorima Lucifer/DES [1]. Fitur utama dari rancangan algoritma yang diajukan ini adalah penggunaan sejumlah relatif prima dari sebuah bilangan untuk menyamarkan pola.

Algoritma Friendzone meningkatkan keamanannya dengan menggunakan kunci yang memiliki panjang 128-bit (dengan panjang efektif 96-bit) daripada algoritma Lucifer yang hanya menggunakan kunci sepanjang 64-bit (dengan panjang efektif 56-bit). Selain itu, algoritma Lucifer merupakan algoritma yang sudah tua (diterbitkan sejak 1977), dan dengan semakin berkembangnya teknologi tentu akan semakin besar kemungkinan untuk bisa memecahkan keamanan algoritma ini .

II. DASAR TEORI

A. *Prinsip Confusion dan Diffusion*

Prinsip *confusion* adalah prinsip untuk menyembunyikan hubungan antara *plaintext*, *ciphertext*, dan kunci sehingga informasi statistik sulit dianalisis. Salah satu caranya adalah dengan melakukan substitusi yang kompleks.

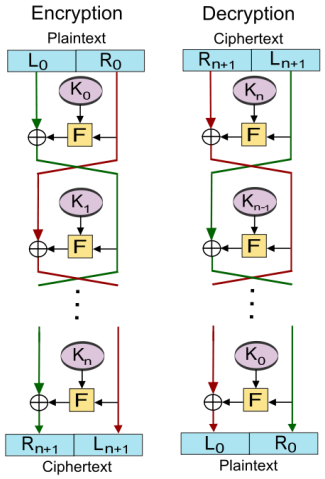
Prinsip *diffusion* adalah prinsip untuk menyebarkan pengaruh perubahan 1 bit dalam *plaintext* atau kunci ke sebanyak mungkin *ciphertext*. Salah satu caranya adalah dengan melakukan permutasi.

B. *Cipher Berulang*

Proses transformasi *plaintext* menjadi *ciphertext* harus dapat diulang/iterasi. Setiap iterasi mengkombinasikan hasil dari proses sebelumnya dengan sebuah kunci yang unik.

C. *Jaringan Feistel*

Skema Jaringan Feistel membuat proses enkripsi menjadi *reversible* sehingga tidak memerlukan algoritma baru untuk dekripsi. Selain itu, *round function* dalam Jaringan Feistel tidak harus *reversible* sehingga dapat dibuat kompleks. Skema Jaringan Feistel dapat dilihat pada Gambar 1.



Gambar 1. Skema Jaringan Feistel [2]

D. *Kotak-S*

Kotak-S merupakan sebuah matriks yang digunakan untuk melakukan substitusi. Biasanya, substitusi dilakukan untuk mengubah ukuran bit dari masukan.

III. RANCANGAN *BLOCK CIPHER*

Algoritma yang dirancang menerima masukan berupa pesan berukuran berapa pun serta sebuah kunci rahasia dengan panjang 128-bit. Proses pembangkitan kunci internal dan enkripsi pesan masukan dirancang sebagai berikut.

A. *Pembangkitan Kunci Internal*

1) Kunci internal pertama dibangkitkan dari kunci masukan pengguna sedangkan kunci internal berikutnya dibangkitkan dari kunci internal sebelumnya.

2) Misalkan K(n) menyatakan kunci sekarang dan K(n+1) menyatakan kunci yang akan dibangkitkan. K(n) dipisah menjadi 16 bagian, setiap bagian merepresentasikan sebuah bilangan berukuran 8-bit. Setiap bilangan disubstitusi MSB-nya dengan 1 dan LSB-nya dengan 0. Misalkan ke-16 bilangan tersebut adalah N(0), N(1), N(2), sampai N(15).

3) Untuk setiap bilangan yang didapatkan dari tahap sebelumnya, N(i), 0 <= i <= 15, diambil 16 bilangan terbesar yang kurang dari N(i) dan relatif prima terhadap N(i). Ke-16 bilangan tersebut diletakkan dalam matriks M berukuran 16 X 16 dengan bilangan pertama pada baris pertama atau posisi (i,0); bilangan kedua (i,1); bilangan ketiga (i,2); dan bilangan keempat (i,3), dan seterusnya sampai terisi baris ke-16.

4) Dilakukan operasi XOR *bitwise* dari elemen-elemen matriks secara diagonal untuk mendapatkan 16 bilangan baru masing-masing berukuran 8 bit. Bilangan pertama adalah hasil dari M(0,0) ^ M(1,1) ^ M(2,2) ^ M(3,3) ^ … ^ M(15,15); bilangan kedua M(0,1) ^ M(1,2) ^ M(2,3) ^ M(3,4) ^ … ^ M(15,0); dan seterusnya.

5) Ke-16 bilangan di-*shift* secara *wrapped* dengan arah *shift* berganti-gantian kiri-kanan (bilangan pertama di-*shift* ke kiri, bilangan kedua di-*shift* ke kanan, bilangan ketiga di-*shift* ke kiri, dan seterusnya) dan dengan banyaknya pergeseran pada setiap bilangan yaitu { 1,2,1,3,2,1,2,2,1,1,2,3,1,1,1,2}.

6) K(n+1) merupakan kunci internal baru berukuran 128 bit yang merupakan hasil *append* dari ke-16 bilangan dari hasil tahap sebelumnya secara berurutan.

7) Proses diulang terus hingga mendapatkan 16 kunci internal.

B. *Enkripsi Pesan*

1) Skema enkripsi pesan menggunakan Jaringan Feistel.

2) Pesan dibagi-bagi ke dalam blok-blok masing-masing berukuran 128 bit.

3) Jika ada blok yang berukuran kurang dari 128 bit, bit dilengkapi dengan cara menambahkan bit 0 di belakang.

4) Setiap blok pesan dibagi menjadi dua bagian berukuran sama (64 bit), L(0) dan R(0).

5) Pada setiap iterasi i, 1 <= i <= 16, dibangkitkan L(i) = R(i-1) dan R(i) = L(i-1) ^ f(K(i), R(i-1)).

6) Round function f didefinisikan sebagai berikut

a) Posisi bit kelipatan 4 (4, 8, 16, ..., 128) dari kunci K(i) dibuang sehingga hanya tersisa 96 bit.

b) Menggunakan tabel yang telah didefinisikan sebelumnya, bit-bit dalam R(i-1) dipermutasi dan diduplikasi sehingga menjadi 96 bit.

c) Hasil dari a di-XOR-kan dengan hasil dari b.

d) Hasil dari c dipartisi menjadi 16 kelompok, masing-masing sepanjang 6 bit.

e) Untuk 0 <= j <= 15, kelompok ke-i disubstitusi dengan biner berukuran 4 bit menggunakan Kotak-S S(i) yang telah didefinisikan.

f) Hasil dari e di-append secara berurutan dan menghasilkan sebuah cipherteks berukuran 64-bit.

7) Pada akhir iterasi, L(16) dan R(16) di-append untuk mendapatkan blok pesan yang telah terenkripsi, ulangi untuk setiap blok pesan.

IV. EKSPERIMEN DAN PEMBAHASAN HASIL

Eksperimen dilakukan terhadap pesan teks untuk ketiga modus operasi yang diimplementasikan. Potongan pesan yang dipakai dapat dilihat pada Tabel I.

TABLE I. HASIL ENKRIPSI DALAM MODUS ECB

REFERENCES

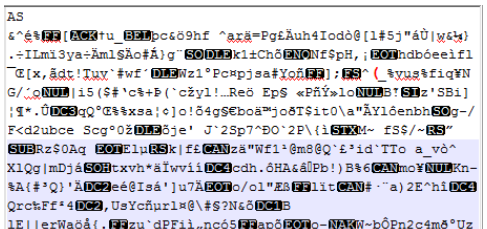
The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first ...”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the reference list. Use letters for table footnotes.

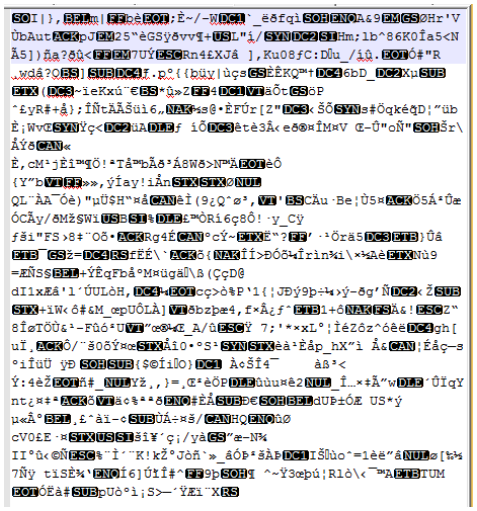
Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

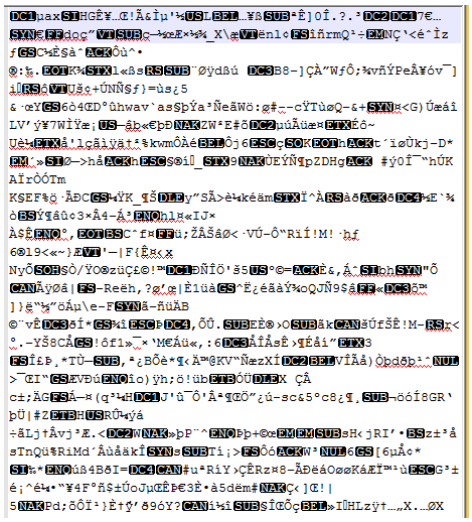
Potongan hasil enkripsi untuk modus ECB, CBC, dan CFB 8-bit masing-masing ditampilkan pada Gambar 2, Gambar 3, dan Gambar 4.



Gambar 2. Potongan Enkripsi dengan Modus ECB



Gambar 3. Potongan Enkripsi dengan Modus CBC

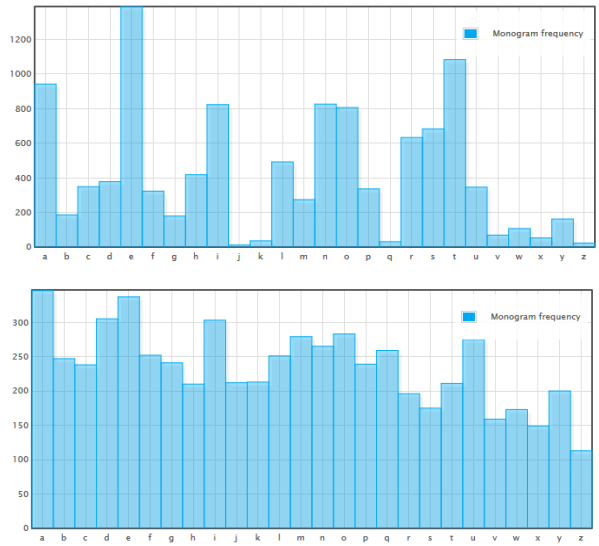


Gambar 4. Potongan Enkripsi dengan Modus CFB 8-bit

Berdasarkan hasil yang didapat, pesan asli tidak terkenali lagi sehingga algoritma yang dirancang mampu mengenkripsi pesan. Selain itu, proses dekripsi juga mengembalikan pesan semula.

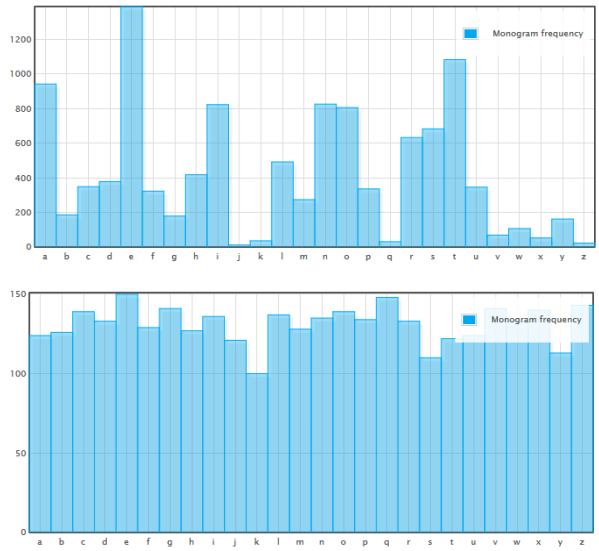
V. ANALISIS KEAMANAN

Analisis keamanan dilakukan dengan membandingkan frekuensi kemunculan huruf alphabet antara plainteks dengan cipherteks. Frekuensi kemunculan kata ditampilkan dalam representasi grafik batang untuk mempermudah pembandingan. Perbandingan frekuensi kemunculan huruf dari pesan asli dan cipherteks dengan modus ECB dapat dilihat pada Gambar 5.



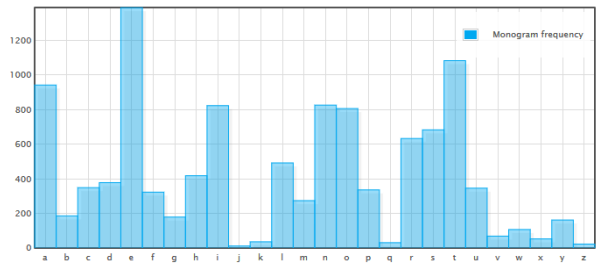
Gambar 5. Frekuensi Kemunculan Huruf Plainteks dengan Modus ECB

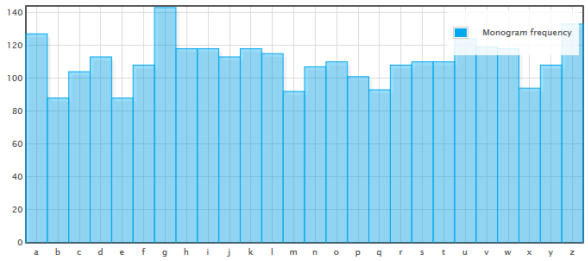
Sementara itu, perbandingan frekuensi kemunculan huruf plainteks dengan cipherteks dengan modus CBC dapat dilihat pada Gambar 6.



Gambar 6. Frekuensi Kemunculan Huruf Plainteks dengan Modus CBC

Perbandingan frekuensi kemunculan huruf plainteks dengan cipherteks dengan modus CFB 8-bit dapat dilihat pada Gambar 7.





Gambar 7. Frekuensi Kemunculan Huruf Plainteks dengan Modus CFB 8-bit

Seperti yang dapat dilihat pada ketiga perbandingan tersebut, cipherteks yang dihasilkan dengna modus ECB masih memiliki kemiripan distribusi frekuensi huruf. Hal tersebut dapat jelas terlihat dari bentuk visualisasi diagram batang yang disajikan. Berbeda dengan ECB, baik pada modus CBC maupun pada modus CFB 8-bit bentuk grafik sudah mulai terlihat lebih baik. Hal tersebut dapat terlihat jelas dengan ratanya distribusi huruf.

Dengan meratanya distribusi kemunculan huruf, penyerang akan kesulitan untuk melakukan deduksi menggunakan analisis frekuensi kemunculan huruf [3]. Oleh karena itu, algoritma yang diajukan memiliki keamanan yang baik.

VI. KESIMPULAN DAN SARAN

Berdasarkan hasil eksperimen dan analisis, dapat disimpulkan algoritma yang diajukan berhasil mengenkripsi dan menghilangkan hubungan frekuensi huruf dari plainteks. Rancangan algoritma yang sudah ada dapat dikembangkan lebih lanjut, salah satunya dengan membuat panjang kunci yang digunakan menjadi lebih fleksibel (tidak harus 128-bit).

REFERENSI

[1] Data Encryption Standard, Federal Information Processing Standards Publication (FIPS PUB) 46, National Bureau of Standards, Washington, DC (1977).

[2] http://upload.wikimedia.org/wikipedia/commons/thumb/f/fa/Feistel_cipher_diagram_en.svg/500px-Feistel_cipher_diagram_en.svg.png, diakses pada 18 Maret 2015.

[3] Rinaldi Munir, Security Analysis of Selective Image Encryption Algorithm Based on Chaos CBC-like Mode. Internstional Conference on Telecommunication Systems, Services, and Applications, 2012.