SnR - Swap and Round Block Cipher

A New Block Cipher Algorithm

Vincent Theophilus Ciputra (*13513005*)
Teknik Informatika
Institut Teknologi Bandung
Bandung, Indonesia
vincent.theophilusc@gmail.com

Edwin Wijaya (*13513040*)
Teknik Informatika
Institut Teknologi Bandung
Bandung, Indonesia
edwinwijaya1994@gmail.com

***Abstrak*—Di era informasi ini, pengiriman dan penerimaan pesan memerlukan protokol keamanan yang terjamin agar pesan yang bersifat rahasia tidak dapat diakses oleh pihak yang tidak memiliki akses . Pada makalah ini, diajukan sebuah algoritma kriptografi *block cipher* yang baru dengan nama *swap and round algorithm* yang menggunakan blok pesan sepanjang 128-bit dan kunci yang panjangnya 128-bit juga. Untuk membuat algoritma yang sulit dipecahkan maka algoritma ini menggunakan operasi seperti permutasi, substitusi, *circular shift*, dan XOR. Selain itu, kunci internal yang digunakan pada setiap *round function* di jaringan feistel dibentuk secara kompleks agar sulit untuk dipecahkan. Hasil eksperimen dan analisis yang dilakukan menunjukkan bahwa algoritma ini mampu melakukan enkripsi dan dekripsi pesan dengan tingkat keamanan yang cukup baik.**

Kata Kunci—block cipher; dekripsi; enkripsi; kunci; swap and round algorithm

I. PENDAHULUAN

Manusia tidak akan terlepas dari yang namanya "komunikasi" dengan manusia yang lain. Dengan komunikasi, manusia dapat saling bertukar informasi. Sebagai contoh, komunikasi antar teman, antar suami dan istri, antar murid dan guru, antar mahasiswa dengan dosennya, dan sebagainya. Komunikasi merupakan suatu hal yang sangat penting dan pasti dilakukan oleh setiap manusia. Di antara informasi-informasi tersebut, pastilah ada informasi yang bersifat rahasia, artinya hanya boleh diketahui oleh orang-orang tertentu saja. Akan sangat berbahaya apabila rahasia-rahasia tersebut diketahui oleh pihak lain, apalagi pihak tersebut adalah musuh yang berusaha untuk mencari kekurangan dan kelemahan suatu pihak.

Dalam dunia komputer, peranan ini dipegang oleh kriptografi yang sangat diperlukan baik untuk menyimpan maupun mengirimkan pesan. Algoritma kriptografi dilakukan untuk mengenkripsi pesan agar keamanannya terjamin dan pesan yang dienkripsi tetap rahasia dan hanya dapat diakses oleh orang-orang tertentu. Proses dalam kriptografi dilakukan dengan cara mengenkripsi pesan dengan suatu kunci dan pesan yang sudah terenkripsi menjadi sebuah cipherteks dan hanya bisa diakses oleh orang yang memiliki kunci tersebut untuk mendekripsi cipherteks menjadi sebuah plainteks kembali.

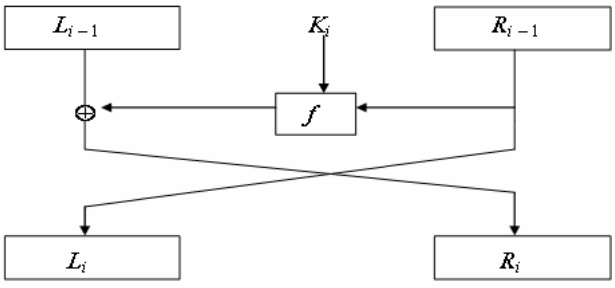
Seiring dengan pesatnya perkembangan teknologi informasi, ancaman keamanan informasi rahasia yang dikirimkan melalui jalur internet dan jaringan seluler menjadi semakin tinggi dan meresahkan. Oleh karena itu, perkembangan kriptografi juga selalu mengikuti bersamaan dengan meningkatnya ancaman tersebut. Jenis cipher yang biasanya dipakai dalam proses enkripsi dan dekripsi adalah *block cipher* dan *stream cipher*. Dalam *block cipher*, proses enkripsi dan dekripsi dilakukan dengan membagi pesan menjadi beberapa blok bit yang panjang suatu bloknya sudah ditentukan.

Untuk mengatasi hal tersebut, dalam makalah ini akan dipaparkan algoritma *block cipher* yang baru yaitu *swap and round algorithm* dengan harapan agar algoritma ini dapat menjaga pesan semaksimal mungkin. Untuk meningkatkan kekuatan enkripsi pada algoritma ini dilakukan metode *confusion* dan *diffusion* dan juga jaringan Feistel. Jaringan feistel pada algoritma ini menggunakan prinsip XOR dan membagi plainteks ke dalam blok berukuran 16 bit.

II. DASAR TEORI

A. Jaringan Feistel

Jaringan feistel merupakan suatu model jaringan yang yang simetris dan digunakan dalam *block* cipher. Jaringan feistel ini membuat algoritma kriptografi menjadi *reversible* atau dengan kata lain proses untuk mengenkripsi dan mendekripsi pesan hampir sama. Oleh karena itu algoritma untuk mendekripsi cipherteks tidak perlu dibuat berbeda.



Gambar 1. Jaringan Feistel

Dalam jaringan feistel, yang dilakukan adalah sebagai berikut:

1. Bagi blok pesan dengan ukuran n bit menjadi dua bagian yang sama panjangnya yaitu Li dan Ri.
2. Untuk setiap n putaran lakukan:

   $L_{i+1} = R_i$

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

   Keterangan:

   n = jumlah putaran

   Ki = kunci internal pada putaran ke-i

   F = fungsi feistel

   Li = bagian kiri pada plainteks

   Ri = bagian kanan pada plainteks
3. Untuk melakukan dekripsi, ulangi langkah 1 dan 2 untuk ciphertext, dengan

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

   $R_{i+1} = L_i$

B. Prinsip Confusion dan Diffusion

Prinsip *confusion* adalah menyembunyikan hubungan apapun yang ada antara plainteks, cipherteks, dan kunci. Kunci yang ada dibuat dengan tidak berhubungan apapun dengan cipherteks. Setiap karakter dalam cipherteks seharusnya bergantung pada beberapa bagian dari kunci tersebut. Hal ini akan membuat seorang kriptanalis sulit untuk mencari pola-pola statistik yang muncul pada cipherteks.

Prinsip *diffusion* adalah menyebarkan pengaruh satu bit plainteks atau kunci ke sebanyak mungkin cipherteks. Jika kita mengubah suatu karakter dalam plainteks, maka beberapa karakter pada cipherteks berubah juga, dan juga sebaliknya. Hal ini akan menyembunyikan hubungan statistik antara plainteks, cipherteks, dan kunci karena plainteks disebarkan ke beberapa karakter dalam cipherteks, artinya membutuhkan lebih banyak cipherteks untuk melakukan serangan dengan melihat statistik.

C. Circular Shift

Circular Shift adalah metode yang digunakan untuk mengubah urutan bit. *Circular Shift* dibagi menjadi 2 cara yaitu *circular left shift* dan *circular right shift*. *Circular left shift* menggeser sejumlah bit ke kiri dan bit paling kiri digeser ke paling kanan, misalnya bit 01101011 jika dilakukan *circular left shift* menjadi 11010110. Begitu juga sebaliknya untuk *circular right shift*, bit 01101011 menjadi 10110101 karena bit paling kanan digeser ke paling kiri dan bit yang lainnya digeser ke kanan. *Circular Shift* ini dilakukan untuk membuat algoritma enkripsi menjadi lebih rumit karena mengubah bit awal menjadi bit yang berbeda.

D. S-Box

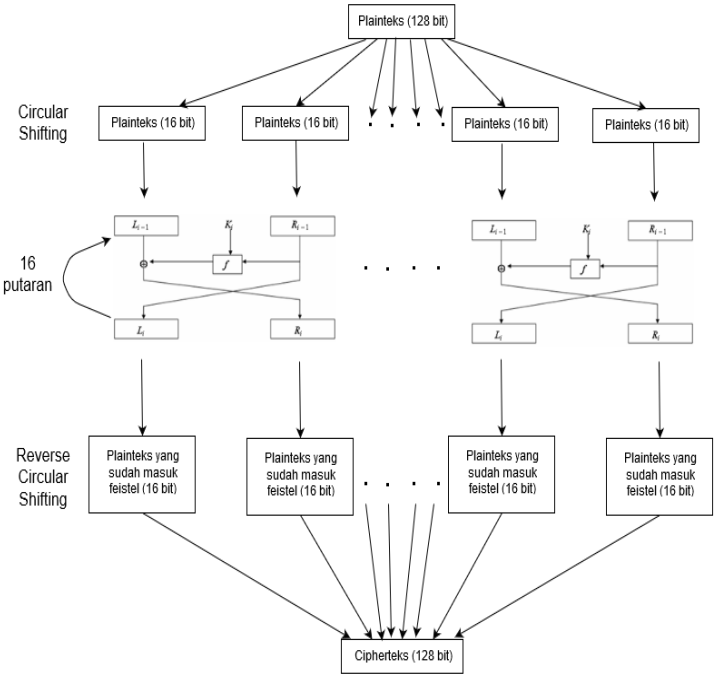
S-Box adalah matriks yang berisi substitusi sederhana yang memetakan satu atau lebih bit dengan satu atau lebih bit yang lain. Untuk membuat *S-Box* yang bagus adalah dengan:

1. Membuat fungsi komponen yang seimbang.
2. Derajat aljabar tinggi.
3. Memenuhi SAC.
4. Fungsi komponen tidak linear.

III. RANCANGAN BLOCK CIPHER

Swap and Round adalah block cipher yang memiliki ciri khas berupa fungsi transposisi yang menggunakan teknik pertukaran (swap) dan putaran (round) dengan terlebih dahulu merepresentasikan blok-blok bit ke dalam matriks. Ukuran plainteks yang diterima bisa berukuran berapa pun dan akan dibagi menjadi blok-blok pesan berukuran 128-bit serta kunci eksternal berukuran 128-bit. Dari kunci eksternal tersebut akan dibentuk kunci internal sebanyak 16 dan masing-masing berukuran 16 bit.

A. Skema Algoritma



Gambar 2. Skema Algoritma

1. Awalnya plainteks yang berupa blok pesan berukuran 128-bit dibagi menjadi 8 upa-blok berukuran 16-bit.
2. Lakukan permutasi awal sebagai berikut : lakukan circular-shifting ke kanan pada blok 16 bit. Kemudian lakukan circular shift ke kiri untuk masing-masing 8 bit pertama dan 8 bit terakhir. Ulangi shifting untuk

setiap 4 bit (shift kanan) juga 2 bit (shift kiri). Misal blok 16 bit

0	1	0	1	1	0	1	1	0	0	1	1	1	0	1	1

Circular-shifting ke kanan pada 16 bit menjadi:

1	0	1	0	1	1	0	1	1	0	0	1	1	1	0	1

Circular-shifting ke kiri untuk masing-masing 8-bit menjadi:

0	1	0	1	1	0	1	1	0	0	1	1	1	0	1	1

Circular-shifting ke kanan untuk masing-masing 4-bit menjadi:

1	0	1	0	1	1	0	1	1	0	0	1	1	1	0	1

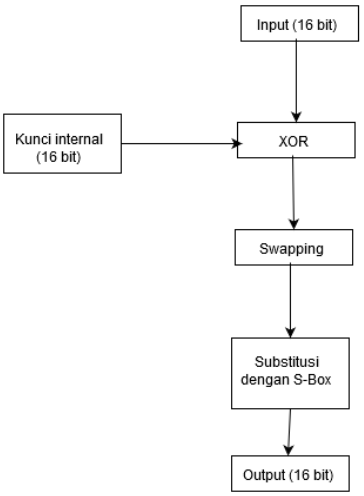
Circular-shifting ke kiri untuk masing-masing 2-bit menjadi:

0	1	0	1	1	1	1	0	0	1	1	0	1	1	1	0

3. Masukkan setiap 2 upa-blok, misalnya Li dan Ri ke dalam sebuah jaringan feistel sehingga ada 4 jaringan feistel.
4. Pada setiap round function yang dilakukan sebanyak 16 kali:
   a. Lakukan XOR terhadap Ri dengan kunci internal yang sudah dibangkitkan.
   b. Bentuk Ri ke dalam matriks 4x4. Lakukan transposisi dengan kunci internal sebagai masukan. Untuk setiap bit genap <= 11 dari kunci (mulai dari 0) dilakukan pertukaran baris matriks hanya jika bit bernilai 1. Sedangkan untuk bit ganjil <= 11 dari kunci, dilakukan pertukaran kolom hanya jika bernilai 1. Untuk menentukan pasangan baris / kolom yang akan ditukar menggunakan semua kombinasi pasangan baris / kolom yang mungkin (sejumlah 4C2 = 6 untuk baris dan 6 untuk kolom) sehingga ada 12 pertukaran. Untuk 4 bit terakhir dari kunci maka akan dilakukan pertukaran elemen sudut dari matriks hanya jika bit kunci bernilai 1 dengan pasangan (ujung kiri atas, ujung kanan atas), (ujung kanan atas, ujung kanan bawah), (ujung kanan bawah, ujung kiri bawah), (ujung kiri bawah, ujung kiri atas).
   c. Lakukan substitusi dengan S-Box yang sudah terdefinisi. Substitusi dilakukan dengan mengambil setiap 4 bit pesan, 2 bit pertama memetakan indeks baris, 2 bit berikutnya memetakan indeks kolom. Kemudian 4 bit pesan tersebut disubstitusi dengan 4 bit pesan pada S-Box berukuran 4 x 4 yang sudah terdefinisi sebagai berikut :

1010	1001	1100	0001
0010	0101	1101	0110
0010	0000	0111	1011
0100	1111	0011	1110

S-Box berukuran 4 x 4 yang sudah didefinisikan



Gambar 3. Skema round function

5. Lakukan permutasi invers berdasarkan langkah 2.
6. Gabungkan semua upa-blok hasil keluaran dari round function menjadi blok berukuran 128 bit kembali. Blok ini menjadi cipherteks.

B. Pembangkitan Kunci Internal

1. Bagi blok menjadi 16 bagian, masing-masing berukuran 8 bit. Untuk menambah kompleksitas bit kunci, lakukan xor terhadap setiap kunci dengan bitset “10101010” untuk kunci dengan indeks ganjil atau “01010101” untuk kunci dengan indeks genap.
2. Bangun 8 buah matriks 4x4 dengan matriks ke-i berisi bit ke-i dari setiap potongan blok plainteks.
3. Untuk setiap matriks, akan dihasilkan sejumlah bit dengan melakukan operasi XOR untuk setiap bit dalam baris, juga dilakukan hal yang sama untuk setiap kolom. Pada proses ini akan menghasilkan 8 bit (4 bit dari baris, 4 bit dari kolom).
4. Selanjutnya putar elemen terluar dari matriks (M[0,0], M[0,1], M[0,2], M[0,3], M[1,3], M[2,3], M[3,3], M[3,2], M[3,1], M[3,0], M[2,0], M[1,0]) satu kotak ke samping searah jarum jam.

Ilustrasi proses 4:

Matriks awal:

1	1	1	0
0	1	1	0
0	0	1	1
0	1	1	0

Matriks akhir setelah dilakukan perputaran elemen terluar:

0	1	1	1
0	1	1	0
0	0	1	0
1	1	0	1

5. Lakukan proses 3 kembali sehingga kembali menghasilkan 8 bit. Sekarang kita mempunyai 16 bit yang kemudian dijadikan kunci internal 1.
6. Lakukan langkah 4, kemudian lakukan langkah 3-5 kembali untuk matriks ke-1 sehingga kita kembali mendapatkan 16 bit yang dijadikan kunci internal 2. Dalam hal ini kita mendapatkan 2 buah kunci internal dari sebuah matriks.
7. Lakukan langkah 3 sampai 6 untuk 7 matriks lainnya, sehingga kita mendapatkan 16 buah kunci internal yang berbeda.

IV. Eksperimen dan Pembahasan Hasil

Pada bagian ini akan dibahas hasil implementasi block cipher menggunakan bahasa pemrograman C++. Pengujian dilakukan dengan melakukan enkripsi dan dekripsi dari suatu pesan dan kunci yang diberikan untuk mengetahui hasil enkripsi dari plainteks tersebut dan dapat didekripsi kembali menjadi pesan yang asli. Hal tersebut bertujuan untuk mengamankan pesan. Kode implementasi dapat diakses pada pranala : https://github.com/edwinwijaya94/swap-round-cipher. Berikut adalah contoh hasil enkripsi dan dekripsi yang dilakukan:

Key: n9#56*)?|`hjn7$@
Plaintext : Lorem Ipsum is simply dummy text of the printing and typesetting industry.
Ciphertext : ¬j¼æñƒ
ü<π│ü■|╥;\UV▒ÆH^$FFÿ☻4Cîí_R╓O5D*┴╨Θ▄sù↓^)Ö╟ª
←h#3Tµ⌐→sù↓^Öq4sƒ)τ*╢P▌ ? a≡Q

Dari hasil enkripsi di atas dapat diamati bahwa cipherteks yang dihasilkan benar–benar *random* sehingga tidak mungkin bagi kriptanalis untuk melakukan analisis frekuensi. Ciphertext yang *random* ini antara lain karena adanya penggunaan round function secara berulang, sehingga setiap iterasi, cipherteks yang dihasilkan semakin sulit dihubungkan dengan plainteks. Hal ini sesuai dengan prinsip *confusion*.

Selanjutnya, pengujian yang dilakukan adalah dengan mengenkripsi pesan–pesan dalam ukuran yang lebih besar untuk menghitung waktu eksekusi yang dibutuhkan. Berikut perbandingan waktu eksekusi *swap and round cipher* dengan DES :

Ukuran Pesan	DES	Swap and Round Cipher
1 KB	47 ms	156 ms
2 KB	74 ms	336 ms
4 KB	155 ms	673 ms

Dari table perbandingan waktu pemrosesan, dapat diamati bahwa *swap and round cipher* membutuhkan waktu yang lebih lama dibandingkan DES. Hal ini antara lain disebabkan oleh perbedaan panjang kunci. DES beroperasi pada blok pesan 64 bit sedangkan *swap and round cipher* beroperasi pada blok pesan berukuran 128 bit sehingga memerlukan waktu pemrosesan yang lebih lama. Semakin panjang pesan maka semakin lama waktu yang dibutuhkan. Rancangan block cipher ini menggunakan komputasi seperti permutasi dan substitusi dalam bentuk bit yang jumlahnya tidak sedikit, karena itu waktu yang diperlukan pun akan berbanding lurus dengan banyaknya komputasi.

V. Analisis Keamanan

Analisis keamanan dilakukan berdasarkan kompleksitas dari algoritma ini sendiri. Semakin panjang kunci yang digunakan maka kompleksitasnya akan semakin meningkat. Beberapa cara menyerang algoritma ini adalah sebagai berikut:

A. *Analisis Frekuensi*

Salah satu serangan yang biasa dilakukan dalam kriptografi adalah serangan tanpa mengetahui kunci untuk memecahkan cipherteks. Biasanya dilakukan dengan menggunakan metode analisis frekuensi. Metode frekuensi analisis menghitung seluruh kemunculan suatu karakter dalam cipherteks dan yang memiliki jumlah kemunculan terbanyak akan disamakan dengan karakter yang tersering muncul dalam suatu bahasa, misalnya dalam bahasa inggris, huruf paling sering muncul adalah karakter ‘E’.

Dengan menggunakan algoritma *Swap and round* ini, suatu karakter dalam plainteks akan dienkripsi menjadi karakter yang berbeda-beda tergantung dari kuncinya. Karena itu cipherteks hasil dari *swap and round algorithm* tidak akan dapat didekripsi dengan mudah jika menggunakan frekuensi analisis.

B. *Brute-force Attack*

Serangan lain yang biasa dilakukan menggunakan serangan *brute force.* Serangan *brute force* adalah serangan dengan mencari kunci satu per satu sampai didapatkan kunci yang benar. Di bawah ini adalah tabel waktu yang diperlukan untuk mendekripsi suatu cipherteks menggunakan *brute force* :

Panjang Kunci	Banyak kemungkinan Kunci	Waktu yang diperlukan (10^6 percobaan per detik)
16 bits	$2^{16} = 65536$	32.7 milidetik
32 bits	$2^{32} = 4.3 \times 10^9$	35.8 menit
56 bits	$2^{56} = 7.2 \times 10^{16}$	1142 tahun
128 bits	$2^{128} = 4.3 \times 10^{38}$	5.4×10^{24} tahun

VI. KESIMPULAN DAN SARAN

Kesimpulan yang didapat melalui makalah ini adalah dalam pengiriman pesan dibutuhkan suatu algoritma kriptografi untuk menjaga isi pesan tersebut. Dengan rancangan *swap and round algorithm* yang diusulkan dalam makalah ini, maka pengiriman pesan akan terjamin dan akan sulit untuk dipecahkan. Algoritma ini menggunakan prinsip *confusion* dan *diffusion* untuk menyembunyikan hubungan antara plainteks, cipherteks, dan kunci. Dalam algoritma ini juga digunakan suatu permutasi menggunakan *circular shifting* dan jaringan feistel agar tidak diperlukan algoritma baru untuk mendekripsi cipherteks. Berdasarkan hasil implementasi algoritma tersebut yang dibuat menggunakan bahasa pemrograman C++, rancangan algoritma ini dapat melakukan enkripsi dan dekripsi dan waktu yang diperlukan bergantung atas jumlah proses komputasi yang dilakukan. Dengan algoritma ini, kunci yang digunakan akan sulit untuk dipecahkan walaupun menggunakan *brute force* maupun metode frekuensi analisis.

Saran yang dapat diberikan adalah sebaiknya dilakukan analisis dan eksperimen lebih lanjut agar dapat memastikan tingkat keamanan dari algoritma *swap and round block cipher* ini.

REFERENSI

[1] http://informatika.stei.itb.ac.id/~rinaldi.munir/

[2] http://www.nku.edu/~christensen/diffusionandconfusion

[3] https://ahriev.wordpress.com/kriptografi-duniaku/design-block-cipher/

[4] http://searchsecurity.techtarget.com/definition/block-cipher

[5] K. Elissa, “Title of paper if known,” unpublished.