MR Cipher

Ridho Pratama-13516032[1] Muhammad Abdullah Munir-13516074[2]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]p.ridho@yahoo.co.id [2]munirabdullahm@gmail.com

Abstrak—Algoritma MR adalah algoritma *block cipher* yang memiliki ukuran blok sebesar 128 bit. Algoritma ini menerapkan jaringan feistel dengan jumlah putaran sebanyak 8 kali. Operasi yang dilakukan pada algoritma MR sangat sederhana, yaitu: rotasi bit, substitusi, serta permutasi. Namun hasil enkripsi yang dihasilkan cukup aman berdasarkan analisis yang telah dilakukan. Selain itu, dengan menggunakan operasi yang sederhana algoritma ini memberikan waktu enkripsi dan dekripsi yang cepat. Sehingga cocok untuk digunakan pada perangkat dengan kemampuan komputasi yang rendah.

Kata Kunci—Cipher Blok; Kriptografi; Jaringan Feistel; Fungsi Putaran; S-Box AES;

I. Pendahuluan

Pada zaman yang telah berkembang saat ini, media yang dapat digunakan untuk berkomunikasi sangatlah beragam. Salah satunya adalah komunikasi digital. Pada komunikasi digital, informasi akan dikirimkan melalui jaringan internet. Akan tetapi, jaringan internet adalah saluran publik yang digunakan oleh orang. Dampaknya adalah semua orang dapat mengetahui pesan yang dikomunikasikan.

Upaya untuk menyelesaikan permasalahan ini adalah dengan adanya kriptografi. Dengan kriptografi, sebelum pesan dikirim akan dienkripsi terlebih dahulu. Sehingga pesan yang melalui jaringan internet berupa data yang tidak diketahui maknanya apabila tidak didekripsi. Kemudian pesan yang terenkripsi tadi akan didekripsi oleh penerima pesan.

Algoritma MR merupakan algoritma block cipher yang dikembangkan dengan operasi-operasi sederhana untuk meminimalkan waktu komputasi dalam enkripsi dan dekripsi. Algoritma MR juga menerapkan prinsip confusion dan diffusion dari Shannon serta jaringan Feistel.

II. Dasar Teori

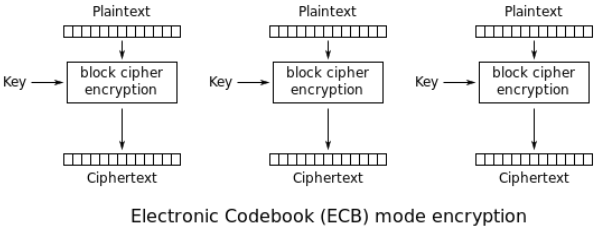
A. *Block Cipher*

Salah satu algoritma kriptografi modern dengan kunci simetris adalah *Block Cipher*. *Plaintext* yang akan di-enkripsi akan dibagi sesuai ukuran blok yang digunakan algoritma. Kemudian operasi kriptografi akan dilakukan terhadap blok *plaintext* dengan menggunakan kunci enkripsi. Panjang kunci enkripsi sesuai dengan panjang blok yang digunakan oleh algoritma. Serta dihasilkan *ciphertext* dengan panjang yang sama. Apabila panjang *plaintext* bukan kelipatan dari ukuran blok, maka akan ditambahkan *padding* sampai panjang pesan menjadi kelipatan panjang blok.

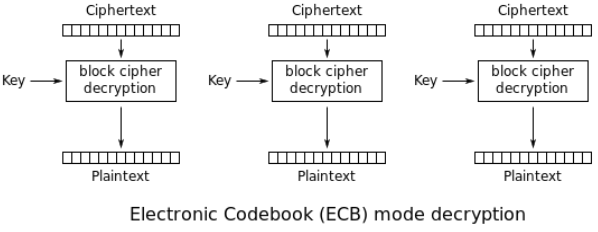
Block Cipher memiliki lima mode operasi. Hal ini berkaitan dengan cara blok dioperasikan pada proses enkripsi dan dekripsi. Lima mode operasi tersebut yaitu:

1. *Electronic Code Book* (ECB)

   Mode yang paling sederhana. Pada mode ini masing-masing blok akan dienkripsi secara independen. Sehingga tidak ada hubungan antar blok. Sehingga suatu blok *plaintext* yang sama akan menghasilkan *ciphertext* yang sama.



Gambar 1. Enkripsi pada mode ECB

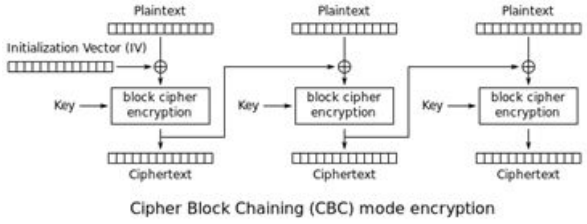


Gambar 2. Dekripsi pada mode ECB

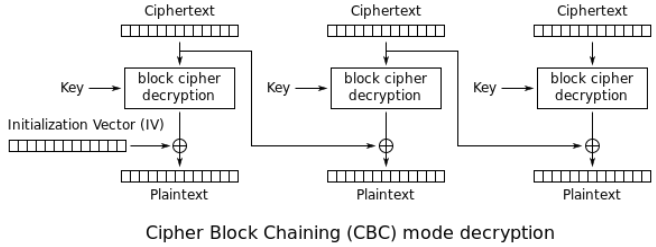
2. *Cipher Block Chaining* (CBC)

   Pada mode ini, ketika proses enkripsi sebuah blok diperlukan *ciphertext* blok sebelumnya. Oleh karena itu, pada operasi blok pertama akan diperlukan sebuah *Initialization Vector* (IV) yang dibangkitkan secara acak. Untuk proses enkripsi, plaintext akan di-XORkan dengan blok *ciphertext* sebelumnya terlebih dahulu sebelum dioperasikan dengan algoritma enkripsi. Kesalahan satu bit pada sebuah

blok *plaintext* akan mempengaruhi blok *ciphertext* saat itu dan berikutnya. Pada proses dekripsi, kesalahan satu bit pada *ciphertext* akan mempengaruhi blok *plaintext* saat itu serta 1 bit blok *plaintext* setelahnya.



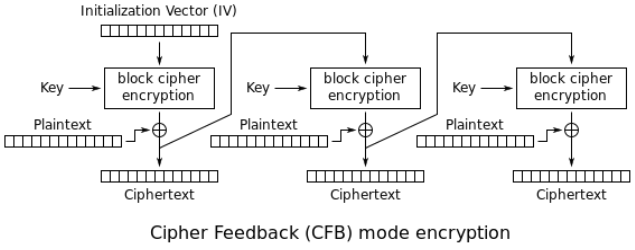
Gambar 3. Enkripsi pada mode CBC



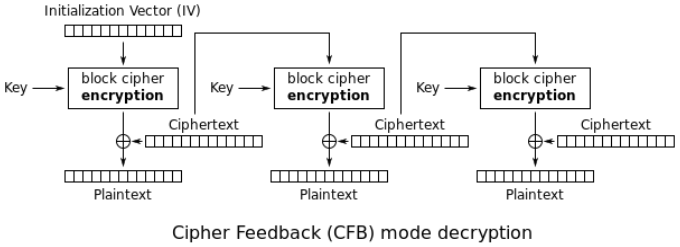
Gambar 4. Dekripsi pada mode CBC

3. *Cipher Feedback* (CFB)

   Mode ini dibuat untuk mengatasi komunikasi data dengan ukuran blok belum lengkap. Pesan dienkripsi dengan cara seperti *stream cipher*. Mode ini memerlukan sebuah antrian dengan ukuran sama dengan blok. Antrian akan digeser dengan *ciphertext* hasil enkripsi sebelumnya.



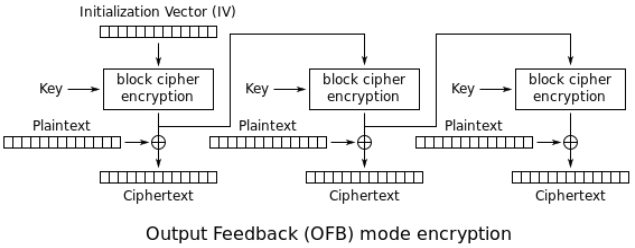
Gambar 5. Enkripsi pada mode CFB



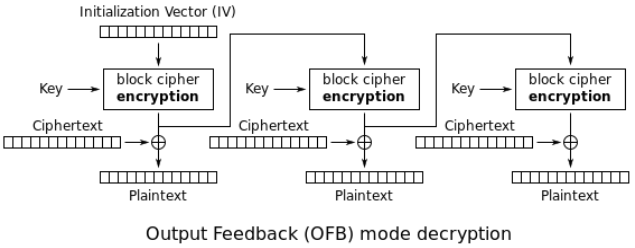
Gambar 6. Dekripsi pada mode CFB

4. *Output Feedback* (OFB)

   Mode ini hampir sama dengan mode CFB, perbedaannya terletak pada pergeseran antrian. Antrian digeser dengan *output* dari proses enkripsi bukan *ciphertext*.



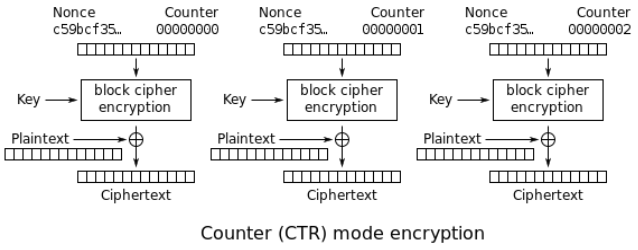
Gambar 7. Enkripsi pada mode OFB



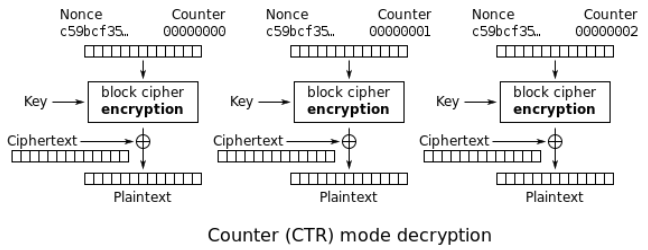
Gambar 8. Dekripsi pada mode OFB

5. *Counter* (CTR)

   Mode *counter* hampir mirip dengan mode ECB. Perbedaannya terletak pada *input* fungsi enkripsi. Pada mode ECB masukan fungsi enkripsi berupa *plaintext* sedangkan pada mode *counter* berupa nilai counter yang direpresentasikan sebagai sebuah blok. Setiap proses enkripsi nilai *counter* akan dinaikkan satu.



Gambar 9. Enkripsi pada mode *Counter*

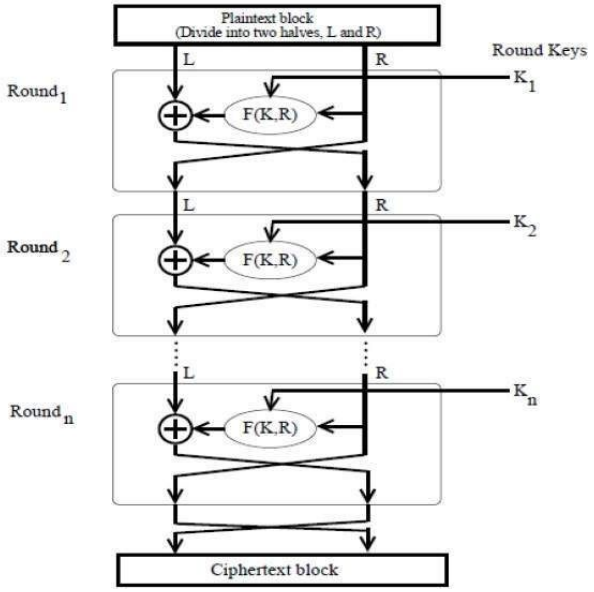


Gambar 10. Dekripsi pada mode *Counter*

B. Jaringan *Feistel*

Jaringan Feistel adalah struktur yang banyak digunakan dalam pembuatan sebuah *cipher*. Kelebihan dari jaringan ini adalah operasi enkripsi dan dekripsi yang sangat mirip, dan kadang dalam beberapa kasus hanya perlu membalikkan urutan kunci yang digunakan.

Dalam Jaringan *Feistel* terdapat fungsi internal yang dijalankan tiap ronde enkripsi dan dekripsi. Fungsi internal ini lah yang dibuat agar serumit mungkin.



Gambar 11. Bentuk struktur Jaringan *Feistel* (sumber: https://www.tutorialspoint.com/cryptography/feistel_block_cipher.htm)

C. Prinsip *Confusion* dan *Diffusion* Shannon

Confusion dan *Diffusion* adalah prinsip yang dikemukakan oleh Claude Shannon dalam makalahnya *Communication Theory of Secrecy Systems*. Prinsip ini dijadikan sebagai panduan dalam merancang algoritma kriptografi modern.

Prinsip *Confusion* adalah dimana pada algoritma kriptografi hubungan antara *plaintext, key,* dan *ciphertext* tidak terlihat jelas. Hal ini berarti analisis kriptografi disulitkan dengan tidak bisa digunakannya teknik seperti frekuensi analisis.

Prinsip *Diffusion* adalah dimana perubahan pada satu bit *plaintext* atau *key* memberikan dampak perubahan yang besar terhadap hasil *ciphertext*.

III. Rancangan Algoritma

A. Pembangkitan Kunci Putaran

Kunci putaran dibangkitkan sebanyak ronde yang digunakan. Kunci pada putaran ke-*i* didapatkan dengan rumus:

Kunci eksternal dibagi 2 menjadi L dan R

$K_0 = L$

$K_i = K_{i-1} \oplus ROL64(R, 7 * i)$

Dimana ROL64(x, n) adalah *circular left shift* x sebanyak n bit dan i adalah putaran saat ini. Cara ini memberikan maksimal 64 kunci berbeda sebelum kembali ke kunci awal.

B. Fungsi internal pada Jaringan *Feistel*

Sementara, dalam fungsi internal Jaringan *Feistel* terjadi proses seperti berikut:

1. XOR dengan K_i
2. Permutasi bit dengan mengelompokkan bit-bit pada posisi ganjil dan genap, sehingga setelah permutasi semua bit pada posisi genap akan berada pada setengah awal *ciphertext* dan bit pada posisi ganjil pada setengah akhir. Contohnya adalah jika urutan bit berupa 0 1 2 3 4 5 6 7, setelah permutasi bit akan menjadi 0 2 4 6 1 3 5 7.
3. Substitusi dengan S-box Rijndael (AES).
4. Permutasi dengan P-box. Pada tahap ini, *byte* pada blok akan disusun ulang sesuai dengan tabel transposisi yang telah didefinisikan, yaitu [0, 7, 2, 5, 3, 1, 4, 6].

IV. Percobaan dan Analisis

A. Percobaan

Percobaan dilakukan pada mode operasi ECB, CBC, EFB, OFB, dan Counter. Pada percobaan ini akan diambil waktu eksekusi pada proses enkripsi dan dekripsi. Pesan yang digunakan dalam percobaan ini memiliki ukuran 256 *bytes* dan 12.228 *bytes*.

Mode	Operasi	256 bytes	12.228 bytes
ECB	Enkripsi	0.00259852409 36279297	0.08591747283 935547
	Dekripsi	0.00209927558 8989258	0.08269095420 837402
CBC	Enkripsi	0.00188493728 63769531	0.08368802070 617676
	Dekripsi	0.00194764137 2680664	0.08465671539 30664

CFB	Enkripsi	0.00194263458 25195312	0.08247590065 002441
	Dekripsi	0.00197601318 359375	0.08471655845 64209
OFB	Enkripsi	0.00265073776 2451172	0.08316779136 657715
	Dekripsi	0.00210118293 76220703	0.08337235450 744629
Counter	Enkripsi	0.00201535224 9145508	0.08443355560 302734
	Dekripsi	0.00196862220 76416016	0.08305597305 297852

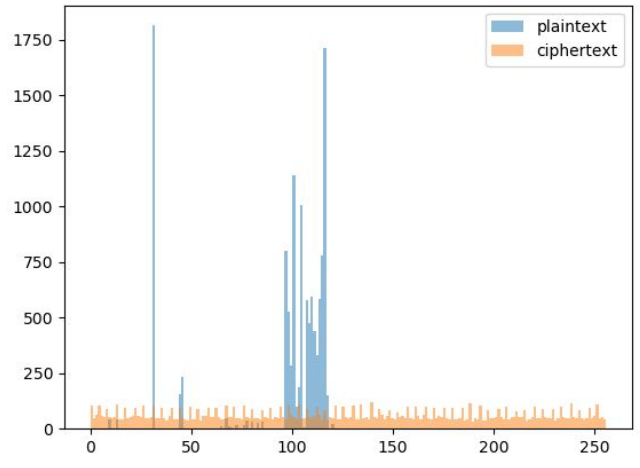
Tabel 1. Waktu eksekusi algoritma MR (dalam detik)

B. Analisis *Brute force*

Teknik *Brute force* dilakukan dengan cara mencoba seluruh kombinasi kunci untuk melakukan dekripsi *ciphertext*. Dengan teknik ini pasti didapatkan kunci yang benar. Namun, permasalahannya terletak pada waktu pencariannya. Pada algoritma MR digunakan kunci dengan ukuran 128 bit. Sehingga seluruh kombinasi kunci akan ada sebanyak 2^{128} kunci. Dengan menggunakan asumsi bahwa sebuah komputer dapat mencoba 10^8 kombinasi kunci per detik. Maka akan diperlukan waktu selama 3.4×10^{30} detik atau sekitar 1.07×10^{23} tahun untuk mencoba semua kunci. Hal ini tentunya merupakan waktu yang sangat lama, disamping itu masih diperlukan pula biaya untuk komputasi. Oleh karena itu, dapat disimpulkan algoritma MR aman dari serangan *brute force*.

C. Analisis *Confusion* dan *Diffusion*

Analisis *Confusion* dilakukan dengan cara membandingkan frekuensi kemunculan karakter pada *plaintext* dan *ciphertext* pada pesan dengan panjang 12.228 *bytes*. Berikut merupakan histogram kemunculan karakter.



Gambar 12. Histogram kemunculan karakter pada *plaintext* dan *ciphertext*

Berdasarkan analisis frekuensi pada histogram, didapatkan distribusi karakter yang merata pada *ciphertext*. Hal ini telah memenuhi tujuan dari prinsip *confusion*.

Pada analisis *Diffusion* dilakukan dengan cara mengubah 1 bit pada kunci yang digunakan untuk enkripsi. Kemudian dilakukan perbandingan pada *ciphertext* yang dihasilkan pada masing-masing enkripsi.

Plaintext
4c 6f 72 65 6d 20 69 70 73 75 6d 20 64 6f 6c 6f 72 20 73 69 74 20 61 6d 65 74 2c 20 63 6f 6e 73 65 63 74 65 74 75 72 20 61 64 69 70 69 73 63 69

Key 1
31 32 33 34 35 36 37 38 39 30 61 62 63 64 65 **66**
Key 2
31 32 33 34 35 36 37 38 39 30 61 62 63 64 65 **67**

Ciphertext (*Key 1*)
6d 30 e9 ab 35 f9 e7 79 f3 47 ee cf 3a 58 77 c7 a2 8a 16 f9 2d 00 6d 3d f4 cc 86 21 8f 39 78 d8 44 a0 ca 7a ee b6 54 37 d6 74 8b 45 41 e5 38 a5 08 24 b8 05 41 58 66 98 67 74 8e 6e a1 15 e4 d7
Ciphertext (*Key* 2)
a2 fb ec 1b 05 c9 52 02 df a0 fe 0e 36 c3 79 45 ec 84 9e 99 fb fb ff 54 11 e8 a3 04 25 f3 42 92 27 09 2e 2f 0f cb a1 38 0a 2a d0 ca 86 47 66 c4 0f 60 c4 81 25 75 81 e0 d8 9e 68 95 d6 d6 4e 9f

Dari hasil percobaan, didapatkan bahwa dengan mengubah 1 bit pada kunci memberikan hasil *ciphertext* yang berbeda serta tidak dapat diprediksi. Oleh karena itu, dapat disimpulkan algoritma MR cipher telah menerapkan prinsip *diffusion*.

V. KESIMPULAN

Algoritma MR cipher yang dirancang sudah cukup baik dalam melakukan enkripsi dan dekripsi pesan. Algoritma telah memenuhi prinsip *confusion* dan *diffusion*, serta ukuran *key* yang panjang yang menyebabkan serangan *bruteforce* sulit dilakukan. Jumlah putaran juga dapat ditambahkan hingga 64 putaran sebelum kunci digunakan berulang. Kecepatan enkripsi dan dekripsi yang cepat juga menyebabkan algoritma ini cocok digunakan pada perangkat dengan kemampuan komputasi rendah.

Saran yang dapat dilakukan selanjutnya adalah untuk melakukan analisis kriptografi yang lebih lanjut untuk mengetahui kelemahan-kelemahan dari algoritma ini.

REFERENSI

[1] Munir, Rinaldi. 2019. Slide Kuliah IF4020 Kriptografi: Kriptografi Modern.

[2] Munir, Rinaldi. 2018. Slide Kuliah IF4020 Kriptografi: Serangan terhadap kriptografi.

[3] C. E. Shannon. "Communication Theory of Secrecy Systems". Bell System Technical Journal, vol. 28-4, pp. 656-715, October 1949.

[4] Schneier, Bruce; Kelsey, John. "Unbalanced Feistel Network and block cipher design". Fast Software Encryption, Lecture Notes in Computer Science. 1039. pp. 121–144.