Algoritma Cipher Blok Mats

Algoritma Cipher Blok dengan Fungsi Acak Modulus

Ramandika Pranamulia 13512078
Teknik Informatika
Institut Teknologi Bandung
Bandung,Indonesia
ramandika@students.itb.ac.id

Mamat Rahmat 13512007
Teknik Informatika
Institut Teknologi Bandung
Bandung, Indonesia
mamat.rahmat@students.itb.ac.id

***Abstrak*—Kriptografi adalah suatu ilmu yang sangat penting dalam bidang informatika/*computer science*. Kriptografi digunakan untuk menyembunyikan data/pesan rahasia dengan cara mengubah pesan menggunakan metode kriptografi (enkripsi) menjadi pesan tidak bermakna/*cipher text* yang dapat dikembalikan ke pesan semula dengan sebuah kunci (dekripsi). Proses enkripsi teks pada komputer dapat dilakukan dalam blok *cipher atau* masing masing huruf. Enkripsi blok *cipher* lebih banyak dipakai karena kesulitannya untuk dipecahkan oleh *cryptanalyst*. Pada paper ini akan dibahas suatu algoritma blok *cipher* baru bernama Algoritma Block Cipher Mats dari segi konsep, implementasi, kompleksitas, keunggulan, dan kelemahannya.**

Keywords—kriptografi; enkripsi; ciphter text; kunci; dekripsi; cryptanalyst; block cipher

I. PENDAHULUAN

Kriptografi merupakan bidang ilmu informatika yang digunakan secara luas untuk keamanan sistem informasi seperti transaksi online (*e-payment*) dan autentikasi seperti *digital signature*. Pada kirptografi terdapat beberapa istilah penting yaitu enkripsi, dekripsi, *cipher, key, cipher text,* dan *plain* text. *Plain text* merupakan pesan yang akan diproses oleh algoritma kriptografi (*cipher)* agar menjadi pesan tidak bermakna(*cipher text*), proses tersebut disebut enkripsi sedangkan proses sebaliknya disebut dekripsi. *Key* sendiri adalah suatu teks rahasia yang digunakan untuk mengenkripsi dan mendekripsi pesan.

Kriptografi sudah ada sejak peradaban romawi dan terus berkembang sampai sekarang. Namun, orientasi algoritma kriptografi/*cipher* antara zaman romawi dengan zaman sekarang sangatlah berbeda. Pada zaman sekarang komputer sebagai salah satu mesin komputasi yang cepat dan dapat memproses data yang besar dilibatkan dalam proses kriptografi. Secara umum algoritma *cipher* modern dapat dibagi menjadi *stream based cipher* dan *block based cipher*. *Stream based cipher* menggunakan bit sebgai basis konversi *plain text* ke *cipher text* sedangkan *block based cipher* menggunakan blok sebagai basis konversi.

Kekuatan dari sistem kriptografi terletak pada *key* dan algoritma yang digunakan, karenanya dalam mendesain algoritma kriptografi harus diperhatikan mengenai aspek aspek penting seperti *confusion* dan *diffusion*. *Confusion* adalah prinsip dimana hubungan antara plainteks, cipherteks, dank unci harus disembunyikan seminimal mungkin. Sedangkan *diffusion* adalah suatu prinsip untuk menyebarkan pengaruh perubahan 1 bit ke seluruh hasil *cipher text*, dengan kata lain perubahan apda 1 bit plainteks dapat menyebabkan *cipher text* yang tidak terduga hasilnya dari hasil sebelum bit tersebut diubah.

A. Blok Cipher

Blok cipher seperti yang sudah dijelaskan sebelumnya adalah metode enkripsi dan dekripsi berdasarkan per blok bits. Pada blok *cipher* terdapat beberapa mode yaitu ECB (*Electronic Code Block)*, CBC (*Chaining Block Cipher)*, dan CFB-n bits. ECB mengenkripsi setiap blok *cipher* secara individual dan independen. CBC mengenkripsi blok *cipher* ke-i dengan memanfaatkan hasil blok *cipher* ke-(i-1), sedangkan untuk blok pertama yang tidak memiliki pendahulu digunakan blok *cipher* random yang disebut *initialization vector* (IV). Pada CBC jika jumlah bits data tidak habis dibagi jumlah blok bits maka harus ditambahkan bits padding pada bits data atau dapat digunakan juga metode CFB n-bits.

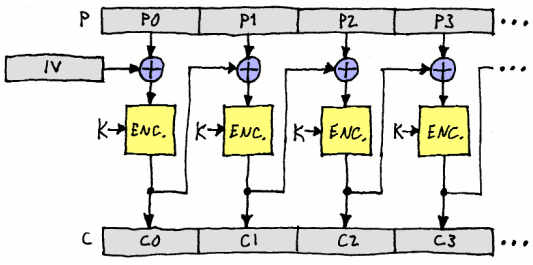
Misalkan kita memiliki plainteks yang memiliki format biner 10100010001110101001 dan key yang digunakan untuk enkripsi dan dekripsi 1011. Pertama kita harus membagi plainteks menjadi blok bits, misalkan 4 bit, maka blok bits yang terbentuk 1010, 0010, 0011, 1010, 1001. Kemudian misalkan operasi yang kita gunakan antara blok bits dengan key adalah melakukan bitwise XOR kemudian geser secara *wrapping* hasil XOR sebanyak 1 ke kiri. Pada mode ECB proses enkripsi dapat dilihat pada tabel di bawah.

Tabel 1 Proses ECB

Blok-i	Plain Teks	Key	XOR	Shift Left *(cihpertext)*
Blok-1	1010	1011	0001	0010
Blok-2	0010	1011	1001	0011
Blok-3	0011	1011	1000	0001
Blok-4	1010	1011	0001	0010
Blok-5	1001	1011	0010	0100

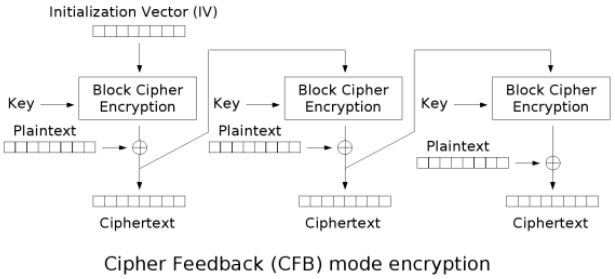
Pada mode CBC, *cipher blok text* sebelumnya akan digunakan dalam proses enkripsi blok sekarang. Hal ini dilakukan agar blok dengan nomor blok >i bergantung pada blok dengan nomor blok <=i, sehingga jika suatu bit saja

berubah pada suatu blok maka akan berpengaruh pada blok blok sesudahnya. Misalkan fungsi enkripsi yang digunakan sama dengan fungsi yang digunakan pada ECB di atas, namun pada CBC sebelum block plainteks di XOR dengan key terlebih dulu di-XOR dengan cipher sebelumnya.



Gambar 1 Proses Enkripsi dengan CBC [https://www.cs.rit.edu/~ark/lectures/https02/blockcipher2.png , diakses 23 Maret 2016]

Pada mode CFB-n bits pesan dienkripsi seperti pada *stream based cipher* sehingga pesan tidak perlu dipadding untuk dapat bias dienkrip menggunakan blok *key*. Konsep dari algoritma CFB-n bits mirip dengan algoritma CBC, plainteks dioperasikan dengan cipherteks sebelumnya sebelum dienkrip menggunakan *key*. Namun, pada CFB yang dienkripsi menggunakan key adalah IV atau *ciphertext* yang akan dioperasikan dengan *plaintext*, kemudian diambil sebanyak n bits untuk dioperasikan dengan *plaintext*. Ilustrasi yang lebih jelas dapat dilihat pada gambar dibawah ini.



Gambar 2 Proses Enkripsi degan CFB [http://i.stack.imgur.com/jaqUc.png, diakses 23 Maret 2016]

B. SHA256

SHA merupakan singkatan dari *Secure Hash Algorithm* yang mana merupakan salah satu algoritma *hashing.* Perbedaan *hashing* dengan enkripsi yaitu *hashing* tidak bisa dikembalikan/*irreversible*, artinya jika suatu pesan dihash maka hasil dari hash tidak dapat dikembalikan menjadi pesan awal. Pada umumnya para *hacker* menggunakan *rainbow table* untuk mengetahui plainteks dari suatu stirng *hashing. Rainbow table* sendiri adalah kumpulan kata pada kamus dengan hasil *hashingnya. Hacker akan mencoba menyocokkan string hashing* dengan hasil *hash* pada *rainbow table*, jika ada *hash* string yang cocok maka plainteks yang berkorespondensi dengan hasil *hashing* itu adalah plainteksnya.

SHA memiliki dua versi kelompok secara umum yaitu SHA-1 dan SHA-2. SHA-1 dan SHA-2 merupakan kelompok dari algoritma SHA. Pada SHA-2 algoritma yang ada yaitu SHA-224, SHA-256, SHA-512, dan lain lain. Nomor SHA menandakan ukuran *hashing*, sebagai contoh pada SHA-256 hasil *hashing* akan berupa string random sebesar 256 bit. Karena seluruh string dengan panjang berapapun akan menjadi string *hashing* dengan panjang 256 bit, maka akan ada kemungkinan hashing *hashing* saling tumpang tindih (*Hash(string_1)=Hash(string_2)* dimana *string_1≠string_2*). Namun kemungkinan akan sangat kecil yaitu sebesar 1/(2^256-1) untuk dua buah string berbeda pada SHA-256.

C. Jaringan Feistel

Jaringan feistel diusulkan oleh Horst Feistel dalam desainnya tentang algoritma Lucifer yang kemudian dipopulerkan oleh DES. Jaringan feistel merupakan metode umum untuk melakukan permutasi terhadap suatu fungsi enkripsi. Konsep jaringan feistel yaitu membagi dua blok pesan(subblok 1 dan subblok 2) dan kemudian meng-XOR kan salah satu blok dengan kunci, untuk ronde berikutnya dilakukan terhadap subblok pesan yang tidak di-XOR pada ronde sebelumnya.

D. Cryptanalysis

Cryptanalysis adalah proses untuk mendapatkan *plaintext* dari string *cipher text* tanpa mengetahui *key* dari *cipher text* tersebut. *Cryptanalysis* pada umumnya dikerjakan dengan mencari pola/hubungan antara *plaintext, ciphertext,* dan kunci. Secara umum metode *cryptanalysis* dapat dibagi menjadi tiga yaitu

1. *Cipher text only attack*, pada serangan ini *attacker* hanya memiliki *cipher text*. Serangan ini merupakan serangan yang paling sulit dilakukan karena tida adanya informasi mengenai *plaintext* maupun *key.* Serangan ini biasanya mengandalkan algoritma *brute force* atau statistik. Algoritma *brute force* dilakukan dengan cara mencoba semua kemungkinan *key* yang mungkin sedangkan statistikan *attack* dapat digunakan untuk membandingkan kemunculan huruf antara *plainteks* dengan *cipher text*.

2. *Known plain text*, pada serangan ini *attacker* memiliki pasangan *cipher text* dan *plain text* tetapi pasangan tersebut tidak mencakup seluruh isi pesan asli. Walaupun terlihat kesulitan masih tetap sama dengan *cipher text only attack,* situasi ini dapat memberi keuntungan sendiri bagi *cryptanalyst*.

3. *Chosen plain text attack*, pada serangan ini *attacker* dapat secara bebas memilih *plain text* dan melihat hasil *cipher text*-nya. Metode ini merupakan metode paling banyak digunakan dibanding dua metode terdahulu karena kemampuannya yang lebih mangkus untuk memecahkan *cipher text*.

II. MATS ALGORITHM

A. Konsep dan Desain

Algoritma Mats yang kami usulkan terdiri dari subbytes substitution menggunakan sbox rijndael, jaringan feistel, pemilihan subblok pada feistel berdasarkan key, dan operasi blok yang juga berdasarkan key. Kami mengharapkan adanya ketergantungan pada key dalam proses enkripsi dan dekripsi sehingga kami mencoba melibatkan key semaksimal mungkin.

Algoritma Mats ini dapat digunakan untuk melakukan enkripsi terhadap blok plain dengan ukuran berapapun dan blok key dengan ukuran berapapun juga. Pada contoh kali ini kami akan menggunakan ukuran blok sebesar 128 bit dan ukuran key sebesar 128 bit juga untuk proses enkripsi dan dekripsinya.

Keunikan yang dimiliki algoritma mats ini yaitu pemilihan subblok feistel yang akan dienrkip dengan key dipilih berdasarkan suatu bilangan acak bergantung pada key. Sehingga untuk setiap ronde bagian subblok yang dienkripsi bisa berbeda, akan dijelaskan selanjutnya. Kemudian ketika subblok sudah didapat, bytes input yang akan dioperasikan dengan key juga tidak bersesuaian seperti halnya pada algoritm rijndael.

B. Proses Enkripsi

Proses enkripsi terdiri dari tiga bagian yaitu *subbytes substitution*, *maximum modulo,* dan *prefix sum modulation round key*. Proses ini berlangsung sebanyak enam belas putaran. Sebelum blok pesan dimasukkan ke dalam fungsi harus dilihat terlebih dahulu mode apa yang digunakan. Terdapat tiga mode yang dapat dipilih yaitu ECB, CBC, dan CFB n-bits. Berikut merupakan tahapan fungsi enkripsi.

1. Pembangkitan kunci internal→kunci internal dibangkitkan dengan cara melakukan hashing terhadap key yang dimasukkan oleh pengguna. Hasil sha-256 adalah 256 bit sedangkan kita membutuhkan 128 bit maka hasil sha-256 kita bagi menjadi dua dan di-XOR berdasarkan significant bitsnya (MSB dengan MSB dan LSB dengan LSB)
2. *Subbytes substitution*, proses sama dengan subbytes substitution pada algoritma rijndael, menggunakan sbox rijndael juga.
3. *Maximum modulo*. Blok pesan berukuran 4x4 byte kita bagi menjadi empat sehingga berukuran 2x2 byte lalu kita beri nomor, kemudian untuk memilih 2 blok yang akan dienkripsi dengan *round key* maka cari modulasi penjumlahan hexa pada setiap elemen *round key*. Dua hasil modulo terbanyak yang muncul akan dijadikan sebagai nomor indeks blok yang dipilih.
4. *Prefix sum modulation round key*. Pada poin dua sudah dipilih dua buah blok berukuran 2x2 byte yang siap di-XOR dengan key yang berukuran 4x4 byte. Operasi XOR antara elemen blok input tidak berdasarkan index seperti pada algoritma rijndael tetapi berdasarkan hasil modulasi penjumlahan hexa setiap elemen key dalam mod8.

C. Proses Dekripsi

Proses dekripsi dapat dilakukan jika pengguna memiliki *cipher text* dan *key*. Proses dekripsi dilakukan dengan mereverse urutan dari fungsi enkripsi sehingga yang harus dijalankan pertama kali yaiu *prefix sum modulation key*, *maximum modulo, subbytes substitution.* Semua tahapan tersebut membutuhkan *round key* sehingga *round key* degenerate di awal round dekripsi sebelum melakukan tahpan di atas.

Pada *prefix sum modulation key* kita harus melakukan *prefix sum modulation key* untuk semua elemen byte kunci. Setelah itu kita XOR elemen key dengan index hasil modulasi tersebut dengan index byte pada dua subblok teratas. Pada maximum modulo posisi subblok dua teratas ditransformasi berdasarkan modulo dari elemen *round key* yang paling sering muncul. Pada subbytes substitution substitusi dilakukan tetapi dengan menggunakan tabel sbox yang sudah diinverse.

III. EKSPERIMEN DAN PEMBAHASAN HASIL

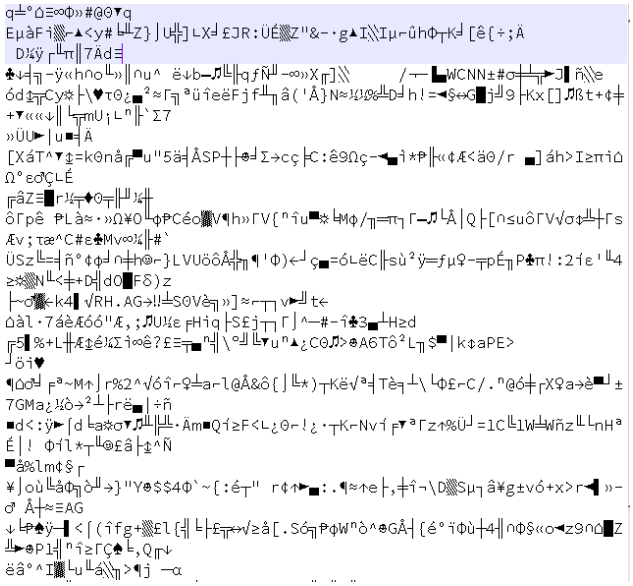
Eksperimen dilakukan terhadap sampel berupa pesan teks dengan ukuran 1331 byte. Potongan sampel pesan yang dipakai dapat dilihat pada tabel di bawah

Tabel 2 *Plaintext*

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

Untuk tiga hasil enkripsi di bawah, kami menggunakan key “mamatrahmatramandikapranamulia”. Berikut adalah hasil enkripsi dari sampel pesan di Tabel 1 dengan mode ECB.

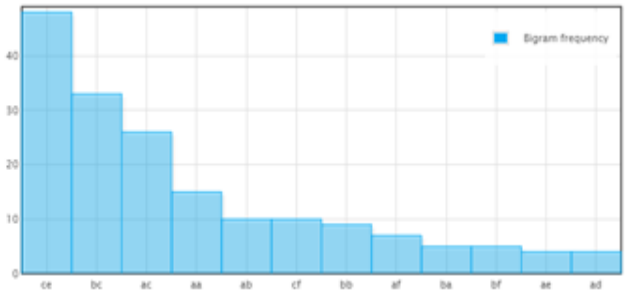
Gambar 3 Mode ECB



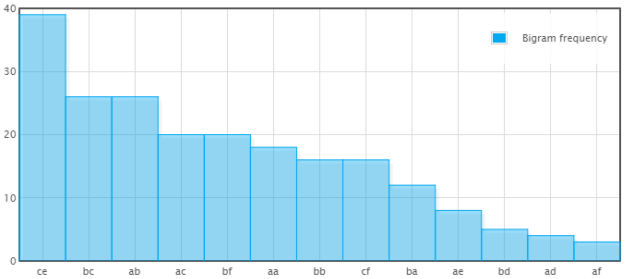
Gambar 4 Mode CBC

Gambar 5 Mode CFB

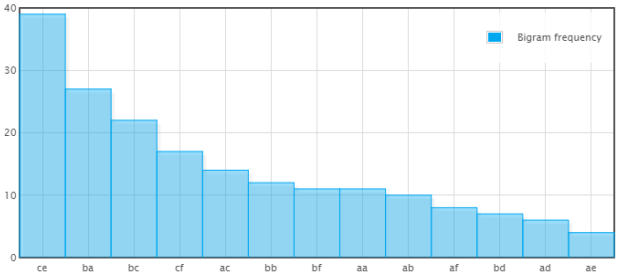
Sedangkan hasil statistiknya dapat dilihat pada gambar dibawah ini



Gambar 6 ECB Statistik

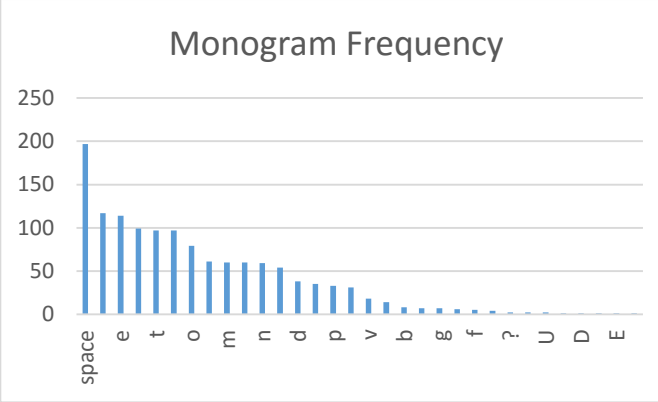


Gambar 7 CBC Statistik



Gambar 8 CFB Statistik

Jika hasil statistik tersebut dibandingkan dengan statistik huruf pada plain teks seperti dibawah ini(Monogram Frequency) maka walaupun analisis statistik bisa dilakukan namun jenis cipher karakter dan plain karakternya tidaklah bersesuaian, sehingga akan sulit ditebak juga.



Berdasarkan hasil yang diperoleh, algoritma yang kami buat berhasil melakukan enkripsi pesan asli menjadi tidak dikenali, sehingga algoritma yang dirancang mampu menenkripsi pesan. Selain itu, proses dekripsi dapat mengembalikan pesan semula baik dengan mode ECB, CBC dan CBF.

Kelemehan algoritma ini yaitu running time yang lama untuk pesan yang relatif kecil. Hal ini disebabkan akibat penggunaan SHA-256 untuk seitap round key. Untuk ukuran teks yang kecil serangan statistik masih mungkin dilakukan karena grafik menunjukkan adanya perbedaan yang cukup signifikan antara hexa yang dihasilkan. Kelompok dua hexa pada diagram kemunculan di atas merepresentasikan satu karakter berukuran satu byte.

IV. KESIMPULAN DAN SARAN

Berdasarkan hasil eksperimen dan analisis yang telah dilakukan, dapat disimpulkan bahwa algoritma yang diajukan berhasil mengenkripsi dan menghilangkan hubungan frekuensi huruf dari plainteks. Rancangan algoritma yang sudah ada dapat dikembangkan lebih lanjut, salah satunya dengan mengganti SHA256 dengan md5 untuk mempercepat prosesn enkripsi dan menggabungkan algoritma rijndael dengan algoritma ini.

V. REFERENSI

- Rouse, Margaret. Block Cipher. http://www.searchsecurity.techtarget.com/definition/block-cipher, diakses tanggal 23 Maret 2016
- Joan Daemen and Vincent Rijmen, “The Block Cipher Rijndael”
- Munir, Rinaldi. Bahan KuliahIF3058 Kriptografi: Algoritma Kriptografi Modern (Bagian 2)