Algoritma Block Cipher Mini-Box

Zulva Fachrina
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jalan Ganesha 10 Bandung 40132, Indonesia
13513010@std.stei.itb.ac.id

Muhtar Hartopo
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jalan Ganesha 10 Bandung 40132, Indonesia
muhtarhartopo@gmail.com

Abstrak — Block Cipher merupakan algoritma beroperasi pada sekumpulan bit dengan panjang sama yang disebut *block*. Algoritma enkripsi yang kami kembangkan merupakan algoritma *block cipher* yang menawarkan solusi dengan keamanan yang berlapis namun tetap sederhana. Mini-Box menerapkan jaringan feistel pada proses enkripsi dan dekripsinya. Algortima ini memiliki S-Box berukuran kecil yang dibangkitkan dinamik secara *pseudorandom* berdasarkan *seed* tertentu. Algoritma ini juga memiliki jumlah perputaran (*round*) yang berbeda bergantung pada kunci eksternal. Eksperimen menunjukkan bahwa algoritma Mini-Box ini memiliki tingkat *cofussion-difussion* yang tinggi.

Kata kunci — *block cipher; S-Box;* jaringan *feistel; enkripsi; Mini-Box*

I. PENDAHULUAN

Kemanan dalam berkomunikasi merupakan salah satu aspek yang sangat penting, terutama jika pesan yang dikirimkan merupakan pesan rahasia yang tidak boleh diketahui oleh pihak ketiga. Sejak zaman Julius Caesar, sudah dikenal berbagai cara untuk mengamankan kerahasiaan pesan melalui modifikasi terhadap data yang dikirimkan. Modifikasi ini bertujuan agar pesan tersebut hanya dimengerti oleh penerima pesan tanpa adanya gangguan dari pihak ketiga. Keahlian atau ilmu untuk menyandikan dan mengenkripsi informasi agar informasi tersebut hanya bisa dipahami oleh pihak yang besangkutan dikenal dengan istilah Kriptografi.

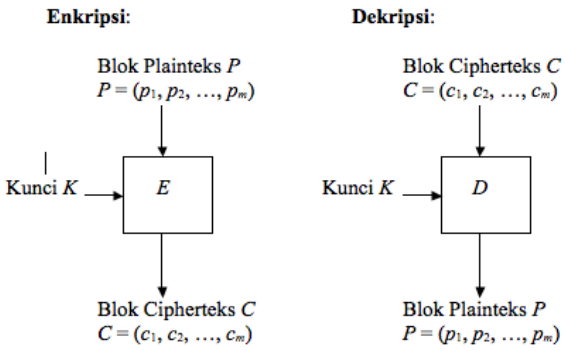
Kriptografi sudah ada sejak abad 19 sebelum masehi, dalam bentuk teks sandi yang diukir pada batu di Mesir. Ilmu kriptografi semakin berkembang dan kompleks sejak dikembangkannya mesin *rotor cipher* pada Perang Dunia I dan penemuan komputer pada Perang Dunia II. Kriptografi pada era komputer tersebut dikenal dengan istilah Kriptografi Modern. Kriptografi modern memanfaatkan teori matematis dan aplikasi komputer, dengan pengoperasian dalam mode bit atau biner. Kriptografi modern saat ini sudah mencakup berbagai teknik seperti *integrity checking*, autentikasi pengirim dan penerima, *digital signature*, bukti interaktif, dan komputasi keamanan, serta sudah diaplikasikan dalam sistem ATM, *password* komputer dan *E-commerce*.

Salah satu cara untuk melakukan enkripsi terhadap bit-bit pada pesan adalah dengan menggunakan metode Block Cipher. Block Cipher membagi bit-bit plainteks menjadi blok-blok bit dengan panjang yang sama. Block Cipher berkontribusi dalam meningkatkan keamanan pesan dengan mengombinasikan operasi sederhana seperti permutasi dan substitusi yang dilakukan berulang-ulang dalam beberapa putaran. Block Cipher telah banyak dikembangkan untuk menghasilkan algoritma-algoritma baru yang terkenal, seperti Algoritma Lucifer/DES, IDEA, RC5 dan Rijndael. Pada makalah ini, dibahas salah satu algoritma baru yang juga memanfaatkan Block Cipher yang diberi nama Mini-Box Block Cipher. Dengan memanfaatkan Jaringan Feistel, S-Box untuk proses substitusi, dan transposisi matriks, diharapkan algoritma ini dapat berkontribusi dalam ilmu kriptografi sebagai salah satu algoritma yang memiliki properti *confusion* dan *diffusion*.

II. DASAR TEORI

A. Block Cipher

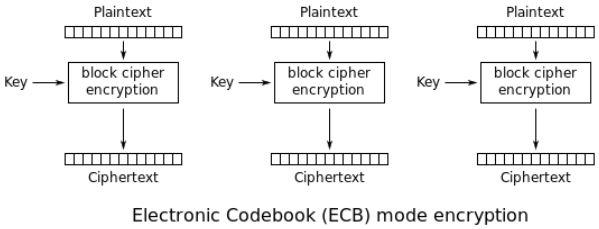
Block Cipher merupakan algoritma beroperasi pada sekumpulan bit dengan panjang sama yang disebut *block*. Plainteks dibagi dalam blok-blok, kemudian dienkripsi dengan kunci yang panjang sama dengan panjang blok, sehingga menghasilkan cipherteks dengan panjang yang sama dengan plainteks. Pada Block Cipher, dikenal 4 mode operasi, yaitu Electronic Code Book (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), dan Output Feedback (OFB).



Gambar 1 Skema Enkripsi dan Dekripsi Block Cipher

1. Electronic Code Book (ECB)

   Electronic Code Book (ECB) merupakan teknik enkripsi sederhana dimana masing-masing blok dienkrispsi secara terpisah. Mekanisme ECB digambarkan dalalam skema di bawah ini

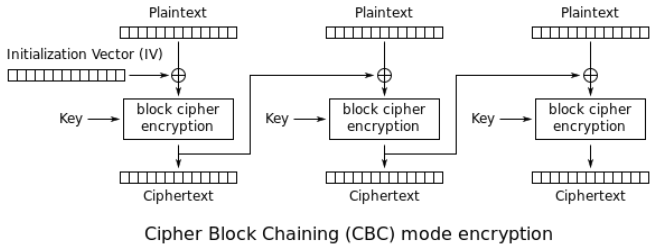


Gambar 2 Skema Enkripsi pada ECB

Kekurangan dari algoritma ECB adalah plainteks yang sama akan dienkripsi menjadi cipherteks yang sama, sehingga tidak menyembunyikan *pattern* data dengan baik.

2. Cipher Block Chaining (CBC)

   Cipher Block Chaining (CBC) merupakan metode dimana masing-masing blok pada plainteks di-XOR dengan hasil ciperteks pada blok sebelumnya sebelum dienkripsi. Dengan begitu, masing-masing block cipherteks bergantung pada seluruh blok plainteks sebelumnya. Untuk membuat masin-masing pesan menghasilkan cipherteks yang unik, digunakan Initialization Vector (IV) pada blok pertama.

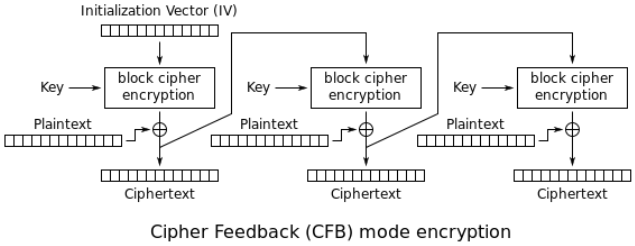


Gambar 3 Skema Enkripsi pada CBC

Mode CBC cukup banyak digunakan karena blok plainteks yang sama tidak akan menghasilkan cipherteks yang sama sehingga menyulitkan proses kriptanalisis. Namun, metode ini masih memiliki kelemahan yaitu kesalahan satu bit pada sebuah blok plainteks akan merambat pada blok cipherteks yang berkoresponden dan semua blok cipherteks selanjutnya.

3. Cipher Feedback (CFB)

   Metode yang digunakan pada Cipher Feedback (CFB) hampir sama dengan CBC, namun CFB mengenkripsi block cipher sama seperti pada cipher aliran (*stream cipher*). CFB dapat digunakan dengan mengombinasikan dengan *shift register* sebagai input pada *block cipher*.



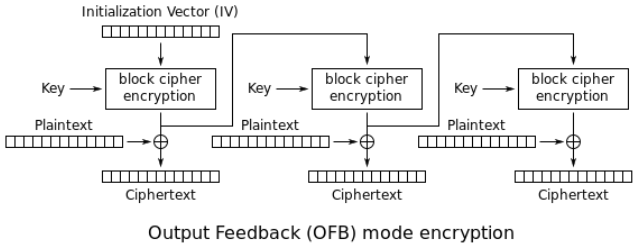
Gambar 4 Skema Enkripsi CFB

Seperti halnya pada CBC, kelemahan CFB terleteak pada pemrosesan yang sekuensial, sehingga kesalahan pada 1 bit plainteks akan merambat pada blok-blok ciperteks selanjutnya.

4. Output Feedback (OFB)

   Metode yang digunakan pada Output Feedback (OFB) hampir sama dengan yang digunakan pada CFB, namun n-bit hasi enkripsi terhadap antiran disalin menjadi elemen paling kanan pada antrian.

Gambar 5 Enkripsi pada OFB



Metode OFB menanggulangi kekurangan pada CBC dan CFB, karena kesalahan 1 bit pada plainteks hanya mempengaruhi cipherteks yang berkoresponden saja.

B. Properti Confusion dan Diffusion

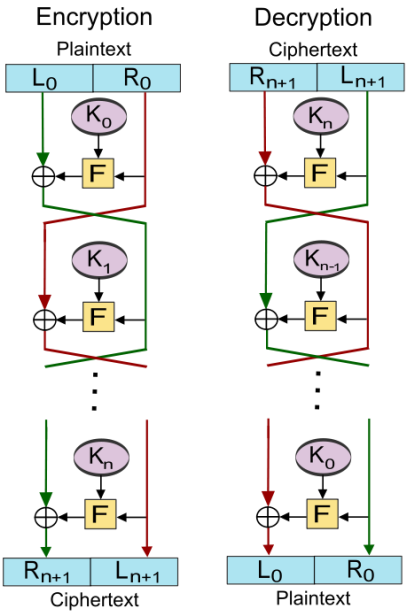
Confusion dan *diffusion* merupakan dua properti dalam kriptografi yang menjamin keamanan cipherteks dengan membuat serangan secara statistik menjadi lebih rumit. Prinsip ini diperkenalkan oleh Claude Shannon pada tahun 1949 dalam makalahnya yang berjudul *Communication theory of secrecy systems.* Prinsip *confusion* bekerja dengan menyembunyikan hubungan apapun yang ada antara plainteks, cipherteks, dan kunci. Prinsip *confusion* dapat direalisasikan dengan menggunakan algoritma substitusi yang kompleks. Sedangkan prinsip *diffusion* menyebarkan pengaruh 1 bit plainteks atau kunci ke sebanyak mungkin cipherteks. Misalnya jika perubahan dilakukan pada satu bit plainteks, maka secara statistik sebagian besar bit pada cipherteks juga akan berubah, begitu pula sebaliknya. Kedua prinsip ini merupakan panduan dalam merancang berbagai algoritma

kriptografi, dan juga menjadi konsep yang penting dalam merancang fungsi *hash* dan *pseudorandom generator*.

C. *Jaringan Feistel (Feistel Network)*

Jaringan Feistel merupakan struktur simetris yang digunakan dalam konstruksi Block Cipher. Jaringan ini ditemukan oleh kriptografer Horst Feistel selama pelaksanaan penelitiannya di IBM. Jaringan Feistel bersifat *reversible*, karena operasi untuk melakukan proses enkripsi dan dekripsi sama, sehingga tidak perlu membuat algoritma baru untuk mendekripsi cipherteks menjadi plainteks. Jaringan Feistel terbentuk dari sejumlah putaran yang terdiri dari operasi-operasi berulang, seperti permutasi, substitusi, dan operasi aljabar menggunakan XOR. Fungsi yang digunakan pada setiap putaran ini disebut *round function*.

Cara kerja Jaringan Feistel adalah sebagai berikut



Gambar 6 Skema pada Jaringan Feistel

Pertama-tama, plainteks dibagi menjadi dua bagian yaitu, L_0 dan R_0. Kemudian untuk masing-masing putaran i=1,2,3, … , kalkulasikan L_{i+1} dan R_{i+1} menggunakan rumus dibawah ini.

$$L_{i+1} = R_i$$
$$R_{i+1} = L_i \oplus \mathrm{F}(R_i, K_i).$$

Hasil cipherteksnya adalah (R_{n+1},L_{n+1}).

Proses dekripsi menggunakan cara yang sama, untuk i=n,n-1,…, hitung L_{i+1} dan R_{i+1} menggunakan rumus dibawah ini.

$$R_i = L_{i+1}$$
$$L_i = R_{i+1} \oplus \mathrm{F}(L_{i+1}, K_i)\,.$$

Perhitungan secara berulang akan menghasilkan plainteks (L_i,R_i).

Jaringan Feistel banyak digunakan pada algoritma kriptografi modern seperti DES, Lucifer, dan GOST karena menerapkan prinsip *confusion* dan *diffusion.*

D. *S-Box*

S Box merupakan matriks berisi substitusi sederhana yang memetakan satu atau lebih bit dengan satu atau lebih bit yang lain. Secara umum, S-Box menerima masukan berupa m bit dan mentransformasikannya menjadi n bit keluran. Sebuah S-Box m x n dapat diimplementasikan sebagai *lookup table* dengan 2^m kata pada masing-masing n bit. Masukan dari *lookup table* dijadikan index pada S-Box sedangkan keluarannya adalah elemen pada S-Box. Pada umunya digunakan tabel dengan elemen-elemen yang tetap, seperti pada algoritma DES, namun terdapat juga tabel yang dibangkitkan dari kunci, seperti Blowfish dan Twofish. Metode S-Box menjamin properti *confusion* pada algoritma.

III. RANCANGAN ALGORITMA

Algoritma yang diajukan diberi yaitu Mini-Box Block Cipher. Algoritma ini merupakan modifikasi dari algoritma Block Cipher dengan memanfaatkan Jaringan Feistel. Pertama-tama, algoritma menerima masukan penjang kunci sepanjang 128 bit atau 16 karakter. Algoritma juga akan menerima masukan plainteks yang kemudian akan dibagi ke dalam blok-blok dengan panjang yang sama seperti panjang kunci yaitu 128 bit. Secara garis besar, algoritma ini terdiri dari 3 tahap, yaitu penentuan jumlah *round*, pembangkitan kunci internal, dan *round function* pada Jaringan Feistel. *Round function* yang digunakan memanfaatkan *matrix modification* dan S-Box dengan *seed* yang dapat dimasukkan oleh pengguna.

A. *Penentuan Jumlah Round*

Pada Mini-Box jumlah putaran pada Jaringan Feistel ditentukan oleh jumlah nilai dari masing-masing karakter pada kunci. Nilai dari karakter sama dengan nilai urutan dalam *alphabet*, misalnya A=0, B=1, dst. Masing-masing karakter pada kata kunci dijumlahkan, kemudian hasil penjumlahan tersebut dimodulus dengan 16 untuk menentukan jumlah putaran pada Jaringan Feitsel. Tahapan untuk menentukan jumlah *round* atau putaran adalah sebagai berikut.

1. Ubah kata kunci sehingga menjadi *uppercase*, kemudian hitung nilai dari masing-masing karakter tersebut sesuai dengan urutannya pada *alphabet.*
2. Jumlahkan nilai dari masing-masing karakter pada kata kunci.
3. Modulus hasil penjumlahan dengan 16. Jika hasil modulus lebih kecil dari 4, maka hasilnya adalah 4. Hasil modulasi tersebut akan ditambahkan 4 sehingga jumlah minimal putaran adalah 8 dan maksimal adalah 19.

Berikut adalah contoh perhitungan jumlah *round* dengan menggunakan kunci = “KRIPTO”

Kunci: KRIPTO Jumlah nilai karakter: “K” + “R” + “I” + “P” + “T” + “O” = 83 Jumlah putaran: 83 mod 16 = 3, 3 < 4, maka jumlah putaran = 4 + 4 = 8

Tabel 1 Contoh Penentuan Jumlah Round dengan Kunci = KRIPTO

B. Pembangkitan Kunci Internal

Kunci internal dibangkitkan sebanyak *round* yang didapatkan dari perhitungan sebelumnya. Kunci pertama yang digunakan adalah kunci yang diberikan oleh pengguna, kemudian kunci untuk putaran selanjutnya ditentukan dengan melakukan pergeseran (*shift)* masing-masing bit satu satuan ke kanan, kemudian bit yang tersisa ditabahkan ke sebelah kiri. Pergeseran ini mengakibatkan *least significant bit* (LSB) dipindah posisi sehingga menjadi *most significant bit* (MSB).

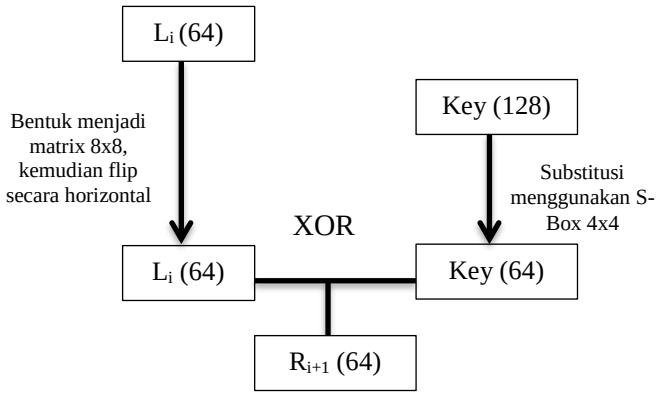
Berikut adalah contoh pembangkitan kunci internal dengan kunci awal = 10110 dan putaran = 8.

K_0 = 10110 K_1 = 01011 K_2 = 10101 K_3 = 11010

Tabel 2 Contoh Proses Pembangkitan Kunci Internal

C. Round Function

Setelah menentukan jumlah putaran (*round*) dan sejumlah kunci internal, plainteks dan kunci dapat diproses dalam Jaringan Feistel. Sebelumnya masing-masing plainteks yang sudah dibentuk dalam blok-blok 128 bit dipecah menjadi dua bagian, yaitu *Left* dan *Right*, yang masing-masing memiliki panjang 64 bit. Cipherteks untuk masing-masing putaran didapatkan dari pemrosesan dengan menggunakan *round function*. *Round function* yang digunakan pada Mini-Box terdiri dari 2 proses utama, yaitu modifikasi matriks dan substitusi dengan menggunakan S-Box. S-Box digunakan untuk mengubah ukuran kunci yang awalnya 128 bit menjadi 64 bit. Berikut adalah gambaran mekanisme *round function* yang digunakan pada Mini-Box.



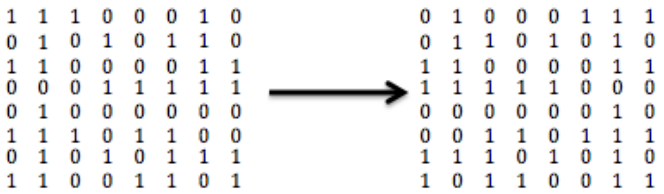
Gambar 7 Mekanisme Round Function

1. Modifikasi Matriks

   Modifikasi matriks dilakuakan melalui flip secara horizontal terhadap blok-blok plainteks baik yang di sebelah kanan (*right*) maupun kiri (*left*). Tahapan untuk melakukan modifikasi adalah sebagai berikut.

   - Untuk masing-masing bagian blok plainteks yang berukuran 64 bit, ubak menjadi matriks berukuran 8 x 8 bit dengan pengisian elemen matriks berurutan dari kiri ke kanan.
   - Hasil matriks di-flip secara horizontal sehingga elemen pada kolom pertama ditransposisi dengan elemen pada kolom kedelapan, elemen pada kolom kedua ditransposisi dengan elemen pada kolom ketujuh, dan seterusnya.

   Berikut adalah gambaran hasil dari modifikasi matriks dengan melakukan flip horizontal.



Gambar 8 Contoh Modifikasi Flip Horizontal Matriks

2. Substitusi S-Box

   S-Box digunakan untuk melakukan substitusi dengan memetakan bit-bit yang terdapat pada blok pada kunci. Kunci yang awalnya memiliki panjang 128 bit ditransformasikan menjadi 64 bit dengan menggunakan S-Box. S-Box dibangkitkan secara pseudorandom dengan menggunakan *seed* yang dimasukkan oleh pengguna. Berikut adalah langkah-langkah untuk melakukan substitusi dengan menggunakna S-Box.

   - Pengguna memasukkan bilangan sembarang yang digunakan untuk membangkitkan S-Box berukuran 4 x 4. Elemen pada S-Box merupakan angka pseudorandom antara 0 – 3 yang diinisiasi berdasarkan seed yang telah ditentukan.
   - Kunci yang memiliki panjang 128 bit dipisah menjadi blok-blok berukuran 4 bit, sehingga terdapat total 32 blok kunci dengan panjang masing-masing blok adalah 4 bit.
   - Masing-masing blok kunci yang berukuran 4 bit disubstitusi dengan menggunakan S-Box yang telah dibangkitkan sebelumnya. Elemen pertama dan terakhir pada blok kunci digunakan untuk menentukan baris pada *lookup table*, sedangkan sisanya digunakan untuk menentukan kolom.

- Elemen yang berada pada baris dan kolom yang ditentukan berdasarkan blok kunci merupakan keluaran untuk mensubstitusi blok kunci sebelumya. Hasil substitusi dari 4 bit blok kunci menghasilkan 2 bit blok kunci.
- Hasil substitusi 2 bit blok kunci dari jumlah 32 blok disatukan sehingga menghasilkan kunci baru dengan panjang 64 bit.

Untuk lebih memahami proses substitusi yang dilakukan, diberikan contoh sebagai berikut. Misalkan terdapat sebuah blok kunci dengan panjang 4 bit = 0111 dan sebuah S-Box seperti tabel di bawah ini.

3	0	2	1
1	3	0	2
2	0	1	3
0	2	3	1

Tabel 3 Contoh S-Box 4 x 4

Bit pertama dan terakhir dari blok kunci adalah **0**11**1** -> **01 = 1**, sedangkan sisanya adalah 0**11**1 -> **11 = 3**, maka nilai yang diambil adalah angka pada *lookup table* dengan baris ke-1 dan kolom ke-3 yaitu **1 = 01.**

Setelah melakukan proses modifikasi matriks pada blok plainteks dan substitusi S-Box pada blok kunci, maka didapatkan hasil berupa 64 bit plainteks dan 64 bit kunci yang telah dimodifikasi. Kedua hasil tersebut kemudian di-XOR untuk mendapatkan cipherteks yang akan digunakan pada proses selanjutnya.

IV. EKSPERIMEN DAN PEMBAHASAN HASIL

Eksperimen dilakukan dengan menggunakan menerapkan Mini-Box dengan menggunakan 3 teknik Block Cipher, yaitu Electronic Code Book (ECB), Cipher Block Chaining (CBC), dan Cipher Feedback (CFB). Data yang digunakan merupakan data berupa paragraf yang disimpan dalam file eksternal *mars.txt*. Kunci yang digunakan adalah kunci dengan panjang 16 karakter, yaitu **INFORMATIKASTEII**, sedangkan nilai yang digunakan sebagai *seed* untuk S-Box adalah **5286**.

Mars is the fourth planet from the Sun and the second-smallest planet in the Solar System, after Mercury. Named after the Roman god of war, it is often referred to as the "Red Planet"[13][14] because the iron oxide prevalent on its surface gives it a reddish appearance.[15] Mars is a terrestrial planet with a thin atmosphere, having surface features reminiscent both of the impact craters of the Moon and the valleys, deserts, and polar ice caps of Earth

Tabel 4 Data Uji Mars.txt

A. Hasil Pengujian dengan ECB

Metode EBC diterapkan dengan menjalankan algoritma yang dibuat secara langsung. Dengan menggunakan metode ECB, didapatkan hasil cipherteks dalam hex sebagai berikut. Representasi hex digunakan untuk mempermudah pembacaan, karena byte yang tidak bisa direpresentasikan dalam bentuk karakter.

19 96 E4 B9 97 6 48 88 8F FE B3 91 69 F3 AF F2 20 8D 76 F0 4B 4E FF AD 8F B6 B0 C3 60 F1 FA F4 64 A2 14 B3 B2 51 B1 49 95 F2 F6 C5 67 F9 FA F3 99 B4 69 6 7B 9A E2 85 9A FA BA D4 7C E8 FA F0 12 AE 48 D A1 73 A BE DB E2 BE D4 2F CF B5 EC AD A7 14 1B CE C4 47 CD 96 BA F6 D0 69 E8 BF F2 2E A8 27 3A 4 4A C1 9B D5 B6 98 D0 62 F9 BE A0 BD 83 E 5D 15 3F 97 8A 9E B6 84 DE 62 FD B4 A0 9B C0 60 EA 28 D1 B1 7 9A E4 FA 91 66 E8 FA E9 47 67 39 E EB B2 5D E0 89 F3 B0 D4 7D EE BF E4 4C 33 5B AA FA 4C 93 96 93 F3 F6 93 5D F9 BE A0 A6 41 EB 7A 38 C1 D B9 CA A5 8B EA 3E A8 87 A0 36 92 F5 AB 32 A4 6 B0 8F FE B3 91 66 EE B5 EE 14 28 26 21 83 5A C3 B8 89 F3 A0 D0 63 F9 B4 F4 94 20 10 88 3E 2B 40 C2 88 E3 A4 D7 6E FF BF A0 19 E 42 E5 4 9F DA 4 DB F7 F6 C3 6A F8 BE E9 47 F 66 69 7 9D 2A B 89 F7 B8 D2 6A B2 81 B1 D9 72 14 5 B4 C5 40 88 92 E5 F6 D0 2F E8 BF F2 C 8A 75 3C D5 56 DA 8E DB E6 BA D0 61 F9 AE A0 1B 1E 32 A2 77 D6 3 64 93 FF B8 91 6E E8 B7 EF 1F 17 36 3D D5 1A BD 8 93 F7 A0 D8 61 FB FA F3 A9 15 78 1 FC 5A 7B 84 9E F7 A2 C4 7D F9 A9 A0 AE F2 73 A3 F9 6 C8 93 9E F8 A2 91 6D F3 AE E8 CC 38 58 68 63 DF F4 0 92 FB A6 D0 6C E8 FA E3 6E 84 6A BD 5 EE 7 7F 9D B6 A2 D9 6A BC 97 EF 3 29 14 C1 39 CB 83 34 93 F3 F6 C7 6E F0 B6 E5 4D BC D2 DA B1 7A 90 A5 89 E2 A5 9D 2F FD B4 E4 9C 37 5B 64 36 E5 7B 8B 98 F3 F6 D2 6E EC A9 A0

Tabel 5 Hasil Enkripsi dengan ECB

B. Hasil Pengujian dengan CBC

Pada metode CBC, cipherteks yang didapatkan dari hasil perhitungan pada satu block akan digunakan sebagai salah satu masukan pada perhitungan block selanjutnya. Sebelum diproses, blok plainteks akan di-XOR terlebih dahulu dengan cipherteks dari perhitungan sebelumnya. Untuk blok plainteks di awal perhitungan, digunakan *initialization vector* yang dibangkitkan secara acak. Berikut adalah cipherteks hasil Mini-Box dengan menggunakan metode CBC.

FB D0 1B 4C C8 1B A7 D0 E6 8A DA F0 5 85 CE 9E 3D 8D B7 32 A2 75 AA 85 E9 BC EA B3 E5 F4 B4 EA 4F 93 75 CD 36 8A B7 1A FC CE 9C F6 2 8D CE 99 68 D5 A4 25 8C 20 A7 87 E6 B4 A6 A2 FE E5 B4 E9 9C D7 8 EC D3 75 1 2F BD D6 98 F6 51 AA 81 85 D 9A 84 19 16 65 46 C2 AB EC EE A6 B8 C2 BE F7 77 84 55 C7 8E ED 7B 37 FE DA F6 F6 5A BB 80 D7 34 DD B5 74 40 8E 6C D7 E0 EC F2 A8 B8 C6 B4 F7 29 AB 1B A F4 BD 71 BE FA 88 88 B9 5E AE CE 9E B0 5C B2 18 E4 FB DE A6 F3 FB B8 ED A3 C0 F1 FA B2 31 75 84 5A 35 43 EE E0 88 CE FE 7E B9 CF DA 92 E0 6C 0 9D E8 3C 8D AA AD C5 94 C0 91 C8 FA 70 46 BB 3 2D 44 A8 E3 A5 D3 F6 85 26 FF FD 94 40 24 73 2 4B 60 55 F3 AC A0 D6 D5 C5 86 C9 E0 60 80 89 A0 57 AB 7 B7 A4 C3 F2 82 2B F9 F6 C0 DD CC 5 85 6 2A 33 31 FF B4 84 C1 C1 81 C8 A9 E4 6F C3 EE 3 B7 8B 2E F6 C3 BC 93 2B B3 C9 98 D3 5F 6B A3 E2 3E D9 3E E4 A6 CA C3 84 DB F6 EA 79 31 CC DD 97 32 ED 66 BF C0 F0 93 65 A2 D8 CA 1E AD 70 37 C7 20 74 D0 AC BF C8 82 8B CA EF A5 B5 C6 D4 CA 42 E8 BF FC BF C8 E8 DA 6A B1 95 D6 60 41 3A 11 69 BE EC 92 A1 BF CA 9E 97 C8 BC F6 CA CF 9B 4A F8 2A 98 EF BF C7 E8 8F 7A BB 92 9E 7A 95 55 52 44 A9 A4 17 AD BC CE DF 96 D3 E8 FD 20 AD CD 95 A9 D 35 56 B0 8A EC 86 7C EF FF 92 AF 54 6F B4 2F B0 C8 AA A3 F9 9A C1 92 9F C9 F7 A6 F6 65 6C 56 B2 4A 61 AA 9B BF DC 3D E2 FD 93 EE 99 42 B2 5D 91 F A2 B2 E8 C9 8E D3 8E D4 B3

Tabel 6 Hasil Enkripsi dengan CBC

C. *Hasil Pengujian dengan CFB*

Teknik yang digunakan pada metode CFB hampir menyerupai metode CBC, namun hasil cipherteks terlebih dahulu dienkripsi sebelum di-XOR dengan plainteks. Selain itu, digunakan metode *shift register* 8 bit pada metode CFB. Berikut adalah hasil enkripsi dengan menggunakan metode CFB

D5 C6 BD 48 AE ED 9 9E E6 8A DA F0 5 85 CE 9E 5E E8 A3 F 74 AA DD 8B 9D D4 8F 93 83 9B C1 98 2E AA 85 E4 A D2 0 DF FC 86 9F A4 B 8F 9B 9F 91 8B 86 9D 8D DE 9A 83 88 98 85 84 9F 82 C1 9C F4 C6 A1 5E FA A4 13 D0 B2 96 D7 B5 43 B9 D4 80 4B F2 A3 2C 61 B8 C7 8B 84 D8 C9 80 8A 82 84 9E 66 82 C0 C5 1C C9 52 C7 BC C2 F1 B1 E 8F DF CC 95 8E 9D 96 9B D3 9D 86 8C D4 BB 8E 81 97 8F CC FF C8 AB 1B E1 E2 5A C9 F3 90 93 F0 A 9E 9B 85 59 A0 EC 19 6C AE DD CE 9B 91 8F 84 9E 84 84 88 66 BB CA 97 1E CF 0 CA FA 87 9F F2 31 8F DF CC A4 84 88 9D 8C 87 CB B5 D8 C7 B4 BA DD C2 BC CC FA C2 AC 5A FB F7 1F 9E E6 8A DA F0 A 98 D4 82 A EF FB 16 7C AE 93 9E 9B 91 9F 80 80 93 8F 98 66 A0 CB 97 16 C8 53 9E E1 97 CD B6 2 89 DE CC 93 81 9F 96 9A D3 80 9A C9 95 C9 93 89 92 85 85 EB CF EF 5A FE F4 1F DF E0 83 D1 B3 6 C4 E0 DD 1F DD A3 32 79 B9 C0 CE 80 87 C9 80 CC 82 84 9E 34 AA D6 C3 D D5 41 D2 B2 92 D3 B1 D 8F CF CC 83 81 9D 9B C9 92 C9 9A 81 9D 87 C1 8D 82 8C 83 EB D7 A7 5E FC E1 56 9E FA 83 C9 B9 D 8D 9B 9F 5F F2 E5 1E 7B AE 93 88 8C 95 9D 94 9E 93 92 CC 34 AA C8 DE 11 D5 53 DD F7 8C CB F0 1 85 CF 84 D4 87 8F D3 9D 9B 8C CE 80 99 99 80 8F 82 C1 8F EA C6 BB 5E FC F7 5A D1 F4 C2 CB B8 6 CA F6 83 45 EE A3 1E 76 AF 93 9A 81 91 C9 97 8D 9A 8D 89 3F BC 89 97 1B D9 53 DB E0 96 CC FC 43 8B D5 88 D4 98 86 9F 88 81 C9 87 8A 91 C9 82 8D 86 92 CC

Tabel 7 Hasil Enkripsi dengan CFB

V. ANALISIS KEAMANAN

Mini-Box block cipher menerapkan property *confusion* dan *diffusion* dengan memanfaatkan Block Cipher yang dimodifikasi dengan menggunakan Jaringan Feistel, modifikasi matriks, dan substitusi S-Box. Dengan adanya properti tersebut, diharapkan berbagai serangan yang dilakukan kriptanalis dapat diatasi dan dicegah. Dengan mengamati hasil enkripsi dan metode yang digunakan, dapat dianalisis pengaruh algoritma terhadap keamanan data. Dalam pembahasan berikutnya akan dijabarkan analisis keamanan algoritma dengan masing-masing jenis serangan yang mungkin terjadi.

A. *Brute Force Attack*

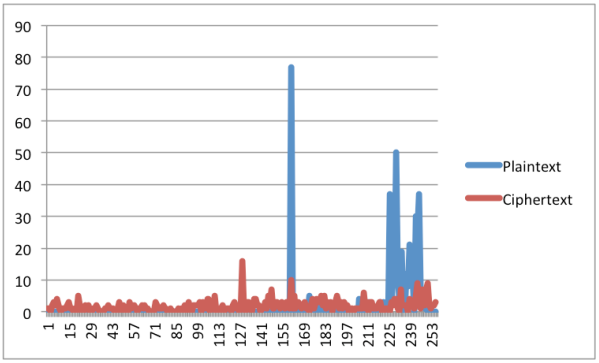
Brute force attack merupakan metode yang dilakukan penyerang dengan menebak dan mencoba kunci satu persatu hingga didapatkan hasil yang diinginkan. *Brute force attack* menggunakan metode *exhaustive search*, dimana penyerang mengenumerasi satu persatu kemungkinan. Dengan menggunakan metode ini, kunci kemungkinan besar akan ditemukan, namun waktu yang dibutuhkan bisa sangat lama, terutama jika kunci sangat panjang.

Pada algoritma Mini-Box kunci yang digunakan memiliki panjang 16 karakter atau setara dengan 128 bit. Kemungkinan kunci yang dapat dibentuk dari 128 bit adalah 2^{128}. Dengan kemampuan komputasi untuk menciptakan 10^6 kunci per detik, maka waktu yang diperlukan untuk mengenumerasi kunci satu per satu adalah $2^{128} / 10^6 = 3.4 \times 10^{32}$ setara dengan 3.93×10^{27} hari atau 1.075×10^{25} tahun. Waktu yang begitu lama untuk menemukan kunci yang digunakan megindikasikan bahwa algoritma ini hampir tidak mungkin dipecahkan. Terlebih lagi, algoritma Mini-Box juga menggunakan bilangan random untuk menjadi *seed* pada S-Box yang bisa merupakan bilangan integer berapapun. Untuk menemukan kunci dan bilangan seed yang digunakan akan membutuhkan waktu yang sangat lama sehingga dapat disimpulkan bahwa algoritma Mini-Box ini sangat sulit untuk diserang menggunakan *brute force attack.*

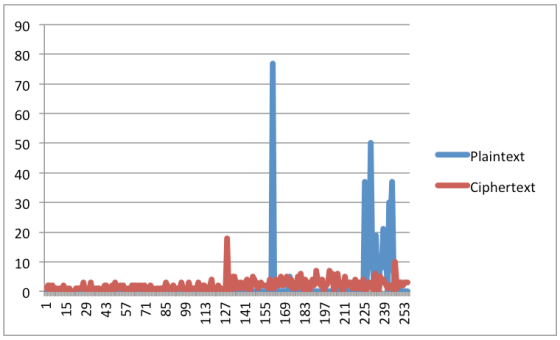
B. *Analisis Frekuensi*

Analisis frekuensi merupakan teknik yang digunakan untuk memecahkan cipherteks dengan memperhatikan frekuensi kemunculan huruf atau bigram pada cipherteks dan membandingkannya dengan huruf atau bigram yang sering muncul pada plainteks. Misalnya untuk Bahasa Inggris, huruf yang sering muncul adalah E, T, A, dan O, sedangkan huruf yang jarang muncul adalah Z, Q, dan X. Pengetahuan ini digunakan untuk memetakan kemunculan kata pada cipherteks.

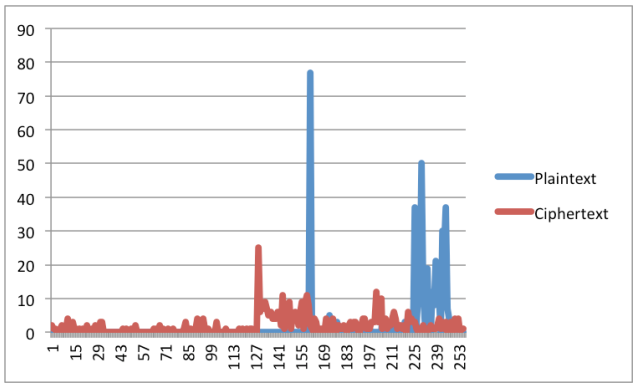
Untuk menganalisis pengaruh analisis frekuensi dengan keamanan algoritma Mini-Box ini dapat dilihat dari diagram perbandingan frekuensi kemunculan berikut ini.



Gambar 9 Diagram Frekuensi Kemunculan Byte dengan ECB



Gambar 10 Diagram Frekuensi Kemunculan Byte pada CBC



Gambar 11 Diagram Frekuensi Kemunculan Byte pada CFB

Ketiga gambar di atas adalah diagram yang menggambarkan frekuensi kemunculan masing-masing byte pada plainteks dan cipherteks dengan menggunakan data dan hasil pengujian pada bab IV. Sumbu x menggambarkan nilai byte atau ASCII dari karakter yang muncul sedangkan sumbu y menggambarkan jumlah kemunculan karakter tersebut. Dari ketiga contoh tersebut, dapat dilihat bahwa pada plainteks, prsebaran karakter yang digunakan tidak merata, karena umumnya karakter yang muncul pada kalimat adalah huruf atau tanda baca yang memang sering digunakan. Namun jika diperhatikan pada cipherteks dari ketiga percobaan, persebaran karakter hasil enkripsi sangat merata dan tidak berhubungan dengan kemunculan pada plainteks. Dapat dilihat bahwa huruf yang paling sering muncul pada plainteks belum tentu sama dengan huruf yang paling sering muncul di cipherteks.

Berdasarkan analisis tersebut, dapat disimpulkan bahwa serangan analisis frekuensi sangat sulit untuk diterapkan pada algoritma Mini-Box. Serangan analisis frekuensi cocok untuk diterapkan pada kasus jika pemetaan plainteks ke cipherteks pasti, namun dengan menggunakan algoritma Mini-Box, huruf yang sama pada satu plainteks belum tentu dienkripsi menjadi huruf yang sama pada cipherteks.

VI. KESIMPULAN DAN SARAN

Algoritma Mini-Box menawarkan solusi algoritma blok cipher yang sederhana dengan S-Box berukuran kecil yang dibangkitkan dinamik secara pseudorandom dan dengan jumlah round yang bergantung pada kunci eksternal. Algoritma ini memiliki tingkat *confussion* dan *difussion* yang baik. Berdasarkan analisis yang dilakukan, algoritma ini memiliki keamanan yang baik sehingga bisa menjadi alternatif dalam enkripsi blok cipher.

UCAPAN TERIMA KASIH

Penulis ingin mengucapkan terima kasih kepada Allah SWT, karena atas rahmat dan karunia-Nya lah makalah ini dapat selesai pada waktunya. Penulis juga ingin mengucapkan terima kasih kepada kedua orang tua yang tidak pernah letih mendukung dan mendoakan anaknya, serta Bapak Dr. Ir. Rinaldi Munir selaku dosen mata kuliah Kriptografi. Tidak lupa penulis juga ingin mengucapkan terima kasih kepada pihak-pihak yang secara langsung maupun tidak telah membantu dalam merampungkan makalah ini.

REFERENSI

[1] A. Menezes, P. van Oorschot, dan S. Vanstone. 1996. *Handbook of Applied Cryptography*. CRC Press

[2] Chandrasekaran, J. et al. 2011. *A Chaos Based Approach for Improving Non Linearity in the S-Box Design of Symmetric Key Cryptosystems.* In Meghanathan, N. et al.

[3] Munir, Rinaldi. 2015. *Slide Kuliah IF4020 Kriptografi: Algoritma Kriptografi Modern*

[4] Munir, Rinaldi. 2015. *Slide Kuliah IF4020 Kriptografi: Pengantar Kriptografi*

[5] Stallings, William. 2014. *Cryptography and Network Security (6th ed.).* Upper Saddle River, N.J.: Prentic Hall