Algoritma Cipher *Block* RG-1

Feryandi Nurdiantoro (13513042)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung,
Jl. Ganesha 10 Bandung 40132, Indonesia
feryandi@gmail.com

Ibrohim Kholilul Islam (13513090)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung,
Jl. Ganesha 10 Bandung 40132, Indonesia
13513090@std.stei.itb.ac.id

Muhamad Fakhrusy (13612020)
Program Studi Teknik Aeronautika & Astronotika
Fakultas Teknik Mesin dan Dirgantara
Institut Teknologi Bandung,
Jl. Ganesha 10 Bandung 40132, Indonesia
mfakhrusy@live.com

Abstrak* — Makalah ini membahas mengenai rancangan algoritma blok cipher "baru". Blok yang diambil oleh algoritma ini adalah 16x16 yang masing-masing sel nya berukuran 1 bit sehingga total blok berukuran 256-bit dari plainteks yang telah diubah ke dalam bentuk bit. Matriks ini akan melalui permutasi sebelum dimasukan ke jaringan feistel. Salah satu bagian blok akan disubtitusi, dan bagian lain akan masuk ke fungsi Vigenere cipher, selanjutnya di XOR lalu hasil keluarannya akan di transpose. Proses ini akan dilakukan sebanyak 16 kali. Kunci dibuat melalui perhitungan modulo dengan sebuah bilangan prima yang besar dan dibuat sebanyak 32 kunci. Kunci pertama akan digunakan pada saat permutasi, dan kunci kedua digunakan saat di dalam jaringan feistel. Terdapat 4 mode operasi yang dapat digunakan untuk algoritma ini, yaitu mode Electronic Code Book, Chain Block Cipher, Cipher Feedback, dan Output Feedback.

***Kata kunci*—block cipher, enkripsi, subtitusi, permutasi, transpose, perhitungan modulo, prima.**

I. PENDAHULUAN

Kriptografi merupakan teknik menyembunyikan pesan yang dapat dibaca oleh semua orang menjadi hanya dapat dibaca oleh orang-orang yang berhak membaca pesan tersebut. Didalam kehidupan sehari-hari kriptografi digunakan untuk mengamankan informasi penting mengenai seseorang.

Dengan penggunaan internet yang semakin berkembang pesat, keamanan menjadi salah satu topik penting yang harus ditangani di dalam mentransmisikan pesan-pesan dari satu tempat ke tempat lain melalui banyak jaringan. Pesan penting dan rahasia yang ditransmisikan melalui banyak jaringan ini dapat menimbulkan masalah dan kerugian jika tidak diamankan melalui suatu cara. Cara yang banyak dipakai didalam jaringan internet untuk melakukan pengamanan pesan ini adalah menggunakan teknik Block Cipher.

Block Cipher merupakan cara kriptografi untuk melakukan enkripsi dan dekripsi kepada sebuah pesan dengan cara membagi pesan-pesan tersebut ke dalam beberapa blok. Tiap blok dapat mempengaruhi hasil enkripsi berikutnya sehingga jika blok yang dikirim sama, belum tentu hasil enkripsi blok tersebut sama juga.

Makalah ini akan membahas mengenai algoritma cipher baru yang berbasis block cipher dengan nama RG-1 yang merupakan singkatan dari Roti Gulung - 1 karena metodenya yang menggulung-gulung blok plaintext hingga isinya tidak terlihat. Algoritma ini menggunakan permutasi, subtitusi, perhitungan modulo, algoritma vigenere dan transposisi matriks didalamnya. Algoritma ini hanya bekerja sempurnya pada komputer dengan arsitektur 64-bit.

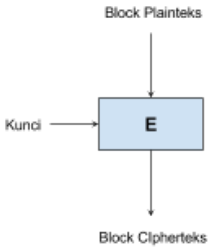
II. DASAR TEORI

A. Cipher Blok

Algoritma cipher blok merupakan cara kriptografi dengan membagi pesan menjadi blok-blok sebesar suatu ukuran. Algoritma ini akan menghasilkan *cipher block* dengan ukuran yang sama dengan *plain block* sehingga sangat menghemat ukuran terlebih saat dikirimkan melalui suatu jaringan seperti internet.

Implementasi algoritma ini pada umumnya terbagi menjadi tiga jenis yakni, Electronic Code Block (ECB), Cipher Block Chaining (CBC), Cipher-Feedback (CFB), dan Output-Feedback (OFB)..

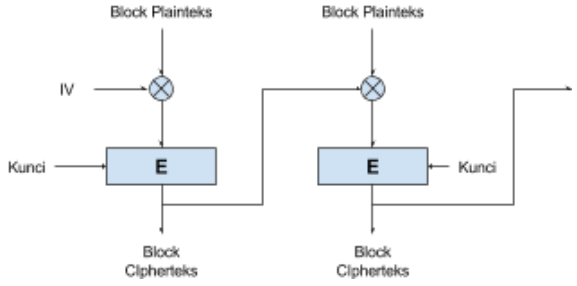
Pada ECB, setiap blok di enkripsi dan dekripsi secara independen dengan cara memasukan blok ke dalam suatu fungsi, yang nantinya akan menjadi blok cipher. Penggunaan metode ini kurang aman karena blok yang sama dapat menghasilkan blok cipherteks yang sama juga sehingga mengurangi keamanan. dan dapat dilakukan analisa kriptanalis.



Gambar 1. Cara kerja EBC

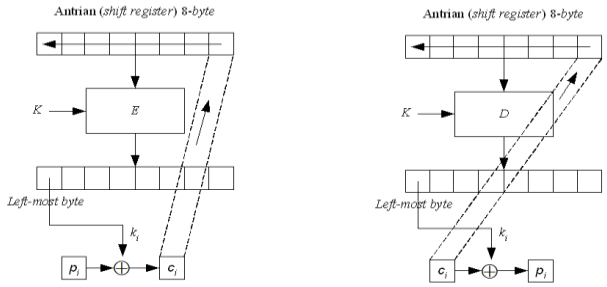
Pada metode CBC, tiap blok bergantung dengan blok yang lain dalam enkripsi dan dekripsinya. Enkripsi dan

dekripsi pada metode ini membutuhkan sebuah blok baru yang disebut IV (*Initialization Vector*) yang akan digunakan pada XOR pertama tahap dekripsi maupun enkripsi. Selanjutnya nilai yang di XOR kan dengan blok berikutnya adalah blok sebelumnya.



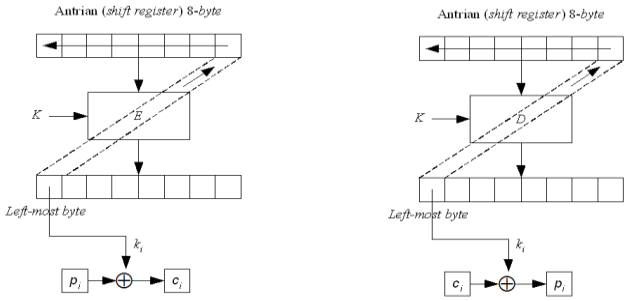
Gambar 2. Cara kerja CBC

Pada CFB, metode ini memperbaiki kelemahan yang ada pada metode CBC, seperti jika terjadi blok yang belum lengkap dan mengurangi kesalahan dekripsi data jika data rusak selama perjalanan. Metode ini dapat bekerja pada unit-unit blok yang cukup kecil sehingga dapat menyerupai *stream-cipher*.



Gambar 3. Cara kerja CFB [*Algoritma Kripto Modern, oleh M, Rinaldi. 2015*]

Pada metode OFB, hanya sedikit berbeda dengan CFB, perbedaannya terdapat pada IV yang digunakan pada fungsi E kedua dan seterusnya berasal dari hasil fungsi E dengan masukan Kunci dan IV sebelumnya. Hal ini menyebabkan OFB bisa mengolah blok selanjutnya tanpa harus menunggu XOR selesai dilakukan pada blok sebelumnya.



Gambar 4. Cara kerja OFB [*Algoritma Kripto Modern, oleh M, Rinaldi. 2015*]

Semua metode diatas, jika dibalik akan menghasilkan metode dekripsi yang dapat digunakan untuk membaca pesan yang telah terenkripsi melalu salah satu metode tersebut.

B. Prinsip Konfusi dan Difusi

Konfusi dan difusi merupakan salah satu cara kriptografer untuk memperkuat kriptografi sehingga tidak dapat melakukan analisis frekuensi dan memberikan perbedaan besar pada *ciphertext* hanya dengan sedikit perubahan pada *plaintext*.

Konfusi adalah metode untuk menghilangkan bentuk statistik yang dapat muncul pada sebuah bahasa atau kata-kata yang beraturan, hal ini dapat dilakukan dengan cara melakukan subtitusi sehingga mengubah *input* secara drastis ketika menjadi *output*.

Difusi adalah metode untuk menyebarkan pengaruh perubahan dari suatu karakter atau bagian pada *input* dapat mengubah banyak atau seluruh bagian *output*. Metode yang dapat digunakan untuk melakukan hal ini adalah menggunakan permutasi. Dengan difusi ini, pola-pola yang ada seharusnya hilang atau tersebar.

C. Jaringan Feistel

Jaringan Feistel merupakan salah satu struktur yang digunakan didalam kriptografi. Sistem yang digunakan pada jaringan Feistel ini adalah dengan membagi *plaintext* menjadi dua bagian. Masing-masing bagian akan diperlakukan dengan berbeda.

Bagian kanan *plaintext* akan di masukan kedalam suatu fungsi F dengan menggunakan kunci K_x dimana x merupakan nomor putaran yang sedang dilakukan. Bagian kiri *plaintext* selanjutnya akan di XOR dengan bagian kanan *plaintext* hasil keluaran setelah melalui fungsi F dengan kunci K_x. Selanjutnya kedua bagian tersebut kembali digabungkan kembali, namun dipertukarkan antara bagian kiri dengan bagian yang kanan.

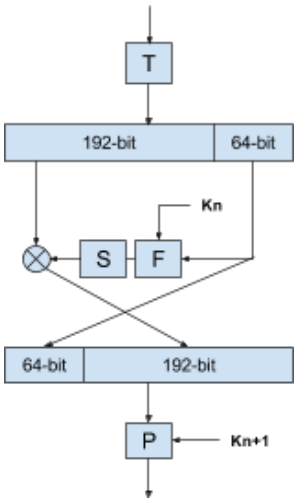
Proses tersebut dapat dilakukan berkali-kali untuk mendapatkan hasil yang lebih *random* dari sebelumnya. Cara untuk mendekripsi jaringan Feistel ini adalah dengna membalikkan proses diatas.

III. RANCANGAN ALGORITMA

Algoritma RG-1 menggunakan jaringan Feistel yang tak seimbang yang diputar sebanyak 16 kali. Besar blok pesan yang digunakan adalah sebesar 256-bit, dimana pada bagian kiri sebesar 192-bit dan pada bagian kanan sebesar 64-bit. Di dalam *round function* digunakan subtitusi yang dapat mengekspansi pesan sebesar 64-bit menjadi 192-bit sehingga dapat di XOR dengan bagian kiri pesan. Besar kunci adalah 128-bit, yang akan di-*generate* sebanyak 32 kali, 16 kunci digunakan untuk *round function* dan 16 kunci lainnya digunakan untuk permutasi. Sebelum sebuah jaringan Feistel dimulai, dilakukan transposisi terhadap pesan awal. Permutasi akan dilakukan setelah suatu jaringan Feistel dilakukan.

A. Jaringan Feistel

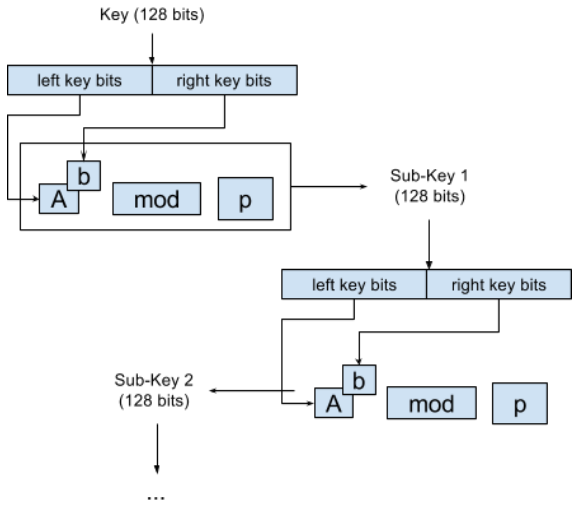
Jaringan Feistel yang digunakan pada algoritma ini sedikit dimodifikasi dengan penambahan fungsi transposisi sebelum jaringan Feistel dan fungsi permutasi setelah jaringan Feistel. Selain itu, juga jaringan Feistel yang digunakan tidak seimbang antara kanan dan kiri sehingga masing-masing ukurannya berbeda. Gambaran jaringan yang digunakan adalah sebagai berikut.



Gambar 5. Cara kerja algoritma RG-1

B. Mekanisme Key Schedule

Penjadwalan kunci yang digunakan pada block cipher akan menggunakan fungsi modulo sebagai fungsi bantu untuk membuat generasi kunci baru. Mekanisme key schedule dapat dilihat pada diagram dibawah ini.



Gambar 6. Cara pembuatan kunci pada algoritma RG-1

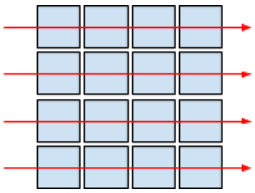
Dimana p merupakan suatu bilangan prima yang telah ditentukan didalam program. Putaran ini dilakukan sebanyak 32 kali untuk mendapatkan 32 kunci yang berbeda yang akan digunakan di masing-masing putaran pada mekanisme *block cipher*.

C. Permutasi

Boks permutasi ini menggunakan sebuah matriks permutasi P 16 x 16 yang dibangkitkan dari kunci internal ke i. Nilai pada matriks merepresentasikan bit urutan dari LSB (*least significant bit*). Matriks P memiliki kolom dari 1 s.d. 16 dimulai dari kiri dan baris 1 s.d. 16 dimulai dari atas, sehingga pojok kiri atas adalah kolom 1, baris 1. Prosedur untuk membangkitkan matriks permutasi ini adalah:

1. Mulai dari kolom 8 baris 8.
2. Isi nilai c dengan 0.
3. Isi nilai k dengan dengan 1.
4. Isi nilai m dengan bit ke k dari LSB kunci internal i.
5. Jika nilai m = 1, tambahkan c dengan 2, jika m = 0, tambahkan c dengan 1.
6. Isi matriks dengan c.
7. Tandai c telah dipakai.
8. Tambahkan k dengan 1.
9. Ulangi langkah 5-6 untuk mengisi matriks membentuk spiral sampai key habis.
10. Isi matriks membentuk spiral untuk nilai 1..256 yang belum dipakai.

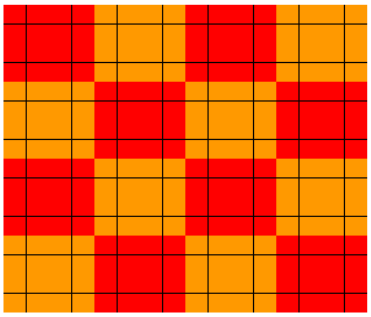
Pesan diisikan menurut nilai m pada masing-masing kotak dengan m merupakan bit ke m pada pesan. Kemudian matriks tersebut dibaca dari kiri ke kanan, dari baris 1 s.d. 16. Seperti ilustrasi pada gambar dibawah ini.



Gambar 7. Cara pengambilan pesan

D. Transposisi

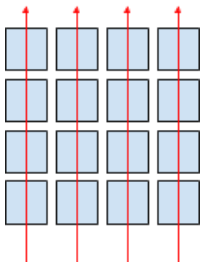
Transpose dilakukan dengan cara membentuk kode pesan yang telah melewati tahap-tahap jaringan Feistel menjadi matriks 16x16. Cara pemasukan bit-bitnya adalah dimulai dari kolom 1 baris 1 lalu diputar berlawanan arah jarum jam, berakhir di kolom 9 baris 9. Lalu, matriks tersebut akan dibagi tiap 4 blok besar, tiap bloknya terdiri dari matriks 4x4.



Gambar 8. Area yang ditranspose

Jadi, terdapat 1 matriks besar 4x4 di mana tiap kotak, adalah matriks kecil 4x4. Lalu operasi yang akan dilakukan adalah melakukan operasi transposisi matriks besar saja. Matriks besar akan di-transposisi matriks (pertukaran baris dan kolom), matriks kecil tidak mengalami transposisi.

Transposisi matriks dilakukan secara sederhana yaitu menukar setiap baris dengan kolom. Pada matriks besar, kotak baris ke satu kolom kedua akan ditukar dengan kotak baris kedua kolom pertama. Dan begitu seterusnya. Pesan diambil dari kotak paling kiri bawah ke atas hingga ujung. Lalu kembali ke baris kolom kedua, dan seterusnya hingga kanan atas. Seperti ilustrasi pada gambar dibawah ini.



Gambar 9. Cara pengambilan pesan

Pada program, transposisi dilakukan dengan cara mengubah pesan menjadi teracak menurut suatu aturan yang telah ditentukan. Aturan tersebut berisi peta yang menunjukan acakan untuk bit ke x diisi dengan bit ke y. Aturan yang digunakan untuk mengacak adalah sebagai berikut.

```
91, 134, 185, 244, 95, 138, 189, 248, 99, 142, 193,
252, 199, 198, 197, 256,
132, 133, 184, 243, 94, 137, 188, 247, 98, 141, 192,
251, 146, 145, 196, 255,
181, 182, 183, 242, 93, 136, 187, 246, 97, 140, 191,
250, 101, 144, 195, 254,
238, 239, 240, 241, 92, 135, 186, 245, 96, 139, 190,
249, 100, 143, 194, 253,
87, 88, 89, 90, 3, 14, 33, 60, 39, 38, 37, 64, 203,
202, 201, 200,
128, 129, 130, 131, 12, 13, 32, 59, 18, 17, 36, 63,
150, 149, 148, 147,
177, 178, 179, 180, 29, 30, 31, 58, 5, 16, 35, 62, 105,
104, 103, 102,
234, 235, 236, 237, 54, 55, 56, 57, 4, 15, 34, 61, 68,
67, 66, 65,
83, 84, 85, 86, 47, 24, 9, 2, 43, 42, 41, 40, 207, 206,
205, 204,
124, 125, 126, 127, 48, 25, 10, 11, 44, 21, 20, 19,
154, 153, 152, 151,
173, 174, 175, 176, 49, 26, 27, 28, 45, 22, 7, 6, 109,
108, 107, 106,
230, 231, 232, 233, 50, 51, 52, 53, 46, 23, 8, 1, 72,
71, 70, 69,
223, 168, 121, 82, 219, 164, 117, 78, 215, 160, 113,
74, 211, 210, 209, 208,
224, 169, 122, 123, 220, 165, 118, 79, 216, 161, 114,
75, 212, 157, 156, 155,
225, 170, 171, 172, 221, 166, 119, 80, 217, 162, 115,
76, 213, 158, 111, 110,
226, 227, 228, 229, 222, 167, 120, 81, 218, 163, 116,
77, 214, 159, 112, 73
```

Cara pembacaan aturan tersebut yakni, pada bit ke 1 diganti dengan bit 91 pada pesan sebelumnya dan seterusnya.

E. Round Function

Round funnction yang digunakan pada algoritma ini menggunakan metode subtitusi yang akan mengekspansi 64-bit pesan menjadi sebesar 192-bit dan dengan menggunakan kunci sebagai *seed* yang menentukan nilai subtitusi dan ekspansinya. Ekspansi dilakukan dengan cara mengalikan key sebesar 128-bit dengan pesan sebesar 64-bit sehingga menghasilkan 196-bit.

Setelah itu digunakan *subtitution box* yang sama dengan yang dimiliki Rijndael untuk melakukan subtitusi kepada pesan yang telah diekspansi menjadi 196-bit tersebut.

```
   | 0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
---|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
00 |63 7c 77 7b f2 6b 6f c5 30 01 67 2b fe d7 ab 76
10 |ca 82 c9 7d fa 59 47 f0 ad d4 a2 af 9c a4 72 c0
20 |b7 fd 93 26 36 3f f7 cc 34 a5 e5 f1 71 d8 31 15
30 |04 c7 23 c3 18 96 05 9a 07 12 80 e2 eb 27 b2 75
40 |09 83 2c 1a 1b 6e 5a a0 52 3b d6 b3 29 e3 2f 84
50 |53 d1 00 ed 20 fc b1 5b 6a cb be 39 4a 4c 58 cf
60 |d0 ef aa fb 43 4d 33 85 45 f9 02 7f 50 3c 9f a8
70 |51 a3 40 8f 92 9d 38 f5 bc b6 da 21 10 ff f3 d2
80 |cd 0c 13 ec 5f 97 44 17 c4 a7 7e 3d 64 5d 19 73
90 |60 81 4f dc 22 2a 90 88 46 ee b8 14 de 5e 0b db
a0 |e0 32 3a 0a 49 06 24 5c c2 d3 ac 62 91 95 e4 79
b0 |e7 c8 37 6d 8d d5 4e a9 6c 56 f4 ea 65 7a ae 08
c0 |ba 78 25 2e 1c a6 b4 c6 e8 dd 74 1f 4b bd 8b 8a
d0 |70 3e b5 66 48 03 f6 0e 61 35 57 b9 86 c1 1d 9e
e0 |e1 f8 98 11 69 d9 8e 94 9b 1e 87 e9 ce 55 28 df
f0 |8c a1 89 0d bf e6 42 68 41 99 2d 0f b0 54 bb 16
```

IV. EKSPERIMEN DAN PEMBAHASAN HASIL

Berikut ini merupakan hasil eksperimen yang dilakukan terhadap *plaintext* yang dimasukan ke dalam algoritma yang telah kami buat.

A. Menggunakan ECB

Menggunakan metode ECB, dilakukan ekspereimen terhadap plainteks berukuran 32-byte yang akan dijadikan 1 blok (berukuran 32-byte). Plainteks direpresentasikan dalam string dan hex tiap byte dan hasil enkripsi enkripsi direpresentasikan menggunakan hex tiap bytenya.

Kunci	`semoga kelar deh`

Plainteks (String)	`the quick brown fox jumps over?`
Plainteks (Hex)	`00 74 68 65 20 71 75 69 63 6B 20 62 72 6F 77 6E 20 66 6F 78 20 6A 75 6D 70 73 20 6F 76 65 72 3F`

Encrypted (Hex)	`2C A0 68 CD 19 6D 26 3F 7C 36 3D ED 0F 68 EA FE C1 F7 FD 14 83 87 C4 86 53 7A BC E0 AD EE A9 1C`

Decrypted (Hex)	`00 74 68 65 20 71 75 69 63 6B 20 62 72 6F 77 6E 20 66 6F 78 20 6A 75 6D 70 73 20 6F 76 65 72 3F`

B. Menggunakan CBC

Dengan menggunakan metode CBC, dilakukan eksperimen terhadap plainteks berukuran 44-byte yang akan dijadikan 2 blok (setiap blok berukuran 32-byte). Plainteks akan direpresentasikan dalam string dan hex tiap byte, sedangkan hasil enkripsi akan direpresentasikan menggunakan hex setiap bytenya untuk memudahkan pembacaan.

Kunci	`semoga kelar deh`

Plainteks (String)	`the quick brown fox jumps over the lazy dog`
Plainteks (Hex)	`74 68 65 20 71 75 69 63 6B 20 62 72 6F 77 6E 20 66 6F 78 20 6A 75 6D 70 73 20 6F 76 65 72 20 74 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 68 65 20 6C 61 7A 79 20 64 6F 67`

Encrypted (Hex)	`33 B2 20 13 DA 68 1E 4F 04 39 8D 58 97 1B E8 6E E1 2F B4 10 30 83 E4 02 63 8F 24 E1 93 A6 91 11 51 8D BD 90 11 EC F6 DC EE 1F AD 04 60 CE 90 F3 EA 17 35 B1 13 83 28 40 E1 38 C9 31 89 12 2A 44`

Decrypted (Hex)	`74 68 65 20 71 75 69 63 6B 20 62 72 6F 77 6E 20 66 6F 78 20 6A 75 6D 70 73 20 6F 76 65 72 20 74 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 68 65 20 6C 61 7A 79 20 64 6F 67`

C. Menggunakan CFB 8-bit pada 256-bit

Dengan menggunakan metode CFB, dilakukan eksperimen terhadap plainteks berukuran 32-byte. Plainteks akan direpresentasikan dalam string dan hex tiap byte, sedangkan hasil enkripsi akan direpresentasikan menggunakan hex setiap bytenya untuk memudahkan pembacaan.

Kunci	`semoga kelar deh`

Plainteks (String)	`the quick brown fox jumps over?`
Plainteks (Hex)	`74 68 65 20 71 75 69 63 6B 20 62 72 6F 77 6E 20 66 6F 78 20 6A 75 6D 70 73 20 6F 76 65 72 3F`

Encrypted (Hex)	`AE 30 F8 9D 9B 3E 2A 11 7B 57 67 7F 56 A1 BA 46 7F 7E 0F 65 D9 C9 AD 15 84 62 99 F0 F5 FC 48`
Decrypted (Hex)	`74 68 65 20 71 75 69 63 6B 20 62 72 6F 77 6E 20 66 6F 78 20 6A 75 6D 70 73 20 6F 76 65 72 3F`

Dari eksperimen tersebut, dapat disimpulkan bahwa algoritma RG-1 dapat digunakan sebagai block cipher karena telah berhasil mengenkripsi pesan dan mendekripsi kembali ke bentuk semula jika menggunakan kunci yang sama.

V. ANALISIS

Pada suatu algoritma kriptografi, hal yang paling disoroti adalah fitur keamanannya di dalam menjamin bahwa hanya yang berhak yang dapat membuka pesan dengan suatu kunci khusus. Untuk itu pada algoritma RG-1 ini juga dilakukan uji coba keamanan sebagai bukti kekuatan algoritma ini sehingga dapat dijadikan acuan untuk mengembangkan algoritma enkripsi lainnya atau digunakan di dalam keseharian. Beberapa hal yang penting harus ada dalam segi keamanan algoritma kriptografi adalah sebagai berikut.

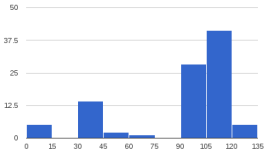
A. Analisis Frekuensi

Salah satu teknik kriptanalisis yang dapat dilakukan adalah dengan analisis frekuensi. Teknik ini merupakan teknik yang sering digunakan oleh kriptanalis untuk mencari plainteks dari cipherteks yang tersedia dengan mencocokan frekuensi kemunculan huruf-huruf.

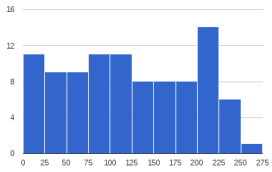
Dengan meratakan persebaran kemunculan huruf-huruf pada cipherteks, dapat mengecoh kriptanalis dalam melakukan analisis frekuensi. Sehingga percobaan persebaran frekuensi teks perlu dilakukan untuk melihat keamanan suatu algoritma kriptografi.

Di dalam melakukan percobaan, digunakan pesan sebagai berikut.

`the quick brown fox jumps over? they do it because of the strange things happens every day.`



Gambar 10. Histogram persebaran huruf sebelum di enkripsi



Gambar 11. Histogram persebaran huruf setelah di enkripsi

Pada histogram diatas, terlihat bahwa persebaran huruf pada plainteks (di representasikan dalam ASCII 256 bit) berubah secara signifikan pada cipherteks. Frekuensi kemunculan huruf-huruf yang tadinya berpola dan terlihat jelas, menjadi tidak jelas dan rata pada histogram cipherteks. Sehingga dapat dikatakan bahwa algoritma ini cukup aman dari serangan analisis frekuensi.

B. *Sedikit Perbedaan Plaintext*

Teks dapat saja berubah hanya sedikit jika merupakan suatu perintah yang terstruktur. Untuk itu, algoritma kriptografi juga harus dapat mengamankan pesan dengan tipe ini. Sehingga dilakukan pengujian terhadap teks yang hanya sedikit berbeda dengan kunci yang sama dan didapatkan hasil seperti di bawah ini.

transfer 3000 in USD from account 176 to 290
E2 F2 CE C7 E0 69 E1 5**E** 54 3E BF 1C 07 D0 C2 4C ED 71 D4 50 80 85 00 42 C3 7B F0 60 E9 C4 69 1**0** E3 EC 83 04 54 98 40 **9**B DF D8 9A **2**2 E5 44 90 75 A1 24 E5 04 00 D7 5B EE D4 39 0A 45 DA 54 62 3**7**

transfer 4000 in USD from account 176 to 290
E2 F2 CE C7 E0 69 E1 5**A** 54 3F BF 1C 07 D0 C2 4C ED 71 D4 50 80 85 00 42 C3 7B F0 60 E9 C4 69 1**1** E3 EC 83 04 54 98 40 **1**B DF D8 9A **A**2 E5 44 90 75 A1 24 E5 04 00 D7 5B EE D4 39 0A 45 DA 54 62 3**6**

Dari hasil pengujian tersebut, terlihat bahwa perubahan yang terjadi tidak terlalu signifikan terhadap cipherteks dari kedua plainteks. Sehingga algoritma kurang baik jika digunakan untuk melakukan pengiriman pesan yang sama secara berulang-ulang. Namun sulit untuk diubah menjadi teks yang bermakna jika hanya dengan mengubah di satu tempat, tetapi harus dibeberapa tempat sekaligus dan mengerti perubahan yang harus dilakukan, sehingga algoritma ini tergolong masih cukup aman.

C. *Sedikit Perbedaan Kunci*

Kunci merupakan hal yang paling penting didalam kriptografi, jika seseorang berhasil mendekati kunci yang sebenarnya, seharusnya algoritma kriptografi yang baik tidak akan mendekatkan hasil dekripsi ke plainteks yang sebenarnya sehingga menyulitkan kriptanalis dalam mengetahui kunci yang benar. Sehingga dilakukan pengujian sebagai berikut untuk mengetahui efek dari perbedaan kunci yang hanya sedikit.

kill him at 2350 when he sleep
d**o**nt tell anyone
15 FC 37 38 00 31 DC 63 B1 39 E9 46 D2 42 66 A4 C7 8C 79 57 88 CA 04 D4 5E B2 88 4E CE 2A A0 4C

kill him at 2350 when he sleep
d**u**nt tell anyone
3B 31 08 E3 18 D8 35 90 6A 2A 67 B 6B BF D9 60 F2 21 77 D2 85 F7 E5 50 DD F1 DA 67 81 4A 00 29

Dengan perbedaan yang sangat sedikit pada kunci, menyebabkan teks cipher yang benar-benar berbeda satu sama lain. Sehingga dapat disimpulkan bahwa penggunaan algoritma ini aman terhadap serangan melalui pencarian kunci.

D. *Repeated Block*

Letak kelemahan pada teknik *repeated block* tidak seluruhnya berada pada algoritma, tetapi pada metode yang digunakan untuk menjadikan algoritma RB-1 ini menjadi blok cipher. Sehingga kelemahan serangan ini akan sangat terekspos jika algoritma digunakan pada metode ECB seperti yang terlihat pada eksperimen dibawah ini.

ECB
grant access user to adarwawan. grant access user to adarwawan.
C9 4C 82 02 F3 E9 B7 FF D1 65 11 21 01 3F C0 09 94 F9 96 19 D4 E4 69 BD 7D CC 97 B1 01 E8 77 78 C9 4C 82 02 F3 E9 B7 FF D1 65 11 21 01 3F C0 09 94 F9 96 19 D4 E4 69 BD 7D CC 97 B1 01 E8 77 78

Hal ini terlihat sangat berbeda jika metode yang digunakan adalah CBC atau CFB dimana masing-masing blok memiliki keterhubungan satu dengan yang lainnya dan memiliki peran didalam dekripsi dan enkripsi. Hasil dalam menggunakan metode tersebut dapat dilihat dibawah ini.

CBC
grant access user to adarwawan. grant access user to adarwawan.
D7 E2 8F 8D 46 A7 EE 0F 9B 85 40 12 F2 E1 F9 6A 48 30 47 B0 49 B4 B2 17 09 86 7B 38 6B 33 6B 45 48 EB E1 0F 84 B0 A2 4B BD EC F9 30 1E 0C 32 CA E0 9D EB B5 82 1E 47 D4 29 B4 D5 73 EC 48 61 B5

CFB 8-bit untuk 32-byte
grant access user to adarwawan. grant access user to adarwawan.
BD 2A FC D3 9E 4B 22 91 F3 12 36 7E B9 A3 87 63 EB 13 04 08 91 5F A5 B5 07 D3 97 76 54 C1 FE C8 77 58 81 02 8B 0B 5B 51 0F EF A9 9D E0 DA 5A EF EB F9 C4 7E 70 28 60 C5 68 BB BB 7F B5 5A A1

E. Brute Force Attack

Algoritma ini menggunakan kunci sebesar 128-bit didalam melakukan enkripsi dan dekripsi. Sehingga terdapat sekitar 2^{128} kemungkinan kunci yang harus dicoba untuk memecahkan kunci ke-16 pada cipher teks. Jika kunci ke-16 telah didapatkan, penyerang harus mendapatkan kunci ke-15 dan seterusnya hingga mendapatkan kunci awal. Sehingga dibutuhkan sekitar 16 * 2^{128} kemungkinan yang harus dicoba untuk mendapatkan kunci awal. Dengan kemampuan komputasi pencobaan kunci sebanyak 10 juta kunci per detik, memerlukan waktu sekitar 17 x 10^{24} tahun untuk memecahkan kunci tersebut. Sehingga dapat diasumsikan bahwa penyerangan secara *brute force* tidak akan efektif untuk memecahkan enkripsi menggunakan algoritma ini.

Kekuatan algoritma ini tidak hanya berada pada kemungkinan yang ada pada kunci, tetapi juga ada pada kekuatan aritmatika modulus yang digunakan untuk mendapatkan 16 kunci yang digunakan selain itu modulus menggunakan bilangna prima yang cukup besar untuk melakukan perhitungan. Sehingga mempersulit penyerang untuk mengetahui nilai yang tepat untuk mendapatkan kunci lainnya.

VI. KESIMPULAN

Algoritma RG-1 ini merupakan algoritma baru yang kami kembangkan menggunakan modulus, subtitusi, transposisi, perkalian dan menggunakan jaringan feistel sebagai penguat algoritma sehingga memiliki faktor confusion dan difusion yang cukup tinggi. Melalui beberapa eksperimen yang telah dijelaskan sebelumnya, diketahui bahwa algoritma RG-1 ini memiliki kekuatan yang cukup baik dalam menangani beberapa serangan yang sering digunakan oleh kriptanalis. Sehingga dapat digunakan sebagai alternatif baru di dalam block cipher.

Dengan adanya algoritma RG-1 yang kami kembangkan ini diharapkan dapat memberikan inspirasi dan masukan kepada pengembang algoritma kriptografi lain dan dapat digunakan untuk memperkuan algoritma kriptografi block cipher simetris.

REFERENCES

[1] Munir, Rinaldi. *Algoritma Kriptografi Modern*. Maret 2015. Presentasi PowerPoint.

[2] Western Association of Schools and Colleges. *Modular Arithmatic.* http://www.artofproblemsolving.com/wiki/index.php/Modular_arithmetic/Introduction diakses pada 19 Maret 2015, 19.00.

[3] Schneier, Bruce and Kelsey, John. *Unbalanced Feistel Networks and Block-Cipher Design*. https://www.schneier.com/cryptography/paperfiles/paper-unbalanced-feistel.pdf diakses pada 20 Maret 2015, 17.00.

[4] LearnCpp. *Bitwise Operators*. http://www.learncpp.com/cpp-tutorial/38-bitwise-operators/ diakses pada 19 Maret 2015, 15.00.

[5] TechTarget. *Rijndael*. http://searchsecurity.techtarget.com/definition/Rijndael diakses pada 21 Maret 2015, 14.30.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 23 Maret 2016

Feryandi Nurdiantoro (13513042)

Ibrohim Kholilul Islam (13513090)

Muhamad Fakhrusy (13612020)