Riffle Block Cipher

Leo Lambarita Nadeak
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
lnadeak97@gmail.com

***Abstract*—Block cipher merupakan salah satu teknik enkripsi yang sering digunakan. Dan banyak algoritma *block cipher* yang dapat digunakan. *Riffle cipher* menjadi salah satu metode baru yang diajukan sebagai salah satu *block cipher*. Cipher ini terinspirasi dari teknik pengocokan kartu bridge *riffle*. Dan untuk menjamin hal ini dilakukan analisis keamanan untuk menunjukkan tingkat keamanan cipher ini. Analisis dilakukan berdasarkan hasil yang diberikan cipher ini, meliputi analisis prinsip *confussion* dan *diffusion*, *brute force attack*, dan juga analisis frekuensi.**

Keywords—block cipher, riffle, analisi keamanan

I. PENDAHULUAN

Internet merupakan jaringan terbuka, yang mana setiap orang dapat mengakses konten di dalamnya dengan bebas. Sehingga pesan apapun yang dikirimkan melalui internet dapat dibaca oleh siapapun. Dikarenakan hal ini, pesan tidak dapat dikirim begitu saja. Pesan perlu dimodifikasi agar orang yang tidak diberi akses tidak dapat memahami informasi di dalam pesan yang dibaca.

Dengna menggunakan metode enkripsi pesan dapat dimodifikasi menjadi pesan lain sehingga tidak dapat dibaca oleh sembarang orang tanpa kehilangan informasi didalamnya. Dan salah satu teknik enkripsi yang sering digunakan hingga sekarang adalah *block cipher*, yang memroses pesan perbloknya.

Block cipher sendiri terdiri dari bermacam – macam algortma yang telah dikembangkan oleh banyak orang, seperti DES, 3DES, AES, dan lainnya. *Paper* ini mengajukan algorimta *block cipher* baru dengan nama *Riffle Block Cipher*. *Block cipher* ini terinspirasi dari metode pengocokan kartu *bridge* yang dinamai dengan teknik *Riffle*.

II. DASAR TEORI

A. *Block Cipher*

Block cipher merupakan *cipher* yang memroses pesan perbloknya. Blok yang dimaksud merupakan pesan sejumlah n-bit dengan nilai n sesuai pendefinisian tergantung algoritma *block cipher* yang digunakan. Plainteks dibagi kedalam beberapa blok pesan, dan operasi enkripsi dilakukan pada setiap blok.

Dalam *block cipher* sendiri terdapat lima mode operasi yang dapat dilakukan, antara lain: [1]

- *Electronic Code Book (ECB)*, metode yang mengenkripsi setiap blok plainteks satu persatu dan secara independen.
- *Cipher Block Chaining (CBC)*, metode yang membuat setiap blok bergantung pada keseluruhan blok sebelumnya.
- *Cipher-Feedback (CFB)*, metode yang mengenkripsi pesan dalam satuan yang lebih kecil dari ukuran blok seperti 1 bit, 2 bit, dan lainnya.
- *Ouput Feedback (OFB)*, metode yang menggunakan n-bit hasil enkripsi sebagai antrian pada posisi paling kanan antrian yang akan dienkripsi.
- *Counter Mode*, metode yang menggunakan perantaian seperti CBC namun tedapat *counter* didalamnya yang nilai ditambah satu untuk setiap proses.

B. Teknik Pengocokan *Riffle*

Riffle merupakan salah satu teknik mengocok satu *pack* kartu *bridge* yang sering digunakan. Teknik ini membagi *pack* kartu menjadi dua bagian, kemudian menggabungkan kedua bagian dengan menyisipkan kartu dari bagian lain di antara kartu – kartu pada bagian lainnya [3], seperti yang dapat dilihat pada Fig. 1 dibawah.

Gambar 1. Teknik mengocok kartu bridge *riffle*

C. *Jaringan Feistel*

Jaringan feistel merupakan sebuah metode yang melakukan konversi terhadap sebuah fungsi ke dalam sebuah permutasi [2]. Struktur dari jaringan feistel sendiri dapat digunakan dalam menyusun algoritma *block cipher*. Keuntungan dari penggunaan jaringan feistel adalah kemudahan implementasi. Dengan menggunakan jaringan ini, dalam implementasi tidak perlu menyusun kembali algoritma dekripsi yang berbeda dengan dengan algoritma enkripsi. Alur algoritma dekripsi dapat disusun seperti algoritma enkripsi dengan mengikuti alur dari jaringan feistel, dengan arah yang berlawanan dari arah alur alogritma enkripsi [1].

D. *S-box*

S-box adalah sebuah matriks atau dapat juga disebut dengan tabel, yang berisi kombinasi nilai – nilai tertentu

yang dapat digunakan untuk mensubstitusi sebuah nilai bit menjadi nilai baru [1]. Cara menggunakan *S-box* adalah dengan melakukan pencarian nilai atau *lookup* tabel pada baris dan kolom tertentu, yang mana nilai baris dan kolom ini diperoleh dari nilai bit milik blok bit yang ingin dipetakan [1].

E. Prinsip Confussion and Diffusion

Kedua prinsip ini disampaikan oleh Claude Shannon pada tahun 1949 [1]. Kedua prinsip ini digunakan sebagai salah satu standar untuk melihat kesukaran dari sebuah algoritma enkripsi untuk dipecahkan melalui proses kriptanalisis.

Prinsip *confussion* adalah prinsip yang membingungkan kriptanalisis dengan cara menyamarkan hubungan antar elemen dalam enkripsi, seperti plainteks, cipherteks, dan kunci [1]. Sedangkan prinsip *diffusion* adalah prinsip yang memberi efek perubahan yang signifikan terhadap cipherteks ketika hanya sedikit (satu atau dua) karakter pada plainteks diubah [1].

F. Brute Force Attack

Brute Force Attack adalah jenis serangan untuk memecahkan sebuah cipherteks dengan menggunakan metode *brute force*, yaitu mencoba satu persatu kemungkinan kunci yang ada pada teknik enkripsi yang digunakan untuk memperoleh cipherteks tersebut [1].

III. RANCANGAN ALGORITMA

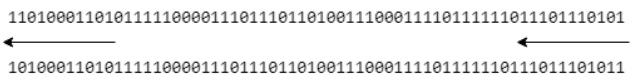
Algoritma ini membagi pesan kedalam blok dengan panjang 64 bit. Kemudian setiap blok diproses, yang mana setiap blok dipasangkan dengan masing – masing kunci internal yang telah dibangkitkan dari kunci masukan pengguna. Pemrosesan blok dilakukan sebanyak 8 putaran, hasil dari setiap putaran digunakan sebagai data untuk menyusun *S-box*. *S-box* ini digunakan untuk melakukan substitusi terhadap hasil enkripsi pada putaran terakhir. Secara garis besar keseluruhan proses di atas dapat dibagi kedalam tiga tahap, pembangkitan kunci internal, fungsi putaran, dan substitusi dengan *S-box*

A. Pembangkitan Kunci Internal

Kunci internal dibangkitkan dari kunci eksternal sepanjang 64 bit yang diterima dari masukan pengguna. Jika kunci tidak mencapai 64 bit, kunci diisi ulang mulai dari bit awal kunci masukan pengguna.

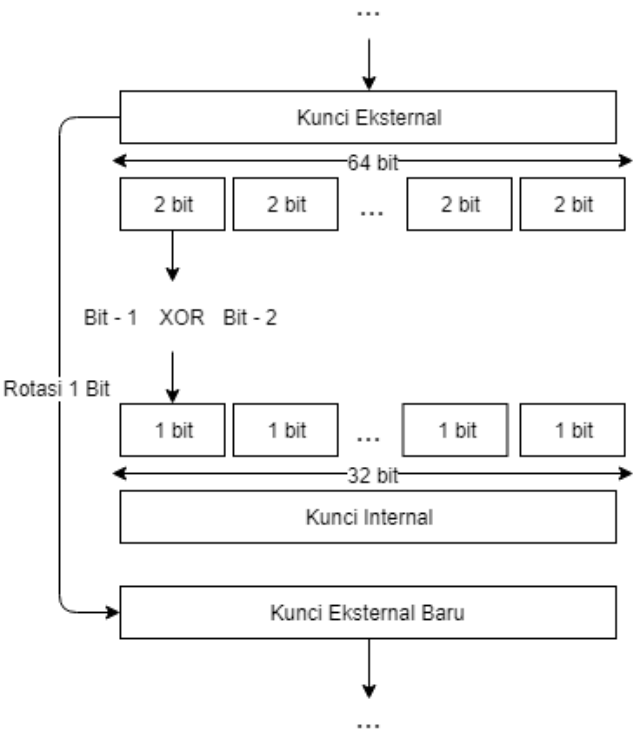
Pembangkitan kunci internal dilakukan dengan menggunakan hasil operasi XOR dari setiap bagian kunci sepanjang 2 bit. Kunci eksternal yang awalnya sepanjang 64 bit dibagi menjadi 32 bagian dengan panjang masing – masing 2 bit. Bit pertama dan kedua kemudian dioperasikan XOR, kemudian hasil dari 32 bagian tersebut disatukan menjadi satu kunci internal sepanjang 32 bit.

Kunci internal dibangkitkan dalam setiap putaran enkripsi. Dan setiap selesai satu putaran, bit kunci eksternal akan dirotasi sejauh 1 bit seperti yang digambarkan pada gambar 2 dibawah.



Gambar 2. Contoh rotasi yang dilakukan pada bit kunci eksternal

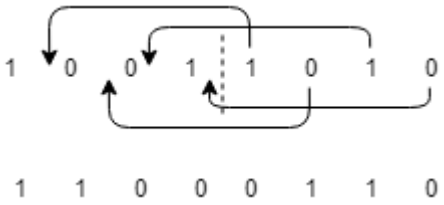
Rotasi kunci eksternal dilakukan untuk meningkatkan variasi kunci internal yang dipakai pada setiap putaran enkripsi. Sehingga secara umum pembangkitan kunci internal dilakukan mengikuti alur yang digambarkan pada gambar 3 di bawah.



Gambar 3. Alur pembangkitan kunci internal

B. Fungsi Putaran

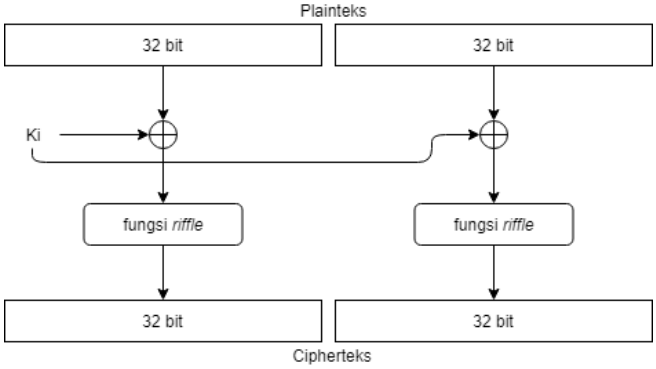
Fungsi yang dilakukan dalam setiap putaran adalah sebuah fungsi *riffle*. Seperti yang dijelaskan sebelumnya fungsi ini mengadopsi sistem pengocokan kartu bridge *riffle*. Namun untuk membuat algoritma lebih terstrukstur, kartu dari bagian lain disipkan diantara dua bit yang berurutan pada bagian tujuan. Untuk lebih jelasnya dapat dilihat pada gambar 4 dibawah yang mendemonstrasikan fungsi swapping card pada angka dengan panjang 8 bit.



Gambar 4. Demonstrasi fungsi *riffle* pada angka sepanjang 8 bit

Pada awal proses, 64 bit blok pesan dibagi menjadi dua bagian yang masing – masing bagian sepanjang 32 bit. Setelah itu setiap bagian dijalankan fungsi *riffle.* Keunikan dari fungsi *riffle* adalah, ketika fungsi diulangi sebanyak n kali, dengan nilai n adalah panjang bit dibagi 2, posisi bit kembali ke pisisi awal. Hal inilah yang mendasari jumlah putaran yang dilakukan adalah sebanyak 8 kali. Karena ketika dilakukan dekripsi pesan, setelah dilakukan 8 putaran lagi, posisi bit kembali ke posisi awal sebelum dienkripsi.

Tidak cukup dengan hal diatas, untuk meningkatkan variasi hasil dari fungsi putaran ini, sebelum fungsi *riffle* dijalankan pada bagian blok, setiap bagian dioperasikan XOR dengan kunci internal. Hal ini meningkatkan keragaman hasil yang diberikan dari setiap proses putaran. Sehingga secara keseluruhan, proses sekali putaran dilakukan sesuai alur yang digambarkan pada gambar 5 dibawah.



Gambar 5. Jaringan fiestel dalam satu putaran enkripsi

C. S-box

S-box digunakan untuk melakukan substitusi terhadap bit hasil enkripsi pada putaran terakhir. *S-box* yang digunakan merupakan sebuah tabel atau matriks dengan jumlah baris dan kolom sebanyak 16. *S-box* ini disusun secara random dengan menggunakan *seed* tertentu yang dikalkulasi berdasarkan kunci eksternal (kunci yang sama digunakan untuk membangkitkan kunci internal putaran enkripsi) dari pengguna.

Setiap *cell* pada tabel *S-box* berisi bilangan *hexadecimal* sepanjang 2 digit. Sehingga diperlukan dua modul random dengan masing – masing memiliki nilai seed berbeda. Modul pertama memiliki *seed* dengan nilai berupa total penjumlahan dari representasi *integer* dari setiap karakter kunci. Dan *seed* pada modul kedua merupakan panjang key jika setiap karakter direpresentasikan dalam bit.

Digit pertama dari nilai *cell* tabel *S-box* adalah representasi *hexadecimal* dari bilangan acak yang dihasilkan oleh modul *random* pertama. Sama halnya dengan digit kedua yang dihasilkan modul *random* kedua. Salah satu contoh *S-box* yang dihasilkan dapat dilihat pada gambar 6 dibawah.

```
e2 62 71 fb a7 f2 ca 92 85 44 eb c8 04 59 83 a5
fe dc 19 14 11 ad 80 13 1d 24 d8 c0 f1 0c bc 93
8c f0 00 52 22 9b 61 80 99 1c 19 44 1f 61 fd 10
b6 d9 96 b7 e1 05 95 04 1d 1c 79 63 e1 57 cd 81
90 58 6a e3 be 89 36 e8 17 a9 49 0c 36 27 22 d5
44 51 cf 67 1f a6 46 3f fd 6a 5b 7d 68 3c 77 96
09 33 1f b5 cc af 9b 55 6b 16 4a da 15 35 3e ac
a5 03 94 ac 9f 54 e5 ea fa f9 64 6b 6b 54 81 a9
55 a5 47 fd 0e b0 6c 6c 26 17 8f f6 4f 39 b6 50
6a 60 15 f5 8f e5 40 43 d9 73 eb c8 c3 9a 22 a4
5c ce d1 39 5c 13 33 a2 6c 5a d7 0f cb a0 a3 59
36 ac f4 9c 6e 76 e5 f3 62 1f dc fa 4a 62 71 a7
d9 97 37 de 16 0a 1b f4 e6 33 d7 e7 ec 38 5c 3a
7c 30 3b e9 2d 2b 62 66 c7 62 b3 a7 9c ce 94 15
58 e2 7d ad da 64 81 30 9c dd c8 ee ca c1 1b 37
0d 38 76 30 84 24 d5 fe e8 ed 22 fc 0d 81 c6 14
```

Gambar 6. Contoh *S-box* yang dihasilkan dari modul *random.*

Metode *lookup* yang dilakukan pada *S-box* adalah dengan mendapatkan nilai baris dan kolom melalui bit hasil enkripsi setelah putaran terakhir. Hasil enkripsi sepanjang 64 bit dibagi kedalam 8 bagian sepanjang 8 bit. Kemudian 4 bit pertama bagian tersebut dikonversi kedalam nilai basis 10 dan menjadi nilai baris, sedangkat 4 bit terakhir digunakan sebagai nilai kolom dalam melakukan *lookup* pada tabel *S-box.*

Dengan metode *lookup* di atas, setiap 64 bit hasil enkripsi, setelah dipetakan menghasilkan 16 digit bilangan hexadecimal.

IV. Simulasi dan Hasil

Implementasi dari algoritma cipher *riffle* dapat dilakukan dengan menggunakan bahasa pemograman apapun. Salah satu contohnya dengan menggunakan bahasa *python.* Dan telah dilakukan implementasi cipher dengan menggunakan operasi ECB dan CBC. Untuk mode CBC sendiri, nilai IV diinisiasi dengan bit 0 sepanjang 64 bit.

Hasil yang diberikan kedua mode operasi ini berbeda cukup signifikan seperti yang terlihat pada tabel I dibawah.

Tabel I. Hasil Enkripsi dari Implementasi Cipher Riffle

Plainteks
Alice tolong kirimkan dokumen A ke rumah saya malam ini
Kunci
72hd9aj0
Hasil dengan mode ECB
8720b7414ff3a37af46e1b2434908eeb827a241623302e89fc200d46235e5d6e5c57681a78b6ca14ff32ef232da6e18f
Hasil dengan mode CBC
8720b7414ff3a37a6873384adf68b27e570b0b0c9a6553fa186e9d2bafe8303ce232b0ce184ed15955168c635e18fe42

Contoh kedua dilakukan dengan menggunakan plainteks yang lebih panjang. Plainteks diambil dari bagian teks yang diperoleh dari salah satu artikel *New York Times.* Hasil yang diberikan dapat dilihat pada tabel II dibawah.

Tabel I. Hasil Enkripsi dari Implementasi Cipher Riffle Pada Teks Panjang

Plainteks
Guy grew up in an impoverished, highly segregated part of West Baltimore near what was now the focal point of the street clashes, but she had long since climbed into a different stratum of the city's society; she was working as an information-technology project manager for T. Rowe Price, the Baltimore-based mutual-fund giant. Seeing her old neighborhood erupt changed her life. After long discussions with her husband, who manages the office of a local trucking company, she quit her job and went to work for a community mediation organization. "It just felt like it was the work I was

supposed to be doing," she said.
Kunci
xuipoiw5
Hasil dengan mode ECB
cea3e520fce0012420da564e2e04634130af5c07830197b9e8e db878f48c4a7a160b0b5a160741fe655fed2b6e564924a75c76 36342480597b13345ca35a9d277607f4f4637698f0766e0d75 dfd82d419f8c7e102cd3621e8307cb785a17e2f3f03047fed5fc 9dfecc5eaa485e162e1e0c726957473cf43da6f99073ae3272b 3fff6d86a560180e66869e323eb3fce5616afff48656a47e17ae 50af45c7897fecc747414ce8b3de682d301734f5742312ecb5d 92f4f4f54e88bbf0ae59b88bb3886ea38299f3aa17cc1790ad6 5521e20d4382c0a07a3af90fe568ea33cfe5690659aff0c4839b 2d69f68aa7dfae62a2378adccaa685d235e0b62fa785656afb7 5ac14f16b816805947b8b04ad8b709ed6807bfed882bff1e5c8 3ea928831173cfe01306e83ccd6f01859620d6e0741a3328c5e 0165744a46a0a3eb97878e7c2307edeb233441d6ae340723c1 1aeb5016527c3f6e575e275e5224c50756daf6d446b2070b1bf 634aa82e6ebdf6ebf6924c8fc947eccb7f66865410756ea7836 68b16ed6c1265e3f5a7610245c7f72207a7ae634f5407a687c4 de1af9a47f69a68cc5eea7853422a27e22af6c17e69bdf04834c 18f7afaf0d67287ce07e68037b8a746c6a368ce8836c8476810 657e473418b8e25653aaaaa675e17ae8165ea3272050d8e618 34caaa5041527e2d5ed85673fa52bffc0c4dd601654d8301300 a7a727b526532f4e3b16916fc32c9f0b3781a774a3777f456
Hasil dengan mode CBC
cea3e520fce00124117d9af45d2d16422a100a8211b1b11442 5af432322a39ed82722d9fccb72eed18c107c9798b233d32a3e b01e1325623822a16cc5ed4d120078e537452d44120136949 2c38524dccf0c9e10cc6ea48df68fcfccb2a42f4ce9f01260bd5 56f473ccfad3cbfe30cc20f0ce23f3c90b7e8febb88f56b35d23 b782e0b7df0d3241411be68827897224319a18070748f93c72 d414b6e6209fd165659ddfedcee36a502e2673fe8bc12042caf cf9e5973f4eed32d6fe88c1a37c68a7143f3223aab12eeb0db1 16d3e2cc2e468f807bb6d4e13d6534240497975e9ff43068e0f 4e2e6a33247fee01b3cc1dfedcaeb0bc9568af37a42a71b6842f 3fe47f6caf95cb7721ee1092e10f4fe8a0bff48a3789d34c1723 78756260bf4b25a574e272688af205a77caef5c18423cc19fdf 165e372707eba08e04b7f410f314e2c1fe0aa3f423d65e7ac57 ad639fe8e34788e7d4a0b4841c8f475233420e52e88463c983 ced80165af0bff51bfced2e3c42d120529f18207216245e23c1 cc14c80c49ad8c688f7ced4e398cfa2616428e1672235d32394 7d1a626ad0b8231945eb2e23d46a6aec1cc16b6fce2485ed45e 40823d8338feebfa327a0187b7ce20b85e342bf58c879216fffa 5565c1552ee35d89470c555727fcf4e334ca0bf442e6d840721 38e240c97a6d552da2c01560b2e52525ced56b316cc42ad7f5 2977ce2770152c15eb365d6ced4837bc127fe3d7e1b04f0eafe a0

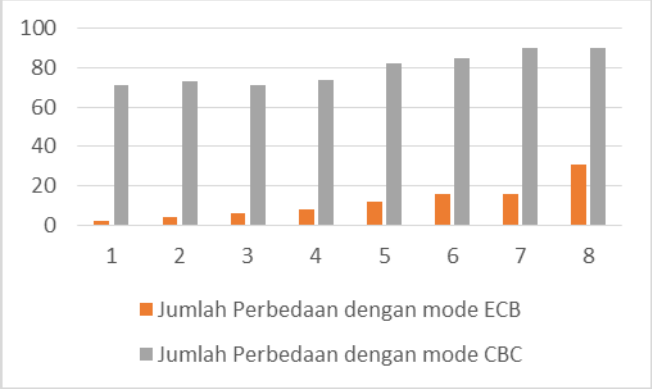
Perlu diperhatikan dibeberapa kasus panjang cipherteks tidak sesuai dengan panjang plainteks, yang mana setiap 8 karakter seharusnya diubah kedalam cipherteks sepanjang 16 karakter. Namun karena terdapat kemungkinan pesan yang diberikan pengguna panjangnya tidak selalu kelipatan 64, perlu dilakukan padding yang juga dapat mempengaruhi hasil enkripsi.

V. Analisis Keamanan

A. *Analisis Confussion dan Diffusion*

Konversi dari karakter ascii kedalam bilangan hexadecim al sudah cukup menyamarkan hubungan plainteks dengan cip herteks. Selain itu setiap karakter plainteks tidak tepat dipeta kan pada karakter cipherteks secara spesifik. Ketika posisi ka rakter diubah, enkripsi yang diberikan juga berbeda. Sebagai contoh, plainteks "abcde" dienkripsi menghasilkan cipher tek s "cea3ad201ed3f323", dan ketika plainteks "edbca" dienkri psi dengan kunci sama diperoleh cipherteks "cea3ad20d47d2 73d".

Dapat dilihat hasil yang diberikan ketika posisi karakter diubah berbeda cukup signifikan, yang mana perlu diperhatikan kunci yang digunakan sama untuk kedua kasus. Dengan kata lain *S-box* yang dihasilkan untuk kedua kasus pasti sama karena seed yang digunakan pun sama.



Gambar 7. Perbandingan jumlah karakter yang berubah pada mode ECB dan CBC

Untuk analisis *diffusion*, algoritma memberikan hasil yang cukup baik namun jika menggunakan mode CBC. Dalam grafik pada gambar 7 di atas (sumbu x adalah banyak karakter yang diubah), jumlah karakter yang berubah pada mode CBC jauh lebih tinggi dibanding ketika menggunakan mode ECB. Meskipun begitu, mode ECB memberikan jumlah perubahan yang linear hubungannya dengan jumlah karakter yang diubah.

B. *Brute Force Attack*

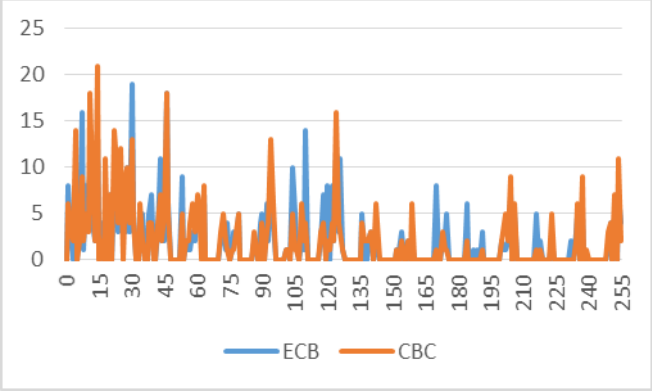
Kunci yang digunakan dalam cipher *riffle* adalah kunci sepanjang 64 bit. Jika dilakukan *brute force attack* dengan mencoba kemungkinan kunci satu persatu, maka jumlah kunci yang perlu dicoba ada sebanyak 2^{64} kunci atau sama dengan 18.446.744.073.709.551.616 kunci.

Misalkan digunakan mesin komputasi yang dapat mencoba 10.000 kunci dalam 1 detik, maka total waktu yang diperlukan untuk mencoba seluruh kunci adalah 18.446.744.073.709.551,616 detik atau sama dengan

2135039823346O1.28 hari atau sekitar 5,8 x 10^{11} tahun. Hasil ini menunjukkan *brute force attack* sangat tidak cocok digunakan untuk memecahkan hasil enkripsi dari cipher *riffle*.

C. *Brute Force Attack*

Setelah diperoleh cipherteks dengan melakukan enkripsi terhadap teks "Guy grew up in an impoverished, highly segregated part of West Baltimore near what was now the focal point of the street clashes, but she had long since climbed into a different stratum of the city's society; she was working as an information-technology project manager for T. Rowe Price, the Baltimore-based mutual-fund giant. Seeing her old neighborhood erupt changed her life. After long discussions with her husband, who manages the office of a local trucking company, she quit her job and went to work for a community mediation organization. "It just felt like it was the work I was supposed to be doing," she said", dilakukan penghitungan frekuensi kemunculan setiap bit (dari bit bernilai 1 sampai dengan bit bernilai 255) pada cipherteks yang dihasilkan. Data frekuensi tersebut dapat dilihat pada gambar 8 di bawah.



Gambar 8. Perbandingan frekuensi kemunculan bit pada mode ECB dan CBC

Perhitungan frekuensi bit dilakukan pada hasil enkripsi dengan menggunakan mode ECB dan CBC. Dapat dilihat bit yang sering mucul pada kedua mode cukup dominan pada bit – bit bernilai kecil yaitu pada range 0 – 45. Meskipun masih cukup banyak bit besar (di atas 45) yang masih muncul, hasil tersebut masih kurang baik, yang mana ketika frekuensinya merata lebih mempersulit dilakukannya analisis frekuensi saat melakukan kriptanalisis.

VI. Kesimpulan dan Saran Pengembangan

A. *Kesimpulan*

Cipher *riffle* sudah cukup baik dalam memenuhi prinsip *confussion* dan juga prinsip *diffusion.* Selain itu, cipher ini efektif dalam menangani *brute force attack* karena kerumitan kunci yang digunakan. Namun dalam hal serangan yang menggunakan analisi frekuensi, cipher ini masih memiliki sedikit kekurangan. Cipherteks yang dihasilkan cipher ini masih relatif dominan pada range tertentu dibanding diluar range tersebut. Yang mana hal ini dapat menjadi celah dalam melakukan kriptanalisis menggunakan analisis frekuensi.

B. *Saran*

- Dalam paper ini belum diperhatikan biaya komputasi yang diperlukan seperti penggunaan memori dan lama komputasi. Alangkah lebih baik jika cipher dapat bekerja sefektif dan efisien mungkin. Oleh karena itu, akan lebih baik jika komputasi dari cipher *riffle* dapat diperhitungkan juga

References

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

[2] Schneier, B., & Kelsey, J. (1996). Unbalanced Feistel networks and block cipher design. Lecture Notes in Computer Science, 121–144. doi:10.1007/3-540-60865-6_49

[3] https://coolcardtricks.net/shuffle/riffle-shuffle

[4] https://www.dailymail.co.uk/sciencetech/article-3011046/How-shuffle-cards-like-pro-Mathematician-shows-riffle-technique-effective-flashy-overhand.html