STANDARD Block Cipher

Algoritma *Block Cipher* dengan Transformasi Fourier dan Cosine

Muhammad Rizki Duwinanto - 13515006
Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13515006@std.stei.itb.ac.id

Dandy Arif Rahman - 13516086
Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13516086@std.stei.itb.ac.id

***Abstract*—Algoritma *Block Cipher* saat ini sangat dibutuhkan dalam melakukan pengenkripsian suatu berkas. Algoritma STANDARD adalah algoritma *Block Cipher* yang bekerja pada blok 256-bit dengan kunci 256-bit. Algoritma STANDARD menggunakan jaringan feistel dalam melakukan enkripsi dan menggunakan fungsi substitusi, permutasi, pencampuran, dan fungsi ronde. Di dalam fungsi ronde terdapat transformasi *fourier* dan *cosine* yang merupakan keunikan dari algoritma tersebut. Algoritma ini dapat dijalankan dalam berbagai mode pengenkripsian. Dalam eksperimen didapatkan bahwa algoritma masih dapat diserang dengan frekuensi analisis, namun kuat terhadap serangan *bruteforce*.**

Keywords—Transformasi, Fourier, Cosine, Block, Cipher, Enkripsi

I. Pendahuluan

Seiring perkembangan zaman, hal yang juga sangat terasa perkembangannya adalah teknologi informasi. Kebutuhan akan pertukaran informasi semakin hari semakin bertambah. Tentunya pertukaran informasi harus didukung juga dengan keamanan akan data didalamnya. Sebenarnya sejak dahulu kala, sudah bermunculan berbagai macam metode untuk mengamankan data yang dipertukarkan, hal ini yang nantinya menjadi cikal bakal dari kriptografi. Tujuan dari enkripsi sendiri adalah memastikan keamanan terhadap data tersebut dari tangan orang yang tidak berkepentingan.

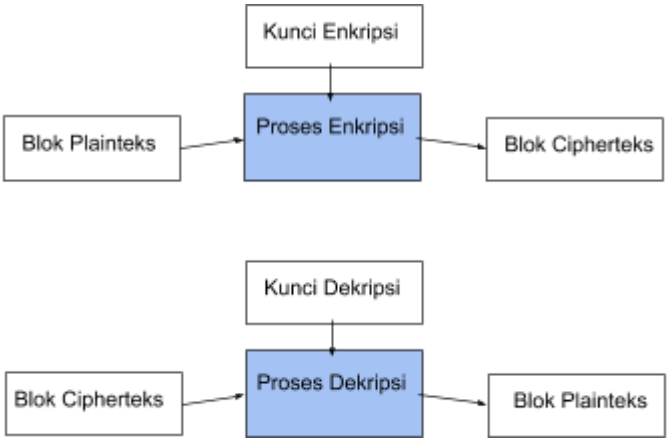
Berdasarkan waktu nya, algoritma kriptografi terbagi menjadi algoritma kriptografi klasik dan algoritma kriptografi modern. Pada algoritma kriptografi klasik, metode yang digunakan adalah substitusi dan transposisi terhadap karakter-karakter pada pesan tersebut. Pada zamannya, memang algoritma kriptografi klasik ini terbilang sulit dipecahkan, tetapi seiring dengan perkembangan teknologi komputer, memecahkan algoritma kriptografi klasik menjadi mudah, maka dari itu dibutuhkan metode yang lebih canggih. Karena kebutuhan tersebut, muncul algoritma kriptografi modern yang berjalan pada komputer dan melakukan kriptografi pada bit-bit nya dibanding karakter-karakter nya.

Jika dibandingkan dengan algoritma kriptografi klasik, algoritma kriptografi modern beroperasi pada sekuens *binary bit,* bukan pada sekuens karakter. Algoritma kriptografi modern memiliki algoritma yang publik, tetapi tetap aman, karena kesulitan terletak pada rumit nya komputasi, dan kerahasiaan *secret key,* berbeda dengan kriptografi klasik yang algoritma nya sendiri rahasia. Berbagai algoritma kriptografi modern telah ditemukan, tetapi seiring dengan pesatnya perkembangan teknologi, banyak juga algoritma yang berhasil dipecahkan dengan waktu yang wajar. Oleh karena itu para kriptografer terus berusaha membuat algoritma-algoritma yang lebih rumit lagi.

II. Dasar Teori

A. Block Cipher

Block cipher mengambil blok bit dari plainteks dan menghasilkan blok bit cipherteks, umumnya berukuran sama. Ukuran blok ditetapkan pada skema yang digunakan. Pilihan ukuran blok tidak secara langsung mempengaruhi kekuatan enkripsi tersebut, selain itu kekuatan cipher juga bergantung pada panjang kunci.



Gambar 1. Skema Enkripsi dan Dekripsi Block Cipher

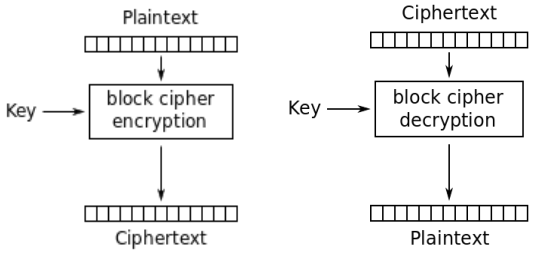
Pada block cipher dikenal lima mode operasi, yaitu *Electronic Code Book* (ECB), *Cipher Block Chaining* (CBC), *Cipher Feedback* (CFB), *Output Feedback* (OFB), dan mode *counter*.

1. *Electronic Code Book* (ECB)

   Electronic Code Book (ECB) adalah mode operasi untuk blok cipher, dengan karakteristik bahwa setiap

kemungkinan blok plainteks akan memiliki suatu nilai cipherteks yang sama dan sebaliknya. Dengan kata lain, nilai plainteks yang sama akan selalu menghasilkan nilai cipherteks yang sama. ECB digunakan ketika volume plainteks dipisahkan menjadi beberapa blok data, yang masing-masing kemudian dienkripsi secara independen dari blok lain atau dapat dikatakan bahwa hasil enkripsi untuk suatu blok tidak akan mempengaruhi proses enkripsi blok yang lain. Bahkan, ECB memiliki kemampuan untuk mendukung kunci enkripsi yang terpisah untuk setiap jenis blok.

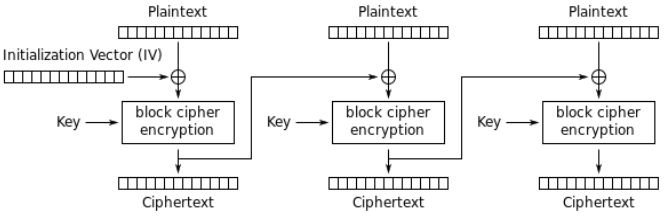
Namun, ECB bukan sistem yang baik untuk digunakan dengan ukuran blok kecil (misalnya, lebih kecil dari 40 bit) dan mode enkripsi yang identik. Ini karena beberapa kata dan frasa dapat digunakan kembali cukup sering, sehingga bagian blok cipherteks yang sama akan muncul, membuat kriptanalis mudah untuk memecahkannya.



Gambar 2. Operasi block cipher dengan mode ECB

2. Cipher Block Chaining (CBC)

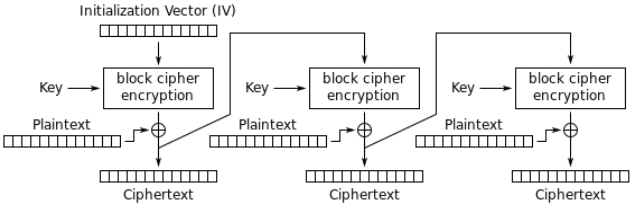
Cipher Block Chaining (CBC) adalah mode dimana setiap blok akan dienkripsi dengan memanfaatkan hasil enkripsi (cipherteks) blok sebelumnya. Pertama kali plainteks di XOR dengan *Initialization Vector* (IV) lalu dienkripsi, kemudian Cipherteks yang dihasilkan pada proses enkripsi akan dilakukan operasi XOR dengan blok plainteks selanjutnya, kemudian hasil operasi tersebut akan dienkripsi. Metode ini sering digunakan karena blok plainteks yang sama tidak akan dienkripsi menjadi blok cipherteks yang sama, sehingga proses kriptanalisis dapat menjadi lebih sulit. Kesalahan satu bit mengakibatkan keseluruhan cipherteks tidak bisa didekripsi.



Gambar 3. Operasi block cipher dengan mode CBC

3. Cipher Feedback (CFB)

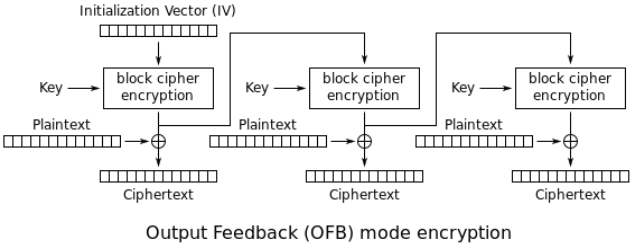
Mode ECB dan CBC harus dilakukan jika jumlah bit dalam satu blok sudah lengkap, sehingga proses akan menunggu terlebih dahulu hingga jumlah bit lengkap. Mode CFB berusaha mengatasi kelemahan tersebut dengan melakukan proses dalam unit yang lebih kecil daripada ukuran blok. Ukuran data yang diproses dapat berupa bit per bit, 2 bit, 3 bit, dan sebagainya. Penggunaan mode CFB ini memerlukan struktur antrian (queue) yang berukuran sama dengan ukuran blok masukan. Contoh penggunaan mode CFB 8-bit pada blok berukuran 64-bit dapat dilihat pada Gambar 4.



Gambar 4. Operasi block cipher dengan mode CFB

4. Output Feedback (OFB)

Pada mode OFB, operasi yang digunakan mirip dengan operasi pada mode CFB, namun n-bit dari hasil enkripsi terhadap antrian disalin menjadi elemen posisi paling kanan di antrian. Contoh penggunaan mode OFB 8-bit pada blok berukuran 64-bit dapat dilihat pada Gambar 5.

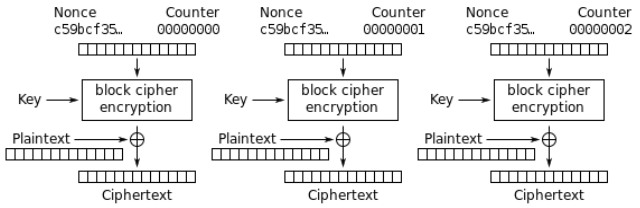


Gambar 5. Operasi block cipher dengan mode OFB.

5. Mode counter

Mode counter diusulkan oleh Diffie dan Hellman pada tahun 1979. Pada mode ini tidak dilakukan

proses perantaian (chaining) seperti pada mode CBC. Untuk melakukan proses enkripsi, digunakan counter berupa blok bit yang ukurannya sama dengan ukuran blok plainteks. Nilai awal counter tersebut harus berbeda dari pada setiap blok yang dienkripsi, kemudian nilai counter tersebut dinaikan nilainya satu persatu pada tiap proses enkripsi. Skema mode counter ini dapat dilihat pada Gambar 6.



Gambar 6. Operasi block cipher dengan mode counter.

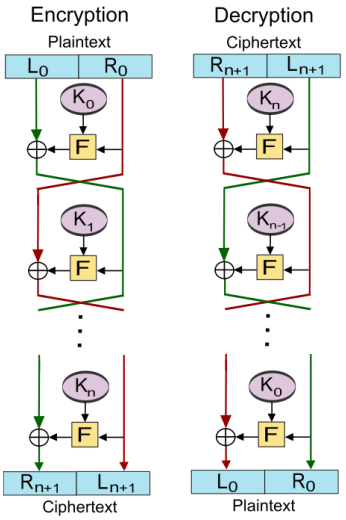
B. *Jaringan Feistel*

Jaringan Feistel mengimplementasikan serangkaian cipher iteratif pada blok data dan umumnya dirancang untuk cipher blok yang mengenkripsi data dalam jumlah besar. Jaringan Feistel bekerja dengan memecah blok data menjadi dua bagian yang sama dan menerapkan enkripsi dalam beberapa putaran. Setiap putaran mengimplementasikan permutasi dan kombinasi yang berasal dari fungsi utama atau kunci. Jumlah putaran bervariasi untuk setiap sandi yang mengimplementasikan jaringan Feistel. Feistel merupakan algoritma yang reversibel. Jaringan Feistel dapat dilihat pada Gambar 7.

C. *Properti Konfusi dan Difusi*

Pada kriptografi terdapat properti yang disebut prinsip *diffusion* dan *confusion,* kedua properti ini diperkenalkan oleh Claude Shannon pada tahun 1945. Kedua prinsip ini bertujuan untuk menggagalkan penggunaan statistik dan metode lainnya dalam kriptanalisis. Prinsip ini dipublikasikan dalam publikasinya yang berjudul *Communication Theory of Secrecy Systems*.

Shannon mendefinisikan confusion mengacu pada proses perancangan hubungan serumit mungkin antara cipherteks dan kunci simetri, sementara *diffusion* mengacu pada penyebaran pengaruh bit plainteks dan kunci seluas



Gambar 7. Feistel Network

mungkin pada *cipher text*. *Confusion* bertujuan agar kriptanalis frustasi dalam menemukan hubungan kunci dengan pesan. Sebagai contoh pada vigenere cipher, prinsip confusion berusaha diterapkan dengan membuat cipherteks yang berasal dari operasi penjumlahan karakter pesan dan karakter kunci. Diffusion bertujuan agar perubahan bit pada cipherteks menghasilkan hasil pesan di luar prediksi kriptanalis. Untuk mendapatkan cipher dengan keamanan yang tinggi, prinsip *diffusion* dan *confusion* diterapkan secara berulang dalam sebuah blok tunggal dengan kombinasi yang berbeda-beda. Metode paling sederhana untuk memenuhi prinsip *confusion* dan *diffusion* adalah menerapkan jaringan substitusi dan permutasi.

III. Rancangan Algoritma

Algoritma yang dibuat menerima *block cipher* sepanjang 256 bit atau 32 byte dan kunci dengan ukuran yang sama. Dipilih jumlah bit tersebut dikarenakan semakin banyak bit yang diproses oleh algoritma, maka akan semakin rumit proses kriptografi yang ada sehingga mempersulit serangan terhadap algoritma. Algoritma ini juga diulang sebanyak 20 kali untuk meningkatkan kompleksitas. Algoritma ini menerapkan prinsip *diffusion* dan *confusion* dengan memproses algoritma tersebut dengan permutasi dengan kotak permutasi dan substitusi dengan kotak substitusi pada fungsi *round*-nya. Fungsi *round* tersebut kemudian juga diproses dengan operasi bit agar semakin kompleks. Algoritma ini juga menerapkan jaringan Feistel dengan membagi kedua *plaintext* menjadi 128 bit di keduanya.

Selanjutnya akan dijelaskan detail dari rancangan algoritma block cipher Standard.

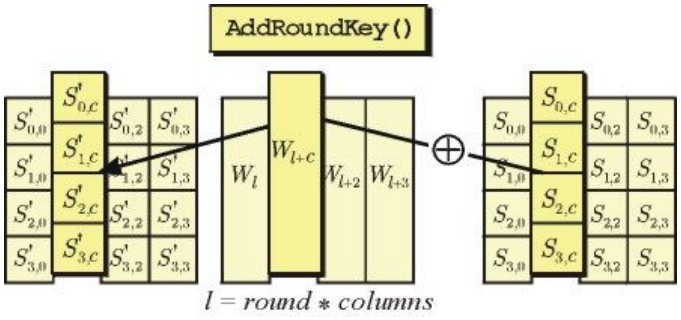
A. *Fungsi Ekspansi Kunci*

Algoritma ini menerima kunci sebanyak dengan jumlah blok yaitu 32 byte. Algoritma ini menggunakan fungsi ronde yang sama dengan DES yaitu (F(Mi, Ki) = Ci +1) untuk

membangkitkan kunci. F menerima 128 bit dari *significant bit* dari kunci dan 128 bit (K1) dari selain dari *significant bit* tersebut (K2) sehingga F menerima F(K1, K2). Dalam Algoritma ini menerima hingga (256 * (n + 128)) bit, dan kemudian setelah dibangkitkan lalu digabungkan hasil invers bit dari kuncinya dengan kunci aslinya menjadi kunci yang dipakai dalam fungsi ronde. Fungsi tersebut akan melakukan iterasi dari ronde 0 hingga selanjutnya.

B. Fungsi Pencampuran

Fungsi Pencampuran di dalam algoritma ini mencampur setengah *block cipher* berukuran 128 bit yang kemudian dilakukan operasi XOR dengan kunci yang berukuran sama 128 bit (Xi = Mi ⊕ Ki) seperti dengan AddRoundKey pada operasi DES. Fungsi Pencampuran dapat dilihat pada Gambar 8.



Gambar 8. Fungsi Pencampuran

C. Generasi *S-Box* dan Substitusi

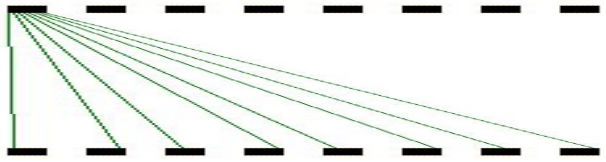
S-Box dalam *block cipher* ini berukuran 128 bit yang dibangkitkan secara acak menggunakan benih yaitu kunci dari *cipher* tersebut. Kemudian *S-Box* digunakan untuk proses substitusi dengan setengah kotak dari *cipher* yang ada. Contoh S-Box dapat dilihat pada Gambar 9.

S_1	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S_2	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S_3	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S_4	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

Gambar 9. Contoh Kotak-S

D. Generasi *P-Box* dan Permutasi

P-Box dalam *block cipher* ini berukuran 128 bit yang dibangkitkan secara acak menggunakan benih yaitu kunci dari *cipher* tersebut dengan rumus (Ci+1 = P(Zi)). *P-Box* kemudian digunakan untuk melakukan permutasi dari setengah *block cipher* berukuran 16 bit agar tersebar dari tiap 2 bit. Contoh Permutasi dengan Kotak-P dapat dilihat pada Gambar 10/



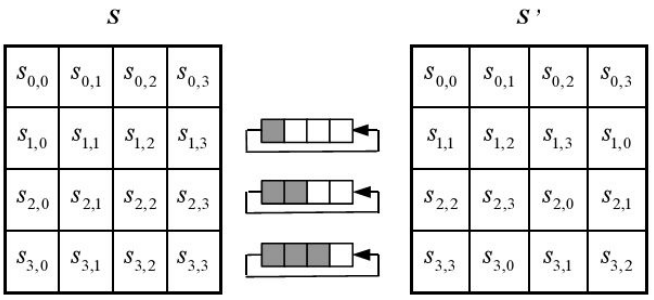
Gambar 10. Contoh Permutasi

E. Fungsi Transformasi

Round Function yang diterapkan terdapat empat proses yaitu rotasi kiri atau kanan dengan kunci, fungsi *shiftrow* ke kanan atau kiri, fungsi rotasi sirkuler dan terakhir fungsi *Discrete Cosine Transform* dan *Discrete Fourier Transform*.

Fungsi transformasi dimulai dengan melakukan transposisi dengan rotasi kiri atau kanan, bergantung terhadap token yaitu bit kunci ke N. N adalah bilangan acak yang dibangkitkan dari benih kunci. Jika N lebih besar daripada delapan, akan dilakukan rotasi kiri pada *block* tersebut dan sebaliknya. Fungsi rotasi dapat diilustrasikan sebagai berikut.

Kemudian dilanjutkan dengan melakukan transposisi dengan *shiftrow* kiri atau kanan, bergantung terhadap token yaitu bit kunci ke N. N adalah bilangan acak yang dibangkitkan dari benih kunci. Jika N lebih besar daripada delapan, akan dilakukan *shiftrow* kiri pada *block* tersebut dan sebaliknya. Fungsi *shiftrow* ini dilakukan sebanyak jumlah *counter* yaitu *less significant bit* dari byte dari key. Fungsi *shiftrow* dapat diilustrasikan sebagai Gambar 11.



Gambar 11. Contoh Shiftrows

Lalu, dilakukan rotasi sirkuler yaitu rotasi penuh yang berlawanan arah jarum jam sebanyak 4x4 sebanyak jumlah *counter* yaitu *less significant bit* dari byte dari key. Hal ini digunakan agar *block* semakin *confusing* dan semakin sulit untuk dilakukan serangan. Nilai *counter* dan *token* dari kunci akan membantu pula dalam diffusion, terlebih lagi kunci adalah seed dari nilai acak. Fungsi rotasi sirkuler dapat diilustrasikan sebagai berikut.

Terakhir dilakukan antara *Discrete Cosine Transform* dan *Discrete Fourier Transform*. Kedua fungsi transformasi itu akan mengeluarkan nilai *integer* baru yang diskrit. Selain itu

juga akan membantu prinsip *confusing* karena nilai yang dikeluarkan adalah setengah dari ukuran *block* yaitu 64 bit.

F. *Mode Operasi*

Block Cipher ini yang berukuran 256 bit yang dibuat dapat dioperasikan dalam mode berikut: ECB, CBC, CFB dan OFB.

IV. Eksperimen

Block cipher dengan Transformasi Fourier dan Cosine diimplementasikan dengan bahasa Python. Untuk melakukan eksperimen terhadap algoritma tersebut, dilakukan pengujian dengan 5 mode operasi block cipher; Electronic Code Book (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB). Data yang digunakan merupakan potongan teks pembukaan UUD 1945 sepanjang 185 karakter yang akan dibaca dari file test.txt. Kunci yang digunakan sepanjang 32 karakter adalah 'KriptografiMerupakanMatkulYgAsik' sedangkan *init vector* yang digunakan sepanjang 32 karakter, yaitu 'KamiBerhrapNilaiKriptografiKamiA'.

Berikut adalah kunci yang digunakan:

```
KriptografiMerupakanMatkulYgAsik
```

Berikut adalah vektor inisialisasi yang digunakan :

```
KamiBerhrapNilaiKriptografiKamiA
```

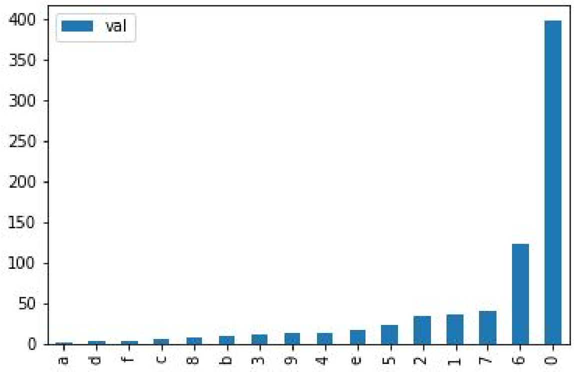
Berikut adalah data yang digunakan:

```
Bahwa sesungguhnya kemerdekaan itu ialah hak segala
bangsa dan oleh sebab itu, maka penjajahan diatas dunia
harus dihapuskan karena tidak sesuai dengan
perikemanusiaan dan perikeadilan.
```

Berikut adalah bentuk HEX dari data tersebut:

```
fffe42006100680077006100200073006500730075006e006
70067007500680006e007900610020006b0065006d0065007
200640065006b00610061006e002000690074007500200006
90061006c006100680020006800610006b002000730065006
70061006c0061002000620061006e006700730061002000 6
40061006e0020006f006c006500680020007300650062006
1006200200069007400075002c0020006d0061006b0061002
000700065006e006a0061006a006100680061006e0020006
4006900610074006100730020006400075006e00690061002
0006800610072007500730020006400690068006100700007
50073006b0061006e0020006b006100720065006e0061002
00074006900640061006b00200073006500730075006100 6
9002000640065006e00670061006e002000700065007200 6
9006b0065006d0061006e0075007300690061006100 6e002
000640061006e002000700065007200690006b00650061006
40069006c0061006e002e00
```

Sehingga jika hasil *cipher* HEX dilakukan plotting pada diagram dihasilkan grafik sesuai Gambar 12.



Gambar 12. Grafik Kemunculan Huruf pada Plaintext

A. Hasil Pengujian dengan ECB

Berikut adalah hasil enkripsi dengan mode ECB:

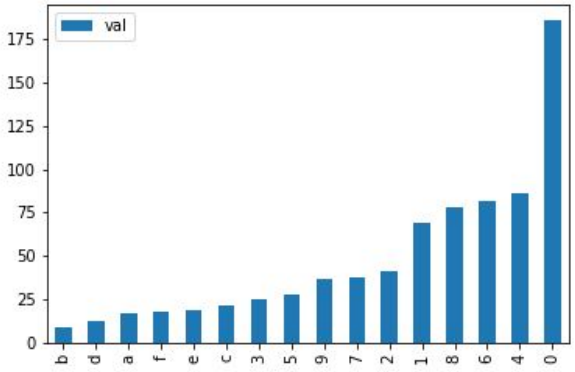
```
滿〱J堘□š貆G萠̀□⠇{□阴ӵǯ⠿□⁸蚕uñӵ聲档□□
□ű蹯4□□□□□ı驭À□«A□ǆ̄ġ□ӹ□aL僙)□□ïɔ̈鏡
楨□☺褠□̦□□ꩻ□□□聡G□潒□□a□ʊ¡鑭□□□
뿱□陭gȸ□□□咨Ġ□□□ɷぬ샵Ųɛ̀iv悜□栱蠵□
□Æ葵□□□聴□孲p贖桓c□□Ű□Ɵ遅□E堹□tɜÁ鑞
恋"□瑀□□ꭡ□¡ᵒ□帋□舤Î鐠檦ɕ⁖ô□Чмw□□儨□a
顮ã □□硾裢p□C葩□□□eŸɛT□□(倀肉□羁
```

Berikut adalah hasil enkripsi dalam HEX:

```
ff6e41204a006858eb406101608c47006584530954
187e287b0066403496f5046e04db2869107820fac8
75006812f504728064684508331060487101 6f8e3
4006910f70856183038644031016d9ac0006810a2
2141006918f6c021017608f904671061004c006150
29006241641cef006794d36948082030f188704126
08b8006f144d294f103040a840224161804700611
04261090869086100744020 0ca1006d94e3084918
390829c060406d9667006a0442496a102948f0c82
0012e1a08006414ca4948106c30f5c072016c0aec0
075044e606a0831683488684065 1cc6007584f021
210874106d8079416898700075805368630069406
64870016318e8047290e64845003958bd08740069
0cc10061944b6022007b1065802340780e6d04691
4a10045082d1072c837402482ce002094f1694410
6a28f4007a416502690461044e407d083b5078406
```

```
1006d98e3002000c74062087e78b5887000651a43
00698448496710210865007801 6c0a54006e148d4
828000050898000410 59e1100
```

Dari informasi yang didapat dari hasil *cipher* HEX, didapatkan kesamaan huruf yaitu 0,26%. Sehingga jika hasil *cipher* HEX dilakukan plotting pada diagram dihasilkan grafik sesuai Gambar 13.



Gambar 13. Grafik Kemunculan Huruf mode ECB

B. Hasil Pengujian dengan CBC

Berikut adalah hasil enkripsi dengan mode CBC:

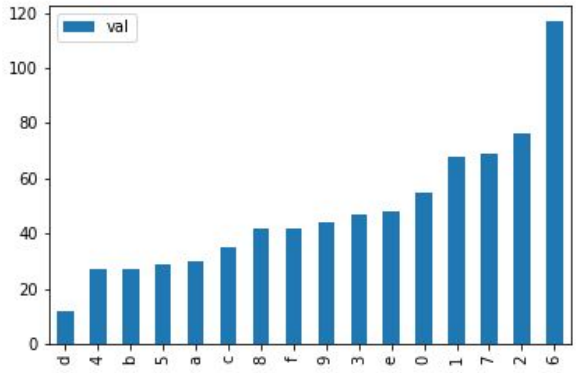
```
¢*gG!ânCifvád)&©1\ì<láÛ+og'Qfb4n&yjf~fX)^á
éx0>~h/
        ¢/N{bf§Xa1V©Xqnùlï Í
|o&Q«¢6ooãYfqfwx*9&&÷!yp"vÁlk3oQ"T/ybr78
1O
)UqCbÑhíàrg)¿¢lngëÌfzæuJ).nviüpmöÄl«B;CQÂb
.ibfX!>K¡
1Lê§hüYkw/q£b..'cIbzfÉpj)v1áå1fõÏhœ
uVI".iœfò¢Pb1
ÐapZò+lç¿#;g;Iñb,n+gçf;âñ-9/v))50w`æhr+Dw@iª
LoCq<bæ0f!Q!pSxšhï»jyt"|.gõHfuâº8f16iÄ0þh
```

Berikut adalah hasil enkripsi dalam HEX:

```
8f98a22a0667472101e2086e436993660c76e11864
291c261da998315cec3c6ce10cdb2b6f6727516662
346e26796a667e66865803297f5ee1e978303e7e99
68889c2f0a107f070903a20c2f4e7b03621666a758
6131165692a95871016ef96cef9ccd0b7c6f2651aba
2366f6fe35966716677782a392626f72179702276c
16c801c016b336f1e510f22542f0f791f6210723738
08314f0e0a2955714362d168ed1ce00272677f29bf
a26c6e67ebcc667ae675184a292e6e7669fc706df6c
```

```
46c8918ab423b7f4351c2620e2e036915620f669b5
801210f3e4ba10d314ceaa768fc88596b11772f71a3
622e2e276349627a66c9706a29761631e1e53166f0
cf688e08bd0a757f56498f22032e0e69bd661bf2a25
062310d16d0610e705af22b6ce718bf233b673b49f
1622c6e2b67e7663be2f1102d392f7629293530776
0e6688688722b4477406904aa4c6f43713c621be61
7306621511e9c2194705378a868ef88bb6a037f791
974227c2e67f5486675e2ba38663111361569c4301
7fe1068
```

Dari informasi yang didapat dari hasil *cipher* HEX, didapatkan kesamaan huruf yaitu 0,0%. Sehingga jika hasil *cipher* HEX dilakukan plotting pada dihasilkan grafik sesuai Gambar 14.



Gambar 14. Grafik Kemunculan Huruf mode CBC

C. Hasil Pengujian dengan CFB

Berikut adalah hasil enkripsi dengan mode CFB:

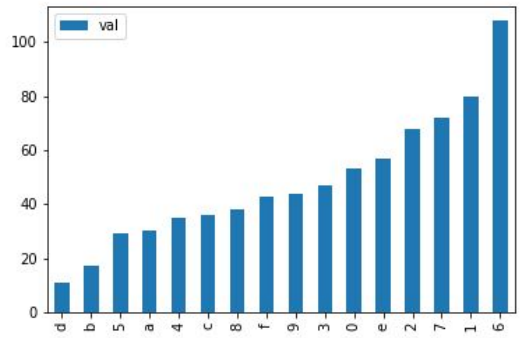
```
#.g_1ªPnFå·fòbN9\viQpæ¬láøgo7úâ}noñ{b~â')7Fé
)p17nl/*wWi
joáfv8I1~a1êDlïOBVg>3â~ocqdbqbup*)&¡10jp\h
Jg^b/o®fõP
)~iÕqàLlí@x')³*%n.{Lfzâ×8J1.v¡i1-pLh)"gQÎâ.Oá
-bæ#1C©0Dð®lüùk;o?I/j?//glbzfÊxH9~=at1.ö{hu
wFQb.CéfæxB!VT©qpŠlçŸ#wsyùb,./ùzf;vòX17v
%)œ17hvlQcNwPIbM.Jñfv7n)
F¡qàhïj)Aüj|nkeXbuæ»8f!A6iÌ1h!l
```

Berikut adalah hasil enkripsi dalam HEX:

```
8f9820232e675f3191aa506e46e5b7660cf262184e
395c761069517019e6ac6ce11cf802676f3711fae27
d6e6ff17b627ee2271801293746e9297031376e056
```

```
c889c2f2a107757690b6a1c6f07e19f66167684384
9311e7e9261103108ea446cef884f4256673e0133e
27e6f63716462716275702a2926067fa131306a705
c68800c834a19675e019f620c2f0e6fae6610f61650
0a290f7e1769d5710ee04c6ced184003787f2729b3
2a256e2e7b4c667ae2d7384a312e1676a1ec312d70
4c68891c292211670351cee20f2e4fe12d620fe61a1
02331071643a91d3044f0ae6cfc1cf96b3b6f3f492f
6a3f2f2f676c627a66ca7848397e163d6174312ef67
b688e0c1e02757746510362022e43e915661be680
784221055654a99e711270a66ce79cbe231b77737
9f9622c2e2ff97a663b76f258053137762529bd313
768766c869c51634e77504908624d2e4af191661b7
637186e29094614a18d711fe09968ef88196a017f2
941fc6a7c6e6b65586275e6bb38662141369c69cc3
11e68216c
```

Dari informasi yang didapat dari hasil *cipher* HEX, didapatkan kesamaan huruf yaitu 0,0%. Sehingga jika hasil *cipher* HEX dilakukan plotting pada dihasilkan grafik sesuai Gambar 15.



Gambar 15. Grafik Kemunculan Huruf mode CFB

D. Hasil Pengujian dengan OFB

Berikut adalah hasil enkripsi dengan mode OFB:

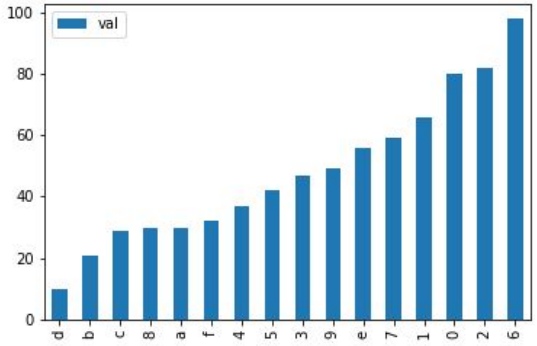
```
 #.g_1ªPnFå·fòbN9\viQpæ¬lâk_)     êEnOñ
brt8v1Q!0
vÌhæ6
0wyj      .s<br0s9naVpàlr¢J#g        Jâ
.Ne¢fâBP)RféVq
òPlf


g
ÊêC.ëbrRP;9~iBpMäðlòJ,VIRjnFk¢brð0{9
.áÇ0Olôlö"g[b
.NçfvÝ0x9f×!W1Lö#hbbGwK)¢/}brÂ8Z)SáÞpCn
žhb#
```

```
g^IV¢Uo})bbXP!niÌqn¥lòc)ga¢/JyfIfA|)>©0øhò+
0gQâPoï2bIötPX9T.R)ql3hâ*
w)ªY.GåbâŸ9*n~éfphþdh
```

Berikut adalah hasil enkripsi dalam HEX:

```
8f9820232e675f3191aa506e46e5b7660cf262184e
395c761069517019e6ac6c1ee2996b0c7f5f2909ea
456e4ff10d621b7274387631511e13218f300b76cc
6819e6360a30770f799b6a092e0e733c6201729030
7339136e9461567012e00c6c1772a24a236716094
ae20a2e4e65a2660ee242105029526683e956710bf
2506c1f660e0b0a671f09caea432e07eb936208725
2503b39037e836942704de4f06c1df2014a2c7f564
9526a186e466ba2620372f0307b390b2e13e1c7304
f6cf46c14f6882204675b110f620a2e4ee795661c76
dd3078390366d72157314cf623680562116247774
b2907a2012f077d81621c72c2385a2953161ce1de7
0436eb868026285230b675e4956a2556f037d2962
086258501b21196e9669cc711d6ea56c19f2630229
671a6110a20e2f4a791666496641107c29103e9fa9
1c3005f8846811f28e2b30670c5107e2506f0eef326
249f674505839542e52298e71046c336819e22a0b0
277162906aa592e47e5056207e2be101b392a6e7ee
9667068fe6468
```

Dari informasi yang didapat dari hasil *cipher* HEX, didapatkan kesamaan huruf yaitu 0,0%. Sehingga jika hasil *cipher* HEX dilakukan plotting pada dihasilkan grafik sesuai Gambar 16.



Gambar 16. Grafik Kemunculan Huruf mode OFB

V. Analisis Keamanan

A. *Analisis Frekuensi*

Jika dilihat dari hasil yang didapatkan dari grafik tiap mode, maka algoritma ini masih sangat lemah dalam menghadapi serangan terhadap analisis frekuensi. Pada mode ECB didapatkan bahwa *byte* 0 masih mendominasi terhadap

jumlah *byte* dalam representasi *cipher* HEX. Pada mode CBC, CFB dan OFB didapatkan bahwa *byte* 6 yang mendominasi terhadap jumlah *byte* dalam representasi *cipher* HEX.

Hal ini sangat buruk dikarenakan dapat dilakukan analisis frekuensi dikarenakan *byte* yang paling sering adalah 0. Namun, hal ini juga diakibatkan program yang melakukan padding nol agar dapat memadatkan byte yang tidak sempurna menjadi *block-block* 32 bit.

Sehingga dapat disimpulkan bahwa Algoritma STANDARD belum dapat menangani analisis frekuensi dengan baik.

B. *Bruteforce*

Ukuran *Block Cipher* yang digunakan dalam algoritma ini adalah 256 bit atau 32 byte. Ukuran Kunci yang digunakan juga sama yaitu 256 bit atau 32 byte. Sehingga secara matematika dibutuhkan 2^{256} atau 1.16×10^{77} percobaan.

Untuk analisis ini, digunakan panjang kunci, yaitu 256-bit, berarti terdapat $2256 = 1.16 \times 10^{77}$ kemungkinan kunci. Kemungkinan sebanyak itu apabila dilakukan *brute force* oleh super komputer tercepat saat ini saja, Sunway TaihuLight milik Tiongkok yang berkecepatan 93 petaflops (93×10^{15} operasi per detik), masih membutuhkan waktu sangat lama untuk memecahkan ARES secara brute force.

Perhitungannya adalah sebagai berikut:

t = jumlah operasi / kecepatan komputasi
t = 2256 /
t = 1.16×10^{77} / 93×10^{15}
t = 1.24×10^{60} detik = 3.92×10^{58} tahun

Hal ini membuktikan bahwa algoritma STANDARD sangat tangguh untuk menghadapi serangan *Bruteforce*. Jika menggunakan

VI. Kesimpulan dan Saran

Algoritma STANDARD adalah algoritma Block Cipher yang bekerja pada blok 256-bit dengan kunci 256-bit. Algoritma STANDARD menggunakan jaringan feistel dalam melakukan enkripsi dan menggunakan fungsi substitusi, permutasi, pencampuran, dan fungsi ronde. Di dalam fungsi ronde terdapat transformasi fourier dan cosine yang merupakan keunikan dari algoritma tersebut.

Algoritma ini dapat dijalankan dalam berbagai mode pengenkripsian. Dalam eksperimen didapatkan bahwa algoritma masih dapat diserang dengan frekuensi analisis, namun kuat terhadap serangan bruteforce.

Saran untuk selanjutnya adalah lebih melakukan operasi yang kompleks dikarenakan grafik yang dihasilkan dari hasil enkripsi masih belum sesuai dengan harapan.

REFERENSI

[1] http://www.wikiwand.com/en/Block_cipher_mode_of_operation

[2] Munir, Rinaldi, 2005. Diktat Kuliah Kriptografi. Bandung: Program Studi Teknik Informatika Institut Teknologi Bandung.

[3] Menezes, Alfred J, 2001, Handbook of Applied Cryptography (Fifth ed.), Waterloo: CRC Press. hal. 251

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, 13 Maret 2019

Muhammad Rizki Duwinanto (13515006)

Dandy Arif Rahman (13516086)