DK: Algoritma Cipher Blok Kombinasi Jaringan Feistel dan *Pseudorandom sub-Key*

Daniar Heri Kurniawan
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jl. Ganesha 10 Bandung 40132, Indonesia
daniar.h.k@gmail.com

***Abstrak*—Makalah ini membahas algoritma kriptografi kunci simetris DK yang berbasis block cipher. Algoritma ini menggunakan kunci dengan panjang minimal 8 karakter dan maksimal 32 karakter. Ukuran blok akan berubah secara dinamis menyesuaikan panjang kunci yang dimasukkan pengguna. Algoritma ini mengacu kepada prinsip diffusion dan confusion dari Shannon dengan mengimplementasikan operasi subtitusi, transposisi, dan iterasi yang dikombinasikan dengan struktur jaringan Feistel pada proses enkripsi dan dekripsinya.**

Kata Kunci—Block cipher; Jaringan Feistel; Pseudorandom sub-Key.

I. PENDAHULUAN

Perkembangan teknologi pada era modern ini menjadi kunci persebaran informasi ke seluruh penjuru dunia. Teknologi pengamanan informasi yang dulu hanya dilakukan melalui selembar kertas, saat ini kompleksitasnya sangat jauh berbeda. Kemampuan komputasi komputer yang semakin meningkat mendorong para peneliti untuk menciptakan algoritma enkripsi yang semakin aman. Keamanan sebuah algoritma dapat dilihat dari penerapan prinsip *diffusion* dan *confusion* yang di perkenalkan oleh Shannon. Saat ini penggunaan algoritma kriptografi tidak hanya oleh badan keamanan negara, namun juga diterapkan secara meluas pada proses pengiriman paket data melalui jaringan internet atau biasa dikenal *Secure Socket Layer*. Dalam praktiknya, sebuah algoritma kriptografi akan mengenkripsi sebuah *plaintext* menjadi *ciphertext* berdasarkan kunci tertentu.

Makalah ini membahas algoritma DK yang merupakan algoritma kriptografi simetrik yang dikembangkan oleh Daniar Heri Kurniawan. Nama DK diambil dari dua inisial nama pembuat, yaitu Daniar dan Kurniawan. Algoritma ini menerapkan teknik iterasi, subtitusi, dan transposisi yang dikombinasikan dengan struktur jaringan Feistel. Pengujian dilakukan dengan mengimplementasikan algoritma DK pada mode ECB, CBC, dan CFB 8-bit. Ukuran blok yang digunakan akan berubah secara dinamis sesuai panjang kunci yang dimasukkan oleh pengguna. Kunci terpendek yang boleh digunakan adalah sepanjang 8 karakter dan yang terpanjang adalah 32 karakter. Dalam algoritma ini juga diterapkan algoritma SHA-256 dam *pseudorandom* sub-Key yang di peroleh dari fungsi khusus yang akan dijelaskan pada Bab 3.

II. DASAR TEORI

A. *Prinsip Diffusion dan Confusion*

Dalam dunia kriptografi, prinsip *confusion* dan *diffusion* adalah dua hal terpenting untuk melakukan operasi enkripsi dan dekripsi yang aman. Kedua prinsip tersebut di perkenalkan oleh Claude Shannon pada bukunya "*A Mathematical Theory of Cryptography*". Dalam bukunya dijelaskan bahwa *ciphertext* yang dihasilkan dari algoritma kriptografi lemah akan mudah di pecahkan dengan teknik analisis statistik dalam ilmu *cryptanalysis.* Dengan diterapkannya prinsip *confusion* dan *diffusion*, kemungkinan keberhasilan serangan kriptoanalisis dengan teknik tersebut dapat diminimalkan. Selain itu, konsep ini juga sangat berguna untuk membuat fungsi hash yang aman.

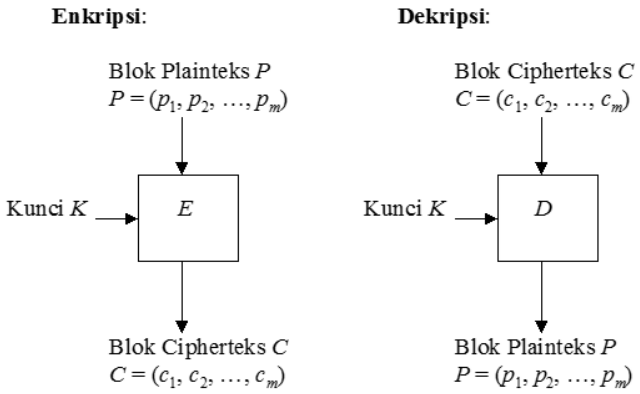
Confusion merupakan proses yang bertujuan untuk merubah data input secara drastis. Prinsip ini bisa diterapkan dengan membuat algoritma subtitusi yang sangat kompleks. Sedangkan *diffusion* adalah prinsip untuk menyebarkan pengaruh perubahan satu bit plainteks atau kunci ke sebanyak mungkin cipherteks. Pada algoritma DES, *diffusion* direalisasikan dengan menggunakan operasi permutasi. Secara umum, kedua prinsip ini bertujuan untuk menyamarkan keterhubungan atau *pattern* antara kunci, *plaintext*, dan *ciphertext.*

B. *Cipher Berulang (Iterated Cipher)*

Fungsi ini cukup sederhana untuk diimplementasikan, yaitu dengan mengulang proses enkripsi beberapa kali seperti pada *tripleDES.* Yang membedakan tiap perulangannya adalah adanya *sub-Key* atau upa-kunci yang bisa dibuat serumit mungkin agar hasil *ciphertext* berubah secara drastis dari *plaintext* awal.

C. *Mode ECB, CBC, dan CFB-8 bit*

Ketiga mode tersebut adalah jenis operasi yang dilakukan pada *cipher* blok. ECB (*Electronic Code Book)* adalah proses enkripsi yang dilakukan independen pada setiap blok. Pada CBC (*Cipher Block Chaining*), setiap blok cipherteks bergantung tidak hanya pada blok plainteksnya tetapi juga pada seluruh blok plainteks sebelumnya. Hal tersebut dikarenakan Hasil enkripsi blok sebelumnya di-umpan-balikkan ke dalam



Gambar 1. Skema enkripsi dan dekripsi pada *cipher* blok

enkripsi blok yang saat ini diproses. Sedangkan CFB (*Cipher Feedback*) diimplementasikan untuk mengatasi kelemahan pada mode CBC, karena data yang dienkripsi mungkin saja tidak sebesar blok yang di deklarasikan. Sehingga dalam mode ini, data dienkripsikan dalam unit yang lebih kecil daripada ukuran blok. CFB-8bit berarti ukuran data yang dienkripsi adalah setiap 8 bit atau satu *Byte*.

D. Fungsi SHA-256

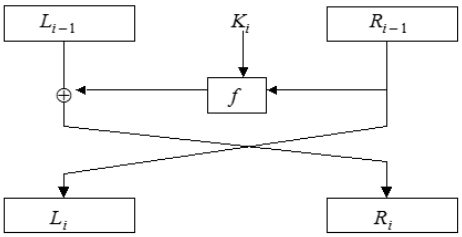
SHA-256 merupakan salah satu algoritma hashing yang telah distandardisasi oleh National Institute of Standards and Technology. Algoritma ini teruji cukup aman dan dapat diimplementasikan pada rangkaian byte data. SHA-256 menghasilkan string dengan panjang 256 bit. Hal ini menyebabkan algoritma hash SHA-256 lebih aman dari pada SHA-1. SHA-256 terdiri atas 64 kali putaran dan menggunakan 8 buah 32 bit register. Hal ini berbeda dengan SHA-1 yang menggunakan lima buah 32 bit register yang terdiri atas 80 kali puaran. Pemrosesan pesan dilakukan dalam blok 512 bit, sehingga data dibagi ke dalam blok-blok sepanjang 512 bit.

E. Fungsi Pseudorandom

Pembangkit bilangan acak secara matematis menggunakan fungsi matematis yang sekuensial, tiap bilangan baru merupakan fungsi deterministik dari bilangan-bilangan sebelumnya (rekuren). Basis dari fungsi rekursif tersebut disebut umpan (seed), yaitu satu atau lebih bilangan awal. Tidak ada fungsi matematis yang dapat menghasilkan deret bilangan acak yang sempurna/natural. Oleh karena itu pembangkit semacam itu disebut *pseudo-random number generator* (PRNG). Dalam makalah ini, PRNG yang digunakan adalah hasil modifikasi dari PRNG yang dimiliki oleh *library* bahasa Java versi 8 dengan menambahkan operasi bit untuk meningkatkan kerumitannya.

F. Jaringan Feistel

Feistel network atau jaringan feistel adalah adalah struktur simetris yang digunakan dalam mengkonstruksi block cipher. Jaringan feistel pertama kali ditemukan oleh Horst Feistel pada tahun 1970. Sejak saat itu jaringan feistel seringkali digunakan dalam banyak algoritma enkripsi, misalnya DES, GOST, Lucifer, Triple DES, dan masih banyak lagi. Jaringan Feistel memiliki sifat reversible yang membuat kita tidak perlu membuat algoritma baru untuk mendekripsi cipherteks menjadi plainteks. Sesuai dengan Gambar 2, sifat reversible tidak bergantung pada fungsi f sehingga fungsi f dapat dibuat serumit mungkin.



$$L_{i-1} \oplus f(R_{i-1}, K_i) \oplus f(R_{i-1}, K_i) = L_{i-1}$$

Gambar 2. Jaringan Feistel

III. RANCANGAN *BLOCK CIPHER*

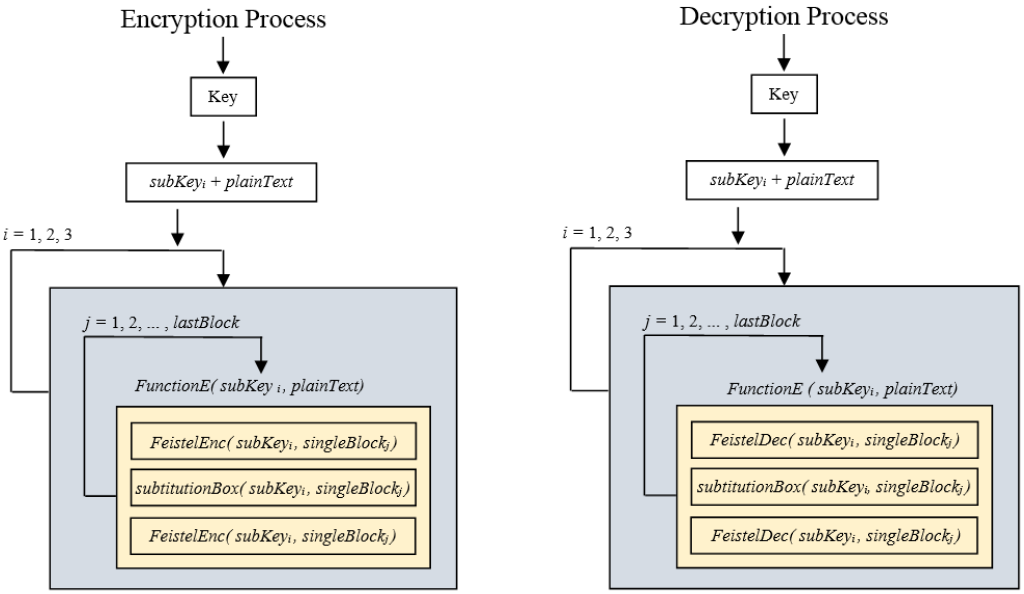
Prinsip konfusi Shannon dapat dicapai dengan menerapkan struktur jaringan Feistel. Dalam algoritma ini, struktur feistel diimplementasikan dengan mendefinisikan fungsi f sebagai algoritma SHA-256. Hasil dari fungsi hash kemudian akan dipotong sesuai panjang blok yang dikehendaki. Algoritma DK terbagi menjadi 3 proses utama, yaitu: perulangan pertama, perulangan kedua, dan pemrosesan pada fungsi E. Fungsi E sendiri terdiri dari 3 proses yang dilakukan secara sekuensial, yaitu: diproses pada jaringan Feistel, subtitusi dengan Sbox, dan diproses kembali pada jaringan Feistel.

Iterasi yang paling luar berjumlah 3 buah dengan masukan *subKey* dan *plainText.* Pada proses ini terjadi proses pembangkitan *subKey* dari key yang sudah ada. Proses ini akan di jelaskan pada bagian fungsi *pseudoRandom*. Kemudian pada setiap perulangan, data akan diproses per blok. Ukuran blok akan berubah secara dinamis sesuai panjang kunci yang diberikan pengguna. Panjang minimal yang diterima adalah 8 karakter dan panjang maksimalnya adalah 32 karakter. Panjang maksimal yang dipilih adalah 32 Byte (32 karakter) karena itu merupakan panjang hasil fungsi hash SHA-256 yang digunakan untuk membangkitkan upa-kunci

A. Pembangkitan upa-Kunci (subKey)

Upakunci di bangkitkan sebanyak 3 kali, masing-masing dilakukan dengan tahapan sebagai berikut:

1. Menginisialisasi sebuah array yang disebut arrayX sebanyak 255 buah dengan angka terurut 0-255.
2. Menghitung hash dari kunci asli dengan algortima SHA-256 yang akan menghasilkan 32 karakter.
3. Mengacak arrayX dengan fungsi pseudoRandom dengan seed berupa jumlah bit 1 pada hasil fungsi hash.
4. Memotong arrayX hasil pengacakan disesuaikan dengan panjang kunci.
5. Merubah representasi kunci dan arrayX tersebut menjadi array of Byte.



Gambar 3. Proses Enkripsi dan Dekripsi

6. Melakukan operasi XOR terhadap array of Byte dari kunci dan array of Byte dari arrayX yang teracak.
7. Langkah 1 - 6 diulang sesuai indeks *subKey* yang dibutuhkan. Pada indeks perulangan yang ke dua, maka akan digunakan $subKey_1$ sebagai kunci masukan pada langkah satu. Hal tersebut berlaku juga untuk indeks selanjutnya.

B. Implementasi Fungsi F pada Jaringan Feistel

Fungsi F di definisikan sebagai hasil fungsi hash dari upa-kunci yang di XOR kan dengan upa-kunci itu sendiri. Panjang dari fungsi SHA-256 akan selalu dipotong menyesuaikan dengan panjang blok yang sedang diproses.

C. Kotak Subtitusi

Dalam algoritma ini diimplementasikan kotakS untuk mendapatkan efek konfusi. Kotak subtitusi untuk proses enkripsi disusun dengan langkah-langkah sebagai berikut:

1. Menginisialisasi sebuah array yang disebut kotakS sebanyak 255 buah dengan angka terurut 0-255.
2. Menghitung hash dari kunci asli dengan algortima SHA-256 yang akan menghasilkan 32 karakter.
3. Mengacak kotakS dengan fungsi pseudoRandom dengan seed berupa jumlah bit 1 pada hasil fungsi hash.
4. KotakS kemudian di reverse atau dibalik urutannya.

Setelah kotakS terbentuk, maka karakter akan disubtitusi satu persatu dengan kotak array tersebut. Pada proses dekripsi, proses yang sejenis juga dilakukan, namun dengan urutan terbalik sebagaimana sehingga hasil kotak subtitusinya merupakan kebalikan dari kotakS pada proses enkripsi. Key yang digunakan untuk membangkitkan kotak ini juga selalu berupbah sesuai upa-kunci yang digunakan.

IV. EKSPERIMEN DAN PEMBAHASAN

DK diimplementasikan dalam bahasa Java versi 8. Untuk mengujinya, maka dibuatlah implementasinya dalam mode ECB, CBC, dan CFB. Algoritma ini akan dicoba pada berbagai kunci dan plainteks untuk menunjukkan bukti secara empiris dan teknis. Pada bagian akhir akan diberikan analisi terhadap hasil pengujian yang telah dilakukan

A. Hasil Enkripsi dan Dekripsi Plainteks Acak

Berikut adalah contoh hasil enkripsi tiga buah plainteks acak dengan kunci yang sama dalam metode CBC. Masing-masing sepanjang satu blok 32 Byte. Tiap data direpresentasikan per Byte. Text berikut dicopy dari text editor notepad.

Key : njhngporeojfeoasc,a'cpwokfpscs;l

Plain CBC:
sdv8453u9tj340tu34t49kfdnskvfndsklv
lsdfvsdkvn;vfdsvsdvsdvdsmlvmd

Cipher CBC:
º¡‡
¥¸?OÐn¾É¼òÃ°-É”æ%® ¢ ¾°Š¨
vÞ{¡£ vI žgãsªE,½tÆ□®ù®ñ•à1sŠwRèÔäÚ k…†¾;
ë©¦ó š ©Ÿö.{Üo Å8Šj*ƒ*

Plain CBC:
;dcmsavmlewnjvohegriowh358y-
924u5t09u345p'gm;'mlmckn43iqh8yh8y*&^T59'

Cipher CBC:
rw¨o/â|¿ ^¸2v
>œSÉÓ býRÆCkí¢³ÏÂv±sT-Ý@°Hbe] ¦±C5ý8`â© n—
òòÅÂ¿ÛÂõ:åÓ¡Z uÈá¸WÃZ C³ ˆñÅJ□×¬Ì□‹ a §

Plain CBC:
HGBVHBLKNBFV^T*hiVFDKNl8vfd*^&986984ugoinj
kvndflbkh93y9t2hinvlkdxs

Cipher CBC:
藿︿□□숮ｰ떾□恭盟琗木ᾶ棬ᡕ廣웃乬ﺩ□ᅡ雈뭉□ﯼ붒□ュ炨냀
젌鏁杙□ｨ□ᢵ弾縪Ꞓ□□揭□□❱軩ﻳ

Berikut adalah hasil dekripsi ketiga cipherteks yang diperoleh di atas dengan kunci yang sama.

Dekripsi mode CBC:
sdv8453u9tj340tu34t49kfdnskvfndsklv
lsdfvsdkvn;vfdsvsdvsdvdsmlvmd

Dekripsi mode CFB:
;dcmsavmlewnjvohegriowh358y-
924u5t09u345p'gm;'mlmckn43iqh8yh8y*&^T59'

Dekripsi mode ECB:
HGBVHBLKNBFV^T*hiVFDKNl8vfd*^&986984ugoinj
kvndflbkh93y9t2hinvlkdxs

Dapat dilihat bahwa plainteks yang dihasilkan pada proses dekripsi sama dengan plainteks awal. Jadi, algoritma tersebut telah mengenkripsi dan mendekripsi dengan benar.

B. *Hasil Enkripsi Plainteks atau Kunci yang Mirip*

Dalam bagian ini akan dilakukan enkripsi dua plainteks yang berbeda satu Byte. Akan dilihat perbedaan hasil enkripsi kedua plainteks tersebut dalam notasi heksadesimal.

Key : daniarhedaniarhedaniarhedaniarhe

Plain Text CBC:
Proposal of New Block Cipher Algorithm
Daniar Heri Kurniawan
Department of Informatics
School of Electrical Engineering and Informatics
Bandung, Indonesia
daniar.h.k@gmail.com

Cipher CBC:
9174 6b3d 0246 a852 8429 c2da a1bc 4c61
9535 a469 b8b2 7229 dcb7 d163 3a55 6c76
3865 2243 2aba 8689 aa1d 4f36 4534 faa6
1f7c 902a e39e 02a4 2488 7d88 1797 49b2
e33d 0968 481b 6780 9ab2 07d7 35a2 c0af
38b1 17f5 9b28 eb71 5acc e79f 7dab 7a6d
45ca e584 336c 4071 10c6 8675 520d 2a9f
8814 db51 0050 9328 2aa4 c7d1 3740 6481
5d56 f9d9 67fd 6bd4 00ad 0d3c 0388 8db8
60f4 3277 2a91 a7ce 1c1f a3d3 0c90 5a1b
fcc3 56be 55b9 cebb 66ba 0e89 8fa5 8334
b27a 1de5 ed94 4146 1c78 cf8e 4c36 043b

Plain Text CBC:
Proposal of New Block Cipher Algorithm
Danial Heri Kurniawan
Department of Informatics
School of Electrical Engineering and Informatics
Bandung, Indonesia
daniar.h.k@gmail.com

Cipher CBC:
fcc0 cdc5 c28c f8c9 d4d8 8cef eeef 96a6
1a38 253a 2539 2b26 6a25 2c6a 042f 3d6a
0826 2529 216a 0923 3a22 2f38 6a0b 262d
3528 332e 3237 501e 3b34 333b 287a 123f
2833 7a11 2f28 3433 3b2d 3b34 501e 3f2a
0d1e 1801 0902 184c 030a 4c25 020a 031e
010d 1805 0f1f 663f 0f04 0303 004c 030a
7015 3c35 3324 2239 3331 3c70 153e 3739
3e35 3522 393e 3770 313e 3470 193e 363f
e2fd f1e4 f9f3 e39a d2f1 fef4 e5fe f7bc
b0d9 fef4 fffe f5e3 f9f1 9af4 f1fe f9f1
8ad6 90d6 93b8 9f95 9991 94d6 9b97 95f2

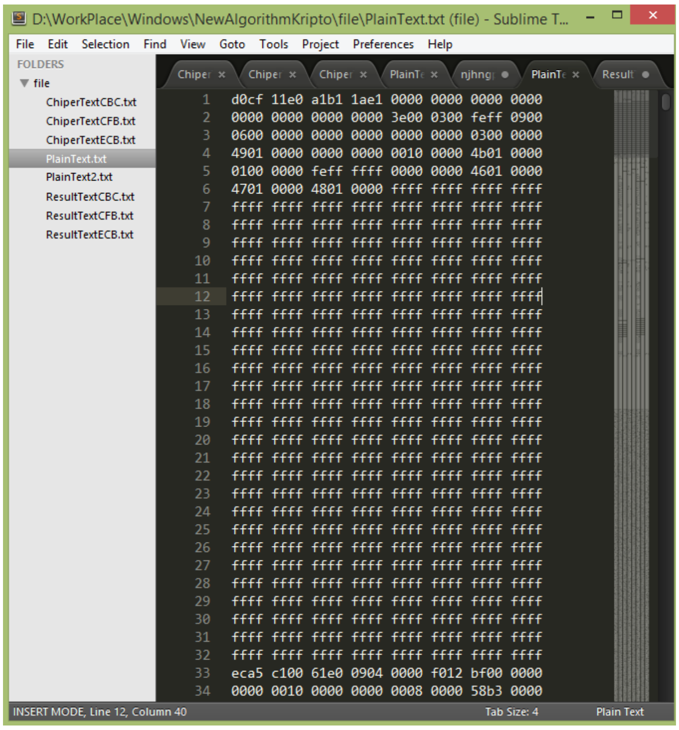
Plain Text CBC:
Proposal of New Block Cipher Algorithm
Danian Heri Kurniawan
Department of Informatics
School of Electrical Engineering and Informatics
Bandung, Indonesia
daniar.h.k@gmail.com

Cipher CBC:
045f 086c ec1c a31e 78ff 0969 7391 600a
78ef 6027 7c5b a07d 289c 67ea 77a5 ed1e
f200 8d0b ddeb bc27 12bb d4e0 8d83 8da6
e65e 4235 66cf 705c 8123 18c9 8edb 6454
9ed3 3d8f dbbb 8a85 2aeb 1721 e4ae 2c62
a149 5d3e 898a 08bf 7278 c281 9f08 fae0
eed6 d276 6b27 d32c eafa dba2 d272 602a
ac99 e88f 7f0f d0dd c47a 3384 8ec0 ae43
e185 a16e af5e c046 1992 aa01 11bc 3af4
b474 8757 5676 abff 9429 d4a1 b5e8 da9a
c718 f175 a005 8e6d 1c5b 0799 0248 df0a
f444 0c30 cf18 c046 9c30 4dc2 85d6 df98

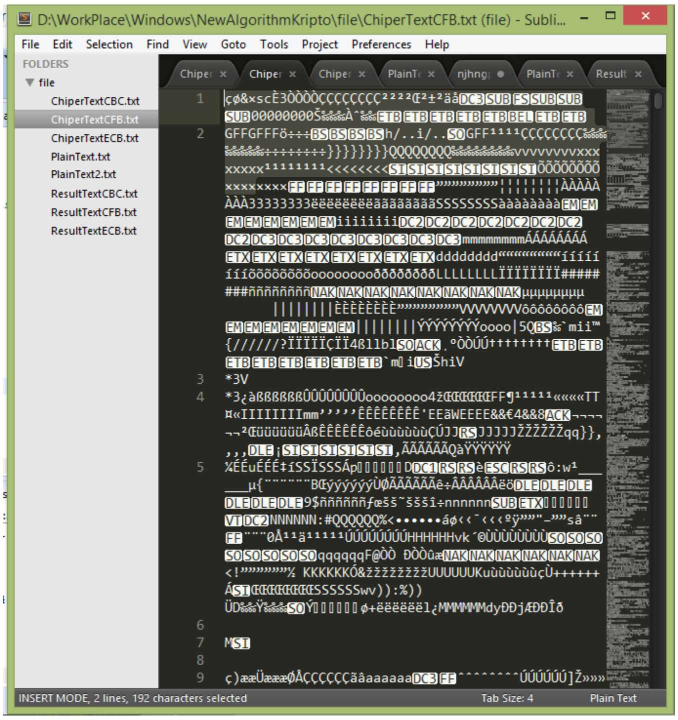
Sama seperti sebelumnya, kedua cipherteks yang dihasilkan berbeda jauh. Maka, prinsip konfusi telah dicapai oleh algoritma ini.

C. *Enkripsi dan Dekripsi Mode ECB, CBC, dan CFB*

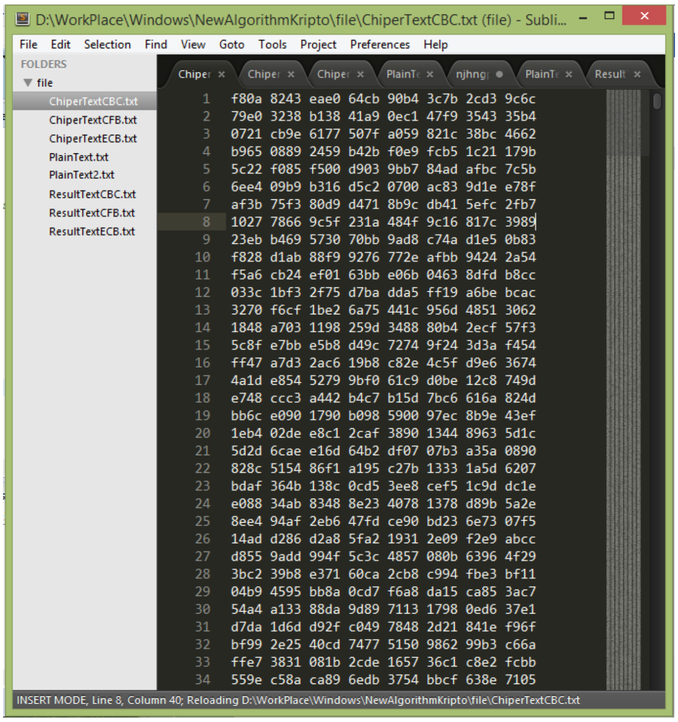
Dalam bagian ini dilakukan enkripsi dan dekripsi berkas dengan format tertentu dengan ketiga mode cipher blok. Kunci yang digunakan sama untuk ketiganya, yaitu “daniarhe”. Ukuran byte yang akan digunakan adalah 8 Byte karena panjang kuncinya 8 karakter. File yang akan di enkripsi adalah template makalah IEEE, “IEEE Paper Template.doc”. Berikurt adalah screenshot dari cipher teks dan plain teks yang dihasilkan. Hasil dari proses dekripsi berhasil mengembalikan dokumen seperti plainteks awal.



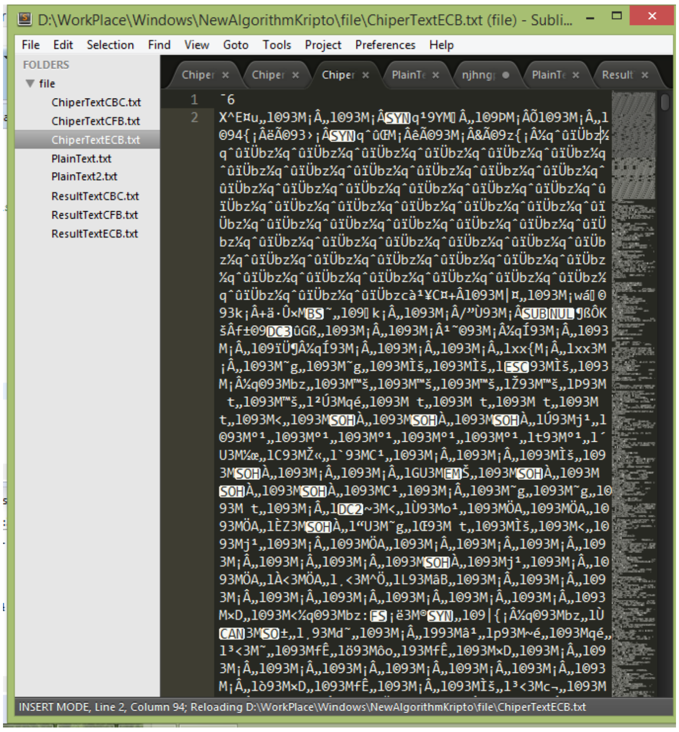
Gambar 4. Teks Asli yang dibuka dengan sublime



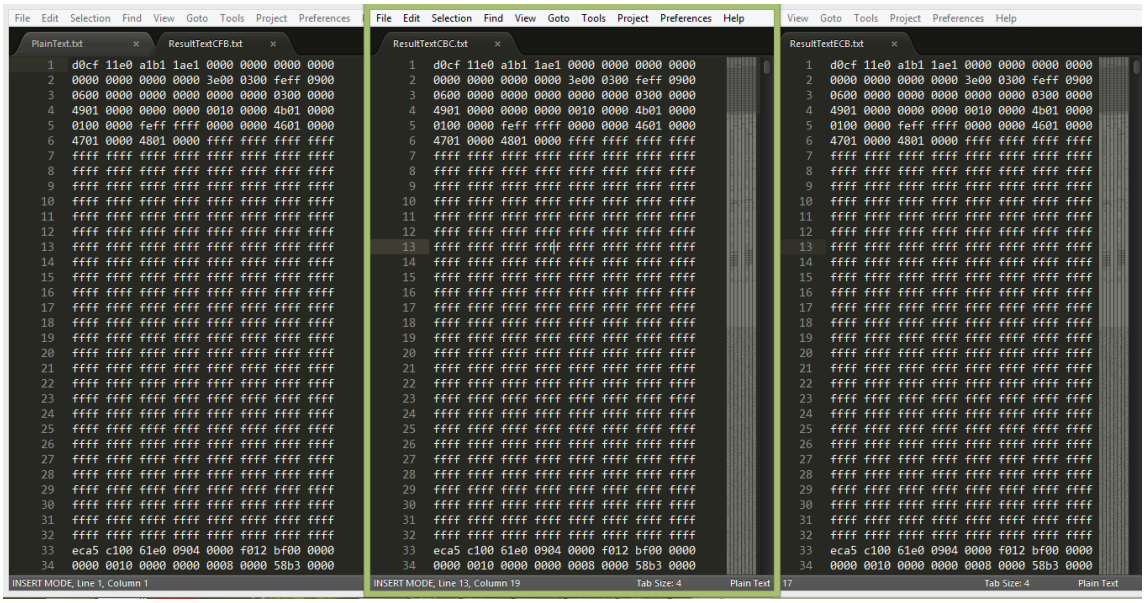
Gambar 5. Cipher text CFB



Gambar 6. Cipher text CBC



Gambar 7. Cipher text ECB



Gambar 8. Plainteks yang dihasilkan

Hasil dekripsi ketiga cipherteks tersebut berhasil mengembalikan dokumen awal. Dapat dilihat bahwa secara sekilas, hasil enkripsi sama sekali tidak menggambarkan plainteks atau kunci yang dipakai.

V. Analisis Keamanan

Lamanya pencarian kunci dengan *brute force attack* sebagian besar tergantung oleh panjang kunci. Karena panjang kunci maksimal ditetapkan sepanjang 32 Byte atau 256 bit, maka ada sebanyak 2^{256} buah kunci berbeda, atau sebanyak 1.1579×10^{77} kunci berbeda. Semakin panjang kunci, maka akan semakin aman algoritma ini.

VI. Kesimpulan dan Saran

Pengujian dilakukan dengan mengimplementasikan algoritma DK pada mode ECB, CBC, dan CFB 8-bit. Ukuran blok yang digunakan akan berubah secara dinamis sesuai panjang kunci yang dimasukkan oleh pengguna. Kunci terpendek yang boleh digunakan adalah sepanjang 8 karakter dan yang terpanjang adalah 32 karakter. Berdasarkan hasil analisa pengujian dan analisa keamanan, Algoritma DK mampu memenuhi prinsip diffusion dan confusion Shanon. Algoritma ini juga dapat diimplementasikan pada mode ECB, CBC, maupun CFB.

Referensi

[1] Xuejia Lai, “On the design and security of block ciphers”, Zurich:Swiss Federal Institute of Technology, 1992

[2] Claude E. Shannon, "Communication theory of secrecy systems", Bell System Technical Journal, vol. 28-4, page 656–715, 1949