Algoritma Blok Cipher OE-CK

Ali Akbar
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Bandung, Jawa Barat
13514080@std.stei.itb.ac.id

Aditio Pangestu
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Bandung, Jawa Barat
13514030@std.stei.itb.ac.id

***Abstrak*—Di era perkembangan teknologi informasi saat ini, enkripsi pesan sangat dibutuhkan untuk menjaga keamanan dalam berbagi informasi antar komunikan. Enkripsi pesan harus dibuat serumit mungkin agar sulit untuk dipecahkan oleh penyerang. Oleh karena itu pada makalah ini penulis mengusulkan sebuah algoritma blok cipher baru bernama OE-CK (*Odd-Even Confusing Key*) yang bekerja pada blok 128-bit dengan kunci 128-bit. Algoritma ini menggunakan struktur feistel dengan memanfaatkan operasi substitusi, pergeseran dan pengacakan pada fungsi f. Untuk meningkatkan kerumitan, fungsi ekspansi kunci juga memanfaatkan aturan bit ganjil genap dan pengacakan. Algoritma ini dirancang sedemikian rupa untuk memenuhi prinsip *confusion* dan *diffusion*.**

Keywords—Enkripsi, Block Cipher, Confusion, Diffusion

I. PENDAHULUAN

Dari sejak dulu, karena kepentingan akan kerahasiaan informasi, bermunculan banyak ide dalam melakukan enkripsi pesan (yang sering dikenal dengan algoritma kriptografi klasik). Algoritma kriptografi klasik kebanyakan hanya memanfaatkan prinsip substitusi dan transposisi sehingga dapat dengan mudah dipecahkan. Ditambah dengan adanya penemuan komputer, memecahkan pesan yang dienkripsi dengan algoritma kriptografi klasik akan jauh lebih mudah. Maka dari itu bermunculan algoritma-algoritma yang bekerja pada komputer secara bit-bit atau disebut juga dengan algoritma kriptografi modern.

Algoritma kriptografi modern jauh lebih sulit dipecahkan karena banyak kombinasi yang dihasilkan dari urutan bit-bit yang jauh lebih besar dibanding algoritma klasik. Namun seiring perkembangan zaman, kemampuan komputer terus meningkat dan semakin canggih. Bahkan algoritma sekuat DES (*Data Encryption Standard*) dapat dipecahkan oleh pemrosesan banyak komputer dalam beberapa tahun setelah perancangan algoritma tersebut. Oleh karena itu banyak dikembangkan algoritma-algoritma kriptografi modern yang sangat rumit.

Pada makalah ini penulis akan memberikan hasil analisis dan implementasi sebuah algoritma blok cipher baru yang bernama OE-CK (*Odd Even Confusing Key*). Algoritma ini menggunakan struktur feistel yang bekerja pada kunci sepanjang 128-bit dan blok sepanjang 128-bit. Sesuai dengan namanya, algoritma ini memanfaatkan posisi ganjil genap pada bit-bit kunci serta pengacakan dengan kotak P untuk meningkatkan *confusion*. Pada fungsi f pada struktur feitselnya, dilakukan proses substitusi, penggeseran dan pengacakan. Substitusi dilakukan dengan kotak S dan pengacakan dilakukan dengan kotak P yang berbeda dengan kotak P pada kunci. Untuk meningkatkan *confusion*, proses *enciphering* juga diulang hingga 10 kali pengulangan.

II. DASAR TEORI

A. Block Cipher

Block Cipher adalah algoritma deterministik yang bekerja pada kelompok bit pesan dengan panjang yang tetap, disebut juga dengan *block*, dengan transformasi-transformasi yang menggunakan kunci simetri. *Block cipher* digunakan sebagai komponen dasar dari berbagai protokol kriptografi pada data yang besar.

Algoritma *block cipher* terdiri dari 2 pasang algoritma yaitu algoritma enkripsi dan algoritma dekripsi. Kedua algoritma ini menerima 2 masukan yaitu kunci sepanjang k-bit dan pesan sepanjang n-bit sesuai dengan spesifikasi *block cipher*. Fungsi enkripsi memetakan pesan menjadi cipherteks dan fungsi dekripsi memetakan cipherteks menjadi pesan yang sebenarnya. Fungsi enkripsi yang dinotasikan dengan E melakukan operasi antara kunci dan blok pesan sehingga dihasilkan blok cipherteks sepanjang n-bit. Fungsi dekripsi yang dinotasikan dengan D adalah invers dari fungsi $E(E^{-1})$ yang juga memetakan kunci dan blok cipher ke blok plainteks yang sebenarnya sepanjang n-bit. Secara matematis, definisi fungsi enkripsi dan dekripsi dapat dinotasikan sebagai berikut.

$$E_k(P) := E(K,P) : \{0,1\}^k \; x \; \{0,1\}^n \rightarrow \{0,1\}^n$$

$$E_k^{-1}(C) = D_k(C) := D(K,C) : \{0,1\}^k \; x \; \{0,1\}^n \rightarrow \{0,1\}^n$$

$$D_k\big(E_k(P)\big) = P$$

Terdapat beberapa desain dalam membuat algoritma *block cipher*. Diantaranya,

1. *Iterated Block Cipher*, desain ini memetakan blok pesan ke cipherteks melalui proses berulang dengan sebuah fungsi *invertible* bernama fungsi *round*.

Enkripsi dilakukan dengan fungsi F sementara dekripsi dilakukan dengan fungsi F invers.

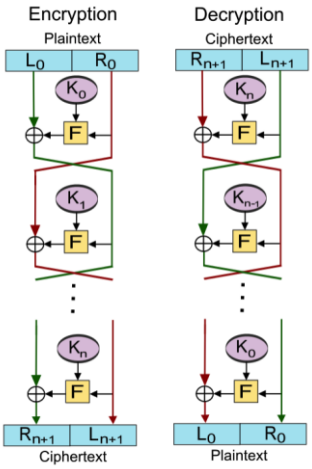
2. *Substitution Permutation Network*, desain ini memetakan pesan ke cipherteks dengan serangkaian proses substitusi menggunakan kotak S (*S-Box*) dan kotak P (*P-Box*) (sehingga membentuk jaringan) dengan penggabungan kunci dan blok menggunakan operasi XOR. Proses dekripsi dengan struktur ini dilakukan dengan *reverse order* proses enkripsi.
3. *Feistel Structure*, desain ini memetakan pesan ke cipherteks dengan membagi pesan menjadi 2 bagian sama besar. Untuk separuh bagian tersebut akan dilakukan transformasi bersama dengan *subkey* melalui sebuah fungsi *round*. Hasil dari fungsi akan di xor ke bagian lainnya yang tidak dikenai operasi fungsi. Operasi tersebut dilakukan dalam serangkaian pengulangan sesuai desain yang dibuat kriptografer. Secara matematis, algoritma enkripsi *feistel structure* dapat dinotasikan sebagai berikut.

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

Algoritma dekripsi dengan struktur feistel hanya kebalikan dari enkripsi.

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

Skema enkripsi dan dekripsi struktur feistel dapat dilihat pada gambar 1.



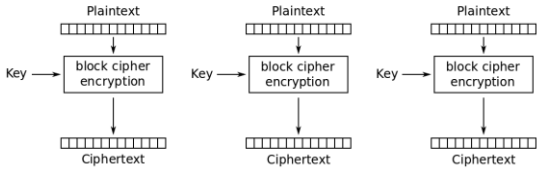
Gambar 1 Skema Enkripsi dan Dekripsi Struktur Feitsel

Karena keunggulan struktur feistel yang tidak mengharuskan fungsi *round* yang *invertible*, dapat didesain *round function* yang serumit mungkin. Fungsi *round* dapat mencakup operasi substitusi, permutasi, pergeseran, penjumlahan modulo, dsb. Maka dari itu akan lebih mudah untuk meningkatkan kerumitan algoritma kriptografi dengan menggunakan struktur feistel.

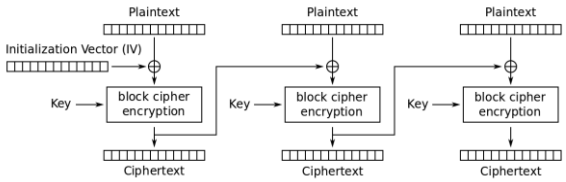
Block cipher dapat dijalankan dalam beberapa mode yaitu mode *Electronic Code Book* (ECB), *Cipher Block Chaining* (CBC), *Cipher Feedback* (CFB), *Output Feedback* (OFB) dan *Counter* (CTR). Adapun penjelasan pada tiap mode adalah sebagai berikut,

1. Pada mode ECB, block P_i dienkripsi secara individual dan independen menjadi sebuah blok cipher C_i melalui algoritma *block cipher* yang didesain. Skema algoritma ini dapat dilihat pada gambar 2. Mode ini memiliki kelemahan yaitu blok pesan yang dienkripsi selalu dipetakan ke blok cipher yang sama sehingga dapat diserang dengan analisis statistik.
2. Mode CBC berusaha untuk mengantisipasi kekurangan mode ECB dengan membuat ketergantungan antar blok dalam proses enkripsi. Blok pesan pertama dikenai dengan operasi xor dengan sebuah vektor sebesar panjang blok sebelum masuk ke dalam algoritma *block cipher*. Hasil dari algoritma *block cipher* akan dipakai sebagai vektor untuk operasi xor pada blok selanjutnya. Skema mode CBC dapat dilihat pada gambar 3. Kekurangan dari mode ini adalah blok selanjutnya baru dapat diproses apabila blok sebelumnya sudah diproses sehingga tidak dapat dijalankan secara parallel. Selain itu mode CBC dan ECB mengharuskan *padding* pada blok pesan apabila pesan tidak sepanjang blok. Akibatnya 2 mode tersebut tidak dapat bekerja untuk data komunikasi yang belum lengkap.
3. Mode CFB berusaha mengantisipasi masalah CBC dan ECB terhadap data yang belum lengkap. Maka dari itu CFB memodifikasi ECB dengan melakukan pendekatan cipher alir yang mana menggunakan sebuah vektor random sebagai input awalan dan menjadikan blok *plaintext* sebagai masukan operasi xor di akhir. Hasil dari operasi xor ini digunakan sebagai input pada operasi blok selanjutnya. Skema mode CFB dapat dilihat pada gambar 4. Kekurangan dari mode CFB adalah mode CFB secara skematik adalah *self-synchronizing* sehingga semua proses harus menunggu proses sebelumnya. Akibatnya tidak dapat diparalelkan
4. Mode OFB berusaha mengantisipasi masalah CFB dengan membuat skema *synchronous stream cipher*. Mode OFB hanya memodifikasi sedikit dari mode CFB yang mana blok yang akan diproses selanjutnya adalah blok hasil algoritma *block cipher*. Pada akhir setiap proses dilakukan operasi xor pada blok plainteks. Karena operasi xor selalu di akhir dan dilakukan secara independen. Proses ini dapat diparalelkan. Skema mode OFB dapat dilihat pada gambar 5.
5. Mode CFB, OFB dan CBC memiliki kekurangan yaitu pemaralalelan tidak dapat dilakukan pada proses *enciphering* dengan *block cipher*. Maka dari itu dibuat mode CTR yang melakukan pendekatan *stream cipher* dalam skemanya. Tiap proses blok akan menerima sebuah vektor masukan unik yang merupakan nilai

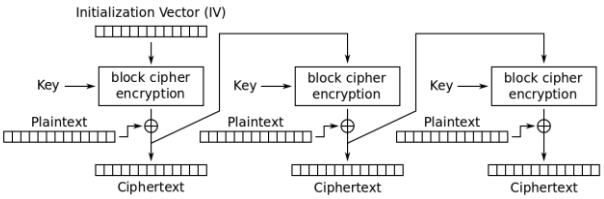
selanjutnya dari vektor masukan pada proses sebelumnya. Vektor ini dapat dibentuk dengan *loss-less operation* dari sebuah vektor acak terhadap bilangan hasil fungsi berurut. Skema dari CTR dapat dilihat pada gambar 6.



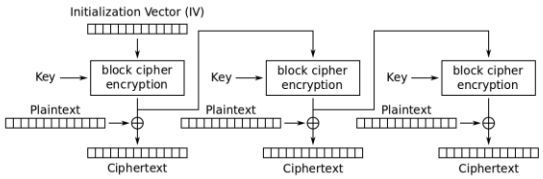
Gambar 2 Skema Enkripsi pada ECB



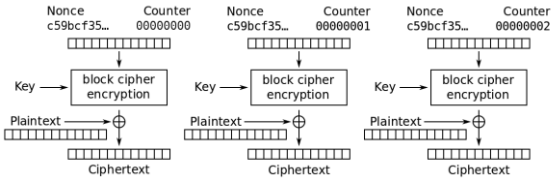
Gambar 3 Skema Enkripsi CBC



Gambar 4 Skema Enkripsi CFB



Gambar 5 Skema Enkripsi OFB



Gambar 6 Skema Enkripsi CTR

B. *Shannon's Confusion dan Diffusion*

Dalam kriptografi prinsip *diffusion* dan *confusion* adalah 2 properti dari *secure cipher* yang diperkenalkan oleh Claude Shannon pada tahun 1945. Kedua prinsip ini bertujuan untuk menggagalkan penggunakan statistik dan metode lainnya dalam kriptaanalisis. Prinsip ini dipublikasikan dalam publikasinya yang berjudul *Communication Theory of Secrecy Systems*.

Shannon mendefinisikan *confusion* mengacu pada proses perancangan hubungan serumit mungkin antara *ciphertext* dan kunci simetri sementara *diffusion* mengacu pada penyebaran pengaruh bit plainteks dan kunci seluas mungkin pada cipherteks. *Confusion* bertujuan agar kriptanalis frustasi dalam menemukan hubungan kunci dengan pesan. Sebagai contoh pada *vigenere cipher*, prinsip *confusion* berusaha diterapkan dengan membuat *ciphertext* yang berasal dari operasi penjumlahan karakter pesan dan karakter kunci. *Diffusion* bertujuan agar perubahan bit pada *ciphertext* menimbulkan 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 BLOK CIPHER

Berdasarkan analisis terhadap algoritma-algoritma kriptografi modern yang banyak berkembang. Kunci yang panjang akan menghasilkan cipherteks yang lebih aman namun secara komputasi akan lebih mahal. Maka dari itu algoritma OE-CK bekerja pada blok cipher sepanjang 128-bit dan kunci 128-bit serta memanfaatkan operasi-operasi pada bit agar komputasi menjadi lebih murah. OE-CK memanfaatkan jaringan feistel dan pada fungsi *round*-nya digunakan operasi-operasi bit yang murah yaitu substitusi, permutasi, pergeseran dan xor. Untuk meningkatkan kerumitan, OE-CK juga menggunakan fungsi ekspansi kunci yang khusus agar *confusion* antara *subkey* dan blok semakin tinggi. *Subkey* yang dihasilkan berasal dari indeks iterasi serta posisi bit ganjil genap pada kunci. Jaringan feistel dari algoritma ini dapat dilihat pada gambar 7. OE-CK menggunakan kombinasi substitusi dan permutasi untuk meningkatkan *confusion*. Terdapat 2 kotak P yang pertama untuk mengacak posisi kunci, dan yang kedua untuk mengacak posisi bit-bit pada pesan. Lalu terdapat 1 kotak S untuk substitusi elemen pada blok di fungsi f.

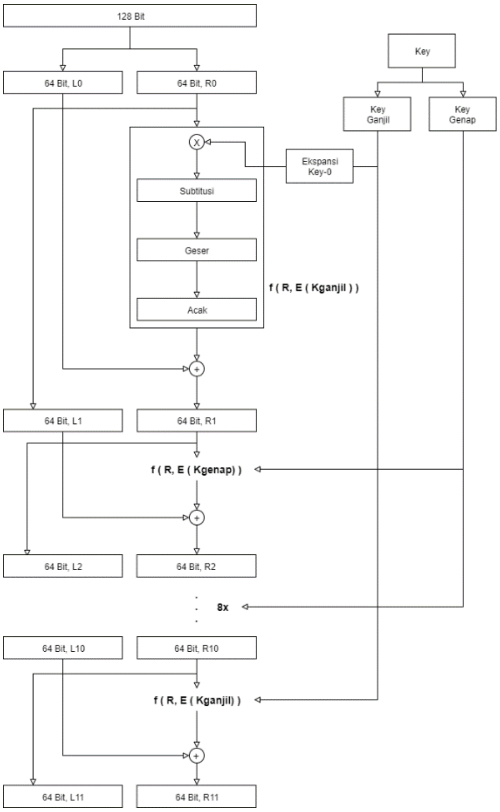
Selanjutnya akan dijelaskan detil dari rancangan algoritma *block cipher* OE-CK.

A. *Fungsi Ekspansi Kunci*

Algoritma OE-CK memiliki 10 kali pengulangan sehingga akan dibentuk 10 *subkey* untuk tiap iterasi. Untuk tiap iterasi, dilakukan proses pembangunan *subkey*. Dalam membangun *subkey* dilakukan pengacakan dengan kotak P untuk kunci. :

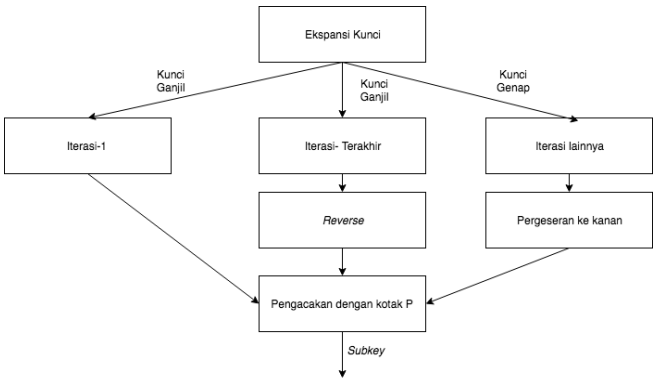
1. Bagi kunci menjadi 2 bagian (masing-masing sepanjang 64 bit). Yaitu kunci genap yang berisi bit-bit posisi genap pada kunci dan kunci ganjil yang berisi bit-bit posisi ganjil.
2. Untuk proses *enchipering* pada iterasi pertama, dibangun *subkey* dari pengacakan kunci ganjil dengan menggunakan kotak P kunci.
3. Untuk proses *enciphering* pada iterasi terakhir. Dilakukan proses pembalikan (*reversing*) terhadap kunci ganjil dan pengacakan menggunakan kotak P kunci.

4. Untuk proses *enciphering* pada iterasi lainnya (2-9). Dilakukan pergeseran ke kanan 1 bit pada kunci genap lalu dilakukan pengacakan dengan kotak P kunci.



Gambar 7 Struktur Feistel untuk algoritma OE-CK

Proses pengacakan, pergeseran serta membagi kunci menjadi 2 bagian dimaksud untuk meningkatkan *confusion* dan *diffusion* dari OE-CK. Proses ekspansi kunci juga digambarkan pada gambar 8.



Gambar 8 Skema Ekspansi Kunci OE-CK

Kotak P untuk mengacak posisi bit pada kunci dihasilkan melalui proses pengacakan *pseudo-random* dari angka 0-63. Berikut adalah kotak P kunci yang digunakan untuk fungsi ekspansi kunci OE-CK.

12	6	46	11	14	50	22	33
20	24	19	34	63	25	9	31
29	13	54	7	47	28	32	3
52	53	61	40	26	41	2	43
35	21	39	48	27	44	45	42
62	55	10	36	49	58	15	5
37	59	38	60	0	8	16	4
1	17	23	18	30	56	57	51

Tabel 1 Kotak P Kunci

0110000101100001011000010110000101100010011000100100010011000100110001101100011011000110110001101100011011000110110001000110010001100100

1001100110011001100010001000100010011001100110010110101010101010
0100010001000100010101010101010101010101010101010100010001000100

Gambar 9 Proses pengubahan dari kunci (paling pertama) ke kunci genap (atas kedua) ganji (bawah kedua)l

B. *Fungsi Round*

Fungsi f pada jaringan feistel algoritma OE-CK menerima blok sepanjang 64 bit dan melakukan operasi dengan bit-bit. Blok pertama kali diubah ke bentuk matriks 4x4 (yang berarti 1 sel berisi 4 bit pesan) untuk dioperasikan selanjutnya. Pembentukan matriks dilakukan dengan cara yang sederhana dimana pesan dibagi menjadi blok sepanjang 4 bit dan dimasukkan secara sekuensial dari kiri ke kanan pada matriks.

Selanjutnya dari matriks yang dihasilkan dilakukan operasi perkalian xor dengan *subkey* pada iterasi tersebut. *Subkey* juga berbentuk dalam matriks agar dapat dioperasikan. Selanjutnya dilakukan substitusi nilai dari hasil perkalian xor dengan kotak S yang dihasilkan secara *pseudo-random*. Substitusi dilakukan dengan mengambil gabungan bit pertama dan ketiga sebagai nilai baris dan gabungan bit kedua dan keempat sebagai nilai kolom. Nilai baris dan kolom menentukan elemen substitusi yang tepat dari kotak S untuk menggantikan nilai pada sel pada blok matriks. Kotak S yang digunakan dapat dilihat pada tabel 2.

Selanjutnya dilakukan pergeseran terhadap matriks. Pergeseran dilakukan terhadap sel pada matriks. Isi sel pada baris pertama dan ketiga digeser ke kanan dan baris kedua dan keempat digeser ke kiri. Setelah dilakukan pergeseran. Dilakukan pengacakan dengan kotak permutasi P. Kotak P dapat dilihat pada tabel 3.

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

Tabel 2 Kotak S Untuk Substitusi Nilai pada blok pesan

5, 15, 7, 2, 4, 0, 9, 8, 12, 1, 10, 14, 3, 6, 11, 13

Tabel 3 Kotak P untuk Permutasi Pesan

1001	1000	0001	0110
0100	1100	1111	1101
0000	0101	0110	1110
1100	0011	0001	0010

Gambar 10 Contoh blok dalam matriks

1000	1001	1110	0000
0100	0101	0010	0011
0111	1101	0000	1011
0101	0001	1110	1100

Gambar 11 Hasil substitusi dari blok pesan matriks menggunakan kotak S

0110	1001	1000	0001
1100	1111	1101	0100
1110	0000	0101	0110
0011	0001	0010	1100

Gambar 12 Hasil pergeseran blok

1001	1000	0001	0110
0100	1100	1111	1101
0000	0101	0110	1110
1100	0011	0001	0010

Gambar 13 Hasil pengacakan blok menggunakan kotak P

C. *Jaringan Feistel*

Berikut adalah langkah-langkah algoritma OE-CK yang digambarkan dengan struktur feistel pada gambar 7.

1. Bagi pesan menjadi 2 bagian pesan sepanjang 64-bit
2. Untuk bagian kanan, ubah bagian menjadi blok matriks
3. Untuk tiap iterasi lakukan pembangkitan kunci dengan fungsi ekspansi kunci sesuai yang telah dijelaskan sebelumnya.
4. Lakukan operasi perkalian xor antara *subkey* dan blok matriks kanan
5. Lakukan operasi substitusi dengan kotak S
6. Lakukan operasi pergeseran terhadap matriks
7. Lakukan operasi pengacakan dengan kotak P
8. Dari hasil langkah di atas, bentuk bagian kiri yang baru dengan bagian kanan lainnya dan bagian kanan yang baru dengan hasil operasi xor dari operasi di atas dan bagian kiri yang awal.
9. Gabung hasil operasi di atas dalam sebuah kesatuan 128-bit
10. Ulangi hingga 10 kali iterasi

IV. EKSPERIMEN DAN PEMBAHASAN HASIL

Dilakukan pengujian algoritma OE-CK dengan 5 mode yaitu mode ECB, mode CBC, mode CFB, mode OFB, mode CTR. Eksperimen dilakukan pada plainteks berukuran 158 karakter (1264 bit yang berarti sekitar 10 blok pesan 128-bit) dan kunci berukuran 16 karakter (128-bit). Berikut adalah hasil dari tiap eksperimen,

Kunci	algoritmaoeckyes
Plainteks	Pemerintah dan DPR melalui panitia khusus (pansus) menyepakati keterlibatan TNI dalam revisi UU Antiterorisme. Pelibatan TNI ini menuai kontroversi dan kritik
Cipherteks (ECB)	Nã=!Ö,ÀìZ□È³±Äw□HW!†'»Ò×ib•]áçÖšâÒ¨ Âüjº Þ·Â ¿ì‰1□OÓ•zF□gØÝ•ÿ#ÊMßÇØ—hN € ? • Ž UvÆËt*ÝCÝlrsFÉñtPïNªù □}‡Ù3/eZƒÃ",FÇúÔá}ß™"ãhJËÃ-†Ô…¢ *ór+ä,fÇ„Ñ0ð ºïh î2}

Tabel 4 Hasil Enkripsi Algoritma OE-CK dengan mode ECB

Kunci	algoritmaoeckyes
Plainteks	Pemerintah dan DPR melalui panitia khusus (pansus) menyepakati keterlibatan TNI dalam revisi UU Antiterorisme. Pelibatan TNI ini menuai kontroversi dan kritik
Cipherteks (CBC)	<ˆlw¾i=KEÕ□ã®□9HÂf□¡ÀVióµH*ƒ* p9Ê□nÜÀqr¡õs•ì□s– ü.□ô(Ï)P¡óH+2ßX=nh□AÑLÿ=™Œ¶qmØ– JÙô4ÈLÏú ž ;»Ph-, ÕòhKR' Ö]H>h·À(QmH- Þ²*ujÿ¿‡ k8™× • wT¾ÎÁc |Š

Tabel 5 Hasil Enkripsi Algoritma OE-CK dengan mode CBC

Kunci	algoritmaoeckyes
Plainteks	Pemerintah dan DPR melalui panitia khusus (pansus) menyepakati keterlibatan TNI dalam revisi UU Antiterorisme. Pelibatan TNI ini menuai kontroversi dan kritik
Cipherteks (CFB	"ç‹™‰{ÓÄÍ¡ûmôó¸L`öÓo+H›¸□q;ó*ƒ*{ÎFXˆev §Ø'Ñ·é0Jî£©Ùó`E|7d4 • Äâ¾j#1CY&uÀ t¯DÄF5UŠàP™□TÚ□MÅ±Õ*É™,âØÆþK» GŠj",íâtëÕ— Þ†75ZQ¸Ê¹£èÖJ‰jñfÂÂÏ□y³/pMöÆLæŠ}Á

Tabel 6 Hasil Enkripsi Algoritma OE-CK dengan mode CFB

Kunci	algoritmaoeckyes
Plainteks	Pemerintah dan DPR melalui panitia khusus (pansus) menyepakati keterlibatan TNI dalam revisi UU Antiterorisme. Pelibatan TNI ini menuai kontroversi dan kritik
Cipherteks (OFB)	”ç‹™‰{ÓÄÍ¡ûmôó6~1%ªö”âNz°SÊùñt×9 • ¾ wö,ñÓ&‘ÄõÇ=œwXJ@¸˜¾«W¸ŠíJs»Dí9Ÿˆ´O Õü™ÃþŠ*]□ˆ…û1þï0Äµ½-µø8býL# Ñ¨ • • j¦ëÿ, YÊû I Þ¯vF¯ZT]ºæÈG— #ËÃ S.À?ì

Tabel 7 Hasil Enkripsi Algoritma OE-CK dengan mode OFB

Kunci	algoritmaoeckyes
Plainteks	Pemerintah dan DPR melalui panitia khusus (pansus) menyepakati keterlibatan TNI dalam revisi UU Antiterorisme. Pelibatan TNI ini menuai kontroversi dan kritik
Cipherteks (CTR)	”ç‹™‰{ÓÄÍ¡ûmôó”·□ƒ□ŒtËÐÌ¡êm½É-Jí… ƒ•fÒ…©ìm§À·d-ƒ□□lÂ¥Äêþxôß¡òhœ‡‰wÆQÄï¾X#□” ma □†ÀgÂ£Ìòò,8□“…>I‡Ÿ…gÈ7ÌòñiCôì¡•äŒŠ” tÉñÏÝ,ºØä÷H□□□|‡Êïê~¢Ð¶„Î□□{‡®×èîe

Tabel 8 Hasil Enkripsi Algoritma OE-CK dengan mode CTR

Berikut adalah hasil eksperimen pengubahan satu kata pada plainteks pada mode ECB dan CBC.

Kunci	algoritmaoeckyes
Plainteks	Pemerintah dan KPK melalui panitia khusus (pansus) menyepakati keterlibatan TNI dalam revisi UU Antiterorisme. Pelibatan TNI ini menuai kontroversi dan kritik
Cipherteks (ECB)	NÚ=!Ö,ÀìJ□È¼±ÄwyH³!†’»Ò'ib“]áçÒšâÒ¨Â üjºÞ·Â ¿ì‰l□OÓ•zF□gØÝ•ÿ#ÊMßÇØ—hN € ? • Ž UvÆÏt*ÝCÝlrsFÉñtPïNªù □}‡Ù3/eZƒÃ",FÇúÔá}ß™”ãhJËÃ-†Ô…¢ *ór+ä,fÇ„Ñ0ð ºïh î2}

Tabel 9 Hasil Enkripsi Algoritma OE-CK dengan mode ECB pada Plainteks yang Mengalami Perubahan Satu Kata

Kunci	algoritmaoeckyes
Plainteks	Pemerintah dan KPK melalui panitia khusus (pansus) menyepakati keterlibatan TNI dalam revisi UU Antiterorisme. Pelibatan TNI ini menuai kontroversi dan kritik
Cipherteks (CBC)	<ˆlw¾i=ŠEÕ • ã® • éHÂf • ¡ÁV–µHƒ pIÊ□aÜÀqt~s•ì□Ó– ü)□ô+Ï!Ð¡óH+2?X=kh□AÔL¥-™Œ¶±mØœ JÙÿ4§üÏú î;»\h-, ¡¢hKR‘ †]L>h¼À‘ ÁmH-Þ¡²*ujÿ¿ • kà™× • wT¾ÎÁg |Ž

Tabel 10 Hasil Enkripsi Algoritma OE-CK dengan mode CFB pada Plainteks yang Mengalami Perubahan Satu Kata

Dapat dilihat pada hasil di atas, pengubahan 1 kata pada plainteks hanya membuat perubahan pada 1 blok pesan pada cipherteks untuk ECB. Sementara pada CFB cipherteks setelah pengubahan kata juga ikut berubah akibat ketergantungan antar blok saat enkripsi.

V. ANALISIS KEAMANAN

Berikut adalah hasil eksperimen dan analisis keamanan yang dilakukan terhadap plainteks yang dienkripsi dengan algoritma OE-CK.

A. Analisis Frekuensi

Dilakukan pengujian dengan plainteks berulang untuk mengetahui apakah algoritma dapat diserang secara statistik. Pengujian dilakukan dengan melihat apakah terdapat pengulangan kata pada cipherteks apabila plainteks memiliki banyak kata berulang. Contoh yang diambil adalah “That that is is that that is not is not. Is that it? It is.”

Mode	CBC	ECB
Kunci	algoritmaoeckyes	algoritmaoeckyes
Plainteks	That that is is that that is not is not. Is that it? It is.	That that is is that that is not is not. Is that it? It is.
Cipherteks	Ï,ì’ìÇ«Ã°mþ□¤‹ÿê • ø Ò c<□6`!™ô³ 5Q‘Ùöûv´=í6c€ 9a ÇÖDt4Í×làt-	o)²Ž D^9qeWLÜ • ïC ÛºÛìRH@S‹§Ã!=¹# W)e«ŠoÙ»Hb9ˆWvQ 57uN2a`ë6I`

Tabel 11 Hasil Enkripsi Algoritma OE-CK pada Plainteks yang Memiliki Banyak Kata Berulang

Pada hasil di atas dapat dilihat bahwa tidak ada karakter yang berulang pada cipherteks walaupun begitu banyak pengulangan kata pada plainteks. Hal ini terjadi karena diterapkannya prinsip *confusion* dan *diffusion* dengan pengacakan dan substitusi pada pesan dan kunci sehingga menyulitkan kriptanalis dalam memecahkan pesan.

B. Perbedaan Kunci Enkripsi

Kriptanalis dapat melakukan penyerangan dengan melihat cipherteks hasil enkripsi dari 2 kunci yang berdekatan untuk melihat pola pemetaan algoritma pada mode ECB. Untuk itu perlu dilakukan percobaan dengan mengganti 1 huruf pada kunci untuk melihat apakah terdapat suatu pola dalam cipherteks. Berikut adalah hasil pengujian pada algoritma OE-CK.

Kunci	algoritmaoeckye**s**	algoritmaoeckye**a**
Plainteks	That that is is that that is not is not. Is that it? It is.	That that is is that that is not is not. Is that it? It is.
Cipherteks	o)²Ž D^9qeWLÜ • ïC ÛºÛìRH@S‹§Ã!=¹# W)e«ŠoÙ»Hb9ˆWvQ 57uN2a`ë6I`	os„ˆ”0$<µƒLë • ¥? • ÿ ¿Ë<Ö …QÐ¤ŒWÃ H]8¸ã·ímOòŸè½vbb¨| êX•W³r • ÀQÉIš

Tabel 12 Hasil Enkripsi Algoritma OE-CK pada Plainteks yang Sama dan Kunci yang Berdekatan

Dapat dilihat pada tabel di atas perbedaan cipherteks sangat signifikan sehingga sangat sulit untuk menemukan pola kunci dan plainteks walaupun kunci diubah 1 karakter. Hal ini dikarenakan diterapkannya prinsip *diffusion* melalui transformasi kunci dan permutasi.

C. Perbedaan Karakter Plainteks

Kriptanalis juga dapat melakukan penyerangan dengan melihat cipherteks hasil enkripsi dari 2 plainteks yang berdekatan untuk melihat pola pemetaan algoritma pada mode ECB. Berikut adalah hasil pengujian pada algoritma OE-CK.

Kunci	algoritmaoeckye**s**	algoritmaoeckye**s**
Plainteks	Alice mengirim surat rahasia ke Bob.	Alise mengirim surat rahasia ke Bob.
Cipherteks	tæà • Êjf-Ì9õ¦Vùî □dGmI=9EÚƒÃkD– □W□@□Û□ì□50œ • ï	yæà • Íhf½Ì9õ¦6Y □dGmI=9EÚƒÃkD– □W□@□Û□ì□50œ • ï

Tabel 13 Hasil Enkripsi Algoritma OE-CK pada Plainteks yang Berdekatan dan Kunci yang Sama

Dari tabel di atas,ditemukan bahwa pengubahan 1 huruf pada plainteks hanya membuat perubahan pada 1 blok pesan pada cipherteks untuk mode ECB.

D. Serangan Brute Force

Keamanan OE-CK bergantung pada panjang kunci yang digunakan untuk mengenkripsi pesan. Semakin panjang kunci maka akan semakin baik. Namun karena OE-CK memiliki kunci minimal sepanjang 128 bit maka melalui perhitungan terdapat minimal 2^{128} yaitu sekitar 3.4 x 10^{38} kunci. Jika diasumsikan setiap 1 detik komputer dapat menjalankan 10^6 percobaan, maka diperlukan waktu sekitar 1.55 x 10^{28} tahun untuk memecahkan kunci dengan 1 komputer. Waktu ini dapat terus bertambah dengan bertambahnya panjang kunci.

VI. KESIMPULAN DAN SARAN

Algoritma OE-CK menggunakan struktur feistel yang bekerja pada kunci sepanjang 128-bit dan blok sepanjang 128-bit. Algoritma ini memanfaatkan posisi ganjil genap pada bit-bit kunci serta pengacakan dengan kotak P untuk meningkatkan *confusion*. Operasi yang dilakukan pada fungsi *round*-nya adalah operasi perkalian xor,operasi substitusi dan operasi pergeseran. Dari hasil uji keamanan, Algoritma OE-CK memiliki tingkat keamanan yang cukup baik kecuali pada mode ECB.

Algoritma OE-CK dapat dikembangkan lebih lanjut dengan meningkatkan panjang kunci yang digunakan atau jumlah iterasi yang dilakukan.

REFERENSI

[1] informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2017-2018/kripto17-18.htm diakses pada 10 Maret 2018 pukul 19.00 WIB.

[2] https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation diakses pada 15 Maret 2018 pukul 09.00 WIB.

[3] https://en.wikipedia.org/wiki/Block_cipher diakses pada 15 Maret 2018 pukul 09.10 WIB.

[4] https://en.wikipedia.org/wiki/Confusion_and_diffusion diakses pada 15 Maret 2018 pukul 09.20 WIB.

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, 16 Maret 2018

Penulis