Multiplex: Algoritma Blok Cipher berbasis Perkalian

Vania Velda[1], **Yoel Susanto**[2].

[1,2] Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika (STEI), Institut Teknologi Bandung (ITB), Jalan Ganesha 10, Bandung 40132
E-mail: 13517090@std.stei.itb.ac.id,13517014@std.stei.itb.ac.id

Abstrak. Kriptografi adalah teknik untuk melakukan scrambling terhadap pesan dengan tujuan menjaga kerahasiaan pesan tersebut. Saat ini, banyak algoritma blok cipher modern yang dikembangkan untuk membuat pesan semakin aman. Algoritma multiplex merupakan algoritma yang kami usulkan berdasarkan algoritma blok cipher yang menggunakan perkalian untuk membuat kunci enkripsi yang berbeda pada tiap blok. Selain itu , Multiplex menggunakan dua jenis S-box untuk membuat hubungan dekripsi semakin sulit.
Keywords: kriptografi, blok cipher, perkalian, multiplex

1. Pendahuluan

Kriptografi berasal dari bahasa Yunani, "kryptós" yang berarti tersembunyi dan "gráphein" yang berarti tulisan [1].Kemunculan pertama kriptografi dapat dilihat dari Bangsa Mesir 4000 tahun yang lalu dimana Bangsa Mesir menggunakan *hieroglyph*[2] yang tidak standar dalam menulis pesan.Tujuan dari kriptografi adalah untuk menjaga kerahasiaan pesan ,integritas data, autentikasi serta sebagai layanan untuk mencegah penyangkalan.Proses dasar kriptografi adalah enkripsi dan dekripsi. Enkripsi merupakan proses untuk mengamankan pesan / *plaintext* agar tidak terbaca orang lain sedangkan dekripsi mengubah *ciphertext* menjadi *plaintext*

Pada saat ini, kriptografi modern telah berkembang dan menghasilkan berbagai algoritma *cipher* yang lebih bagus sebagai hasil dari perbaikan dari kriptografi klasik. Salah satu jenis kriptografi modern adalah *block cipher*. *Block cipher* termasuk dalam algoritma cipher kunci simetris dimana cipher blok beroperasi pada blok atau kelompok bit dengan panjang tetap dan menggunakan transformasi yang tetap untuk semua digit pada blok. Algoritma AES dan DES merupakan contoh algoritma block cipher yang paling umum digunakan.

Cipher blok yang ideal akan terwujud apabila hubungan antara input blok dan output blok sangat random namun masih dapat didekripsi kembali [3]. Pada prinsipnya, algoritma cipher blok didasarkan dari struktur Feistel untuk melakukan dekripsi secara efisien. Bentuk dasar dari cipher block didasarkan dari prinsip shannon untuk memberikan hubungan antar ciphertext dan kuncinya serumit mungkin.

2. Studi Pustaka

2.1. Jaringan Feistel

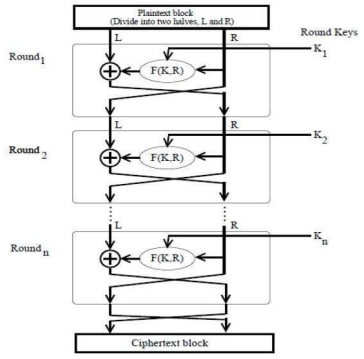
Jaringan Feistel merupakan struktur yang diperkenalkan oleh Horst Feistel.Jaringan Feistel dimana jaringan Feistel dibuat berdasarkan struktur Shannon yang diperkenalkan pada tahun 1945. Proses enkripsi pada jaringan Feistel terdiri dari beberapa putaran pemrosesan *plaintext* dimana pada setiap putaran terdiri dari langkah substitusi ditambah dengan langkah permutasi.Penentuan jumlah putaran

dari desain algoritma yang umumnya digunakan sebanyak 16 putaran. Semakin banyak jumlah putaran, maka akan semakin terjamin kerahasiaannya namun akan membuat proses enkripsi dan dekripsi menjadi tidak efisien dan lambat.

Proses enkripsi pada jaringan Feistel adalah sebagai berikut [4]:

1. Input dipecah menjadi dua, misal namanya L dan R.
2. Untuk setiap putaran , R tidak berubah . L dieksekusi dengan bergantung pada R dan kunci enkripsi K.Hasil berupa f(R,K) $\oplus$ L
3. Pada implementasi dunia nyata , K yang digunakan merupakan subkey dari kunci enkripsi sehingga setiap putaran menggunakan kunci yang berbeda
4. Pada langkah permutasi, L dan R ditukar sehingga L untuk putaran berikutnya merupakan R dari putaran sekarang dan R untuk putaran berikutnya merupakan hasil L .
5. Ulangi hingga semua round. Pada putaran terakhir, hasil R dan L akan dikonkat untuk menghasilkan blok ciphertext.

Ilustrasi dari jaringan Feistel[4]



Proses deskripsi dari jaringan Feistel hampir sama dengan proses enkripsi. Penggunaan subkey dilakukan dalam urutan terbalik

The title is set 17 point Times Bold, flush left, unjustified. The first letter of the title should be capitalized with the rest in lower case. It should not be indented. Leave 28 mm of space above the title and 10 mm after the title.

2.2. Diffusion and Confusion

Pada paper mengenai fondasi dari kriptografi [“Communication theory of secrecy systems,” Bell Systems Technical Journal 28 (1949), 656 – 715]yang ditulis oleh Claude Shannon terdapat dua hal penting untuk sistem kriptografi yang berfungsi untuk menyembunyikan analisis statistik. Dua hal tersebut adalah *diffusion* dan *confusion*..

Diffusion memiliki pemahaman bahwa apabila sebuah karakter dalam plaintext diubah, maka beberapa karakter pada ciphertext akan berubah dan apabila sebuah karakter pada ciphertext diubah, maka beberapa karakter pada plaintext akan berubah. Perubahan tersebut akan membuat hubungan statistik antara ciphertext dan plaintext semakin rumit. serta menghasilkan cipherteks yang memiliki perbedaan signifikan dengan cipherteks tanpa *diffusion*.

Confusion merupakan prinsip untuk membuat hubungan antara kunci dan ciphertext serumit mungkin khususnya dimana setiap karakter pada ciphertext harus bergantung pada beberapa bagian dari kunci.Perubahan yang diharapkan tentu perubahan kunci secara keseluruhan. Apabila kunci dapat diubah seluruhnya, kriptanalisis akan semakin sulit mencari kunci dan perlu mencari kunci secara keseluruhan daripada mencari bagian bagian kunci.

2.3. Block Cipher

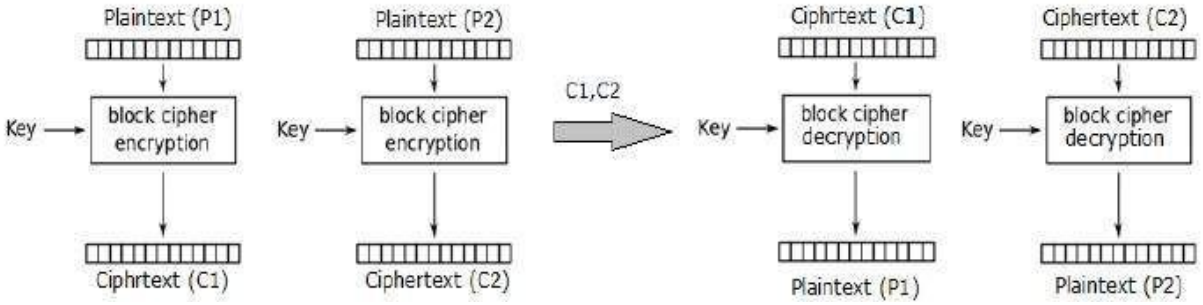
Algoritma *block cipher* melakukan enkripsi pesan dengan membagi pesan tersebut ke blok blok dengan ukuran yang sama. Setiap blok kemudian akan dienkripsi secara independen dari blok lain yang menghasilkan perubahan blok plaintext menjadi blok ciphertext dengan ukuran yang sama. Pada saat ini, blok cipher modern telah digunakan secara luas dikarenakan block cipher mudah dimengerti dan mudah untuk diimplementasikan pada software maupun hardware.

Terdapat 5 mode operasi yang berkaitan dengan operasi blok sebelum dilakukan enkripsi dan enkripsi. Kelima mode tersebut adalah :

1. *Electronic Code Book / ECB*

   Mode ECB merupakan algoritma tercepat dan paling mudah diimplementasikan dikarenakan menggunakan substitusi yang simple. ECB merupakan algoritma terlemah dibanding mode mode lainnya. Input akan dibagi menjadi beberapa blok dan di enkripsi secara individual menggunakan kunci. Dekripsi setiap blok juga dilakukan secara individual. Tetapi , mode ECB tidak akan cocok digunakan untuk block berukuran kecil, misalnya apabila ukurannya lebih kecil dari 40 bits. Hal ini dikarenakan beberapa kata atau kalimat dapat digunakan berulang yang mengakibatkan adanya kemuncul blok ciphertext yang repetitif. Untuk mengatasi hal tersebut, dapat dilakukan dengan menambah pad bits pada setiap blok.

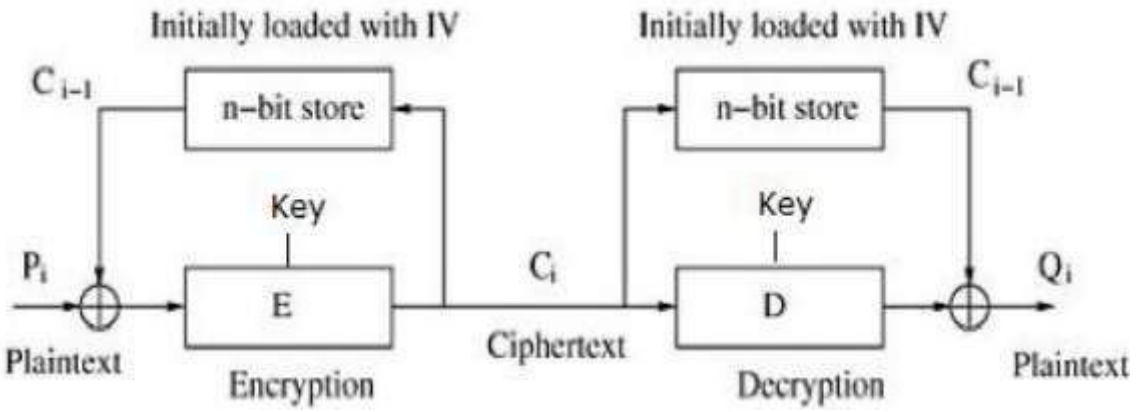
   Ilustrasi Enkripsi dan Dekripsi pada ECB
   (Sumber https://www.tutorialspoint.com/cryptography/block_cipher_modes_of_operation.htm)



2. *Cipher Block Chaining / CBC*

   Sesuai dengan namanya, blok blok pada algoritma ini memiliki ketergantungan antara satu sama lain.Ketergantungan ini membuat sistem yang menggunakan CBC menjadi tidak deterministic. Pada CBC, blok pertama dari plaintext digunakan sebagai fungsi binary untuk XOR yang akan membandingkan dua bit dan mengubah hasil ciphertext dengan bit ketiga dengan menggunakan vektor inisialisasi. Hasil ciphertext setiap blok akan digunakan untuk enkripsi blok plaintext berikutnya. Kesalahan pada satu ciphertext akan memberikan efek domino yang mempengaruhi ciphertext ciphertext berikutnya

   Ilustrasi Enkripsi dan Dekripsi pada CBC
   (Sumber https://www.tutorialspoint.com/cryptography/block_cipher_modes_of_operation.htm)

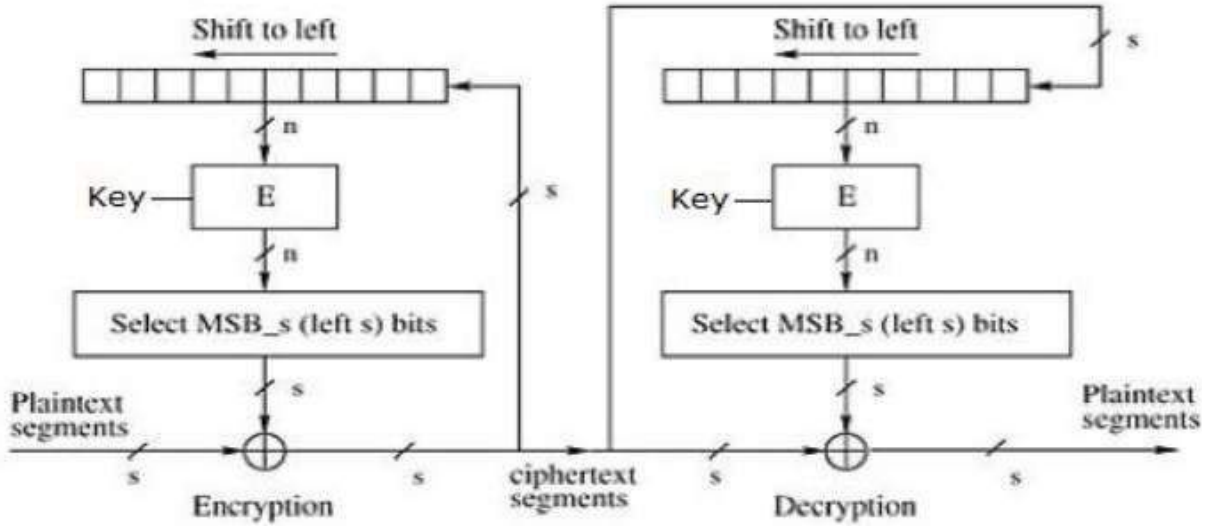


3. *Cipher Feedback / CFB*

Perbedaan utama dari CFB adalah CFB merupakan *stream mode*.*Stream* mode berarti enkripsi pesan dilakukan dengan menggunakan pseudo random cipher digit stream dimana setiap bit akan dienkripsi dengan satu cipher digit yang bersesuaian.CFB menggunakan inisialisasi vektor seperti pada CBC.Pada CFB, ciphertext blok sebelumnya dienkripsi dan hasilnya di XOR dengan blok plaintext sekarang untuk menghasil blok ciphertext sekarang.Adanya data loss dikarenakan shift membuat kriptanalisis sulit meng dekripsi pesan.

Ilustrasi Enkripsi dan Dekripsi pada CFB

(Sumber https://www.tutorialspoint.com/cryptography/block_cipher_modes_of_operation.htm)



4. *Output Feedback / OFB*

Struktur dari OFB hampir mirip dengan CFB dimana pada OFB hasil dari fungsi enkripsi menjadi feedback bagi shift register sedangkan pada CFB unit ciphertext yang menjadi feedback untuk shift register pada blok. Perbedaan lainnya adalah OFB beroperasi untuk full blok dari plaintext dan ciphertext. Keuntungan utama dari OFB adalah adanya error bit pada transmisi tidak akan memberikan efek domino pada enkripsi. Namun OFB lebih rentan terhadap serangan modifikasi pesan *stream* dibandingkan dengan mode CFB.

Ilustrasi Enkripsi dan Dekripsi pada OFB

(Sumber :
https://www.includehelp.com/cryptography/output-feedback-mode-ofb-in-cryptography.aspx)

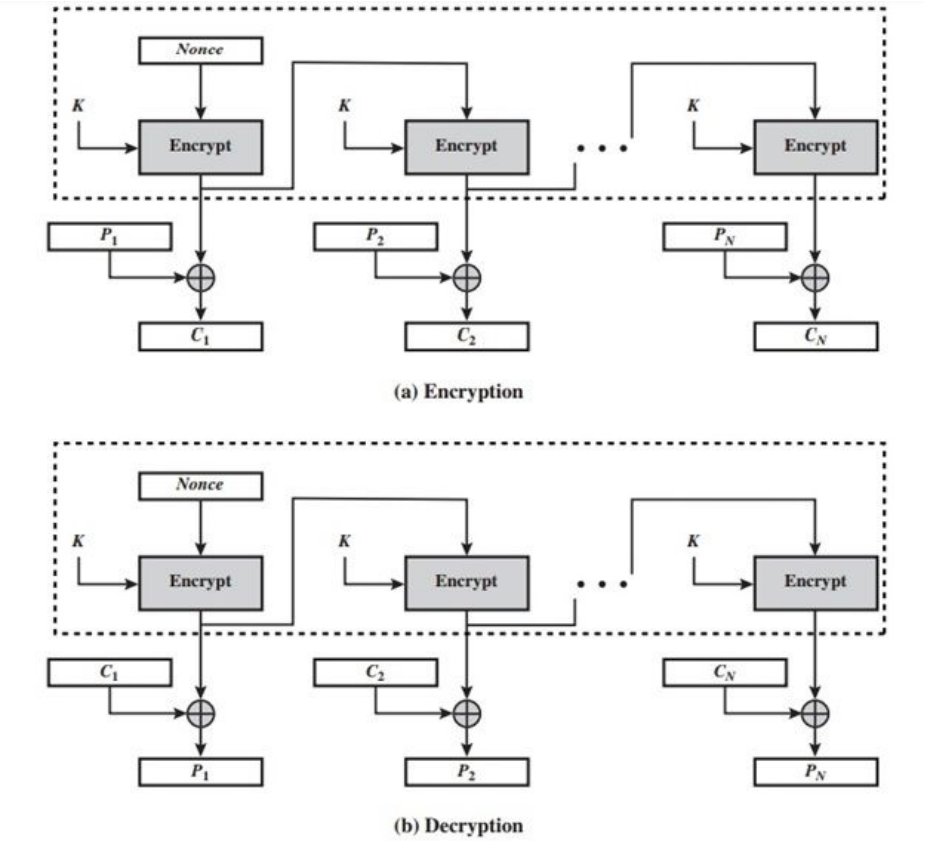


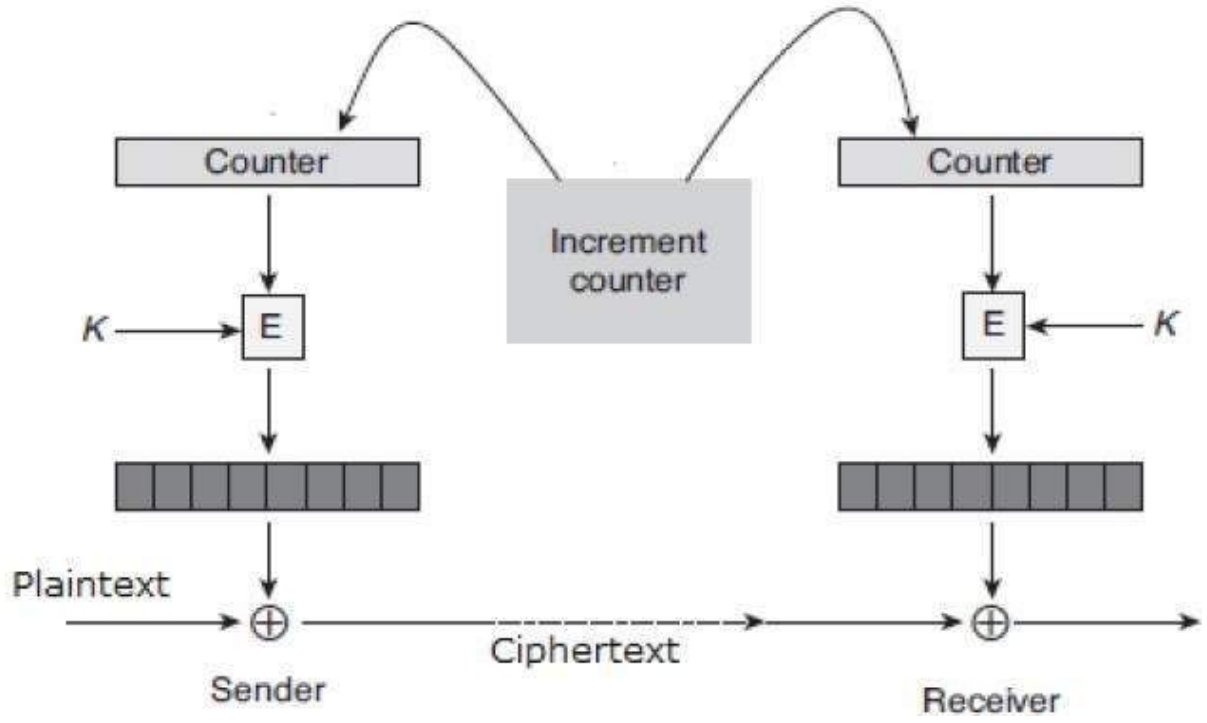
Figure 6.6 Output Feedback (OFB) Mode

5. *Counter Mode* / CTR *mode*

Diperkenalkan pertama kali oleh Diffie dan Hellman pada tahun 1979, CTR menggunakan nomor random sebagai counter yang berubah untuk setiap blok yang akan dienkripsi. Counter akan di enkripsi dengan cipher dimana hasilnya akan di XOR menjadi ciphertext. Counter yang sering berubah menyelesaikan permasalahan yang ada pada ECB.CTR menjamin tidak dapat idi attack selama jumlah blok yang dienkripsi kurang dari $2^{n/2}$ [7].

Ilustrasi Enkripsi dan Dekripsi pada Counter

(Sumber https://www.tutorialspoint.com/cryptography/block_cipher_modes_of_operation.htm)



2.4. S-box

S-box merupakan komponen dasar dari algoritma kunci simetris yang berfungsi untuk melakukan substitusi. Pada blok cipher, S-box digunakan untuk membuat hubungan kunci dan ciphertext menjadi sulit dilacak dan digunakan pada Confusion dari Shannon. S-box yang digunakan merupakan Rijndael S-Box yang digunakan pada algoritma AES.S-box jenis ini melakukan map antara input 8 bit menjadi output 8 bit. Selain itu terdapat inverse S-box yang merupakan reverse dari S-box.

Rjindael S-box (Sumber : https://www.researchgate.net/publication/318906543_Enhanced_Hybrid_Algorithm_of_Secure_and_Fast_Chaos-based_AES_RSA_and_ElGamal_Cryptosystems/figures?lo=1)

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1x	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2x	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3x	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4x	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5x	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6x	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7x	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8x	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9x	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
ax	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
bx	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
cx	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
dx	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
ex	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
fx	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

3. Rancangan Algoritma

Multiplex adalah algoritma *block cipher* yang bekerja dengan melakukan manipulasi bit informasi dalam beberapa metode. Dalam perancangannya, Multiplex menerapkan prinsip-prinsip perancangan algoritma *block cipher.* Confusion serta diffusion diterapkan melalui *substitution-permutation network* yang melibatkan substitusi oleh kotak-S dan transformasi posisi informasi. Multiplex juga menggunakan prinsip *cipher* berulang menggunakan upa kunci yang berbeda pada tiap iterasi fungsi enkripsi. Pada akhirnya agar proses dekripsi dan enkripsi tidak perlu dilakukan dengan algoritma yang berbeda, Multiplex menggunakan jaringan Feistel.

Multiplex dikembangkan secara modular agar operasi enkripsi dapat dilakukan pada berbagai mode. Multiplex mendukung operasi pada lima mode enkripsi yaitu *Electronic Code Book(ECB), Cipher Block Chaining(CBC), Cipher Feedback (CFB), Output Feedback(OFB)* serta *Counter Mode.* Fungsi enkripsi Multiplex melakukan operasi pada *block* informasi berukuran 64 bit. Setiap block informasi akan melalui iterasi enkripsi berulang sebanyak 16 *round*. Rancangan ini membuat kriptanalisis sulit dilakukan untuk memecahkan algoritma Multiplex.

Dalam implementasinya, algoritma Multiplex memiliki fungsi enkripsi yang melakukan 8 operasi untuk *information scrambling*. Berikut 8 operasi yang dilakukan pada fungsi inti Multiplex:

A. Substitusi S-Box

Pada operasi substitusi S_Box, Multiplex menggunakan dua jenis S_Box yaitu AES S-Box atau dikenal juga sebagai Rijndael S-box. S-Box kedua yang digunakan adalah AES S-Box yang sudah melalui proses inverse. Masing-masing informasi dibagi menjadi panjang 1 *byte*, lalu dengan menggunakan representasi bentuk hexadecimalnya, kita melakukan substitusi dengan S-Box pada posisi sesuai nilai hexadesimal dari informasi tersebut. Pada operasi substitusi S-Box pertama, S-Box yang digunakan adalah Rijndael S-Box sedangkan pada operasi S-Box kedua, digunakan S-Box *inverse* nya.

Contoh Rijndael S-Box beserta Inversenya [5]

AES S-box

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

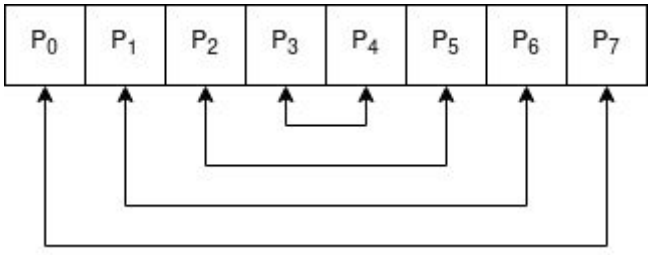
Inverse S-box

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

B. Transposisi Genap dan Ganjil

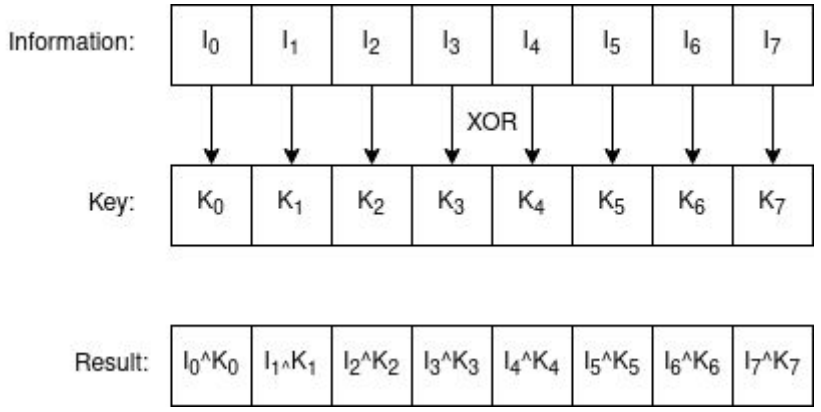
Pada transposisi ganjil dan genap, Multiplex terlebih dahulu memotong informasi *block* berukuran 64 bit menjadi *array* berisi 8 potongan berukuran 8 bit. Lalu untuk setiap potongan informasi tersebut. Kami melakukan penjumlahan antara nilai *integer* dari potongan[i] dengan potongan[7-i]. Pada transposisi genap, pertukaran nilai hanya terjadi apabila hasil penjumlahan berupa nilai genap dan begitu pula pada transposisi ganjil.

Ilustrasi transposisi genap dan ganjil yang digunakan



C. XOR dengan kunci
Pada operasi XOR dengan kunci. Informasi *block* dengan panjang 64 bit kembali dipecah menjadi 8 bagian dengan ukuran yang sama. Hal tersebut dilakukan juga pada kunci. Dengan demikian kita dapat melakukan operasi biner XOR pada potongan informasi dan kunci pada indeks yang sama.

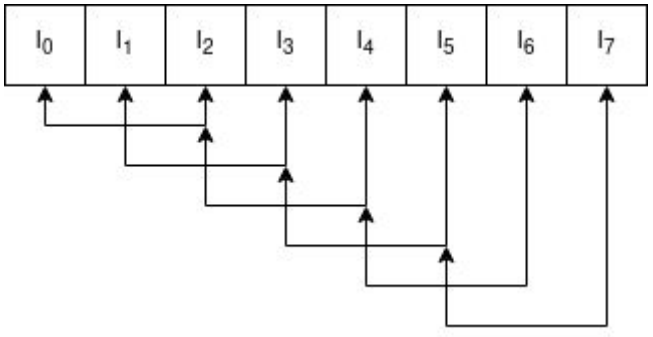
Ilustrasi operasi kunci menggunakan XOR



D. Substitusi Prima
Operasi substitusi prima melakukan pertukaran informasi jika indeks dari informasi tersebut termasuk nilai prima. Nilai prima yang dipertimbangkan adalah nilai prima dari 0 - 63 sesuai dengan panjang *block* operasi algoritma Multiplex.

E. Transposisi Flip
Transposisi flip melakukan operasi pertukaran bit informasi pada indeks *i* dengan *i+2*. Pertama, operasi ini membagi blok informasi menjadi bagian berukuran 1 byte. Lalu dilakukan operasi pertukaran informasi pada masing-masing bagian 1 byte tersebut. Dimulai dari indeks 0 sampai dengan 5, pasangan index *i* dan *i+2* akan ditukar nilainya.

Ilustrasi Transposisi Flip



F. Pembangkitan Upa Kunci
Pada akhir dari setiap round, Multiplex akan melakukan pembangkitan upa kunci yang baru. Upa kunci ini dibangkitkan dengan cara mengalikan bagian 1 byte dari kunci dan bagian 1 byte dari teks biasa. Lalu algoritma Multiplex akan melakukan modulo 256 terhadap nilai yang dihasilkan. Setiap nilai tersebut disimpan lalu akan gabungkan menjadi upa kunci yang baru. Dengan demikian setiap round enkripsi akan menggunakan key yang berbeda.

4. Eksperimen dan Analisis Hasil

A. Eksperimen

Eksperimen dilakukan dengan menjalankan algoritma Multiplex pada semua mode operasi *block cipher*. Untuk setiap mode operasi, kami melakukan pengujian pada 3 berkas dengan masing-masing berkas berukuran 10,100 dan 1000 bytes. Kami mencatat kecepatan algoritma Multiplex dalam satuan waktu sekon.

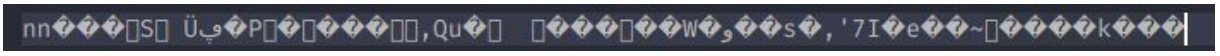
Tabel 1 Hasil multiplex dalam sekon

	ECB		**CBC**		**CFB**		**OFB**		**counter**	
	Enkripsi	Dekripsi	Enkripsi	Dekripsi	Enkripsi	Dekripsi	Enkripsi	Dekripsi	Enkripsi	Dekripsi
10 Bytes	0.02	0.03	0.02	0.02	0.26	0.29	0.23	0.36	0.02	0.02
100 Bytes	0.14	0.13	0.12	0.15	1.85	1.67	1.64	2.08	0.15	0.21
1000 Bytes	1.01	1.01	0.99	0.90	14.26	15.23	16.35	16.71	0.84	0.82

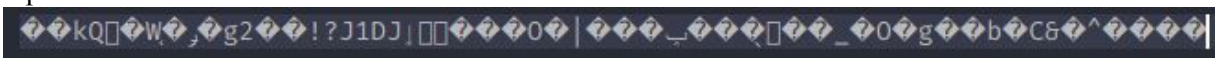
B. Analisis

Pada algoritma Multiplex, fitur confusion dan diffusion merupakan bagian yang sangat penting agar kriptanalisis tidak dapat dilakukan dengan mudah. Kita akan menguji coba dengan melakukan perubahan 1 byte text dan membandingkan hasil enkripsi.

Kasus 1:
Plain Text: Multiplex: Algoritma Blok Cipher berbasis Perkalian
Cipher Text:



Kasus 2:
Plain Text: Multiplex: algoritma Blok Cipher berbasis Perkalian
Cipher Text:



Hanya dengan mengubah kapitalisasi dari sebuah char pada teks biasa, kita dapat menghasilkan output enkripsi yang sama sekali berbeda.

Dengan demikian diffusion dan confusion pada algoritma Multiplex berhasil bekerja.

5. Kesimpulan dan Saran

Algoritma multiplexing yang dijelaskan pada paper ini menggunakan prinsip blok cipher dalam pengimplementasiannya. Selain itu, Multiplexing telah dapat melakukan enkripsi dan dekripsi dengan baik sehingga dapat mengatasi serangan brute force. Dapat dilihat dari hasil pada bagian IV, hasil cipher text yang dihasilkan sangat jauh berbeda dari plain text. Selain itu, perubahan satu huruf saja membuat perubahan yang sangat berdasar pada ciphertext yang dihasilkan.Namun dapat dilihat bahwa pengimplementasian untuk CFB dan OFB relatif lambat. Pada bagian IV A, dapat dilihat bahwa untuk ukuran 1000 Bytes, CFB dan OFB relatif lambat dibandingkan mode lainnya. Sehingga kedepannya, diperlukan perbaikan untuk mempercepat proses dari CFB dan OFB.

6. Referensi

[1] Rinaldi Munir. Pengantar Kriptografi. Slide Kuliah IF4020 Kriptografi, 2018.
[2] https://www.tutorialspoint.com/cryptography/origin_of_cryptography.htm
[3] http://www.cs.man.ac.uk/~banach/COMP61411.Info/CourseSlides/Wk2.1.DES.pdf
[4] https://www.tutorialspoint.com/cryptography/feistel_block_cipher.htm

[5] https://en.wikipedia.org/wiki/Rijndael_S-box
[6] https://www.sciencedirect.com/topics/computer-science/cipher-feedback
[7] https://link.springer.com/chapter/10.1007/978-3-319-78375-8_24

7. Acknowledgments

Puji syukur ke hadirat Tuhan yang Maha Esa karena telah memberikan kesempatan kepada penulis untuk menyelesaikan makalah ini. Penulis juga berterima kasih kepada dosen pengampu mata pelajaran IF4020 Kriptografi, Dr. Ir. Rinaldi Munir, MT., yang telah memberikan pengetahuan mengenai kriptografi dan block cipher sehingga Multiplex dapat dibuat. Selain itu penulis juga berterima kasih kepada semua pihak yang turut ikut serta membantu menyelesaikan makalah ini. Penulis berharap agar makalah ini dapat membantu menambah wawasan bagi pembaca mengenai block cipher.