Vibranium Cipher

Simple and Secure Cipher

Muhammad Naufal
Teknik Informatika / Sekolah Tinggi Elektro dan Informatika
Institut Teknologi Bandung
Bandung, Indonesia
mnaufal75@gmail.com

Cut Meurah Rudi
Teknik Informatika / Sekolah Tinggi Elektro dan Informatika
Institut Teknologi Bandung
Bandung, Indonesia
cutmeurahrudi@gmail.com

***Abstract*—Dalam makalah ini, penulis mengusulkan sebuah algoritma blok cipher baru, bernama Vibranium Cipher. Cipher ini mengandalkan berbagai operasi matematika seperti pertambahan modulo, pergeseran bit, dan operasi XOR. Selain itu, cipher ini juga menggunakan tabel permutasi dan tabel S-Box untuk meningkatkan kompleksitas.**

Keyword—cipher; block; XOR; S-Box; modulo; bit

I. PENDAHULUAN

Kriptogragfi merupakan ilmu mengamankan pesan, sehingga pesan tidak dibaca/diakses oleh yang tidak berkepentingan. Ilmu ini telah digunakan sejak zaman Julius Caesar. Dengan berkembangnya ilmu pengetahuan, maka ilmu kriptografi juga semakin berkembang

Block Cipher merupakan salah satu algoritma kriptografi, dimana setiap plaintext dibagi menjadi blok-blok. Kemudian pengoperasian akan dilakukan berdasarkan blok-blok tersebut.

Dalam paper ini, penulis ingin mengajukan sebuah algoritma *block cipher* yang baru, yang dinamakan Vibranium Cipher. Kami mengajukan algoritma ini karena kami ingin adanya suatu algoritma *block cipher* yang mudah diimplementasikan serta sederhana.

II. DASAR TEORI

A. Block Cipher

Blok cipher adalah suatu metode pengenkripsian yang beroperasi berdasarkan blok. Cara kerjanya adalah dengan cara membagi plaintext menjadi blok-blok yang besarnya telah ditentukan sebelumnya.

Terdapat 5 mode pengoperasian dalam *Block Cipher*:

1. Electronic Code Book (ECB)

Plaintext dibagi menjadi blok-blok, dan lakukan enkripsi pada setiap blok tersebut secara terpisah

2. Cipher Block Chaining (CBC)

Setiap blok plaintext di-XOR-kan dengan *ciphertext* sebelumnya sebelum melakukan enkripsi

3. Cipher Feedback (CFB)

Pada mode ini, membuat *block cipher* menjadi *stream cipher*. Mode ini sangat mirip dengan mode CBC, dimana ciphertext sebelumnya dienkripsi terlebih dahulu, kemudian di-XOR-kan dengan plaintext.

4. Output Feedback (OFB)

Mode ini juga membuat *block cipher* menjadi *stream cipher*. Ciphertext didapatkan dari hasil operasi XOR antara *plaintext* dan hasil enkripsi dari tahap sebelumnya. Pada round pertama, digunakan initialization Vector (IV).

5. Mode counter

Dalam mode ini, setiap pengirim dan penerima harus dapat mengakses sebuah *counter* yang *reliable,* di mana nilai dari *counter* tersebut akan diakses setiap *ciphertext* blok dipertukarkan. Nilai *counter* ini tidak harus rahasia.

B. Prinsip Diffusion dan Confussion Shannon

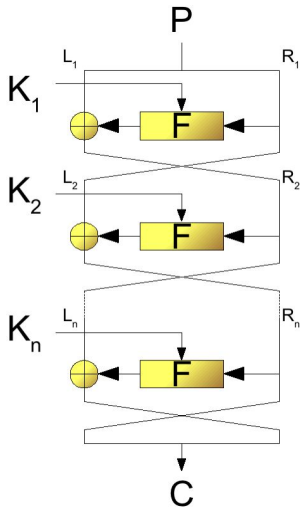
Prinsip diffusion dan confussion merupakan dua properti dari cipher yang aman, sebagaimana dituliskan oleh Claude Shannon pada tahun 1945 dalam bukunya yang berjudul *A Mathematical Theory of Cryptography*.

Confussion berarti setiap bit, byte, atau digit dari plaintext harus bergantung pada beberapa bagian dari kunci, untuk menghilangkan hubungan antara keduanya

Diffusion berarti jika kita mengubah satu bagian dari *plaintext*, maka secara statistik setengah dari *ciphertext* akan berubah dari sebelumnya.

C. Jaringan Feistel

Jaringan Feistel merupakan struktur simetris yang digunakan dalam *block cipher*. Setiap blok yang akan dioperasikan akan dibagi menjadi dua sub-blok yang sama panjang. Sub-blok yang kiri akan menjadi bagian kanan dari blok yang baru. Sub-blok yang kanan akan di-XOR-kan dengan hasil round function antara sub-blok yang kanan dan kunci.



Gambar 1. Struktur Feistel

III. RANCANGAN BLOK CIPHER

Algoritma ini akan membagi plaintext menjadi blok-blok berukuran 128 bit, dan kunci eksternal yang juga berukuran 128 bit.

A. Pembangkitan Kunci Internal

Untuk membangkitkan kunci setiap *round* ke-i, kunci eksternal digeser (shift) ke kiri sebanyak i kali. Kemudian kunci dipecah menjadi dua bagian, masing-masing beukuran 64 bit. Kemudian lakukan operasi XOR pada keduanya. Maka didapatkan kunci internal yang berukuran 64 bit.

B. Round function

Dalam struktur Feistel setiap blok dibagi menjadi dua bagian, anggap saja Li dan Ri. Kemudian lakukan pengoperasian sebagai berikut:

$L_{i+1} = R_i$

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

Algoritma yang digunakan sebagai round function adalah sebagai berikut:

1. Lakukan permutasi terhadap Ri, dengan mengacu pada tabel permutasi berikut:

53	23	60	4	36	42	61	3
21	62	50	45	56	6	52	40
55	20	8	48	1	63	49	26
46	7	24	38	17	15	58	39
64	34	14	5	57	11	41	18
13	29	30	35	10	22	31	2
43	33	32	44	54	19	28	47
16	59	12	9	51	27	25	37

Tabel 1.Tabel Permutasi

2. Geser R_i ke kanan sejauh i bit.
3. Lakukan operasi modulo 2^{64} antara R_i dan K_i
4. Hasil dari tahap tiga dibagi menjadi 8 bagian: H_1, H_2, H_3, H_4, H_5, H_6, H_7, dan H_8. Lakukan pengurutan berdasarkan dengan cara membagi H_i menjadi 8 bagian, dan hitung jumlah bit bernilai 1 pada setiap bagian.

   Contoh: Misalkan H = 10100001 10101010 10000001 11010011 01010000 00101000 10100001 00111011. Bagi menjadi 8 bagian, dan hitung frekuensi bit 1 pada setiap bagian:

i	Bagian ke-i	Total bit 1
1	10100001	3
2	10101010	4
3	10000001	2
4	11010011	5
5	01010000	2
6	00101000	2
7	10100001	3
8	00111011	5

Tabel 2. Frekuensi bit 1 pada setiap bagian

Kemudian urutkan berdasarkan jumlah total bit 1 pada potongan kunci. Karena bagian 4 mepunyai bit 1 terbanyak, maka letakkan H_4 pada urutan pertama. Kemudian letakkan H_8 pada urutan kedua, karena mempunyai bit 1 sebanyak 5. Jika terdapat bagian yang dengan jumlah bit 1 yang sama, diutamakan untuk i yang lebih kecil. Hasil pengurutan adalah sebagai berikut: H_4, H_8, H_2, H_1, H_7, H_3, H_5, H_6. Hasil pertukaran digabungkan menjadi 11010011 00111011 10101010 10100001 10100001 10000001 01010000 00101000.

5. Letakkan setiap bit R_i dalam matriks berukuran 8x8, dan lakukan rotasi ke kanan.

$k_{1,1}$	$k_{1,2}$	$k_{1,3}$	$k_{1,4}$	$k_{1,5}$	$k_{1,6}$	$k_{1,7}$	$k_{1,8}$
$k_{2,1}$	$k_{2,2}$	$k_{2,3}$	$k_{2,4}$	$k_{2,5}$	$k_{2,6}$	$k_{2,7}$	$k_{2,8}$
$k_{3,1}$	$k_{3,2}$	$k_{3,3}$	$k_{3,4}$	$k_{3,5}$	$k_{3,6}$	$k_{3,7}$	$k_{3,8}$
$k_{4,1}$	$k_{4,2}$	$k_{4,3}$	$k_{4,4}$	$k_{4,5}$	$k_{4,6}$	$k_{4,7}$	$k_{4,8}$
$k_{5,1}$	$k_{5,2}$	$k_{5,3}$	$k_{5,4}$	$k_{5,5}$	$k_{5,6}$	$k_{5,7}$	$k_{5,8}$
$k_{6,1}$	$k_{6,2}$	$k_{6,3}$	$k_{6,4}$	$k_{6,5}$	$k_{6,6}$	$k_{6,7}$	$k_{6,8}$
$k_{7,1}$	$k_{7,2}$	$k_{7,3}$	$k_{7,4}$	$k_{7,5}$	$k_{7,6}$	$k_{7,7}$	$k_{7,8}$
$k_{8,1}$	$k_{8,2}$	$k_{8,3}$	$k_{8,4}$	$k_{8,5}$	$k_{8,6}$	$k_{8,7}$	$k_{8,8}$

Tabel 4a. Matriks awal

$k_{8,1}$	$k_{7,1}$	$k_{6,1}$	$k_{5,1}$	$k_{4,1}$	$k_{3,1}$	$k_{2,1}$	$k_{1,1}$
$k_{8,2}$	$k_{7,2}$	$k_{6,2}$	$k_{5,2}$	$k_{4,2}$	$k_{3,2}$	$k_{2,2}$	$k_{1,2}$
$k_{8,3}$	$k_{7,3}$	$k_{6,3}$	$k_{5,3}$	$k_{4,3}$	$k_{3,3}$	$k_{2,3}$	$k_{1,3}$
$k_{8,4}$	$k_{7,4}$	$k_{6,4}$	$k_{5,4}$	$k_{4,4}$	$k_{3,4}$	$k_{2,4}$	$k_{1,4}$
$k_{8,5}$	$k_{7,5}$	$k_{6,5}$	$k_{5,5}$	$k_{4,5}$	$k_{3,5}$	$k_{2,5}$	$k_{1,5}$
$k_{8,6}$	$k_{7,6}$	$k_{6,6}$	$k_{5,6}$	$k_{4,6}$	$k_{3,6}$	$k_{2,6}$	$k_{1,6}$
$k_{8,7}$	$k_{7,7}$	$k_{6,7}$	$k_{5,7}$	$k_{4,7}$	$k_{3,7}$	$k_{2,7}$	$k_{1,7}$
$k_{8,8}$	$k_{7,7}$	$k_{6,8}$	$k_{5,8}$	$k_{4,8}$	$k_{3,8}$	$k_{2,8}$	$k_{1,8}$

Tabel 4b. Matriks sesudah dirotasi kekanan

6. Transformasi matriks menjadi string kembali
7. Lakukan substitusi dengan S-Box. Berikut adalah S-Box yang digunakan di sini:

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

Tabel 5. Kotak S-Box

IV. PENGUJIAN

Berikut merupakan pengujian menggunakan kunci "informatika2014". Plainteks yang digunakan sebagai berikut:

LONDON — As she probes for Russia's vulnerabilities during a week of deepening crisis, Prime Minister Theresa May does not have to look far. Within a few blocks of No. 10 Downing Street, she could find opulent homes owned by members of President Vladimir V. Putin's inner circle.A short walk from Mrs. May's office is an apartment registered to a company owned by First Deputy Prime Minister Igor I. Shuvalov, with a value estimated at $16 million. Roman Abramovich, a former member of the Russian Parliament and a longtime Putin associate, lives opposite Kensington Palace, in a house whose value has been estimated at $163 million. The rupture between Russia and the United Kingdom deepened on Thursday, eleven days after a former Russian spy was poisoned with a military-grade nerve agent in a sleepy town in southwest England. In a rare joint statement, the United States, Germany and France condemned the attack,

calling it the first offensive use of a nerve agent in Europe since World War II.

Cipherteks menggunakan mode ECB dalam representasi base64

UFV9ZStiXraow3VwC2ZlVl0QOiRATCpTSidWDnJoMjozP
Xra+YoBLD4oYiQCGVkGN1xvJn9lMXASIHcABj9VO0Ap
NWN7KnBnDAZOIyI7EFpeC19aCDseVxsNDDVeHU45Bh
BEW24TCGEAZGoLVUN5W2FUBWFzYREVJUY4O1sSI3
VBA10sOC8BDm1weExMU0cZEgtbOGdlMzdRdnt5fh0OV
Wc9BkY5H1tIckYQFlA1HhENVAQuFGVgQVAWHUImPll
XEHF0DS4OaGBQNyNtJCBMEW0RWR8mUBRyFTEydD
UhRBJEV3hTXU5LL1Zeaz5OCiQ1DldPP0kgaUZ8HWpUdg
cbSm1nZHokAGV9ZiFR85eXXlwvdxgODiRbEkQ3am8bZD
ZLelVzWyobOCADX3QhNS99DRR/UBcxQDbSzuh2cUUfP
kBzBDcLQH9nSipfMSwyRUtZDCNYOGZGPXJqQB5gfxIh
YBw+IzoSYVMlHEZzGF53VxoUEw1RVlMgGFoiMDByfg
Z6PkYqQl9zOn4nEX4qAWw8SBUdMStWACRlIRoLFmZ9
VXA2QmdBFyc/S3whMlRzAhwrLXZBQEtNBU0uVmlTBT
R6YR1NClIUQjU/GCEmIFAIRAJyIxsdS104YDQDNgFSDg
MvZ2UnFTxTN21nWVEaK3ILICM1SHNhKBEkIkg1fX1+R
lBLZG5WOxEnP2ReYW8RGkAtQyVgAhw5ZTRAIxx+LD
QFIkEyKWwnHzVYJ2khWW0Icy57XmA0IxB5DREIelcuS3
sFY3V/IUdyGUVQSCAnXnVVdlk+AURTF2VIAnADHQkn
ND8ZHDMcTGdhOAEPWA8tGjJodmkPYUEFRGcWM20fc
j8bNSx9an1RWmQ1AmEoAm1gczEuLgJ2XkREaXUsJmZY
Phs2UFpnUTJ0ICIcawRFQ2V0EShNMG1UaD4OH3psSjYb
AjgUbzBuP1oyPWJ2DCwEYXYpQjteKmoTCEoAMlQ9LhZ
baTJfaHtWPX8IC1wKDBQsdgkcUz8VaApYUW98UiljZEIr
OGEwdXAxOToZQV1oCiZBFyEuQ3w5PBRiUxBBCSVxeE
UVVVJjSH98XF0BE0BuXWFnBGhdV34XEg96VnYCOxZ8
YkUbPQw9dCpwSGYsHDUrAV5WLy0bGRdrd3hCL3sOK
BVaJj1RRgsaaVcZAwYOH1BgayIzcXo1b3ZIWxMrcDQ6Y
HliHWQYYzFsPk9QSys/YXpfRBUNCRB3DURcbWBLRg
U1ERR+dAlCYBkrNGN4WAFDfG5vIXgCMhcaWSVqJlVS
emZmfDADbTkEdTJ6H0B7EkVDX3BGa3QxIR9XbWxmW
EsTdS01XRBnCHZ2AT1eCU5ORWxt

V. ANALISIS KEAMANAN

Algoritma block cipher ini menerapkan properti confusion dan diffusion dengan memanfaatkan Block Cipher yang dimodifikasi dengan menggunakan Jaringan Feistel dan substitusi S-Box. Dengan adanya properti tersebut, diharapkan berbagai serangan yang dilakukan kriptanalis dapat diatasi dan dicegah. Dengan mengamati hasil enkripsi dan metode yang digunakan, dapat dianalisis pengaruh algoritma terhadap keamanan data. Dalam pembahasan berikutnya akan dijabarkan analisis keamanan algoritma dengan masing-masing jenis serangan yang mungkin terjadi.

1. Brute Force Attack

Brute force attack merupakan metode yang dilakukan penyerang dengan menebak dan mencoba kunci satu persatu hingga didapatkan hasil yang diinginkan. Brute force attack menggunakan metode exhaustive search, dimana penyerang mengenumerasi satu persatu kemungkinan. Dengan menggunakan metode ini, kunci kemungkinan besar akan ditemukan, namun waktu yang dibutuhkan bisa sangat lama, terutama jika kunci sangat panjang

Panjang kunci adalah adalah 128 bit. Sehingga kemungkinan kunci adalah $2^{128} \approx 3{,}4 \times 10^{38}$. Jika untuk setiap percobaan dapat dilakukan dalam waktu 1 ms, maka dibutuhkan waktu selama 10^{28} tahun untuk mecoba semua kemungkinannya.

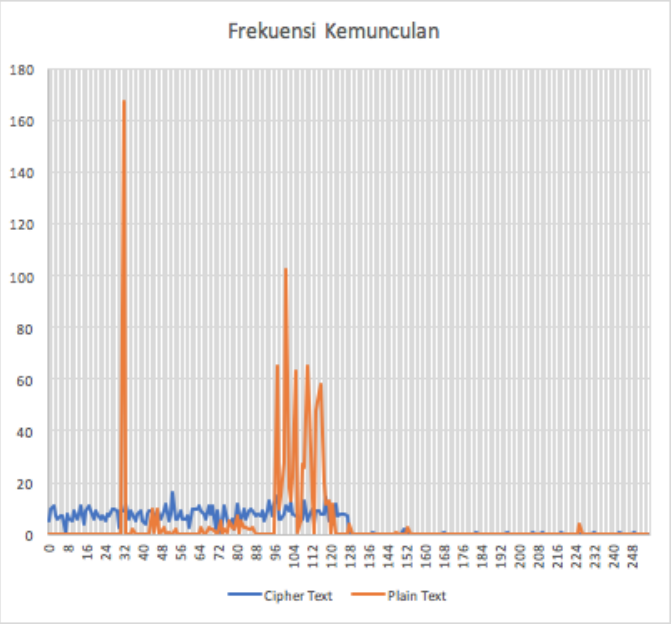
2. Known Plain Text Attack

Serangan dengan jenis ini memerlukan attacker untuk mengetahui pasangan plain teks dan cipher teks. Keamanan block cipher ditentukan oleh struktur utamanya, yaitu skema feistel network beserta modus operasinya dan fungsi round yang terdapat di dalamnya. Bagaimana membangkitkan kunci internal juga menjadi faktor keamanan serangan ini. Dapat dilihat pada modus operasi ECB, blok pada plainteks yang sama akan menjadi blok cipherteks yang sama pula. Dengan demikian, apabila serangan dengan jenis ini memerlukan analisis yang lebih mudah dibandingkan mode-mode lainnya. Keamanan block cipher ditentukan oleh struktur utamanya, yaitu skema feistel network beserta modus operasinya dan fungsi round yang terdapat di dalamnya. Pembangkitan kunci internal juga menjadi faktor keamanan dalam hal ini. Dapat dilihat pada modus operasi ECB, blok pada plainteks yang sama akan menjadi blok cipherteks yang sama pula. Dengan demikian, apabila sebuah blok pada plainteks sudah diketahui hasil enkripsinya, dekripsi algoritma ini pada modus operasi ECB akan sangat cepat. Hal ini dapat diatasi dengan menggunakan modus operasi yang berbasis stream cipher seperti CBC dan CFB.

3. Analisis Frekuensi

Analisis frekuensi merupakan teknik yang digunakan untuk memecahkan cipherteks dengan memperhatikan

frekuensi kemunculan huruf atau bigram ataupun poligram pada cipherteks dan membandingkannya dengan huruf atau bigram atau poligram yang sering muncul pada plainteks. Analisis frekuensi memanfaatkan huruf-huruf yang sering muncul pada suatu bahasa dan dibandingkan dengan huruf-huruf yang sering muncul pada cipher teks, untuk Bahasa Inggris, huruf yang sering muncul adalah E, T, A, dan O, sedangkan huruf yang jarang muncul adalah Z, Q, dan X. Pengetahuan ini digunakan untuk memetakan kemunculan kata pada cipherteks. Untuk menganalisis pengaruh analisis frekuensi dengan keamanan algoritma Vibranium Cipher ini dapat dilihat dari diagram frekuensi kemunculan berikut ini. Diagram frekuensi tersebut merupapakan perbandingan frekuensi plain teks dan cipher teks yang menggunakan mode ECB.



Gambar 2. Diagram Frekuensi Kemunculan

VI. KESIMPULAN DAN SARAN

Dari hasil eksperimen yang telah dilakukan, dapat ditarik kesimpulan bahwa:

1. Algoritma ini telah memenuhi prinsip diffusion dan confussion Shannon, karena dengan perubahan satu karakter pada plaintext, menyebabkan perubahan yang signifikan pada ciphertext
2. Algoritma ini mempunyai kompleksitas yang baik, disebabkan kesulitan dalam melakukan bruteforce terhadap ciphertext.

Saran yang dapat kami berikan sebagai penulis adalah dalam proses pembuatan kunci internal, dapat dilakukan randomisasi untuk shift yang terhadap untuk setiap round, sehingga dapat meningkatkan kompleksitasnya.

DAFTAR PUSTAKA

[1] https://www.block-cipher.com/
[2] https://www.tutorialspoint.com/cryptography/block_cipher.html
[3] http://www.crypto-it.net/eng/theory/modes-of-block-ciphers.html