Algoritma Block Cipher JANE :

Just Another Normal Encryption

Christian Kevin Saputra
13516073
Studi Teknik Informatika
Institut Teknologi Bandung
Bandung, Indonesia
13516073@std.stei.itb.ac.id

Ahmad Faishol Huda
13516094
Studi Teknik Informatika
Institut Teknologi Bandung
Bandung, Indonesia
13516094@std.stei.itb.ac.id

***Abstract*—Algoritma JANE merupakan sebuah algoritma blok cipher kunci simetris dengan blok sebesar 256 bit. Agar sulit untuk didekripsi, algoritma JANE menerapkan prinsip *diffusion* dan *confusion*, pergeseran bit, dan juga cipher berulang menggunakan struktur feistel serta proses substitusi s-box. Untuk mempersulit kriptanalisis, kunci putaran untuk setiap iterasi didapatkan dari kunci pengguna yang telah dimasukkan ke dalam fungsi *hash*.**

Keywords—confusion; diffusion; feistel; s-box; hash; block cipher;

I. Pendahuluan

Kriptografi berasal dari kata *cryptós* yang berarti *secret* (rahasia) dan juga *gráphein* yang berarti *writing* (tulisan) sehingga dapat diartikan sebagai *secret writing* atau tulisan rahasia. Kriptografi ini dapat juga diartikan sebagai ilmu dan seni untuk menjaga keamanan sebuah pesan (Schneier, 1996). Di masa sekarang ini dimana kehidupan manusia terhubung dengan internet dan banyak sekali terjadi pertukaran data, kriptografi memiliki peran yang penting dalam menjaga kerahasiaan, integritas, dan otentikasi data.

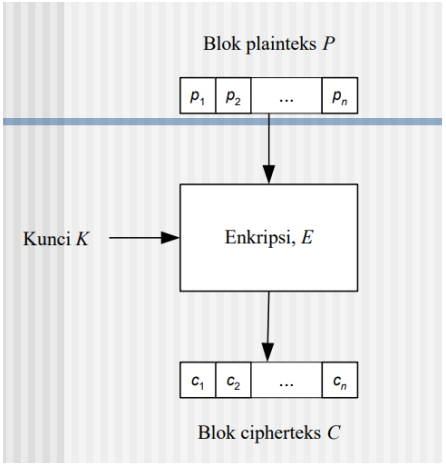
Kriptografi modern yang dilakukan untuk menjaga kerahasiaan data adalah operasi terhadap bit - bit di dalam data. Salah satu kategori algoritma cipher yang beroperasi pada bit adalah *block cipher*. *Block cipher* bekerja dengan membagi data yang akan dienkripsi ke dalam beberapa blok dengan jumlah bit yang sama dan operasi enkripsi atau dekripsi dilakukan pada blok tersebut. Beberapa metode block cipher yang terkenal adalah DES dan AES.

Makalah ini akan menjelaskan sebuah algoritma *block cipher* baru yang dikenal dengan nama JANE. JANE merupakan singkatan dari *just another normal encryption*. Agar JANE tidak mudah untuk dipecahkan dengan metode kriptanalisis, JANE menerapkan prinsip *diffusion* dan *confusion* , pergeseran bit, dan juga cipher berulang menggunakan struktur feistel. Hasil dari feistel tersebut kemudian akan disubstitusi menggunakan s-box 4x4 yang akan terbentuk berdasarkan key dari pengguna. Untuk cipher berulang, kunci untuk setiap putaran didapatkan dari kunci pengguna yang dimasukkan ke dalam fungsi hash.

II. Dasar Teori

A. Algoritma Block Cipher

Block cipher merupakan algoritma kriptografi dengan kunci simetris yang bekerja dengan membagi pesan yang akan dienkripsi ke dalam blok - blok yang memiliki ukuran yang sama besar. (Misalkan cipher blok 64 bit, maka pesan akan dibagi menjadi blok - blok dengan setiap blok berukuran 64 bit). *Block cipher* juga akan menerima kunci dari pengguna yang akan digunakan oleh *block cipher* untuk mengenkripsi / mendekripsi setiap blok pesan. Skema dari blok cipher dapat digambarkan seperti berikut :



Gambar 1. Skema Enkripsi / Dekripsi Block Cipher
(Sumber : Algoritma Kriptografi Modern(2019) oleh Rinaldi M)

Terdapat beberapa operasi yang dapat dilakukan pada *block cipher* yaitu :

1. *Electronic Code Block (ECB)*
   Operasi ECB melakukan enkripsi / dekripsi pada blok tanpa adanya ketergantungan dengan blok lain
2. *Cipher Block Chaining(CBC)*
   Operasi CBC melakukan enkripsi / dekripsi sedimikian mungkin sehingga terjadi ketergantungan terhadap blok sebelumnya.
3. *Cipher Feedback*
   Operasi *cipher feedback* melakukan enkripsi / dekripsi terhadap bit yang lebih kecil daripada ukuran blok.
4. *Output Feedback*
   Operasi *output feedback* melakukan enkripsi / dekripsi dengan menyalin hasil enkripsi menjadi elemen paling kanan.
5. *Counter Mode*
   Operasi dengan menggunakan *counter* yang berbeda setiap bloknya. Operasi *Counter Mode* tidak memiliki ketergantungan seperti pada operasi CBC, *cipher*, atau *output feedback*.

B. *Confusion and Diffusion*

Menurut Claude Shannon, untuk mempersulit proses kriptanalisis pada sebuah algoritma sebuah dibutuhkan prinsip *confusion* dan *diffusion.*

Confusion merupakan cara yang dapat dilakukan untuk menyembunyikan hubungan antara plainteks dengan cipherteks ataupun dengan kunci yang ada. Dengan menerapkan prinsip *confusion* ini, proses kriptanalisis seperti analisis frekuensi akan sulit untuk dilakukan. Prinsip *confusion* ini dapat dicapai dengan melakukan substitusi misalkan substitusi dengan menggunakan s-box.

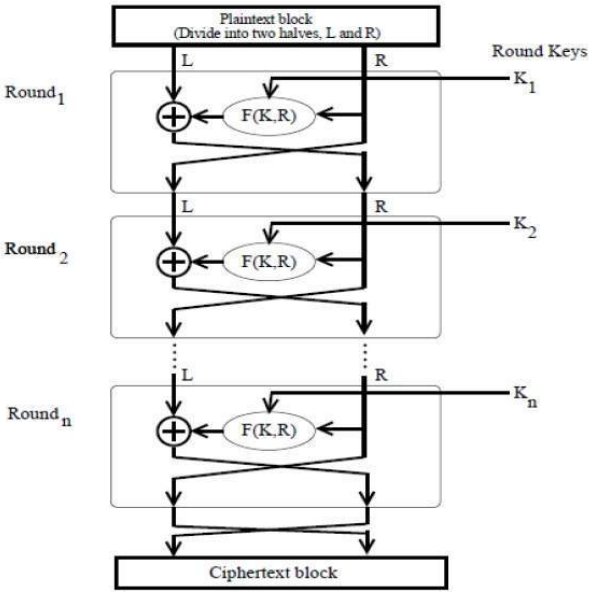
Diffusion merupakan prinsip yang menyatakan bahwa perubahan satu karakter pada plainteks akan menyebabkan cipherteks berubah dan begitu pula perubahan satu karakter pada cipherteks akan menyebabkan plainteks berubah. Prinsip *diffusion* ini dapat diterapkan dengan melakukan transposisi ataupun permutasi.

C. *Jaringan Feistel*

Jaringan feistel merupakan sebuah struktur yang bersifat reversible yang memungkinkan penggunaan satu algoritma dapat digunakan untuk mengenkripsi serta mendekripsi pesan yang ada. Sifatnya yang reversible inilah yang menyebabkan tidak dibutuhkanya algoritma dekripsi yang berbeda dengan algoritma enkripsi. Cara kerja dari jaringan feistel adalah :

1. Setiap blok dibagi 2 menjadi blok kiri dan blok kanan yang memiliki ukuran yang sama besar.
2. Dalam setiap putaran, blok kanan tidak akan berubah sedangkan blok kiri akan berubah melalui sebuah operasi yang bergantu pada kunci dan juga blok kanan.
3. Setelah operasi pada putaran tersebut selesai, blok kanan diubah menjadi blok kiri dan blok kiri diubah menjadi blok kanan.
4. Terdapat beberapa putaran dalam struktur feistel tergantung pada desain algoritma. Biasanya terdapat pula kunci yang berbeda untuk setiap putarannya namun kunci tersebut masih berhubungan dengan kunci utama yang diberikan oleh pengguna.

Gambar dari jaringan feistel dapat dilihat pada gambar berikut:



Gambar 2 Jaringan Feistel (Sumber:https://www.tutorialspoint.com/cryptography/feistel_block_cipher.htm)

D. Substitution Box (S-box)

Substitution box atau *s-box* adalah sebuah *lookup table* yang digunakan untuk menyembunyikan hubungan antara plainteks dengan cipherteks. S-box memiliku ukuran m x n dimana m merupakan ukuran bit input dan n merupakan ukuran bit output. Contohnya adalah s-box pada algoritma DES yang berukuran 6x4, maka 6 bit input dan output dari s-box adalah 4 bit. Contoh s-box :

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Gambar 3. AES S-box (Sumber : http://home.sandiego.edu/~cparker/old_classes/crypto_su17/aes_sbox.jpg)

E. Fungsi Hash

Fungsi hash merupakan sebuah fungsi yang dapat menerima input sebuah pesan baik panjang maupun pendek dan menghasilkan sebuah pesan lain dalam ukuran yang tetap. Hash sering digunakan untuk keamanan ataupun mempermudah mengingat data yang besar seperti pada bitcoin. Salah satu contoh fungsi hash adalah :

1. MD5
2. SHA-256
3. BLAKE2

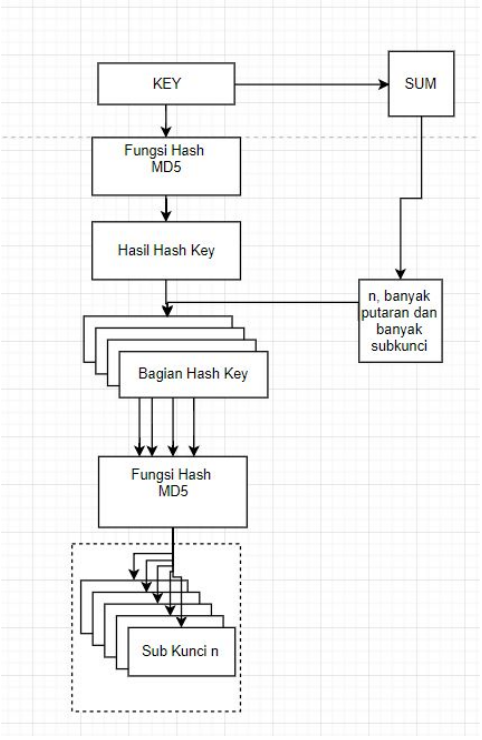
III. Rancangan Blok Cipher

Algoritma blok cipher JANE adalah sebuah algoritma blok cipher yang memanfaatkan jaringan Feistel. Besar blok plaintext/pesan adalah 256-bit atau 32 karakter. Algoritma ini juga memerlukan input kunci yang panjangnya dibebaskan. Panjang kunci tidak berpengaruh karena kunci tersebut nantinya akan dimasukkan ke dalam fungsi hashing. Penggunaan fungsi hashing ditujukan untuk mempermudah kunci yang bisa digunakan tanpa berpengaruh pada kekuatan dan proses algoritma ini sendiri.

Jaringan feistel pada algoritma ini melakukan putaran sebanyak 8*n kali dimana n ditentukan dari kunci yang dimasukkan. Algoritma ini dimulai dengan pembangkitan upakunci dari kunci yang dimasukkan beserta penentuan banyak putaran dari kunci tersebut (n). setelah itu blok pesan akan dibagi menjadi 2 bagian, masing-masing 128 bit, dilakukan pergeseran bitwise terhadap masing-masing bagian. Setelah itu digabungkan kembali lalu dilanjutkan dengan jaringan feistel. Pada jaringan feistel akan dilakukan permutasi ataupun operasi XOR dengan upakunci secara bergantian. Dilanjutkan dengan substitusi S-box sebelum akhirnya menjadi blok ciphertext.

A. Pembangkitan Upakunci

Jaringan feistel memerlukan kunci berbeda untuk setiap putarannya, untuk itu diperlukan upakunci yang akan dibangkitkan dari kunci internal yang telah diberikan oleh pengguna. Banyak putaran jaringan feistel ditentukan dengan menjumlahkan kunci internal setelah dikonversi ke dalam integer, banyak putaran = SUM(INT(key)). Upakunci akan dibangkitkan dengan memasukkan kunci internal kedalam sebuah fungsi hashing, fungsi hashing yang dipakai adalah fungsi hash MD5-128bit, setelah itu hasilnya akan dibagi sesuai banyak putaran yang diperlukan, lalu setiap bagiannya akan dimasukkan ke dalam fungsi hash lagi untuk mendapatkan upakunci yang diperlukan.

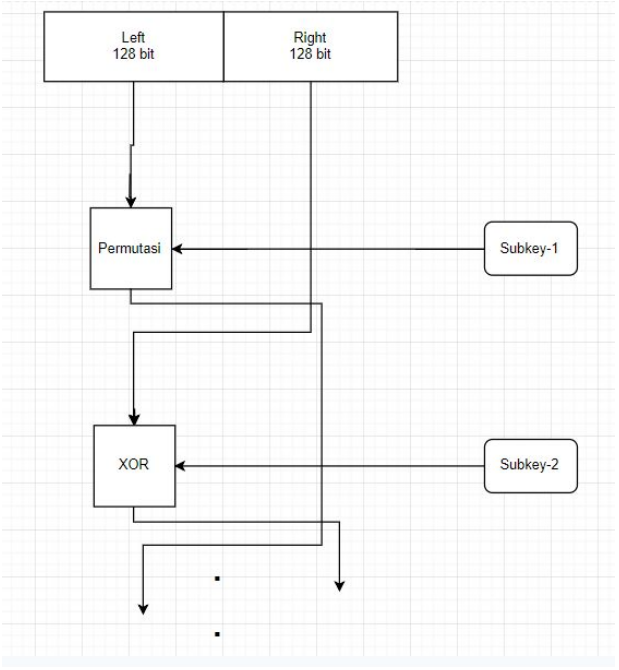


B. Fungsi pergeseran

Sebelum blok masuk ke dalam jaringan feistel maka akan dilakukan pergeseran bitwise terhadap blok tersebut. pergeseran akan dilakukan dengan arah kiri sebanyak banyak putaran jaringan feistel.

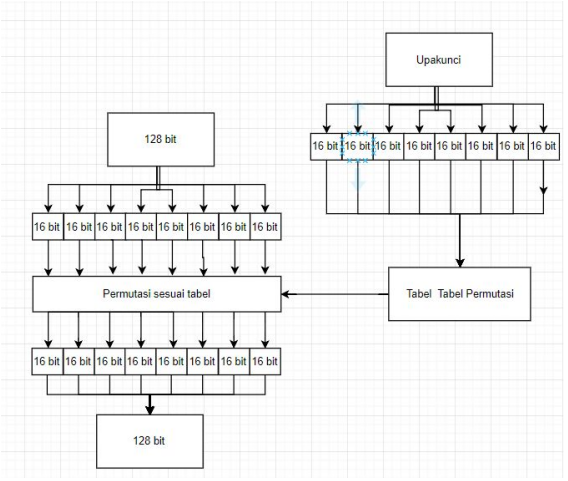
C. Fungsi Putaran Jaringan feistel

Salah satu bagian terpenting dalam algoritma blok cipher yang memanfaatkan jaringan feistel adalah fungsi putaran ini. Pada fungsi putaran di algoritma ini blok akan dibagi menjadi 2 bagian terlebih dahulu, masing masing 128 bit, yaitu bagian kiri dan kanan. Dilanjutkan dengan putaran dimana setiap putarannya akan dilakukan permutasi ataupun operasi XOR secara bergantian pada bagian kiri, lalu pada putaran selanjutnya dilakukan pertukaran yaitu bagian kiri menjadi bagian kanan dan sebaliknya. setelah melewati fungsi putaran kedua bagian tersebut akan disatukan kembali.



i. Permutasi

Pada proses permutasi, pertama adalah membagi bagian tersebut menjadi masing masing 16 bit. setelah itu bangkitkan tabel permutasi untuk melakukan pertukaran elemen. Tabel permutasi dibangkitkan secara dinamis tergantung dengan upakunci pasangan blok tersebut. Setelah itu dilakukan pertukaran elemen berdasarkan tabel permutasi tersebut



ii. XOR

Pada proses XOR blok akan di XOR kan dengan upakunci yang sesuai dengan putaran tersebut.

D. Transposisi

Setelah diproses oleh jaringan feistel, hasilnya akan di transpose sebelum memasuki S-box. Transposisi dilakukan dengan mengubah 256 bit hasil jaringan feisel menjadi sebuah matriks berukuran 16x16. Transposisi dilakukan pada matriks tersebut (baris pertama menjadi kolom pertama, baris kedua menjadi kolom kedua, dan seterusnya) kemudian pembacaan bit dilakukan dari elemen pertama (elemen pojok kiri atas) ke kanan hingga elemen terakhir pada baris tersebut lalu pindah ke baris selanjutnya.

E. S-Box

S-box yang digunakan pada algoritma JANE adalah S-box dengan ukuran 4x4. S-box di-generate berdasarkan karakter pada kunci pengguna. Sebelum dimasukkan ke dalam s-box, pesan dibagi ke dalam blok - blok sebesar 4 bit. Bit pertama dan bit terakhir akan digunakan menjadi baris dan bit kedua serta bit ketiga akan digunakan untuk menjadi kolom. Hasil yang didapat dari pencarian pada baris dan kolom pada s-box akan mengganti blok tersebut.

Contoh kunci adalah “kriptografi”, maka s-box untuk blok pertama akan di-generate dengan menggunakan karakter ‘k’ sebagai seed, blok kedua dengan ‘r’ sebagai seed, dan seterusnya. Apabila kunci lebih pendek, maka setelah karakter terakhir pada kunci, proses dimulai kembali dari awal kunci.

Misalkan blok 1 dengan 4 bit yaitu 1001, maka bit pertama dan bit terakhir yaitu 11 yang berarti 3 akan menjadi baris dan 00 yang berarti 0 akan menjadi kolom. Maka pencarian pada s-box dilakukan pada baris ke 3 dan kolom ke 0. Misalkan hasil yang didapat adalah 8, maka blok 1 yang tadinya bernilai 1001 akan diubah menjadi 1000 (bit dari angka 8).

IV. Pengujian dan Analisis

A. *Pengujian*

Akan dilakukan beberapa pengujian dan analisis terhadap algoritma blok cipher ini. Algoritma ini akan jalankan dengan beberapa metode :

1. Metode EBC

Kunci	kriptografi
Plain	INI PLAINTEXTS BLOCKCIPHER JANE JUST ANOTHER NORMAL ENCRYPTION USING FEISTEL NETWORK AND SBOX SUBSTITUSION WITH SOME TWIST IN IT
Cipher	E>ßˆ¥e [¿8‰‘oòçêÑwR¸Œ?iF˜…¿‘HŠ+e(¡¥peW ý1rÑÃ·›H-àõ†)d NŽR-æ!áþ‘à©{(!ò; Ñ,wò¸-ìur:lÍ™ªRwµÇÊ˜ø 0§¥ÑÌçßh üâQˆ

Kasus	**Kunci diganti**
Kunci	KRIPTOGRAFI
Plain	INI PLAINTEXTS BLOCKCIPHER JANE JUST ANOTHER NORMAL ENCRYPTION USING FEISTEL NETWORK AND SBOX SUBSTITUSION WITH SOME TWIST IN IT
Cipher	Ï>P×x¶õP ±¥VÁq¤¹þ °OÒÀOï7Âüqu§‘PÈñn%aËc5 3¯äP³,ßíhˆ®Ðz"ÎP»‘Õî”h V8,@tê” ¼€O×‘÷MË‹•P¿x õéBaVî€Ç²´£ƒ”P\…Óožª

Kasus	**1 bit plainteks diubah**
Kunci	kriptografi
Plain	INI **L**LAINTEXTS BLOCKCIPHER JANE JUST ANOTHER NORMAL ENCRYPTION USING FEISTEL NETWORK AND SBOX SUBSTITUSION WITH SOME TWIST IN IT
Cipher	E>ßˆ¥a [¿8‰‘oòçêÑw¸Œ5iF˜…¿‘HŠ+e(¡¥pe Wý1rÑÃ·›H-àõ†)d NŽR-æ!áþ‘à©{(!ò; Ñ,wò¸-ìur:lÍ™ªRwµÇÊ˜ø 0§¥ÑÌçßh üâQˆ

Pada metode EBC, tidak ada ketergantungan antara 1 blok dengan blok yang lain. Plainteks terdiri atas 1024 bit sehingga akan dibagi menjadi 4 blok sebesar 256 bit. Apabila terjadi perubahan 1 bit pada pesan, akan terjadi perubahan pada cipherteks seperti yang di*highlight* warna kuning. Perubahan kunci juga menyebabkan cipherteks yang dihasilkan sangat berbeda.

2. Metode CBC

Kunci	kriptografi
Plain	INI PLAINTEXTS BLOCKCIPHER JANE JUST ANOTHER NORMAL ENCRYPTION USING FEISTEL NETWORK AND SBOX SUBSTITUSION WITH SOME TWIST IN IT
Cipher	E>ßˆ¥e [¿8‰‘oòçêÑwR¸Œ?iF˜…ú¯— ; (ú ù}Æ’ïÕ˜ ÛÀÉð lÊhÀ±áª1 W*ƒ* @ìEÙÜï x AÀið €ôØ9ä ZS¬ü9[xQp9Už N-Ð a‰›kä

Kasus	**Kunci diubah**
Kunci	KRIPTOGRAFI
Plain	INI PLAINTEXTS BLOCKCIPHER JANE JUST ANOTHER NORMAL ENCRYPTION USING FEISTEL NETWORK AND SBOX SUBSTITUSION WITH SOME TWIST IN IT
Cipher	Ï>P×x¶õP ±¥VÁq¤¹þ °OÒÀOï7Âülº°¯ ‰¬›E%Ðn@¢DŠQ ãàRaB_JR¡…_ s`Ï€± Â 5"1±ïv épð°:ùŽY±©[éuðŠ*h@¸—ÿ0ô× i ð àâþ˜]

Kasus	**1 bit plainteks diubah**
Kunci	kriptografi
Plain	INI **L**LAINTEXTS BLOCKCIPHER JANE JUST ANOTHER NORMAL ENCRYPTION USING FEISTEL NETWORK AND SBOX SUBSTITUSION WITH SOME TWIST IN IT
Cipher	E>□Őa [8□ɘγ□w_□iF·:· ˜□□); □□ □ Úlh1 WÉ@씵□□ x Ai□??9□ ZS□9[xQp9U➚ NƏPaɛ𝕏

Pada metode CBC, terdapat hubungan antara 1 blok dengan blok yang lain. Hasil dari enkripsi pada blok sebelumnya akan digunakan oleh blok berikutnya untuk melakukan operasi XOR terhadap cipherteks sehingga menimbulkan ketergantungan antara 1 blok dengan blok lainnya. Pada kasus kunci diganti, cipherteks memiliki awalan yang mirip dengan cipherteks pada kasus kunci diganti pada metode EBC. Namun karena blok pada CBC saling bergantung, cipherteks yang dihasilkan jadi berbeda dari EBC.

Pada kasus satu bit plainteks diubah, hasil cipherteks yang dihasilkan sangat berbeda dari cipherteks pada plainteks asli dikarenakan terdapat blok yang berubah sehingga mempengaruhi blok - blok selanjutnya dan membuat cipherteks berbeda.

3. Metode Counter

Kunci	kriptografi
Plain	INI PLAINTEXTS BLOCKCIPHER JANE JUST ANOTHER NORMAL ENCRYPTION USING FEISTEL NETWORK AND SBOX SUBSTITUSION WITH SOME TWIST IN IT
Cipher	E>ßˆ¥e [¿8‰‘oòçêÑwR¸Œ?iF˜…¿‘HŠ+e(¡¥pe Wý1rÑÃ·›H-àõ†)e NŽR-æ!áþ‘â©{(!ò; Ñ,wò¸-ìur:nÍ™ªRwµÇÊ˜ø 0§¥ÑÌçßh üâQ‹

Kasus	**Kunci diubah**
Kunci	KRIPTOGRAFI
Plain	INI PLAINTEXTS BLOCKCIPHER JANE JUST ANOTHER NORMAL ENCRYPTION USING FEISTEL NETWORK AND SBOX SUBSTITUSION WITH SOME TWIST IN IT
Cipher	Ï>P×x¶õP ±¥VÁq¤¹þ °OÒÀOï7Âüqu§‘PÈñn%aËc5 3¯äP³,ßíhˆ¯Ðz"ÎP»‘Õî”h V8,@tê” ¼€O×‘õMË‹•P¿x õéBaVî€Ç²´£ƒ”P\…Óož©

Kasus	**1 bit plainteks diubah**
Kunci	kriptografi
Plain	INI **L**LAINTEXTS BLOCKCIPHER JANE JUST ANOTHER NORMAL ENCRYPTION USING FEISTEL NETWORK AND SBOX SUBSTITUSION WITH SOME TWIST IN

	IThttps://docs.google.com/document/d/1V5iXKRJIiU83UOpjQqsG2j3AafbRt5cQT2-JwoAtB9w/edit#
Cipher	E>ßˆ¥a [¿8‰‘oòçêÑw_‚Œ5iF˜…¿‘HŠ+e(¡¥peWý1rÑÃ·›H-àõ†)e NŽR-æ!áþ‘à©{(!ò;Ñ‚wò¸-ìur:nÍ™ªRwµÇÊ˜ø 0§¥ÑÌçßh üâQ‹

Pada metode counter, tidak ada hubungan antar blok seperti pada metode EBC. Namun, pada setiap blok terdapat sebuah counter . Counter ini digunakan setelah substitusi S-box dengan melakukan operasi XOR pada cipherteks dengan counter.

B. *Analisis Keamanan*

Algoritma Jane menerapkan konsep *confusion* and *diffusion* sehingga secara statistik persebaran karakter menjadi lebih merata. Hubungan antara kunci juga sangat kompleks karena menggunakan fungsi hashing dan setiap upakunci dibangkitkan secara dinamik. Tidak hanya itu setiap fungsi pada jaringan feistel bergantung kepada kunci yang dimasukkan oleh user, entah itu untuk menentukan tabel permutasi atau pada operasi XOR untuk upakunci.

1. Serangan *Brute force*

Serangan yang paling sering digunakan oleh kriptanalis untuk memecahkan ciphertext adalah dengan mencoba semua kemungkinan jenis kemungkinan kunci untuk melakukan dekripsi terhadap ciphertext. Serangan ini juga disebut dengan serangan brute force atau exhaustive search. Serangan ini bisa mencari kunci yang digunakan untuk algoritma enkripsi.

Algoritma Jane menggunakan panjang kunci yang tidak beraturan(tidak spesifik panjangnya) hal ini akan mempersulit penebakan. kunci tersebut nantinya akan menjadi sebuah kunci internal 128 bit dengan fungsi hashing. setelah itu akan di bangkitkan upakunci eksternal sebanyak 8* banyak round. untuk itu minimal terdapat $8*2^{128}$ atau 10^{39} atau $2{,}722589353675077077069968594541e^{39}$. Jika sebuah komputer bisa mencoba sebanyak 10 juta kunci per detik maka akan memerlukan 10^{32} detik atau $3{,}17 * 10^{25}$ tahun.

2. Analisis Frekuensi

Perbandingan antara ciphertext dan plaintext:

Plaintext:

‗: 18; I: 13; T: 13; N: 12; S: 10; E: 9; O: 8; A: 5; R: 5; L: 4; U: 4; P: 3; B: 3; C: 3 ; H: 3; W: 3; X: 2; K: 2 J: 2; M: 2; Y: 1; G: 1; F: 1

Ciphertext :

Ñ: 4 : 4 ↲: 4 ¥: 3 e: 3 ‘: 3 ò: 3 w: 3; R: 3 -: 3; ß: 2 ‗: 2 ¿: 2; : 2; ç: 2; ¸: 2; ˜: 2; H: 2 (: 2 r: 2 à: 2 !: 2 0: 1 1: 1 8: 1 E: 1 >: 1 ˆ: 1 [: 1 ‰: 1 o: 1 ê: 1 Œ: 1 ?: 1 i: 1 F: 1 …: 1 Š: 1 +: 1 ¡: 1 p: 1 W: 1 ý: 1 Ã: 1 ·: 1 ›: 1 õ: 1 †: 1): 1 d: 1 N: 1 Ž: 1 æ: 1 á: 1 þ: 1 ©: 1 {: 1 ;: 1 ,: 1 ì: 1 u: 1 :: 1 l: 1 Í: 1 ™: 1 ª: 1 : 1 µ: 1 Ç: 1 Ê: 1 ø: 1 §: 1 Ì: 1 h: 1 ü: 1 â: 1 Q: 1 Ñ: 4 : 4 ↲: 4 ¥: 3 e: 3 ‘: 3 ò: 3 w: 3 R: 3 -: 3 ß: 2 ‗: 2 ¿: 2 : 2 ç: 2 ¸: 2 ˜: 2 H: 2 (: 2 r: 2 à: 2 !: 2 0: 1 1: 1 8: 1 E: 1 >: 1 ˆ: 1 [: 1 ‰: 1 o: 1 ê: 1 Œ: 1?: 1i: 1F: 1…: 1Š: 1+: 1¡: 1p: 1W: 1ý: 1Ã: 1·: 1›: 1õ: 1†: 1): 1d: 1N: 1Ž: 1æ: 1á: 1þ: 1©: 1{: 1 ;: 1 ,: 1 ì: 1 u: 1 :: 1 l: 1 Í: 1 ™: 1 ª: 1 : 1 µ: 1 Ç: 1 Ê: 1 ø: 1 §: 1 Ì: 1 h: 1 ü: 1 â: 1 Q: 1 D: 1

Terlihat bahwa pada ciphertext persebaran nya sudah cukup merata. Persebaran yang merata akan mempersulit kriptanalisis untuk melakukan analisis frekuensi untuk memecahkan kunci pada algoritma ini.

V. Kesimpulan dan Saran

Algoritma JANE merupakan sebuah algoritma *block cipher* berukuran 256 bit yang menerapkan prinsip *confusion* dan *diffusion* untuk mempersulit kriptanalisis. Selain itu, JANE juga menerapkan jaringan feistel dengan 8 * n putaran dan penggunaan s-box 4x4 setelah selesai dari jaringan feistel. Kunci untuk setiap putaran didapatkan dari fungsi hash yang dilakukan terhadap kunci awal yang dimasukkan oleh pengguna. Terdapat 3 operasi yang dapat dilakukan dengan *block cipher* JANE yaitu EBC, CBC, dan counter. Dengan penerapannya yang dibuat untuk mempersulit kriptanalisis, algoritma JANE dapat digunakan sebagai salah satu algoritma alternatif.

Pengembangan dapat dilakukan terhadap algoritma JANE dengan memperkecil ukuran tiap blok agar lebih banyak perubahan yang terjadi pada setiap bloknya.

References

[1] R. Munir, Slide Kuliah IF4020 Kriptografi, Pengantar Kriptografi, 2019.

[2] R. Munir, Slide Kuliah IF4020 Kriptografi, Algoritma Kriptografi Modern, 2019.

[3] Wade Trappe and Lawrence C. Washington, Introduction to Cryptography and Coding Theory.

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

Christian Kevin Saputra
13516073

Ahmad Faishol Huda
13516094