Algoritma *Spiral shifting*

Algoritma Gabungan Feistel Network dan Rijndael dengan Transformasi *Spiral shifting* dan Dependent SubBytes

Muhammad Harits Shalahuddin Adil Haqqi Elfahmi
Sekolah Teknik Elektro dan Informatika (STEI)
Institut Teknologi Bandung, ITB
Bandung, Indonesia
13511046@std.stei.itb.ac.id

Dinah Kamilah Ulfa
Sekolah Teknik Elektro dan Informatika (STEI)
Institut Teknologi Bandung, ITB
Bandung, Indonesia
13511087@std.stei.itb.ac.id

***Abstract*— Makalah ini mengajukan algoritma block cipher baru yang didasarkan pada algoritma Rijndael dan Feistel Network. Algoritma ini menggunakan Feistel Network dengan fungsi berupa proses-proses round pada algoritma rijndael. Pada proses round tersebut sendiri, digunakan modifikasi berupa shifting spiral serta dependent SubBytes untuk lebih jauh mengaburkan hubungan antara urutan byte data pada plainteks dan hasil cipherteks.**

Keywords—block cipher, shifting spiral, feistel network, rijnael, AES.

I. PENDAHULUAN

Pada saat ini, seiring dengan semakin umumnya penggunaan teknologi informasi digital seperti internet, tindak kriminal melalui media ini juga semakin sering terjadi. Contoh dari perbuatan kriminal ini adalah pencurian dan penggunaan data secara ilegal. Karena itu, keamanan informasi merupakan aspek penting dalam teknologi informasi digital berbasis internet. Salah satu ilmu keamanan informasi adalah kriptografi.

Bidang ilmu kriptografi telah banyak menghasilkan algoritma yang melindungi keamanan informasi digital. Contohnya adalah Data Encryption Standard (DES) yang sempat menjadi standar algoritma enkripsi data. Akan tetapi, perkembangan teknologi menyebabkan algoritma enkripsi ini tak lagi aman, dan karena itu standar ini diganti dengan Advanced Encryption Standard (AES) yang masih digunakan sampai saat ini. Selain itu, terdapat pula beberapa algoritma enkripsi lain seperti Serpent, Twofish, RC6, dan MARS.

Dalam makalah ini, kami mengusulkan algoritma cipher yang menggunakan transformasi spiral pada blok cipher yang dienkripsi. Algoritma ini merupakan modifikasi dari algoritma yang telah ada, yaitu algoritma Rijndael yang saat ini digunakan sebagai AES dan Feistel Network.

Makalah ini disusun sebagai berikut. Pada Bagian 2, kami menjelaskan teori yang menjadi dasar dari penyusunan algoritma kami. Bagian 3 menyajikan menyajikan spesifikasi serta motivasi desain dari algoritma. Bagian 4 menyajikan hasil dari pengujian algoritma dan pembahasannya. Dalam Bagian 5, kami memberikan analisis keamanan dari beberapa tipe serangan yang ada. Bagian 6 berisi kesimpulan makalah ini.

II. DASAR TEORI

A. *Prinsip Desain Block Cipher*

Dalam merancang Block Cipher, ada empat prinsip yang dapat digunakan untuk mendesain, yaitu:

1. *Confusion & Diffusion*

   Prinsip *confusion* artinya menyembukan hubungan apapun antara plainteks, cipherteks dan kunci, seperti hubungan statistik. Sedangkan prinsip *diffusion* artinya meyebarkan pengaruh satu bit plainteks ke sebanyak mungkin cipherteks.

2. Iterated Cipher

   Iterated Cipher merupakan fungsi transformasi sederhana yang mengubah plainteks menjadi cipherteks diulang sejumlah kali. Pada setiap putaran digunakan subkey atau round key yang dikombinasikan dengan plainteks. Cipher berulang dinyatakan sebagai:

   $$Ci = f(Ci\text{-}1, Ki) \quad (1)$$

3. Feistel Network

   Feistel network merupakan struktur simetrik yang digunakan untuk mengkontruksi block cipher. Feistel network adalah iterated cipher dengan fungsi internal yang disebut round function.

4. S-Box

   S-Box adalah matriks yang berisi subtitusi sederhana yang memetakan satu atau lebih bit dengan satu atau lebih bit yang lain.

B. *Rijndael Algorithm*

Algoritma yang dibahas di makalah ini menggunakan algoritma Rijndael sebagai salah satu dasarnya. Algoritma ini beroperasi dengan byte. Proses enkripsi algoritma ini memiliki beberapa tahap, dan dalam tahap ini dilakukan *round* transformasi sebanyak *Nr*-kali. Nilai *Nr* ditentukan oleh besar

kolom blok (*Nb*) dan kolom *key* (*Nk*). Adapun tahap-tahap algoritma adalah sebagai berikut:

1. AddRoundKey

Tahap ini adalah initial *round* dengan cara melakukan operasi XOR antara plainteks dengan *cipher key*.

2. The Round Transformation

Tahap ini tersusun dari 4 jenis transformasi yang akan diulang sebanyak Nr-1 kali. Empat transformasi tersebut adalah:

a. SubBytes

SubBytes adalah substitusi *byte* non-linear yang beroperasi secara independen pada masing-masing *array state byte*. Tabel yang digunakan untuk subtitusi *byte* dinamakan *S-box*.

b. ShiftRows

Pada proses ShiftRows, dilalukan pergeseran secara siklis pada tiap baris *array state* dengan *offset* berbeda. Baris pertama tidak digeser. Baris ke-2 digeser sebanyak 1 bit kekiri, baris ke-3 2 bit kekiri dan baris ke-4 3 bit kekiri.

c. MixColumns

Dalam tahap ini, setiap kolom *array state* dianggap sebagai polinomial dan dikalikan dengan polinom a(x) mod ($x^4 + 1$). Nilai a(x) yang digunakan adalah: a(x) = {03}x3 + {01}x2 + {01}x + {02}.

d. AddRoundKey

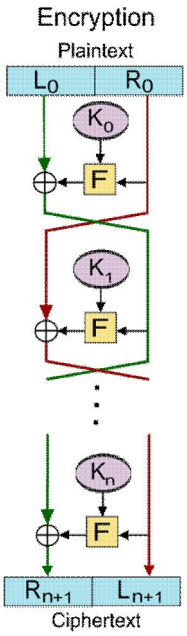
Tahap ini melakukan XOR antara *array state* dengan *round key*. *Round key* didapat dari *cipher key* menggunakan *key schedule*.

3. Final Round

Tahap ini terdiri dari transformasi yang hampir sama dengan tahap kedua, yaitu *SubBytes, ShiftRows,* dan *AddRoundKey*.

C. *Feistel Network*

Feistel network adalah struktur simetris yang digunakan untuk membangun cipher blok, dimana struktur ini memiliki keuntungan operasi enkripsi dan dekripsi sangatlah mirip dan tidak memerlukan fungsi yang invertible. Struktur ini merupakan cipher berulang dengan fungsi internal bermama fungsi round. Contoh skema enkripsi feistel network sendiri sebagai berikut:



Gambar 1. Skema enkripsi Feistel Network

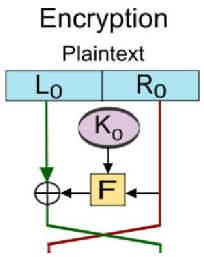
III. RANCANGAN BLOCK CIPHER

Algoritma Rijndael [1] mengutamakan kesederhanaan dalam desainnya. Salah satu buktinya dapat dilihat dari deskripsi fungsi transformasi MixColumn. Pada fungsi tersebut rijndael menentukan enam kriteria desain, yaitu:

1. Invertibilitas
2. Linearitas di GF(2)
3. Kekuatan difusi yang relevan
4. Kecepatan pada prosesor 8-bit
5. Simetri
6. Kesederhanaan penjelasan

Khususnya untuk kriteria 4, Rijndael menggunakannya untuk menentukan nilai koefisien yang kecil agar dapat digunakan dengan baik pada prosesor-prosesor 8-bit maupun hardware. Hal ini juga dapat dilihat pada fungsi-fungsi transformasi lainnya. Oleh karena itu, Block Cipher yang dibuat pada dalam makalah ini berusaha untuk mengurangi kesederhanaan yang ada di algoritma Rijndael dengan harapan mendapatkan suatu algoritma yang dengan kompleks sulit untuk diserang. Algoritma ini, dalam makalah ini selanjutnya akan disebut Algoritma Shifting Spiral.

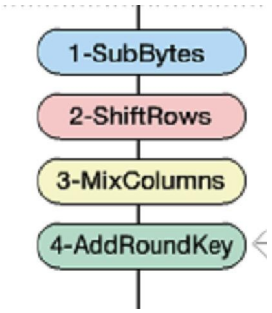
Pada Algoritma Shifting Spiral, struktur yang digunakan sebagai struktur utama adalah feistel network, sedangkan fungsi internal yang digunakan dalam feistel network itu sendiri adalah komponen The Round Transformation (Gbr. 3) dari algoritma Rijndael. Struktur feistel network yang digunakan dapat dilihat pada Gbr. 2.



Gambar 2. Struktur Feistel Network pada Algoritma Shifting Spiral

Karena pada Feistel Network plainteks akan dibagi menjadi dua bagian, dimana masing-masing bagian merupakan suatu state, dan satu state berukuran 16 byte (4x4), maka ukuran block cipher yang digunakan dalam algoritma ini adalah 16x2 = 32 byte (256 bit). Apabila ada suatu blok yang bukan merupakan kelipatan tersebut dilakukan padding untuk memenuhi bit-bit kosongnya.

Pada Gambar 1, komponen F (fungsi internal) yang digunakan seperti yang sudah disebutkan diawal merupakan komponen The Round Transformation dari algoritma Rijndael dengan sedikit modifikasi. Round key yang digunakan pada setiap round juga didasarkan pada algoritma rijndael. Perubahan yang dilakukan pada komponen Rijndael tersebut antara lain berlaku untuk fungsi SubBytes dan ShiftRows dengan tujuan untuk memperkompleks hubungan antara plainteks, cipherteks dan key (mengikuti prinsip *confusion* dan *diffusion*). Proses-proses cipher selengkapnya di algoritma Rijndael antara lain:



Gambar 3. Proses yang dilakukan pada Algoritma Rijndael

1. AddRoundKey

Pada Rijndael, ada yang disebut initial round, dimana dilakukan operasi XOR antara state awal (plainteks) dengan cipher key untuk mendapatkan round key 0. Pada algoritma ini, akan digunakan metode yang sama.

2. SubBytes

Pada Rijndael, proses SubBytes dilakukan dengan memetakan setiap byte dari array state dengan menggunakan S-Box. S-Box yang digunakan sendiri independen terhadap kunci yang digunakan. Hal ini dapat membuat algoritma menjadi lebih lemah. Oleh karena itu, kami membuat S-Box yang dependen terhadap kunci yang digunakan untuk enkripsi. Tiap sel pada S-Box akan di-XOR dengan hasil dari fungsi f (key) yang implementasinya seperti berikut (dalam kode Java):

```
private static byte getKeySubValue(byte[] byteKey){
        String key = new String(byteKey);
        int val = 0;
        for(int i=0;i<key.length();i++){
                val = (val + ((int) key.charAt(i)) * (i % 10)) % 128;
        }
        return (byte) val;
    }
```

Hasil XOR pada S-Box akan tetap membuat nilai S-Box unik sehingga cocok untuk digunakan dalam kasus ini.

3. ShiftRows

Pada algoritma kami, proses ShiftRows diubah sehingga pergeseran pada round ke i akan akan menggeser seluruh bit sebanyak i-bit secara spiral searah jarum jam, seperti pada ilustrasi berikut:

d4	e0	b8	1e
27	bf	b4	41
11	98	**5d**	52
ae	f1	e5	30

Gambar 4. State awal

Pergeseran pada round 1 akan menggeser state awal sebanyak 1 bit, menjadi:

27	d4	e0	b8
11	98	bf	1e
ae	f1	b4	41
5d	e5	30	52

Gambar 4. State akhir

Sel yang berada pada index (2, 2) berpindah ke sel pada indeks (3, 0).

4. MixColumns

Pada AES, digunakan transformasi MixColumns() dengan mengalikan setiap kolom dari array state dengan polinom a(x) mod (x^4 + 1). Pada algoritma ini metode yang digunakan adalah sama.

IV. EKSPERIMEN DAN PEMBAHASAN HASIL

Dalam mengembangkan algoritma sudah disebutkan bahwa ada beberapa prinsip yang digunakan untuk mengembangkan block cipher, salah satunya adalah *confusion* dan *diffusion*. Dalam tabel dibawah, kami menyajikan beberapa hasil enkripsi menggunakan algoritma shifting spiral serta algoritma Rijndael sebagai perbandingan, yang digunakan sebagai untuk pembahasan hasil enkripsi dan dekripsi.

Tabel 1 Perbandingan Hasil Enkripsi dengan Algoritma *Spiral shifting* dan Rijndael

No	Input String	Mode Of Operation	Waktu	Key	Enkripsi Rijndael (dalam nilai byte)	Enkripsi Shifting (dalam nilai byte)
1	Harits Elfahmi Harits Elfahmi .	ECB	0s	Ganteng	117 -25 59 13 121 -109 -74 -31 -18 118 42 95 48 112 45 88 -65 -12 49 116 101 -20 -40 73 49 -96 -32 106 117 -53 112 80	-29 75 -32 59 -42 -116 25 47 -3 55 -27 -35 -17 116 25 -91 -51 70 -37 23 -107 -32 -10 -49 47 -11 12 -38 -89 -81 -57 35
2	Harits Elfahmi Harits Elfahmy .	ECB	0s	Ganteng	117 -25 59 13 121 -109 -74 -31 -18 118 42 95 48 112 45 88 -111 42 26 -48 124 -71 21 115 -112 -68 39 45 9 35 -101 -124	105 126 107 -84 -99 -34 91 -24 -22 -102 19 -125 -50 111 -21 -63 24 71 109 -45 14 45 -15 -81 113 -1 65 -85 -84 127 11 90
3	Dinah Kamilah Ulfa Dinah Kamilah Ulfa Dinah Kamilah Ulfa Dinah .	CBC	0s	Kriptografi Kriptografi Kriptograf	43 123 -105 -90 -115 62 53 13 -69 -128 -77 98 -112 -19 17 112 123 49 -115 -18 -87 -102 12 -60 -21 -92 -56 -76 53 46 75 -104 19 37 68 -69 9 26 -59 48 123 -20 -126 -32 26 55 73 -109 -50 32 -117 -43 -67 32 -25 8 -88 116 37 27 12 87 -31 -100 101 -23 -55 -110 90 -49 -71 34 -17 -54 -32 80 5 -68 103 99	5 -57 -108 -22 85 -101 54 72 21 123 -22 86 -49 82 37 99 -76 -4 0 -81 -68 -6 120 119 93 -85 -115 -24 89 -73 -32 -112 -22 57 -103 -7 29 -17 94 -88 -13 -71 3 -50 -28 40 87 -56 -104 -28 -94 -73 -114 107 -26 -2 36 -14 111 18 116 -1 8 -32 -86 21 39 -62 71 -28 -93 117 -72 -103 66 -40 -124 125 -85 122 16 -75 -4 -54 -46 52 72 51 35 -80 38 -54 -34 115 100 37
4	Dinah Kamilah Ulfa Dinah Kamilah Ulfa Dinah Kamilah Ulfa Dinah .	CBC	0s	Kriptografi Kriptografi Kriptograp	-95 103 -67 34 40 93 -125 -79 73 43 -19 -83 -70 122 103 -64 35 -78 -18 -70 -100 -124 -58 -125 -25 18 102 115 108 27 115 49 -17 -49 -59 55 -59 29 47 35 72 -84 -13 64 -119 25 -120 114 -57 71 -98 99 51 -2 -111 -33 -35 61 117 -111 -103 53 -65 -22 -1 -127 -102 83 52 71 113 -21 107 -121 -96 -94 -43 -106 -53 -128	-35 32 127 41 -127 45 84 -36 56 37 116 -88 -26 45 -77 64 55 104 -44 88 -13 70 33 -125 91 78 104 -55 -79 -57 40 -17 -116 -71 100 -3 41 -57 -111 -42 13 -112 91 -42 -27 1 74 -38 -121 -75 -60 -115 52 48 -85 80 55 -17 25 110 -125 41 27 114 117 84 -100 74 78 -40 -72 -34 111 83 -108 102 -34 17 76 -94 118 -30 -92 -13 -116 -92 95 -2 -58 24 -116 -117 28 -67 -96 39

Pada Tabel 1, diperlihatkan perbandingan hasil enkripsi antara algoritma Rijndael serta Shifting Spiral. Pada kasus No. 1 dan 2, string input diubah satu karakter (byte) menggunakan mode ECB dan dienkripsi menggunakan algoritma Rijndael dan Shifting Spiral menggunakan string key yang sama. Dari kasus tersebut dapat dilihat pada hasil byte enkripsi yang dihasilkan, bahwa perubahan 1 byte pada input hanya merubah setengah bagian hasil enkripsi dibandingkan dengan input awal. Sedangkan pada algoritma *Spiral shifting*, hampir seluruh byte berubah, sehingga untuk kasus ini, dapat diketahui bahwa efek *diffusion* dari algoritma *Spiral shifting* lebih baik dari algoritma Rijndael.

Pada kasus No. 2, string key sepanjang 32 byte diubah satu karakter (byte) dan diberik masukan input yang sama. Dari hasil enkripsi, dapat dilihat bahwa kedua algoritma saling memberikan hasil enkripsi yang sangat berbeda dari input string key yang belum diubah, walaupun key hanya diubah satu karakter. Dari kasus ini, dapat diketahui bahwa efek *diffusion* dari algoritma Rijndael.

Walaupun tidak semua kasus dapat dicakup pada pembahasan ini, namun beberapa studi kasus diatas sudah membuktikan kekuatan algoritma *Spiral shifting* tidak terlalu buruk dibandingkan dengan algoritma Rijndael. Hal tersebut tidak aneh mengingat bahwa algoritma *Spiral shifting* sendiri didasarkan pada algoritma Rijndael dan juga mematuhi prinsip desain block cipher, yaitu: *confusion* & *diffusion*, iterated cipher, feistel network, dan S-Box. *Confusion* dan *diffusion* diperkuat menggunakan *spiral shifting*, *iterated cipher* telah ditangani dengan struktur feistel network, dan S-Box yang dependent terhadap key.

V. Analisis Keamanan

A. Related-key attack

Related-key attack adalah serangan dimana kriptoanalis melakukan operasi cipher menggunakan key yang berbeda-beda dan tidak diketahui nilainya pada awalnya, namun memiliki hubungan tertentu (contohnya selalu memiliki bagian yang sama di posisi yang sama). Karena algoritma ini merupakan modifikasi dari Rijndael yang menggunakan *key-schedule* dengan dengan difusi tinggi dan non-linearitas, tipe serangan ini akan sangat sulit dilakukan untuk algoritma ini.

B. Linear cryptanalysis

Linear cryptanalysis merupakan serangan yang berdasarkan pada menemukan pendekatan *affine* untuk proses cipher. Serangan ini dapat dilakukan bila terdapat korelasi linear yang dapat diprediksi antara input dan output pada sebagian besar *round*. Agar tahan terhadap serangan ini, tidak boleh ada *trail* linear dengan koefisien korelasi lebih tinggi dari $2^{n/2}$. Pada algoritma Rijndael, telah dibuktikan bahwa koefisien korelasi linearnya lebih rendah dari $2^{n/2}$ [1]. Hal ini dipengaruhi oleh pengacakan pattern lewat berbagai macam transformasi pada algoritma tersebut. Algoritma *Spiral Shifting* meningkatkan kerandoman pada output dengan memodifikasi proses transformasi SubBytes dan ShiftRows.

C. Differential cryptanalysis

Differential cryptanalysis merupakan tipe serangan yang menganalisis efek dari perbedaan tertentu dalam sepasang plainteks pada perbedaan dari pasangan cipherteks yang dihasilkan. Perbedaan ini dapat digunakan untuk menetapkan probabilitas ke kandidat key yang mungkin dan menemukan kunci yang paling mungkin[2]. Agar tahan terhadap serangan ini, tidak boleh ada Serangan ini mungkin dilakukan jika ada banyak propagasi perbedaan yang dapat diprediksi yang memiliki rasio prop (jumlah relatif dari semua pasangan masukan dimana setiap perbedaan masukan yang diberikan menimbulkan perbedaan output) lebih besar dari 2^{1-n}, dengan n adalah panjang blok. Agar tahan terhadap serangan ini, tidak boleh ada trail diferensial dengan rasio prop lebih tinggi dari 2^{1-n}. Pada algoritma Rijndael, telah dibuktikan bahwa nilai rasio prop lebih rendah dari 2^{1-n} [1]. Hal ini dipengaruhi oleh jumlah minimal S-box aktif dalam sejumlah putaran. Karena algoritma *Spiral Shifting* memiliki jumlah S-box aktif yang sama dengan algoritma Rijndael pada putarannya, algoritma ini memiliki nilai rasio prop yang sama.

VI. Eksperimen dan Pembahasan Hasil

Pada paper ini, kami berusaha membuat sebuah algoritma baru berdasarkan algoritma Rijndael dan Feistel Network. Desain yang kami buat menitikberatkan pada kompleksitas dengan mengubah beberapa aspek yang pada awalnya dibuat sederhana pada algoritma Rijndael. Karena kompleksitas enkripsi yang meningkat tersebut, diperkirakan bahwa algoritma ini kurang cocok untuk digunakan pada mesin dengan kekuatan pemrosesan yang kecil. Namun, perbedaan tersebut tidak dapat terlihat pada komputer-komputer modern.

Hasil studi kasus pada bab IV telah menunjukkan bahwa kekuatan hasil cipher algoritma Spiral Shifting berdasarkan prinsip desain block cipher tidak kalah hasilnya dan kadang-kadang melebihi algoritma Rijndael. Hal tersebut tidak dapat digunakan sebagai dasar untuk mengatakan bahwa algoritma Spiral Shifting lebih baik dari algoritma Rijndael, namun dari hasil yang sudah ditampilkan terlihat bahwa algoritma Spiral Shifting memiliki potensi yang baik.

Akhir kata, tidak ada suatu cara yang cukup jelas dimana sebuah algoritma Spiral Shifting dapat menjadi lebih lemah karena sudah mengikuti prinsip-prinsip desain block cipher dengan baik. Kami mengajak para pembaca untuk membantu kami menentukan kekuatan dari algoritma Spiral Shifting ini.

References

[1] AES Proposal: Rijndael, http://csrc.nist.gov/archive/aes/rijndael/Rijndael-ammended.pdf, diakses 18 Maret 2015 7:18 am.

[2] Eli Biham, Adi Shamir, *Differential Cryptanalysis of DES-like Cryptosystem*, The Weizmann Institute of Science, 1990.