Anemo : A Fresh New Dynamic Block Cipher Algorithm

T. Antra Oksidian Tafly[1], **Vincent Budianto**[2].

[1,2] Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika (STEI), Institut Teknologi Bandung (ITB), Jalan Ganesha 10, Bandung 40132
E-mail: [1]13517020@std.stei.itb.ac.id [2]13517137@std.stei.itb.ac.id

Abstract. Kriptografi merupakan suatu teknik yang paling untuk mengamankan informasi di jaman sekarang. Informasi sudah sangat cepat dan menjadikannya tidak terlalu aman. Block cipher adalah salah satu teknik pengaplikasian kriptografi modern dengan melakukan enkripsi terhadap blok yang terus bit-bit data. Pada makalah ini, kami mengusulkan algoritma baru dengan nama Anemo yang merupakan variasi block cipher berbasis algoritma *Advanced Encryption Standard* (AES). Pada algoritma ini, aspek kedinamisan merupakan aspek yang paling diutamakan sehingga tidak ada yang konstan atau statis dalam algoritma ini.

Keywords: **Block Cipher, Block Cipher Variation, Confussion, Diffusion, Feistel Network, Anemo Algorithm**

1. Pendahuluan

Kriptografi (atau kriptologi; dari bahasa Yunani κρυπτός kryptós, "tersembunyi, rahasia"; dan γράφειν graphein, "menulis", atau -λογία logi, "ilmu") merupakan keahlian dan ilmu dari cara-cara untuk berkomunikasi dengan aman dari kehadiran pihak ketiga. Algoritma kriptografi dapat dibagi menjadi dua, klasik dan modern. Algoritma kriptografi klasik salah satu contoh adalah Caesar Cipher yang diciptakan oleh Julius Caesar yang digunakan pada zaman Romawi Kuno. Sedangkan algoritma modern yang umum digunakan sekarang adalah AES, RSA, 3DES, DES, dan algoritma lainnya. Algoritma tersebut dibangun dengan tujuan keamanan sehingga dari cipher yang dihasilkan, maka plainteks/pesan awal yang disematkan tidak mudah ditebak dan diketahui dengan menggunakan teknik-teknik umum seperti analisis frekuensi.

Advanced Encryption Standard (AES) merupakan standar enkripsi dengan kunci simetris. Standar ini terdiri dari tiga penyandian blok, yaitu AES-128, AES-192, dan AES-256. Tiap-tiap penyandian memiliki ukuran blok 128 bit dengan ukuran kunci masing-masing 128 bit, 192 bit, dan 256 bit. AES telah dianalisis secara luas dan sekarang digunakan di seluruh dunia, seperti halnya dengan pendahulunya, *Data Encryption Standard* (DES).

AES didesain berdasarkan jaringan substitusi–permutasi. Hal ini dicapai dengan menggunakan *P-box* dan *S-box* untuk melakukan operasi permutasi dan substitusi. Berbeda dengan DES, AES tidak menggunakan *Feistel Network*.

Block Size	***Key Size***	***Rounds***
128 bit	128 bit	10 *rounds*
	192 bit	12 *rounds*
	256 bit	14 *rounds*

Tabel 1: Jumlah Putaran Algoritma AES

Rounds	***Steps***
Round 0	1. *Key Expansion* *Round key* diturunkan dari *cipher key* menggunakan *AES key schedule*. 2. *Add Round Key* Setiap bit digabung dengan satu bit dari *round key* menggunakan operasi *bitwise XOR* (⊕).
Round 1-9 | *Round* 1-11 | *Round* 1-13	1. *SubBytes* Setiap bit dilakukan substitusi non-linear berdasarkan *S-Box* [$b_{ij} = S(a_{ij})$]. 2. *Shift Rows* Bit pada setiap baris digeser ke kiri. Jumlah pergeseran diinkremen setiap barisnya. 3. *Mix Columns* Setiap bit pada kolom dikali dengan polinomial tetap *c(x)*. 4. *Add Round Key* Setiap bit digabung dengan satu bit dari *round key* menggunakan operasi *bitwise xor* (⊕).
Round 10 | *Round* 12 | *Round* 14 (*Last Round*)	1. *SubBytes* Setiap bit dilakukan substitusi non-linear berdasarkan S-Box [bij = S(aij)]. 2. *Shift Rows* Bit pada setiap baris digeser ke kiri. Jumlah pergeseran diinkremen setiap barisnya. 3. *Add Round Key* Setiap bit digabung dengan satu bit dari *round key* menggunakan operasi *bitwise xor* (⊕).

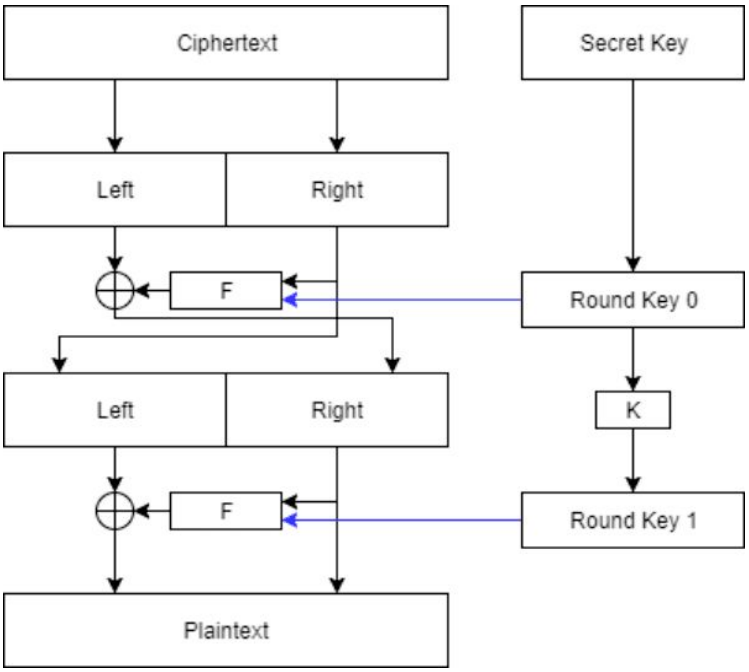
Tabel 2: Gambaran Umum Algoritma AES

Algoritma Anemo yang kami usulkan merupakan variansi dari algoritma AES yang lebih disimplifikasi dengan modifikasi sistem *diffusion*.

2. Studi Pustaka

2.1. *Feistel Network*

Feistel Network adalah sebuah teknik kriptografi yang dinamai atas kriptografer dari Jerman bernama *Horst Feistel*. *Feistel Network* menerapkan *Feistel Cipher* berulang kali pada data yang dituju. *Feistel Cipher* itu sendiri adalah sebuah struktur kriptografi yang melakukan substitusi dan permutasi terhadap data dengan membagi dua data, lalu menerapkan fungsi yang ditetapkan pengguna (Fungsi *Feistel*) dari satu bagian ke bagian lain lalu menukarnya. *Feistel Network* mempermudah proses enkripsi dan dekripsi dengan menyederhanakan struktur *Feistel Cipher* dan menyerahkan *obfuscation* kepada *Feistel Function* dan jumlah *Cipher* di *Feistel Network*, hal ini menyebabkan tidak diperlukannya algoritma khusus untuk dekripsi. Kemudahan inilah yang membuat *Feistel Network* menjadi salah satu standar yang digunakan banyak algoritma enkripsi (AES salah satunya).



Gambar 1: Ilustrasi Feistel Network

2.2. *Shannon Principles*

Shannon Principles adalah prinsip-prinsip desain yang dikemukakan oleh Claude Shannon pada tahun 1949. Prinsip-prinsip ini menentukan kualitas dari *symmetric cryptosystem*. Terdapat banyak *Shannon Principle* namun ada dua yang paling utama, yaitu:

- *Confusion*: Algoritma yang menghubungkan antara *Plaintext* dan *Ciphertext* harus dibuat serumit mungkin.
- *Diffusion*: Tiap satu bagian dari *Plaintext* dan kunci harus mempengaruhi banyak bagian dari *Ciphertext*.

2.3. *S-Box*

S-Box adalah sebuah *lookup table* berukuran MxN (M dan N tidak harus sama) untuk mencari nilai baru yang ditentukan dari nilai lama. *S-Box* diperlukan dalam sebuah algoritma sebagai basis metode *substitution* dan bekerja dengan menggunakan *input* sebagai *indeks* dari tabel. *S-Box* digunakan sebagai *substitution* agar hubungan antara *Plaintext* dan *Ciphertext* semakin rumit sesuai dengan prinsip Shannon.

hex	y															
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	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 2: S-Box yang Digunakan AES

2.4. *Iterated Cipher*

Iterated Cipher adalah sebuah teknik kriptografi dimana dilakukan enkripsi terhadap *Plaintext* berulang kali dengan sebuah upakunci atau *round key* yang bisa berubah-ubah setiap iterasinya. *Iterated Cipher* digunakan untuk menambah kerumitan dari algoritma enkripsi sesuai prinsip *Confusion* Shannon.

2.5. *Dynamic S-Box*

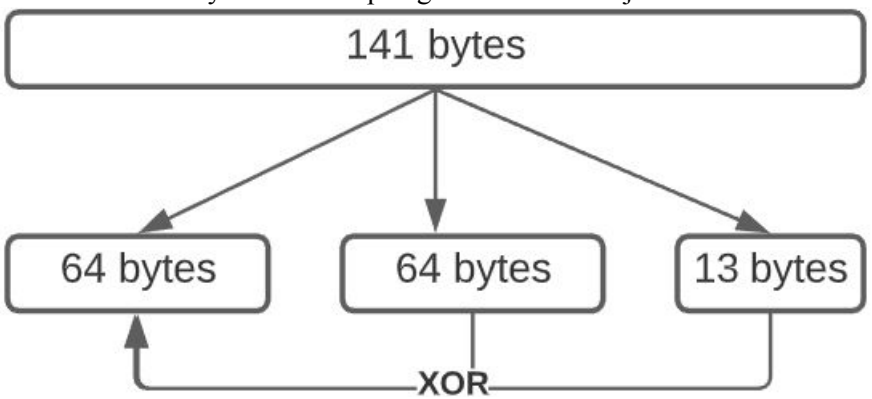
Dynamic S-Box adalah sebuah teknik kriptografi yang dimana isi dari *S-Box* berubah-ubah sesuai dengan kunci yang digunakan. Hal ini dilakukan untuk menambah kerumitan dari algoritma yang dilakukan. Selain itu, hal ini juga akan membuat detail teknik kriptografi lebih aman untuk disebarluaskan karena *Dynamic S-Box* mengurangi adanya sebuah konstanta yang mempermudah pemecahan teknik. (Hosseinkhani, 2012)

3. *Proposed Block Cipher*

3.1. *Struktur algoritma*

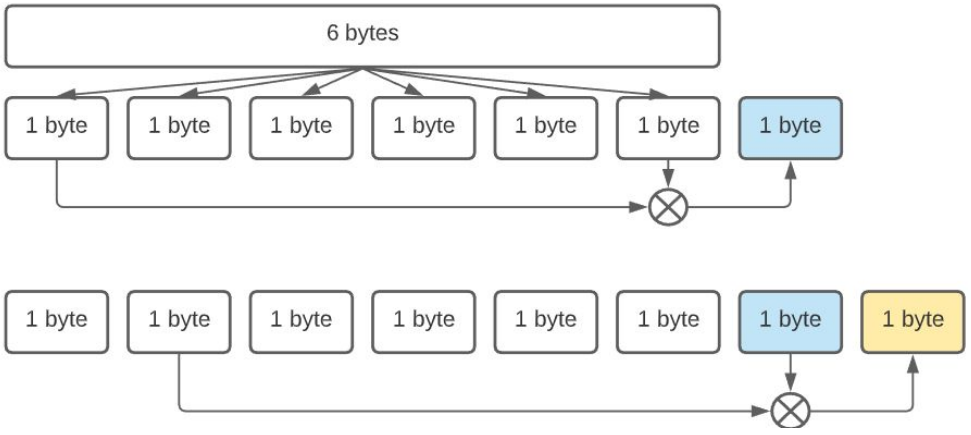
Algoritma Anemo memiliki 5 fase yang dilakukan, yaitu:

1. Fase *fitting* kunci, dimana algoritma membangkitkan sebuah kunci berukuran 64 bytes dari kunci dengan ukuran sembarang lebih dari 6 bytes. Hal ini dicapai dengan cara berikut:
    a. Jika ukuran kunci lebih besar daripada 64 bytes, kunci dipecah menjadi beberapa bagian berukuran 64 bytes lalu setiap bagian di XOR menjadi satu.



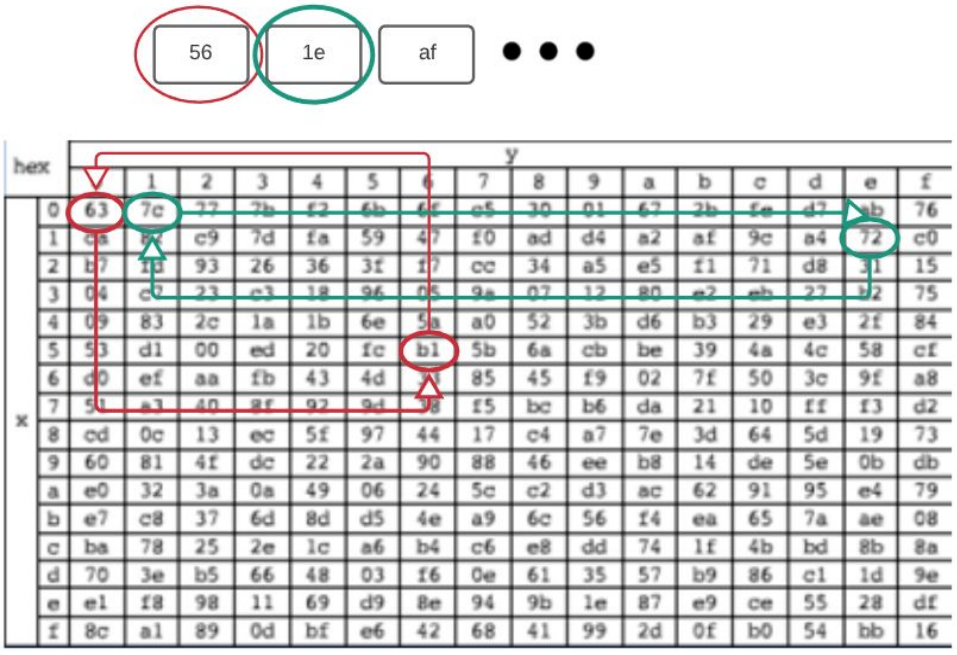
Gambar 3: Pengecilan Ukuran Kunci

    b. Jika ukuran kunci lebih kecil daripada 64 bytes, maka akan digunakan sebuah *block based keystream generator* klasik dimana *byte* keenam akan di XOR dengan *byte* pertama. Setelah itu, hasil XOR tersebut akan di sambung ke kunci awal. Hal ini dilakukan berulang kali sampai panjang kunci mencapai 64 bytes.



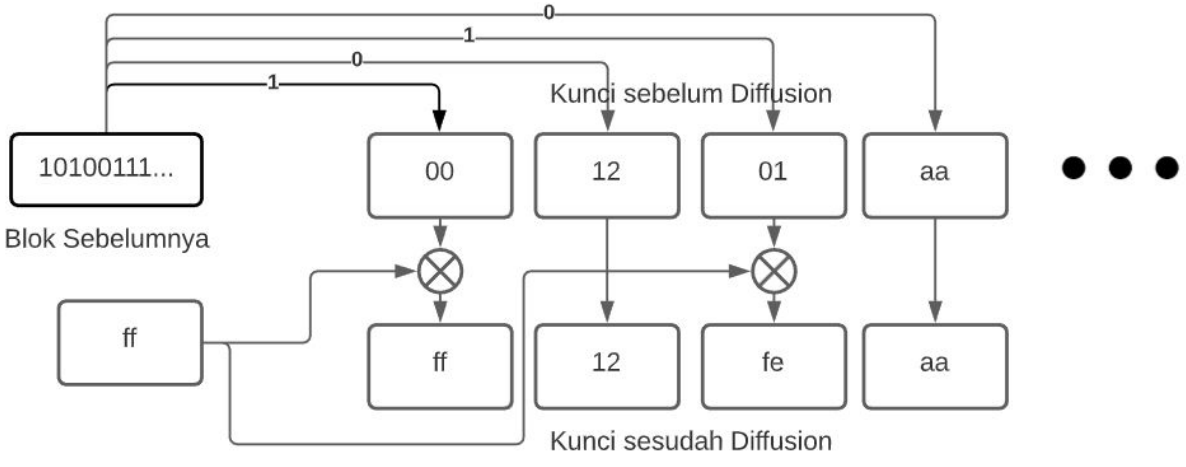
Gambar 4: Pembesaran Ukuran Kunci

2. Fase *S-Box generation*, dimana algoritma membuat sebuah *S-Box* berdasarkan kunci 64 *bytes* yang telah dibuat pada fase sebelumnya. Pembuatan *S-Box* baru dilakukan dengan cara melakukan iterasi terhadap elemen dari tabel satu-per-satu, dengan tiap iterasi dilakukan *swap* elemen dengan elemen yang memiliki indeks sama dengan nilai kunci. Setelah itu, Hal ini diulang dengan menggunakan kunci yang dibalik.



Gambar 5: Ilustrasi S-Box Generation Menggunakan Swapping

3. *Diffusion*, dimana digunakan nilai dari hasil enkripsi blok sebelumnya untuk menentukan kunci yang akan digunakan dalam enkripsi blok saat ini. Penentuan dari kunci dilakukan dengan cara melakukan XOR pada sebuah *byte* dengan *byte* "ff" jika bit pada blok sebelumnya yang berkoresponden terhadap *byte* bernilai 1.



Gambar 6: Ilustrasi Diffusion pada Kunci

4. *Round Key*, dimana akan dibuat sebuah kunci berukuran 4 bytes menggunakan seed random berdasarkan *block* sebelumnya atau counter. Seed yang digunakan adalah blok sebelumnya.
5. *Feistel Network* akan menggunakan *S-Box* dengan *Round Key* sebagai indeks untuk menentukan nilai yang akan di XOR dengan satu bagian dari *Feistel Network*. Jumlah ronde dari setiap *Feistel Network* ditentukan dengan cara sebagai berikut:

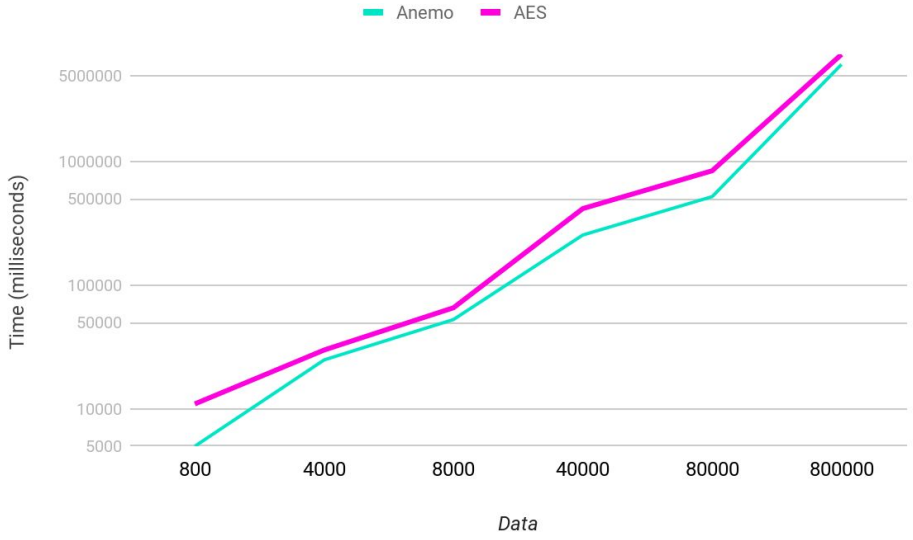
a. Setiap *Feistel Network* dimulai dengan melakukan satu ronde.
b. Dengan menggunakan *byte* pertama *round key* dari ronde pertama, didapatkan berapa kali lagi ronde yang dilakukan dengan melakukan operasi modulo *byte* pertama tersebut dengan 16.

3.2. *Perbandingan dengan AES*

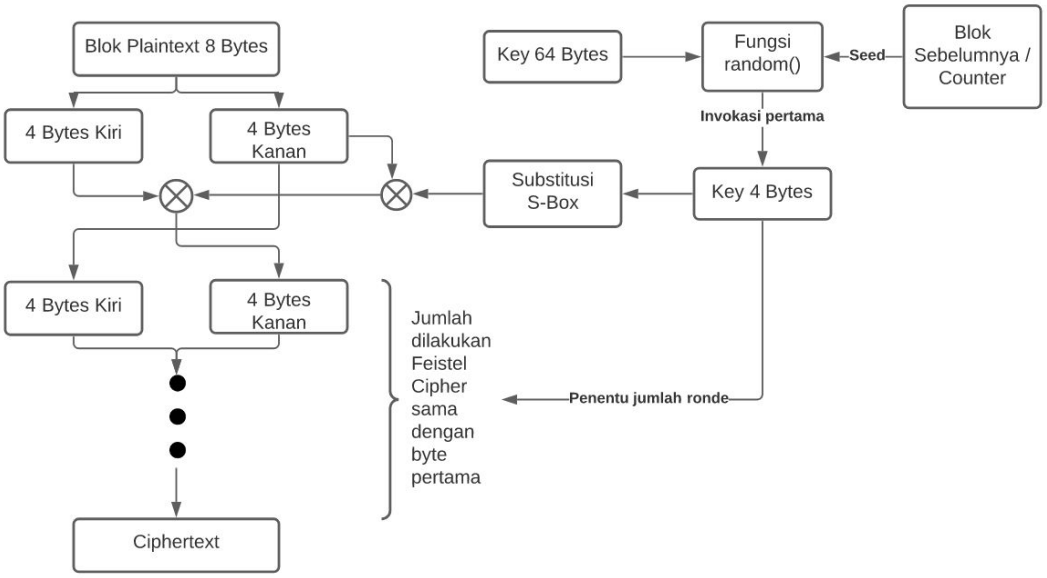
Algoritma Anemo menyederhanakan proses key expansion pada algoritma AES (AES menggunakan *AES key schedule*). Berbeda dengan AES, Anemo menerapkan prinsip *diffusion* dengan cara melakukan operasi XOR pada sebuah *byte* jika *byte* sebelumnya berkoresponden dengan nilai 1. Selain itu, tidak ada yang konstan di dalam algoritma Anemo. Hal ini menyebabkan jumlah *round* setiap *block*-nya berbeda.

4. Eksperimen dan Analisis Hasil

Algoritma Anemo lebih mengutamakan aspek kedinamisan algoritma sebagai komponen *confusion* dibandingkan dengan algoritma AES yang mengutamakan kompleksitas algoritma sehingga waktu komputasi algoritma Anemo dapat lebih cepat dibandingkan dengan algoritma AES.



Gambar 7: Perbandingan Big O Algoritma Anemo dengan Algoritma AES



Gambar 8: Feistel Network Algoritma Anemo

5. Kesimpulan dan Saran

Algoritma Anemo menggunakan AES (*Advanced Encryption Standard*) sebagai basis konsep dan implementasinya. Algoritma Anemo yang dirancang sudah cukup baik dalam melakukan enkripsi pesan. Algoritma ini telah memenuhi prinsip *confusion* dan *diffusion*.

Algoritma Anemo dapat dikembangkan lebih lanjut dengan menerapkan operasi-operasi tambahan untuk memperumit proses pada *feistel network*.

6. References

[1] Hosseinkhani, R. (2012). Using Cipher Key to Generate Dynamic S-Box in AES Cipher System. *International Journal of Computer Science and Security (IJCSS), Volume (6) : Issue (1) : 2012*, *6*, 19–28.

[2] Maxfield, M. (2006, December 20). *Tutorial: Linear Feedback Shift Registers (LFSRs) – Part 1 | EE Times*. https://www.eetimes.com/tutorial-linear-feedback-shift-registers-lfsrs-part-1/

[3] Munir, Rinaldi. Pengantar Kriptografi (2020). Slide Presentasi Kuliah IF4020. Diakses pada tanggal 21 Oktober 2020

[4] Munir, Rinaldi. Kriptografi Modern (Bagian 1). Slide Presentasi Kuliah IF4020, Diakses pada tanggal 18 Oktober 2020.

[5] Munir, Rinaldi. Kriptografi Modern (Bagian 3). Slide Presentasi Kuliah IF4020. Diakses pada tanggal 18 Oktober 2020.

Acknowledgments

Kami berterima kasih atas Tuhan YME, beserta seluruh anggota kelompok Tugas Pengganti UTS ini sehingga *paper* ini bisa diselesaikan dengan baik dan tepat waktu. Kami juga berterima kasih kepada Bapak Rinaldi Munir yang telah memberikan pengajaran dan materi sehingga tugas ini bisa selesai dengan baik.