REPOSITOR, Vol. 1, No. 2, Desember 2019, Pp. 95-104
ISSN : 2714-7975
E-ISSN : 2716-1382

Improvisasi Algoritma Advanced Encryption Standard (AES) Dengan Melakukan Pemetaan S-Box Pada Modifikasi Mixcolumns

Dana Putri Harum[*1], Aminudin[2], Sofyan Arifianto[3]
[1,2,3]Teknik Informatika/Universitas Muhammadiyah Malang
danaharum@webmail.umm.ac.id[1], aminudin2008@umm.ac.id[2], sofyan.arifianto@gmail.com[3]

Abstrak

Algoritma kriptografi merupakan salah satu unsur penting dalam pengamanan data. Berbagai improvisasi pengembangan algoritma telah dilakukan untuk mengoptimalisasikan kemanan proses pertukaran dalam suata jaringan data. Efisiensi performa menjadi salah satu pertimbangan penggunaan algoritma tertentu. AES memiliki keamanan super yang hingga saat ini keamanannya hanya dapat ditembus dengan waktu sekitar 10^{10} tahun dan semilyar processor. Kompleksitas keamanan AES sebanding dengan penggunaan penggunaan memori serta waktu yang dibutuhkan untuk memproses enkripsi dan dekripsi data begitu besar. Penelitian ini melakukan modifikasi lookup table sbox dan constant matrix pada mixcolumns. Improvisasi tersebut dilakukan dengan harapan mampu meningkatkan performa agar menjadi lebih efisien. Pengujian yang dilakukan terhadap penggunaan memori, waktu komputasi serta persentase avalanche effect masing-masing memiliki selisih sebesar 24mb, 18.9 detik, serta peningkatan avalanche effect sebesar 0.92%. berdasarkan data tersebut dapat diketahui bahwa performa pada improvisasi AES ini telah mampu meningkatkan performa algoritma dengan mereduksi waktu dan memori serta meningkatkan persentase avalanche effect.

Kata Kunci: *AES, Mixcolumns, Subbytes, Lookup Table, Constant Matrix*

Abstract

Cryptographic algorithm is an important element in data security. Any improvised of algorithm development has been carried out to optimize the security of the exchange process in a data network environment. Performance efficiency is one of the considerations of using algorithms. AES has a super security nowaday and its security can only be broken about 10^{10} years and a billion processors. The complexity of AES security is proportional with the memory usage and the time are needed to process encryption and decryption of data is huge. This study modified the sbox table and constant matrix in mixcolumns. That Improvement is expected of being able to improve the performance to become more efficient. Tests carried out on memory usage, computation time and the percentage of avalanche effect which have a difference of 24MB, 18.9 seconds, and an increase in the avalanche effect of 0.92%. based on the data, it can be known that the performance of the AES has been able to improve the performance of the algorithm by reducing time and memory also it is increasing the percentage of avalanche effect

Keywords: *AES, Mixcolumns, Subbytes, Lookup Table, Constant Matrix*

1. Pendahuluan

AES merupakan algoritma enkripsi simetris dengan panjang kunci 2^{128}[1]. Dibutuhkan waktu sekitar 10^{10} tahun dengan semilyar processor untuk memecahkannya [2]. Sehingga hanya yang memiliki private key yang dapat masuk ke dalam data maupun system [3]. Prinsip dasar kriptografi akan membuat hubungan statistik antara plaintext, ciphertext serta kunci menjadi rumit sehingga cryptanalys sulit mengetahui pola penyerangan [4]. Lebih dalam lagi Claude shannon mengatakan terdapat dua hal penting dalam kriptografi yaitu konfusi dan difusi [5]. Konsep konfusi merupakan struktur statistik dari plaintext yang dibuat secara acak tak beraturan menjadi statistik dengan rentang yang panjang dengan menggunakan kopleksitasi algoritma substitusi[6][7]. Konsep difusi pada AES mengubah setiap bit input plaintext kemudian merubah sebagian pada nilai rata–rata.

AES memiliki lookup table konfusi yang sangat bagus sehingga mampu menghancurkan pola non-linear. Sebagaimana konfusi, konsep difusi pada AES mampu meningkatkan keamanan

AES yang diketahui sebagai algoritma simetris yang dalam jangka panjang rawan untuk diserang. Struktur blockcipher pada AES memiliki empat transformasi yang mampu meningkatkan konfusi dan difusi. Transformasi addroundkey, shiftrows, subbytes yang memiliki konsep konfusi dan mixcolumns yang memiliki konsep difusi. Beberapa penelitian terkait improvisasi AES dilakukan dengan melakukan transpose matrix mixcolumns yang terbukti meningkatkan persentase avalanche effect serta improvisasi pada subbytes, shiftrows serta mixcolumns yang terbukti mampu mereduksi waktu komputasi [8][9].

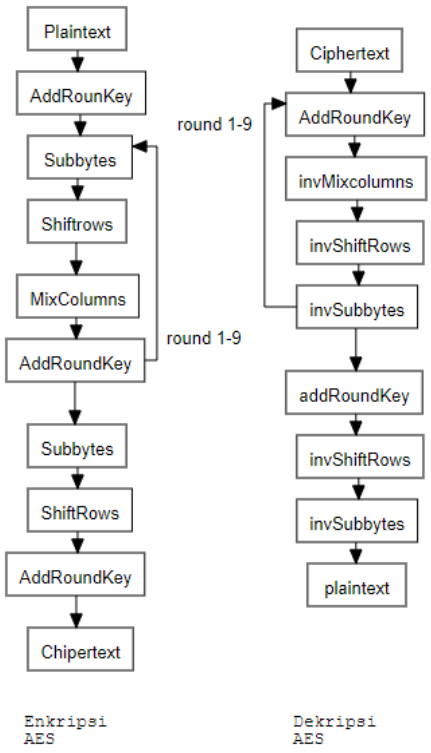
Improvisasi yang akan dilakukan pada penelitian ini akan melakukan modifikasi pada transformasi subbytes dan mixcolumns berdasarkan penelitian yang telah disebutkan sebelumnya. Tujuan penelitian ini untuk mengetahui seberapa besar peingkatan performa algoritma jika dilakukan improvisasi pada beberapa transformasi yang melibatkan konfusi dan difusi di dalamnya. Improvisasi yang dilakukan serta simulasi pengujian akan dijelaskan pada bab selanjutnya.

2. Metode Penelitian

Algoritma kriptografi pada dasarnya beropeperasi mengikuti hakikat operasi komputer digital yaitu pengolahan data terhadap simbol biner. Sebagaimana disebutkan oleh william stalling bahwa pada dasarnya aljabar polinomial yang digunakan pada AES meliputi perkalian, penjumlahan, invers dan substitusi[10]. Aljabar polinomial yang digunakan pada operasi AES adalah finite field atau galois field berderajat 4 yang dinotasikan pada Persamaan 1. satuan unit yang digunakan pada AES meliputi state yang terbagi atas blok matriks ordo 4x4 yang berukuran 16 bytes setara 128 bit. Masing – masing blok terdiri atas empat word yang berukuran 4 byte atau setara 128 bit. Satuan notasi pada AES dipengaruhi oleh Persamaan 1 yang jika diuraikan dalam operasi polinomial akan menemui hasil 128 bit sebagai operasi biner pada komputasi.

$$a_3x^3 + a_2x^2 + ax + 1 \quad (1)$$

AES pada dasarnya terdiri atas empat transformasi yaitu Addroundkey, Shiftrows, Subbytes, dan Mixcolumns. Masing – masing transformasi saling terhubung dalam suatu tahapan berulang berdasarkan panjang kunci. Kunci dengan panjang 128 bit setara 16 bytes, akan menghasilkan algoritma AES 128 bit dengan 10 perulangan yang memuat empat transformasi tersebut. algoritma kriptografi AES enkripsi dan dekripsi dapat dilihat pada Gambar 1.



Gambar 1. Enkripsi dan Dekripsi AES 128 bit

2.1. AES Standar

Addroundkey merupakan proses tarnasformasi pencampuran dua byte data dari plaintext dengan kunci. Secara dasar aljabar, proses pada addroundkey menggunakan dasar perkalian finite field $f(x) \bullet g(x)$. Operasi exclusive or digunakan sebagai operator dalam pencampuran value stateplain dan kunci sebagaimana Persamaan 2. Addroundkey pada putaran selanjutnya melakukan pencampuran data antara key schedule dengan stateplain.

$$k_{i,j} = s_{i,j} \otimes a_{i,j} \tag{2}$$

AES sebagai algoritma kriptografi block cipher, operasi yang digunakan adalah matrix yang terdiri atas 4 word yang menjadi satu kesatuan state 16 bytes. Transformasi shiftrows merupakan proses pergeseran baris matriks stateplain. Baris r = 1 digeser sebanyak 1 *byte*, r = 2 digeser sebanyak 2 *byte*, r = 3 digeser sebanyak 3 *byte*, sedangkan baris r = 0 tidak mengalami proses pergeseran [11]. Landasan matematika pada shiftrows adalah permutasi siklis. Permutasi siklis pada shiftrows pada Persamaan 3 yang kemudian menghasilkan Persamaan 4 sebagai suatu matriks stateplain yang barus dari hasil pergeseran baris.

$$P(n-1)! \Leftrightarrow P(4-1)! \tag{3}$$

$$S = \begin{matrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,3} & s_{1,0} & s_{1,1} & s_{1,2} \\ s_{2,2} & s_{2,3} & s_{2,0} & s_{2,1} \\ s_{3,1} & s_{3,2} & s_{3,3} & s_{3,0} \end{matrix} \tag{4}$$

Subbytes merupakan transformasi yang melibatkan proses substitusi dengan lookup table yang disebut sebagai s-box. Pada dasarnya operasi pada subbytes menggunakan operasi polinomial sebagaimana Persamaaan 5. Penggunaan lookup table sebagai pengembangan untuk mempermudah proses transformasi. Adapun lookup table s-box dapat dilihat pada Gambar 2. Substitusi dilaukan berdasarkan value index stateplain yang kemudian digunakan untuk mendaptkan value baru beerdasarkan kesamaan index baris dan kolom pada stateplain baru.

$$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} \tag{5}$$

	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 2. Lookup Table s-box

Perkalian matriks antara kedua stateplain dan constant matrix digunakan pada transformasi mixcolumns. konsep difusi yang disebut oleh Stalling pada AES terdapat pada transformasi mixcolumns. operasi EBC yang menyatukan bit data dan menggantikan dengan value baru mengakibatkan persebaran data pada ciphertext lebih rumit untuk diketahui pola nya. perkalian yang digunakan pada transformasi tidak lepas dari kaidah perkalian polinomial yang menggunakan operator exclusive OR(XOR). Persamaan transformasi mixcolumns dipaparkan pada Persamaan 6.

$$\begin{aligned} S'_{0,j} &= S_{0,j} * (x) \otimes S_{1,j} * (x+1) \otimes S_{2,j} \otimes S_{3,j} \\ S'_{0,j} &= S_{0,j} \otimes S_{1,j} * (x) \otimes S_{2,j} * (x+1) \otimes S_{3,j} \\ S'_{0,j} &= S_{0,j} \otimes S_{1,j} \otimes S_{2,j} * (x) \otimes S_{3,j} * (x+1) \\ S'_{0,j} &= S_{0,j} * (x+1) \otimes S_{1,j} \otimes S_{2,j} \otimes S_{3,j} * (x) \end{aligned} \quad (6)$$

2.2. AES Modifikasi

Improvisasi yang dilakukan pada penelitian ini terletak pada pengembangan transformasi subbytes dan mixcolumn. Secara utuh subalgortima transformasi keduanya tidak mengalami perubahan namun improvisasi dilakukan pada lookup table s-box dan constant matrix. Substitusi pada transformasi subbytes melibatkan lookup table yang akan menjadi value baru pada stateplain. Improvisasi dengan melakukan perkalian pada value di setiap index telah dilakukan pada penelitian sebelumnya [9]. Value s-box standar merupakan bilangan heksadesimal yang dilakukan perkalian dengan bilangan heksadesimal dengan value 02 dan 03. Sebaliknya penelitian ini hanya melakukan salah satu improvisasi yaitu perkalian 02 lookup table s-box. Berdasarkan hasil perkalian dari setiap value index beberapa menghasilkan value lebih dari 8 bit sehingga operasi tambahan digunakan untuk mereduksi kembali menjadi 8 bit sesuai dengan kaidah dasar finite fiels (2^8). operasi reduksi value index s-box dapat menggunakan operasi xor dengan 11B atau menggunakan operasi polinomial sebagaimana Persamaan 7. Setelah seluruh value berukuran 8bit maka dapat dilihat value index s-box modifikasi menjadi seperti Gambar 3.

$$m(x) = x^8 + x^4 + x^3 + x + 1 \quad (7)$$

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

Gambar 3. Table s-box Modifikasi

Constant matrix pada transformasi mixcolumns merupakan matriks 4x4. Penelitian sebelumnya telah melakukan mekaisme transpose matrix dengan tujuan meningkatkan persentase avalanche effect [8]. Transpose pada constant matrix diharapkan mampu meningkatkan persebaran data sebagai difusi pada improvisasi AES. Modifikasi contant matrix selanjutnya menghasilkan Persamaan 8.

$$\begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix} \bullet \begin{bmatrix} 02 & 01 & 01 & 03 \\ 03 & 02 & 01 & 01 \\ 01 & 03 & 02 & 01 \\ 01 & 01 & 03 & 02 \end{bmatrix} \quad (8)$$

Secara tahapan operasi transformasi mixcolumns sebagai operasi perkalian kedua matrix tersebut tidak mengalami perubahan. Value pada constant matrix yang telah dirubah dapat berpengaruh pada hasil operasi yang menjadi value dari stateplain yang baru dan akan digunakan pada tahap transformasi selanjutnya. Hasil dari modifikasi kedua transformasi tersebut harapannya mampu meningkatkan performa AES baik konfusi, difusi, serta efisiensi perangkat.

2.3. Pengujian

Mekanisme pengujian yang digunakan pada penelitian ini terbagi atas dua tahap berdasarkan rumusan masalah yaitu persentase keamanan data serta performa AES standar dengan AES modifikasi. Pengujian dengan melakukan perbandingan performa bedasarkan variabel waktu dan memori serta persentase persebaran pola data dengan avalanche effect. Sample data yang digunakan merupakan 5 file dengan format docx. dengan ukuran dan konten yang berbeda. Sistematika pengambilan data yaitu dengan membuat dample data, pengujian sample data, pengambilan data hasil pengujian kemudian analisa hasil. Berikut rancangan pengujian dan pengambilan sample data:

a. Performa

Pengambilan data perbandingan performa AES standar dengan AES modifikasi dilakukan melalui dua tahap yaitu pengukuran data berdasarkan variabel waktu komputasi dengan memori. Perbandingan performa dilakukan pada proses enkripsi dan dekripsi. Penggunaan variabel waktu dan memori sebagai pertimbangan efisiensi performa [12]. Semakin kompleks suatu algoritma maka akan semakin banyak menghabiskan memori dan waktu komputasi dan bisa jadi berpengaruh pada biaya komputasi.

b. Keamanan Data

Konfusi dan difusi yang dimiliki AES merupakan suatu sistem keamanan data. Pengukuran seberapa besar penyebaran bit data pada hasil enkripsi suatua lgoritma dapat menggunakan pengujian sederhana menggunakan avalanche effect. Avalanche effect menghitung seberapa besar persentase perbedaan bit data antara plaintext dan ciphertext sehingga dapat diketahui perbandingan persentase AE pada AES standar dan AES modifikasi. Semakin besar persentase AE maka semakin baik pula perubahan bit data nya sehingga semakin sulit dipahami pola persebaran data nya dan semakin sulit dipecahkan [13].

3. Hasil Penelitian dan Pembahasan

Implementasi dilakukan dengan membuat program berdasarkan penjabaran rumus dan transformasi AES. Sampel data kemudian diproses pada kedua program implementasi dari AES standar dan AES modifikasi. Hasil dari pengujian dari kedua lagoritma kemudian dibandingkan dan dianalisa sehingga didapatkan hasil dan kesimpulan dari penelitian.

3.1. Avalanche Effect

Tabel 1. Hasil Pengujian Avalanche Effect AES Standar

No	Nama file (.docx)	Ukuran (bytes)	Jumlah bit (bit)	Perubahan bit (bit)	Avalanche effect(%)
1	01A	13.187	560	304	54,29
2	02A	13.550	2.304	1.176	51,04
3	03A	14.590	3.200	1.649	51,53
4	04A	15.764	3.600	1.824	50,67
5	05A	15.905	4.480	2.259	50,42
Rata - rata			2.828	1.485	52,51

Berdasarkan Tabel 1 dapt diketahui hasil avalanche effect pada AES standar rata – rata sebesar 52,51%. Persentase tertinggi terdapat pada file dengan ukuran terkecil sedangkan persentase pada ukuran sampel data lainnya memiliki rata- rata 50%. Berdasarkan keacakan data yang dihasilkan pada perhitungan AE menunjukkan bahwa besarnya ukuran file tidak mempengaruhi besaran persentase keacakan data. Sehingga jika dilihat berdasarkan pola persebaran data ciphertext memiliki sifat konfusi dan difusi yang besar.

Tabel 2. Hasil Pengujian Avalanche Effect AES Modifikasi

No	Nama file (.docx)	Ukuran (bytes)	Jumlah bit (bit)	Perubahan bit (bit)	Avalanche effect(%)
1	01A	13.187	560	313	55,89
2	02A	13.550	2.304	1.203	52,21
3	03A	14.590	3.200	1.705	53,28
4	04A	15.764	3.600	1.884	52,33
5	05A	15.905	4.480	2.324	51,87
Rata - rata			2.828	1.485	52,51

Pada Tabel 2, hasil persentase avalanche effect pada AES modifikasi cenderung lebih tinggi dibandingkan AES standar. Sebagaimana rata – rata dari kedua algoritma memiliki selisih persentase sebesar 0.92% lebih besar AES modifikasi. Hal ini dipengaruhi adanya perubahan value s-box yang telah dimodifikasi sehingga mampu memberikan tingkat persebaran bit data semakin tinggi. Semakin besar persentase AE semakin besar pula tingkat keamanan data berdasarkan konfusi dan difusi algoritma.

3.2. Waktu Komputasi

Tabel 3. Hasil Pengujian Perbandingan Waktu Komputasi Enkripsi

No	Nama file (.docx)	Ukuran (bytes)	Standar (ms)	Modifikasi (ms)
1	01A	13.187	3.316	116
2	02A	13.550	5.540	160
3.	03A	14.590	9.724	265
4	04A	15.764	22.110	1.049
5	05A	15.905	43.701	1.693
Rata - rata			18.905	656

Berdasarkan hasil perbandingan dari kedua proses enkripsi memiliki perubahan waktu eksekusi yang cukup signifikan. Selisih kedua algoritma ini memiliki rata – rata sebesar 18.249ms lebih besar AES standar. Berdasarkan analisa pengukuran performa variabel waktu, efisiensi waktu pada AES modifikasi dapat dikatakan berhasil karena dapat mereduksi waktu komputasi saat eksekusi enkripsi sampel data.

Tabel 4. Hasil Pengujian Perbandingan Waktu Komputasi Dekripsi

No	Nama file (.docx)	Ukuran (bytes)	Standar (ms)	Modifikasi (ms)
1	01A	13.187	2.261	141
2	02A	13.550	8.238	280
3.	03A	14.590	12.779	375
4	04A	15.764	33.496	1.197
5	05A	15.905	61.282	2.101
Rata - rata			23.612	818

Performa algoritma dapat dipertimbangkan berdasarkan kemanan data pada saat komputasi yaitu selisih waktu enkripsi dan dekripsi. Waktu komputasi dekripsi dianjurkan lebih besar dibandingkan enkripsi. Semakin rendah waktu komputasi pada dekripsi, semakin besar pula kemungkinan serangan oleh cryptanalys untuk memecahkan kunci dan plaintext. Selisih waktu dari masing – masing algoritma pada saat proses enkripsi dan dekripsi memiliki selisih waktu lebih besar pada dekripsi sebagaimana pada Tabel 3 dan Tabel 4. Secara keseluruhan perbandingan waktu komputasi kedua algoritma memiliki selisih rata–rata yang cukup besar. Selisih yang terjadi dikarenakan adanya perubahan skema pada transformasi mixcolumns.

Secara umum metode yang digunakan pada mixcolumns adalah perkalian matriks ordo 4x4. Operator shift left digunakan pada saat value dari constant matrix bernilai 02 atau 03. Implementasi pada modifikasi mixcolumn tetap menggunakan kaidah perkalian matriks namun mengurangi salah satu operasi shift left. Perubahan difusi tetap meningkat pada AES modifikasi

dikarenakan kaidah percampuran data untuk meningkatkan difusi tetap digunakan dengan merubah constant matrix sebagaimana persamaan (8). Sehingga efisiensi waktu serta peningkatan difusi data menjadikan performa pada AES modifikasi mengalami peningkatan.

3.3. Memory Profiling

Tabel 5. Memory Profiling Enkripsi AES Standar dan Modifikasi

No	Nama file (.docx)	Ukuran (bytes)	Heap used	
			Standar (bytes)	Modifikasi (bytes)
1	0	0	11.613.416	10.950.208
2	01A	13.187	21.606.680	19.779.600
3	02A	13.550	40.421.128	28.770.400
4	03A	14.590	50.510.984	42.560.176
5	04A	15.764	81.592.616	63.202.640
6	05A	15.905	102.240.100	88.456.840
Rata - rata			61.596.984	34.821.761

Tabel 5 menunjukkan hasil memory profiling dengna mengukur jumlah heap yang digunakan pada saat eksekusi program. Analisa pengujian ini dilakukan berdasarkan besarnya heap yang digunakan ketika program dijalankan tanpa diberikan beban proses enkripsi maupun dekripsi kemudian mengukur jika program diberikan beban proses enkripsi dan dekripsi. Eksekusi awal AES standar tanpa beban enkripsi, jumlah heap yang digunakan sebesar 11.613.416 bytes. AES modifikasi ketika diberikan beban enkripsi sebesar 10.950.208 bytes. Selisih waktu eksekusi kedua program ini mencapai sebesar 663.208 bytes. Secara eksekusi program tanpa ditambahkan dengan beban proses enkripsi dari kedua program berarti AES modifikasi dari segi penggunaan memory memiliki sisa ruang lebih daripada AES standar.

Penambahan beban proses enkripsi dari kelima file sampel data memberikan beban ukuran heap menjadi meningkat. Rata – rata selisih heap yang digunakan pada masing – masing AES standar dan modifikasi mencapai 26.775.223 bytes. Hasil ini membuktikan perbedaan ukuran heap pada AES modifikasi dapat mereduksi alokasi memori saat enkripsi maupun tanpa beban proses enkripsi.

Tabel 6. Memory Profiling Dekripsi AES Standar dan Modifikasi

No	Nama file (.docx)	Ukuran (bytes)	Heap used	
			Standar (bytes)	Modifikasi (bytes)
1	0	0	11.613.416	10.950.208
2	01A	13.187	21.606.680	19.779.600
3	02A	13.550	40.421.128	28.770.400
4	03A	14.590	50.510.984	42.560.176
5	04A	15.764	81.592.616	63.202.640
6	05A	15.905	102.240.100	88.456.840
Rata - rata			61.596.984	34.821.761

Eksekusi program tanpa diberikan beban dari kedua algoritma masing – masing AES standar dan AES modifikasi sebesar 11.613.416 dan 10.950.208. perubahan ukuran heap memiliki peningkatan ketika diberikan beban dekripsi 5 file dengan ukuran berbeda sehingga menghasilkan ukuran heap masing – masing AES standar dan AES modifikasi sebesar 67.136.860 bytes dan 42.290.180 bytes. Berdasarkan Tabel 5 dan Tabel 6 menghasilkan rata–rata ukuran heap pada saat mendapat bebean enkripsi dan dekripsi dari kedua lagoritma memiliki selisih dimana heap used cenderung lebih tinggi ketika dekripsi dibandingkan enkripsi.

Selisih data hasil dari pengujian memory profiling disebabkan adanya improvisasi pada transformasi mixcolumns. pengurangan operasi shift left dapat memberikan efek reduksi memori ketika program mendapatkan beban enkripsi dan dekripsi. Berkurangnya operasi shift left tidak memberikan dampak menurunnya persentase avalanche effect sebagaimana pada Tabel 1 dan 2.

3.4. Analisa performa AES standar dan AES modifikasi

Pengujian performa suatu algoritma dapat menggunakan beberapa variabel seperti waktu dan memori. Besarnya memori yang digunakan dapat diukur berdasarkan memory profiling. Semakin besar alokasi memori yang digunakan semakin rendah performa suatu algorit. Sama hal nya dengan pengujian seberapa besar waktu yang digunakan program untuk menyelesaikan proses enkripsi dan dekripsi data. Besarnya waktu komputasi menggambarkan seberapa besar efisiensi waktu yang digunakan pada algoritma maupun implementasi program nya. Variabel waktu dan memori menjadi pertimbangan seberapa efisien AES standar dan AES modifikasi.

Pengujian performa suatu algoritma dapat menggunakan beberapa variabel seperti waktu dan memori. Besarnya memori yang digunakan dapat diukur berdasarkan memory profiling. Semakin besar alokasi memori yang digunakan semakin rendah performa suatu algorit. Sama hal nya dengan pengujian seberapa besar waktu yang digunakan program untuk menyelesaikan proses enkripsi dan dekripsi data. Besarnya waktu komputasi menggambarkan seberapa besar efisiensi waktu yang digunakan pada algoritma maupun implementasi program nya. Variabel waktu dan memori menjadi pertimbangan seberapa efisien AES standar dan AES modifikasi.

Kecepatan komputasi antara AES standar dan AES modifikasi memiliki perbandingan rata–rata waktu enkripsi sebesar 18.905,8 ms dan 656,6 ms. Rata-rata waktu dekripsi cenderung lebih tinggi daripada proses enkripsi yaitu sebesar 23.612 ms dan 818 ms. Performa dari kedua algoritma diukur berdasarkan rata – rata heap used pada eksekusi enkripsi 5 file memiliki selisih sebesar 26 mb. Hasil pengujian pada proses dekripsi kedua algoritma menghasilkan selisih rata – rata heap size sebesar 24 mb. Selisih data yang dihasilkan pada pengujian memory profiling dapat disimpulkan bahwa heap used pada AES modifikasi telah mampu meningkatkan performa berdasarkan alokasi memori yang digunakan pada saat eksekusi program.

AES modifikasi terbukti mampu mereduksi penggunaan memori serta waktu eksekusi enkripsi dan dekripsi berdasarkan data pengujian pada waktu eksekusi dan memory profiling. Data pengujian menunjukkan bahwa performa algoritma meningkat seiring menurunnya penggunaan memori dan waktu yang digunakan untuk menyelesaikan proses enkripsi maupun dekripsi.

4. Kesimpulan

Performa sutau algoritma kriptografi dapat diukur berdasarkan variabel waktu dan memori. Improvisasi AES dengna tujuan meningkatkan performa komputasi harus mempertimbangkan efisiensi daya, biaya, serta keamanan data. Berdasarkan penelitian yang telah dilakukan performa AES modifikasi terbukti mampu meningkatkan keamanan data yaitu AE sebesar 0.92%. seiring dengan meningkatnya kemanan data, performa AES modifikasi mempu mereduksi waktu komputasi serta penggunaan memory berdasarkan heap used.

Meskipun terdapat peningkatan konfusi dan difusi pada AES modifikasi, namun peningkatan AE masih relatif memiliki jarak yang sangat dekat dengan hasil AE dari AES standar. Maka dari itu pada penelitian selanjutnya perlu dilakukan pengembangan pada transformasi mixcolumn dengan landasan utama aljabar polinomial GF (2^8) dengan modulo $x^4 + 1$ serta invers nya. sehingga segala perubahan pada constant matrix mixcolumn akan didapatkan invers dari persamaan polinomial yang baru.

Referensi

[1] Muharram F. Analisis Algoritma pada Proses Enkripsi dan Dekripsi File Menggunakan Advanced Encryption Standard. *Prosiding Seminar Nasional Ilmu Komputer dan Teknologi Informasi*; 3.
[2] Renaldi Munir. *KRIPTOGRAFI*. Bandung, 2006.
[3] Salim MA. Analisa Algoritma AES Modifikasi dengan Teknik Blum Blum Shub - Chaotic Function dan Modifikasi ShiftRows.
[4] Putera A, Siahaan U. Rail Fence Cryptography in Securing Information Dining Philosophers Theory and Concept in Operating System Scheduling View project A Fast Induction Motor Speed Estimation based on Hybrid Particle Swarm Optimization (HPSO) View project Rail Fence Cryptography in Securing Information. *International Journal of Scientific & Engineering Research*; 7, http://www.ijser.org (2016).
[5] Shannon CE. *Communication Theory of Secrecy Systems*.
[6] Stallings W. *Cryptography and Network Security Principles and Practices, Fourth Edition*. 2005.

[7] Abdulah HS, AHamood Al-Rawi M, Hammod DN, et al. *Analysis of AES Algorithm Effects on the Diffusion Property*.
[8] Perolihin EB. Penerapan Transpose Matriks pada Algoritma Kriptografi Aes untuk Content Database.
[9] Riyaldhi R, Rojali, Kurniawan A. Improvement of Advanced Encryption Standard Algorithm with Shift Row and S.Box Modification Mapping in Mix Column. *Procedia Computer Science* 2017; 116: 401–407.
[10] Stallings W. *Cryptography and Network Security principles and practices* . Fourth. Prentice Hall, 2015.
[11] Daemen J, Rijmen V, Pwi B, et al. AES Proposal : Rijndael.
[12] Marek L, Zheng Y, Ansaloni D, et al. Introduction to dynamic program analysis with DiSL. *Science of Computer Programming* 2015; 98: 100–115.
[13] Aminudin A, Helmi AF, Arifianto S. Analisa Kombinasi Algoritma Merkle-Hellman Knapscak dan Logaritma Diskrit pada Aplikasi Chat. *Jurnal Teknologi Informasi dan Ilmu Komputer* 2018; 5: 325.

ISSN: 2714-7975; E-ISSN: 2716-1382