http://ejournal.unisba.ac.id
ISSN: 1412-5056

Kriptografi *Advanced Encryption Standard* (AES) Untuk Penyandian File Dokumen

Aditia Rahmat Tulloh, Yurika Permanasari, Erwin Harahap
Program Studi Matematika, FMIPA, Universitas Islam Bandung,

adit.rahmat051994@gmail.com, yurikape@unisba.ac.id, erwin2h@unisba.ac.id

Abstrak. *Advanced Encryption Standard (AES)* adalah algoritma kriptografi yang menjadi standar algoritma enkripsi kunci simetris pada saat ini. Dalam algoritma kriptografi AES 128, 1blok plainteks berukuran 128 bit terlebih dahulu dikonversi menjadi matriks heksadesimal berukuran 4x4 yang disebut state. Setiap elemen *state* berukuran 1 *byte*. Proses enkripsi pada AES merupakan transformasi terhadap state secara berulang dalam 10 ronde. Setiap ronde AES membutuhkan satu kunci hasil dari generasi kunci yang menggunakan 2 transformasi yaitu subtitusi dan transformasi. Pada proses enkripsi AES mengunakan 4 transformasi dasar dengan urutan trasformasi *subbytes, shiftrows, mixcolumns,* dan *addroundkey*. Sedangkan pada proses dekripsi mengunakan invers semua transformasi dasar pada algoritma AES kecuali *addroundkey* dengan urutan transformasi *invshiftrows, invsubbytes, addroundkey,dan invmixcolumns.* Pada data teks, proses enkripsi diawali dengan mengkonversi teks menjadi kode ASCII dalam bilangan heksadesimal yang dibentuk menjadi matriks byte 4x4. Selanjutnya dilakukan beberapa trnsformasi dasar seperti *subbytes, shiftrows, mixcolumns,* dan *addroundkey.* Akan tetapi ketika melakukan trasformasi data yang diproses pada setiap trasformasi berupa data biner dari matriks heksadesimal. Kriptografi AES 128 bit memiliki ruang kunci 2^{128} yang merupakan nilai yang sangat besar dan dianggap aman untuk digunakan sehingga terhindar dari *brute force attack.*

Kata Kunci: AES, Penyandian file, Algoritma kunci simetris.

Abstract. (*Cryptography Advanced Encryption Standard (AES) for File Document Encryption*). Advanced Encryption Standard (AES) is a cryptographic algorithms as a standard symmetric key encryption algorithm that used in current time. AES 128 has 1 blok plaintext with 128 bit sized, where in the process of cryptographic algorithms, first the plaintext is converted into hexadecimal-sized 4 x 4 matrices called the state, where each element of state has 1 byte size. The process of encryption on AES is the transformation towards the state repeatedly in the 10th round. Each round of AES requires one key result of the key generation using 2 basic transformation, i.e. substitution and transformation. AES encryption using 4 transformation by the following sequence: subbytes, shiftrows, mixcolumns, and addroundkey. On the other hand, the process of decryption is using the inverse of all the basic transformation of AES algorithm, except addroundkey. Therefore, the sequence of transformation on the decription is invshiftrows, invsubbytes, invmixcolumns, and addroundkey. In the data text, the encryption process is initiated with convertion the data text into ASCII code in hexadecimal numbers that are molded into the matrix 4 x 4 bytes. Next, do some basic transformation such as subbytes, shiftrows, mixcolumns, and addroundkey. However, when performing the transformation, the processed data on every trasformation is in the form of binary data obtained from the hexadecimal matrix. AES 128 bit cryptography have room 2^{128} keys which is a tremendous value and is considered secure to use to avoid the brute force attack.

Keywords: AES, file Encryption, symmetric key algorithm.

1. Pendahuluan

Perkembangan teknologi informasi dan komunikasi saat ini, mengakibatkan manusia dapat berkomunikasi dan saling bertukar data dan informasi tanpa dihalangi oleh jarak dan waktu. Seiring dengan tuntutan akan keamanan untuk kerahasiaan informasi yang saling dipertukarkan tersebut semakin meningkat menimbulkan tuntutan tersedianya suatu sistem pengamanan data dan informasi yang lebih baik agar dapat mengamankan data dari berbagai ancaman. Oleh karena itu berkembangnya cabang ilmu yang mempelajari tentang cara-cara pengamanan data merupakan dampak positif dari tuntutan tersedianya sistem keamanan data yang berfungsi untuk melindungi

data yang ditransmisikan atau dikirimkan melalui suatu jaringan komunikasi. Ilmu yang mempelajari tentang cara-cara pengamanan data dikenal dengan nama Kriptografi.

Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan data dan informasi seperti keabsahan data, integritas data, serta autentifikasi data. Sistem kriptografi adalah suatu fasilitas untuk mengkonversikan pesan jelas (plainteks) ke pesan yang telah disandikan (cipherteks). Proses konversi ini disebut enkripsi (*encryption*). Sebaliknya, menerjemahkan cipherteks menjadi plainteks disebut dengan dekripsi (*decryption*). Proses enkripsi dan dekripsi menggunakan satu atau beberapa kunci kriptografi.

Pada tahun 2000, *National institute of standards and technology* (NIST) sebagai agensi departemen perdagangan AS menetapkan sebuah standard kriptografi yang baru yaitu Algoritma Rijandel dan ditetapkan sebagai *Advanced Encryption Standard* (AES) (Munir, 2006). *Advanced Encryption Standard* (AES) secara garis besar beroperasi pada blok 128-bit atau 16 karakter, yang berarti dapat digunakan untuk enkripsi teks. File dokumen terdiri dari barisan teks yang tentu saja berukuran lebih dari 16 karakter, akan tetapi AES dapat digunakan untuk penyandian yaitu dengan melakukan enkripsi perblok (128 bit) secara pararel untuk memudahkan proses enkripsi maupun dekripsi digunakan *software* aplikasi MATLAB. Berdasarkan latar belakang yang telah diuraikan, maka perumusan masalah dalam penelitian ini sebagai berikut: "Bagaimana proses penyandian dengan *Advanced Encryption Standard* (AES)?". Selanjutnya, tujuan dalam penelitian ini diuraikan dalam pokok-pokok sbb.

1. Memahami proses penyandian dengan *Advanced Encryption Standard* (AES).
2. Mengetahui penerapan Algoritma kriptografi AES pada file teks.
3. Dapat merancang dan menggunakan program pengamanan data teks metode Kriptografi AES dengan menggunakan *Graphical User Interface* (GUI) MATLAB.

2. Landasan Teori

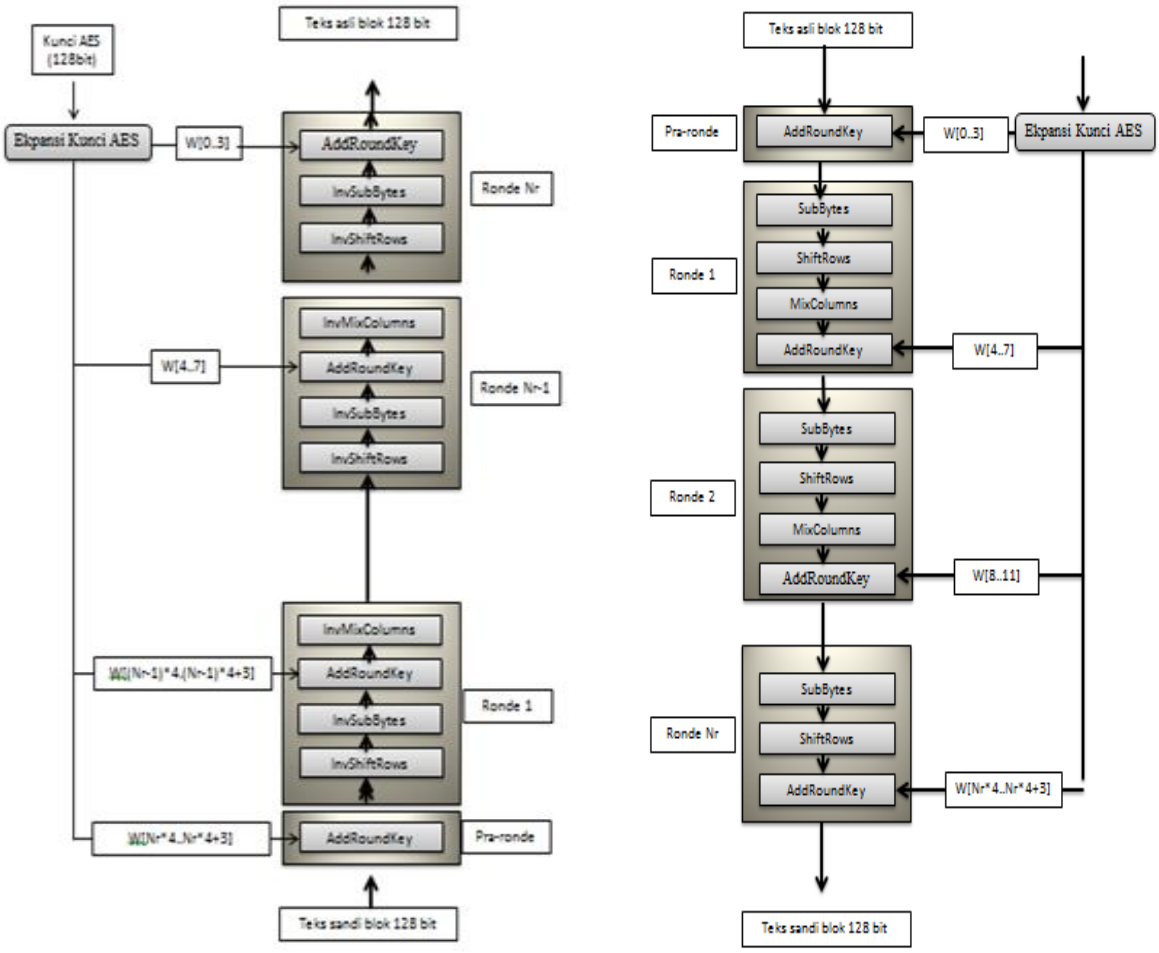
AES merupakan sistem penyandian blok yang bersifat non-Feistel karena AES menggunakan komponen yang selalu memiliki invers dengan panjang blok 128 bit. Kunci AES menggunakan proses yang berulang yang disebut dengan ronde. Proses di dalam AES merupakan transformasi terhadap *state*. Sebuah teks asli dalam blok (128 bit) terlebih dahulu diorganisir sebagai *state*. Enkripsi AES adalah transformasi terhadap *state* secara berulang dalam beberapa ronde. *State* yang menjadi keluara ronde k menjadi masukan untuk ronde ke-k +1.

Pada Proses enkripsi awalnya teks asli dibentuk sebagai sebuah *state*. Kemudian sebelum ronde 1 dimulai blok teks asli dicampur dengan kunci ronde ke-0 (transformasi ini disebut *AddRoundKey*). Setelah itu, ronde ke-1 sampai dengan ronde ke-(Nr-1) dengan Nr adalah jumlah ronde. AES menggunakan 4 jenis transformasi yaitu:

1. *SubBytes*, sebagai transformasi subtutusi.
2. *ShiftRows*, sebagai transformasi permutasi.
3. *MixColumns*, sebagai transformasi pengacakan.
4. *AddRoundKey*, sebagai transformasi penambahan kunci.

Pada ronde terakhir, yaitu ronde ke-Nr dilakukan transformsi serupa dengan ronde lain namun tanpa transfomasi serupa dengan ronde lain namun tanpa transformasi MixColumns.

Algotirma dekripsi AES dapat diilustrasikan seperti Gambar 1. Secara ringkas algoritma deskripsi merupakan kebalikan algoritma enkripsi AES. Algoritma dekripsi AES menggunakan Tansformasi invers semua transformasi dasar yang digunakan pada algoritma enkripsi AES. Setiap tranformasi dasar dari algoritma kriptografi AES memiliki transformasi invers, yaitu: InvSubBytes, InvShiftRows dan InvMixColumns. AddroundKey merupakan transformasi yang bersifat self-invers dengan syarat mengunakan kunci yang sama (Stalling, 2003).



Gambar 1. Proses enkripsi dan dekripsi

Penyandian AES membutuhkan kunci ronde untuk setiap ronde transformasi kunci ronde ini di bangkitkan (di ekspansi) dari kunci AES. Pada bagian ini di bahas bagaimana kunci ronde di bangkitkan oleh kunci AES. Kunci AES 128 bit atau 4 word menghasilkan sebuah larik sebanyak 44 word yang menjadi kunci. Berikut adalah langkah langkah mengekspansi kunci:

1. Pertama kunci AES 128 bit di organisir menjadi 4 word dan disalin ke word keluaran (*W*) pada 4 elemen pertama (*W[0], W[1], W[2], W[3]*).
2. Untuk elemen keluaran selanjutnya *W[i]* dengan i ={4,…,43} dihitung sebagai berikut:

    a. Salin *W [i-1]* pada word *t*.
    b. Jika i mod 4 = 0 (I habis dibagi 4) maka lakukan $W[i] = f(t,i) \oplus W[i-4]$,dengan fungsi f(t,i) adalah sebagai berikut:

    $$f(t,i) = Subword\ (rotword(t)) \oplus RC[i/4]$$

    c. Jika I mod 4 tidak sama dengan 0, lakukan $W[i] = t \oplus W[i-4]$.

3. Hasil Penelitian dan Pembahasan

Misalkan seseorang akan mengirim sebuah plainteks yang berisi 128 bit atau 16 byte atau 16 karakter seperti berikut:

Plainteks: *Kriptografi AES* dengan Kunci: *Aditia*

Maka plainteks ini dapat dibuat menjadi state berikut:

Plainteks : Kunci :

K	t	a	A
R	o	f	E
I	g	i	S
P	r	(spase)	(null)

A	i	(null)	(null)
D	a	(null)	(null)
I	(null)	(null)	(null)
T	(null)	(null)	(null)

Karena pada AES menggunakan representasi *byte* maka menggunakan bilangan heksaadesimal. Sehingga karakter teks diatas di konversi menjadi bilangan heksaadesimal dengan melihat table KODE ASCII maka di dapat seperti berikut:

Plainteks : Kunci :

4B	74	61	41
72	6F	66	45
69	67	69	53
70	72	20	00

41	69	00	00
64	61	00	00
69	00	00	00
74	00	00	00

Proses Ekspansi Kunci

Proses ekspansi kunci seperti berikut:

41	69	00	00
64	61	00	00
69	00	00	00
74	00	00	00

Atau dapat ditulis
W_0= 41 64 69 74
W_1= 69 61 00 00
W_2= 00 00 00 00
W_3= 00 00 00 00

Maka untuk mencari W_4 adalah seperti berikut

$W_4 = W_0 \oplus subword\ (rotword\ (W_3)) \oplus RC[1]$
$= 41\ 64\ 69\ 74 \oplus subword\ (rotword\ (00\ 00\ 00\ 00)) \oplus 01\ 00\ 00\ 00$
$= 41\ 64\ 69\ 74 \oplus subword\ (00\ 00\ 00\ 00) \oplus 01\ 00\ 00\ 00$
$= 23\ 07\ 0A\ 17$

Hasil perhitungan untuk ekspansi untuk 1 ronde terdapat pada table 3.2 seperti berikut ini:

Tabel 3.2 Ekspansi Kunci

W(i)	***W(i-1)***	***After subword ⊕ rcon ⊕ W(i-NK)***
W4	00 00 00 00	23 07 0A 17
W5	23 07 0A 17	4A 66 0A 17
W6	4A 66 0A 17	4A 66 0A 17
W7	4A 66 0A 17	4A 66 0A 17

Proses Enkripsi

Hal pertama yang harus dilakukan adalah Melakukan xor antara plainteks dan kunci, seperti berikut:

4B	74	61	41
72	6F	66	45
69	67	69	53
70	72	20	00

$\oplus$

41	69	00	00
64	61	00	00
69	00	00	00
74	00	00	00

$=$

0A	1D	61	41
16	0E	66	45
00	67	69	53
04	72	20	00

Ronde 1

a. Hasil Proses *Subbyte* (menggunakan table s- box)

67	A4	EF	83
47	AB	33	6E
63	85	F9	ED
F2	40	B7	63

b. Transformasi *ShiftRows*

67	A4	EF	83
47	AB	33	6E
63	85	F9	ED
F2	40	B7	63

ShiftRows →

67	A4	EF	83
AB	33	6E	47
F9	ED	63	85
63	F2	40	B7

c. Proses *MixColumns*

Proses ini merupakan proses terbanyak dari pada proses proses lain setiap rounde. Kali ini penulis membagi proses *mixcolumns* menjadi 4 bagian untuk sebuah matriks atau state karena dikerjakan untuk setiap kolom sebagai berikut:

1. Proses *mixcolumn* untuk kolom pertama.

$$\begin{bmatrix} S'(0,1) \\ S'(1,1) \\ S'(2,1) \\ S'(3,1) \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} * \begin{bmatrix} 67 & A4 & EF & 83 \\ AB & 33 & 6E & 47 \\ F9 & ED & 63 & 85 \\ 63 & F2 & 40 & B7 \end{bmatrix}$$

$$= \begin{bmatrix} B2 & 19 & 54 & E6 \\ 59 & 1C & B8 & 2E \\ 80 & 5B & 87 & 17 \\ 3D & D6 & A7 & 29 \end{bmatrix}$$

Pada ronde pertama didapat Cipherteks yang akan menjadi masukan atau input untuk ronde 2, begitu juga cipherteks yang didapat pada ronde 2 kan digunakan menjadi input pada ronde 3. Proses seperti ini berlangsung hingga ronde 10. Pada ronde 10 didapat hasil enkripsi sebagai berikut:

Ronde-10 :

Sub-byte =

D9	36	01	59
EE	D4	FF	36
EF	AD	D1	AE
2B	B7	BC	56

Shift Rows =

D9	36	01	59
D4	FF	36	EE
D1	AE	EF	AD
56	2B	B7	BC

Addround key =

9A	79	9E	64
53	8C	31	C2
F4	99	B6	38
59	64	64	68

Pada ronde 10 transformasi yang dilakukan hanya 3 transformasi yaitu *Subbyte*, *ShiftRows*, *Addroundkey*. Dan didapat cipherteks yang sesungguhnya yaitu:

cipherteks =

9A	79	9E	64
53	8C	31	C2
F4	99	B6	38
59	64	64	68

jika didalam bentuk ASCII maka di dapat cipherteks: š S ô Y y Œ ™ d ž 1¶ d d Â 8 h

Untuk mengubah kembali cipherteks menjadi plainteks maka dilakukan proses dekripsi dengan menggunakan transformasi invers semua transformasi dasar yang digunakan pada algoritma enkripsi AES. Setiap transformasi dasar AES memiliki transformasi invers, yaitu L: *invsubbytes*, *invshiftrows*, dan *invmixcolumns.* Dari proses dekripsi yang dilaksanakan 10 ronde didapat:

Plainteks =

4B	74	61	41
72	6F	66	45
69	67	69	53
70	72	20	00

Plainteks yang di konversi menjadi bentuk ASCII menjadi: “Kriptografi AES”. Maka pada Algoritma kriptografi AES untuk Plainteks= “Kriptografi AES” dan kunci= “Aditia” didapat Cipherteks= “š S ô Y y Œ ™ d ž 1¶ d d Â 8 h”.

4. Kesimpulan

Pada data teks, proses enkripsi dalam algoritma kriptografi AES 128, 128 bit (1 blok) plainteks terlebih dahulu dikonversi menjadi kode ASCII dalam bilangan heksadesimal dan dibentuk sebagai matriks byte berukuran 4x4 yang disebut *state*. Proses enkripsi pada AES 128 merupakan transformasi terhadap *state* secara berulang dalam 10 ronde. Data yang diproses pada setiap ronde berupa data biner. Setiap ronde AES membutuhkan satu kunci hasil generasi kunci dan menggunakan 4 transformasi dasar yaitu *subbytes, shiftrows, mixcolumns,* dan *addroundkey.* Sedangkan pada proses dekripsi mempunyai transformasi-transformasi dengan urutan *invshiftrows, invsubbytes, addroundkey,* dan *invmixcolumns.*

Pada file dokumen yang sudah dipastikan memiliki jumlah karakter lebih dari 16 karakter akan dilakukan proses enkripsi dan dekripsi setiap 128 bit atau 16 karakter. Sehingga proses enkripsi dan dekripsi AES dilakukan secara pararel. Sedangkan untuk file teks yang jumlah karakternya kurang dari 16 karakter maka akan dilakukan padding. Padding adalah penggunaan karakter ASCII null untuk mengisi jumlah karakter yang kurang agar dapat di proses dan tidak akan mempengaruhui hasil enkripsi maupun dekripsi.

Dengan bantuan MATLAB proses enkripsi dan dekripsi dapat dilaksanakan dengan cepat tepat dan efisien. Yang dibutuhkan hanya menginputkan plainteks dan kunci maka proses enkripsi dan dekripsi dapat menghasilkan *output* dengan cepat.

5. Saran

Saran Teoritis

Berdasarkan kesimpulan yang telah disebutkan, harapan penulis dimasa yang akan datang dapat ditemukan atau diteliti lebih lanjut mengenai perluasan Algoritma AES sehingga dapat mengenkripsi plainteks dengan jumlah karakter lebih dari 16 digit.

Saran Praktis

Selanjutnya, bagaimana Algoritma AES dapat diimplementasikan untuk mengenkripsi tidak hanya pada teks tetapi juga pada media lain seperti gambar, suara, dan video. Sebagai tambahan disarankannya implementasi program dapat dilakukan dengan menggunakan software selain dari MATLAB yang dapat meng akomodasi kebutuhan sistem.

Referensi

[1] Munir, Rinaldi. *Kriptografi*. Bandung : Penerbit Informatika, 2006.

[2] Sadikin, Rifki. *Kriptografi Untuk Keamanan Jaringan*. Yogyakarta: Penerbit Andi. 2012.

[3] Stalling, Wiliams. *Komputer Security And Cryptography*. New Jersey: A Jhon Wiley&Sons, Inc. 2003.

[4] E. Harahap, J. Wijekoon, R. Tennekoon, F. Yamaguchi, S. Ishida. *Modeling of Router-based Request Redirection For Content Distribution Network.* International Journal of Computer Applications, volume 76, issue 13, pp. 37-46. New York 76.13 (2013). DOI: 10.5120/13310-0857

[4] Y. Permanasari, E. Harahap. *Algoritma Data Encryption Standard (DES) Pada Electronic Code Book (ECB)*. Jurnal Matematika UNISBA, Vol. 6, No. 1. 2007. pp. 77-84.

[5] R. Tennekoon, J. Wijekoon, E. Harahap, dan H. Nishi. *Per-hop data encryption protocol for transmitting data securely over public netrwork*. Procedia Computer Science. Volume. 32. 2014. pp. 965-972. DOI: 10.1016/i.procs.2014.05.519

[6] R. Tennekoon, J. Wijekoon, E. Harahap, H. Nishi, E. Saito, S *Katsura. Per hop data encryption protocol for transmission of motion control data over public networks*. Proceeding Advanced Motion Control (AMC) on IEEE 13th International Workshop, Yokohama, Japan. 2014. Pp.128-133