Analisis dan Implementasi Algoritma AES dalam Enkripsi Suara

Shirley - 13508094
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
If18094@students.if.itb.ac.id

***Abstract*—Berkembangnya teknologi, termasuk teknologi pengiriman pesan saat ini belum diikuti dengan adanya suatu standar keamanan pengiriman pesan. Makalah ini memberikan salah satu solusi untuk keamanan pengiriman pesan suara dengan menggunakan algoritma Advanced Encryption Standard (AES). Algoritma AES merupakan algoritma yang cukup kuat, yang hingga saat ini belum ada yang melaporkan secara resmi bahwa algoritma ini telah berhasil dipecahkan. Algoritma ini merupakan algoritma enkripsi cipher blok yang melakukan enkripsi dalam bentuk blok-blok bit suara sehingga delay yang ditimbulkan cukup besar sehingga harus disesuaikan dengan menggunakan mode operasi Counter. Pengiriman pesan suara yang akan dibahas adalah pengiriman suara bersifat real-time yang berlangsung dua arah.**

***Index Terms*—Enkripsi, Advanced Encryption Standard, mode operasi Counter, real-time, pesan suara, Rijndael**

I. INTRODUCTION

Pada saat ini, dengan kemajuan teknologi yang semakin berkembang pesat, pengiriman pesan menggunakan suara sudah menjadi suatu hal yang lumrah. Hampir semua orang dari seluruh kalangan saat ini sudah menggunakan peralatan seperti handphone untuk bertukar pesan suara, selain itu juga ada orang-orang yang menggunakan *voice call* di Internet, dan sebagainya. Bahkan, saat ini komunikasi suara dapat dilakukan melalui jaringan yang lebih dikenal dengan *Voice over Internet Protocol* (VoIP).. Akan tetapi, sampai saat ini keamanan dalam bertukar pesan seperti itu masih belum terjamin sepenuhnya.

Ada beberapa cara yang bisa diterapkan untuk meningkatkan keamanan pengiriman pesan suara seperti itu. Salah satunya adalah dengan *voice scrambling*. *Voice scrambling* adalah pengubahan sinyal telekomunikasi dalam pengiriman suara sehingga tidak bisa terdeteksi dan terbaca kecuali oleh penerima yang memiliki alat khusus untuk menerima sinyal tersebut. Hal ini kurang lebih sama dengan cara penggunaan radio, yang bisa menerima sinyal hanyalah beberapa penerima yang memang sedang dipasang ke frekuensi siaran tersebut. Walaupun demikian, teknik ini masih memiliki tingkat keamanan yang cukup rendah.

Cara yang lainnya yang bisa digunakan adalah dengan menggunakan enkripsi pada pesan suara yang akan dikirim. Enkripsi suara ini dilakukan sebelum pesan suara dikirim sehingga walaupun ada pihak ketiga yang berhasil mendapatkan pesan suara tersebut, pesan suara tersebut tidak bisa dipahami lagi artinya. Enkripsi pesan suara ini biasanya dilakukan menggunakan suatu alat atau aplikasi pengenkripsi

Ada berbagai macam algoritma enkripsi dengan karakteristiknya masing-masing yang bisa digunakan untuk melakukan enkripsi pesan suara. Akan tetapi, karena belum ada standar pasti yang ditetapkan untuk enkripsi pesan suara tersebut, diperlukan upaya ekstra jika ingin mengimplementasikan algoritma baru untuk enkripsi suara tersebut. Biasanya algoritma yang digunakan untuk enkripsi pesan suara adalah algoritma cipher aliran karena pengiriman pesan suara seperti itu bersifat *real-time*.

Advanced Encryption Standard (AES) adalah algoritma kriptografi yang dikembangkan oleh dua orang Belgia bernama Joan Daemen dan Vincent Rijmen dibawah nama Rijndael. AES sampai saat ini digunakan sebagai standar enkripsi dengan kunci-simetris yang diadopsi oleh pemerintah Amerika Serikat [1]. Hingga saat ini, masih belum ada laporan resmi bahwa algoritma ini berhasil dipecahkan orang. Dengan kata lain, algoritma ini sangat aman karena hingga saat ini belum ada yang berhasil memecahkannya.

Algoritma AES merupakan algoritma cipher blok, hal ini merupakan hambatan jika diterapkan pada enkripsi komunikasi suara. Algoritma cipher blok beroperasi dalam bentuk blok bit, yang dalam hal ini rangkaian bit dibagi menjadi blok-blok bit yang panjangnya sudah ditentukan sebelumnya [2]. Oleh karena itu, delay yang akan ditimbulkan menjadi besar karena harus menunggu data-data sejumlah blok tersebut. Untuk memperkecil delay-nya, harus dilakukan penyesuaian pada algoritma ini.

Salah satu cara yang dapat digunakan adalah dengan menyesuaikan mode operasi yang digunakan. Saat ini, mode operasi yang banyak digunakan pada algoritma AES adalah *Cipher Block Chaining* (CBC). Tetapi mode operasi ini tidak akan meningkatkan kecepatan enkripsi AES karena enkripsi dilakukan secara sekuensial. Salah

satu mode operasi yang dapat digunakan untuk mengubah kecepatan dan efisiensi enkripsi cipher blok menjadi menyerupai cipher aliran adalah mode operasi counter. Oleh karena itu, pada tugas akhir ini dipilih penerapan. Algoritma AES dengan mode operasi yang disesuaikan menjadi mode operasi counter untuk melakukan enkripsi pada aliran pesan suara dalam dua arah.

II. KRIPTOGRAFI

Kriptografi adalah suatu teknik yang digunakan untuk menjamin aspek keamanan dari pertukaran data, seperti kerahasiaan data, kebenaran data, integritas data, serta autentikasi data [2]. Untuk menjamin keamanan pertukaran data, dapat dilakukan dengan berbagai cara, salah satunya adalah dengan proses penyandian dengan menggunakan algoritma sandi. Proses penyandian dilakukan agar data yang dikirim tidak dapat dimengerti oleh pihaklain selain yang memiliki akses terhadap data tersebut. Dalam proses penyandian terdapat dua konsep utama yaitu enkripsi dan dekripsi.

Enkripsi adalah proses yang mengubah data atau informasi yang akan dikirim menjadi bentuk yang hampir tidak dikenali sebagai informasi awalnya. Enkripsi biasanya dilakukan sebelum data atau informasi tersebut dikirimkan. Dalam kriptografi, data atau informasi yang dapat dimengerti maknanya dikenal dengan plainteks (plainteks) atau teks-jelas (cleartext) sedangkan informasi yang telah tersamarkan tersebut dikenal dengan cipherteks (cipherteks) [2]. Untuk meningkatkan keamanan enkripsi informasi, pada proses enkripsi tersebut ditambahkan kunci. Dekripsi adalah kebalikan dari enkripsi.

Kunci yang digunakan untuk melakukan enkripsi dan dekripsi bisa sama atau berbeda. Jika kunci yang digunakan berbeda, dikenal dengan kriptografi kunci publik. Sebaliknya, jika kunci yang digunakan sama, disebut juga kriptografi kunci simetri. Dalam makalah ini digunakan mekanisme kunci simetri.

A. Mode Operasi Cipher Blok

Cipher blok adalah algoritma yang datanya dibagi kedalam blok-blok berukuran sama dan proses enkripsi dilakukan pada setiap blok tersebut. Dalam cipher blok masih ada kemungkinan dihasilkannya suatu cipherteks yang sama dari plainteks yang sama yang akan mengurangi tingkat keamanan algoritma enkripsi. Oleh karena itu dibutuhkan suatu mekanisme tambahan untuk meningkatkan tingkat keamanannya yaitu dengan penggunaan berbagai mode operasi dalam cipher blok.

Mode operasi yang sering digunakan oleh algoritma cipher blok adalah CBC (Cipher Block Chaining). Dalam CBC, plainteks di-XOR dengan cipherteks dari blok sebelumnya, kemudian hasilnya dimasukkan ke dalam algoritma enkripsi dan menghasilkan cipherteks. Kelemahan dari mode operasi CBC adalah prosesnya yang sekuensial. Untuk melakukan enkripsi dari suatu blok harus menunggu cipherteks dari hasil enkripsi blok yang sebelumnya. Sehingga hal ini bisa menimbulkan penundaan (*delay*).

Salah satu mode operasi cipher blok yang bisa beroperasi seperti cipher aliran adalah mode operasi *Counter*. Mode operasi ini menghasilkan sebuah blok *keystream* dengan cara mengenkripsi nilai dari sebuah fungsi penghitung ("*counter*"). Counter ini berupa fungsi yang menghasilkan suatu rangkaian nilai yang pasti berbeda satu dengan yang lain untuk waktu yang lama. Dengan kata lain, untuk semua enkripsi blok dengan masukan suatu kunci tertentu, nilai counter yang dihasilkan selalu unik.

B. Algoritma AES

Algoritma AES adalah algoritma enkripsi cipher blok dengan kunci simetris yang bisa memproses blok-blok data 128 bit, menggunakan kunci cipher berukuran 128, 192, dan 256 bit.

Garis besar algoritma AES yang beroperasi blok 128-bit dengan kunci 128-bit adalah sebagai berikut [1]:

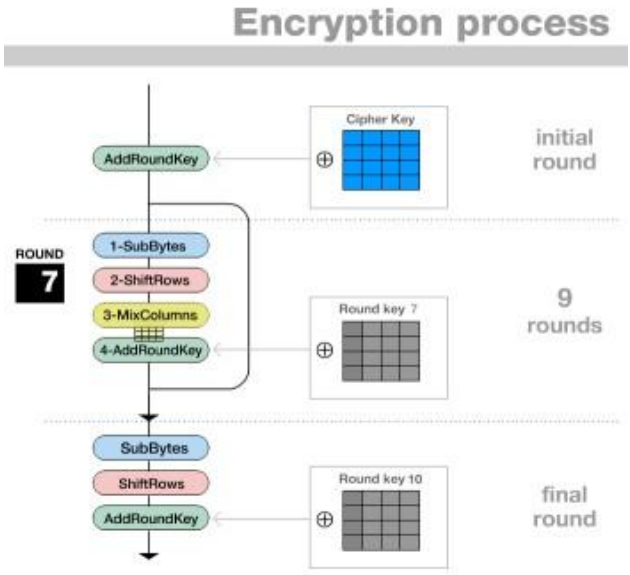
1. AddRoundKey: melakukan XOR antara state awal (plainteks) dengan cipher key. Tahap ini disebut juga initial round.
2. Putaran sebanyak Nr – 1 kali. Proses yang dilakukan pada setiap putaran adalah:
    a. SubBytes: substitusi byte dengan menggunakan tabel substitusi (Sbox). Tabel substitusi dapat dilihat pada tabel 1, sedangkan ilustrasi ByteSub dapat dilihat pada gambar 2.
    b. ShiftRows: pergeseran baris-baris array state secara wrapping. Ilustarsi ShiftRow dapat dilihat pada gambar 4.
    c. MixColumns: mengacak data di masing-masing kolom array state. Ilustarsi MixColumn dapat dilihat pada gambar 5.
    d. AddRoundKey: melakukan XOR antara state sekarang dengan round key. Ilustarsi AddRoundKey dapat dilihat pada gambar 6.
3. Final round: proses untuk putaran terakhir:
    a. SubByte.
    b. ShiftRow.
    c. AddRoundKey.

Diagram proses enkripsi AES dapat dilihat pada Gambar 1. Algoritma AES mempunyai 3 parameter sebagai berikut:

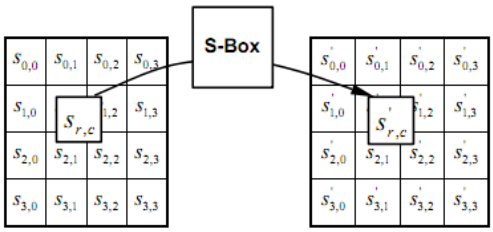
- Plainteks : array yang berukuran 16 byte, yang berisi data masukan.
- Cipherteks : array yang berukuran 16 byte, yang berisi hasil enkripsi.
- Key : array yang berukuran 16 byte, yang berisi kunci ciphering (disebut juga cipher key).

Dengan 16 byte, maka baik blok data dan kunci yang berukuran 128-bit dapat disimpan di dalam ketiga array tersebut (128 = 16 x 8).

Selama kalkulasi plainteks menjadi cipherteks, status sekarang dari data disimpan di dalam array of byte dua dimensi, state, yang berukuran NROWS x NCOLS. Elemen array state diacu sebagai S[r,c], dengan $0 \leq r < 4$ dan $0 \leq c < Nc$ (Nc adalah panjang blok dibagi 32). Pada AES, Nc = 128/32 = 4.



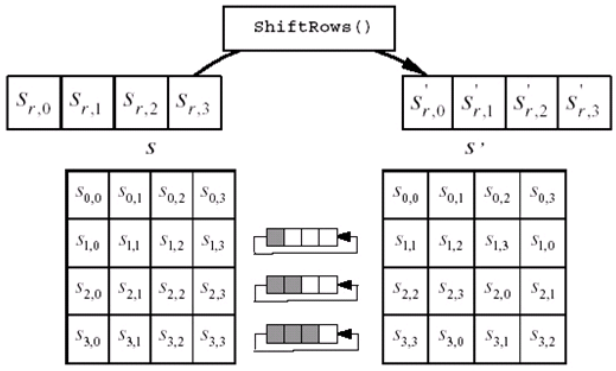
Gambar 1 – Diagram Proses Enkripsi AES



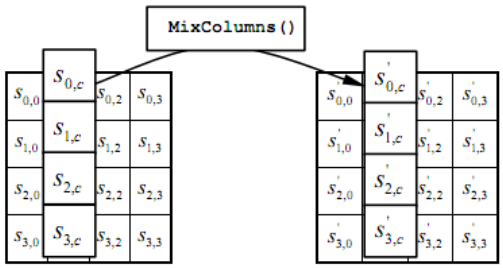
Gambar 2 – Ilustrasi Transformasi SubBytes() pada AES

x \ y	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

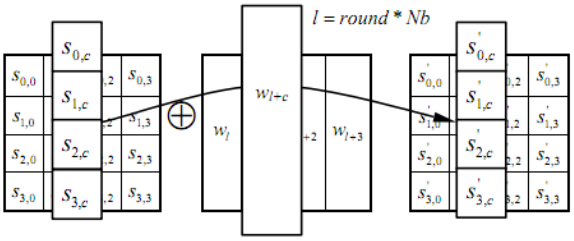
Gambar 3 – Tabel Sbox yang digunakan pada SubBytes()



Gambar 4 – Ilustrasi Transformasi ShiftRows() pada AES



Gambar 5 – Ilustrasi Transformasi MixColumns() pada AES



Gambar 6 – Ilustrasi Transformasi AddRoundKey() pada AES

III. ANALISIS

Permasalahan yang akan diselesaikan dalam pelaksanaan tugas akhir ini adalah menerapkan algoritma AES agar dapat digunakan untuk melakukan enkripsi suara dalam proses pengiriman pesan suara antar dua buah komputer melalui jaringan. Pengiriman suara bersifat dua arah.

Suara dimasukkan melalui dua sumber, yaitu *microphone* dan *file audio*. Untuk masukan dari *microphone* dibutuhkan suatu mekanisme digitalisasi. Kemudian dilakukan proses kompresi untuk memperkecil ukuran untuk dimasukkan ke dalam proses enkripsi dan pengiriman. Pengiriman dilakukan dengan menggunakan paket-paket data. Proses yang ada dalam penerima merupakan kebalikan proses dari pengirim.

Algoritma AES digunakan untuk enkripsi aliran pesan suara dengan mengubah mode operasi yang digunakan hingga karakteristiknya menyerupai cipher aliran, yaitu dengan mode operasi Counter.

Cara untuk membangkitkan blok counter yaitu [3] :

1. Dari satu blok counter awal (T1), akan diterapkan fungsi penambah untuk membangkitkan blok counter selanjutnya
2. Blok counter akan terbagi menjadi dua bagian, yaitu message nonce dan bit yang akan bertambah (increment). Message nonce akan diambil dari angka acak.
3. Fungsi penambah yang digunakan, didasarkan pada definisi yang diberikan oleh *National Institute of Standards and Technology* (NIST), yaitu: $[X]_m = [X + 1 \bmod 2^m]_m$
   m = jumlah bit dalam fungsi penambah

Proses dekripsi dengan mode operasi counter membutuhkan masukan blok counter yang digunakan pada proses enkripsi. Oleh karena itu, blok counter yang digunakan dalam proses enkripsi akan ikut dikirimkan bersama dengan cipherteks hasil enkripsi.

IV. IMPLEMENTASI

Pada implementasinya, ada beberapa tahap yang harus dilakukan untuk pengenkripsian pesan suara tersebut, yaitu :

A. Pendigitalisasian Suara

Suara yang ditangkap melalui alat seperti *microphone* harus diubah menjadi digital data byte sebelum bisa dienkripsi [4]. Setelah diubah menjadi data digital, barulah pesan suara tersebut bisa dienkripsi dan dikirim.

Cara paling mudah yang bisa digunakan untuk pendigitalisasian suara adalah dengan memasukkannya ke suatu buffer sementara dan mengubahnya menjadi deretan byte di sana, kemudian memasukkannya ke blok-blok untuk dienkripsi.

Cara pengubahan suara menjadi byte tergantung dari frekuensi suara tersebut. Setiap suara yang dihasilkan manusia memiliki frekuensi yang unik, sehingga ketika kita walaupun kita mengubahnya menjadi byte, suara tersebut tidak akan rusak, karena pada dasarnya pengubahan didasarkan pada frekuensi suara tersebut.

B. Proses Enkripsi

Setelah data diubah menjadi byte, kemudian data-data tersebut dimasukkan ke dalam blok-blok untuk dienkripsi. Ukuran setiap blok ditentukan dari proses enkripsi yang akan digunakan, bisa 128, 192, ataupun 256 bit per blok.

Setelah itu, proses enkripsi akan dilakukan, dimulai dari AddRoundKey yang dilakukan dengan cara melakukan XOR pada blok plainteks dengan cipher kunci yang dimasukkan.

Kemudian, akan dilakukan transformasi SubBytes menggunakan Sbox. Pada transformasi ini, setiap byte yang ada pada blok tersebut akan digantikan sesuai dengan tabel Sbox yang ada.

Selanjutnya dilakukan transformasi ShiftRows, yaitu penggantian baris misalnya dimajukan satu-satu setiap kolom, dan sebagainya. Tujuan dari hal ini adalah untuk semakin mengacak blok data yang dienkripsi sehingga semakin sulit untuk dipecahkan.

Jika hal tersebut sudah selesai, maka transformasi MixColumns pun dilakukan, yaitu dengan mengubah-ubah letak-letak kolom pada suatu blok secara acak.

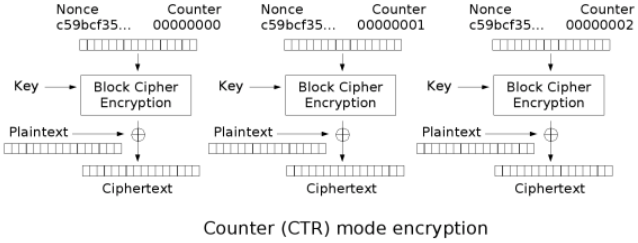
Kemudian, enkripsi akan berulang ke bagian SubBytes, berlanjut ke ShiftRows, dan seterusnya hingga pengulangan yang ditentukan. Misalnya pengulangannya sebanyak 9 kali, maka sistem akan mengulangnya sebanyak 7 kali (1 putaran awal bersama AddRoundKey dan 1 putaran akhir tidak dihitung).

Jika putaran yang dilakukan sudah n-1, maka proses enkripsi hanya akan melakukan SubBytes, ShiftRows, dan AddRoundKey sebagai satu putaran terakhir. Hasilnya akan dikeluarkan sebagai cipherteks.

Akan tetapi, proses tersebut dilakukan jika yang digunakan hanyalah algoritma AES. Jika menggunakan cara seperti itu, akan terjadi delay yang cukup lama karena suara harus dimasukkan ke buffer, dienkripsi per blok, baru dikirimkan. Hal tersebut terjadi jika kita menggunakan mode operasi biasa seperti misalnya CBC.

Pada makalah ini, kita akan membahas implementasi untuk mode operasi counter. Mode operasi counter dibuat menyerupai mode cipher aliran sehingga bisa mengurangi delay waktu yang terjadi ketika kita menggunakan mode operasi cipher blok yang biasa [5].

Jika menggunakan mode operasi counter, pendiitalisasian suara dilakukan hampir sama dengan mode operasi cipher blok yang biasa, yaitu suara dimasukkan ke dalam buffer kemudian di ubah ke byte dan dienkripsi lalu dikirimkan per blok. Yang membedakan mode operasi counter ini adalah pada saat blok pertama dbuat dan diisi dengan data yang akan dikirim, blok selanjutnya sudah dibangkitkan. Dengan kata lain, akan terjadi pembangkitan dua blok secara paralel. Ketika blok pertama sedang mengenkripsi, blok kedua sudah diisi. Jadi ketika blok pertama sedang dalam pengiriman, blok kedua sudah tahap enkripsi sehingga ketika pengiriman blok pertama selesai, blok kedua sudah siap kirim. Dengan demikian, delay waktu yang terjadi hanyalah di awal pengiriman saja, sedangkan untuk seterusnya, delay waktu tidak akan terlalu lama lagi karena pengiriman tidak harus menunggu blok hasil enkripsi lagi.



Gambar 7 – Diagram Proses Enkripsi Mode Counter

Untuk melakukan mode counter seperti itu, kita membutuhkan message nonce dan counter bit yang bertambah (increment) yang dimasukkan ke cipher blok. Setelah itu, barulah blok cipher tesebut dimasukkan ke proses enkripsi AES untuk diproses, yang menghasilkan cipherteks.

Beberapa algoritma untuk fungsi-fungsi utama enkripsi :

```
Function AddRounKey(state, w, Nr,
Nb)
Begin
   Int r, c
   for (r=0,r<4,r++)
       for(c=0,c<Nb,c++)
             state[r][c] =
w[Nr*4+c][r];
       end for
   end for
   return state
end
```

```
Function SubBytes(s, Nb)
Begin
   Int r, c
   for (r=0,r<4,r++)
       for(c=0,c<Nb,c++)
             s[r][c] = Sbox[s[r][c]];
       end for
   end for
   return s
end
```

```
Function ShiftRows(s, Nb)
Begin
   Int r, c
   for (r=0,r<4,r++)
       for(c=0,c<4,c++)
             t[r][c] = s[r][(c+r)%Nb]
       end for
       for (c=0,c<4,c++)
             s[r][c] = t[c]
       end for
   end for
   return s
end
```

```
Function MixColumn(s, Nb)
Begin
   Int r, c
   for (r=0,r<4,r++)
       a = array(4)
       b = array(4)
       for(c=0,c<Nb,c++)
             a[r] = s[r][c]
             if (s[r][c]<<1)
                   b[r] = s[r][c]
             end if
       end for
   end for
   return s
```

```
Enkripsi(byte byte)
Begin
   Nb = 4
   Nk = lengthkey() / 4
   W = array(Nb*(Nr+1))
   byte state
   nonce(byte)
   counterBit(byte)
   state = byte
   AddRoundKey(state, w, Nr, Nk)

   for pengulangan downto 1
       InvSubBytes(state, Nb)
       InvShiftRows(state, Nb)
       InvMixColumns(state, Nb)
       AddRoundKey(state, w, Nr, Nk)
   end for

   InvSubBytes(state, Nb)
   InvShiftRows(state, Nb)
   AddRoundKey(state, w, Nr, Nk)
end
```

Untuk melakukan dekripsi, cukup membalikkan proses enkripsi saja, tidak ada fungsi-fungsi yang harus ditambah atau diubah.

V. KESIMPULAN

Kesimpulan yang dapat diambil dari hasil analisis enkripsi suara menggunakan AES ini adalah :

1. Enkripsi suara menggunakan algoritma AES biasa tanpa menggunakan mode operasi counter akan menghasilkan delay waktu yang cukup lama
2. Enkripsi suara menggunakan algoritma AES biasadengan menggunakan mode operasi counter tetap akan menghasilkan delay waktu, tetapi hanya di awal saja sehingga bisa mengurangi delay yang seharusnya terjadi
3. Walaupun algoritma enkripsinya cukup kuat dan sulit untuk dipecahkan, tetapi dengan adanya delay waktu di awal, penggunaan algoritma AES untuk mengenkripsi pesan suara yang real-time masih kurang mangkus dibandingkan dengan algoritma cipher aliran.

REFERENCES

[1] *Announcing the Advanced Encryption Standard* , URL : http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[2] Munir, Rinaldi, (2007), Bahan Kuliah IF5054 Kriptografi. Program Studi Teknik Informatika, Institut Teknologi Bandung.

[3] Dworkin, Morris. 2001. *Recommendation for Block Cipher Mode of Operation.* NIST

[4] *Implementatiion of Real-Time Voice Encryption System* , URL : http://upcommons.upc.edu/pfc/bitstream/2099.1/4858/1/MarkusBrandau.pdf

[5] *Counter with CBC-MAC* , URL : http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/ccm.pdf

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, 22 Maret 2010

Shirley - 13508094