Studi dan Implementasi Enkripsi Pengiriman Pesan Suara dengan Algoritma *Serpent*

Anggi Alisia Putri

Laboratorium Rekayasa dan Komputasi
Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung
e-mail: if15096@students.if.itb.ac.id

Abstrak – *Banyaknya jenis komunikasi suara saat ini masih belum diikuti dengan adanya suatu standar keamanan. Makalah ini memberikan salah satu solusi untuk keamanan pengiriman pesan suara dengan menggunakan algoritma Serpent. Algoritma Serpent merupakan algoritma yang cukup kuat dalam hal keamanan dan memiliki tipe enkripsi blok cipher 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 terjadi berlangsung dua arah antara pengirim dan penerima.*

Kata Kunci*: Enkripsi, Serpent, mode operasi Counter.*

1. PENDAHULUAN

Saat ini komunikasi suara menjadi hal yang penting dalam kehidupan sehari-hari, seperti komunikasi suara dengan telepon yang berbasis analog dan telepon seluler yang berbasis digital. Bahkan, saat ini komunikasi suara dapat dilakukan melalui jaringan yang lebih dikenal dengan *Voice over Internet Protocol* (VoIP). Berbagai alat komunikasi yang ada saat ini belum tentu aman untuk digunakan, karena belum ada standar keamanan yang dapat digunakan oleh alat-alat tersebut. Oleh karena itu, komunikasi ini sangat rentan terhadap serangan pihak ketiga yang seringkali sangat merugikan.

Salah satu solusi yang ditawarkan untuk permasalahan ini adalah *voice scrambling*, yaitu perubahan pada sinyal telekomunikasi agar tidak dapat diketahui oleh siapapun selain pihak yang memiliki alat penerima khusus. Namun teknik ini masih memiliki tingkat keamanan yang sangat rendah.

Solusi lain yang ditawarkan adalah enkripsi suara yang memiliki tingkat keamanan yang lebih tinggi. Enkripsi pada data digital ini dilakukan sebelum data dikirimkan sehingga pihak ketiga tidak dapat memahami arti dari data yang berhasil diambilnya. Proses enkripsi ini biasanya dilakukan oleh alat atau aplikasi pengenkripsi.

Ada berbagai macam algoritma enkripsi dengan karakteristiknya masing-masing yang dapat digunakan untuk proses enkripsi suara. Karena belum ada standar tertentu yang dapat digunakan, diperlukan usaha untuk menerapkan algoritma lain untuk mengetahui sebaik apa algoritma tersebut. Untuk proses enkripsi komunikasi suara yang bersifat real time biasanya digunakan algoritma cipher aliran untuk mempercepat prosesnya.

Algoritma Serpent merupakan algoritma yang menempati urutan kedua pada kompetisi *Advance Encryption Standard* (AES). Algoritma Serpent merupakan algoritma kuat yang sampai saat ini belum ada laporan serangan dari kriptanalis yang berhasil merusaknya. Algoritma ini juga tidak dipatenkan, sehingga penggunaannya pada alat pengenkripsi tidak memerlukan biaya. Algoritma ini juga dapat bekerja baik pada *hardware* dan *smart card* [SER09].

Algoritma Serpent 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 [RIN07]. 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 Serpent adalah *Cipher Block Chaining* (CBC) [SER09]. Tetapi mode operasi ini tidak akan meningkatkan kecepatan enkripsi Serpent 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 Serpent dengan mode operasi yang disesuaikan menjadi mode operasi counter untuk melakukan enkripsi pada aliran pesan suara dalam dua arah.

2. KRIPTOGRAFI

Kriptografi adalah suatu teknik yang digunakan untuk menjamin aspek keamanan dari pertukaran data, seperti kerahasiaan data, kebenaran data, integritas data, serta autentikasi data [RIN07]. 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 pihak lain 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) [RIN07]. 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.

2.1 Mode Operasi Cipher Blok

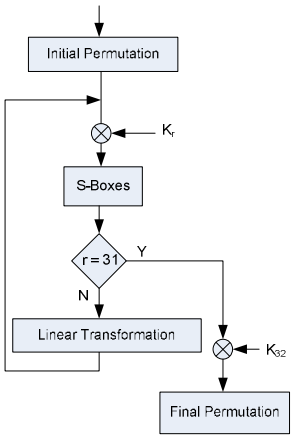
Cipher blok adalah algoritma yang datanya dibagi kedalam blok-blok berukuran sama dan proses enkripsi dilakukan pada setiap blok tersebut. Dalam cipher blok masih ada kemungkinan dihasilkannya suatu cipherteks yang sama dari plainteks yang sama yang akan mengurangi tingkat keamanan algoritma enkripsi. 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).

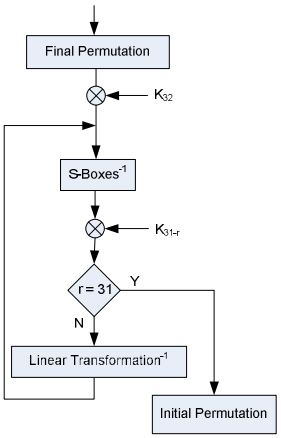
Salah satu mode operasi cipher blok yang bisa beroperasi seperti cipher aliran adalah mode operasi Counter. Mode operasi ini menghasilkan sebuah blok keystream dengan cara mengenkripsi nilai dari sebuah fungsi penghitung ("counter"). Counter ini berupa fungsi yang menghasilkan suatu rangkaian nilai yang 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.

2.2 Algoritma Serpent

Serpent merupakan algoritma cipher blok yang memiliki ukuran blok sebesar 128 bit dan mendukung ukuran kunci sebesar 128, 192, atau 256 bit. Cipher ini berbentuk Substitution-Permutation Network (SP-network) yang merupakan rangkaian operasi-operasi matematis yang saling berhubungan. SP-network memiliki S-boxes dan P-boxes yang mengubah blok bit masukan menjadi suatu bit keluaran.



Gambar 1 Proses Enkripsi



Gambar 2 Proses Dekripsi

Serpent mendukung masukan kunci sepanjang 128 bit, 192 bit, dan 256 bit. Kenyataannya, dalam mekanisme penjadwalan kunci dibutuhkan kunci sepanjang 256 bit. Oleh karena itu, untuk masukan kunci sepanjang 128 bit dan 192 bit memerlukan mekanisme tambahan, yaitu padding. Padding menambahkan bit "1" pada bit terpenting (most significant bit) dan beberapa bit "0" sampai ukuran kunci mencapai 256 bit.

Untuk proses enkripsi, Serpent membutuhkan 32 upakunci 128 bit yang dinotasikan dengan K0, … , K32. Tahapan untuk mendapatkan ke-33 upakunci yaitu [SER09]:

1. Membagi kunci masukan K menjadi delapan bagian, masing-masing 32 bit yang dinotasikan dengan w_{-8}, … , w_{-1}
2. Membentuk 132 kunci antara (*prekey*) yang dinotasikan dengan w_0, … , w_{131} melalui persamaan:

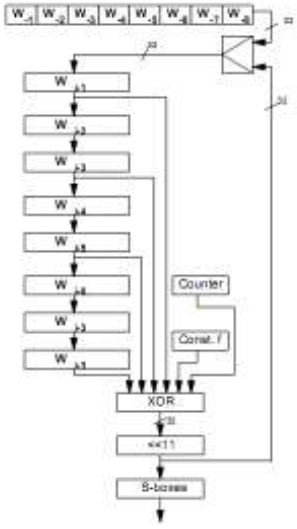
   $$w_i = (w_{i-8} \oplus w_{i-5} \oplus w_{i-3} \oplus w_{i-1} \oplus Ø \oplus i) <<< 11$$

   Notasi Ø merupakan bagian kecil dari *golden ratio* ($\sqrt{5}$ + 1) / 2 atau 0x9e3779b9 dalam heksadesimal.
3. Membentuk 132 kunci putaran (*round key*) k_0 sampai k_{131} yang dibentuk dari kunci antara yang dihasilkan dari proses sebelumnya dengan menggunakan *S-boxes*. *S-boxes* digunakan untuk mengubah kunci antara w_i menjadi k_i dengan ketentuan berikut ini :

   $$\{k_0, k_1, k_2, k_3\} = S_3 (w_0, w_1, w_2, w_3)$$
   $$\{k_4, k_5, k_6, k_7\} = S_2 (w_4, w_5, w_6, w_7)$$

$$\{k_8, k_9, k_{10}, k_{11}\} = S_1 (w_8, w_8, w_{10}, w_{11})$$
$$\{k_{12}, k_{13}, k_{14}, k_{15}\} = S_0 (w_{12}, w_{13}, w_{14}, w_{15})$$
. . .
$$\{k_{124}, k_{125}, k_{126}, k_{127}\} = S_4 (w_{124}, w_{125}, w_{126}, w_{127})$$
$$\{k_{128}, k_{129}, k_{130}, k_{131}\} = S_3 (w_{128}, w_{129}, w_{130}, w_{131})$$

Pembentukan kunci putaran untuk tahap (1) sampai tahap (3) dapat digambarkan dalam gambar 3.



Gambar 3 Pembentukan kunci putaran [BOSZC]

4. Membentuk upakunci 128 bit K_i (untuk I € {0, … , 32}) dari 32 bit nilai k_j dengan cara:
$$K_i = \{k_{4i}, k_{4i+1}, k_{4i+2}, k_{4i+3}\}$$

5. Menerapkan IP pada upakunci yang dihasilkan untuk menempatkan bit-bitnya ke dalam urutan yang sesuai.
$$\hat{K}_i = IP\ (K_i)$$

3. ANALISIS

Permasalahan yang akan diselesaikan dalam pelaksanaan tugas akhir ini adalah menerapkan algoritma Serpent 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 Serpent digunakan untuk enkripsi aliran pesan suara dengan mengubah mode operasi yang digunakan hingga karakteristiknya menyerupai cipher aliran, yaitu dengan mode operasi Counter.

3.1 Penerapan blok Counter

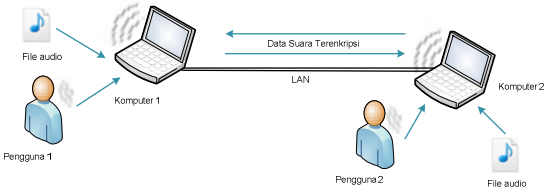
Cara untuk membangkitkan blok *counter* yaitu [DWO01] :

1. Dari satu blok counter awal (T_1), 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

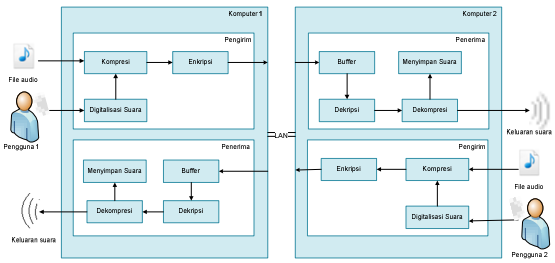
Proses dekripsi dengan mode operasi counter membutuhkan masukan blok *counter* yang digunakan pada proses enkripsi. Oleh karena itu, blok *counter* yang digunakan dalam proses enkripsi akan ikut dikirimkan bersama dengan cipherteks hasil enkripsi.

3.2 Analisis Perangkat Lunak

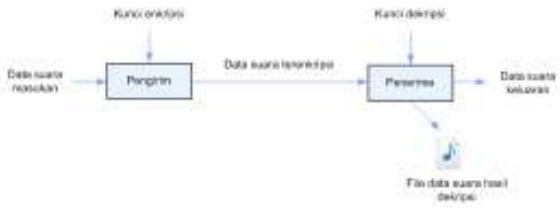
Pengirim akan menerima masukan data suara melalui *microphone* kemudian diubah menjadi bit digital. Untuk memperkecil ukuran data suara yang akan diproses, dilakukan proses kompresi. Hasil dari kompresi ini nantinya akan dienkripsi dengan algortima Serpent dan dikirimkan melalui kabel jaringan. Penerima akan melakukan dekripsi terhadap data suara yang dikirimkan. Hasil dekripsi kemudian di didekompresi dan dikeluarkan melalui speaker agar dapat didengarkan kembali. Masukan suara selain berasal dari microphone juga bisa berasal dari suatu file audio dengan format wav. Secara umum, arsitektur global sistem tampak pada gambar 4. Untuk pengiriman suara dalam dua arah, pengirim dan penerima berada di satu komputer dalam satu waktu dan berbagi sumber daya komputer untuk menjalankan kedua fungsi tersebut secara bersamaan.



Gambar 4 Arsitektur Global Sistem



Gambar 5 Arsitektur Detail Sistem

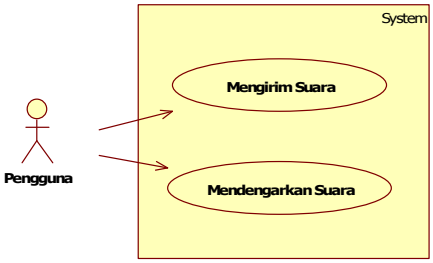


Gambar 6 Skema Pengiriman satu arah

4. PERANCANGAN PERANGKAT LUNAK

4.1 Diagram use case

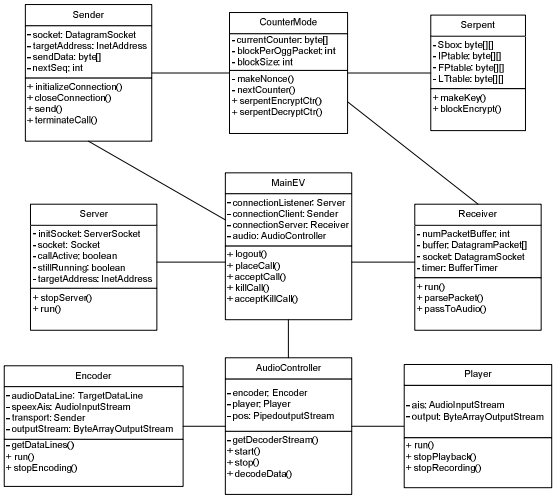
Dalam sistem ini, pengguna dapat melakukan dua hal, yaitu mengirimkan suara dan mendengarkan suara yang dikirimkan. Diagram use case dapat dilihat pada gambar 7.



Gambar 7 diagram use case

4.2 Diagram Kelas

Identifikasi kelas dilakukan berdasarkan analisis kelas. Terdapat sembilan kelas utama pada perangkat lunak ini, yaitu kelas *Sender*, *CounterMode, Serpent, Server, MainEV, Receiver, Encoder, AudioController,* dan *kelas Player*. Rancangan diagram kelas dapat dilihat pada gambar 8.



Gambar 8 Diagram Kelas

5. IMPLEMENTASI

Pada tugas akhir ini, perangkat lunak yang dikembangkan untuk melakukan pengiriman pesan suara memiliki batasan sebagai berikut:

1. Perangkat lunak hanya melibatkan dua komputer.
2. Proses digitalisasi dan kompresi sinyal suara tidak diimplementasikan, tetapi menggunakan *library* dan API yang telah tersedia.
3. *File* audio masukan memiliki format *encoding* yang sama dengan aplikasi, yaitu dengan nilai *sample* 8000Hz, ukuran *sample* 16 bit, dan *channel* mono.
4. Jenis masukan hanya bisa berasal dari satu sumber, yaitu dari *microphone* atau *file* audio, tidak dapat dilakukan secara bersamaan dalam satu komputer.

5.1 Implementasi Kelas

Kelas-kelas yang telah dirancang diimplementasikan dengan menggunakan bahasa pemrograman Java. Pada tabel di bawah dapat dilihat daftar implementasi kelas-kelas yang ada pada perangkat lunak beserta keterangannya.

Tabel 1 Penjelasan Kelas Implementasi

Nama kelas	Nama File	Keterangan
MainEV	`MainEV.java`	mengimplementasikan kelas java.awt.event.ActionListener
Server	`Server.java`	mengimplementasikan kelas java.awt.event.ActionListener
Sender	`Sender.java`	Fungsi random yang digunakan untuk meng-isi nomor urutan meng-gunakan fungsi random dari java.util.Random
Receiver	`Receiver.java`	mengimplementasikan kelas java.awt.event.ActionListener
Counter Mode	`CounterMode.java`	
Serpent	`Serpent.java`	
AudioController	`AudioController.java`	menggunakan library JSpeex dan API Java Sound untuk proses sinyal suara
Player	`Player.java`	merupakan turunan dari kelas Thread yang menggunakan API Java Sound untuk proses sinyal suara
Encoder	`Encoder.java`	merupakan turunan dari kelas Thread yang menggunakan library JSpeex dan API Java Sound untuk proses sinyal suara

5.2 Implementasi Antarmuka

Antarmuka perangkat lunak dibangun dengan menggunakan IDE Netbeans 6.5.1 yang memiliki

peralatan untuk membuat *Graphical User Interface* (GUI). Antarmuka perangkat lunak ini terdiri dari satu layar utama dan empat layar tambahan, yaitu layar awal, layar peringatan, layar notifikasi penerimaan panggilan, dan layar status komunikasi.

Gambar 9 Layar Awal

Gambar 10 Layar Utama

Gambar 11 Layar Notifikasi Penerimaan Panggilan

Gambar 12 Layar Status Komunikasi

Gambar 13 Layar Peringatan

6. PENGUJIAN

Pengujian pada perangkat lunak bertujuan untuk:

1. Menguji kebenaran hasil enkripsi pada *byte* data suara dan proses dekripsi *byte* data suara.
2. Mengetahui kinerja perangkat lunak yang telah dibuat dalam proses pengiriman dan penerimaan *byte* data suara.

Berdasarkan pengujian yang dilakukan, proses enkripsi dan dekripsi yang dilakukan dapat terbukti kebenarannya, karena hasil sebelum enkripsi dan sesudah dekripsi sama.

Pengujian kinerja perangkat lunak mendapatkan bahwa waktu tunda yang dihasilkan cukup besar, yaitu 1,4 detik untuk masukan dari *microphone* dan 569 sampai 380 milidetik untuk masukan dari *file* audio. Dalam hal ini sumber daya komputer yang digunakan sangat berpengaruh karena mekanisme pengirim dan penerima berada dalam satu komputer yang dijalankan secara bersamaan (komunikasi dua arah).

7. KESIMPULAN

Kesimpulan yang dapat diambil dari setelah pambahasan tentang penerapan algoritma Serpent pada aliran pesan dua arah melalui jaringan adalah:

1. Algoritma Serpent merupakan algoritma yang dapat diterapkan untuk melakukan enkripsi aliran pesan suara dengan cukup baik setelah mengalami modifikasi pada mode operasinya. Namun ada beberapa hal yang harus diperhatikan:
   a. Data suara yang dienkripsi harus diambil secara bertahap agar tetap menjaga property *real time*
   b. Proses enkripsi dan dekripsi merupakan proses yang sama karena menggunakan mode Counter, tidak perlu menggunakan proses dekripsi seperti Serpent pada umumnya
2. Kualitas suara setelah mengalami kompresi dan enkripsi tetap memiliki kualitas yang cukup baik, dan distorsi yang ada dapat diacuhkan.
3. Delay yang dihasilkan cukup besar karena proses pengiriman dan penerimaan dilakukan di satu komputer pada waktu bersamaan.
4. Proses pengiriman paket audio dari pengirim ke penerima dapat dijamin keamanannya dari modifikasi byte yang terjadi di tengah proses pengiriman.
5. Keberlangsungan komunikasi tetap dapat terjaga walaupun terjadi kerusakan atau hilangnya beberapa paket saat pengiriman.
6. Kecepatan pengiriman suara tidak dipengaruhi oleh besarnya data suara masukan.

REFERENSI

[BLE01] Blelloch, Guy E. 2001. *Introduction to Data Compression*. Carnegie Mellon University.

[BOSZC] Bora, Piotr, Tomasz Czakka. *Implementation of Serpent Algorithm Using Altera FPGA Devices*. Military Communication Institute.

[CAV] Cavagnolo, J.Bier. *Introduction to Digital Audio Compression*. Berkeley Design Technology.

[CISDL] Understanding Delay in Packet Voice Networks. URL: http://www.cisco.com/en/US/tech/tk652/tk698/technologies_white_paper09186a00800a8993.shtml

[DAVCS] Peters, Davidson. *VOIP Fundamental*. Cisco System.

[DWO01] Dworkin, Morris. 2001. *Recommendation for Block Cipher Mode of Operation*. NIST Centennials.

[MOH02] Moh, T. 2002. AES is not Broken. URL: http://www.usdsi.com/aes.html

[RAT07] Ratih. (2007). Tugas Akhir : Studi dan Implementasi Enkripsi Pengiriman Suara Menggunakan Algoritma Twofish. Jurusan Teknik Informatika, Fakultas Teknologi Industri, Institut Teknologi Bandung.

[REP00] Nechvatal, James, dkk. 2000. *Report on the Development of the Advance Encryption Standard (AES)*. U.S. Department of Commerce.

[RIN07] Munir, Rinaldi, (2007), Bahan Kuliah IF5054 Kriptografi. Program Studi Teknik Informatika, Institut Teknologi Bandung.

[SCH96] Schneier, Bruce. 1996. *Applied Cryptography 2nd*. John Wiley & Sons.

[SER09] 2009. Serpent Home Page. Official Serpent Homepage. URL : *http://www.cl.cam.ac.uk/~Erja14/serpent.html*

[SPX07] Valin, Jean-Marc. 2007. *The Speex Codec Manual Version 1.2 Beta 3*.

[TAN01] Tanenbaum, Andrew S. 2001. *Modern Operating Systems 2nd*. Prentice Hall.

[TAN03] Tanenbaum, Andrew S. 2003. *Computer Networks 4th*. Prentice Hall.

[VALIN] Valin, Jean-Marc. Speex: *A Free Codec for Free Speech*. Xiph.org Foundation.

[WCH99] Wichman, Shanon. 1999. *A Comparison of Speech Coding Algorithms ADPCM vs CELP*. University of Texas.

[XOGG] OGG Vorbis Documentation. URL: http://www.xiph.org/vorbis/doc