Studi dan Implementasi Enkripsi Pengiriman Pesan Suara Menggunakan Algoritma Twofish

Ratih

Laboratorium Ilmu Rekayasa dan Komputasi
Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung
e-mail: if13016@students.if.itb.ac.id

ABSTRAK

Komunikasi suara dengan menggunakan jaringan internet saat ini telah banyak digunakan, namun komunikasi suara yang digunakan tersebut belum tentu aman. Makalah ini akan membahas tentang solusi pengamanan pesan suara melalui jaringan internet dengan menggunakan enkripsi. Enkripsi berarti melakukan pengkodean pesan suara agar pihak yang tidak berhak tidap dapat memahaminya. Algoritma enkripsi yang digunakan pada makalah ini adalah algoritma enkripsi *cipher* blok Twofish. Algoritma *cipher* blok akan menimbulkan *delay* yang lebih besar daripada algoritma *cipher* aliran. Oleh karena itu, penerapan algoritma Twofish harus disesuaikan agar *delay* yang ditimbulkan kecil dan property *real time* terjaga. Pada makalah ini, perubahan tersebut dilakukan dengan mengubah mode operasi Twofish menjadi mode operasi *counter* yang dikatakan dapat merubah efisiensi *cipher* blok menjadi menyerupai *cipher* aliran.

Kata kunci: Enkripsi, *cipher* blok, mode operasi *counter*.

Makalah diterima 27 September 2007 . Revisi akhir [tanggal bulan tahun].

1. PENDAHULUAN

Saat ini komunikasi suara telah menjadi bagian dari kehidupan sehari-hari. Teknologi terbaru dalam komunikasi suara merupakan komunikasi melalui internet atau jaringan yang biasa disebut voice over internet protocol (VOIP). Berbagai macam jenis komunikasi suara tersebut belum tentu aman untuk digunakan, karena belum tentu ada suatu standar keamanan yang diterapkan untuk masing-masing fasilitas komunikasi suara tersebut.

Salah satu solusi untuk mengamankan data suara tersebut adalah dengan melakukan *voice scrambling*, yaitu perubahan pada sinyal telekomunikasi untuk membuatnya menjadi tidak dapat diketahui oleh siapapun kecuali pihak yang memiliki alat penerima khusus [ANS07]. Namun teknik ini memiliki tingkat keamanan yang sangat rendah.

Solusi lain yang memiliki tingkat keamanan jauh lebih tinggi adalah enkripsi suara. Enkripsi dilakukan pada data suara sebelum data suara dikirimkan, sehingga pihak lain yang tidak berhak tidak dapat memahami data suara yang dikirimkan tersebut meskipun data suara berhasil diakses. Algoritma enkripsi yang akan dibahas pada tugas akhir ini adalah algoritma Twofish.

Algoritma Twofish merupakan algoritma kuat yang sampai saat ini dinyatakan aman karena masih belum ada serangan kriptanalisis yang benar-benar dapat mematahkan algoritma ini [SCH05]. Algoritma ini juga tidak dipatenkan, sehingga penggunaannya pada alat enkripsi tidak perlu mengeluarkan biaya.

Algoritma Twofish merupakan algoritma *cipher* blok, hal ini merupakan hambatan jika digunakan untuk melakukan enkripsi aliran suara. *Cipher* blok melakukan enkripsi berdasarkan blok-blok bit, *delay* yang ditimbulkan menjadi besar karena harus menunggu data sejumlah blok tersebut. Untuk memperkecil *delay* yang ditimbulkan, harus dilakukan penyesuaian terhadap penerapan algoritma enkripsi yang berbasis *cipher* blok.

Salah satu cara yang dapat digunakan adalah dengan menyesuaikan mode operasi yang digunakan. Salah satu mode operasi yang dapat digunakan untuk mengubah efisiensi dan kecepatan enkripsi *cipher* blok menjadi menyerupai *cipher* aliran adalah mode operasi counter.

Oleh karena itu, pada tugas akhir ini dipilih penerapan algoritma Twofish dengan mode operasi yang disesuaikan menjadi mode operasi counter untuk melakukan enkripsi pada aliran pesan suara.

2. Kriptografi

Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan [SCH96]. Pada intinya kriptografi adalah teknik penyandian data untuk menjaga keamanan pesan. Proses penyandian pada kriptografi terdiri atas dua tahapan, yaitu enkripsi dan dekripsi. Enkripsi merupakan proses perubahan plainteks menjadi cipherteks yang tidak

dapat dimengerti, sedangkan dekripsi adalah proses perubahan kembali cipherteks menjadi plainteks.

Kriptografi biasanya dilakukan dengan kunci, untuk menambah tingkat keamanannya. Kriptografi yang dilakukan dengan menggunakan kunci yang sama untuk enkripsi dan dekripsi disebut algoritma kunci simetri.

Terdapat dua macam tipe yang pada umumnya digunakan pada algoritma kunci simetri, yaitu *cipher* blok dan *cipher* aliran. *Cipher* blok beroperasi pada satu blok bit pada setiap waktu, sedangkan *cipher* aliran dapat langsung beroperasi pada satu bit atau *byte*. Algoritma Twofish karena termasuk kedalam *cipher* blok, maka harus beroperasi pada satu blok bit plainteks.

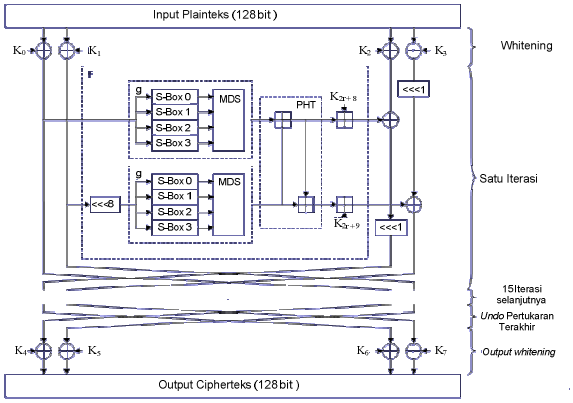
2.2 Mode Operasi *Cipher* Blok

Mode operasi pada *cipher* blok diperlukan untuk meningkatkan keamanan hasil enkripsi karena pada *cipher* blok, kunci dan plainteks yang sama akan menghasilkan cipherteks yang sama sehingga memudahkan serangan kriptanalisis.

Mode operasi yang pada umumnya diterapkan pada algoritma Twofish adalah mode operasi CBC (*cipher block chaining*). Kelemahan dari mode operasi CBC jika diterapkan untuk enkripsi pesan suara adalah *delay* yang ditimbulkan. *Delay* dapat terjadi karena pada CBC setiap enkripsi dari blok pada satu pesan harus menunggu hasil cipherteks dari blok sebelumnya. Untuk mengatasi masalah *delay* ini, perlu digunakan mode operasi lain yang memiliki efisiensi mendekati *cipher* aliran agar *delay* yang dihasilkan kecil.

2.1. Algoritma Kriptografi Twofish

Seperti telah dibahas sebelumnya, Twofish beroperasi pada satu blok plainteks, blok plainteks pada Twofish terdiri dari 128 bit. Struktur Algoritma Twofish seperti pada Gambar 2.



Gambar 1. Struktur Algoritma Twofish

Pada implementasi algoritma Twofish, terdapat beberapa hal yang harus diperhatikan, antara lain:

1. Bit masukan sebanyak 128 bit akan dibagi menjadi empat bagian masing-masing sebesar 32 bit menggunakan konvensi *little-endian*. Dua bagian bit akan menjadi bagian kanan, dua bagian bit lainnya akan menjadi bagian kiri.
2. Bit *input* akan di-XOR terlebih dahulu dengan empat bagian kunci, atau dengan kata lain mengalami proses *whitening*.
   $R_{0,i} = P_i \oplus K_i$ i = 0, ..., 3
   Dimana K adalah kunci, Ki berarti sub kunci yang ke-i.
3. Seperti telah dibahas diatas, algoritma Twofish menggunakan struktur jaringan Feistel. Jaringan Feistel yang digunakan oleh Twofish terdiri dari 16 iterasi. Fungsi f dari Twofish terdiri dari beberapa tahap, yaitu:
   a. Fungsi g, yang terdiri dari empat s-box dan matriks MDS
   b. PHT (*pseudo-hadamard transform*/ perubahan *pseudo hadamard*)
   c. Penambahan hasil PHT dengan kunci

3. Analisis

Masalah yang ingin diselesaikan pada tugas akhir ini antara lain adalah menerapkan algoritma Twofish agar bisa digunakan untuk enkripsi suara dan digunakan untuk pengiriman pesan suara antara dua buah komputer melalui jaringan. Perangkat lunak yang dikembangkan diharapkan dapat memberikan simulasi terhadap solusi yang ditawarkan.

Algoritma twofish digunakan untuk enkripsi aliran pesan suara dengan cara merubah mode operasi yang digunakan sehingga memiliki efisiensi menyerupai cipher aliran, yaitu dengan menggunakan mode operasi *counter*.

3.1 Penerapan Metode Counter

Cara membangkitkan blok *counter* yang akan diterapkan, dapat dirangkum menjadi:

1. Dari satu blok *counter* awal, T_1, akan diterapkan fungsi *increment* untuk membangkitkan blok *counter* selanjutnya.
2. Blok *counter* akan terbagi menjadi dua bagian, yaitu *message nonce* dan bit-bit yang akan dipakai untuk *increment*. *Message nonce* akan diambil dari waktu milidetik saat blok *counter* diinisialisasi.
3. Fungsi *increment* yang digunakan merupakan fungsi *increment* standar, berdasarkan definisi

oleh *National Institute of Standards and Technology* (NIST), yaitu:

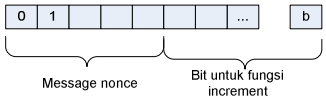
Jika,

m = jumlah bit fungsi *increment* (1)

maka,

$[X]_m = [X+1 \bmod 2^m]$ (2)

Misalkan panjang blok *counter* yang digunakan merupakan *b*. Blok *counter* yang digunakan akan memiliki bentuk seperti pada Gambar III-1.

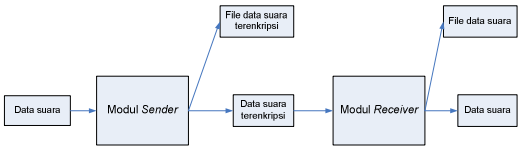


Gambar 2. Blok *Counter*

3.2 Analisis Perangkat Lunak

Perangkat lunak dapat dijalankan dalam dua mode, yaitu mode *Sender* yang menggunakan dan mode *Receiver*. Komputer yang menjalankan mode *Sender* akan berfungsi sebagai penerima masukan dan pengirim pesan suara, sementara komputer yang menjalankan mode *Receiver* akan berfungsi sebagai penerima pesan suara.

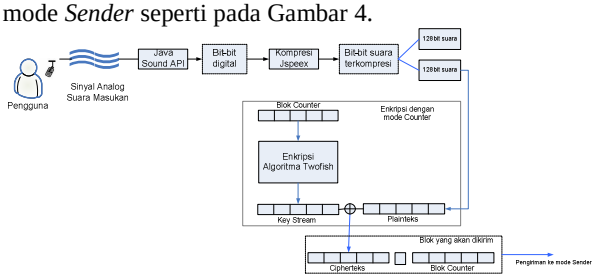
Berdasarkan penjelasan diatas, maka arsitektur perangkat lunak dapat digambarkan seperti pada Gambar 3.



Gambar 3. Arsitektur perangkat lunak

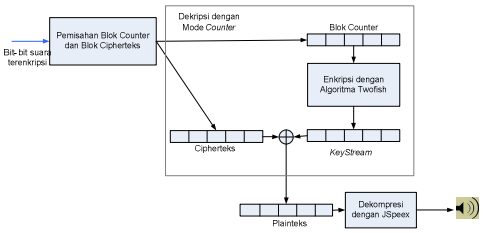
File data suara terenkripsi akan menyimpan data suara pengguna setelah keluar dari modul *Sender*. Sementara *file* data suara akan menyimpan data suara setelah dikeluarkan oleh modul *Receiver*.

Secara lebih detail, proses yang terjadi pada mode *Sender* seperti pada Gambar 4.



Gambar 4. Proses Detail Mode *Sender*

Setelah bit-bit suara selesai diproses, bit-bit suara tersebut dikirimkan ke mode *receiver* untuk diproses kembali. Secara lebih detail proses yang terjadi pada mode *receiver* seperti pada Gambar 5.



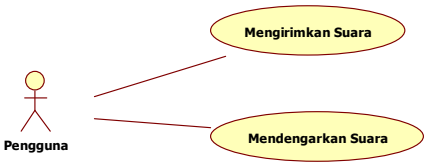
Gambar 5. Proses Detail Mode *Receiver*

4. Perancangan Perangkat Lunak

Perancangan perangkat lunak dilakukan dengan membuat diagram *use case* dan diagram kelas.

4.1. Diagram *Use Case*

Pengguna dapat melakukan dua hal, yaitu mengirimkan suara dan mendengarkan suara. Diagram *use case* dapat dilihat pada Gambar 6.

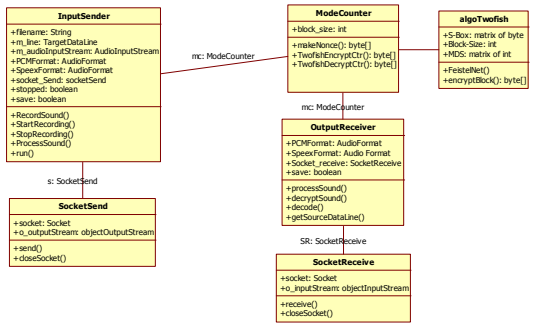


Gambar 6. Diagram *use case*

Untuk mengirimkan suara, pengguna harus berada pada mode *Sender*. Untuk mendengarkan suara, pengguna harus berada pada mode *Receiver*. Kedua mode tersebut harus digunakan pada dua komputer yang berbeda dan dihubungkan oleh kabel LAN.

4.2. Diagram Kelas

Identifikasi kelas dilakukan berdasarkan hasil analisis perangkat lunak. Terdapat enam kelas pada perangkat lunak ini, yaitu kelas *InputSender*, *OutputReceiver*, *SocketSend*, *SocketReceive*, *ModeCounter*, dan kelas AlgoTwofish. Rancangan diagram kelas dapat dilihat pada Gambar 5.



Gambar 5. Diagram Kelas

5. Implementasi

Perangkat lunak yang dikembangkan untuk melakukan pengiriman pesan suara memiliki batasan sebagai berikut:

1. Perangkat lunak yang dibuat hanya melibatkan dua macam entitas yaitu komputer pengirim pesan dan komputer penerima pesan.
2. Proses digitalisasi dan kompresi sinyal suara tidak diimplementasikan melainkan memakai *library* dan API yang telah tersedia.

5.1. Implementasi Kelas

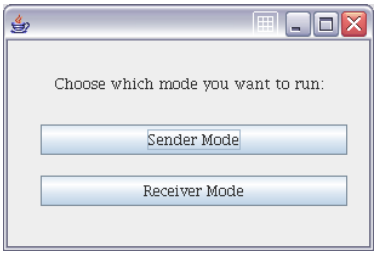
Kelas-kelas yang telah dirancang pada bagian perancangan diimplementasikan menjadi kelas-kelas dalam bahasa Java. Kelas-kelas yang digunakan pada proses implementasi ini akan dijelaskan pada Tabel IV-1.

Tabel 1. Penjelasan kelas Implementasi

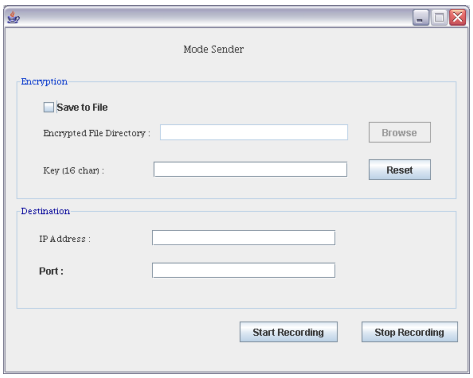
No.	Kelas	Nama *File*	Keterangan
1.	*InputSender*	`InputSender.java`	Menggunakan library *JSpeex* dan API Java Sound untuk proses sinyal suara.
2.	*SocketSend*	`SocketSend.java`	
3.	*OutputReceiver*	`Output Receiver.java`	Menggunakan library *JSpeex* dan API Java Sound untuk proses sinyal suara.
4.	*SocketReceive*	`Socket Receive.java`	
5.	algoTwofish	`Algo Twofish.java`	
6.	*ModeCounter*	`ModeCounter.java`	

5.2. Implementasi Antarmuka

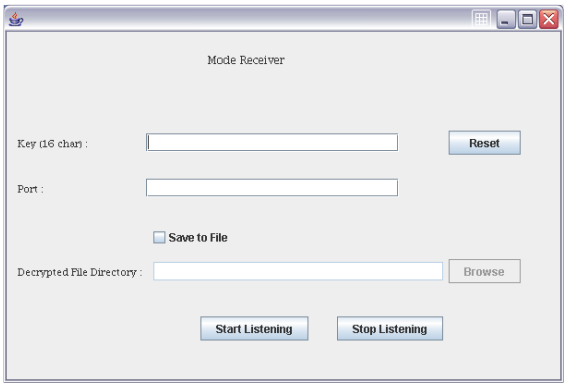
Antarmuka perangkat lunak dibangun dengan menggunakan *IDE Netbeans* 5.0, yang memiliki kelas-kelas untuk membuat *Graphical User Interface (GUI)*. Antarmuka perangkat lunak ini terbagi menjadi tiga layar, yaitu layar utama, layar mode *Sender*, dan layar mode *Receiver*.



Gambar 6. Layar Utama



Gambar 7. Layar Mode *Sender*



Gambar 8. Layar Mode *Receiver*

6. Pengujian

Pengujian yang dilakukan bertujuan membutikan dua hal, yaitu kebenaran proses enkripsi dekripsi dan kinerja perangkat lunak yang dihasilkan. Kinerja perangkat lunak maksudnya adalah menguji waktu pengiriman pesan suara menggunakan perangkat lunak setelah melewati proses enkripsi dan kompresi.

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

Pengujian kinerja perangkat lunak juga memperoleh hasil yang memuaskan karena *delay* yang dihasilkan relative dapat diterima. *Delay* yang dihasilkan bervariasi, namun memiliki rentang antara 31 sampai 94 milidetik. *Delay* seperti itu masih dapat diterima karena *delay* yang dapat ditoleransi untuk komunikasi *real time* merupakan 250 milidetik [IFK07].

7. Kesimpulan

Makalah ini membahas tentang penerapan algoritma Twofish pada aliran pesan suara melalui jaringan. Beberapa kesimpulan yang dapat diambil adalah:

1. Algoritma Twofish 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 properti *real time*.
    b. Proses enkripsi dan dekripsi merupakan proses yang sama karena menggunakan mode *counter*, tidak perlu menggunakan proses dekripsi seperti Twofish pada umumnya.
2. Kualitas suara setelah mengalami kompresi dan enkripsi tetap memiliki kualitas yang cukup baik (distorsi yang ada dapat diacuhkan).
3. *Delay* yang dihasilkan meskipun tetap terasa, dapat dianggap tidak terlalu mengganggu karena dibawah 250 milidetik dan suara yang dihasilkan dapat didengar tanpa terputus-putus.

REFERENSI

[ANS07] URL: http://www.answers.com/voice+compression?cat=technology, diakses mulai 13 Juni 2007

[DWO01] Dworkin, Morris. (2001). Recommendation for Block Cipher Modes of Operation.

[IFK07] URL: http://www.infokomputer.com/arsip/012000/infotek/infotek.shtml, diakses mulai 10 September 2007.

[KEL98] Kelsey, John, et al. (1998). Twofish: a 128-Bit Block *Cipher*.

[MUN04] Munir, Rinaldi. (2004). Bahan Kuliah IF5054 Kriptografi. Departemen Teknik Informatika, Institut Teknologi Bandung.

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

[SCH05] URL: http://www.schneier.com/blog/archives/2005/11/Twofish_cryptan.html.com/, diakses mulai 1 Desember 2006.

[WIK07] URL: http://en.wikipedia.org/wiki/Secure_voice, diakses mulai 13 Juni 2007