Algoritma Twofish : kinerja dan implementasinya sebagai salah satu kandidat algoritma AES (Advanced Encryption Standard)

Indra Mukmin – NIM : 13506082 [1)]

1) Jurusan Teknik Informatika ITB, Bandung 40135, email: if16082@students.if.itb.ac.id

Abstract – *Makalah ini membahas Twofish, salah satu algoritma yang digunakan dalam bidang kriptrografi. Kriptografi sendiri merupakan ilmu sekaligus seni untuk menjaga kerahasiaan pesan dengan cara menyamarkannya melalui proses enkripsi. Agar memiliki keseragaman, diperlukan suatu standar dalam proses enkripsi tersebut yang merupakan alasan utama dibuatnya DES (Data Encryption Standard) oleh NIST (National Institute of Standards and Technology).*

Sejalan berkembangnya waktu, diperlukan suatu algoritma enkripsi standar yang lebih efisien mengingat keinginan para kriptografer yang menginginkan proses yang "closed door" sehingga kemudian muncul AES (Advanced Encryption Standard). Salah satu algoritma yang disarankan sebagai standar adalah Twofish, mengingat beberapa keunggulan yang dimiliki algoritma ini.

Dalam makalah ini akan dibahas mengenai tujuan desain, keunggulan, kinerja, dan salah satu contoh implementasi Twofish yaitu proses enkripsi pengiriman pesan suara, serta beberapa hal lain yang berkaitan dengan algoritma Twofish.

Kata Kunci*: Twofish, kriptografi, enkripsi, AES, fietsel network, key setup, chiper, Mode Counter*

1. PENDAHULUAN

Pada zaman sekarang ini, menjaga kerahasiaan informasi merupakan hal yang sangat penting. Sebagai contoh bagi – bagi perusahaan besar, penyimpanan dokumen serta data – data penting adalah kewajiban yang mesti dilakukan. Penyalahgunaan data – data rahasia perusahaan tersebut oleh pihak tertentu tentunya bisa saja menimbulkan kerugian yang sangat besar pada perusahaan tersebut.

Contoh lainnya adalah komunikasi suara lewat jaringan internet. Kemungkinan pihak lain untuk mencuri informasi yang disampaikan lewat komunikasi elektronik tersebut sangat besar mengingat belum adanya sekuritas khusus terhadap aplikasi tersebut. Karenanya, salah satu alternatif yang dapat digunakan untuk menjaga kerahasiaan informasi tersebut adalah dengan menyamarkannya menjadi bentuk tersandi yang tidak bermakna. Hal tersebut dapat dilakukan dalam kriptografi. Proses penyandian tersebut dilakukan melalui proses enkripsi, yaitu mengubah pesan asli menjadi bentuk – bentuk tersandi yang tidak bermakna.

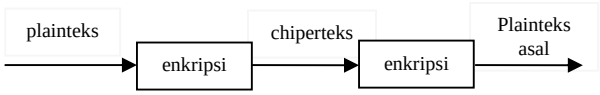
Kriptografi amat luas penggunaannya, oleh karena itulah, pada tahun 1972 dan 1974, National of Standart (sekarang bernama NIST) mengumumkan adanya standar enkripsi, yaitu DES (Data Encryption Standard).

Dalam proses perkembangannya ternyata kunci dalam DES dirasa terlalu pendek bagi keamanan komersial Akhirnya, NIST mengumumkan AES (Advanced Encryption Standard) pada tahun 1997. Salah satu kandidat AES adalah Twofish. Hal ini disebabkan Twofish memenuhi semua kriteria yang dibutuhkan NIST, yaitu 128-bit block, 128 bit, 192 bit dan 256 bit key (kata kunci), efisien pada *platform* manapun dan lain-lain, serta beberapa desain berat lainnya.

2. KRIPTOGRAFI

Kriptografi merupakan ilmu sekaligus seni untuk menjaga kerahasiaan pesan (data atau informasi) dengan cara menyamarkannya (*to cripyt*) menjadi bentuk tersandi yang tidak bermakna.[1]

Pesan yang dirahasiakan dinamakan plainteks, sedangkan pesan hasil penyamaran dinamakan chiperteks. Proses penyamaran dari plainteks ke chiperteks disebut enkripsi (encryption) dan proses pembalikan dari chiperteks ke plainteks disebut dekripsi (decryption). Gambar dibawah memperlihatkan diagram kedua proses yang dimaksud.



Gambar 1: Proses Enkripsi dan Dekripsi

2.1. Notasi Matematis

Jika chiperteks dilambangkan dengan C dan plainteks dilambangkan dengan P, maka fungsi enkripsi E memetakan P ke C,

$$E(P) = C$$

Pada proses kebalikannya, fungsi dekripsi D memetakan C ke P,

$$D(C) = P$$

Kekuatan suatu algoritma diukur dari banyaknya kerja yang dibutuhkan untuk memecahkan data chiperteks menjadi plainteksnya. Untuk kriptografi modern, kekuatan algoritma terletak pada kuncinya, yaitu berupa deretan karakter atau bilangan bulat yang

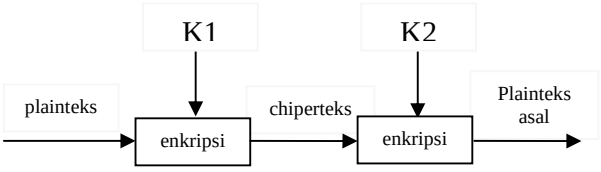
dijaga kerahasiaannya. Kunci ini dapat dianalogikan dengan penggunaan PIN pada ATM.
Secara matematis, pada sistem kriptografi yang menggunakan kunci K, maka fungsi enkripsi dan dekripsi menjadi,

$$E_{k1}(P) = C$$

$$D_{k2}(C) = P$$

dan kedua fungsi ini memenuhi

$$D_{k2}(E_{k1}(P)) = P$$



Gambar 2: Enkripsi dan Dekripsi pada algoritma kriptografi modern

Jika K1 = K2 (yaitu, kunci untuk proses enkripsi sama dengan kunci untuk dekripsi), maka algoritma kriptografinya disebut algoritma simetri. Contoh algoritma yang menjadi standar algoritma simetri adalah DES (Data Encryption Standard). Sebaliknya, apabila K1 tidak sama dengan K2, maka disebut algoritma nirsimetri. Contohnya adalah algoritma RSA (Rivest-Shamir-Adleman)

2.2. DES (Data Encryption Standard)
DES mulai digunakan sebagai standar untuk proses enkripsi pada tahun 1972. DES memadukan teknik permutasi, ekspansi, kompaksi, dan substitusi, yang semuanya dilakukan dalam 16 kali perulangan. Panjang kunci DES adalah 8 karakter atau 64 bit. Dari 64 bit tersebut, hanya 56 bit saja yang dipakai dalam proses enkripsi. Namun demikian, hanya dengan 56 bit itu saja akan terdapat 2^{56} atau 72.057.594.037.927.936 kemungkinan kunci. Diandaikan dalam satu detik dapat dicobakan satu juta kemungkinan kunci, maka akan diperlukan waktu 2284 tahun untuk menemukan kunci yang benar.
Atas pertimbangan perkembangan teknologi yang semakin maju, dirasakan kunci DES menjadi terlalu pendek bagi keamanan komersial. Akhirnya pada tahun 1997, NIST sebagai Lembaga Standarisasi menetapkan AES (Advanced Encryption Standard) sebagai pengganti DES. Salah satu algoritma yang direkomendasikan sebagai AES adalah algoritma Twofish.

3. 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 [2]. Algoritma ini juga tidak dipatenkan sehingga penggunaannya pada alat enkripsi tidak perlu mengeluarkan biaya.

3.1 Desain dan Keunggulan Twofish
Algoritma Twofish merupakan salah satu algoritma yang direkomendasikan sebagai AES. Hal ini disebabkan pemenuhan kriteria desain oleh NIST sebagai standar AES yaitu :

a. Blok cliper simetris 128-bit
b. Memiliki panjang kunci antara lain : 128 bit, 192 bit, dan 256 bit.
c. Tidak terdapat kunci – kunci yang lemah.
d. Memiliki efisiensi pada *software* dan *hardware* dari *platform* yang berbeda.
e. Memiliki rancangan yang fleksibel, misalnya menerima panjang kunci tambahan, dapat diterapkan pada *software* dan *hardware* dari *platform* berbeda, cocok untuk *stream chipper*, fungsi hash dan MAC.
f. Desain yang simpel, memudahkan baik untuk analisa maupun implementasi.

Sementara itu, tujuan NIST dalam hubungannya dengan Twofish adalah sebagai berikut:

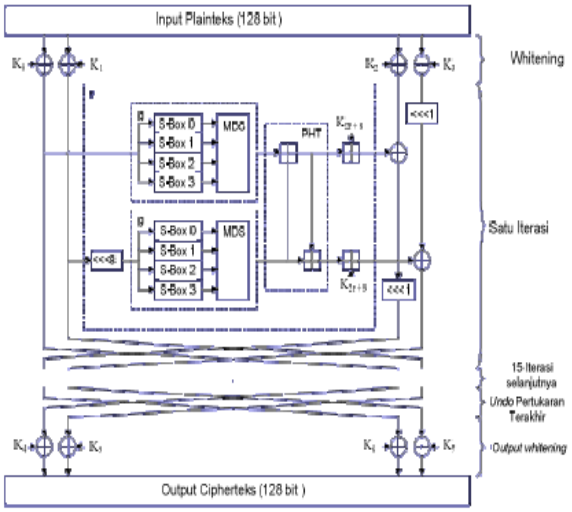
- Twofish 16-round tidak boleh memiliki *chosen-plaintext attack* yang memerlukan kurang dari 2^{80} *chosen-plaintext* dan menggunakan waktu dari 2^N dimana N adalah panjang kunci.
- Twofish 12-round tidak boleh memiliki suatu *related-key attack* yang memerlukan kurang dari 2^{64} *chosen-plaintext* dan menggunakan waktu kurang dari 2^N dimana N adalah panjang kunci.

Beberapa keunggulan algoritma kriptografi Twofish yaitu :

- Memiliki varian dengan sebuah nomor variabel dari setiap *round*.
- Memiliki *key schedule* yang dapat di-prakomputasikan untuk kecepatan maksimum dan penggunaan memori manimum.
- Cocok sebagai *stream chipper*, fungsi hash satu arah, MAC dan *pseudo random number generator*, dengan menggunakan metode konstruksi yang dapat dimengerti.
- Memiliki varian *famili-key* untuk memungkinkan versi *chipper* yang berbeda dan *non interrupterable*.

Namun pada intinya, keunggulan algoritma Twofish adalah pemenuhan terhadap kriteria – kriteria yang telah ditetapkan oleh NIST.
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 32 bit menggunakan konvensi little-Endian. Dua bagian bit akan menjadi bagian kanan, dan dua lainnya



Gambar 3: Struktur Algoritma Twofish

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. Algoritma Twofish menggunakan struktur jaringan Feistel. Jaringan Feistel yang digunakan oleh Twofish terdiri atas 16 perulangan. Fungsi f pada algoritma Twofish terdiri atas beberapa tahap yaitu :
   a. Fungsi g, yang terdiri dari 4 s-box dan matriks MDS
   b. PHT (Pseudo-Hadamard Transformation) atau Transformasi Pseudo-Hadamard
   c. Penambahan hasil PHT dengan kunci

3.2 Jaringan Fietsel

Jaringan Fietsel adalah metode umum untuk mentransformasi suatu fungsi menjadi bentuk permutasi. Bagian paling fundamental dari Jaringan Fietsel adalah fungsi f: sebuah pemetaan *key-dependent* dari suatu masukan *string* menjadi keluaran *string*. Dalam Twofish dilakukan Fietsel *Network* sebanyak 16 kali. Prosedur Jaringan Fietsel sebenarnya terdiri dari masukan Whitening, S-boxes, keluaran Transformasi Pseudo Hadamard, dan keluaran Whitening[3].

3.3 S-Boxes

S-Box adalah operasi subsitusi *table-driven* non linear yang digunakan dalam blok chipper. S-boxes bervariasi antara setiap ukuran masukan dan ukuran keluarannya, dan bisa diciptakan secara acak atau dengan algoritma.
Twofish menggunakan empat *bijective, key-dependent* dan *8-by-8-bit* S-boxes. S-boxes ini dibuat menggunakan dua permutasi *8-by-8-bit* dan *material key*.

3.4 Matriks MDS

Kode MDS (Maximum Distance Separable) melalui **a** adalah pemetaan linear dari elemen field **a** ke elemen field **b**, menghasilkan campuran dari vector **a+ b** elemen, dengan properti jumlah minimum angka tidak nol dalam vektor tidak nol paling kurang b + 1. Dengan kata lain "Distance" adalah jumlah elemen yang berbeda antara dua vector yang berbeda yang dihasilkan oleh MDS paling kurang b+1. Pemetaan MDS bisa direpresentasikan oleh matriks MDS yang terdiri dari a x b elemen. Twofish menggunakan matriks MDS 4 x 4 tunggal.

3.5 PHT atau Transformasi Pseudo-Hadamard

Transformasi Pseudo-Hadamard (PHT) adalah operasi sederhana yang bekerja dengan cepat dalam *software*. Diberikan dua masukan, a dan b, dan PHT 32 bit didefinisikan sebagai :

$A_0 = a + b \bmod 2^{32}$
$B_0 = a + 2b \bmod 2^{32}$

SAFER menggunakan PHT 8 bit secara meluas untuk proses difusi. Sementara itu, Twofish menggunakan PHT 32 bit untuk melakukan *mixing* terhadap keluarannya dari dua buah fungsi g 32 bit paralel. PHT ini dapat dieksekusi dalam dua *opcode* diatas kebanyakan mikroprosesor modern, termasuk keluarga Pentium

3.6 Whitening

Whitening merupakan teknik meng-XOR-kan *key material* sebelum ronde pertama dan sesudah ronde terakhir. Dalam serangan terhadap Twofish, terbukti bahwa whitening secara substansial meningkatkan kesulitan menyerang chipper, dengan jalan menyembunyikan masukan spesifik untuk awal dan akhir ronde dari Twofish.

3.7 Fungsi f

Pondasi dasar dari jaringan Feistel adalah fungsi *f*, yaitu suatu permutasi yang *key-dependent* terhadap nilai 64-bit. Fungsi *f* memerlukan tiga buah argumen, dua *input* word R_0 dan R_1, dan bilangan bulat *r* yang digunakan untuk memilih *subkey* yang besesuaian. R_0 dilewatkan fungsi *g*, yang menghasilkan T_0. R_1 dirotasikan dalam sebuah PHT dan dua *word* dari *key* yang diekspansi kemudian ditambahkan kepadanya.

$T_0 = g(R_0)$
$T_1 = g(ROL(R_1,8))$
$F_0 = (T_0 + T_1 + K_{2r+8}) \bmod 2^{32}$
$F_1 = (T_0 + T_1 + K_{2r+9}) \bmod 2^{32}$

Dimana (F_0, F_1) merupakan hasil dari *F*, *ROL* adalah rotasi ke kiri terhadap R_1 sejauh 8 bit.
Fungsi *F* selalu nonlinear dan kemungkinan nonsurjektif, yaitu tidak semua *output* yang dimungkinkan berada dalam ruang *output* dapat terjadi semua.

3.8 Key Schedule

Key schedule adalah suatu cara dimana bit-bit kunci diubah menjadi kunci – kunci bilangan bulat yang

dapat digunakan oleh chipper. Twofish memerlukan *material key* yang sangat banyak, dan memiliki *key schedule* yang rumit. Untuk memudahkan analisis, *key schedule* menggunakan primitif yang sama dengan fungsi pembulatan biasa.

Key schedule harus menyediakan 40 *word*, yaitu key $K_0 \ldots K_{39}$ dan 4 *key-dependent* S-boxes yang digunakan dalam fungsi g. Twofish didefinisikan untuk panjang N = 128, N = 192, dan N = 256. Kunci yang lebih pendek dari 256 bit dapat dipergunakan dengan cara mengisinya dengan nilai nol sampai panjang kunci yang lebih besar berikutnya.

4. KINERJA TWOFISH

Twofish telah didesain dari awal dengan menekankan pada kinerjanya. Twofish sangat efisien diimplementasikan pada beragam *platform*, yaitu CPU 32 bit, *smart card* 8 bit, dan perangkat keras VLSI. Yang lebih penting lagi, Twofish didesain untuk memungkinkan beberapa *layer* kinerja, tergantung pada kepentingan relatif terhadap kecepatan enkripsi, *key setup*, penggunaan memori, *hardware gate count*, dan parameter implementasi yang lain. Hasilnya merupakan algoritma yang sangat fleksibel yang dapat diimplementasikan secara efisien dalam beragam aplikasi kriptografi.

Sebagai contoh adalah kinerja Twofish pada mikroprosesor berukuran besar.. Enkripsi dan dekripsi untuk pilihan *key scheduling* yang berbeda dan pada beberapa mikroprosesor modern dengan menggunakan bahasa pemrograman dan kompiler berbeda. Selisih waktu untuk enkripsi dan dekripsi cenderung tipis. Tidak diperlukan waktu untuk men-*setup* algoritmanya kecuali untuk *key setup*. Waktu untuk mengubah sama dengan waktu yang digunakan untuk men-*setup* suatu *key*.

Dalam algoritma Twofish juga telah diimplementasi kan empat macam pilihan *keying* yang berbeda. Terdapat beberapa pilihan *keying* yang mungkin, dimana masing-masing mempunyai perbedaan tipis dalam hal *key setup*.

4.1 Full Keying

Pilihan ini melakukan prakomputasi terhadap kunci. Dalam menggunakan pilihan ini, suatu komputasi dari *g* berisi empat buah tabel pencarian, dan tiga buah operasi XOR. Sementara itu, kecepatan enkripsi dan dekripsinya bernilai konstan tanpa menghiraukan ukuran kunci.

4.2 Partial Keying

Untuk aplikasi dimana sebagian kecil blok dienkripsi dengan kunci tunggal, tidak akan menjadi masalah dalam membangun *key schedule* yang lengkap. Pilihan ini melakukan prakomputasi terhadap empat S-boxes dalam tabel berukuran 8 x 8 bit, dan menggunakan empat buah tabel MDS 8 x 32 bit untuk melakukan perkalian MDS. Dan sekali lagi, kecepatan enkripsi dan dekripsinya tidak menghiraukan ukuran kunci.

4.3 Minimal Keying

Untuk aplikasi yang mengenkripsi sangat sedikit bagian dari blok dengan kunci tunggal, disini terdapat optimasi lebih jauh yang mungkin. Penggunaan pilihan *Minimal Keying* ini hanya memerlukan sebuah tabel 1 Kb untuk menamping S-boxes yang diprakomputasi secara parsial. Pentingnya *byte key* dari S yang diprakomputasi adalah layaknya mereka diperlukan dalam setiap *round*.

4.4 Zero Keying

Pilihan ini tidak melakukan prakomputasi terhadap S-boxes, dan juga tidak memerlukan tabel ekstra. Sebagai gantinya, setiap entri di komputasi secara melayang. Waktu *key setup* secara murni digunakan untuk melakukan komputasi terhadap nilai K_i dan *S*. Untuk suatu aplikasi yang tidak memiliki waktu *key setup* sama sekali, waktu yang digunakannya untuk mengenkripsi satu blok adalah penjumlahan dari waktu *key setup* dan waktu enkripsi *zero keying*.

5. IMPLEMENTASI TWOFISH [4]

Komunikasi suara dengan menggunakan jaringan internet saat ini telah banyak digunakan, namun komunikasi suara yang digunakan tersebut belum tentu aman. 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. 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.

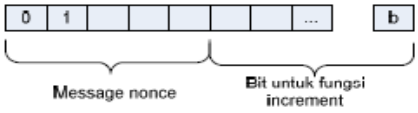
Algoritma twofish digunakan untuk enkripsi aliran pesan suara dengan cara merubah mode operasi yang digunakan sehingga memiliki efisiensi menyerupai cipher aliran, yaitu dengan menggunakan mode operasi *counter*.

5.1 Penerapan Metode Counter

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

1. Dari satu blok *counter* awal, *T1*, 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 mod 2m] (2) Misalkan panjang blok *counter* yang digunakan merupakan *b*. Blok *counter* yang digunakan akan memiliki bentuk seperti dibawah ini.

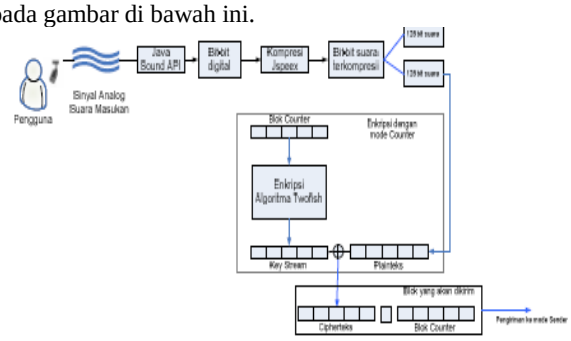


Gambar 4: Blok *Counter*

Untuk melakukan proses enkripsi suara tersebut, kemudian dibuat perangkat lunak yang mengaplikasikan algoritma Twofish tersebut.

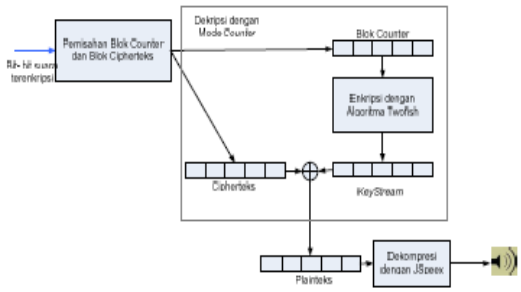
Perangkat lunak tersebut dapat dijalankan dalam dua mode, yaitu mode *Sender* yang menggunakan dan mode *Receiver*. Komputer yang menjalankan mode *Sender* akan berfungsi sebagai penerima masukan dan pengirim pesan suara, sementara komputer yang menjalankan mode *Receiver* akan berfungsi sebagai penerima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

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 di bawah ini.



Gambar 5 : Proses Detail Mode *Sender*

Setelah bit-bit suara selesai diproses, bit-bit suara tersebut dikirimkan ke mode *receiver* untuk diproses kembali seperti pada gambar di bawah ini.

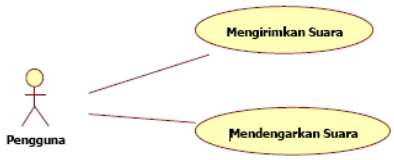


Gambar 6 : Proses Detail Mode *Receiver*

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

5.2 Diagram *Use Case*

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

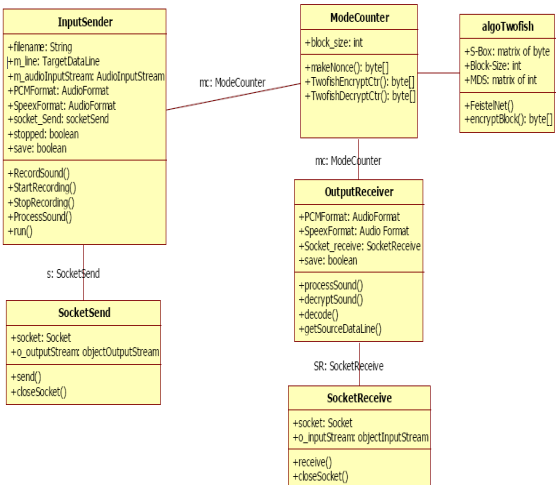


Gambar 7 : Diagram *Use Case*

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.

5.3 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 di bawah ini.



Gambar 8 : Diagram Kelas

Perangkat lunak yang dikembangkan untuk mengirimkan pesan suara tersebut memiliki batasan sebagai berikut :

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

6. KESIMPULAN

- Twofish adalah cipher blok 128 bit yang menerima key dengan panjang variabel diatas 256 bits dan tidak memiliki kunci – kunci yang lemah
- Twofish memiliki empat macam key schedule dalam implementasinya yaitu : full keying, partial keying, minimal keying, dan zero keying dengan perbedaan dalam hal *key setup*
- Twofish dibentuk berdasarkan jaringan Fietsel yang terdiri atas masukan Whitening, S-boxes, keluaran Transformasi Pseudo Hadamard, dan keluaran Whitening
- Twofish memiliki kehandalan- kehandalan dalam implementasinya diatas berbagai platform microprocessor, smart card dan hardware yang dibuat sebagai perangkat enkripsi data
- Twofish memiliki resistensi yang tinggi terhadap related key attack, dan hanya dapat ditembus dengan menggunakan *brute force*
- Salah satu contoh implementasi Algoritma Twofish adalah penerapannya dalam proses enkripsi aliran pesan suara.
- Untuk mendapatkan hasil yang maksimal, dilakukan modifikasi pada mode operasinya yaitu dengan mengganti mode operasinya menjadi Mode *Counter*.
- Berdasarkan kelebihan yang dimilikinya, algoritma Twofish dapat dijadikan standar AES. Namun demikian, pada tahun 2002 akhirnya diumumkan bahwa algoritma standar AES adalah algoritma Rijndael, salah satu dari lima kandidat algoritma yang diajukan.[5]

DAFTAR REFERENSI

[1] Rinaldi Munir, *Matematika Diskrit*, Prodi Teknik Informatika ITB, 2006

[2] URL : http://www.schneier.com/twofish.html
Tanggal akses : 24 Desember 13:00

[3] URL : http://prastowo.staff.ugm.ac.id/kuliah/ kriptografi /tugas-akhir/kel7-twofish/ kriptoTWOFISH.doc.
Tanggal akses : 24 Desember 2007 11:50

[4] Ratih," Studi dan Implementasi Enkripsi Pengiriman Pesan Suara Menggunakan Algoritma Twofish", National Conference On Computer Science & Information Technology VII, 2007.

[5] F.Sapty R, "Kriptografi",FIKOM UI, 2005
URL : http://bebas.vlsm.org/v06/Kuliah/MTI-Keamanan-Sistem-Informasi/2005/124/124P-04-final 2.0-Cryptography.pdf
Tanggal akses : 25 Desember 2007 11:30