STUDI PERBANDINGAN ALGORITMA SIMETRI *BLOWFISH* DAN *ADVANCED ENCRYPTION STANDARD*

Mohammad Riftadi – NIM : 13505029

Program Studi Informatika, Institut Teknologi Bandung
Jl. Ganesha No. 10, Bandung
E-mail : if15029@students.if.itb.ac.id

Abstrak

Makalah ini membahas tentang studi perbandingan algoritma simetri *Blowfish* dan algoritma *AES* (atau yang biasa juga disebut algoritma *Rijndael*). Algoritma *Blowfish* diciptakan oleh Bruce Schneier sebagai salah satu alternatif pengganti DES yang dirasa sudah tidak aman lagi. Algoritma ini menggunakan kunci yang panjangnya bisa bervariasi antara 32-bit hingga 448-bit. Algoritma ini telah diterima oleh kalangan internasional sebagai suatu algoritma enkripsi yang sukar dipecahkan. Di lain pihak, pada tahun 2001 dirilislah algoritma AES yang dimaksudkan untuk menggantikan DES sebagai standar baru di Amerika Serikat. Algoritma AES ini masih menggunakan kunci simetrik dengan panjang yang dapat bervariasi antara 128-bit, 192-bit, maupun 256-bit. Algoritma AES bekerja dalam mode penyandian blok (block cipher) 128-bit.

Dalam makalah ini juga telah dibuat program kecil yang ditulis dalam bahasa C++ untuk menguji kedua algoritma tersebut. Pengujian dalam makalah ini dibatasi hanya dari segi performansi dan keamanan data dalam mode *CBC* saja. Pengujian untuk performansi dilakukan dengan cara menghitung berapa lama proses enkripsi maupun dekripsi dari masing-masing algoritma dieksekusi untuk mengenkripsi arsip yang sama. Untuk pengujian keamanan data, dilakukan modifikasi pada blok-blok cipherteks untuk kemudian didekripsi kembali dan dilihat perubahan dari teks asalnya. Hasil uji menunjukkan bahwa AES jauh lebih baik secara dari segi performansi ketimbang algoritma *Blowfish*, sedangkan untuk segi keamanan data kedua algoritma menunjukkan hasil yang tidak jauh berbeda.

***Kata kunci**: Blowfish Cipher, Advanced Encryption Standard,* perbandingan algoritma*,* algoritma *Rijndael,* enkripsi, dekripsi.

1. Pendahuluan

Transmisi data melalui media elektronik telah menjadi suatu kebutuhan sehari-hari dalam dunia yang telah memasuki era informasi seperti sekarang ini. Akan tetapi, arus data yang dialirkan melalui media elektronik tersebut biasanya dikirimkan melalui sebuah saluran publik (contohnya Internet) yang sangat rentan terhadap terjadinya penyadapan, padahal data yang ditransmisikan di saluran publik seringkali bersifat penting dan rahasia sehingga tidak boleh sampai diketahui oleh pihak yang tidak berhak.

Untuk mengatasi hal tersebut dibutuhkan suatu metode untuk merahasiakan pesan yang dikirim. Salah satu cara merahasiakannya adalah dengan cara mengacak (atau sering juga disebut menyandikan) data yang akan ditransmisikan sehingga hanya pihak yang memiliki kunci saja yang dapat membaca isi pesan tersebut. Proses mengacak data tersebut biasanya disebut enkripsi, sedang proses kebalikannya (mengembalikan kembali pesan yang telah diacak ke bentuk normal) biasa disebut dekripsi.

Pada tahun 1976 dipilihlah algoritma DES oleh National Bureau of Standards (NBS) milik Amerika Serikat sebagai standar enkripsi dalam pemerintahan. Sebagai efeknya, algoritma DES digunakan secara luas oleh kalangan internasional sebagai standar untuk enkripsi. Algoritma DES ini adalah algoritma kriptografi simetris yang menggunakan kunci sepanjang 56-bit. Namun, seiring dengan berkembangnya kemampuan perangkat keras untuk komputasi dan juga berkembangnya sistem terdistribusi yang bisa melakukan komputasi menggunakan banyak komputer yang terhubung melalui suatu jaringan, DES menjadi dapat diserang melalui

exhaustive search dalam waktu beberapa jam saja.

Berusaha mengatasi kelemahan algoritma DES, Bruce Schneier, seorang pakar keamanan, pada tahun 1993 merilis algoritma yang ia namakan Blowfish. Algoritma Blowfish ini diciptakan sebagai salah satu alternatif pengganti DES yang dirasa sudah tidak aman lagi. Algoritma Blowfish menggunakan kunci yang panjangnya bisa bervariasi antara 32-bit hingga 448-bit. Setelah itu algoritma ini telah diterima oleh kalangan internasional sebagai suatu algoritma enkripsi yang sukar dipecahkan. Blowfish tidak dipatenkan dan bebas lisensi, yang berarti dapat digunakan oleh semua orang secara bebas tanpa dipungut bayaran.

Di lain pihak, pada tahun 2001 dirilislah algoritma AES yang dimaksudkan untuk menggantikan DES sebagai standar baru di Amerika Serikat. Algoritma AES ini masih menggunakan kunci simetrik dengan panjang yang dapat bervariasi antara 128-bit, 192-bit, maupun 256-bit. Algoritma AES bekerja dalam mode penyandian blok (*block cipher*) 128-bit. Panjang 128-bit ini merupakan panjang pasti yang tidak dapat dirubah.

Perbandingan antara algoritma *Blowfish* dengan AES inilah yang akan menjadi topik dalam makalah ini. Sebagai batasan masalah, perbandingan serta pengujian antara kedua algoritma ini dibatasi hanya pada performansi saat enkripsi/dekripsi dan keamanan data yang telah dienkripsi dengan masing-masing algoritma. Pengujian dilakukan menggunakan program kecil yang dibuat dengan bahasa C++.

2. Tipe dan Mode Algoritma Simetri

Algoritma kriptografi (*cipher*) simetri dapat dikelompokkan kedalam dua kategori, yaitu:

1. *Cipher* arus (*stream cipher*)
   Algoritma kriptografi jenis ini beroperasi pada plainteks/chiperteks dalam bentuk penyandian bit per bit. Mode operasi penyandian bit satu persatu inilah yang membuat nampak seolah-olah data yang masuk seperti arus yang mengalir.
2. *Cipher* blok (*block cipher*)
   Algoritma kriptografi beroperasi pada plainteks/cipherteks dalam bentuk blok bit, yang dalam hal ini rangkaian bit dibagi menjadi blok-blok bit yang sama panjang (dengan panjang yang sudah ditentukan sebelumnya).

2.1 *Cipher* Blok

Dalam *cipher* blok, rangkaian bit-bit plainteks dibagi menjadi blok-blok bit dengan panjang yang sama. Enkripsi dilakukan terhadap blok bit plainteks menggunakan bit-bit kunci (yang ukurannya sama dengan blok plainteks). Dekripsi dapat dilakukan dengan cara yang sama seperti enkripsi.

Misalkan blok plainteks (P) yang berukuran m bit dinyatakan sebagai vektor

$$P = (p_1, p_2, ..., p_m)$$

yang dalam hal ini p_i adalah bit 0 atau bit 1 untuk i = 1, 2, …, m, dan blok cipherteks (C) adalah

$$C = (c_1, c_2, ..., c_m)$$

yang dalam hal ini c_i adalah bit 0 atau bit 1 untuk i = 1, 2, …, m.

Bila plainteks dibagi menjadi n buah blok, barisan blok-blok plainteks dinyatakan sebagai

$$(P_1, P_2, \ldots, P_n)$$

Untuk setiap blok plainteks P_i, bit-bit penyusunnya dapat dinyatakan sebagai vektor

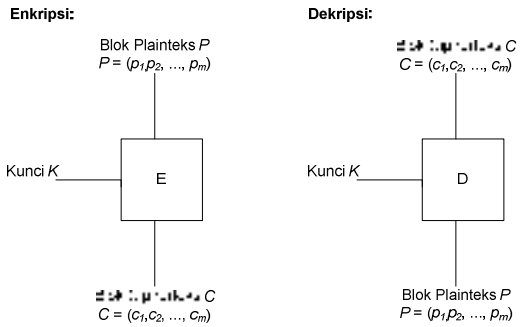
$$P_i = (p_{i1}, p_{i2}, ..., p_{im})$$

Enkripsi dengan kunci K dinyatakan dengan persamaan

$$E_k(P) = C,$$

sedangkan dekripsi dengan kunci K dinyatakan dengan persamaan

$$D_k(C) = P$$



Gambar 1 Skema Enkripsi dan Dekripsi dengan *Cipher* Blok

2.2 Mode Operasi *Cipher* Blok

Plainteks dibagi dalam beberapa blok dengan panjang tetap. Beberapa mode operasi dapat diterapkan untuk melakukan enkripsi terhadap keseluruhan blok plainteks. Empat mode operasi yang biasa diterapkan pada sistem blok *cipher* adalah:

1. *Electronic Code Book* (ECB)
2. *Cipher Block Chaining* (CBC)
3. *Cipher Feedback* (CFB)
4. *Output Feedback* (OFB)

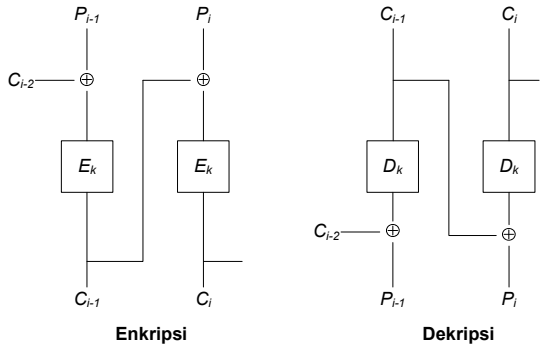
Untuk menyederhanakan permasalahan, mode operasi *cipher* blok yang digunakan dalam pengujian dalam makalah ini hanya mode operasi CBC saja.

2.2.1 *Cipher Block Chaining* (*CBC*)

Mode ini menerapkan mekanisme umpan balik pada sebuah blok, yang dalam hal ini hasil enkripsi blok sebelumnya diumpanbalikkan ke dalam enkripsi blok yang *current*. Caranya adalah hasil pada blok sebelumnya di-*XOR*-kan dengan plainteks blok sesudahnya.

Dekripsi dilakukan dengan cara sebaliknya, yaitu memasukkan blok cipherteks yang *current* ke fungsi dekripsi, kemudia meng-*XOR*-kan hasilnya dengan blok cipherteks sebelumnya.

Skema enkripsi dan dekripsi dengan mode *CBC* dapat dilihat pada Gambar 2.



Gambar 2 Enkripsi dan Dekripsi dengan Mode *CBC*

Secara matematis, enkripsi dengan mode *CBC* dinyatakan sebagai

$$C_i = E_k(P_1 \oplus C_{i-1})$$

dan dekripsi sebagai

$$P_i = D_k(C_i) \oplus C_{i-1}$$

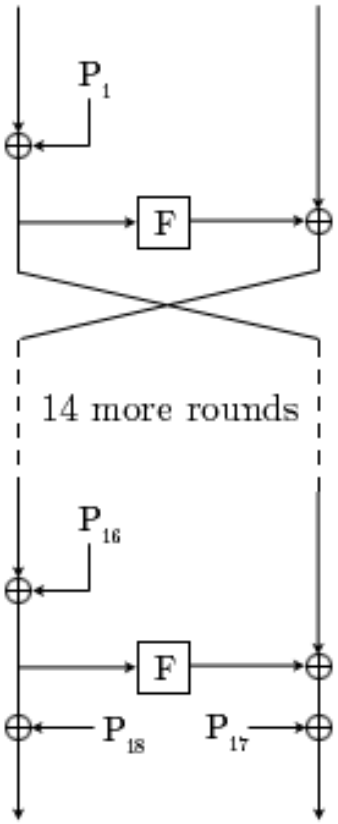
Yang dalam hal ini, $C_0 = IV$ (*initialization vector*). *IV* dapat diberikan oleh pengguna atau dibangkitkan secara acak oleh program. Jadi, untuk menghasilkan blok cipherteks pertama (C_1), *IV* digunakan untuk menggantikan blok cipherteks sebelumnya, C_0. Sebaliknya pada dekripsi, blok plainteks diperoleh dengan cara meng-*XOR*-kan *IV* dengan hasil dekripsi terhadap blok cipherteks pertama.

Pada mode *CBC*, blok plainteks yang sama akan menghasilkan blok cipherteks yang berbeda jika blok-blok plainteks sebelumnya berbeda. Hal ini disebabkan adanya pengaruh dari blok-blok sebelumnya tersebut.

3. *Blowfish Cipher*

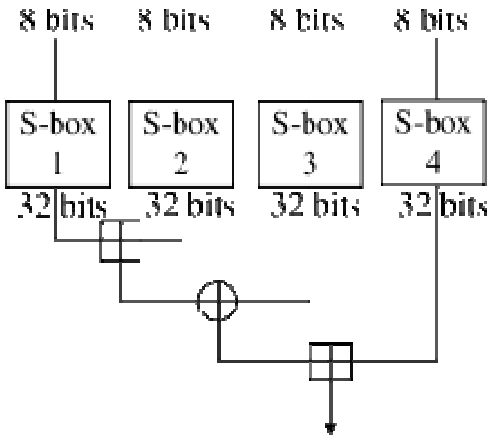
3.1. Algoritma *Blowfish*

Algoritma Blowfish beroperasi pada mode blok 64-bit dan memiliki sebuah kunci yang panjangnya bisa bervariasi antara 32-bit hingga 448-bit. Algoritma Blowfish juga menggunakan jaringan Feistel sebanyak 16 putaran dan menggunakan beberapa buah S-box yang sudah dispesifikasikan.



Gambar 3 Ilustrasi jaringan *Feistel* pada algoritma *Blowfish*

Blowfish menyimpan larik (array) dari upakunci (subkey) yang berisi 18 matriks-P dan 4 buah S-box. S-box disini menerima masukan 8-bit dan memberikan keluaran 32-bit. Sebuah matriks-P akan digunakan pada tiap putaran, dan setalah putaran terakhir, kedua bagian blok data akan di-XOR-kan dengan dua matriks-P yang belum digunakan.



Gambar 4 Ilustrasi fungsi pembulatan pada *Blowfish*

Untuk proses inversinya, dikarenakan menggunakan jaringan Feistel, maka Blowfish dapat diinversi cukup dengan meng-XOR-kan P17 dan P18 ke blok chiperteks, lalu menggunakan matriks-P dengan urutan terbalik.

4. *Advanced Encryption Standard* (AES)

4.1 Algoritma *Rijndael*

Algoritma AES ini sering juga disebut sebagai algoritma *Rijndael* dikarenakan oleh algoritma *Rijndael* inilah yang memenangkan sayembara untuk menjadi standar baru.

Masih seperti pada *DES*, *Rijndael* menggunakan substitusi dan permutasi, dan sejumlah putaran dalam jaringan *Feistel*. Untuk setiap putarannya, *Rijndael* menggunakan kunci yang berbeda. Kunci setiap putaran disebut *round key*. Namun berbeda daru *DES* yang bekerja dalam bit, *Rijndael* bekerja dalam *byte* sehingga lebih memungkinkan untuk implementasi algoritma yang efisien ke dalam *software* maupun *hardware*.

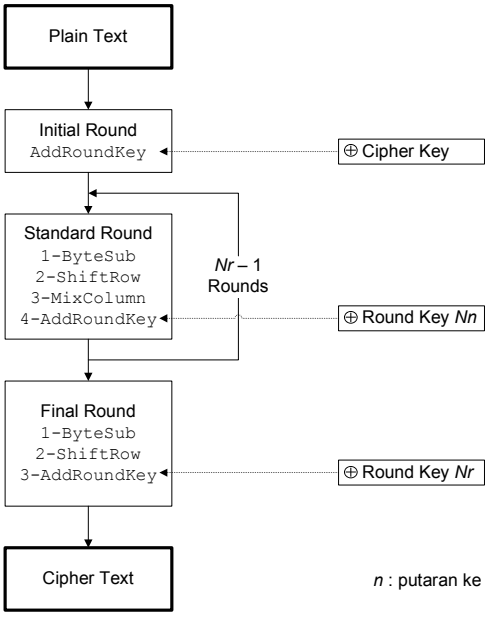
Garis besar dari algoritma *Rijndael* yang beroperasi dalam blok 128-bit dengan kunci 128-bit adalah sebagai berikut:

1. *AddRoundKey*: melakukan *XOR* antara plainteks dengan *cipher key*. Tahap ini sering disebut sebagai *initial round*.
2. Putaran sebanyak *Nr* – 1 kali. Proses yang dilakukan pada setiap putaran adalah:
   a. *ByteSub*: substitusi byte dengan menggunakan tabel substitusi (*S-box*).
   b. *ShiftRow:* pergeseran baris-baris *array state* secara *wrapping*.

c. *MixColumn*: mengacak data di masing-masing kolom *array state*. Ilustarsi
d. *AddRoundKey:* melakukan *XOR* antara *state* sekarang dengan *round key*. Ilustarsi

3. *Final round*: proses untuk putaran terakhir:
   a. *ByteSub*.
   b. *ShiftRow*.
   c. *AddRoundKey*.

Diagram proses enkripsi *AES* dapat dilihat pada Gambar 5.



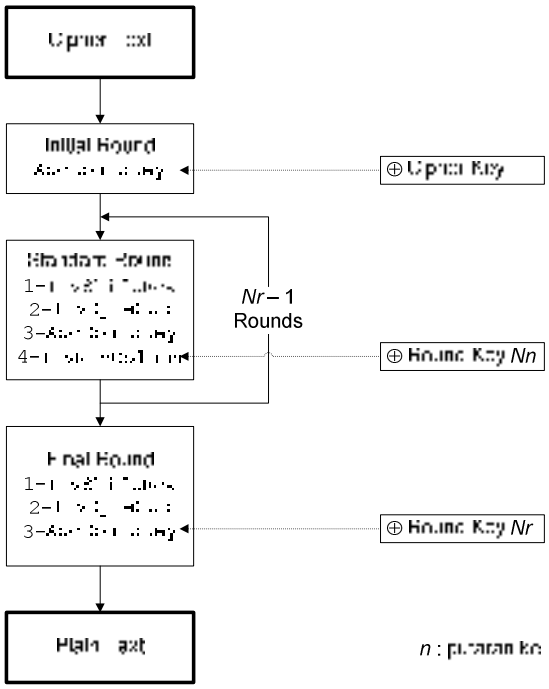
Gambar 5 Diagram Proses Enkripsi *AES*

4.3 Cipher Kebalikan (*Inverse Cipher*)

Cipher kebalikan merupakan kebalikan dari proses enkripsi, yaitu proses dekripsi dari *AES*.

Secara garis besar, *cipher* kebalikan yang beroperasi blok 128-bit dengan kunci 128-bit adalah sebagai berikut:

1. *AddRoundKey*: melakukan *XOR* antara *state* awal (cipherteks) dengan *cipher key*. Tahap ini disebut juga *initial round*.
2. Putaran sebanyak *Nr* – 1 kali. Proses yang dilakukan pada setiap putaran adalah:
   a. *InvShiftRow:* pergeseran baris-baris *array state* secara *wrapping*.
   b. *InvByteSub*: substitusi byte dengan menggunakan tabel substitusi kebalikan (*inverse S-box*).
   c. *AddRoundKey:* melakukan *XOR* antara *state* sekarang dengan *round key*.
   d. *InvMixColumn*: mengacak data di masing-masing kolom *array state*.
3. *Final round*: proses untuk putaran terakhir:
   a. *InvShiftRow*.
   b. *InvByteSub*.
   c. *AddRoundKey*.



Gambar 6 Diagram Proses Dekripsi *AES*

5. Pengujian

5.1 Perancangan Kasus Uji

Berdasarkan teori-teori dan batasan masalah yang diberikan diatas, maka dirancanglah beberapa kasus uji untuk diujikan pada program kecil hasil implementasi AES maupun Blowfish yang telah dibuat sebagai berikut:

1. Kasus Uji 1

Kasus uji 1 memberikan skenario pengujian kecepatan masing-masing algoritma dalam mengenkripsi dan mendekripsi file teks yang berukuran relatif kecil (192 kb).

2. Kasus Uji 2
   Kasus uji 2 memberikan skenario pengujian kecepatan masing-masing algoritma dalam mengenkripsi dan mendekripsi file multimedia yang berukuran relatif besar (162MB).

3. Kasus Uji 3
   Kasus uji 3 memberikan skenario dimana file hasil enkripsi menggunakan kedua algoritma dihapus beberapa bit sebanyak jumlah yang acak, lalu hasil akhirnya didekripsi kembali untuk melihat perubahannya.

4. Kasus Uji 4
   Kasus uji 4 memberikan skenario dimana file hasil enkripsi menggunakan kedua algoritma ditambah beberapa bit sebanyak jumlah yang acak, lalu hasil akhirnya didekripsi kembali untuk melihat perubahannya.

5. Kasus Uji 5
   Kasus uji 4 memberikan skenario dimana file hasil enkripsi menggunakan kedua algoritma dimodifikasi beberapa bit sebanyak jumlah yang acak, lalu hasil akhirnya didekripsi kembali untuk melihat perubahannya.

Selain itu perlu diketahui pula, lingkungan pengujian dilakukan pada sebuah PC berprosesor Pentium IV 3 GHz dengan sistem operasi Windows.

5.2 Evaluasi Hasil Pengujian

Dari kasus uji 1, hasil yang didapat adalah waktu yang digunakan oleh kedua algoritma relatif sama singkat, yaitu sama-sama dibawah 1 detik. Untuk mendapat perbedaan waktu yang lebih signifikan digunakanlah kasus uji 2 yang memiliki ukuran file lebih besar.

Kasus uji 2 memberikan perbedaan hasil yang cukup jauh, yaitu untuk AES proses enkripsi berlangsung selama 16,4 detik dan proses dekripsi selama 10,7 detik, meninggalkan jauh *Blowfish* yang hanya membukukan proses enkripsi selama 41,8 detik dan dekripsi 37,2 detik.

Dari hasil uji 2 dapat dilihat bahwa sebenarnya AES memiliki performansi yang jauh lebih tinggi daripada Blowfish baik dalam segi enkripsi maupun dekripsi.

Kasus uji 3 memberikan hasil, baik untuk AES maupun *Blowfish*, berupa sampai dengan bagian blok yang dikurangi beberapa bit, file teks tidak mengalami perubahan, sedangkan pada blok yang dikurangi terjadi perubahan beberapa karakter dan dilanjutkan rentetan blok-blok acak yang sama sekali tidak bersesuaian dengan file teks asal hingga akhir file.

Kasus uji 4 memberikan hasil serupa dengan kasus uji 3 untuk kedua algoritma, yaitu blok yang ditambah berubah beberapa karakter, seterusnya file menjadi acak hingga akhir file.

Kasus uji 5 memberikan hasil perubahan yang terjadi hanya terdapat pada blok yang dirubah. Perubahan tersebut hanya berupa ketidaksesuaian beberapa karakter saja, namun tidak terjadi perubahan sama sekali pada blok-blok sebelum dan sesudah bagian yang termodifikasi. Kedua algoritma sama-sama mendemonstrasikan kesalahan yang tidak jauh berbeda.

Dari kasus uji 3 dan 4 serta perbandingannya dengan kasus uji 5, dapat dianalisis bahwa kerusakan file yang muncul dikarenakan bergesernya titik awal semua blok setelah terjadinya penambahan atau pengurangan bit, sehingga proses dekripsi file memberikan hasil yang berbeda jauh dari file asal.

6. Kesimpulan

Kesimpulan yang dapat dimbil dari studi dan perbandingan algoritma enkripsi AES dengan *Blowfish* ini adalah:

1. *Advanced Encryption Standard* (*AES*) dan *Blowfish Cipher* dapat memenuhi kebutuhan untuk pengacakan data sehingga keamanan transmisi informasi yang sensitif dapat lebih terjaga.
2. *AES* memberikan performa yang jauh lebih cepar daripada *Blowfish* baik untuk proses enkripsi maupun dekripsi. Perbedaan waktu ini bisa mencapai perbandingan rasio kasar sebesar 1 :3.

3. Penambahan maupun pengurangan sebesar beberapa bit pada blok cipherteks *AES* maupun *Blowfish* akan merusak seluruh informasi yang terdapat dalam file tersebut.
4. Modifikasi beberapa bit hanya akan merusak informasi yang terkandung dalam blok tersebut saja.

DAFTAR PUSTAKA

[1] Munir, Rinaldi. (2009). Bahan Kuliah IF3058 Kriptografi. Program Studi Informatika, Institut Teknologi Bandung.

[2] Schneier, Bruce. (1993). Halaman *web* resmi dari *Blowfish Cipher*. *http://www.schneier.com/blowfish.html*. Tanggal akses: 30 Maret 2009 pukul 19:00.

[3] Schneier, Bruce. (1996). Applied Cryptography 2nd. John Wiley & Sons.

[4] *NIST*. (2009). National Institute of Standards and Technology. *http://www.nist.gov*. Tanggal akses: 30 Maret 2009 pukul 19:00.