Analisis dan Studi Kriptografi TwoFish

Adi Nugraha Setiadi - 13508062
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
If18062@students.if.itb.ac.id

Abstrak*—Twofish merupakan algoritma yang beroperasi dalam mode block. Algoritma Twofish sendiri merupakan pengembangan dari algoritma Blowfish. Twofish merupakan 128-bit block sandi/cipher yang bisa menerima panjang varibel kunci/key sebesar 256 bit. Cipher tersebut berasal 16-round jaringan Feitsel dengan fungsi bijektif F yang dilanjutkan dengan empat key-dependent 8-by-b-bit S-boxes, satu fixed 4-by-4 maximum distance separable matrix over GF(2^8), satu pseudo-Hadamard transform, satu rotasi bitwise, dan satu desain key schedule. Twofish dapat diterapkan pada perangkat keras dengan 14000 gerbang. Perancangan kedua putaran berfungsi untuk pemberian ijin jadwal kunci suatu tradeoff yang luas antara kecepatan, ukuran perangkat lunak, setting waktu, susunan gerbang, dan memori. Kriptanalisis Twofish memiliki sifat ekstensif, yakni serangan dapat dipecahkan dalam 5 putaran dengan $2^{22,5}$ pilihan plaintexts dan 2^{51} effort.*

Kata kunci*—Twofish*, Kriptografi, Kriptanalisis.

I. PENDAHULUAN

Dokumen tertulis dalam dunia teknologi informasi yang semakin berkembang pesat merupakan entitas yang perlu dijaga kerahasiaannya hanya bagi orang-orang yang berkepentingan. Selain itu dokumen juga harus dapat dijaga keasliannya agar tidak ada orang yang tidak berkepentingan merubah dokumen. Oleh karena itu selain algoritma untuk melakukan enkripsi, juga terdapat algoritma untuk melakukan *hashing* yang salah satu fungsinya adalah untuk menjaga keaslian dokumen tersebut. Algoritma MD5 merupakan salah satu algoritma *hashing* yang diciptakan untuk keperluan tersebut.

Pada tahun 1972 dan 1974, *National Bureau of Standart* (yang sekarang bernama NIST) mengumumkan adanya standar enkripsi, yaitu DES yang sangat beralasan karena penggunaannya yang luas dan merupakan algoritma yang sangat sukses di dunia Dalam proses perkembangannya ternyata *key-key* dalam DES dirasa terlalu pendek bagi keamanan komersial sehingga membuat gusar para kriptografer yang menginginkan proses algoritma yang "closed door". Triple*DES* telah muncul sebagai suatu solusi sementara dalam banyak aplikasi keamanan tingkat tinggi, seperti perbankan, tetapi itu terlalu lambat untuk beberapa penggunaan. Terlebih pada dasarnya, panjang 64-bit *clock* dihubungkan dengan *DES* dan hampir semua cipher well-known terbuka untuk serangan ketika sejumlah data yang besar dienkripsi dengan kunci yang sama.

Untuk menggantikan *NES, NIST* mengumumkan program *Advanced Encryption Standard (AES)* pada tahun 1997. *NIST* menunggu komentar dari publik atas standard yang diusulkan, dan secepatnya mengeluarkan suatu algoritma untuk memenuhi standard tersebut. Keinginan *NIST* adalah untuk membuat publik mengetahui secepatnya, melalui suatu proses tinjauan kembali dari publik dan pemberian komentar, untuk memilih suatu standar enkripsi baru menggantikan *DES*.

NIST menggunakan suatu block cipher. *Block cipher* dapat digunakan untuk merancang aliran cipher dengan berbagai sinkronisasi untuk properti-properti kesalahan yang luas, sebuah cara fungsi hash, kode pengesahan pesan, dan pseudo-random generator nomor. *NIST* menetapkan beberapa kriteria disain lain: kunci yang lebih panjang, ukuran blok lebih luas, lebih cepat, dan exibilitas lebih besar. Selagi tidak ada algoritma tunggal yang dapat dioptimalkan untuk semua kebutuhan, *NIST* menjadikan *AES* sebagai standar algoritma simetris untuk dekade selanjutnya. Salah satu kandidat AES adalah Twofish. Mengapa demikian? Karena Twofish memenuhi semua criteria yang dibutuhkan NIST, yaitu 128-bit block, 128 bit, 192 bit dan 256 bit key (kata kunci), efisien pada plaform manapun dan lain-lain, serta beberapa desain berat lainnya Twofish dapat melakukan:

a. Melakukan enkripsi data pada 285 siklus per block di atas Pentium Pro setelah menjalankan key setup 12700 siklus clock.

b. Melakukan enkripsi data pada 860 siklus per blok sdi atas Pentium Pro setelah menjalankan key setup 1250 siklus clock.
c. Melakukan enkripsi data pada 26500 siklus per block di atas sebuah 6805 smart card setelah mejalankan key setup 1750 siklus clock.

II. TWOFISH BUILDING BLOCKS

Sebuah Fietsel *Network* adalah metoda umum untuk mentransformasi suatu fungsi menjadi bentuk permutasi. Bagian paling fundamental dari Jaringan Fietsel adalah fungsi F: sebuah pemetaan key-dependent dari suatu input string menjadi output string. Dalam Twofish dilakukan Fietsel *Network* sebanyak 16 kali. *Procedure* Fietsel *Network* sebenarnya terdiri dari Input *Whitening*, S-*boxes*, Transformasi Pseudo Hadamard, Output dan Output *Whitening*.

S-Box

Sebuah S-box adalah operasi subsitusi table-driven non linear yang digunakan dalam block chipper. S-boxes bervariasi antara setiap ukuran input dan ukuran outputnya, dan bisa diciptakan secara random 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.

MDS Matrix

Code Maximum Distance Separable(MDS) melalui a adalah pemetaan linear dari elemen *field* a ke elemen *field* b, menghasilkan campuran dari *vector* a + b elemen, dengan property jumlah minimum angka tidak nol dalam vector tidak nol paling kurang b + 1. Dengan kata lain "Distance" adalah jumlah element 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 element. Twofish menggunakan matriks MDS 4 x 4 tunggal.

Transformasi Pesudo-Hadamard

Transformasi Pseudo-Hadamard (PHT) adalah operasi sederhana yang bekerja dengan cepat dalam software. Diberikandua input, 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 outputnya dari dua buah fungsi g 32 bit parallel. PHT ini dapat dieksekusi dalam dua opcode diatas kebanyakan microprocessor modern, termasuk keluarga Pentium.

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 input spesifik untuk awal dan akhir ronde dari Twofish.

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 argument, 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 di-*expand* 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 non linear dan kemungkinan non surjektif, yaitu bahwa tidak semua output yang dimungkinkan berada dalam ruang output dapat terjadi semua.

Key Schedule

Key schedule adalah suatu cara dimana bit-bit *key* diubah menjadi *key-key* 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. Key yang lebih pendek dari 256 bit dapat dipergunakan dengan cara mengisinya dengan nilai nol samapai panjang kunci yang lebih besar berikutnya.

III. TEKNIK KRIPTOGRAFI TWOFISH

Gambar 1 menunjukkan suatu diagram dari operasi blok *cipher Twofish*. Twofish menggunakan struktur Feistel 16-round dengan *whitening* tambahan dalam *input* dan *output*-nya. Satu-satunya elemen yang bukan Feistel adalah rotasi 1 bit. Rotasi tersebut dapat dipindahkan ke fungsi *F* untuk menciptakan output berjalan.

Plaintext dipecah menjadi empat buah *word* 32-bit. Pada *whitening input*, keempat *word* itu di XOR-kan dengan empat key word. Dan diikuti dengan keenam belas round. Dalam tiap *round*, dua *word* di kiri digunakan sebagai *input* fungsi *g* (Salah satunya dirotasikan dengan 8 bit terlebih dahulu).

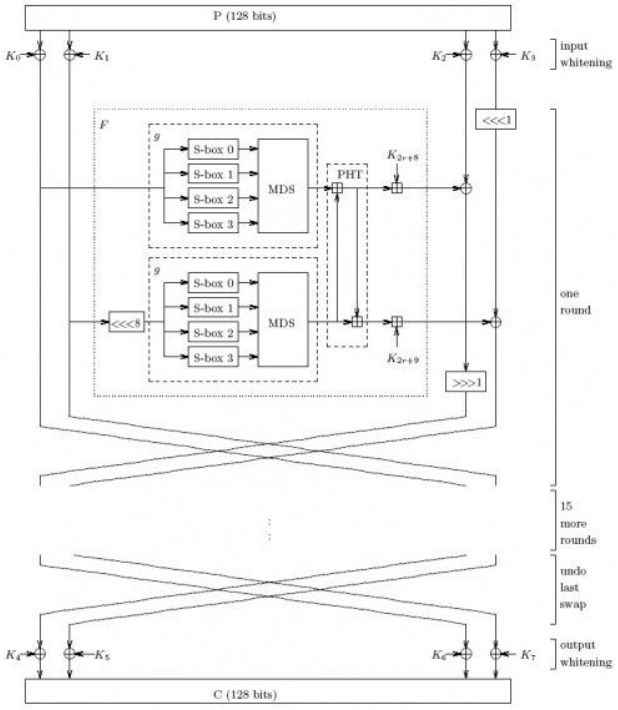

Figure 1: Twofish

IV. KINERJA DARI 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.

Telah diimplementasikan empat macam pilihan *keying* yang berbeda. Terdapat beberapa pilihan *keying* yang mungkin, dimana masing-masing mempunyai perbedaan tipis dalam hal *key* setup.

Full Keying. Pilihan ini melakukan prekomputasi terhadap *key*. 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 *key*.

Partial Keying. Untuk aplikasi dimana sebagian kecil blok dienkripsi dengan *key tunggal*, tidak akan menjadi masalah dalam membangun *key schedule* yang lengkap. Pilihan ini melakukan prekomputasi 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 *key*.

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

Zero Keying. Pilihan ini tidak melakukan prekomputasi 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*.

Perbandingan kinerja dari TwoFish dengan Blok Cipher lain

Algorithm	Key Length	Width (bits)	Rounds	Cycles	Clocks/Byte
Twofish	variable	128	16	8	18.1
Blowfish	variable	64	16	8	19.8
Square	128	128	8	8	20.3
RC5-32/16	variable	64	32	16	24.8
CAST-128	128	64	16	8	29.5
DES	56	64	16	8	43
Serpent	128, 192, 256	128	32	32	45
SAFER (S)K-128	128	64	8	8	52
FEAL-32	64, 128	64	32	16	65
IDEA	128	64	8	8	74
Triple-DES	112	64	48	24	116

V. KRIPTANALIS TERHADAP TWOFISH

Lebih dari seribu jam telah dilakukan *cryptanalist* terhadap twofish. Sebuah catatan penting dari attack yang berhasil dilakukan terhadapnya adalah sebagai berikut:

a. 5- round twofish dengan $2^{22,5}$ pasangan plaintext terpilih dan 2^{51} komputasi fungsi g
b. 10 round twofish dengan sebuah chosen key attack,memerlukan 2^{32} palintext terpilih dan sekitar 2^{32} chosen –plaintext yang adaptif dan sekitar 2^{32} usaha.

Fakta bahawa twofish mampu menahan related *key* attack dengan baik merupakan fakta yang paling menarik dan beralasan karena *related-key* memberikan kepada *attacker* hampir semua kontrol terhadap *input cipher*. Cryptanalis konvensional memungkinkan suatu attacker mengontrol input *plaintext* dan *ciphertext* didalam cipher, yaitu *key-schedule*.
Berdasarkan hasil analisis ini, dapat diterka bahwa tidak lagi terdapat attack yang efisien terhadap twofish selain *brute force*., yaitu *attack* yang paling efisien untuk melawan twofish dengan key 128 bit harus memiliki kompleksitas 2^{128}, sementara untuk twofish dengan *key* 192 bit harus menggunakan attack dengan kompleksitas 2^{192} dan untuk twofish dengan key 256 bit harus menggunakan attack dengan kompleksitas 2^{256}.

Salah satu attack yang dibahas disini adalah adalah *Partial Key Guessing Attack*. Sebuah *key Schedule* yang bagus harus memiliki *property* dimana ketika *attacker* menebak beberapa subset dari bit-bit *key*, *attacker* tidak memahami tentang urutan subkey atau operasi internal lainnya di dalam cipher tersebut. Dan twofish memiliki tipe *key schedule* seperti itu.

Dianggap ada sebuah attacker yang menebak suatu *word* genap dari key M_c. Attacker tidak mempelajari apapun dari key S, untuk tiap *round* blok *subkey* , ia mengetahui A_i. Jika ia menebak dengan suatu K_o, ia dapat menghitung K_1 yang besesuaian. Ia dapat melakukan attack round subkey sebanyak yang dia suka, tapi tiap tebakan memakan 32 bit. Dapat dilihat bahwa tidak ada jalan bagi attacker untuk menguji tebakan 96 bit sekalipun hanya satu *round subkey* dengan cara ini terhadap *full* Twofish.

Jalan lainnya adalah dengan menebak *input* key S terhadap G. Cara ini hanyalah setengah jalan dari full key M , tapi tidak memberikan informasi tentang round key A_I. Dapat dilihat bahwa dengan cara ini pun tak ada jalan bagi *attacker* untuk menguji tebakan s terhadap twofish 16 round yang penuh sehingga analisis menyarankan bahwa *attack* terhadap *full* Twofish dengan menggunakan diferensial *reated key*, adalah suatu pekerjaan yang sia-sia karena resistensi cipher Twofish yang handal terhadap attacker sejenis.

VI. ANALISIS

Twofish adalah cipher blok 128 bit yang menerima key dengan panjang variabel diatas 256 bits. Cipher tersebut merupakan sebuat network Feitsel 16 round dengan suatu fungsi bijektif F yang membuat empat buah *key dependent* s-*boxes* 8 x 8 , jarak maksimum 4 x 4 yang dapat dipisahkan atas GF(2^8) suatu transformasi pseudo-Hadamard, rotasi *bitwise*, dan penjadwalan key dengan seksama.

Sementara itu , pondasi dasar dari jaringan Feistel adalah fungsi F , yaitu suatu permutasi yang key-dependent terhadap nilai 64 bit. Dari fungsi F ini dihasilkan output-output yang kemuduan diolah lagi dalam fugsi g. Dan melalui proses – proses yang lain lagi sehingga dihasilkan sebuah cipher blok 128 bit yang handal dalam menghadapi *attack – attack related key*. Dan perlu diketahui bahwa fungsi f selalu non linear dan kemudian non surjektif , yaitu bahwa tidak semua output yang dimungkinkan berada dalam ruang output dapat terjadi semua.

Twofish memiliki kehandalan-kehandalan dalam implementasinya diatas berbagai platform *microprocessor*, smart card dan hardware yang dibuat sebagai perangkat enkripsi data. Hal ini dapt dilihat dari tabel – tabel yang menunjukkan kinerja twofish secara keseluruhan dalam berbagai format *message* dan ukuran *key*. Kinerja twofish pada microprocessor besar meliputi pilihan- pilihan performance, yaitu full keying dan partial keying , minimal keying, *zero keying* dan *compiled*. Dan dengan jelas telah ditunjukkan bahwa dengan zero keying , dan kehandalan twofish diatas mikroprosessor besar sangat optimal. Namun hal tersebut memiliki faktor- faktor yang penting yaitu ukuran *plaintext*, pilihan *keying*, dan pemakaian waktu dalam *key setup*.

Beberapa cryptanalis yang dilakukkan terhadap cipher blok twofish memberikan hasil yang menakjubkan, dimana attack yang diterapkan dengan menggunakan *related key* dinyatakan sia-sia karena untuk bisa menembus pertahanan twofish diperlukan attack yang berupa brute force. Berdasarkan hasil analisis ini , dapat diterka bahwa tidak lagi terdapat attack yang lebih efisien terhadap twofish selain *brute force*. Yaitu *attack* yang paling efisien untuk melawan twofish dengan key 128 bit harus memliki kompleksitas $2^{128,}$ sementara untuk twofish dengan key 192 bit harus mengunakan *attack* dengan kompleksitas 2^{192} dan

untuk Twofish dengan key 256 bit harus menggunakan attack dengan kompleksitas 2^{256} .

Dari sini dapat disimpulkan bahwa algoritma enkripsi twofish yang merupakan cipher blok 128 bit dan dijdikan kandidat AES adalah algoritma enkripsi yang memilki kehandalan dan keunggulan. implementasi pada berbagai macam platform mikroprocessor, beragam smart card dan *hardware-hardware* yang dispesifikasikan untuk proses enkripsi. Hal ini disebabkan oleh keunggulannya dalam menggunakan siklus *clock* dan kebutuhan spesifikasi hardware dalam implementasinya. Selain itu, twofish memiliki resistensi yang tinggi terhadap *related key attack*, dan hanya dapat ditembus dengan menggunakan *brute force*. Maka Twofish merupakan kandidat AES yang handal.

VII. IMPLEMENTASI TWOFISH

Meski tidak terpilih menjadi algoritma AES, TwoFish cukup bayak diimplementasikan ke dalam perangkat-perangkat lunak penyandian data. Beberapa perangkat lunak yang dimplementasikan TwoFish antara lain:

1. Away32 Deluxe and Away IDS Deluxe
   pengenkripsi file dan folder untuk Windows.
2. Bassline WinPopUp
   Program Internet Manager
3. Cryptix
   Ekstensi kriptografi untuk Java.
4. Crypto++
   Pustaka kelas C++ untuk TwoFish.
5. CuteZIP
   Kompresi file dan audio ripping menggunakan TwoFish 128 bit.
6. Foxtrot
   HTTP Server didesain sebagai application server professional. Foxtrot memfasilitasi eksekusi *query* SQL langsung dari Address Line browser serta *secure transaction* menggunakan TwoFish.
7. Password Creator Professional
   Mengimpelementasikan TwoFish untuk memunculkan *password* secara acak.
8. XPDFViewer
   Menggunakan TwoFish untuk mengenkripsi file.

Implementasi TwoFish dalam bahasa C disediakan oleh pembuatnya, Bruce Scheneier, dalam situsnya, http://www.schneier.com

VIII. KESIMPULAN

Twofish telah dipresentasikan sebagai disain dan cryptanalysis yang berjalan dari tangan ke tangan , dalam artian mustahil melakukan suatu hal tanpa hal yang lain dikerjakan juga dan hanya dalam analisis yang kemampuan algoritmanya dapat didemonstrasikan. Selama proses disain, telah dipelajari beberapa pengetahuan mengenai desain cipher, yang disajiakan sebagai berikut

- Algoritma enkripsi dan key schedule harus didisain secara tandem (bersamaan), yaitu perubahan yang tak kentara dapat mempengaruhi proses yang lain. Dirasa tidak cukup mendesain fungsi round yang hebat dan kemudian dilanjutkan dengan mencabangkan key schedule yang hebat pula padanya(kecuali jika sudah puas dengan konstruksi yang kurang efisien dan kurang elegan yang dimiliki oleh blowfish) Jadi keduanya harus berjalan bersamaan.
- Tak terdapat suatu hal serupa dengan key dependent s-*box*, yang ada hanyalah fungsi non linear multi *stage* yang rumit yang diimplementasikan sebagai sebuah *key* dependent s-box untuk efisiensi.
- Key harusnya dibuat sependek mungkin. Dianggap sangat berat untuk mendesain suatu algoritma dengan *key* panjang jika dibadingkan dengan algoritma yang menggunakan *key* yang pendek . Dari keseluruhan proses desain ini , dirasa lebih mudah mendesain dan menganalisa Twofish dengan sebuah *key* 128 bit daripada Twofish yang menggunakan key 192 bit ataupun key 256 bit.
- Membangun sebuah cipher dengan enkripsi lokal yang kuat dan membiarkan fungsi pembulatan menangani difusi global . Perancangan Twofish dalam hal ini menjadikannya sangat sukar untuk menyusun *statistical cryptanalis attack*.

Diangap bahwa Twofish merupakan algoritma yang ideal bagi AES. Karena Twofish efisien pada mikroprosesor besar, smart card dan hardware yang dikhususkan untuk itu. Selain itu juga karena twofish memiliki *key schedule* yang handal dan membuatnya cocok untuk bermacam–macam implementasi . perhatian terhadap detail, baik sisi fungsi enkripsi, maupun sisi key schedule, membuatnya layak sebagai *codebook*, *output feedback*, dan fungsi hash satu arah serta *pseudo random number*.

REFERENCES

[1] http://www.schneier.com/twofish.html
[2] http://en.wikipedia.org/wiki/Twofish
[3] http://www.tropsoft.com/strongenc/twofish.htm
[4] Rinaldi Munir (2010). IF3058 Kriptografi.
[5] C. Adams, “Constructing Symmetric Ciphers Using the CAST Design Procedure,"Designs, Codes and Cryptography, v.12, n.3,Nov 1997.

[6] M. Bellare, R. Canetti, and H. Karwczyk, \Keying Hash Functions for Message Authentication," Advances in Cryptology |CRYPTO '96 Proceedings, Springer-Verlag, 1996.

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, 9 Mei 2010

Adi Nugraha Setiadi