Studi Perbandingan Algoritma Kunci-Simetris Serpent dan Twofish

Moch. Yusup Soleh / 13507051[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]if17051@students.if.itb.ac.id

***Abstrak*—Makalah yang dibuat membahas tentang studi perbandingan algoritma kunci-simetri Serpent dan Twofish yang mendapatkan peringkat ke-2 dan ke-3 dalam kompetisi Advanced Encryption System (AES), yaitu menyangkut tentang kelebihan dan kekurangan masing-masing algoritma tersebut. Algoritma Serpent dirancang oleh Ross Anderson, Eli Biham dan Lars Knudsen. Sedangkan algoritma Twofish dirancang oleh Bruce Schneier yang merupakan kelanjutan dari algoritma sebelumnya yang telah dirancang, yaitu Blowfish. Kedua algoritma ini memiliki panjang blok yang sama yaitu 128 bit dengan panjang kunci 128 bit, 192 bit atau 256 bit.**

***Kata Kunci*—Serpent, Twofish, Cipher Block, Kunci Simetris.**

I. PENDAHULUAN

Keamanan data merupakan salah satu aspek penting dalam era teknologi seperti saat ini, terutama dalam masalah pengiriman dan penerimaan data. Salah satu upaya untuk menjamin keamanan data ini adalah dengan menjaga kerahasiaan data tersebut, sehingga hanya orang-orang atau instansi tertentu saja yang dapat mengelola data tersebut. Hal ini dapat dilakukan dengan melakukan enkripsi dan dekripsi terhadap data tersebut.

Ilmu yang khusus mendalami teknik untuk menjaga kerahasian data ini dinamakan Kriptografi. Algoritma kriptografi ialah mengubah tulisan yang semula bermakna menjadi tidak bermakna isinya dengan menggunakan berbagai algoritma yang dirancang kriptografer. Kriptografi telah berkembang sejak zaman dahulu. Pada saat ini kriptografi terus berkembang pesat sehingga algoritma-algoritma kriptografi yang ada semakin bertambah jumlahnya.

Dalam Kriptografi modern, algoritma-algoritma yang dibuat oleh para kriptografer semakin rumit. Hal ini karena algoritma-algoritma pada kriptografi modern beroprasi pada mode bit, dimana pada kriptografi klasik beroprasi pada mode karakter. Salah satu metode kriptografi mode bit adalah metode block cipher, dimana setiap masukan akan dibagi menjadi blok-blok untuk selanjutnya dienkripsi atau didekripsi. Diantara algoritma *block cipher* yang dikenal adalah algoritma Serpent dan Twofish, dimana ke dua algoritma ini pernah menjadi finalis dalam kompetisi *Advanced Encryption Standard* (AES) dengan masing-masing menduduki peringkat ke-2 untuk algoritma Serpent, dan peringkat ke-3 untuk Twofish.

II. ALGORITMA KUNCI-SIMETRIS CIPHER BLOCK

Algoritma kunci simetris adalah algoritma kriptografi dimana kunci yang digunakan dalam mengenkripsi dan kunci yang digunakan dalam mendekripsi adalah sama. Algoritma kunci simetris ini dibagi menjadi dua, yaitu *cipher stream* dan *cipher block* dimana keduanya merupakan algoritma kriptografi modern yang beroperasi dalam mode bit.

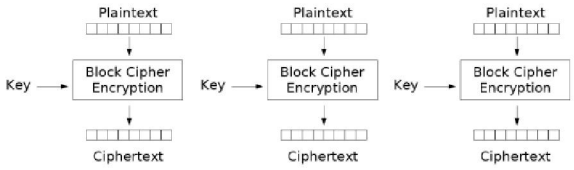
Cipher stream adalah algoritma kunci simetris dimana enkripsi dilakukan pada aliran (*stream*) bit dari data yang akan dienkrip maupun didekrip. Dengan kata lain algoritma ini mengenkripsi setiap bit dari plain teksnya satu per satu.

Cipher block adalah algoritma kunci simetris dimana enkripsi dilakukan pada blok-blok bit data yang akan dienkrip maupun didekrip. Blok merupakan kumpulan bit dengan ukuran tetap yang terdapat pada data.

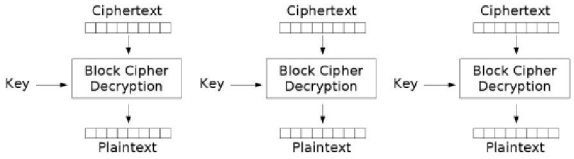
A. Mode Cipher Block

Pada algoritma cipher block enkripsi yang dilakukan terhadap blok-blok dilakukan dengan berbagai mode pengoperasian. Terdapat beberapa mode dalam cipher block, namun yang sering digunakan adalah:

1. Electronic Codebook (ECB)
   Pada mode ini, pengenkripsian atau pendekripsian dilakukan pada setiap blok dari data secara independent. Sehingga setiap blok yang memiliki rangkaian bit yang sama akan menghasilkan blok hasil yang sama.

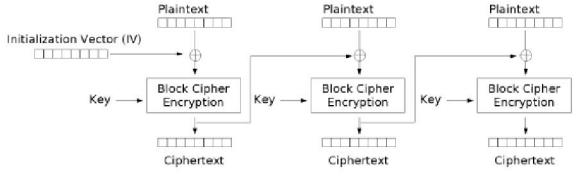


Gambar 1. Skema Enkripsi dengan mode ECB

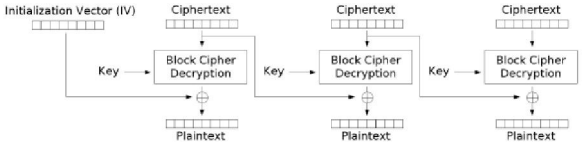


Gambar 2. Skema dekripsi dengan mode ECB

2. Cipher Block Chaining (CBC)
   Pada mode ini, enkripsi dilakukan setelah sebelumnya blok data yang akan dienkripsi di-XOR-kan dengan blok data hasil sebelumnya (khusus untuk blok data pertama akan di-XOR-kan dengan Initial Vector). Demikian halnya dengan dekripsi, data blok dari cipher teks akan didecrypt terlebih dahulu kemudian di-XOR-kan dengan data blok sebelumnya (khusus untuk blok data pertama akan di-XOR-kan dengan Initial Vector).

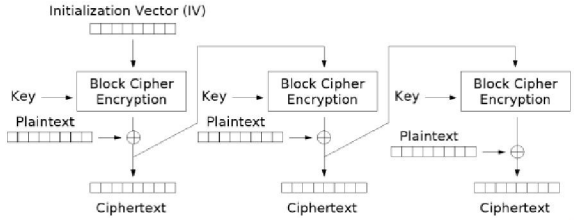


Gambar 3. Skema enkripsi dengan mode CBC

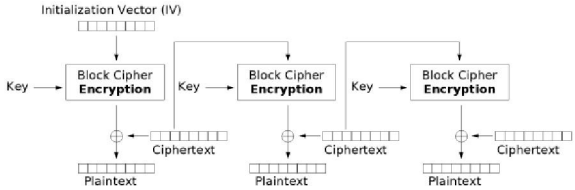


Gambar 4. Skema dekripsi dengan mode CBC

3. Cipher Feedback (CFB)
   Pada mode ini, enkripsi dilakukan dengan men-XOR-kan blok data dengan blok data hasil sebelumnya (kecuali untuk blok data pertama diganti dengan inisial vektor) yang sudah terenkripsi. Sedangkan untuk dekripsi dengan men-XOR-kan blok data dengan blok data sebelumnya (kecuali untuk blok data pertama diganti dengan inisial vektor) yang dienkripsi terlebih dahulu.

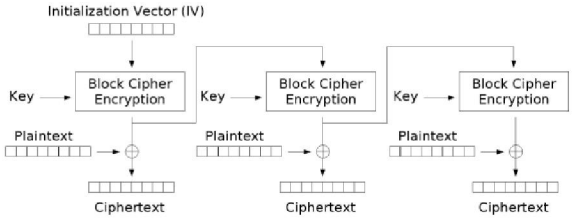


Gambar 5. Skema enkripsi dengan mode CFB

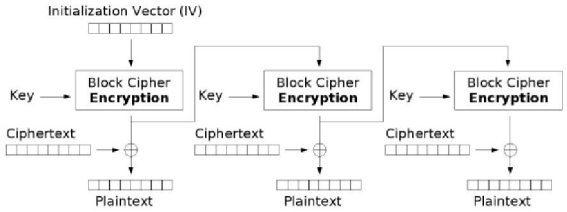


Gambar 6. Skema dekripsi dengan mode CFB

4. Output Feedback (OFB)
   Pada mode ini, baik untuk enkripsi maupun dekripsi dilakukan dengan men-XOR-kan blok data dengan hasil enkripsi inisial vektor untuk blok data pertama, untuk blok data selanjutnya dengan hasil enkripsi dari hasil enkripsi sebelumnya.



Gambar 7. Skema enkripsi dengan mode OFB



Gambar 8. Skema derkipsi dengan mode OFB

B. Prinsip Perancangan

Pada perancangan algoritma *cipher block* digunakan prinsip-prinsip sebagai berikut:

1. Prinsip *confussion* dan *diffusion* dari Shannon
   Fungsi dari *confussion* adalah untuk menyembunyikan keterkaitan antara pain teks, cipher teks dan kunci. Sedangkan *diffusion* berfungsi agar setiap satu bit yang berubah akan memperngaruhi bit-bit lainya.
2. Cipher berulang
   Prinsip ini adalah mengulang proses atau metode dalam pengenkripsian atau pendekripsian selama beberapa kali. Fungsi dari cipher berulang ini adalah untuk menambah ketidak teraturan dari cipher teks yang dihasilkan.
3. Jaringan Feistle
   Prinsip ini adalah membuat suatu model jaringan yang bersifat *reversible*. Fungsi dari jaringan feistle ini adalah untuk menambah kerumitan algoritma.
4. Kunci lemah
   Algoritma yang memiliki kunci lemah adalah

algoritma yang mana jika lakukan enkripsi plain teks dengan menggunakan kunci yang sama akan menghasilkan plain teks itu kembali.

5. Kotak-Substitusi (S-Box)
   Kotak-Substitusi atau yang lebih dikenal sebagai S-Box adalah sebuah kotak atau tabel sederhana yang memetakan sebuah nilai blok data kedalam nilai lainnya.

III. STUDI ALGORITMA

A. Serpent

Algoritma cipher block Serpent adalah algoritma dengan 32 putaran jaringan SP yang beroperasi pada empat *word* 32 bit, yang berarti ukuran bloknya adalah 128 bit. Untuk komputasi internal, semua nilai direpresentasikan dalam *little-endian*, di mana word pertama adalah *least-significant word*, dan *word terakhir* adalah *most-significant word*.

Algoritma Serpent mengenkripsi plainteks P 128 bit menjadi cipherteks C 128 bit dalam 32 putaran dengan kontrol dari 33 sub-kunci 128 bit K0,...,K32. Panjang kunci masukan user 128, 192, dan 256 bit. Kunci yang lebih pendek dari 256 bit dipetakan menjadi kunci sepanjang 256 bit dengan menambahkan satu "1" bit pada akhir MSB, dan diikuti dengan "0" bit sampai mencapai 256 bit.

Algoritma Serpent ini terdiri dari:

1. Initial Permutation (IP)
2. Terdiri dari 32 putaran, masing-masing terdiri dari sebuah operasi pengacakan kunci, operasi menggunakan S-Box, dan transformasi linear. Pada putaran terakhir, transformasi ini digantikan dengan penambahan operasi pengacakan kunci.
3. Final Permutition (FP)

IP atau Initial Permutation diterapkan pada plainteks P menghasilkan B0, yang merupakan input dari putaran pertama, yaitu putaran-0 (putaran diberi nomor dari 0 sampai 31). Hasil dari putaran pertama (putaran-0) dinamakan B1, hasil putaran kedua (putaran-1) dinamakan B2, dan seterusnya sampai B32. Permutasi akhir akan menghasilkan cipherteks C.

Masing-masing fungsi putaran Ri (i = 0,...,31) hanya menggunakan sebuah S-Box ter-replikasi. Misalnya, R0 menggunakan S0, 32 copy yang diterapkan secara paralel, sehingga copy dari S0 menggunakan bit 0,1,2, dan 3 dari B0 $\oplus$ K0 sebagai input dan mengembalikan empat bit pertama dari vektor intermediate sebagai output, copy selanjutnya menerima masukan bit ke 4-7 dari B0 $\oplus$ K0 dan mengembalikan empat bit selanjutnya dari vektor intemediate, dan seterusnya. Vektor intermediate kemudian ditransformasi menggunakan linear transformasi, menghasilkan B1. Dengan cara yang sama, R1 menggunakan 32 copy S1 secara paralel pada B1 $\oplus$ K1 dan mentransformasi outputnya menggunakan transformasi linear, menghasilkan B2.

Himpunan delapan S-Box digunakan sebanyak empat kali. Oleh karena itu, setelah menggunakan S7 pada putaran-7, S0 digunakan kembali pada putaran-8, kemudian S1 pada putaran-9, dan seterusnya. Putaran terakhir R31 sedikir berbeda dengan lainnya, yaitu dengan menggunakan S7 pada B31 $\oplus$ K31, dan meng-XOR-kan hasilnya dengan K32, bukan menggunakan transformasi linear. Hasil B32 kemudian dipermutasikan dengan FP, menghasilkan cipherteks.

Setiap putaran menggunakan 9 S-Box yang berbeda yang memetakan empat input bit ke empat output bit. Masing-masing S-Box digunakan tepat pada empat putaran, dan masing-masing digunakan 32 kali secara paralel.

Final Permutation (FP) adalah invers dari permutasi inisial. Dengan demikian, cipher secara formal dapat dideskripsikan sebagai berikut.

Bo := IP(P)
Bi+1 := Ri(Bi)
C := FP(B32)

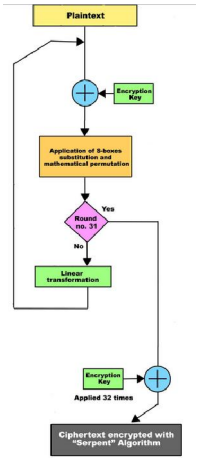
di mana:
Ri(X) = L(S(X $\oplus$ Ki)) i = 0,...,30
Ri(X) = Si(S $\oplus$ Ki) K32 i = 31

di mana Si adalah aplikasi S-Box Si mod 8 32 kali secara paralel, dan L adalah transformasi linear.

Substitution-Box atau S-Box Serpent adalah permutasi 4 bit dengan ketentuan sebagai berikut.

1. Setiap karakteristik diferensial memiliki probabilitas maksimal ¼, dan sebuah input dengan perbedaan satu bit tidak akan meghasilkan output dengan perbedaan satu bit
2. Setiap karakteristik linear memiliki probabilitas antara ½ ± ¼, dan hubungan linear antara sebuah bit pada input dan sebuah bit pada output memiliki probabilitas ½ ±1/8
3. Urutan non-linear bit output sebagai fungsi dari bit input maksimal 3.

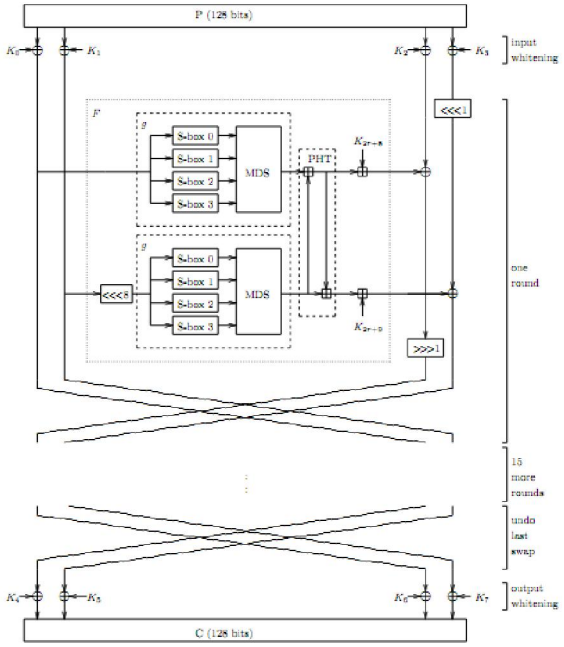
S-Box dibangkitkan dengan menggunakan matriks dengan 32 array yang masing-masing memiliki 16 entri. Matriks diinisialisasi dengan 32 baris S-Box DES dan ditransformasikan dengan menukar entri pada array ke-r bergantung pada nilai entri ke-(r+1) array dan pada inisial string yang merepresentasikan kunci. Jika array hasilnya memenuhi ketentuan yang telah disebutkan sebelumnya, maka simpan array sebagai Serpent S-Box. Ulangi prosedur tadi sampai 8 S-Box berhasil dibangkitkan.



Gambar 7. Skema Algoritma Serpent

B. Twofish

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.



Gambar 8. Skema algoritma twofish

Struktur Feistel 16-round pada Twofish, terdapat penambahan *whitening* pada masukan dan keluaran. untuk Untuk membuat suatu struktur Feistel murni perputaran dapat dipindah ke dalam fungsi F, tapi memerlukan suatu tambahan perputaran *word-word* yang tepat sebelum keluaran *whitening*. Plain teks dipecah menjadi empat *word* 32-bit. Pada langkah *whitening*, ke empat *word* ini di-XOR-kan dengan empat *word* dari kunci. Ini diikuti oleh 16 putaran. Pada setiap putaran, dua *word* pada sisi kiri digunakan sebagai masukan kepada fungsi g (Salah satunya diputarkan pada 8 bit pertama). Fungsi g terdiri dari empat byte-wide S-Box key-dependent, yang diikuti oleh suatu langkah pencampuran linier berdasar pada suatu matriks *MDS*. Hasil kedua fungsi g dikombinasikan menggunakan suatu *Pseudo Hadamard Transform (PHT)*, dan ditambahkan dua *word* kunci. Kedua hasil ini kemudian di-*XOR* ke dalam *word-word* pada sisi kanan (salah satunya diputarkan ke kanan 1 bit pertama, yang lainnya diputarkan ke kanan setelahnya). Bagian kiri dan kanan dibelah dua kemudian ditukar untuk putaran berikutnya, pertukaran yang terakhir dilakukan dengan cara dibalik, dan empat word di-*XOR* dengan lebih dari empat *word* kunci untuk menghasilkan cipher teks. Secara formal, 16 byte plain teks $p_0,\ldots\ p_{15}$ yang yang pertama dipecah menjadi 4 *word* $P_{0,\ldots}P_3$ dari 32 bit masing-masing menggunakan konvensi *little-endian.*

$$\sum_{j=0}^{3} P(4i+j).2^{8j} \quad i=0,\ldots,3$$

Di dalam langkah w*hitening*, *word-word* ini di-*XOR*-kan dengan 4 *word* dari kunci yang diperluas.

$$R_{0,i}=P \oplus K_i \qquad i=0,\ldots,3$$

B.1. Fungsi F

Fungsi F adalah permutasi kunci terhadap nilai 64-bit. Fungsi ini memerlukan tiga argumen, dua *word* R0 dan R1, dan *round-key* ke-r. R0 dilewatkan ke fungsi g, dimana akan menghasilkan T0. R1 diputar kiri sebanyak 8 bit kemudian dilewatkan ke fungsi g untuk menghasilkan T1. Hasil dari T0 dan T1 dikombinasikan dalam PHT dan dua *word* dari kunci yang telah diekspand ditambahkan.

T0 = g(R0)
T1 = g(ROL(R1; 8))
F0 = (T0 + T1 + K2r+8) mod 232
F1 = (T0 + 2T1 + K2r+9) mod 232

dimana (F0; F1) adalah hasil dari F.

B.2. Fungsi g

Inti dari fungsi f adalah memetakan setiap *byte* dari *word* pada S-Box-nya masing-masing. Hal ini dilakukan dengan cara memcah masukan untuk fungsi g ini menjadi 4 buah masing-masing 1 byte (8bit) kemudian dipetakan terhadap S-Box.

B.3. Maximum Distance Separable (MDS)

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 [MS77]. 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.

B.4. Pseudo Hadamard Transform (PHT)

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

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

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

B.5. Penjadwalan Kunci

Penjadwalan kunci adalah suatu cara dimana bit-bit key diubah menjadi key-key bulat yang dapat digunakan oleh cipher. 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.

Penjadwalan kunci harus menyediakan 40 word, yaitu key $K_0 ... K_{39}$ 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.

B.6. Whitening

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

IV. ANALISIS DAN PERBANDINGAN

Perbedaan yang paling sederhana dilihat dari kedua algoritma block cipher di atas adalah perbedaan jumlah putaran. Namun hal tersebut tidak terlalu relevan dalam segi keinformatikaan.

Perbandingan yang dilakukan antara algoritma kriptografi adalah dalam hal keamanan dan algoritma nya itu sendiri.

Algoritma block cipher Serpent dan Twofish keduanya didesain untuk memenuhi kebutuhan *Advanced Encryption Standard* (AES). Salah satu diantaranya, keduanya merupakan block cipher 128 bit. Meskipun spesifikasinya sama namun arsitektur dari kedua algoritma tersebut berbeda dan juga keduanya didesain dengan kelebihan masing-masing dalam platform-platform tertentu dalam lingkungan yang berbeda.

Twofish didasarkan pada jaringan Feistle sedangkan Serpent didasarkan pada jaringan substitusi-permutasi, atau SPN, yang mana kebanyakan *Advanced Encryption Algorithm* atau AEA yang dimasukan dalam ajang AES merupakan struktur berbasis jaringan Feistle atau modifikasi dari jaringan tersebut. Hal ini ditujukan untuk memenuhi kebutuhan akan keamanan.

Secara umum, keamanan suatu block cipher dapat dilihat berdasarkan berapa banyak putaran atau *round* yang sudah berhasil dikriptanalisis, dibandingkan dengan total jumlah putaran yang dipakai dalam algoritma block cipher tersebut. Dalam hal ini, Serpent memiliki rentang keamanan yang cukup besar dibanding dengan twofish, yang juga memiliki rentan keamanan yang lebih besar dari pada Rijndael (AEA yang terpilih).

Jumlah putaran yang banyak tidak selalu mempertinggi keamanan suatu algoritma block cipher. Dengan kata lain, semakin banyak jumlah putaran belum tentu semakin aman. Namun jumlah putaran yang kecil juga rantan, kemungkinan dikriptanalisisnya sangat besar. Jumlah putaran yang tidak aman untuk Serpent adalah 9 dari total 32 putaran, sedangkan Twofish adalah 6 dari total 16 putaran.

Secara umum, Serpent lebih konservatif dengan jumlah putarannya dari pada twofish. Twofish dalam hal ini lebih mengarah pada efisiensi *mixing conservatism*. Oleh karena itu lebih seimbang dalam hal performansi dan keamanan dalam berbagai macam implementasinya.

Dalam hal performansi, akan tergantung pada software atau hardware dimana algoritma diimplementasikan, karena adakalanya performansinya akan lebih cepat, ataupun lebih lambat dari algoritma cipher block yang lainnya. Pada umumnya perbandingan tersebut dilakukan dengan cara mengimplementasi algoritma-algoritma cipher block ke dalam suatu perangkat keras tertentu. Dari implementasi tersebut, diukur performansi implementasi algoritma-algoritma tersebut dengan mengacu pada kecepatan algoritma, kebutuhan (requirement) terhadap resource, dan sebagainya. Dari semua percobaan membandingkan algoritma-algoritma tersebut, hampir semuanya menghasilkan hasil yang berbeda dalam hal algoritma

mana yang terbaik performansinya. Contohnya untuk implementasi dalam smart card, Serpent menjadi algoritma yang paling efisien, namun untuk implementasi pada suatu processor tertentu, Serpent justru menjadi algoritma yang membutuhkan waktu paling lama dalam proses enkripsi dan dekripsi dibanding dengan twofish.

Perbedaan-perbedaan tersebut menunjukkan bahwa masing-masing algoritma memiliki kelebihan dan kekurangan masing-masing. Namun di samping itu, perbedaan ini mungkin disebabkan karena memang ada beberapa percobaan yang dilakukan oleh desainernya sendiri, jadi memungkinkan adanya bias dalam melakukan percobaan.

V. KESIMPULAN

Algoritma kriptografi kunci simetris menggunakan kunci yang sama dalam proses enkripsi maupun dekripsinya. Block cipher merupakan algoritma yang berbasis kunci simetris dengan melakukan enkripsi/dekripsi pada tiap block-block data plain/cipher teks, dimana blok merupakan sejumlah bit dengan ukuran tetap yang terdapat pada data tersebut.

Teknologi kriptografi semakin pesat sehingga kebutuhan akan keamanan data sangat tinggi. Hal inilah yang menyebabkan dilakukan peningkatan standar dari DES ke AES.

Diantara algoritma-algoritma yang masuk menjadi finalis AES adalah Serpent dan Twofish. Masing-masing memiliki perbedaan serta kelebihan dan kekurangan tertentu.

Algoritma serpent menggunakan struktur berbasis jaringan substitusi-permutasi sepertihalnya algoritma Rijndael yang merupakan AEA yang terpilih. Sedangkan Twofish menggunakan struktur jaringan Feistle.

Dalam segi keamanan, secara umum, Serpent lebih konservatif dengan jumlah putarannya dari pada twofish. Twofish dalam hal ini lebih mengarah pada efisiensi *mixing conservatism*. Oleh karena itu lebih seimbang dalam hal performansi dan keamanan dalam berbagai macam implementasinya.

Dalam segi performansi, sebenarnya tidak bisa diputuskan algoritma mana yang lebih baik, karena untuk performansi sendiri bergantung pada hardware atau software dimana algoritma ini diimplementasikan.

DAFTAR PUSTAKA

[1] http://en.wikipedia.org/wiki/Serpent_(cipher)
[2] http://id.wikipedia.org/wiki/Twofish
[3] Schneier, Bruce, et al. 1998. Twofish: A 128-bit Block Cipher.
[4] http://www.schneier.com/twofish.html
[5] http://www.cl.cam.ac.uk/~rja14/serpent.html
[6] Kohno, Tadayoshi et al. Preliminary Cryptanalisis of Reduced-Round Serpent.
[7] Anderson, Ross, et al. Serpent: A Proposal for the Advanced Encryption, Standard. Cambridge University.

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, 23 Maret 2011

ttd

Moch. Yusup Soleh / 13507051