Analisis Sandi Diferensial terhadap AES, DES dan AE1

Yusuf Kurniawan[1], Adang Suwandi Ahmad[2], M. Sukrisno Mardiyanto[3], Iping Supriana[3]& Sarwono Sutikno[2]

[1]Jurusan Teknik Informatika Universitas Pasundan Bandung
[2]Program Studi Teknik Elektro Institut Teknologi Bandung
[3]Program Studi Teknik Informatika Institut Teknologi Bandung

Abstrak. Sejak tahun 1977, DES resmi menjadi algoritma enkripsi Amerika. Kemudian, hampir seluruh dunia menggunakan algoritma ini untuk keperluan keuangan, perbankan dan perekonomian. Karena dianggap terlalu pendek kuncinya, pada tahun 2001 DES diganti AES. Salah satu masalah paling penting dalam kriptologi adalah pembuktian keamanan algoritmanya terhadap analisis sandi. Pada makalah ini akan dibahas analisis sandi diferensial (ASD) untuk memeriksa keamanan DES, AES dan AE1, algoritma yang baru kami buat. Meskipun ASD telah diusulkan sejak tahun 1990, namun para pakar kriptografi di seluruh dunia tetap menggunakannya hingga kini untuk memeriksa keamanan algoritma enkripsi. Kontribusi kami di sini adalah memberikan pembuktian bahwa algoritma kami dapat bertahan menghadapi ASD dan bahwa AE1 memiliki ketahanan yang lebih baik dari pada DES dan AES, namun AE1 memiliki kelemahan, yaitu lebih lambat dibandingkan dengan DES ataupun AES.

Kata kunci: *AES; DES; AE1; Analisis Sandi Diferensial*

Abstract. Since 1977, DES officially became America encryption algorithm. And then this algorithm spreads over the world to be used in banking, financial industries, and economics. Because of the short of the key, in 2001, DES is replaced by AES. One of the most important problem in cryptology is how to prove the security of algorithm against cryptanalysis. In this paper, we will discuss differential attack to examine the security of DES, AES and AE1. AE1 is block cipher that we proposed. Although differential attack has been proposed since 1990, all of cryptographers in the world still use it to examine the strength of ciphers till now. Our contribution is providing a proof that AE1 can be resistant to differential attack and that our cipher is more secure than DES and AES against differential attack. Unfortunately, our cipher is slower than DES or AES.

Keywords: *AES; DES; AE1; Differential cryptanalysis.*

1 Pendahuluan

Kriptologi merupakan ilmu gabungan antara kriptografi dan analisis sandi. Kriptografi berusaha mengamankan data agar hanya yang dituju yang dapat

Makalah diterima redaksi tanggal 1 Desember 2005, revisi diterima tanggal 22 Maret 2006.

membaca pesan kita (enkripsi). Kriptografi juga dapat memberi keyakinan bahwa pengirimnya merupakan orang yang sudah kita percaya (otentikasi) dan berfungsi juga agar pengirim tidak dapat mengingkari telah mengirim pesan (non repudiation). Sebaliknya, analisis sandi berusaha memecahkan algoritma kriptografi. Tanpa analisis sandi, sangat sulit bagi kita untuk mengukur kekuatan algoritma kriptografi.

Kebanyakan peneliti di tanah air hanya menitikberatkan penelitiannya pada bagaimana membuat algoritma kriptografi yang dianggap aman, tanpa sedikitpun memberikan bukti bahwa algoritmanya aman, setidaknya terhadap serangan (analisis sandi) tertentu. Seringkali algoritma dianggap aman hanya berdasar kompleksitasnya. Oleh karena itu, kami mencoba melakukan penelitian bagaimana membuktikan keamanan algoritma enkripsi, khususnya terhadap analisis sandi diferensial (ASD). ASD sendiri dapat diterapkan untuk memeriksa keamanan *stream cipher, block cipher,* dan fungsi *hash*.

Hal menarik yang juga perlu mendapat perhatian adalah apakah setiap modifikasi terhadap algoritma yang ada, akan selalu memberi peningkatan keamanan atau tidak. Dan seberapa besarkah perubahan tingkat keamanannya? Dalam 1, seorang pakar kriptologi, Schneier, menyatakan ”*It's amazing how a few subtle changes can make such a big difference*” ketika mengomentari perubahan IPES menjadi IDEA (Salah satu *block cipher* yang paling terkenal di seluruh dunia). Sedikitnya modifikasi IPES menjadi IDEA, yang menyebabkan peningkatan keamanan yang sangat besar membuat kagum Schneier. Demikian pula, dalam 2, penulis memberi contoh beberapa modifikasi DES yang ternyata justru meningkatkan kelemahan DES. Contoh lain adalah algoritma peserta NESSIE 3, Grand Cru. Algoritma ini didesain untuk meningkatkan keamanan AES dengan memodifikasi beberapa bagian AES yang konstan menjadi bergantung kunci. Namun modifikasi ini justru memperlemah AES, sehingga Grand Cru gugur di tahap awal seleksi. Hal ini menunjukkan bahwa modifikasi suatu algoritma tidak bisa dipandang sebelah mata. Karena hanya dengan sedikit modifikasi, tingkat keamanan sebuah algoritma bisa berubah drastis. Apalagi jika dilakukan banyak modifikasi.

Pada penelitian ini kami membuat algoritma enkripsi yang kami beri nama AE1 (Algoritma Enkripsi 1) yang didasarkan pada AES. Namun, terdapat beberapa perbedaan dari AES. Perbedaan itu adalah pada ukuran matrik MDS (*Maximum Distance Separable*), di mana AE1 menggunakan matrik 16x16, sedangkan AES menggunakan matrik 4x4. Kami juga menggunakan fungsi FN yang tidak terdapat pada AES. Fungsi FN ini berfungsi untuk mempertahankan AE1 dari serangan analisis sandi *linear hull* dan *impossible differential attack*. Perbedaan lainnya dengan AES adalah pada ekspansi kunci. Pada ekspansi kunci, kami menggunakan struktur yang sama seperti pada bagian pengacakan. AES

menggunakan struktur yang berbeda. Penggunaan struktur yang sama ini dimaksudkan agar lebih efisien jika diimplementasikan pada perangkat keras. Penggunaan ulang struktur pengacakan untuk ekspansi kunci juga bertujuan agar setiap bit kunci utama memiliki pengaruh yang sama ke setiap ronde, karena memiliki difusi yang sangat besar. Sehingga ekspansi kunci pada AE1 memiliki difusi yang jauh lebih cepat daripada AES. Hal ini bertujuan untuk menggagalkan serangan *related key attack*. Akibat lanjutnya adalah, setiap perubahan satu bit kunci utama akan mengakibatkan perubahan sejumlah besar bit *ciphertext*. Dalam makalah ini, penelitian ditekankan pada bagaimana cara mengukur kekuatan AE1 terhadap analisis sandi diferensial, dan perbandingannya dengan DES serta AES.

Pembahasan selanjutnya adalah sebagai berikut. Pada bab 2 akan dibahas cara kerja AES, bab 3 akan membahas dasar-dasar analisis sandi diferensial , bab 4 membahas analisis sandi diferensial (ASD) terhadap AES, bab 5 membahas ASD terhadap DES, bab 6 membahas ASD terhadap AE1, dan sebagai penutup adalah bab 7 yang berisi penutup.

2 AES (Advanced Encryption Standard)

Pada tahun 2001, Algoritma Rijndael, karya peneliti dari universitas di Belgia ditetapkan menjadi AES. Rijndael merupakan algoritma yang dapat menerima masukan data 128 bit dan menghasilkan data 128 bit pula. Bila digunakan dengan kunci 128 bit, maka kita menyebutnya sebagai AES-128. Selain 128 bit, AES juga dapat menerima kunci 192 dan 256 bit. Dalam tulisan ini akan dibahas AES-128. Plaintext diletakkan pada matrik segiempat yang disebut *state* berukuran 4x4 (setiap sel berisi 1 byte) untuk AES-128 (Gambar 1).

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$

Gambar 1 Contoh State AES.

Kemudian dilakukan operasi pada setiap ronde yang dalam notasi pseudo C sebagai berikut :

```
AddRoundKey(State);
Ronde(State,KunciRonde)
{
  ByteSub(State);
```

```
  ShiftRow(State);
  MixColumn(State);
  AddRoundKey(State);
}
```

Pada bagian akhir terdapat sedikit perbedaan operasi agar struktur dekripsinya menyerupai struktur enkripsinya. Operasi Mixcolumn ditiadakan pada ronde terakhir.

```
Ronde(State,KunciRonde)
{
  ByteSub(State);
  ShiftRow(State);
  AddRoundKey(State);
}
```

Kotak subtitusi (untuk ByteSub) dibentuk dari fungsi inversi perkalian pada GF(2^8) dan kemudian ditambahkan operasi XOR terhadap fungsi *affine* (semacam fungsi linear) yang didefinisikan sebagai berikut:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

di mana x adalah keluaran fungsi inversi dan y menjadi keluaran kotak subtitusi. Atau dengan kata lain, bila masukan kotak subtitusi adalah w dan keluarannya y, maka di dalam kotak subtitusi ini, w ini akan diinversi menjadi $w^{-1} = x$ pada GF(2^8), dan kemudian x ini akan dimasukkan ke dalam persamaan affine di atas untuk mendapatkan y.

Operasi ShiftRow menggeser baris ke-1 ke kiri 1 byte, baris ke-2 ke kiri 2 byte, dan baris ke-3 ke kiri 3 byte. Baris ke-0 tidak digeser. Kemudian operasi Mixcolumn merupakan operasi perkalian satu kolom dengan polinomial c(x) mod (x^4+1) di mana c(x) = '03'x^3 + '01'x^2 + '01'x + '02'. Persamaan ini dapat dituliskan juga dalam bentuk matrik sebagai berikut:

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

di mana a merupakan masukan dan b keluaran Mixcolumn. Sedangkan operasi AddRoundKey merupakan operasi sederhana berupa pengXORan data dengan key.

Penjelasan lebih lanjut mengenai Rijndael dapat dilihat pada 4.

3 Analisis Sandi Diferensial

Analisis sandi diferensial (ASD) berusaha mendapatkan beda masukan yang menghasilkan beda keluaran dengan peluang sebesar mungkin.

Definisi 1. *Untuk sebarang masukan / keluaran x,y ,dan beda masukan / keluaran Δx, $\Delta y \in GF(2^m)$, peluang diferensial kotak subtitusi : $GF(2^m) \rightarrow GF(2^m)$ didefinisikan sebagai berikut :*

$$P_{dif}[s(x) \oplus s(x \oplus \delta)=\Delta] = \frac{\#\{x \in GF(2^m) \mid s(x) \oplus s(x \oplus \delta)=\Delta\}}{2^m} \quad (1)$$

Bagian utama kebanyakan algoritma enkripsi adalah perulangan ronde. Struktur T ronde dapat kita nyatakan sebagai berikut :

$$E_T = \chi[k_T] \, o \, \chi[k_{T-1}] \ldots \chi[k_1] \quad (2)$$

$$r_i = \chi[k_i](r_{i-1}), \; i=1,\ldots,T \quad (3)$$

di mana masukan adalah $r = r_0$ dan keluaran = r_T.

Definisi 2.(MDP). 5 *Bila vektor masukan x, vektor keluaran y, dan kunci adalah k, maka Maximum Differential Probability (MDP) untuk enkripsi E_T sebanyak T ronde didefinisikan sebagai berikut:*

$$MDP(E_T) \equiv \underset{\Delta x \neq 0, \Delta y, k}{maks} \sum_{\Delta x_1, \Delta x_2, \ldots \Delta x_{T-1}} \prod_{i=1}^{T} DP^{\rho_i[k_i]}(\Delta x_{i-1} \rightarrow \Delta x_i) \quad (4)$$

Jika MDP cukup kecil, maka dijamin tidak akan ada kunci yang lemah menghadapi ASD. Karena MDP sulit dihitung, maka digunakanlah cara perhitungan lain yaitu dihitung rata-rata MDP nya, yang disebut sebagai MADP (Maximum Average Differential Probability). Kesulitan perhitungan MDP diakibatkan karena banyaknya lintasan diferensial yang mungkin terjadi.

Definisi 3 (MADP). *Maximum Average Differential Probability (MADP) didefinisikan sebagai berikut:*

$$MADP(E_T) \equiv \underset{\Delta x \neq 0, \Delta y}{maks} \; \underset{k}{rata^2} \sum_{\Delta x_1, \Delta x_2, \ldots \Delta x_{T-1}} \prod_{i=1}^{T} DP^{\rho_i[k_i]}(\Delta x_{i-1} \rightarrow \Delta x_i) \quad (5)$$

Tetapi ternyata, perhitungan MADP juga masih sulit karena terdapat penjumlahan peluang pada bagian tengah *cipher*. Karena itu maka digunakan metode perhitungan lain yaitu menggunakan pendekatan jalur tunggal MDCP (*Maximum Differential Characteristic Probability*). Metode ini bertujuan mendapatkan sebuah lintasan diferensial dari awal (plaintext) hingga satu atau dua ronde sebelum ciphertext yang memiliki peluang sebesar mungkin.

Definisi 4 (MDCP). *Maximum Differential Characteristic Probability didefinisikan sebagai berikut:*

$$MDCP(E_T) \equiv \underset{\Delta x \neq 0, \Delta y, k}{maks} \prod_{i=1}^{T} DP^{\rho_i[k_i]}(\Delta x_{i-1} \rightarrow \Delta x_i) \tag{6}$$

Contoh fungsi yang memiliki sifat yang diinginkan, diberikan oleh 6 yaitu fungsi inversi perkalian yang juga digunakan dalam AES.

Proposisi 1. *Bila $F(x) = x^{-1}$, $x \neq 0$ dan $F(0) = 0$, maka $F(x)$ adalah uniform-4 secara diferensial 6, atau*

$$\#\{x \in GF(2^n) | F(x+\Delta)-F(x)=\beta\} \leq \delta=4 \tag{7}$$

Bukti: Misalkan masukan kotak subtitusi adalah x dan $(x+\Delta)$ serta menghasilkan y dan $(y+\beta)$, maka beda masukan Δ akan menghasilkan beda keluaran β dengan peluang tertentu.

$$(x+\Delta)^{-1} + x^{-1} = \beta \tag{8}$$

Dengan mengalikan kedua sisi pers (8) dengan x(x+Δ) akan diperoleh :

$$x + (x+\Delta) = \beta x(x+\Delta) = \beta x^2 + \beta \Delta x \tag{9}$$

$$\beta x^2 + \beta \Delta x + \Delta = 0 \tag{10}$$

Persamaan (10) ini akan memiliki solusi paling banyak dua buah bila $x \neq 0$ dan $x \neq \Delta$. Dan bila $x = 0$ serta $x = \Delta$ dianggap solusi, maka persamaan ini akan memiliki paling banyak 4 (= δ) buah solusi. Untuk memperoleh penjelasan lebih detil mengenai ASD, silakan membaca 2 dan 7 .

4 Analisis Sandi Diferensial pada AES

Untuk membuktikan apakah AES kebal terhadap ASD, kita harus dapat memastikan jumlah kotak subtitusi yang aktif setelah beberapa ronde. Telah diketahui bahwa jumlah minimal kotak subtitusi yang aktif pada jalur diferensial adalah 25 buah untuk 4 ronde. Karena itu peluang diferensial maksimal 4 ronde = $(2^{-6})^{25} = 2^{-150}$. Jadi, diperlukan 2^{150} pasang plaintext, sedangkan diketahui bahwa hanya terdapat maksimal 2^{128} plaintext yang

mungkin, sehingga karakteristik AES 4R akan dapat menghadang serangan terhadap AES 7 ronde (3R *attack*).

Misalkan kita memasukkan karakteristik plaintext $a'_{0,0}$ =1 dan $a'_{i,j}$ lain =0 untuk i,j=0...3 (perhatikan lagi gambar 1 dengan asumsi bahwa $a_{i,j}$ menunjukkan posisi pada baris i kolom j) dengan peluang maksimal 2^{-6}. Maka pada keluaran ByteSub (BS) ronde pertama, karekteristik akan tetap, demikian juga setelah ShiftRow(SR) di ronde pertama. Namun, Mixcolumn (MC) pada ronde pertama menyebabkan diferensial pada posisi $a_{0,0}$ menyebar ke $a_{i,0}$. Operasi AddRoundKey(AK) tidak mengubah kondisi, karena operasi XOR tidak mempengaruhi diferensial. Perhatikan bahwa bila kita memiliki 2 masukan x_1 dan x_2 maka $x_1 \oplus k = y_1$ dan $x_2 \oplus k = y_2$. Kemudian $\Delta x \oplus k \oplus k = x_1 \oplus x_2$, jadi operator XOR tidak mempengaruhi diferensial. Sementara itu, diferensial pada byte-byte yang lain tetap nol, karena tidak ada operasi yang mengubahnya

Akibatnya, masukan pada ronde kedua menjadi $a'_{i,0}$ = 1 untuk i=0..3 dan $a'_{i,j}$ = 0 untuk i=0..3 dan j=1..3. Bila kita telusuri lebih lanjut, maka SB pada ronde ke-2 tidak mempengaruhi diferensial, sedangkan SR menyebarkan posisi diferensial "tidak nol" ke seluruh kolom. Akibatnya, pada keluaran MC, seluruh byte pada *state* memiliki diferensial = 1. Sehingga dalam 2 ronde, terdapat 5 buah kotak subtitusi yang aktif, dan pada ronde ketiga terdapat 16 buah kotak subtitusi yang aktif. Dan bila kita berhasil mendapatkan jalur diferensial ke ronde keempat, maka karena pada setiap kolom di ronde ketiga terdapat 4 byte aktif, maka pada ronde keempat paling sedikit akan terdapat 4 kotak subtitusi yang aktif. Akibatnya, dalam 4 ronde, akan terdapat minimal 25 kotak subtitusi yang aktif, sama seperti penjelasan di atas.

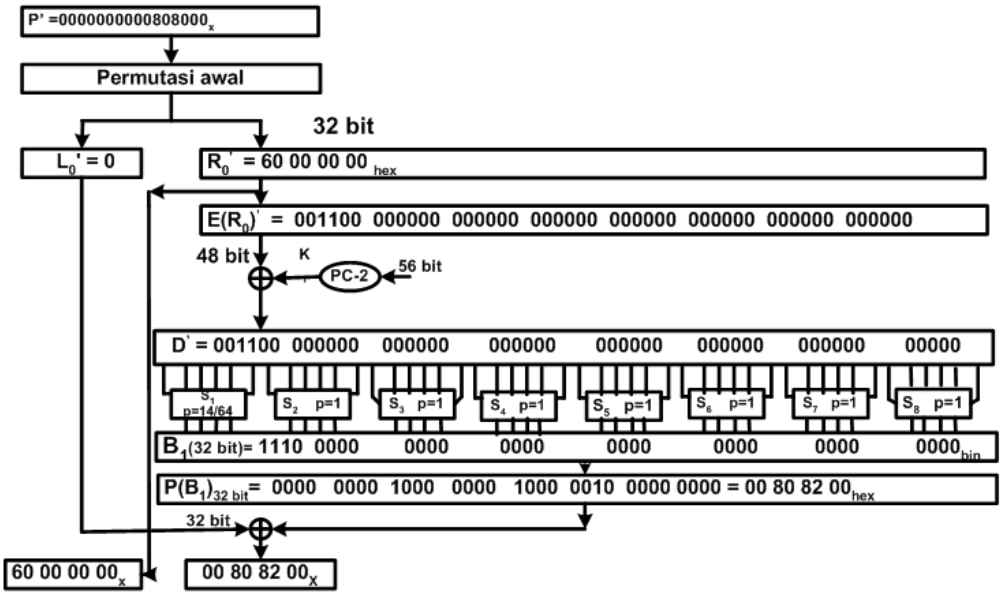
5 ASD terhadap DES

DES merupakan algoritma enkripsi yang memiliki struktur Feistel sehingga struktur enkripsi dan dekripsinya sama. DES lengkap terdiri dari 16 ronde serta memiliki masukan dan keluaran 64 bit. Kunci DES hanya 56 bit sehingga dianggap tidak aman lagi. Namun DES masih digunakan hingga kini karena masalah kompatibilitas dengan perangkat lama, dan DES yang sering digunakan adalah adalah triple DES, yang menggunakan 3 kali enkripsi DES. Deskripsi lengkap tentang cara kerja DES dapat dilihat pada 8.

Karena sedemikian kompleknya mendapatkan karakteristik yang diharapkan pada DES, maka digunakanlah karakteristik iteratif. Karakteristik iteratif ini tidak menjamin bahwa nilai yang diperoleh adalah nilai peluang diferensial yang tertinggi, namun hanya untuk mempermudah saja, ditambah kenyataan,

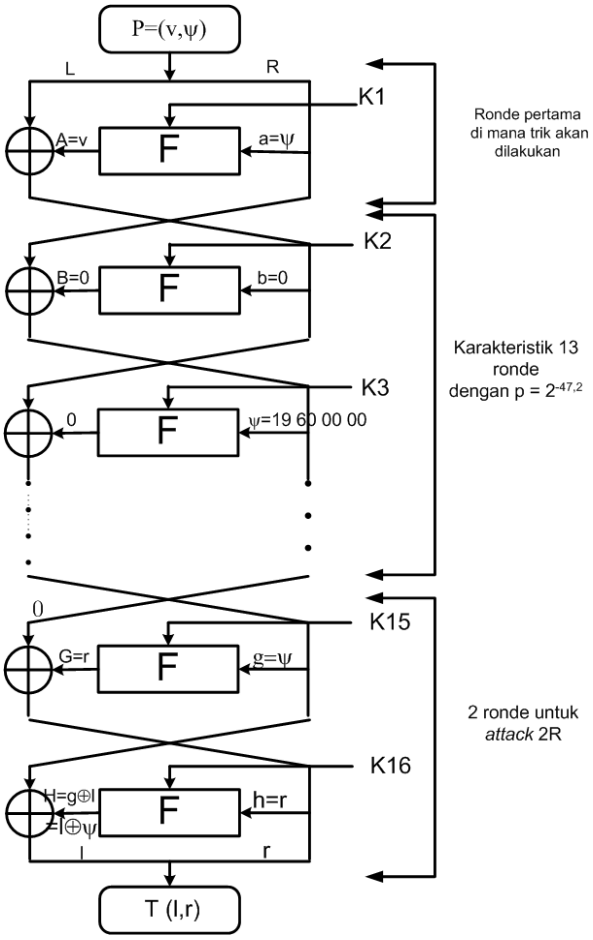
setelah 14 tahun diusulkan, belum ditemukan karakteristik DES yang lebih baik dari pada yang ditemukan Biham dan Shamir.

Untuk mendapatkan karakteristik iteratif yang diperlukan, mula-mula kita periksa peluang diferensial seluruh kotak subtitusi yang ada. Sebagai contoh, perhatikan gambar 2. Gambar 2 memperlihatkan bahwa bila selisih masukan adalah 00 00 00 00 00 80 80 00_{hex} (64 bit), maka setelah melalui 1 ronde DES, akan diperoleh beda = 60 00 00 00 00 80 82 00_{hex} dengan peluang 14/16 x 1^7 = 14/16. Kerena proses permutasi diketahui (karena algoritma dibuka ke umum), maka permutasi tidak meningkatkan keamanan. Keamanan hanya bergantung pada kotak subtitusi. Karakteristik 1 ronde ini diperluas hingga mencapai DES 16 ronde lengkap seperti diperlihatkan pada Gambar 3.



Gambar 2 ASD pada DES 1 ronde.

Jadi, DES 16 ronde dapat dipecahkan menggunakan karakteristik seperti pada gambar 3 dengan 2^{47} pasang plaintext yang dipilih. Meskipun hal ini sangat tidak praktis, karena tentu saja lebih mudah melakukan *brute force attack* menggunakan satu atau dua pasang plaintext/ciphertext yang diketahui dengan mencoba rata-rata 2^{55} enkripsi, namun ASD dianggap sebagai metoda pertama yang berhasil menaklukkan DES. Untuk mendapatkan cara kerja ASD terhadap DES secara lebih detail, silakan membaca 2 dan 9.



Gambar 3 ASD pada DES 16 ronde.

6 ASD pada AE1

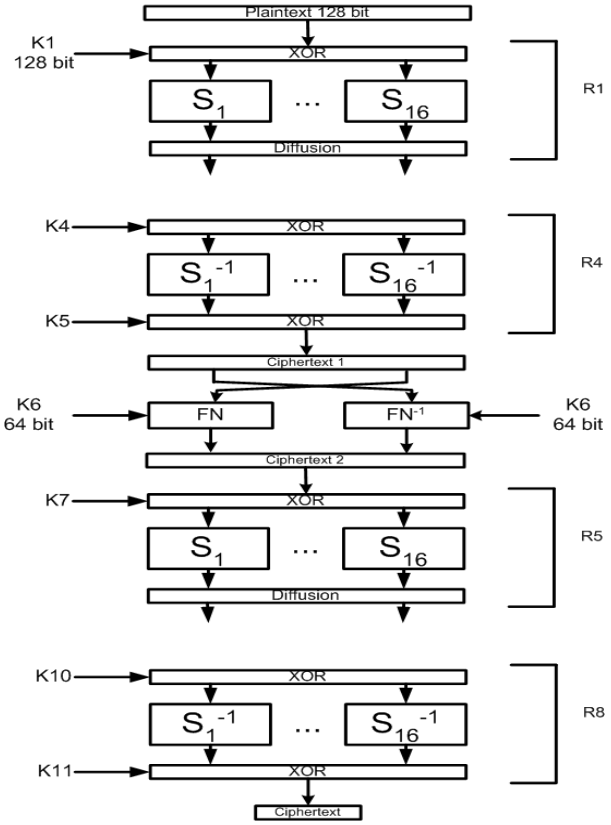
AE1 (Algoritma Enkripsi 1) adalah algoritma enkripsi yang kami rancang berdasar algoritma keluarga AES. Algoritma AE1 memiliki struktur yang serupa dengan AES, yaitu SPN. Namun AE1 dirancang agar memiliki struktur yang *involutional*, yaitu struktur enkripsi dan dekripsinya sama, tidak seperti AES. Struktur AE1 menyerupai *block cipher* ARIA asal korea 10. Keistimewaan yang diharapkan dari AE1 adalah kebal terhadap berbagai macam analisis sandi seperti ASD, ASL, Square Attack, Slide Attack, related key attack, Boomerang attack, interpolation attack dan impossible attack.

Namun AE1 juga memiliki kekurangan, yaitu kurang cepat dalam implementasi di perangkat lunak bila dibandingkan AES. Hal ini disebabkan karena AE1 menggunakan matrik MDS (Maximum Distance Separable) yang berukuran besar yaitu 16 x 16. MDS ini menjadi salah satu faktor terpenting keamanan AE1, namun juga memberikan pengaruh melambatnya algoritma.

6.1 Deskripsi AE1

AE1 merupakan algoritma yang memiliki masukan keluaran 128 bit, dan kunci 128 bit. AE1 memiliki 6 ronde regular (ronde 1,2,3,5,6,7) dan ronde khusus (ronde 4,8 dan fungsi FN). Setiap ronde regular AE1 dapat dinyatakan sebagai $\tau=\psi\circ\sigma\circ\kappa\circ x$ di mana x adalah masukan 128 bit yang diXORkan dengan subkey κ dan hasilnya dimasukkan ke dalam 16 buah kotak subtitusi σ yang sama dan kemudian dilalukan ke lapis difusi, yaitu matrik MDS ψ yang berukuran 16x16. Kotak subtitusi pada ronde genap merupakan inversi dari kotak subtitusi pada ronde ganjil serta MDS yang merupakan matrik *involutional* digunakan untuk membuat sistem menjadi *involutional* secara total. Struktur AE1 dapat dilihat pada gambar 4.

MDS AE1 diilhami *cipher* Khazad 11 yang menggunakan matrik Hadamard untuk mempermudah mendapatkan matrik yang memenuhi kriteria MDS dan sekaligus *involution*. Meskipun Khazad juga *involution*, AE1 tidak mengikuti struktur Khazad karena kuatir sifat kotak subtitusi *involution* yang dimiliki Khazad 12. Matrik Hadamard memiliki sifat $m_{i,j} = d_{i\oplus j}$ di mana m adalah matrik berukuran n x n dan d adalah vektor kode MDS. AE1 menggunakan d = {14, 1, 4, 5, 8, 11, 16, 11, 17, 13, 10, 12, 15, 3, 6, 7}. Matrik ini kami peroleh secara semi acak berdasarkan lemma 1.



Gambar 4 Algoritma AE1.

Lemma 1. *Sebuah kode (n,k,d) yang dibentuk dari generator G=[I A], di mana A adalah matrik k x (n-k), adalah MDS, jika dan hanya jika setiap submatrik bujursangkar A (yang tersusun dari sebarang baris i dan kolom i untuk i = 1,2,...min{k,n-k}) merupakan matrik nonsingular.*

Definisi 5. *Jumlah cabang diferensial dari transformasi* $\lambda : \{0,1\}^n \rightarrow \{0,1\}^n$ *adalah :*

$$B_d(\lambda) = \min_{a \neq b}\{w_b(a \oplus b) + w_b(\lambda(a) \oplus \lambda(b))\} \quad (11)$$

di mana w_b merupakan jumlah kotak subtitusi yang aktif.

Jumlah cabang memiliki sifat:

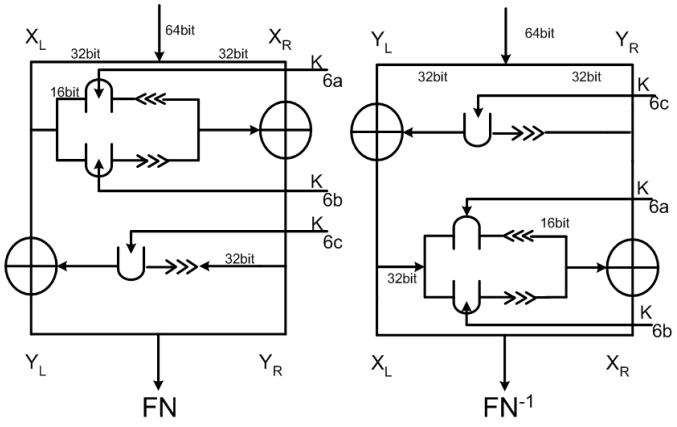
a. Jumlah cabang dari suatu permutasi sama dengan inversinya
b. Jumlah cabang tidak dipengaruhi oleh penambahan kunci.

Beberapa metode untuk mendapatkan kode MDS dapat dilihat pada 13.

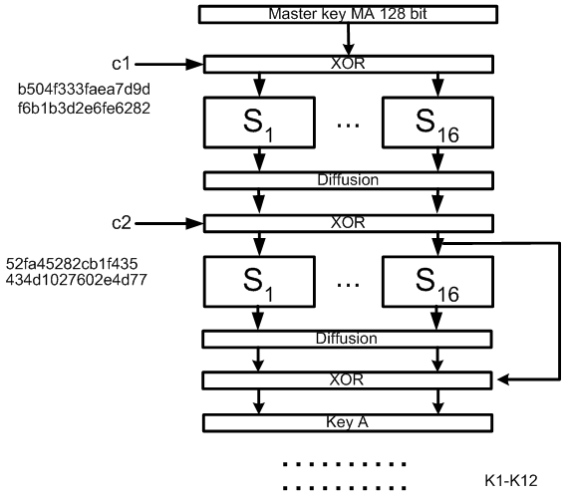
Untuk membuat *involutional*, diperlukan pengolahan subkey pada dekripsi sebagai berikut. Kunci pertama pada dekripsi adalah kunci terakhir pada enkripsi atau K_1 dekripsi (K_{1d}) = K_{11} enkripsi (K_{11e}). Kemudian $K_{2d} = \psi^{\circ}K_{10e}$, $K_{3d} = \psi^{\circ}K_{9e}$, $K_{4d} = \psi^{\circ}K_{8e}$, $K_{5d} = \psi^{\circ}K_{7e}$, $K_{6d} = \psi^{\circ}K_{6e}$, $K_{7d} = \psi^{\circ}K_{5e}$, $K_{8d} = \psi^{\circ}K_{4e}$, $K_{9d} = \psi^{\circ}K_{3e}$, $K_{10d} = \psi^{\circ}K_{2e}$, $K_{11d} = \psi^{\circ}K_{1e}$. Jadi perbedaan antara subkey enkripsi dengan dekripsi terletak pada urutan dan proses difusi beberapa subkey dekripsi.

Fungsi FN (gambar 5) dibuat berdasar fungsi FL pada Camellia 14. Struktur FN dan inversi FN dibuat sedemikian rupa agar cipher keseluruhan tetap involutional. Lambang U merupakan OR, ∩ adalah AND, simbol <<< adalah rotasi ke kiri 1 bit, dan >>> menyatakan rotasi 1 bit ke kanan.

Ekspansi kunci diperlihatkan pada gambar 6. Ekspansi kunci ini menggunakan struktur yang serupa dengan struktur enkripsi untuk penghematan ruang bila diimplementasikan pada perangkat keras. Konstanta c_1 (pecahan $2^{-0,5}$ yang dinyatakan dalam hexadesimal) dan c_2 ($5^{-0,7}$) digunakan sebagai kunci enkripsi dengan plaintext masterkey untuk menghasilkan kunci A. Dan dari kunci A inilah diturunkan subkey K_1 - K_{11}. Caranya adalah dengan merotasi ke kiri 11 bit kunci A untuk mendapatkan K_1. Dan ini diulang sampai K_{11}. Dengan struktur tersebut diharapkan bahwa apabila analis sandi berhasil mendapatkan subkey, K_{11}, misalnya, dia tetap tidak akan dapat menurunkan masterkey-nya. Berbeda dengan DES, karena sangat sederhananya proses ekspansi kunci, bila subkey berhasil diketahui, maka masterkey-nya akan mudah ditemukan. Dengan proses difusi dan *confusion* (yang diberikan oleh kotak subtitusi) yang kuat pada ekspansi kunci diharapkan bahwa *related-key attack* akan sangat dipersulit.



Gambar 5 Struktur FN.



Gambar 6 Ekspansi kunci AE1.

6.2 ASD terhadap AE1

Karena sejak awal, AE1 telah dirancang untuk dapat menahan ASD, maka AE1 dapat dibuktikan akan sanggup menghadapi ASD. Telah diketahui dari 15 bahwa ASD sangat dipengaruhi oleh jumlah kotak subtitusi yang aktif. Semakin besar jumlah kotak subtitusi yang aktif, semakin kecil pula peluang ASD akan berhasil. Pada kasus *block cipher* SPN sederhana, di mana digunakan permutasi bit untuk lapis difusinya, dengan mudah kita mendapatkan jalur diferensial dan linear yang menggunakan sesedikit mungkin kotak subtitusi, sehingga ASD dan ASL dengan mudah pula menembusnya.

Definisi 6. *Kotak subtitusi aktif diferensial adalah kotak subtitusi yang memiliki beda masukan tidak nol 5.*

Teorema 1. *Dalam 4 ronde pertama AE1,* $DP_{min} = 2^{204}$ *sehingga AE1 4 ronde tidak memiliki lintasan diferensial yang dapat digunakan oleh ASD.*

Bukti: Karena MDS yang digunakan memiliki jumlah cabang minimal = 17, maka jumlah kotak subtitusi yang aktif pada lintasan diferensial dalam dua ronde yang berturutan minimal adalah 17. Jika pada ronde pertama terdapat sebanyak k kotak-S aktif, maka pada ronde kedua akan terdapat minimal (17-k) kotak-S yang aktif. Kemudian karena jumlah cabang ronde 2 dan 3 juga 17, maka jumlah kotak subtitusi yang aktif di ronde 2 dan 3 adalah 17-k+k, atau jumlah kotak subtitusi aktif minimal pada ronde 3 sebanyak k, kemudian agar jumlah cabang pada ronde 3 dan 4 dipenuhi, maka jumlah minimal kotak subtitusi aktif pada ronde keempat haruslah sebanyak (17-k). Sehingga dalam 4

ronde, minimal terdapat 34 kotak-S aktif. Karena itu, untuk 4 ronde, Differential Probability $DP_{min}= (2^{-6})^{34} = 2^{204}$.

Dari teorema tersebut dapat disimpulkan bahwa bila lintasan 4 ronde AE1 dapat digunakan untuk memecahkan 6 ronde AE1 (2R- attack), maka diperlukan 2^{204} pasangan plaintext yang dipilih untuk memecahkan AE1 dengan ASD. Sedangkan diketahui bahwa hanya terdapat 2^{128} pasang plaintext yang mungkin ada. Artinya, AE1 terbukti mampu menghadapi ASD. Dan bila *differential hull* ikut diperhitungkan, maka fungsi FN akan menghancurkan lintasannya, itupun bila dalam 4 ronde memiliki peluang yang lebih besar dari pada 2^{-128}.

Fungsi FN diharapkan mampu menahan diferensial hull (bila ada) karena memiliki sifat sebagai berikut:

- Dengan kunci ronde $k_i=0$ dan operator AND, maka $\Delta x=0 \rightarrow \Delta y=0$ dan $\Delta x=1 \rightarrow \Delta y=0$. Artinya, k=0 akan memaksa Δ keluaran menjadi 0.
- Dengan kunci ronde $k_i=1$ dan operator AND, maka $\Delta x=0 \rightarrow \Delta y=0$ dan $\Delta x=1 \rightarrow \Delta y=1$. Artinya, k=1 tidak memberi pengaruh apapun, sehingga Δ keluaran = Δ masukan.
- Dengan kunci ronde $k_i=0$ dan operator OR, maka $\Delta x=0 \rightarrow \Delta y=0$ dan $\Delta x=1 \rightarrow \Delta y=1$. Artinya, k=0 tidak memberi pengaruh apapun, sehingga Δ keluaran = Δ masukan.
- Dengan kunci ronde $k_i=1$ dan operator OR, maka $\Delta x=0 \rightarrow \Delta y=0$ dan $\Delta x=1 \rightarrow \Delta y=0$. Artinya, k=1 akan memaksa Δ keluaran menjadi 1.

Dengan rotasi 1 bit pada FN, diharapkan keteraturan lintasan byte yang digunakan dalam analisis sandi akan pecah. Dari sifat-sifat komponen yang terdapat dalam FN, diharapkan, lintasan *diferensial hull* akan rusak. Dan karena FN berada di tengah-tengah cipher serta difusi yang sangat kuat pada ronde sebelumnya, maka hampir mustahil untuk mengatur masukan FN, sehingga diharapkan, berbagai analisis sandi yang lain pun akan berantakan.

7 Penutup

Dari pembahasan di atas, kita dapat menyampaikan hal-hal berikut:

- AES-128 memiliki ketahanan yang besar untuk menghadapi ASD, karena dalam 4 ronde memiliki $DP_{min}=2^{-150}$, sedangkan AES-128 memiliki 10 ronde. Bandingkan dengan DES lengkap yang dapat dipecahkan ASD dengan 2^{47} pasang plaintext yang dipilih.
- AE1 memiliki ketahanan yang lebih besar terhadap ASD dibandingkan DES dan AES untuk jumlah ronde yang sama (serangan 2 ronde terhadap AE1 6 ronde membutuhkan 2^{204} pasang plaintext, serangan terhadap DES 16 ronde membutuhkan 2^{47} pasang plaintext, dan

serangan terhadap AES 6 ronde membutuhkan 2^{150} pasang plaintext yang dipilih), namun memiliki masalah dalam hal kecepatan eksekusi. Sedangkan untuk ronde lengkap, AE1 sedikit lebih baik daripada AES, di mana untuk serangan 2R, AE1 8 ronde membutuhkan 2^{306} plaintext sedangkan AES 10 ronde membutuhkan 2^{300} plaintext, sementara DES tetap hanya membutuhkan 2^{47} plaintext.

- Kelambatan AE1 diakibatkan penggunaan matrik MDS 16x16 sehingga lebih lambat dioperasikan pada prosesor 8 bit hingga 32 bit, bila dibandingkan AES. Namun, pada prosesor 64 bit, kami perkirakan AE1 dapat menyetarai AES. Karena operasi MDS pada AE1 dapat dilakukan 2 kali lebih cepat dari pada di prosesor 32 bit. Sedangkan AES tidak mendapatkan keuntungan tersebut karena operasi MDS pada AES selalu per 32 bit. Dan jika di waktu yang akan terdapat prosesor 128 bit, AE1 akan lebih unggul performansinya dibanding AES, karena operasi MDS pada setiap ronde AE1 dapat dilakukan dengan sekali operasi, sementara pada AES harus dilakukan dengan empat kali operasi.
- Dengan Fungsi FN yang hanya menggunakan kode dasar komputer, diharapkan cipher akan cepat. Dan dengan sifat-sifat yang dimiliki FN, diharapkan, berbagai analisis sandi akan sulit menembusnya.
- Pembuktian keamanan algoritma kriptografi lebih sulit dari pada pembuatan algoritmanya, sehingga penulis menyarankan agar penelitian mengenai analisis sandi ditingkatkan.

Daftar Pustaka

1. Schneier, B., *Applied Cryptography*, 2nd edition, John Wiley & Sons, Inc., pp. 323 (1996).
2. Biham, E. & Shamir, A., *Differential Cryptanalysis of DES-like Cryptosystems*, In Advances in Cryptology: Journal of Cryptology, Vol. 4, No. 1, pp. 3-72 (1991).
3. NESSIE project, *New European Schemes for Signatures, Integrity and Encryption*, http://www.cryptonessie.org (2000).
4. Daemen, J. & Rijmen, V., *AES Proposal: Rijndael*, AES submission, http://www.nist.gov/aes (1999).
5. Ohkuma, K., Shimizu, H., Sano, F. & Kawamura, S., *Security Assessment of Hierocrypt and Rijndael against the Differential and Linear Cryptanalysis*, In Proceedings of the 2nd NESSIE workshop (2001).
6. Kaisa Nyberg, *Differentially uniform mapping for Cryptography*, Advances in Cryptology, Proc. Eurocrypt'93, LNCS 765 T. Helleseth, Ed., Springer- Verlag, pp. 439-444 (1994).
7. Yusuf Kurniawan, M. Sukrisno Mardiyanto, dan Iping Supriana S. *Kriteria Keamanan Blok Cipher dan Analisis Sandi Diferensial*, Teknoin, Jurnal Teknologi Industri, FTI UII, vol 10 (Maret 2005).

8. Man Young Rhee, *Cryptography and Secure Communications*, textbook, Mc Graw-Hill (1994).
9. Yusuf Kurniawan, M. Sukrisno Mardiyanto, dan Iping Supriana S. Analisis Sandi Diferensial Terhadap Full Data Encryption Standard. *Proceeding Seminar Nasional SNATI 2005*, Teknik Informatika UII Yogyakarta, (April 2005).
10. Kwon, D., Kim, J., Park, S., Sung, S., Sohn, Y., Song, J., Yeom, Yoon, Y., Lee, S., Lee, J., Chee, S., Han, D. & Jin Hong, *New Block Cipher ARIA*, Korean National Security Research Institute, Specification of ARIA (2003).
11. Barreto, P. S. L. M. & Rijmen, V., *The Khazad legacy-level block cipher*, Primitive submitted to NESSIE (Sept. 2000).
12. Biryukov, A., *Analysis of involutional ciphers: Khazad and Anubis*, in Proceedings of Fast Software Encryption (FSE' 03), T.Johansson,ed., Lecture Notes in Computer Science, Springer-Verlag (2003).
13. A. M. Youssef, S. & Tavares, S. E., On the design of linear transformations for substitution permutation encryption networks, *Workshop on Selected Areas of Cryptography, SAC'97*, Workshop record, pp. 40–48 (1997).
14. K. Aoki et al, Camellia: *A 128 Bit Block Cipher Suitable for Multiple Platforms*, NTT and Mitsubishi Electric Corporation, Primitive submitted to NESSIE Sept. (2000).
15. Yusuf Kurniawan, *Analisis Sandi Diferensial pada Jaringan Subtitusi Permutasi 7 Ronde*, Proceeding Seminar Nasional Universitas Widyatama, Bandung (2004).