ANALISIS PERBAIKAN KEAMANAN ALGORITMA ENKRIPSI AES SEBAGAI STANDAR ENKRIPSI BARU DIBANDINGKAN DES

Raditya Arief - 13507030

Program Studi Teknik Informatika Institut Teknologi Bandung
Jalan Ganesha 10 Bandung
e-mail: if17030@students.if.itb.ac.id

ABSTRAK

DES (*Data Encryption Standard*) merupakan standar enkripsi yang digunakan oleh NIST (*National Institute of Standards and Technology*) sejak tahun 1983. DES merupakan sebuah standar algortima, sedangkan algoritmanya sendiri adalah DEA (*Data Encryption Algorithm*). Secara umum DES termasuk ke dalam kriptografi kunci simetri dan tergolong *cipher* blok. Ukuran blok yang digunakan pada DES berkisar pada 64 bit. DES sudah mengalami empat kali revisi sejak pembuatannya demi meningkatkan keamanannya. Namun setelah beberapa kali revisi, NIST menganggap DES sudah tidak aman dan harus diganti dengan algoritma enkripsi baru yang lebih aman. Pada 26 Mei 2002 akhirnya secara resmi standar DES digantikan dengan AES (*Advance Encryption Standard*). Algoritma yang digunakan pada AES ini yaitu algoritma *Rijndael*. AES secara umum termasuk ke dalam algoritma kriptografi semetris berbasis *cipher* blok. Panjang blok yang digunakan fleksibel dan lebih panjang dibandingkan DES, yaitu 128, 192, dan 256 bit. Standar algoritma baik DES dan AES digunakan oleh Pemerintah Federal AS setelah kekuatan algoritma ini dinilai oleh NSA (*National Security Agency*). Tingkat keamanan suatu algoritma standar menjadi sangat penting karena digunakan untuk mengirim informasi yang bersifat sensitif. Karena AES merupakan sebuah revisi dari DES, sudah sepatutnya AES mengalami perbaikan pada sisi tingkat keamanan algoritmanya.

Kata kunci: DES, AES, Enkripsi, NIST.

1. PENDAHULUAN

Kriptografi berasal dari bahasa Yunani κρυπτο (tersembunyi atau rahasia) dan γραφη (menulis) yang artinya "secret writing". Definisi lama mengenai kriptografi yaitu, ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Seiring perkembangan zaman, ilmu kriptografi pun berkembang sedemikian rupa sehingga tidak sebatas mengacak kata, namun juga memberikan aspek keamanan. Kemudian muncul sebuah definisi baru kriptografi: ilmu dan seni untuk menjaga keamanan pesan [Schneier, 1996].

Ilmu kriptografi ini memiliki sejarah yang cukup panjang, dan tercatat sudah digunakan dan berkembang sejak lama. Bangsa Mesir 4000 tahun lalu diketahui menggunakan *hieroglyph* tidak standar untuk menulis pesan. Di Yunani juga tercatat sudah menggunakan kriptografi sejak 400 SM. Sejarah kriptografi juga tercatat ditemukan di negara lain, seperti Jepang dan Cina di abad ke-15, di Inggris di abad ke-17, dan di Jerman pada saat pemerintahan Nazi saat perang dunia ke-2.

Penggunaan kriptografi yang sudah memiliki sejarah panjang menunjukan kebutuhan manusia akan keamanan suatu informasi. Terkadang suatu informasi yang kita miliki ingin kita bagi kepada orang lain namun tidak untuk beberapa orang lainnya, saat seperti inilah penggunaan ilmu kriptografi dibutuhkan. Beberapa kalangan yang diketahui paling membutuhkan kriptografi yaitu:

- Militer (termasuk intelijen dan mata-mata)
- Korps diplomatik
- Diarist
- Lovers

Saat sekelompok orang membutuhan akan keamanan dengan menggunakan kriptografi, maka sekelompok orang tersebut juga akan mengembangkan ilmu kriptografi ini demi menunjang kepentingan mereka. Kelompok orang yang sudah disebutkan di atas tadi *lah* yang sudah banyak berkontribusi dalam memajukan ilmu kriptografi sampai saat ini.

1.1 Enkripsi dan Dekripsi

Secara garis besar terdapat dua buah proses dalam ilmu kriptografi, enkripsi dan dekripsi. Suatu proses untuk mengubah sekelompok informasi agar menjadi tidak dapat dimengerti dengan menggunakan ilmu kriptografi, dinamakan **enkripsi**. Sebaliknya, proses mengubah sekelompok informasi yang sudah diacak kembali menjadi

dapat dimengerti dinamakan **dekripsi**. Sebuah informasi atau pesan sebelum dienkripsi, dinamakan **plainteks**. Plainteks masih dapat dimengerti dan merupakan input dari proses enkripsi. Sedangkan informasi atau pesan yang sudah tidak dapat dimengerti setelah melewati proses enkripsi disebut dengan **cipherteks**.

Proses enkripsi dapat digambarkan dalam notasi matematis sebagai berikut:

P = plainteks
C = cipherteks

$$E(P) = C \quad (1)$$

Sebuah fungsi enkripsi E digunakan untuk mengenkripsi plainteks P sehingga menghasilkan cipherteks C. Kemudian proses dekripsi dapat dinotasikan sebagai berikut:

$$D(C) = P \quad (2)$$

Sebuah fungsi dekripsi D digunakan untuk mendekripsi cipherteks C menjadi plainteks P.

Agar lebih aman, fungsi algoritma saat ini banyak yang menggunakan kunci sebagai masukan selain plainteks. Dengan adanya kunci sebagai masukan, maka dua buah plainteks yang sama dapat dibuat menjadi cipherteks yang berbeda dengan menggunakan kunci yang berbeda.

1.2 Kriptografer dan Kriptanalisis

Seperti yang sudah dijelaskan sebelumnya, tujuan dari kriptografi adalah menyembunyikan suatu pesan agar tidak dimengerti oleh mereka yang tidak berhak. Namun dalam kenyataannya walaupun sudah dienkripsi, orang-orang yang tidak berhak tetap berusaha untuk memecahkan enkripsi tersebut. Karena itu, teknik enkripsi yang aman semakin dibutuhkan oleh orang-orang menggunakan kriptografi untuk menyembunyikan makna suatu pesan atau informasi.

Orang-orang yang berkontribusi dalam mengembangkan algoritma atau teknik kriptografi disebut dengan **kriptografer**. Sedangkan orang-orang yang berusaha untuk memecahkan suatu cipherteks tanpa legitimasi disebut dengan **kriptanalis**. Sedangkan ilmu atau seni untuk untuk memecahkan cipherteks menjadi plainteks tanpa mengetahui kunci yang digunakan disebut **kriptanalisis**.

2. KRIPTOGRAFI

2.1 Klasik atau Modern

Kriptografi secara umum dibagi menjadi dua bagian selama perkembangannya, yaitu kriptografi klasik dan kriptografi modern. Perbedaan yang paling mendasar dari kriptografi klasik dan modern yaitu kriptografi modern beroperasi dalam mode **bit**. Kritptografi klasik secara umum beroperasi dengan mode **karakter**. Yang dimaksud dengan mode disini adalah bagaimana fungsi enkripsi tersebut memanipulasi sebuah plainteks. Sebuah fungsi enkripsi yang bekerja dalam mode bit berarti fungsi enrkipsi tersebut menggunakan setiap bit dari plainteks sebagai masukan. Serupa dengan mode bit, enkripsi dengan mode karakter berarti fungsi enkripsi tersebut menggunakan setiap karakter dalam plainteks sebagai masukan.

Kriptografi dengan manipulasi bit secara teori lebih aman dibandingkan dengan manipulasi karakter. Hal ini disebabkan karena manipulasi bit tidak selalu menghasilkan karakter huruf atau angka seperti yang dihasilkan pada mode karakter, namun dapat menghasilkan karakter yang tidak dapat dibaca. Saat kriptografi dengan mode bit sudah dapat dilakukan, mode karakter sudah jarang digunakan.

2.2 Cipher Aliran dan Cipher Blok

Teknik enkripsi pada enkripsi berbasis bit terbagi dalam dua teknik dasar, yaitu cipher aliran (*stream*) dan cipher blok (*block*). Perbedaan utama dalam keduanya terletak pada **jumlah** bit yang digunakan sebagai masukan fungsi enkripsi. Pada cipher aliran, plainteks diubah menjadi cipherteks setiap bit per bit atau byte per byte. Sedangkan pada cipher blok, plainteks diubah menjadi cipherteks dalam blok-blok yang sama panjangnya.

Sebuah cipher aliran merupakan enkripsi yang bersifat stateful, artinya proses enkripsi tiap bitnya bergantung pada state saat ini. Setiap plainteks akan dikombinasikan dengan sebuah keystream aliran *pseudorandom*. Disebut *pseudorandom* karena bit-bit yang dihasilkan tidak benar-benar random, melainkan dibangkitkan dengan menggunakan suatu variabel. Contoh dari cipher aliran antara lain RC4 dan SEAL. Cipher aliran dapat dieksekusi dengan kecepatan yang tinggi dan tidak membutuhkan kompleksitas tinggi dalam perangkat yang menggunakannya, namun cipher aliran dapat memiliki masalah keamanan yang serius apabila salah dalam menggunakannya.

Cipher blok adalah enkripsi yang mengeksekusi sejumlah bit dengan jumlah/panjang tertentu. Panjang tertentu ini disebut dengan blok. Saat akan dieksekusi, plainteks dibagi menjadi blok-blok bit dengan panjang yang sama.

Panjang kunci pada cipher blok harus sama dengan panjang blok, karena setiap bit blok plainteks akan dienkripsi dengan blok kunci dengan panjang yang sama. Cipher blok dapat bersifat stateless atau stateful.

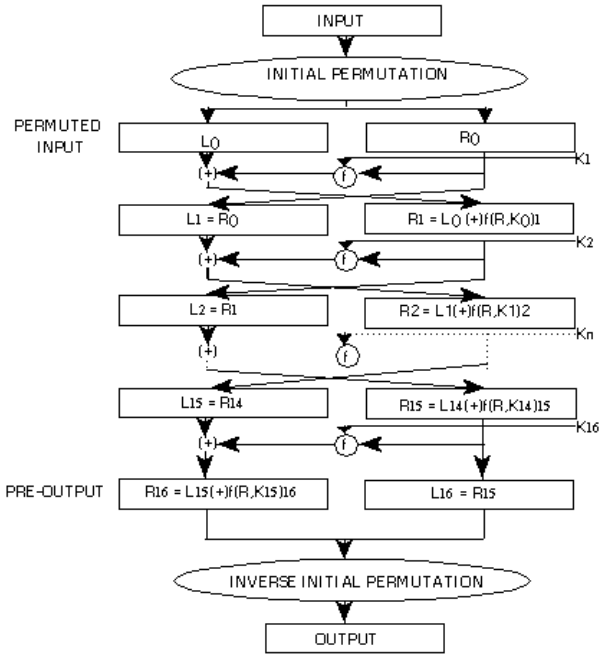
3. DATA ENCRYPTION STANDARD

Data Encryption Standard (atau mulai saat ini disebut DES) sering disalahartikan sebagai suatu algoritma, sebenarnya DES adalah suatu standar. Algoritma sebenarnya pada DES adalah DEA (*Data Encryption Algorithm*) yang diturunkan dari algoritma *Lucifer*. DES memiliki spesifikasi blok sepanjang 64 bit dan kunci sepanjang 56 bit.

Mulai dikembangkan oleh IBM pada tahun 1972 karena adanya studi yang dilakukan pada Pemerintahan US saat itu mengenai kebutuhan keamanan informasi komputer. IBM mengembangkan suatu algoritma yang berdasarkan pada algoritma yang sudah dibuat pada tahun 1973-1974, yaitu algoritma Lucifer yang dirancang oleh Horst Feistel. NBU (*National Bureau of Standards*) Amerika Serikat, saat ini bernama NIST (*National Institute of Standards and Technology*), menetapkan algoritma buatan IBM tersebut sebagai DES pada tahun 1976. DES saat ini telah digunakan dalam berbagai bidang, seperti agen federal, mesin ATM, SSL, dan lain-lain.

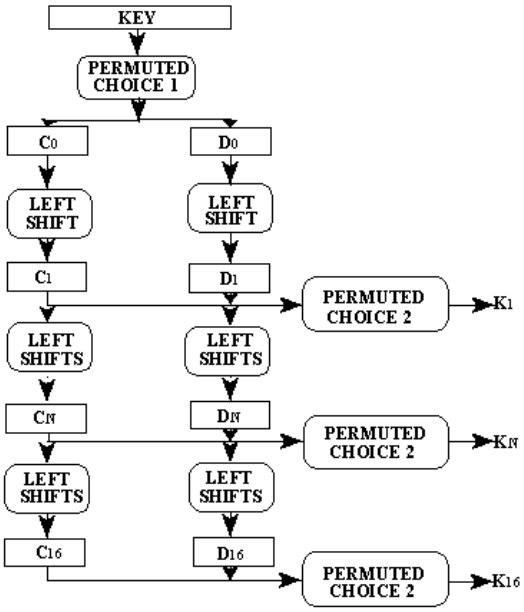
Dalam pengembangan algortima untuk DES oleh IBM ini terdapat kontroversi mengenai keterlibatan NSA (*National Security Agency*) dalam pembangunannya. Dicurigai oleh beberapa kalangan bahwa NSA berusaha membuat *backdoor* dalam DES. Beberapa hal yang dicurigai adalah desakan dari NSA kepada IBM untuk membuat kunci dari DES menjadi 48 bit dari 64 bit, yang pada akhirnya disetujui untuk membuat kunci menjadi sepanjang 56 bit. Hal lain yang dicurigai adalah pembuatan S-box yang dinilai melibatkan NSA secara mendalam.

Pada proses enkripsi DES, setiap blok akan dienkripsi dalam 16 putaran. Setiap putaran digunakan kunci yang berbeda yang dibangkitkan secara internal. Kunci internal ini dibangkitkan oleh kunci eksternal. Intinya setiap blok akan dikenakan dengan IP (*initial permutation*), kemudian kepada sebuah perhitungan dengan masukan sebuah kunci, dan terakhir kepada sebuah permutasi yang merupakan invers dari IP. Perhitungan dengan masukan kunci ini biasa disebut dengan fungsi ***f***, atau disebut juga *cipher function*, dan juga fungsi **KS**, atau disebut juga *key schedule*.



Gambar 1. Komputasi Enkripsi pada DES

Gambar 1 di atas menunjukan bagaimana perputaran 16 kali yang dilakukan pada tiap blok dengan masukan kunci internal yang berbeda. Artinya terdapat 16 kunci internal mulai dari K1, K2, K3, sampai K16. Panjang kunci internal ini adalah 48 bit, yang dibangkitkan secara acak dari kunci masukan 64 bit. Pembangkitan secara acak ini dilakukan dengan memilih blok-blok yang berbeda dari kunci masukan pada tiap iterasinya. Lebih jelasnya proses ini diperlihatkan pada gambar 2.

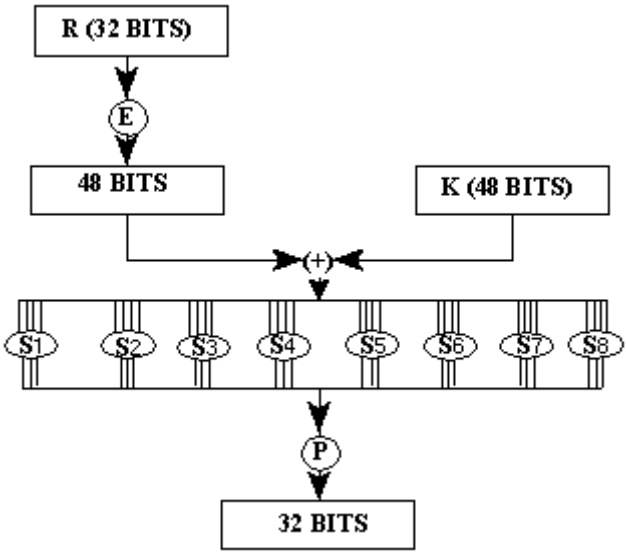


Gambar 2. Pembangkitan Kunci Internal

Proses enkripsi pertama-tama dilakukan dengan membagi dua blok masukan, sehingga didapat dua blok L dan R yang masing-masing sepanjang 32 bit. Kemudian R akan diubah menjadi L pada iterasi selanjutnya, dan L akan dimasukkan ke dalam fungsi ***f*** bersama dengan kunci K.

$$L' = R$$
$$R' = L \,(+)\, f(R,K) \qquad (3)$$

Lebih jelasnya dapat dilihat pada gambar berikut.



Gambar 3. Fungsi *f*(R,K)

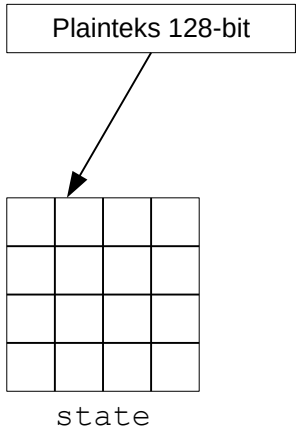
Kemudian langkah ini akan diulangi sebanyak 16 kali dengan masukan kunci yang berbeda pada tiap iterasinya. Langkah dekripsi pada DES hanya merupakan kebalikan dari langkah enkripsinya.

4. ADVANCED ENCRYPTION STANDARD

Advanced Encryption Standard, atau sering juga disebut dengan AES, merupakan sebuah standar baru yang dibuat untuk menggantikan DES. Setelah DES menjadi standar selama beberapa puluh tahun sejak 1976, NIST menilai bahwa DES sudah tidak lagi aman. Setelah diadakan seleksi kepada 15 desain selama kurang lebih 5 tahun, akhirnya pada 26 November 2001 NIST memilih algoritma *Rijndael* sebagai algoritma untuk AES. AES juga merupakan algoritma enkripsi pertama yang digunakan untuk informasi *Top Secret* NSA, namun juga dapat diakses oleh publik.

Dikembangkan oleh dua kriptografer asal Belgia, Joan Daemen dan Vincent Rijmen, Rijndael merupakan algoritma kriptografi tipe block cipher. Walaupun didasarkan pada algoritam Rijndael, sesungguhnya AES tidak persis sama dengan Rijndael. Perbedaan utama diantara keduanya terdapat pada ukuran blok yang didukung. Rijndael mendukung ukuran blok yang lebih banyak dibandingkan dengan AES. Rijndael mendukung ukuran blok dan kunci yang bervariasi, dan mendukung kombinasi ukuran blok dan kunci kelipatan 32 bit dengan minimum 128 bit dan maksimum 256 bit. Sedangkan AES memiliki ukuran blok yang tetap, yaitu 128 bit, serta hanya mendukung tiga ukuran bit kunci, yaitu 128, 192, dan 256 bit.

Berbeda dengan pendahulunya, DES, AES tidak menggunakan jarigan Feistel. AES menggunakan jaringan subtitusi permutasi dalam proses enkripsinya. AES juga berbeda dengan dengan DES karena AES berorientasi byte, sedangkan DES berorientasi bit. Selama kalkulasi plainteks menjadi cipherteks, digunakan sebuah array of bytes berukuran NROWS x NCOLS. Untuk data 128 bit, maka dapat digunakan state 4 x 4.



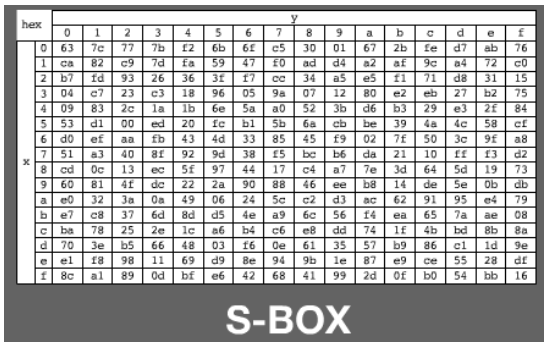
Gambar 4. State Pada AES

Secara umum algoritma enkripsi ini didefnisikan sebagai berikut:

- Key Expansion
- Initial Round
    1. AddRoundKey
- Rounds
    1. SubBytes
    2. ShiftRows
    3. MixColumns
    4. AddRoundKey
- Final Round
    1. SubBytes
    2. ShiftRows
    3. AddRoundKey

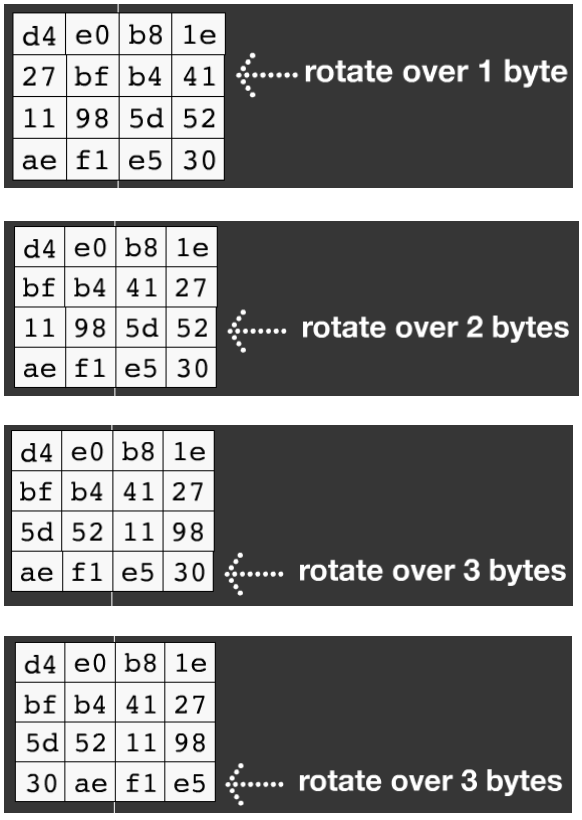
AddRoundKey: melakukan XOR antara state awal (plainteks) dengan cipher key. Tahap ini disebut juga dengan *initial round*.

SubBytes: memetakan setiap byte dari array state dengan menggunakan S-box.



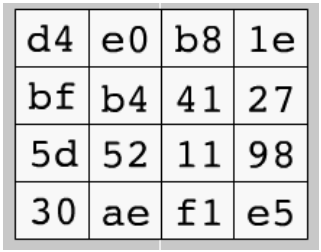
Gambar 5. S-box

ShiftRows: melakukan pergeseran secara wrapping pada 3 baris terakhir dari array state.

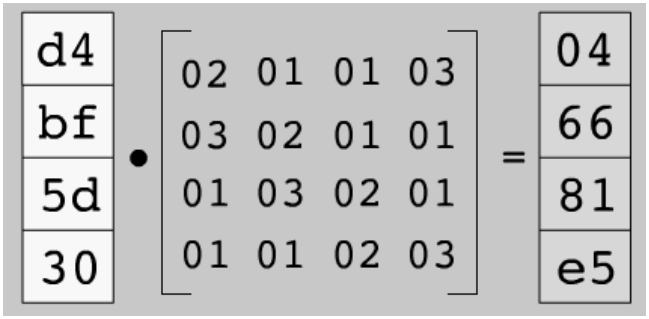


Gambar 6. Proses ShiftRows

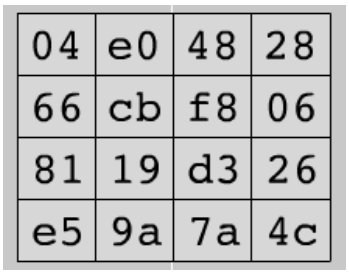
MixColumns: transformasi ini mengalikan setiap kolom dari array state dengan polinom a(x) mod (x^4+1). Setiap kolom diperlakukan sebagai polinom 4-suku pada GF(2^8).



Gambar 7. Sebelum MixColumns



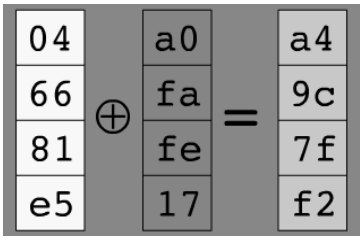
Gambar 8. Operasi MixColumns Kolom Pertama



Gambar 9. Hasil Transformasi MixColumns

Transformasi AddRoundKey: Transformasi ini melakukan operasi XOR terhadap sebuah *round key* dengan *array state*, dan hasilnya disimpan di *array state*.

Gambar 10. Plainteks dan *Round Key*



Gambar 11. Proses XOR kolom pertama State dengan kolom pertama *Round Key*

5. PERBANDINGAN TINGKAT KEAMANAN

Setelah melihat bagaimana DES dan AES bekerja, sekarang saatnya membandingkan tingkat keamanan dari kedua algoritma ini.

Dari fungsi algoritma yang digunakan, fungsi yang digunakan oleh AES dapat dikatakan lebih sederhana dibandingkan dengan DES karena tidak melibatkan jaringan Feistel. Algoritma pada AES juga menggunakan lebih sedikit putaran dibandingkan dengan DES, AES memiliki maksimum 14 putaran sedangkan DES 16 putaran.

Walaupun ternyata AES memiliki fungsi yang lebih sederhana dibandingkan DES, namun AES memiliki kunci yang lebih panjang dibandingkan dengan DES. Apabila DES memiliki kunci sepanjang 64 bit (56 bit), AES memiliki beberapa tipe dengan panjang kunci 128, 192, dan 256 bit.

DES sudah dinilai sebagai algoritma yang kurang aman. Dalam data yang telah ada, tercatat bahwa DES dapat dipecahkan dengan menggunakan sebuah mesin khusus pencari kunci seharga $250.000. Dengan menggunakan mesin ini, sebuah cipherteks DES dapat dipecahkan dalam 56 jam. Kemudian juga terdapat 3 buah teori serangan terhadap DES, walaupun sebenarya secara praktikal tidak dapat dilakukan. Ketiga teori tersebut adalah *Differential Cryptanalysis*, *Linear Cryptanalisis*, dan *Improved Davies Attack*. Pada *Differential Cryptanalysis*, untuk memecahkan 16 putaran kode, maka dibutuhkan pengecekan terhadap 2^{47} plainteks. DES memang sudah dibuat untuk tahan terhadap serangan jenis ini. Kemudian pada serangan jenis *Linear Cryptanalisis* menciptakan 2^{43} kemungkinan, yang juga mustahil untuk dilakukan. Yang terakhir yaitu *Improved Davies Attack* memiliki kompleksitas komputasional sebanyak 2^{50} dan juga membutuhkan 2^{50} plainteks.

Kemudian beberapa catatan serangan terhadap AES pun juga ada. Untuk AES 128 bit, maka setidaknya terdapat 2^{128} kemungkinan kunci. Serangkaian serangan kirptografik (serangan apapun yang lebih cepat dibandingkan *exhaustive search*) juga dicatat terjadi kepada AES. Walaupun terjadi serangkaian serangan terhadap AES, namun serangan tersebut terjadi kepada AES yang tidak memiliki putaran penuh. Salah satu serangan yang paling sukses yaitu dilakukan oleh Alex Biryukov, Orr Dunkelman, Nathan Keller, Dmitry Khovratovich, dan Adi Shamir, dan membutuhkan waktu 2^{39} untuk mendapatkan kunci 256 bit dengan 9 putaran, 2^{45} untuk 10 putaran, dan 2^{70} untuk 11 putaran. Karena AES full memiliki 14 putaran, maka jenis serangan ini tidak cocok.

6. KESIMPULAN

Sampai saat ini belum ditemukan suatu serangan kriptografik yang benar-benar efektif terhadap DES maupun AES. Walaupun begitu, serangan *exhaustive search* terhadap DES 16 putaran telah terbukti berhasil memecahkannya. Namun untuk AES, karena memiliki panjang kunci yang jauh lebih panjang dibandingkan DES, maka dibutuhkan waktu jauh lebih lama untuk menemukan kunci AES dibandingkan DES secara *brute force*. Apabila terdapat suatu mesin yang membutuhkan 1 detik untuk memecahkan suatu kunci DES, maka dibutuhkan kurang lebih 149 triliun tahun bagi mesin tersebut untuk menemukan kunci AES 128 bit. Karena jenis serangan non-*brute force* belum ditemukan efektif terhadap AES, dan serangan *brute force* juga tidak memungkinkan secara constraint waktu, maka dapat disimpulkan bahwa algoritma AES lebih aman dibandingkan dengan DES.

REFERENSI

1. http://simple.wikipedia.org/wiki/Advanced_Encryption_Standard
2. http://www.itl.nist.gov/fipspubs/fip46-2.htm
3. http://en.wikipedia.org/wiki/Data_Encryption_Standard
4. http://www.nist.gov/public_affairs/releases/aesq&a.htm
5. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
6. http://www.networkworld.com/research/2001/0730feat2.html
7. http://armandoon.wordpress.com/2008/05/14/aes-strength-vs-des-strength/
8. http://www.nexiondata.com/products/options/encrypt/desaes.htm
9. http://people.seas.harvard.edu/~salil/cs120/docs/lec13.pdf