STUDI KRIPTOGRAFI MENGENAI *TRIPLE DES* DAN *AES*

Mohammad Gilang Kautzar – NIM : 13505101

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if115101@students.if.itb.ac.id

Abstrak

Makalah ini membahas tentang studi mengenai *Data Encryption Standard (DES)* dalam bentuk *Triple DES* dan *AES*. *DES* merupakan sebuah algoritma enkripsi sandi blok kunci simetri. DES ini berukuran blok 64-bit dan ukuran kunci 56-bit. Untuk saat ini, ukuran kunci 56-bit tergolong pendek, sehingga dapat menambah resiko keamanan karena penyerangan dengan metode *Brute Force Attack* hanya membutuhkan waktu beberapa hari.

Sebagai solusi dari hal di atas, lahirlah metode *Triple DES* yang dapat digunakan sebagai pengenkripsi namun tidak mengubah algoritma dari *DES*. *Triple DES* terbagi menjadi dua variasi, yaitu *2TDES* dan *3TDES*. Jenis *2TDES* hanya menggunakan 2 buah kunci, sementara *3TDES* menggunakan 3 buah kunci. *2TDES* memiliki kunci berukuran 112-bit (2 kali lipat *DES*), dan *3TDES* memiliki kunci berukuran 168-bit (3 kali lipat *DES*).

Solusi lain adalah algoritma baru, yaitu *Advanced Encryption Standard (AES)*. Saat ini, AES digunakan sebagai standar algoritma kriptografi yang terbaru. AES menggantikan DES (*Data Encryption Standar*) yang pada tahun 2002 sudah berakhir masa penggunaannya. DES juga dianggap tidak mampu lagi untuk menjawab tantangan perkembangan teknologi komunikasi yang sangat cepat. AES sendiri adalah algoritma kriptografi dengan menggunakan algoritma Rijndael yang dapat mengenkripsi dan mendekripsi blok data sepanjang 128 bit dengan panjang kunci 128 bit, 192 bit, atau 256 bit.

Kata kunci: *Data Encryption Standard*, *Triple Data Encryption Standard*, *Advanced Encryption Standard,* enkripsi, dekripsi.

1. Pendahuluan

Seiring dengan perkembangan zaman, kebutuhan manusia mengingkat. Termasuk kebutuhan akan informasi. Salah satu contohnya adalah internet. Dalam perkembangannya, internet tidak lagi hanya dimonopoli oleh beberapa elemen industri, tetapi juga digunakan oleh sebagian besar industri kecil dan menengah untuk membantu usaha mereka. Di tengah-tengah pergolakan teknologi informasi yang kian semarak, internet tidak lagi menjamin penyediaan informasi yang bersifat privat. Berbagai mesin-pencari (*search-engine*) dan layanan *e-commerce* juga berkembang. Ditambah serangan *virus* dan *spam* yang kian berkembang dan terus mengintai.

Oleh sebab itu, pengiriman dan penyimpanan data melalui media elektronik memerlukan suatu proses yang mampu menjamin keamanan dan keutuhan dari data tersebut. Untuk menjamin keamanan dan keutuhan dari suatu data, dibutuhkan suatu proses enkripsi-dekripsi (penyandian). Enkripsi dilakukan ketika data akan dikirim. Proses ini akan mengubah suatu data asal menjadi data rahasia yang tidak dapat dibaca. Sementara itu, proses dekripsi dilakukan oleh penerima data yang dikirim tersebut. Data rahasia yang diterima akan diubah kembali menjadi data asal. Dengan cara penyandian tadi, data asli tidak akan terbaca oleh pihak yang tidak berkepentingan, melainkan hanya oleh penerima yang memiliki kunci dekripsi.

Didorong oleh kegunaan yang penting tadi, teknik (algoritma) penyandian telah berkembang sejak zaman dahulu kala. Mulai dari era sebelum masehi, hingga sekarang algoritma penyandian ini selalu berkembang. Mulai dari algoritma *Caesar Cipher* yang

tergolong sederhana hingga algoritma *Advanced Encryption Standar* (*AES*) yang digunakan saat ini. Namun demikian, penelitian akademis yang ekstensif dalam bidang kriptografi masih tergolong baru, yaitu sekitar pertengahan 1970-an. Pada waktu tersebut, dua buah algoritma dikeluarkan, yaitu metode *RSA* dan metode *DES* yang dikembangkan oleh *IBM*. Algoritma *DES* inilah yang akan dibahas dalam makalah ini.

Algoritma *DES* telah dijadikan standard oleh *NBS*, biro standard nasional Amerika, sejak November 1976. *DES* menggunakan kunci berukuran 56-bit. Ukuran kunci mencerminkan kekuatan dari suatu algoritma enkripsi. Perkembangan perangkat keras maupun perangkat lunak dan meluasnya penggunaan jaringan komputer terdistribusi menyebabkan penemuan bahwa algoritma *DES* sudah tidak aman lagi untuk digunakan, terutama dalam hal pengiriman data melalui internet. Perangkat keras yang khusus digunakan untuk mencari kunci dari *DES* dengan metode paling mendasar, *brute force attack*, dapat dibangun dalam beberapa jam saja. Hal ini disebabkan oleh ukuran kunci dari *DES* yang relatif pendek tadi, yaitu 56-bit. Pertimbangan-pertimbangan tersebut menandakan bahwa sebuah standard algoritma yang baru sangatlah diperlukan untuk tetap menjaga kerahasiaan dari suata data. Dalam hal ini, kunci yang lebih panjang juga merupakan keharusan.

Pada tanggal 26 November 2006, National Institute of Standards and Technology (*NIST*) mengumumkan *Advanced Encryption Standard* (*AES*) kepada publik setelah mengalami proses standardisasi selama 5 tahun. Metode *AES* dikembangkan oleh dua orang kriptografer asal Belgia, yaitu Joan Daemen dan Vincent Rijmen. Proyek *AES* ini dibuat dengan tujuan utama untuk menggantikan algoritma *DES* yang cenderung tidak aman lagi untuk digunakan. Algoritma *AES* merupakan algoritma kriptografi simetri yang beroperasi dalam mode penyandi blok (*block cipher*) yang memproses data 128-bit dengan panjang kunci 128-bit, 192-bit, atau 256-bit.

Salah satu alternatif lain adalah penggunan *Triple DES* untuk menggantikan *DES*. Sebetulnya *Triple DES* memiliki algoritma yang sama dengan *DES*, namun metode ini memperbesar ukuran kunci karena *Triple DES* dibentuk dari algoritma *DES* dengan cara menggunakannya tiga kali.

2. Kriptografi

Kriptografi berasal dari dua kata Yunani, yaitu *Crypto* yang berarti rahasia dan *Grapho* yang berarti menulis. Secara umum kriptografi dapat diartikan sebagai ilmu dan seni penyandian yang bertujuan untuk menjaga keamanan dan kerahasiaan suatu pesan. Kriptografi pada dasarnya sudah dikenal sejak lama. Menurut catatan sejarah, kriptografi sudah digunakan oleh bangsa Mesir sejak 4000 tahun yang lalu oleh raja-raja Mesir pada saat perang untuk mengirimkan pesan rahasia kepada panglima perangnya melalui kurir-kurinya. Orang yang melakukan penyandian ini disebut *kriptografer*, sedangkan orang yang mendalami ilmu dan seni dalam membuka atau memecahkan suatu algoritma kriptografi tanpa harus mengetahui kuncinya disebut *kriptanalis*.

Seiring dengan perkembangan teknologi, algoritma kriptografi pun mulai berubah menuju ke arah algoritma kriptografi yang lebih rumit dan kompleks. Kriptografi mau tidak mau harus diakui mempunyai peranan yang paling penting dalam peperangan sehingga algoritma kriptografi berkembang cukup pesat pada saat Perang Dunia I dan Perang Dunia II. Menurut catatan sejarah, terdapat beberapa algoritma kriptografi yang pernah digunakan dalam peperangan, diantaranya adalah ADFVGX yang dipakai oleh Jerman pada Perang Dunia I, *Sigaba/M-134* yang digunakan oleh Amerika Serikat pada Perang Dunia II, *Typex* oleh Inggris, dan *Purple* oleh Jepang. Selain itu Jerman juga mempunyai mesin legendaris yang dipakai untuk memecahkan sandi yang dikirim oleh pihak musuh dalam peperangan yaitu, *Enigma*.

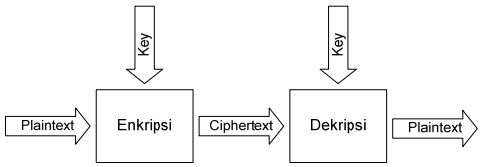
Algoritma kriptografi yang baik tidak ditentukan oleh kerumitan dalam mengolah data atau pesan yang akan disampaikan. Ada 4 syarat yang perlu dipenuhi, yaitu:

1. **Kerahasiaan**. Pesan (*plaintext*) hanya dapat dibaca oleh pihak yang memliki kewenangan.
2. **Autentikasi**. Pengirim pesan harus dapat diidentifikasi dengan pasti, penyusup harus dipastikan tidak bisa berpura-pura menjadi orang lain.

3. **Integritas**. Penerima pesan harus dapat memastikan bahwa pesan yang dia terima tidak dimodifikasi ketika sedang dalam proses transmisi data.
4. ***Non-Repudiation***. Pengirim pesan harus tidak bisa menyangkal pesan yang dia kirimkan.

Kriptografi pada dasarnya terdiri dari dua proses, yaitu proses enkripsi dan proses dekripsi. Proses enkripsi adalah proses penyandian pesan terbuka menjadi pesan rahasia (*ciphertext*). *Ciphertext* inilah yang nantinya akan dikirimkan melalui saluran komunikasi terbuka. Pada saat *ciphertext* diterima oleh penerima pesan, maka pesan rahasia tersebut diubah lagi menjadi pesan terbuka melalui proses dekripsi sehingga pesan tadi dapat dibaca kembali oleh penerima pesan. Secara umum, proses enkripsi dan dekripsi dapat digambarkan sebagai berikut :

Gambar 1



Dalam sistem komputer, pesan terbuka (*plaintext*) diberi lambang M, yang merupakan singkatan dari *Message*. *Plaintext* ini dapat berupa tulisan, foto, atau video yang berbentuk data biner. *Plaintext* inilah yang nantinya akan dienkripsi menjadi pesan rahasia atau *ciphertext* yang dilambangkan dengan C (*Ciphertext*). Secara matematis, fungsi enkripsi ini dinotasikan dengan :

$$E(M) = C$$

Sedangkan fungsi dekripsi adalah proses pembalikan dari *ciphertext* menjadi *plaintext* kembali. Secara matematis dinotasikan sebagai berikut :

$$D(C) = M$$
$$D(E(M)) = M$$

3. Panjang Kunci

Keamanan dari sebuah teknik penyandian tergantung dari dua hal : algoritma penyandian dan panjang kunci (*key*). Algoritma sangat menentukan kekuatan dari sebuah teknik penyandian, tetapi panjang kunci juga tidak kalah penting dalam menentukan kekuatan sebuah teknik penyandian.

Sebagai contoh, apabila seorang kriptanalis mengetahui algoritma yang dipakai untuk melakukan teknik penyandian terhadap suatu pesan, maka kriptanalis tersebut harus mendapatkan kunci yang dipakai terlebih dahulu sebelum dapat melakukan dekripsi terhadap semua *ciphertext* yang dia punya. Satu-satunya cara untuk mendapatkan kunci yang dipakai adalah dengan cara mencoba semua variasi kunci yang ada. Teknik serangan ini sering dikenal dengan nama *brute force attack.*

Adalah mudah untuk menghitung banyaknya variasi kunci yang ada. Apabila panjang kunci adalah 8 bit, maka ada 2^8 atau 256 kemungkinan kunci yang dapat dicoba. Dari 256 percobaan ini, peluang untuk mendapatkan kunci yang benar adalah 50 persen setelah melalui setengah usaha percobaan. Bila panjang kunci 56 bit, maka ada 2^{56} kemungkinan variasi kunci. Dengan menganggap sebuah superkomputer dapat mencoba satu juta kunci per detik, maka diperkirakan sekitar 2285 tahun untuk menemukan kunci yang benar. Bila menggunakan panjang kunci 64 bit, maka dengan superkomputer yang sama akan membutuhkan 585 ribu tahun. Dengan jangka waktu yang lama ini, maka dapat dipastikan bahwa pesan yang disandikan tersebut tidak mempunyai arti lagi apabila telah berhasil dilakukan dekripsi.

Dengan melihat situasi ini, maka kriptografi yang baik akan memilih untuk menggunakan sepanjang mungkin kunci yang akan digunakan, namun hal ini tidak dapat diterapkan begitu saja. Semakin panjang kunci, semakin lama pula waktu yang digunakan oleh komputer untuk melakukan proses enkripsi. Oleh sebab itu, panjang kunci yang akan digunakan hendaknya memperhatikan 3 hal, yaitu seberapa penting data yang akan dirahasiakan, berapa lama waktu yang dibutuhkan agar data tersebut tetap aman, dan seberapa kuat kemampuan kriptanalis dalam memecahkan teknik penyandian kita. Saat ini yang paling banyak dipakai adalah kunci dengan panjang 128 bit

karena panjang kunci ini dianggap paling optimal untuk saat ini.

4. Algoritma Kriptografi Kunci Simetri

Bidang ilmu kriptografi modern terbagi menjadi beberapa area studi, seperti algoritma kriptografi kunci simetri, algoritma kriptografi kunci publik, kriptanalasis, kriptanalisis primitif, dan kriptanalitis protokol. Dalam subab ini akan dibahas algoritma kriptografi kunci simetri.

Pada algoritma kriptografi kunci simetri, kunci enkripsi memiliki hubungan dengan kunci dekripsi. Dalam hal ini, kunci enkripsi bisa identik dengan kunci enkripsi, ataupun ada perubahan yang simpel di antara keduanya. Kunci enkripsi dan dekripsi tersebut merepresentasikan rahasia bersama antara dua atau lebih pihak yang dapat digunakan sebagai penghubung informasi privat.

Algoritma kriptografi kunci simetri ini secara umum tidak intensif dalam perhitungan. Hal tersebut menyebabkan kualitas algoritma kriptografi kunci simetri ini lebih cepat beratus-ratus atau bahkan beribu-ribu kali dari algoritma kriptografi kunci publik (kunci asimetri).

Algoritma kriptografi kunci simetri dapat dikelompokkan menjadi dua kategori, yaitu:

1. *Cipher* aliran (*stream ciphers*)
   Algoritma kriptografi beroperasi pada plainteks/cipherteks dalam bentuk bit tunggal, yang dalam hal ini rangkaian bit dienkripsikan dan didekripsikan bit per bit.
2. *Cipher* blok (*block ciphers*)
   Algoritma kriptografi beroperasi pada plainteks/cipherteks dalam bentuk blok bit, yang dalam hal ini rangkaian bit dibagi menjadi blok-blok bit yang panjangnya sudah ditentukan sebelumnya.

4.1 *Reversibility* dari Algoritma Kriptografi Kunci Simetri

Fungsi enkripsi, sesuai dengan definisinya, harus dapat dibalik karena suatu pesan yang telah dienkripsi harus dapat didekripsi lagi untuk mendapatkan pesan aslinya.

Beberapa metode telah digunakan untuk masalah ini. Sejak dulu telah terdapat buku-buku sandi (*book ciphers*), di mana di dalamnya terdapat kunci bersama yang berhubungan dengan sebagian isi dari sebuah buku, atau kunci-otomatis (*auto-key ciphers*) di mana di dalamnya sebuah kunci diturunkan dari *plaintext*, *grill chiphers*, dll. Di masa modern ini, sejak komputer menjadi hal yang sangat umum di masyarakat, sebagian besar algoritma kunci simetri beradasarkan pada ronde (*rounds*) yang diulang-ulang. Biasanya sebuah skema yang cenderung simpel untuk setiap ronde digunakan berulang seperti di contoh yang akan dibahas berikut. Contoh ini disebut-sebut sebagai milik Horst Feistel.

Bit-bit yang akan dienkripsi dibagi menjadi dua bagian p1 dan p2. p1 tidak diubah, sementara p2 ditambahkan ke fungsi *hash* satu arah (*one-way hashed function*). Kedua hasil tadi kemudian ditukar. Proses ini disebut ronde.

Contoh.:

Dimana p1, p2, key adalah vektor bit, dan ‘,’ adalah sebuah operator konkatenasi dan f adalah fungsi

$$p1, p2 \mapsto p2', p1$$

di mana:

$$p2' = p2 + \mathrm{f}(p1, key)$$

Karena keluaran dari ronde tersebut masih memiliki akses ke p1, dan penambahan tersebut adalah proses yang dapat dibalik, maka operasi ini bisa dikembalikan, untuk seluruh fungsi f satu arah.

Karena satu buah ronde saja kurang aman, tanpa mengubah p1, mengulang operasi tadi lebih dari sekali (biasanya dengan fungsi dan kunci ronde yang berbeda) dapat menambah kekuatan/keamanannya.

Untuk mendekripsi ronde yang banyak (*multiple rounds*), setiap ronde dikembalikan dalam urutan terbalik, dan kunci-kuncinya dipakai dalam urutan yang terbalik pula.

Setelah beberapa proses ronde (biasanya antara 8 sampai 64), keluarannya akan menjadi teracak, di mana dalam sebuah

penyandian yang baik, tidak ada metode penyerangan yang lebih praktis dari *brute force key search*. Dengan kunci yang cukup panjang, serangan *brute force key search* dapat dibuat menjadi tidak praktis.

4.2 Kekurangan Algoritma Kriptografi Kunci Simetri

Kekurangan dari algoritma kriptografi kunci simetri ini adalah diperlukannya sebuah kunci rahasia bersama (*shared secret key*) dengan sebuah kopi di setiap ujung. Karena kunci adalah hal yang memiliki petensi untuk ditemukan oleh dalam suatu pembobolan, kunci-kunci tersebut perlu diganti sesering mungkin dan diamankan ketika proses distribusi dan sedang digunakan. Syarat yang harus dipenuhi, yaitu memilih, menyebarkan, dan menyimpan kunci-kunci tanpa kesalahan dan cacat (disebut manajemen kunci), sangat sulit untuk dicapai.

Dalam rangka untuk mengamankan komunikasi antar orang dalam suatu populasi yang terdiri dari n orang, sejumlah n(n-1)/2 kunci diperlukan. Saat ini, sangat umum bahwa algoritma kriptografi kunci asimetri digunakan untuk menyebarkan kunci-kunci simetri di awal sesi, kemudian, kemudian dilanjutkan dengan pengambil-alihan oleh algoritma kunci simetri yang lebih cepat. Masalah yang sama dari penyebaran kunci yang aman masih muncul di tingkat asimetri, namun masih lebih mudah diatasi. Bagaimanapun juga, kunci simetri hampir selalu dipakai kenyataannya.

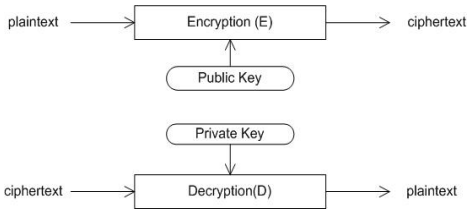
Algoritma kunci simetri tidak dapat digunakan untuk tujuan autentifikasi atau proses perjanjian. Kriptografi kunci simetri seringkali disebut kriptografi kunci konvensional atau kunci rahasia.

5. Algoritma Kriptografi Kunci Publik

Kriptografi *public key* sering disebut dengan kriptografi asimetri. Berbeda dengan kriptografi kunci simetri, kunci yang digunakan pada proses enkripsi dan proses dekripsi pada kriptografi public key ini berbeda satu sama lain. Jadi dalam kriptografi *public key*, suatu *key generator* akan menghasilkan dua kunci berbeda dimana satu kunci digunakan untuk melakukan proses enkripsi dan kunci yang lain digunakan untuk melakukan proses dekripsi.

Kunci yang digunakan untuk melakukan enkripsi akan dipublikasikan kepada umum untuk dipergunakan secara bebas. Oleh sebab itu, kunci yang digunakan untuk melakukan enkripsi disebut juga sebagai *public key*. Sedangkan kunci yang digunakan untuk melakukan dekripsi akan disimpan oleh pembuat kunci dan tidak akan dipublikasikan kepada umum. Kunci untuk melakukan dekripsi ini disebut *private key.*

Gambar 2 Skema Kriptografi Asimetris



Dengan cara demikian, semua orang yang akan mengirimkan pesan kepada pembuat kunci dapat melakukan proses enkripsi terhadap pesan tersebut, sedangkan proses dekripsi hanya dapat dilakukan oleh pembuat atau pemilik kunci dekripsi. Dalam kenyataannya, kriptografi asimetris ini dipakai dalam ssh, suatu layanan untuk mengakses suatu server.

6. *Cipher* Blok

Dalam kriptografi, *cipher* blok adalah sebuah algoritma kunci simetri yang beroperasi dengan bilangan bit yang panjangnya sudah *fixed*. Ketika mengenkripsi, sebuah *cipher* blok mungkin mengambil 128-bit *plaintext* sebagai masukan, dan menghasilkan 128-bit blok *ciphertext* sebagai keluaran. Transformasi ini diatur dengan input kedua, yaitu kunci rahasia. Pada proses dekripsi, hal yang mirip terjadi: algoritma dekripsi akan mengambil 128 blok *ciphertext* bersama dengan kunci rahasia, dan mengeluarkan 128-bit *plaintext* yang asli.

Sebuah *cipher* blok terdiri dari dua buah pasang algoritma, satu untuk enkripsi, E, dan yang lain untuk dekripsi, E^{-1}. Kedua algoritma menerima dua buah masukan: sebuah blok sebesar n-bit dan sebuah kunci sebesar k-bit, dan mengeluarkan *output* n-bit. Untuk setiap kunci tetap (*fixed key*), proses dekripsi adalah invers dari proses enkripsi, jadi:

$$E_K^{-1}(E_K(M)) = M$$

Untuk seluruh blok M dan kunci K.

Untuk setiap kunci K, E_K adalah permutasi (pemetaan bijektif) dari blok masukan. Setiap kunci mengambil satu buah permutasi dari 2^n! set yang mungkin.

Ukuran blok, n, biasanya adalah 64-bit atau 128-bit, namin beberapa *cipher* memiliki ukuran yang bervariasi. 64-bit adalah ukuran yang paling umum sampa pertengahan 1990an, ketika desain-desain baru mulai berpindah ke ukuran 128-bit yang lebih panjang. Salah satu dari beberapa mode operasi umumnya digunakan bersama dengan skema *padding* untuk mengenkripsi suatu *plaintext* yang ukurannya telah ditentukan. Setiap mode memiliki karakter yang berbeda mengenai kesalahan (*error*) dihasilkan, kemudahan mengakses secara random, dan kerapuhan terhadap sekian jenis serangan. Ukuran tipikal dari kunci (k) adalah 40-bit, 56-bit, 64-bit, 80-bit, 128-bit, 192-bit, dan 256-bit. Di tahun 2006 ini, ukuran 80 bit biasanya diambil sebagai ukuran minimum dari kunci yang diperlukan untuk mencegah *brute force attacks*.

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

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

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

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

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

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

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

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

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

Enkripsi dengan kunci K dinyatakan dengan persamaan

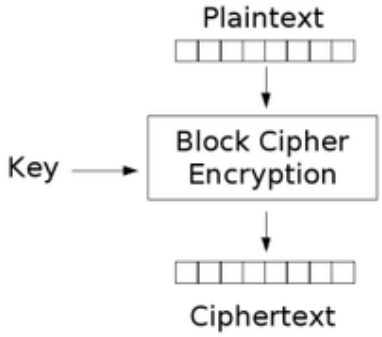
$$E_k(P) = C,$$

sedangkan dekripsi dengan kunci K dinyatakan dengan persamaan

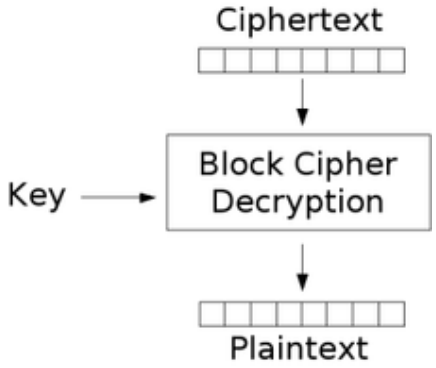
$$D_k(C) = P$$

Skema enkripsi dan dekripsi dengan *cipher* blok dapat dilihat pada Gambar 1.

Gambar 3 Skema Enkripsi dengan *Cipher* Blok



Gambar 4 Skema Dekripsi dengan *Cipher* Blok



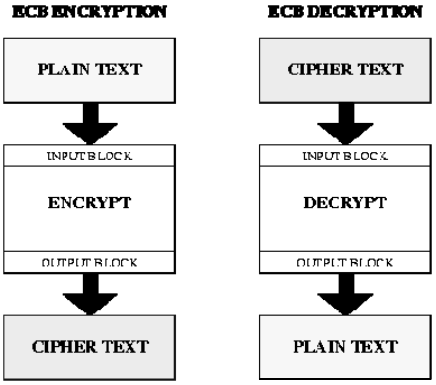
Pada *cipher* blok terdapat empat jenis mode operasi, yaitu *Electronic Code Book (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB).*

6.1. *Electronic Code Book* (*ECB*)

Mode ECB adalah mode yang paling umum dan paling mudah untuk diimplementasikan. Cara yang digunakan adalah dengan membagi data ke dalam blok-blok data terlebih dahulu

yang besarnya sudah ditentukan. Blok-blok data inilah yang disebut *plaintext* karena blok data ini belum disandikan. Proses enkripsi akan langsung mengolah *plaintext* menjadi *ciphertext* tanpa melakukan operasi tambahan. Suatu blok *plaintext* yang dienkripsi dengan menggunakan kunci yang sama akan menghasilkan *ciphertext* yang sama.

Gambar 5 Mode Operasi ECB

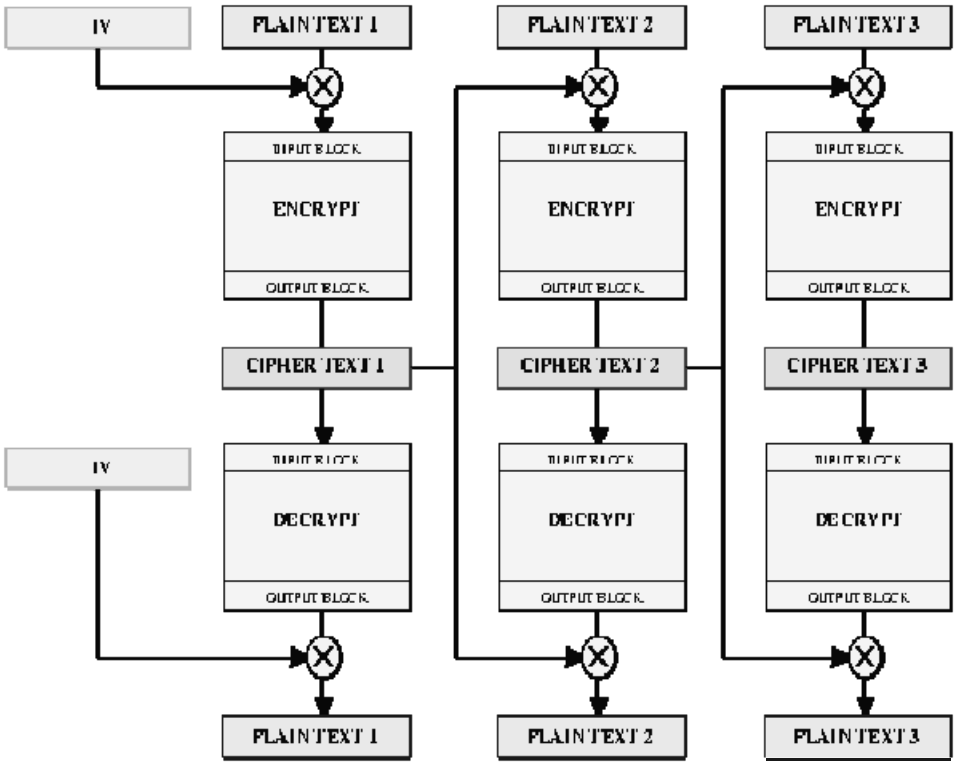


Keuntungan dari mode OBC ini adalah kemudahan dalam implementasi dan pengurangan resiko salahnya semua *plaintext* akibat kesalahan pada satu *plaintext*. Namun mode ini memiliki kelemahan pada aspek keamanannya. Dengan mengetahui pasangan *plaintext* dan *ciphertext*, seorang kriptanalis dapat menyusun suatu *code book* tanpa perlu mengetahui kuncinya.

6.2. *Cipher Block Chaining* (*CBC*)

Pada CBC digunakan operasi umpan balik atau dikenal dengan operasi berantai (*chaining*). Pada CBC, hasil enkripsi dari blok sebelumnya adalah *feedback* untuk enkripsi dan dekripsi pada blok berikutnya. Dengan kata lain, setiap blok *ciphertext* dipakai untuk memodifikasi proses enkripsi dan dekripsi pada blok berikutnya.

Gambar 6 Mode Operasi CBC

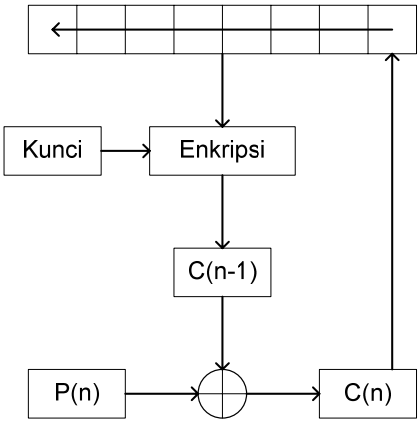


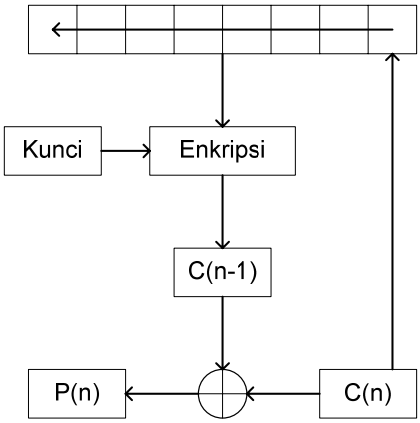
Pada CBC diperlukan data acak sebagai blok pertama. Blok data acak ini sering disebut *initialization vector* atau IV. IV digunakan hanya untuk membuat suatu pesan menjadi unik dan IV tidak mempunyai arti yang penting sehingga IV tidak perlu dirahasiakan.

6.3. *Cipher Feedback* (*CFB*)

Pada mode CBC, proses enkripsi atau dekripsi tidak dapat dilakukan sebelum blok data yang diterima lengkap terlebih dahulu. Masalah ini diatasi pada mode *Cipher Feedback* (CFB). Pada mode CFB, data dapat dienkripsi pada unit-unit yang lebih kecil atau sama dengan ukuran satu blok. Misalkan pada CFB 8 bit, maka data akan diproses tiap 8 bit.

Gambar 7 Mode Operasi CFB



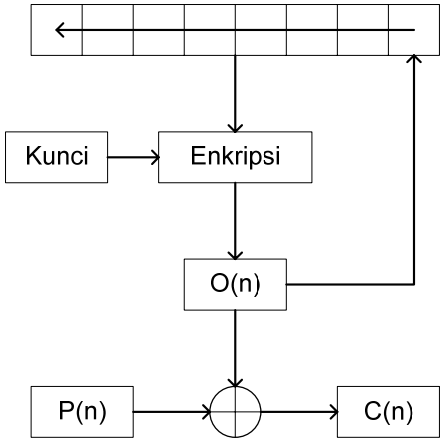


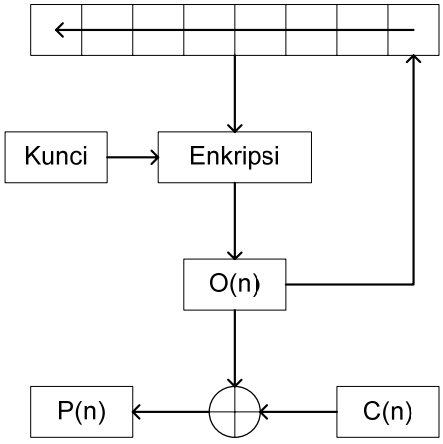
Pada permulaan proses enkripsi, IV akan dimasukkan dalam suatu *register* geser. IV ini akan dienkripsi dengan menggunakan kunci yang sudah ada. Dari hasil enkripsi tersebut, akan diambil 8 bit paling kiri atau *Most Significant Bit* untuk di-XOR dengan 8 bit dari *plaintext*. Hasil operasi XOR inilah yang akan menjadi *ciphertext* dimana *ciphertext* ini tidak hanya dikirim untuk ditransmisikan tetapi juga dikirim sebagai *feedback* ke dalam *register* geser untuk dilakukan proses enkripsi untuk 8 bit berikutnya.

6.4. *Output Feedback* (*OFB*)

Sama pada mode CFB, mode OFB juga memerlukan sebuah *register* geser dalam pengoperasiannya. Pertama kali, IV akan masuk ke dalam *register* geser dan dilakukan enkripsi terhadap IV tersebut. Dari hasil proses enkripsi tersebut akan diambil 8 bit paling kiri untuk dilakukan XOR dengan *plaintext* yang nantinya akan menghasilkan *ciphertext*. *Ciphertext* tidak akan diumpan balik ke dalam *register* geser, tetapi yang akan diumpan balik adalah hasil dari enkripsi IV.

Gambar 8 Mode Operasi OFB





7. *Triple DES* (*TDES*)

Dalam kriptografi, *Triple DES* adalah sebuah *cipher* blok yang dibentuk oleh *DES* dengan menggunakannya tiga kali.

Ketika diketahui bahwa kunci berukuran 56-bit dari DES tidak cukup kuat untuk menjaga dari *brute force attacks*, *Triple DES* dipilih sebagai cara simpel untuk memperbesar ukuran kunci tanpa perlu mengganti algoritma. Penggunaan dari tiga tahap tersebut penting untuk mencegah *meet-in-the-middle attacks* yang efektif untuk digunakan terhadap enkripsi *Double DES*. Catat bahwa *DES* bukanlah sebuah grup (dalam matematika), karena jika merupakan grup, pembangunan *Triple DES* akan ekivalen dengan operasi *Single DES* yang berarti tidak lagi aman.

Variasi paling simpel dari *Triple DES* adalah:

$DES(k_3;DES(k_2;DES(k_1;M)))$

di mana M adalah blok pesan yang akan dienkripsi, k1, k2, dan k3, adalah kunci *DES*. Variasi ini umumnya diketahui sebagai *EEE* karena ketiga operasi *DES* adalah proses enkripsi. Untuk menyederhanakan operasi antara *DES* dan *TDES*, langkah tengah biasanya diganti dengan proses dekripsi (*EDE mode*):

$DES(k_3;DES^{-1}(k_2;DES(k_1;M)))$

maka sebuah enkripsi *DES* dengan kunci k dapat direpresentasikan sebagai *TDES-EDE* dengan k1=k2=k3=k. Pemilihan proses dekripsi pada langkah tengah tidak mempengaruhi keamanan dari algoritma.

7.1 Keamanan *Triple DES* (*TDES*)

Secara umum *TDES* dengan tiga kunci berbeda memiliki kunci berukuran 168-bit (3 kali kunci 56-bit dari *DES*), namun dengan metode *meet-in-the-middle* keamanan yang diberikan hanyalah 112-bit. Sebuah varian, *Double TDES*, menggunakan kunci k1=k3, yang berarti mengecilkan ukuran kunci ke 112-bit dan ukuran *storage* menjadi 128-bit. Namun mode ini lemah terhadap beberapa serangan jenis *chose-plaintext* atau *known-plaintext*. Oleh sebab itu, mode ini biasanya hanya didesain dengan keamanan 80-bit.

7.2 Penggunaan *Triple DES* (*TDES*)

Penggunaan *TDES* semakin hari semakin menurun digantikan oleh *Advanced Encryption Standard AES*. Sebuah pengecualian dalam skala besar adalah dalam industri pembayaran elektronik yang masih menggunakan *Double TDES* dan secara ekstensif mengembangaman standardnya. Ini menjamin bahwa *TDES* akan tetap aktif di dunia kriptografi hingga masa yang belum dapat ditentukan.

Secara desain, *DES* dan juga *TDES*, cenderung lambat pada perangkat lunak, pada prosesor modern, *AES* cenderung lebih cepat. *TDE* lebih cocok untuk implementasi perangkat keras, walaupun *AES* masih tetap lebih cepat.

8. *Advanced Encryption Standard* (*AES*)

Pada algoritma AES, jumlah blok input, blok output, dan *state* adalah 128 bit. Dengan besar data 128 bit, berarti *Nb* = 4 yang menunjukkan panjang data tiap baris adalah 4 byte. Dengan blok input atau blok data sebesar 128 bit, *key* yang digunakan pada algoritma AES tidak harus mempunyai besar yang sama dengan blok input. *Cipher key* pada algoritma AES bisa menggunakan kunci dengan panjang 128 bit, 192 bit, atau 256 bit. Perbedaan panjang kunci akan mempengaruhi jumlah *round* yang akan diimplementasikan pada algoritma AES ini. Di bawah ini adalah tabel yang memperlihatkan jumlah *round* (*Nr*) yang harus diimplementasikan pada masing-masing panjang kunci.

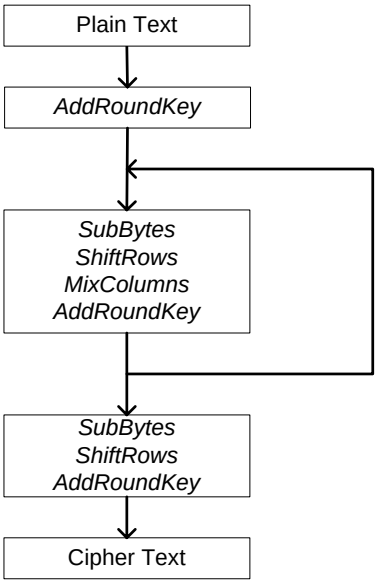
Tabel 1 Perbandingan jumlah Round dan Key

Varian *AES*	Panjang Kunci (*Nk words*)	Ukuran Blok (*Nb words*)	Jumlah Putaran (*Nr*)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

8. 1 Enkripsi Adv*anced Encryption Standard* (*AES*)

Proses enkripsi pada algoritma AES terdiri dari 4 jenis transformasi bytes, yaitu SubBytes, ShiftRows, Mixcolumns, dan AddRoundKey. Pada awal proses enkripsi, input yang telah dikopikan ke dalam *state* akan mengalami transformasi byte AddRoundKey. Setelah itu, *state* akan mengalami transformasi SubBytes, ShiftRows, MixColumns, dan AddRoundKey secara berulang-ulang sebanyak *Nr*. Proses ini dalam algoritma AES disebut sebagai *round function*. *Round* yang terakhir agak berbeda dengan *round-round* sebelumnya dimana pada *round* terakhir, *state* tidak mengalami transformasi MixColumns.

Gambar 9 Diagram Alir Proses Enkripsi



DAFTAR PUSTAKA

[1] *Vocal Technologies Ltd. Triple DES* http://csrc.nist.gov/cryptval/des.htm Tanggal akses: 2 Januari 2007 pukul 22:00.

[2] *The Rijndael Page (old version) http://www.iaik.tu-graz.ac.at/research/krypto/AES/old/%7Erijmen/rijndael/* Tanggal akses:2 Januari 2007 pukul 22:00

[3] Munir, Rinaldi. (2004). Bahan Kuliah IF5054 Kriptografi. Departemen Teknik Informatika, Institut Teknologi Bandung.

[4] *What is a block cipher http://www.rsasecurity.com/rsalabs/faq/2-1-4.html* Tanggal akses:2 Januari 2007 pukul 22:00

[5] *Coppersmith, D. (May 1994). "The Data Encryption Standard (DES) and its strength against attacks" (PDF).* IBM Journal of Research and Development ***38*** *(3): 243.*

[6] *Triple DES Encryption*
http://www.tropsoft.com/strongenc/des.htm
Tanggal akses:2 Januari 2007 pukul 22:00

[7] http://en.wikipedia.org/