STUDI PERBANDINGAN KRIPTOGRAPI MENGGUNAKAN METODE DES, TRIPLE DES DAN RSA

Amat Suroso

Program Studi Manajemen Informatika Sekolah Tinggi Manajemen Informatika dan Komputer
Bani Saleh
suroso@banisaleh.ac.id

Disetujui, 14 Februari 2018

Abstrak

Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, serta otentikasi. Algoritma RSA dan Triple DES adalah dua metode yang digunakan untuk proses enkripsidan dekripsi pada tugas akhir ini. Proses enkripsi dan dekripsi dengan ke dua algoritma tersebut digunakan pada enkripsi dan dekripsi file teks. Pada dasarnya ke dua algoritma ini berbeda berdasarkan kesamaan kuncinya. Pada algoritma RSA, menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsinya. Sedangkan Triple DES setiap proses enkripsi maupun dekripsi data secara keseluruhan digunakan kunci yang sama. Pada tugas akhir ini dibuat perangkat lunak yang menggunakan bahasa pemrograman Visual Basic 6.0 untuk membandingkan ke dua algoritma tersebut. Perbandingan dilakukan dalam hal lama proses dekripsi antara algoritma RSA dan Triple DES.

Kunci : Kriptografi, Algoritma RSA, Triple DES

Abstract

Cryptography is the study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, and authentication. RSA and Triple DES algorithms are two methods used for the encryption process and decryption in this thesis. Encryption and decryption process with these two algorithms used to encrypt and decrypt the text files. Basically to have two different algorithms based on common key. In the RSA algorithm, using the key different to the encryption and decryption process. While Triple DES encryption and decryption of each process overall data use the same key. In this final project created software using Visual Basic 6.0 programming language to compare to these two algorithms. The comparison is done in terms of long decryption process between RSA and Triple DES algorithms.

Keywords: Cryptography, RSA algorithm, Triple DES

1. Pendahuluan

Komputer sebagai *general purpose machine*, selama bertahun-tahun telah dimanfaatkan manusia untuk membantu berbagai pekerjaan. Komputer telah digunakan secara luas sebagai alat hitung, mesin ketik, mesin gambar, alat komunikasi pada jaringan global (*internet*) bahkan sampai pengontrol reaktor nuklir.

Dalam perkembangannya, ketika komputer hanya dipakai sebagai alat penelitian di laboratoriom-laboratoriom perguruan tinggi, sebagai mesin ketik dirumah-rumah, atau sekedar untuk bermain *game*, masalah keamanan data belum diperhatikan. Akan tetapi ketika komputer telah dipakai secara luas dan global seperti : perbankan, perbelanjaan *online*, ataupun untuk alat komunikasi melalui internet maka persoalan keamanan dan kerahasiaan data menjadi hal yang penting. Perlindungan terhadap kerahasiaan dan keamanan data pun meningkat. Akhirnya orang-orang pun mengembangkan berbagai cara untuk mengatasi persoalan keamanan data yang pada intinya adalah bagaimana agar orang-orang yang tidak berhak tidak mungkin dapat membaca atau bahkan merusak data yang bukan ditujukan kepadanya,

Untuk mengatasi masalah ini digunakanlah suatu teknik penyandian data yang dikenal dengan nama kriptografi. Dengan teknik ini maka pesan akan dienkripsi (diubah menjadi kode/sandi yang

tidak dapat dimengerti oleh orang lain) terlebih dahulu menjadi pesan rahasia untuk kemudian baru dikirimkan kepada orang yang diinginkan. Pesan sandi yang diterima penerima akan diubah kembali menjadi pesan asli yang mudah dipahami seperti pesan aslinya.

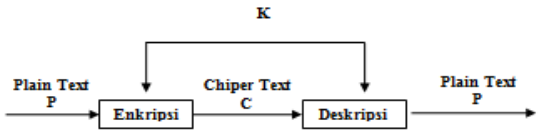
Enkripsi pada saat ini telah berkembang pesat, mencari konfigurasi yang terbaik. Dari sekian banyak algoritma enkripsi yang ada, algoritma DES, algoritma Triple DES dan algoritma RSA yang banyak diimplentasikan dalam aplikasi enkripsi data.

2. Metodologi Penelitian

2.1 Terminologi Kriptografi

Kriptografi (*cryptography*) adalah studi mengenai metode penyandian pesan yang bertujuan untuk menghindari perolehan pesan secara tidak sah. Ditinjau dari terminologinya, kata kriptografi berasal dari bahasa Yunani yaitu kryptos, 'menyembunyikan', dan graphein 'menulis', sehingga dapat didefinisikan sebagai ilmu yang mengubah informasi dari keadaan/bentuk normal (dapat dipahami) menjadi bentuk yang tidak dapat dipahami.

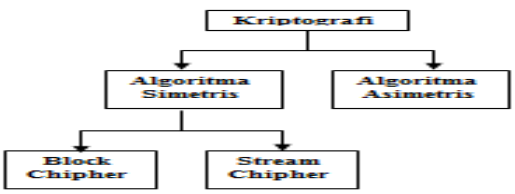
Algoritma Kriptografi selalu terdiri dari dua bagian, yaitu enkripsi dan dekripsi. **Enkripsi** (encryption) merupakan proses yang dilakukan untuk mengubah pesan yang tidak disandikan (plaintext atau cleartext) ke dalam bentuk yang tidak dapat dibaca (ciphertext) Sedangkan **dekripsi** (decryption) adalah proses kebalikannya. Proses enkripsi dan dekripsi diatur oleh satu atau beberapa kunci kriptografi. Secara sederhana proses kriptografi dapat digambarkan sebagai berikut:



Gambar 1. Proses Enkripsi

2.2 Algoritma Kriptografi Berdasakan Key

Terdapat dua macam algoritma kriptografi yaitu algortima ***Public Key*** *(Asymetric)* dan ***Secret Key*** *(Simetric).* Algoritma simetrik adalah sendiri terdiri atas *block chipper* dan *stream chipper*. Gambar di bawah mempersentasikan jenis Kriptografi berdasarkan key yang dipakai.



Gambar 2. Kriptografi berdasarkan key

2.2.1. Algoritma Simetrik

Key yang digunakan pada algoritma ini, antara pengirim dan penerima adalah sama, sedangkan proses yang dilakukan untuk dekripsinya melakukan kebalikan dari proses enkripsi. Kekuatan algoritma simetrik sangat bergantung pada satu key yang digunakan. Jika key dapat dikirimkan secara aman maka kemungkinan mendapatkan *plaintext* dan *chipertext* yang dikirimkan akan semakin kecil. algoritma Simetrik memiliki dua tipe dasar yaitu *Block Chiper* dan *Stream Chiper*. Dengan menggunakan *block chiper* , *plaintext chiper* yang sama dengan key yang sama akan dienkripsikan ke dalam *Chipertext Block* yang sama. Pada *Stream Chiper* , *Plaintext* atau byte yang sama akan dienkripsikan ke dalam bit yang berbeda setiap enkripsinya.

a. Stream Cipher

Stream Chiper melakukan pengkodean 1 bit atau byte dalam satu kali prosesnya *Stream Chiper* lebih muda diimplementasikan dalam hardware. Hardware bekerja berdasarkan bit-bit yang merupakan satuan terkecilnya dalam melakukan proses perhitungannya. Yang temasuk algoritma kunci simetris stream chipper adalah OTP, A5 dan RC4.

b. Block Chiper

Block Chiper melakukan pengkodean 1 block dalam sekali proses. Ukuran block ini sendiri dapat ditentukan sesuai keinginan. Namun dalam prakteknya ukuran block yang digunakan memenuhi rumus 2^n dengan n bilangan integer. Yang termasuk algoritma kunci simetris block chipher adalah DES, AES, Blowfish, IDEA, Triple DES.

2.2.2.Algoritma Asimetrik

Public Key menggunakan dua key yang berbeda dalam melakukan proses enkripsi dan dekripsi. *Public Key* yang digunakan untuk melakukan enkripsi dan boleh diketahui umum. Sedangkan *Private Key* hanya boleh diketahui oleh pihak penerima. Yang termasuk algoritma kunci Asimetris adalah RSA, Diffie Hillman (DH), Quantum, ECC dan DSA.

2.3 Kriteria Algoritma Kriptografi

Kriteria standar yang harus dimiliki oleh suatu algoritma Kriptografi menurut *National Bureau Of Standarts* (NBS), sekarang dinamakan *National Institute of Standarts and Tehcnology* (NIST), adalah

- Algoritma harus memiliki tingkat keamanan yang tinggi.
- Algoritma harus spesifik dan mudah dimengerti

- Tingkat keamanan algoritma harus terletak pada key, bukan dari kerahasiaan algoritmanya
- Algoritma harus dapat diadaptasi pada aplikasi yang beragam
- Algoritma harus ekonomis dalam pengimplementasiannya pada perangkat keras.
- Algoritma harus efisien dalam penggunaannya
- Algortima harus dapat berlaku secara umum
- Algoritma dapat dipisahkan (tidak bergantung)

2.4 Authentication, Integrity, and NonRepudiation

Sebagai tambahan dalam menyediakan kerahasiaan, kriptografi juga sering melakukan pekerjaan yang lain, diantaranya:

a. Autentication
Mengidentifikasikan keaslian suatu pesan dan memberikan jaminan keotentikannya serta untuk menguji identitas seseorang apabila ia akan memasuki sebuah sistem

b. Integrity
Memberikan jaminan bahwa untuk tiap bagian pesan tidak akan mengalami perubahan dari saat data dibuat/dikirim oleh pengirim sampai dengan saat data tersebut dibuka oleh penerima data

c. NonRepudiation
Memberikan cara untuk membuktikan bahwa suatu dokumen datang dari seseorang tertentu sehingga apabila ada seseorang yang mencoba mengakui memiliki dokumen tersebut, dapat dibuktikan kebenarannya dari pengakuan orang tersebut.

2.5 Teori Bilangan

Teori bilangan banyak digunakan dalam teknik kriptografi, teori bilangan yang berhubungan dengan skripsi ini, diantaranya :

a. Aritmatika Modulo
Misalkan *a* adalah bilangan bulat dan *m* adalah bilangan bulat > 0. Operasi *a* mod *m* (dibaca "*a* modulo *m*") memberikan sisa jika *a* dibagi dengan *m*.
Notasi: $a \bmod m = r$ sedemikian sehingga $a = mq + r$, dengan $0 \le r < m$.
Bilangan *m* disebut **modulus** atau **modulo**, dan hasil aritmetika modulo *m* terletak di dalam himpunan $\{0, 1, 2, \ldots, m-1\}$

b. Bilangan Prima
Bilangan bulat positif p ($p > 1$) disebut bilangan prima jika pembaginya hanya 1 dan p.
Contoh: 23 adalah bilangan prima karena ia hanya habis dibagi oleh 1 dan 23.

Karena bilangan prima harus lebih besar dari 1, maka barisan bilangan prima dimulai dari 2, yaitu 2, 3, 5, 7, 11, 13, …. Seluruh bilangan prima adalah bilangan ganjil, kecuali 2 yang merupakan bilangan genap.

Bilangan selain prima disebut bilangan **komposit** (*composite*). Misalnya 20 adalah bilangan komposit karena 20 dapat dibagi oleh 2, 4, 5, dan 10, selain 1 dan 20 sendiri.

Teorema. (The Fundamental Theorem of Arithmetic). Setiap bilangan bulat positif yang lebih besar atau sama dengan 2 dapat dinyatakan sebagai perkalian satu atau lebih bilangan prima

c. **GCD (*Greatest Common Divisor) atau Pembagi Bersama Terbesar (PBB)***
Dengan notasi matematika
Gcd (a,b) = d
Misalkan *a* dan *b* adalah dua buah bilangan bulat tidak nol. Pembagi bersama terbesar (PBB – **greatest common divisor** atau gcd) dari *a* dan *b* adalah bilangan bulat terbesar *d* sedemikian sehingga $d \mid a$ dan $d \mid b$. Dalam hal ini kita nyatakan bahwa PBB(a, b) = d.

d. Relatif Prima
Relatif Prima adalah jika dua buah bilangan bulat a dan b (baik prima ataupun tidak prima) dalam GCD bernilai sama dengan 1.
Dalam notasi matematika ***GCD (a, b) = 1.***

e. Euler Phi
Φ (n) (dibaca fee dari n) sebagai himpunan bilangan positif yang < dari n dan relative prima terhadap n. ingat disebut relative prima jika mempunyai GCD = 1.

f. Teori Euclidean
Digunakan untuk menghitung GCD antara 2 bilangan bulat a dan b (bukan nol) dengan syarat jika r adalah **sisa dari a dibagi dengan b** kemudian GCD (a,b) = GCD (b,r). Proses ini terus berulang sampai tidak ada sisa sama sekali
Rumus : GCD (a,b) =GCD(b,r_1) = GCD (r_1,r_2) = … = GCD (r_{n-1},r_n)

g. **Teori Fiestel**
Teori Fiestel disebut juga dengan fungsi f, dimana berisi expansion, kotak –S dan permutasi yang dikombinasikan dengan kunci Ki
Rumus : $L_n = R_{n-1}$
$R_n = L_{n-1}$ XOR f (R_{n-1} , K_n)

h. **Teori XOR**
Memberikan hasil true jika salah satu operand nya adalah true tetapi tidak jika keduanya adalah true

Tabel 1. Tabel Kebenaran Operator XOR

A	B	A XOR B
True	True	False
True	False	True
False	True	True
False	False	True

3. Metode Penelitiam

3.1. Kompleksitas Algoritma

3.1.1. Kompleksitas Waktu

3.1.2.

Proses dari suatu algoritma didalam mencari solusi dari suatu masalah memerlukan waktu tertentu. Satuan waktu yang dibutuhkan diharapkan dalam waktu yang relative singkat (efesien). Adapun hal-hal yang mempengaruhi waktu tempuh tersebut adalah (Suryadi, 1996:9):

1. Banyak langkah, makin banyak langkah atau instruksi yang digunakan maka makin lama waktu tempuh yang dibutuhkan dalam proses tersebut.
2. Besar dan jenis input data, ukuran atau besar serta jenis input yang digunakan akan sangat berpengaruh pada proses perhitungan. Jika jenis data yang diinginkan adalah dengan tingkat ketelitian tunggal (single precision) maka waktu tempuh relative lebih cepat apabila digunakan data dengan tingkat ketelitian ganda (double precisian)
3. Jenis operasi, waktu tempuh juga dipengaruhi oleh jenis operasi yang digunakan. Jenis operasi tersebut meliputi operasi aritmatik, operasi nalar atau logika dan lain-lain.
4. computer atau kompilator. Faktor ini diluar dari rancangan atau pembuatan algoritma yang efesien walaupun algoritma yang dibuat sudah mencapai waktu tempuh yang sangat efesien (dengan memperhatikan ketiga hal tersebut diatas). Namun bila digunakan computer yang berkemampuan lambat, maka waktu tempuhnya akan menjadi lebih lambat. Kompilator yang digunakan juga akan berpengaruh terhadap waktu tempuh suatu algoritma

Kompleksitas algoritma, T (n) diukur dari jumlah tahapan komputasi yang dibutuhkan untuk menjalani algoritma sebagai fungsi dari ukuran masukan n. Dalam praktek, kompleksitas waktu dihitung berdasarkan jumlah operasi abstrak yang mendasari suatu algoritma dan memisahkan analisisnya dari implementasi

Kompleksitas waktu dari suatu fungsi polynomial F (N) dengan N input dapat dibedakan dengan 3 keadaan yaitu : (Suryadi, 1996:11)

a. Worst Case
   Suatu keadaan yang analog atau merupakan nilai maximal dari fungsi F(N) untuk setiap input yang mungkin. Dengan perkataan lain, hal ini merupakan suatu keadaan yang "terburuk" dari proses didalam suatu algoritma, sehingga waktu tempuh oleh algoritma tersebut adalah waktu maximal
b. Average Case
   Hal ini merupakan suatu keadaan dari waktu tempuh yang ekuivalen dengan nilai ekspektasi dari fungsi F(N) untuk setiap input data yang mungkin. Nilai ekspekstasi dari fungsi F(N)=E yang didefinisikan sebagai berikut:
   E=n1p1+n2p2+…+.nnpn
   Dengan :
   N1,n2,n3…. :Merupakan nilai-nilai yang muncul
   P1,p2,p3 … : Merupakan probabilitas dari setiap nilai (ni) yang muncul
c. Best Case
   Suatu keadaan yang analog atau merupakan nilai minimum dari fungsi F(N) untuk setiap input yang mungkin. Dengan perkataan lain hal ini merupakan suatu keadaan yang "terbaik" dari proses didalam suatu algoritma. Dengan demikian waktu yang ditempuh oleh algoritma tersebut adalah waktu yang minimum.
   Menghitung O Besar untuk setiap instruksi didalam algoritma kemudian menerapkan teorema O Besar sebagai berikut:

a. Pengisian nilai (assignment) perbandingan, operasi aritmatik membutuhkan waktu O(1)
b. Pengaksesan elemen larik atau memilih field tertentu dari sebuah record membutuhkan O(1)
c. If C then S1 else S2, membutuhkan waktu Tc + Max (Ts1,Ts2)
d. Kalang for. Kompleksitas waktu kalang for adalah jumlah pengulangan dikali dengan kompleksitas waktu badan (body) kalang.
e. While C do S, dan repeat S until C, untuk kedua buah kalang kompleksitas waktu badan C dan S
f. Prosedure dan fungsi, waktu yang dibutuhkan untuk memindahkan kendali kerutin yang dipanggil adalah O(1)

Kompleksitas waktu algoritma dapat diketahui dari perulangan berderet yang dieksekusi |n|-1 kali dan karena masing-masing operasi perulangan menyokong O (n Log n) kepada waktu proses. Jadi total waktu proses algoritma RSA satu set n karakter adalah O(n log n).berarti bahwa waktu pelaksanaan n log-n terdapat pada algoritma yang memecahkan persoalan menjadi beberapa persoalan yang lebih kecil, menyelesaikan tiap persoalan menjadi secara independent dan menggabung solusi masing-masing persoalan. Bila n = 1000 maka n log n menjadi 20.000. Bila n dijadikan dua kali semua maka n log n menjadi dua kali semua (tetapi tidak terlalu banyak)

Sedangkan algoritma DES dan 3DES kompleksitas algoritmanya adalah O(log n) yang

berarti bahwa laju pertumbuhan waktunya berjalan lebih lambat dari pada pertumbuhan n algoritma ini memecahkan persoalan besar dengan mentrasformasikan menjadi beberapa persoalan yang lebih kecil yang berukuran sama (Thomas H. Cormen,dkk,1990:340).

3.1.3. Kompleksitas Ruang

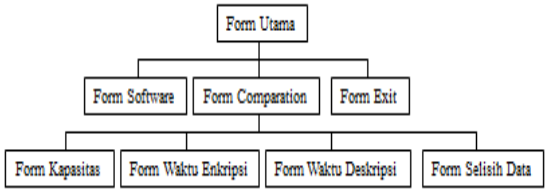
Banyak langkah yang digunakan dan jenis variable atau data yang dipakai dalam suatu algoritma akan mempengaruhi penggunaan memori. Dalam hal ini diharapkan dapat memperkirakan seberapa banyak kebutuhan mmemory yang diperlukan selama proses berlangsung hingga diperoleh penyelesaiannya.. dengan demikian dapat disiapkan storage yang memadai agar proses dari suatu algoritma berlangsung tanpa ada hambatan (kekurangan memory).kompleksitas ruang, S(n) diukur dengan memory yang digunakan oleh struktur data yang terdapat didalam algoritma sebagai fungsi dari ukuran masukan n (suryadi,1996:12)

Dengan menggunakan besaran kompleksitas waktu dan ruang algoritma dapat menetukan laju peningkatan waktu (ruang) yang diperlukan algoritma dengan meningkatnya ukuran masukan n.

Ditentukan dengan kekuatan perhitungan yang diperlukan untuk mengeksekusinya. Perhitungan kompleksitas ditentukan berdasarkan dua variable *T* (*Time*) waktu dan *S* (*Space*) ruang memory yang dibutuhkan. Jika waktu yang diperlukan untuk melakukan proses selalu tetap maka kompleksitasnya dikatakan konstan. Bila waktu berbanding lurus dengan proses dikatakan linear.

3.2. Rancangan Perangkat Lunak

Untuk dapat mengetahui kehandalan ketiga algoritma enkripsi, data yang peneliti bandingkan maka perlu diimplementasikan pada suatu perangkat lunak. Dalam implementasi tersebut peneliti merancang dengan antarmuka sebagai berikut:



Gambar 3. Rancangan Antarmuka

4. Hasil Dan Pembahasan

4.1. Prosedur Operasional

Dalam skripsi ini peneliti melakukan penelitian terhadap algoritma enkripsi data yaitu : DES, Triple DES, dan RSA yang diimplementasikan dengan program Microsoft Visual Basic 6.0 untuk mengetahui perbedaan kapasitas data hasil enkripsi, waktu proses dan selisih data setelah deskripsi. Perangkat lunak hasil penelitian ini dapat digunakan untuk mengenkripsi data atau file dengan berbagai extension

Dalam pengoperasiannya, perangkat lunak ini dapat mengenkripsi data yaitu dengan memilih data yang akan dienkripsi kemudian memilih algoritma enkripsi data dan selanjutnya memasukan key, kemudian tekan tombol enkrip, data akan diproses untuk dienkripsi, setelah proses selesai, akan diketahui kapasitas data setelah enkripsi dan waktu proses enkripsi.

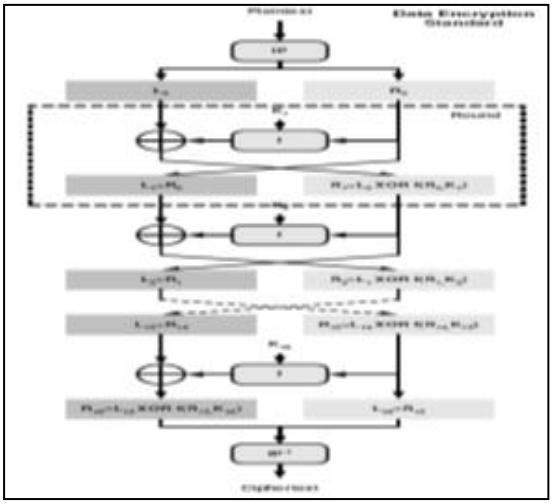
Untuk mengembalikan data kebentuk semula dilakukan dengan cara memilih data hasil enkripsi, memasukan key, kemudian dilakukan deskripsi, maka akan diketahui lama waktu proses deskripsi dan selisih data antara data sebelum enkripsi dan data setelah deskripsi kebentuk semula. Untuk mengetahui perbandingan algoritma enkripsi dalam mengenkripsi data maka dapat dilihat dalam grafik.

4.2. Metode Enkripsi

4.2.1. Algoritma DES

Berikut adalah skema global dari algoritma DES:

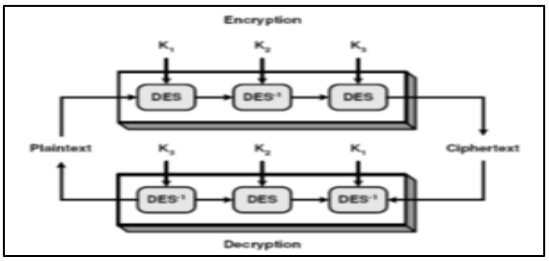
1. Blok plainteks dipermutasi dengan matriks permutasi awal (initial permutation)
2. Hasil permutasi awal kemudian dienchipering sebanyak 16 kali (16 putaran). Setiap putaran menggunakan kunci internal yang berbeda.
3. Hasil enciphering kemudian dipermutasi dengan matriks permutasi balikan (invers initial permutation) menjadi blok cipherteks.



Gambar 4. Aalgoritma DES

4.2.2. Algoritma Triple DES

Seperti yang sudah dijelaskan bahwa Triple DES menggunakan panjang kunci tiga kali lebih panjang dari DES itu sendiri. Masing –masing kunci digunakan untuk mengenkripsi cipher teks kembali sampai tiga kali



Gambar 5. Triple des Algoritma

4.2.3. Algoritma RSA

Dalam algoritma RSA terbagi dua bagian proses yaitu : Key generation dan Enkripsi.

Key generation :

1) Hasilkan dua buah integer prima besar, p dan q, untuk memperoleh tingkat keamanan yang tinggi pilih p dan q yang berukuran besar, misalnya 1024 bit.
2) Hitung phi = (p-1)*(q-1)
3) Hitung n = p*q
4) Pilih d yg relatively prime terhadap m, e relatively prime thd m artinya faktor pembagi terbesar keduanya adalah 1, secara matematis disebut gcd (e, phi) = 1. Untuk mencarinya dapat digunakan algoritma Euclid.
5) Cari d, sehingga e*d = 1 mod (m), atau d = (1+nm)/e, untuk bilangan besar, dapat digunakan algoritma extended Euclid.
6) Kunci publik : e, n
7) Kunci private : d, n

Public key encryption

B mengenkripsi message M untuk A

Yg harus dilakukan B :

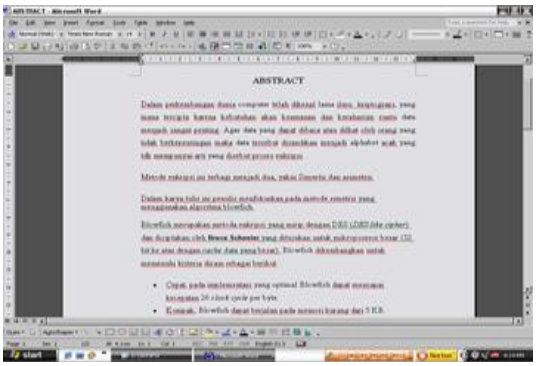
1) Ambil kunci publik A yg otentik (n, e)
2) Representasikan message sbg integer M dalam interval [0,n-1]
3) Hitung C = M ^ e (mod n)
4) Kirim C ke A

Untuk mendekripsi, A melakukan :

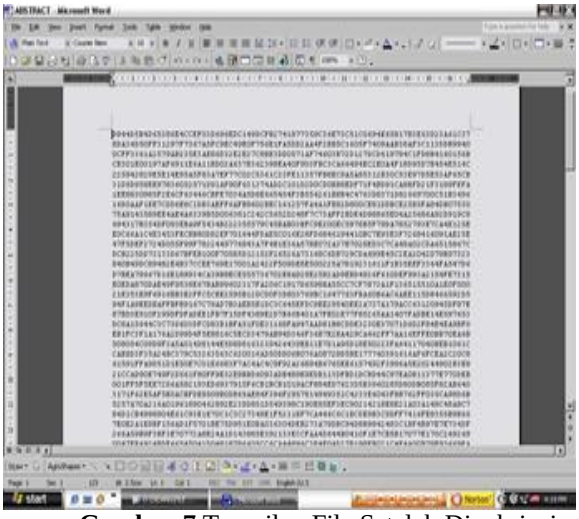
Gunakan kunci pribadi d untuk menghasilkan M = C^(d) (mod n)

4.3. Implementasi Perangkat Lunak

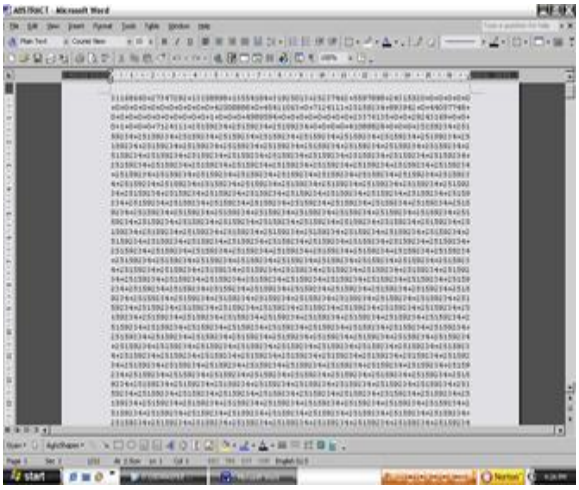
1) Uji 1 File Text (Abstrak.doc)



Gambar 6. Tampilan File Asli Abstrac.doc



Gambar 7. Tampilan File Setelah Dienkripsi DES



Gambar 8. Tampilan File Setelah Dienkripsi RSA

Tabel 2. Uji File Text

Algoritma / Tolok ukur	Kapasitas Awal	DES	3DES	RSA
Kapasitas	26.624 Kb	200%	200%	341%
Waktu Enkripsi		0.15625 sec	0.09375 sec	4.40991 sec
Waktu Deskripsi		0.1563 sec	0 sec	9.60938 sec
Selisih Kapasitas		0 byte	0 byte	0 byte

Berdasarkan table diatas dapat diketahui bahwa file uji1 (26.624 Kb) adalah file dokumen dari hasil pengetikan dengan menggunakan Microsoft Office 2000, setelah dienkripsi ternyata algoritma DES mampu mengenkripsi data sebesar 200 % (53.2646 Kb) sedangkan untuk algoritma Triple DES sebesar 200 % (53.2646 Kb) dan algoritma RSA sebesar 341 % (90.991 Kb)

Waktu yang dibutuhkan algoritma DES untuk melakukan proses enkrip data uji1.doc sangat singkat yaitu kurang dari 1 detik (0.15625 second) dan algoritma Triple DES membutuhkan waktu

kurang dari 1 detik (0.09375 second), sedangkan algoritma RSA mencapai 4 detik (4.40991 sec)

Sementara itu waktu deskripsi yang dibutuhkan algoritma DES kurang dari 1 detik (0.1563 sec) disusul algoritma Triple DES 0 detik dan algortima RSA 9 detik (9.60938 sec), hal ini dikarenakan algoritma RSA memilki kompleksitas algoritma yang lebih rumit dibandingkan kedua algoritma tersebut

Hasil deskripsi data menunjukkan bahwa tidak terdapat selisih kapasitas data antara kapasitas data sebelum dengan kapasitas data setelah deskripsi.

2) Uji 2 File Multimedia (shot02.mp3)

Gambar 9. Tampilan File Asli shot02.mp3 dengan Notepad

Gambar 10. Tampilan File setelah Dienkripsi DES

Gambar 11. Tampilan File Setelah Dienkripsi RSA

Tabel 3. Uji2 File Multimedia

Algoritma / Tolok ukur	Kapasitas Awal	DES	3DES	RSA
Kapasitas	12.436 Kb	200 %	200 %	743 %
Waktu Enkripsi		0.01563 sec	0.03125 sec	1.9375 sec
Waktu Deskripsi		0 .01563sec	0.01563 sec	4.20313 sec
Selisih Kapasitas		0 byte	0 byte	0 byte

Berdasarkan table diatas dapat diketahui bahwa file uji2 (12.436 Kb) adalah file multimedia dengan extention .mp3, setelah dienkripsi ternyata algoritma DES mampu mengenkripsi data sebesar 200 % (24.88 Kb) sedangkan untuk algoritma Triple DES sebesar 200 % (24.88 Kb) dan algoritma RSA sebesar 743 % (92.495 Kb)

Waktu yang dibutuhkan algoritma DES untuk melakukan proses enkrip data uji2.mp3 sangat singkat yaitu kurang dari 1 detik (0.15625 second) dan algoritma Triple DES membutuhkan waktu kurang dari 1 detik (0.09375 second), sedangkan algoritma RSA mencapai 4 detik (4.40991 sec)

Sementara itu waktu deskripsi yang dibutuhkan algoritma DES kurang dari 1 detik (0.1563 sec) disusul algoritma 3DES 0 detik dan algortima RSA 9 detik (9.60938 sec), Hasil deskripsi data menunjukkan bahwa tidak terdapat selisih kapasitas data antara kapasitas data sebelum dengan kapasitas data setelah deskripsi.

4.4. Pengujian Komplesitas Waktu Algoritma (Time Complexity Algoritm)

Kompleksitas algoritma DES dan algortima Triple DES adalah O (log n) yang berarti bahwa laju pertumbuhan waktunya berjalan lebih lambat dari pada pertumbuhan n, sedangkan pada algoritma RSA kompleksitas waktunya adalah O (n log n) yang berarti bahwa waktu pelaksanaan n log n dengan pemecahan persoalan secara independent dan menggabungkan solusi masing-masing persoalan, untuk itu dalam algoritma RSA memerlukan waktu yang lebih banyak dibandingkan kedua algoritma yang lain

Setelah dilakukan penghitungan dengan data masukan n = 5 karakter , n =250 karakter, n= 1000 karakter, yang dapat diuraikan sebagai berikut :

n	Algoritma					
	DES		Triple DES		RSA	
	Big O	Uji 1	Big O	Uji 2	Big O	Uji3
5	0.6990	0,32813	0.6990	0.4688	3.4949	2.9375
250	2.3979	0,4688	2.3979	0.625	599.4850	2.984375
1000	3.0000	0,625	3.0000	0.7813	3.000.0000	4.15625

Berdasarkan tabel diatas dapat diketahui bahwa pada algoritma DES dengan data masukan n = 5, n = 250, dan n = 1000 (berbeda-beda) membutuhkan waktu pelaksaan ekripsi yang tetap yaitu dibawah 0 detik, hal ini sesuai dengan kompleksitas waktu dengan dengan menggunakan big O yaitu O (1). Kompleksitas waktu algoritma 3DES dengan data masukan n= 5 menghasilkan 0.6990, n = 250 menghasilkan 2.3979 dan n = 1000 mengasilkan 3.0000. hal ini dapat dibuktikan dengan melakukan pengujian yaitu waktu pelaksanaan enkripsi berjalan lebih lambaat dari pada pertumbuhan n.

Sedangkan kompleksitas algoritma RSA yang dicari dengan menggunakan big O dengan masukan n = 5 menghasilkan 3,4949, n = 250 menghasilkan 599,4850 dan n = 1000 menghasilkan 3.000.000. Hasil pengujian waktu pelaksanaan enkripsi untuk ketiga data masukan tersebut yaitu sebanding dengan kompleksitas waktu algoritma RSA yaitu O (nlog n) yang berati bahwa semakin besar data masukan yang diberikan akan membutuhkan waktu pelaksanaan yang semakin banyak atau dapat dikatakan data masukan berbanding lurus dengan waktu pelaksanaan enkripsi

5. Penutup

4.1. Kesimpulan

4.2.

Penelitian ini dilakukan dengan tujuan untuk mengetahui perbandingan algoritma DES, Triple DES dan RSA dalam mengenkripsi data yaitu dengan tolok ukur kapasitas waktu proses enkripsi, waktu proses deskripsi dan selisih kapasitas data antara sebelum dan setelah desripsi.

Berdasarkan hasil pengujian terhadap ketiga algoritma tersebut dan analisis yang telah diuraikan pada bab– bab sebelumnya maka dapat disimpulkan sebagai berikut :

1) Proses enkripsi menggunakan metode DES, TripleDES dan RSA akan menambah kapasitas file tersebut dikarenakan perubahan struktur variable dan maximum key.
2) Untuk algoritma RSA semakin besar kapasitas file asal maka akan semakin lama proses enkripsi dan deskripsi serta semakin besar kapasitas file hasil enkripsi.

Hasil pengujian pada perangkat lunak untuk ketiga algoritma enkripsi data yaitu DES, Triple DES dan RSA dari penelitian ini belum sepenuhnya dapat dijadikan parameter yang menunjukkan ratio enkripsi data untuk ketiga algoritma tersebut, selain tolok ukur kapasitas data , waktu proses enkripsi dan deskripsi, serta selisih data masih banyak faktor –faktor lain yang bisa dijadikan tolok ukur untuk mengukur ratio enkripsi ketiga algoritma.

Kondisi tersebut terlihat dari perbedaan ratio waktu proses enkripsi dan deskripsi pada masing-masing komputer dikarenakan spesifikasi perangkat keras dan perangkat lunak yang digunakan berbeda. Oleh sebab itu keterbatasan dalam menentukan tolok ukur yang dijadikan perbandingan terhadap kehandalan ketiga algoritma tersebut menjadi salah satu keterbatasan dari penelitian ini.

Selain dari sisi metode yang digunakan penelitian ini juga memiliki beberapa keterbatasan lain yaitu : Pertama, jumlah kasus uji yang diujicobakan, dengan demikian tingkat generalisasi kesimpulan penelitian masih terbatas. Kedua, isi data kasus uji yang digunakan untuk setiap kasus uji belum dapat mewakili isi data yang ada.

4.3. Saran

Adapun saran –saran yang ingin peneliti sampaikan berdasarkan hasil pengujian terhadap ketiga algoritma tersebut adalah sebagai berikut:

1) Untuk mengatasi kapasitas yang membesar setelah enkripsi diperlukan kombinasi dengan metode kompresi.
2) Untuk mengamankan data yang sangat penting bisa menggunakan algoritma RSA walupun lebih lama dan lebih besar kapasitas hasil enkripsi, karena sebanding dengan berharganya data tersebut.
3) Adanya penambahan tolok ukur serta jumlah kasus uji yang diuji cobakan agar penelitian ini dapat lebih akurat

Daftar Pustaka

[1] Anonim. ***Memahami model enkripsi dan Security Data***. Penerbit Andi, Yogyakarta, 2003.

[2] Ariyus, Doni**.** ***Kamus Hacker***. Penerbit Andi,Yogyakarta, 2005.

[3] Wiley, John & Sons. (eBook) Bruce Schneier - ***Applied Cryptography***, Second Edition - [ISBN0471128457].

[4] Hasan, Rusdi. ***Mengenal Algoritma DES***. www.ilmukomputer.com..8 Agustus 2007 17:10 wib.

[5] Kurniawan, Yusuf. ***Kriptografi Keamanan Internet dan Jaringan Komunikasi***. Penerbit Informatika, 2004.

[6] Kusumo Suryo, Ario. ***Kriptografi menggunakan VB.Net.*** www.ilmukomputer.com, Ario Suryo Kusumo, 2003.

[7] Munir, Rinaldi & Lidya Leoni, ***Algoritma & Pemograman*** , Informatika Bandung, 2002.

[8] Peleeger, Charles & Peleeger, Shari , ***Security In Computing***, Pearson Education Inc., USA.

[9] Seymour Phd, & Marc Phd. ***Matematika Diskrit***, Salemba Teknika ,2001.

[10] Stalling, William. ***Crypthography and Network Security***. Prentice-hall, 2003.

[11] Stiawan, Deris. ***Sistem Keamanan Komputer***. Elex Media Komputindo, 2005.