JUSIKOM PRIMA (Junal Sistem Informasi Ilmu Komputer Prima)
Vol. 2 No. 1, Juli 2018 e-ISSN : 2580-2879

ANALISIS KINERJA KOMBINASI ALGORITMA MESSAGE-DIGEST ALGORTIHM 5 (MD5), RIVEST SHAMIR ADLEMAN (RSA) DAN RIVEST CIPHER 4 (RC4) PADA KEAMANAN E-DOKUMEN

Sumarno[1], Indra Gunawan[2], Heru Satria Tambunan[3], Eka Irawan[4]

STIKOM Tunas Bangsa
Jalan Jenderal Sudirman Blok A, No. 1, 2 & 3 Pematangsiantar, 21127
Sumatera Utara-Indonesia
E-mail : sumarno@amiktunasbangsa.ac.id[1], indra@ amiktunasbangsa.ac.id[2], heru@amiktunasbangsa.ac.id[3], eka.irawan@ amiktunasbangsa.ac.id[4]

ABSTRAK

Pertukaran dokumen di dunia maya sudah banyak digunakan dalam transaksi pada saat ini. Dokumen tersebut harus dijamin oleh setiap pengguna. Untuk menjaga kerahasiaan informasi, terutama untuk isi informasi yang seharusnya diketahui hanya pihak berwenang. Pengiriman data atau informasi tanpa jaminan apapun akan beresiko terhadap penyadapan dan informasi di dalamnya dapat dengan mudah diidentifikasi oleh pihak yang tidak berwenang. Salah satu cara untuk mengamankan dokumen adalah dengan menggunakan algoritma kriptografi. Prinsip keamanan dokumen ini adalah bagaimana sistem dapat mengamankan penyimpanan dan pengiriman dokumen dengan menggunakan algoritma kombinasi. Dalam penelitian ini, penggunaan algoritma kombinasi adalah kombinasi algoritma MD5, RSA dan RC4. Tahapan dokumen keamanan mencakup tiga tahap: proses keamanan data, kunci keamanan, dan menguji integritas dari file.
Kata Kunci : Kriptografi, MD5, RSA dan RC4

1. PENDAHULUAN

Kemudahan pengaksesan informasi, baik itu langsung atau tidak langsung, tentunya berdampak pada munculnya resiko dan ancaman keamanan dan integritas data. Ancaman yang diperkirakan akan terjadi adalah akses yang tidak berhak terhadap informasi atau sumber informasi, seperti penggandaan, perubahan atau bahkan perusakan informasi itu sendiri, sehingga membawa kerugian. Untuk itu diperlukan suatu manajemen keamanan yang dapat melindungi atau paling tidak menahan suatu akses yang tidak berhak untuk menjaga data tersebut dalam durasi waktu tertentu.

Pertukaran dokumen berbasis komputer seperti pesan e-mail atau dokumen dalam pesan e-mail di internet sudah luas digunakan sebagai transaksi komersil. Dokumen sering berisi informasi penting seperti kontrak resmi, transaksi keuangan, record penjualan dan lain-lain. Keamanan dari suatu data merupakan hal yang perlu diperhatikan dalam menjaga kerahasiaan data terutama bagi dokumen yang isinya hanya boleh diketahui oleh pihak yang berhak saja. Pengiriman data atau dokumen tanpa dilakukan pengamanan akan beresiko terhadap penyadapan, kerahasiaan dan keautentikan data. Oleh karena itu diperlukan suatu sistem pengamanan data yang bertujuan untuk meningkatkan keamanan data, melindungi suatu data atau pesan agar tidak dibaca oleh pihak yang tidak berwenang, dan mencegah pihak yang tidak berwenang untuk menyisipkan, menghapus, ataupun merubah data.

Kriptografi adalah ilmu tentang mengkonversi atau merubah pesan atau teks asli yang disebut plaintext menjadi untuk pesan atau teks yang diacak disebut cipher teks dengan menggunakan proses enkripsi dan dekripsi.. Sistem ini ditandai dengan tiga tahapan dimensi sebagai jenis operasi yang digunakan untuk mengkonversi plaintext untuk ciphertext, jumlah kunci yang digunakan dan cara di mana plaintext diproses. Dengan kriptografi dimensi ini dibagi dalam dua kategori yaitu sebagai kriptografi Symmetric kunci atau enkripsi konvensional dan kriptografi asimetris atau enkripsi kunci public. (Diffie, W., Hellman, M.E., 1976)

Untuk membangun sistem penyimpanan dokumen yang hasil simpanannya tidak dapat dibaca oleh orang, dalam penelitian ini telah dikembangkan model sistem pengamanan dengan proses enkripsi dan dekripsi dengan metode simetrik kriptosistem dan asimetrik kriptosistem. Gabungan algoritma simetrik kriptosistem dan asimetrik kriptosistem disebut sebagai *hybrid cryptosystem* (Fauziah, 2008).

Penggunaan *hybrid cryptosystem* dalam penelitian ini merupakan gabungan algoritma MD5, RSA dan RC4 yang digunakan dalam tiga tahapan pengamanan data pada dokumen. Adapun tahapan pengamanan data meliputi : proses setup key, proses enkripsi data, dan proses dekripsi data. Maka dari itu, pada penelitian ini penulis akan menggabungkan tiga algoritma sekaligus yaitu algoritma MD5 digunakan sebagai integritas key, RSA digunakan dengan alasan tingkat keamanannya sangat tinggi sebagai pengamanan kunci dan RC4 digunakan sebagai enkripsi sebuah berkas.

2. ISI PENELITIAN

2.1 Dokumen

Dokumen merupakan salah satu hal yang sangat penting karena merupakan sumber informasi

yang diperlukan oleh suatu instansi, organisasi, atau Negara.

2.1.1 Pengertian Dokumen

Dokumen adalah surat penting atau berharga yang sifatnya tertulis atau tercetak yang berfungsi atau dapat di pakai sebagai bukti ataupun keterangan (Alwi, 2000).

2.1.2 Jenis-jenis Dokumen

Menurut Alwi Hasan (2000), ada tiga jenis dokumen yang harus dipahami dan diketahui sebagai berikut :

1. Jenis dokumen dari segi pemakaian
2. Jenis dokumen dari segi fungsinya
3. Jenis dokumen dari segi ruang lingkupnya

2.1.3 E-Dokumen (Elektronik Dokumen)

Menurut undang-undang ITE pasal 1 angka 4, dokumen elektronik (e-dokumen) adalah setiap Informasi Elektronik yang dibuat, diteruskan, dikirimkan, diterima, atau disimpan dalam bentuk analog, digital, elektromagnetik, optikal, atau sejenisnya, yang dapat dilihat, ditampilkan, dan/atau didengar melalui Komputer atau Sistem Elektronik, termasuk tetapi tidak terbatas pada tulisan, suara, gambar, peta, rancangan, foto atau sejenisnya, huruf, tanda, angka, Kode Akses, simbol atau perforasi yang memiliki makan atau arti atau dapat dipahami oleh orang yang mampu memahaminya.

2.2 Konsep Dasar Kriptografi

2.2.1 Pengertian Kriptografi

Kata kriptografi berasal dari bahasa Yunani. Dalam bahasa Yunani kriptografi terdiri dari dua buah kata yaitu *cryptos* dan *graphia*. Kata *crypto* berarti rahasia (*secret*) sedangkan *graphia* berarti tulisan. Sehingga makna dari kriptografi adalah tulisan rahasia. Menurut terminologinya kriptografi adalah ilmu yang mempelajari tentang bagaimana menjaga kerahasiaan suatu pesan, agar isi pesan yang disampaikan tersebut aman sampai ke penerima pesan (Ariyus, 2008).

Dalam ilmu kriptografi suatu pesan yang akan dirahasiakan atau disandikan disebut dengan *plaintext*, sedangkan pesan yang telah disandikan sehingga tidak bermakna lagi yang bertujuan agar pesan tidak dapat dibaca oleh pihak yang tidak berhak disebut *chipertext*. Lalu dalam ilmu kriptografi terdapat istilah enkripsi dan dekripsi. Enkripsi adalah proses menyandikan *plaintext* menjadi *chipertext*. Sedangkan proses mengembalikan *chipertext* menjadi *plaintext* semula disebut sebagai Dekripsi

2.2.2 Tujuan Kriptografi

Aspek-aspek keamanan di dalam kriptografi adalah sebagai berikut (Munir, 2006):

1. Kerahasiaan
   adalah layanan yang digunakan untuk menjaga isi informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka/mengupas informasi yang telah disandi.
2. Integritas data
   adalah berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya.
3. Autentikasi
   adalah berhubungan dengan identifikasi/pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan melalui kanal harus diautentikasi keaslian, isi datanya, waktu pengiriman, dan lain-lain.
4. Non-repudiasi
   adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/terciptanya suatu informasi oleh yang mengirimkan/membuat.

2.2.3 Cryptanalysis

Cryptanalysis adalah usaha-usaha yang dilakukan seseorang untuk memperoleh informasi ataupun data yang telah dienkripsi. Orang yang melakukan kegiatan cryptanalysis disebut dengan *cryptanalyst*.

2.2.4 Cryptosystem dan Hybrid Cryptosystem

Menurut Munir (2006), sistem kriptografi (*cryptosystem*) adalah kumpulan yang terdiri dari algoritma, semua plaintext dan ciphertext. Hybrid *cryptosystem* yaitu kombinasi kriptografi dengan menggabungkan algoritma simetris dan algoritma asimetris atau dengan *public key* dan *private key*(Fauziah, 2008).

2.2.4.1 Symetric Cryptosystem

Dalam *Symetric cryptosystem* ini, kunci yang digunakan untuk proses enkripsi dan dekripsi pada prinsipnya sama tetapi satu buah kunci dapat pula diturunkan dari kunci yang lainnya, kunci-kunci ini harus dirahasiakan. Contoh dari sistem ini adalah Data Encryption Standard (DES), Blowfish, IDEA, RC2, RC3, RC4, dan lain-lain.

Kelebihan *Symetric Cryptosystem* adalah:

1. Proses enkripsi atau dekripsi *Symetric Cryptosystem* membutuhkan waktu yang singkat.
2. Ukuran kunci simetri relatif lebih pendek
3. Otentikasi pengiriman pesan langsung diketahui dari cipherteks yang diterima, karena kunci hanya diketahui oleh penerima dan pengirim saja.

Kekurangan dari *Symetric Cryptosystem* adalah:

1. Kunci simetris harus dikirim melalui saluran komunikasi yang aman dan kedua entitas yang berkomunikasi harus menjaga kerahasian kunci.

2. Kunci harus sering diubah, setiap kali melaksaanakan komunikasi

2.2.4.2 Assymetric Cryptosystem

Dalam *Assymetric Cryptosystem* ini digunakan dua buah kunci yang berbeda dalam proses enkripsi dan dekripsinya. Satu kunci yang disebut kunci publik (public key) yang dapat dipublikasikan, sedangkan kunci yang lain yang disebut kunci privat (private key) yang harus dirahasiakan. Contoh dari sistem ini antara lain RSA Scheme dan Merkie Hellman Scheme.

Assymetric Cryptosystem dikembangkan para pakar kriptografi untuk menanggulangi kesulitan distribusi kunci pada kriptografi kunci simeteri. Distribusi kunci pada kritografi kunci asimetri sangat mudah, karena kunci enkrispsi bersifat *pubik* atau umum maka distribusi kunci dapat dilakukan di jalur mana saja bahkan jalur yang ingin diamankan sekalipun. Terdapat banyak algoritma yang dikembangkan pakar-pakar kriptografi untuk algoritma kunci asimetri, diantaranya : Algoritma RSA, Algoritma McEliece, Algoritma Rabin, Algoritma Knapsack, Algoritma LUC, Algoritma El Gamal.

Kelebihan dari *Assymetric Cryptosystem* adalah:

1. Hanya kunci privat yang perlu dijaga kerahasiannya oleh setiap entitas yang berkomunikasi. Tidak ada kebutuhan mengirim kunci privat sebagaimana pada kunci simetri.
2. Pasangan kunci privat dan kunci public tidak perlu diubah dalam jangka waktu yang sangat lama.
3. Dapat digunakan dalam pengamanan pengirim kunci simetri.
4. Beberapa algoritma kunci public dapat digunakan untuk memberi tanda tangan digital pada pesan

Kelemahan dari *Assymetric Cyptosystem* adalah:

1. Proses enkripsi dan dekripsi umumnya lebih lambat dari algoritma simetri, karena menggunakan bilangan yang lebih besar dan operasi bilangan yang besar.
2. Ukuran cipherteks lebih besar dari pada plainteks.
3. Ukuran kunci relatif lebih besar daripada ukuran kunci simetri.

Untuk saling menutupi kekurangan dari *Symetric Cryptosystem* dan *Assymetric Cryptosystem*, para pakar kriptografi mengembangkan penggabungan dua metode kriptografi ini. Metode penggabungan ini disebut *Hybrid Cryptosystem*. Pada metode ini algoritma kunci simetri digunakan untuk mengamankan pesan atau data yang akan dikirimkan. Sedangkan algoritma kunci asimetri digunakan untuk mengamankan kunci dari proses kriptografi simetri.

Keterangan Gambar:

Dimana kunci yang digunakan untuk proses enkripsi (K1) atau sering disebut dengan public key dan dekripsi (K2) atau sering disebut dengan private key menggunakan kunci yang berbeda.

2.3 Algoritma Message Digest 5 (MD5)

Message Digest 5 (MD5) adalah salah satu dari serangkaian algoritma Message Digest yang didesain oleh Professor Ronald Rivest dari MIT. Saat kerja analitik menunjukkan bahwa pendahulu MD5 yaitu MD4 mulai tidak aman, MD5 kemudian di desain pada tahun 1991 sebagai pengganti dari MD4 dimana kelemahan MD4 telah ditemukan oleh Hans Dobbertin. MD5 banyak digunakan pada bermacam-macam aplikasi termasuk SSL/TLS, IPSec dan protocol-protokol kriptografi lainnya. MD5 juga biasa digunakan pada implementasi *Timestamping Mechanism, Commitment Schemes*, dan aplikasi pengecekan integritas pada *online software*. MD5 tidak memiliki sistim pengamanan seperti persamaan matematika, namun untuk setiap fungsi hash h, domain D dan range R membutuhkan tiga hal berikut:

1. Pre Image Resistance : jika diberi suatu nilai y ε R, maka kita tidak akan dapat mencari suatu nilai x ε D dimana h(x)=y.
2. Second Pre Image Resistance : jika diberi suatu nilai x ε D, maka kita tidak akan dapat mencari nilai x’ ε D dimana h(x)=h(x’).
3. Collision Resistance : kita tidak akan dapat mencari nilai x,x’ ε D dimana h(x)=h(x’).

Algoritma MD5 menerima masukan berupa pesan dengan ukuran sembarang dan menghasilkan message digest yang panjangnya 128 bit. Dengan panjang message digest 128 bit, maka secara brute force dibutuhkan percobaan sebanyak 2128 kali untuk menemukan dua buah pesan atau lebih yang mempunyai message digest yang sama (Rinaldi, 2006).

2.4 Algoritma Rivest Shamir Adleman (RSA)

Algoritma RSA pertama kali ditemukan oleh Ron Rivest, Adi Shamir dan Leonard Aldeman. RSA adalah singkatan dari huruf depan 3 orang yang menemukannya pada tahun 1977 di MIT(*Massachussets Institute of Technology*). Pada tahun 1983, *Massachusets Institute of Technology* menerima hak paten atas sebuah makalah yang berjudul “*Cryptography Communication System and Method*” yang mengaplikasikan pengguna algoritma kriptografi RSA.(Kurniawan, 2004).

Dari sekian banyak algoritma kriptografi kunci-publik yang pernah dibuat, algoritma yang popular adalah algoritma RSA. Langkah dalam algoritma RSA adalah membuat pasangan kunci yaitu Kunci public dan kunci private. Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima. Pemfaktoran dilakukan untuk memperoleh kunci private. Selama pemfaktoran bilangan besar menjadi faktor-faktor prima belum ditemukan algoritma yang

mangkus, maka selama itu pula keamanan algoritma RSA tetap terjamin. (Listiyono, 2009)

Pada algoritma RSA terdapat 3 langkah utama yaitu *keygeneration* (pembangkit kunci), enkripsi dan dekripsi (Munir, 2005). Kunci pada RSA mencakup dua buah kunci, yaitu *public key* dan *private key. Public key* digunakan untuk melakukan enkripsi, dan dapat diketahui oleh orang lain. Sedangkan *private key* tetap dirahasiakan dan digunakan untuk melakukan dekripsi.

Sistem kriptografi yang baik adalah sistem kriptografi yang memang dirancang sedemikian rupa sehingga sulit untuk dipecahkan. Secara teori sebuah metode kriptografi dengan sebuah kunci akan dapat dipecahkan dengan mencoba semua kemungkinan rangkaian kunci. Satu-satunya cara yang diketahui untuk mendobrak sandi RSA adalah dengan mencoba satu persatu berbagai kombinasi kunci dengan istilah *brute force attack*. Sebenarnya keamanan dari RSA banyak bergantung dari ukuran kunci yang digunakan yaitu dalam bit. Jadi semakin panjang ukuran kunci maka semakin sulit untuk dipecahkan.

Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima. Pemfaktoran dilakukan untuk memperoleh kunci privat. Selama pemfaktoran bilangan besar menjadi faktor-faktor prima belum ditemukan algoritma yang mangkus, maka selama itu pula keamanan algoritma RSA tetap terjamin.

Algoritma RSA memiliki besaran-besaran sebagai berikut :

1. p dan q bilangan prima (rahasia)
2. n = p x q (tidak rahasia)
3. $\phi(n) = (p-1)(q-1)$ (rahasia)
4. e (kunci enkripsi) (tidak rahasia)
5. d (kunci dekripsi) (rahasia)
6. m (plainteks) (rahasia)
7. c (cipherteks) (tidak rahasia)

RSA adalah suatu blok sandi rahasia tempat teks asli dan teks rahasia merupakan bilangan bulat antara 0 dan n-1 untuk beberapa n. Enkripsi dan dekripsi berasal dari beberapa bentuk berikut ini, untuk beberapa blok teks asli M dan blok teks rahasia C.

$C = M' \bmod n$

$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \text{ od } n$

Blok pengirim meupun penerima harus mengetahui nilai n dan e, dan hanya penerima saja yang mengetahui nilai d. ini merupakan algoritma enkripsi kunci umum dengan kunci umum sebesar KU = {e,n} dan kunci khusus sebesar KR = {d,n}. Agar algoritma ini bisa memenuhi syarat sebagai enkripsi kunci umum yang baik, maka harus memenuhi ketentuan-ketentuan seperti berikut :

1. Kemungkinan menemukan nilai e,d,n sedemikian rupa sehingga $M^{ed} = M \bmod n$ untuk semua $M < n$
2. Relative mudah menghitung M^e dan C^d untuk semua nilai $M < n$
3. Tidak mudah menghitung menentukan d, yang diberi e dan n.

dua ketentuan pertama bisa terpenuhi dengan mudah. Sedangkan ketentuan ketiga baru bisa terpenuhi untuk nilai e dan n yang besar.

Pembangkitan Kunci

1. Memilih dua bilangan prima p, q . bilangan ini harus cukup besar (minimal 100 digit).
2. Menghitung n = p . q. Bilangan n disebut *parameter security* (sebaiknya $p \neq q$, sebab jika p = q maka $n = p^2$ sehingga p dapat diperoleh dengan menarik akar pangkat dua dari n).
3. Menghitung $\varphi(n) = (p-1)(q-1)$.
4. Memilih bilangan bulat e dengan algoritma Euclid yaitu gcd $(\varphi(n),e)=1$; dimana $1< e < \varphi(n)$.
5. Menghitung d dengan rumus $d = e\text{-}1 \bmod \varphi(n)$
   Atau $e \,.\, d \equiv 1 \pmod{\varphi(n)}$.
   Perhatikan bahwa $e \,.\, d \equiv 1 \pmod{\varphi(n)}$ ekivalen dengan $e \,.\, d = 1 + k\,\varphi(n)$,
   sehingga secara sederhana d dapat dihitung dengan :
   $d = (1 + k \,.\, \varphi(n)) / e$
6. Kunci umum (kunci public) adalah KU = {e,n}
7. Kunci pribadi (kunci privat) adalah KR = {d,n}

Catatan : n tidak bersifat rahasia, sebab ia diperlukan pada perhitungan enkripsi/dekripsi

Enkripsi:

B mengenkripsi message M untuk A, yang harus dilakukan B :

1. Teks asli dengan syarat $M < n$
2. Ambil kunci public A yang otentik (n,e)
3. Representasikan message sebagai integer M dalam interval [0,n-1]
4. Teks Rahasia didapat dari $C = M^e \pmod n$
5. Kirim C ke A

Dekripsi:

Untuk mendekripsi, A melakukan:

1. Gunakan kunci pribadi d untuk menghasilkan M
2. Teks rahasia adalah C
3. Teks asli didapat dari $M = C^d \pmod n$

2.5 Algoritma Rivest's Cipher (RC4)

Menurut Rinaldi Munir, RC4 merupakan salah satu jenis stream cipher yang didesain oleh Ronald Rivest di laboratorium RSA (RSA Data Security inc) pada tahun 1987. RC4 sendiri merupakan kepanjangan dari Ronald Code atau Rivest's Cipher. RC4 stream cipher ini merupakan salah satu jenis algoritma yang mempunyai S-Box dan menggunakan variable yang panjang kuncinya 1 sampai 256 bit yang digunakan untuk menginisialisasikan tabel sepanjang 256 bit.

RC4 diinisialisasi dari sebuah kunci rahasia. Kemudian di-generate sebuah "keystream" yang disederhanakan dengan XOR dengan plaintext untuk menghasilkan ciphertext. Proses dekripsi sama dengan proses enkripsi. Salah satu alasan untuk kepopuleran RC4 adalah kesederhanaannya. Algoritma RC4 dapat diingat dan mudah diimplementasikan. Algoritma RC4 menggunakan 256 byte dari memori, S[0] hingga S[255], dan menggunakan variable integer i, j, dan k. RC4 adalah salah satu cipher yang tercepat yang dipergunakan secara luas untuk pekerjaan yang serius. RC4 menggunakan panjang kunci variabel dari 1–256 byte (mempunyai kemampuan antara 1−2048 bit) untuk menginisialisasi 256−byte *state table*. Algoritma RC4 dibagi menjadi dua tahap, yaitu membentuk kunci dan ciphering (enkripsi/dekripsi). Pembentukan kunci merupakan tahap pertama dan tersulit. Selama pembentukan kunci, kunci enkripsi digunakan untuk menghasilkan sebuah variabel enkrip menggunakan 2 array (state array & key array) dan sejumlah operasi penjumlahan. (Tanto, 2010).

3. METODOLOGI PENELITIAN

Tujuan dari penelitian ini, dapat mengetahui metode mana yang tepat dalam melakukan proses keamanan pesan yang akan dikirim, sehingga proses pengiriman semakin cepat. Agar dokumen yang dikirimkan terjaga kerahasiaannya maka dilakukanlah proses enkripsi – dekripsi dengan mengunakan algoritma kunci public. Pada penelitian ini diambil tiga buah metode untuk melakukan analisis, dimana metode tersebut adalah algoritma simetrik kriptosistem dan asimetrik kriptosistem.

Pada metode algoritma RC4 proses enkripsi dan dekripsi menggunakan cipher yang memiliki kunci simetris dan mengenkripsi atau mendekripsi plainteks secara digit per digit atau bit per bit dengan cara mengkombinasikan secara operasi biner (biasanya operasi XOR) dengan sebuah angka semiacak kemudian algoritma Rivest Shamir Adleman (RSA) proses enkripsi – dekripsi menggunakan dua buah kunci yaitu kunci public untuk proses enkripsi dan kunci privat untuk proses dekripsi serta sama – sama menggunakan pemfaktoran bilangan prima untuk melakukan pengamanan data kemudian Algoritma hash MD5 sendiri menerima *input* berupa data dengan panjang bebas, dan menghasilkan *output* heksadesimal sepanjang 32 karakter. Jadi, seberapapun panjang data input, output yang dihasilkan akan selalu sepanjang 32 karakter. Perubahan sedikit saja di *input* akan mengubah *output* dengan drastis sehingga dengan demikian proses untuk proses dekripsi isi dari dokumen semakin susah dibaca oleh orang yang tidak bertanggungjawab.

3.1 Teknik Pengembangan

Sistem yang akan penulis rancang terdiri dari tiga buah metode yaitu metode dengan algoritma MD5, RSA dan RC4 dimana pada ketiga metode tersebut memiliki tahap – tahap seperti tahap pembangkitan kunci untuk mendapatkan kunci privat dan kunci public, tahap enkripsi dan tahap dekripsi.

3.2 Algoritma Message-Digest Algortihm 5 (MD5)

3.2.1 Metode Algoritma Message-Digest Algortihm 5 (MD5)

Langkah-langkah pembuatan message digest secara garis besar:

1. Penambahan bit-bit pengganjal (padding bits).
2. Penambahan nilai panjang pesan semula.
3. Inisialisasi penyangga (buffer) MD.
4. Pengolahan pesan dalam blok berukuran 512 bit.

3.3 Algoritma Rivest Shamir Adleman (RSA)

3.3.1 Metode Algoritma Rivest Shamir Adleman (RSA)

Algoritma RSA memiliki dua kunci yang berbeda untuk proses enkripsi dan dekripsi. Dalam menentukan dua bilangan prima sebagai kunci adalah bilangan prima yang besar, karena pemfaktoran bilangan dari dua bilangan prima yang besar sangat sulit, sehingga keamanan pesan lebih terjamin. Pasangan kunci adalah elemen penting dari algoritma RSA. Berikut ini langkah – langkah dalam membangkitkan dua kunci algoritma RSA.

1. Pilih dua bilangan prima sembarang p dan q.
2. Hitung n = p.q
3. Hitung φ (n) = (p – 1) (q – 1)
4. Pilih kunci public e, yang relative prima terhadap φ (n)
5. Bangkitkan kunci pribadi dengan menggunakan e.d ≡ 1 mod φ (n)

Hasil dari algoritma tersebut akan menghasilkan dua kunci, yaitu kunci public (e, n) dan kunci pribadi (d,n).

3.4 Metode Algoritma Rivest Cipher 4 (RC4)

3.4.1. Tahap Pembentukan Kunci

Algoritma RC4 memiliki dua fase, setup kunci dan pengenkripsian. Setup untuk kunci adalah fase pertama dan yang paling sulit dalam algoritma ini. Dalam setup S bit kunci (S merupakan panjang dari kunci), kunci enkripsi digunakan untuk menghasilkan variabel enkripsi yang menggunakan dua buah array, state dan kunci, dan sejumlah-S hasil dari operasi penggabungan. Operasi penggabungan ini terdiri dari pemindahan (*swapping*) byte, operasi modulo, dan rumus lain. Operasi modulo merupakan proses yang menghasilkan nilaisisa dari satu pembagian. Sebagai contoh, 11 dibagi 4 adalah 2 dengan sisa pembagian

3, begitu juga jika tujuh modulo empat maka akan dihasilkan nilai tiga.

Variabel enkripsi dihasikan dari setup kunci dimana kunci akan di XOR-kan dengan plain text untuk menghasilkan teks yang sudah terenkripsi. XOR merupakan operasi logik yang membandingkan dua bit biner. Jika bernilai beda maka akan dihasilkan nilai 1. Jika kedua bit sama maka hasilnya adalah 0. Kemudian penerima pesan akan mendekripnya dengan meng XOR-kan kembali dengan kunci yang sama agar dihasilkan pesan dari plain text tersebut.

RC4 menggunakan dua buah kotak substitusi (S-Box) array 256 byte yang berisi permutasi dari bilangan 0 sampai 255 dan S-Box kedua yang berisi permutasi fungsi dari kunci dengan panjang yang variabel. Cara kerja algoritma RC4 yaitu inisialisasi Sbox pertama, S[0],S[1],...,S[255], dengan bilangan 0 sampai 255. Pertama isi secara berurutan S[0] = 0, S[1] = 1,...,S[255] = 255. Kemudian inisialisasi array lain (S-Box lain), misal array K dengan panjang 256. Isi array K dengan kunci yang diulangi sampai seluruh array K[0], K[1],...,K[255] terisi seluruhnya.

4. HASIL DAN PEMBAHASAN

4.1. Pendahuluan

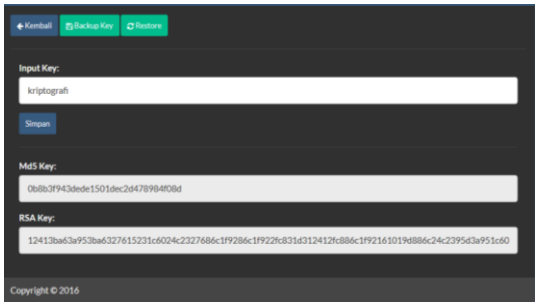
Hasil penelitian dengan menggunakan metode MD5, RSA dan RC4 dalam melakukan pengamanan data. Penelitian akan melakukan perbandingan proses kerja dari ketiga metode terhadap beberapa ukuran file berbeda. Spesifikasi dari perangkat keras dan perangkat lunak yang digunakan adalah sebagai berikut:

1. Sistem Operasi Microsoft Windows 7 Ultimate 32 bit.
2. Ukuran Harddisk tempat penyimpanan sistem operasi 350 GB.
3. Prosesor AMD Dual – Core Processor C 70
4. RAM 2 GB.

4.2. Pengujian

4.2.1 Pembentukan key

Langkah pertama masukkan key (*input key*) simpan key dengan menekan tombol Simpan maka akan menghasilkan key MD5 dan key MD5 yang dihasilkan akan di enkripsi lagi dengan algoritma RSA. Hasil dari enkripsi key MD5 dan RSA tersebut dapat dilihat pada gambar 4.1 dibawah ini.



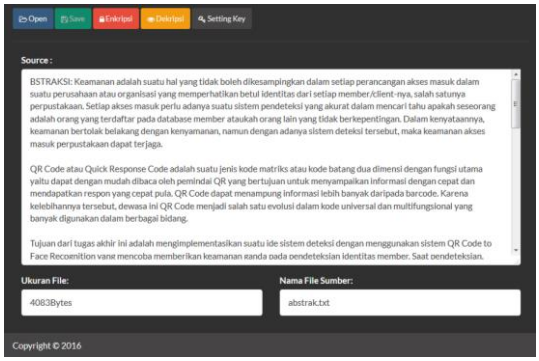
Gambar 4.1 Hasil Enkripsi Key dengan Algoritma MD5 dan RSA

Dari hasil yang dihasilkan oleh key yang di inputkan yaitu : kriptografi akan menghasilkan enkripsi key MD5 yaitu : 0b8b3f943dede1501dec2d478984f08d dan hasil enkripsi key MD5 yang di enkripsi dengan algoritma RSA akan menghasilkan :

12413ba63a953ba6327615231c6024c2327686c1f92
86c1f922fc831d312412fc886c1f92161019d886c24c
2395d3a951c603a9524c2152312413a9586c

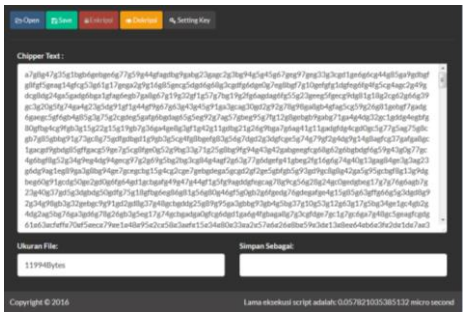
4.2.2. Proses Enkripsi File

Untuk proses enkripsi file, dapat dipilih tombol Open kemudian cari file yang akan dienkripsi. File yang akan di enkrip bisa berupa file yang berextension .txt dan .doc. File yang sudah dipilih akan ditampilkan pada layar Source dan akan ditampilkan ukuran dari file tersebut dan nama filenya. Proses tersebut dapat dilihat pada gambar 4.2 dibawah ini:



Gambar 4.2 Tampilan isi file yang akan di enkripsi dengan ukuran 4038Bytes

Selanjutnya pilih atau klik tombol Enkripsi maka akan menampilkan hasil enkripsi file seperti gambar 4.3 dibawah ini:

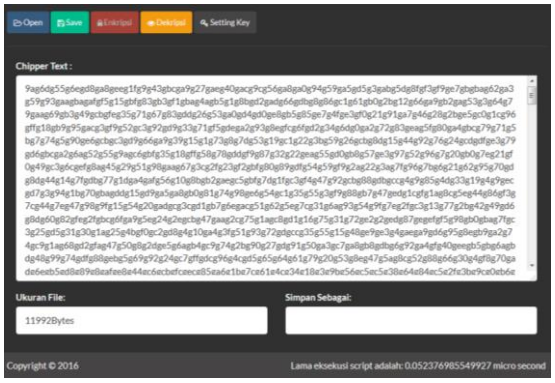


Gambar 4.3 Hasil Enkripsi file dengan ukuran 11994Bytes

Dari hasil enkripsi yang telah dilakukan dengan menggunakan key “kriptografi” dapat dihasilkan dengan perubahan ukuran file 4038Bytes menjadi 11994Bytes waktu yang diperlukan untuk memproses yaitu: 0.057821035385132 micro second. File yang dienkrip disimpan dengan menekan tombol Save.

Enkripsi dengan panjang key yang berbeda dengan dengan isi file yang sama ukuran akan menghasilkan hasil yang berbeda juga perubahan

ukuran filenya, dapat dilihat pada gambar 4.4 dibawah ini:



Gambar 4.4 Hasil enkripsi dengan ukuran key yang berbeda

Hasil enkripsi yang diperoleh dari key yang berbeda panjang karakternya yaitu "kriptografi rahasia" dihasilkan ukuran file menjadi 11992Bytes dan lama prosesnya 0.052376985549927 micro second.

4.2.3. Proses Dekripsi File

Untuk proses dekripsi file dapat menekan tombol Open pilih File yang sudah di enkrip setelah tampil isi file yang di enkripsi pilih tombol Dekripsi maka akan menghasilkan isi file aslinya. Proses ini menggunakan algoritma RC4 untuk dekripsinya dengan key yang sudah disimpan. Hasil dekripsi dapat dilihat pada gambar 4.5 berikut ini.

Gambar 4.5 Hasil Dekripsi file

Hasil dekripsi juga dapat disimpan kembali dengan membuat nama file kemudian pilih Save. Hasil dekripsi ukuran file sama dengan file aslinya, waktu yang diperlukan untuk proses dekripsi 0.1027090549469 micro second

5. PENUTUP

5.1 Kesimpulan

1. Pengamanan sebuah dokumen dengan menggunakan algoritma RC4 tidak mengambil nonce yang terpisah bersamaan dengan kunci. Hal ini berarti jika kunci *single long-term* digunakan untuk mengenkripsi beberapa stream, kriptosistemnya harus menentukan bagaimana cara mengombinasikan *nonce* tersebut dan kunci *long-term* untuk menghasilkan kunci stream untuk RC4. Untuk menangani hal tersebut adalah dengan membuat sebuah kunci RC4 dengan menggunakan fungsi *hash*. Enkripsi dengan menggunakan RC4 dapat diterobos dan rentan terhadap *bit-flipping attack*. Untuk menanggulangi hal ini, skema enkripsi harus dikombinasikan dengan *message authentication* code yang kuat.
2. Algoritma kriptografi dapat digunakan sebagai salah satu pilihan dalam menjaga dan mengamankan pesan rahasia karena sulitnya melakukan faktorisasi terhadap bilangan yang terbentuk dari 2 bilangan prima yang besar sehingga tingkat keamanan algoritma kriptografi RSA cukup tinggi.

5.2 Saran

1. Dalam pengembangan RSA penelitian ini disarankan ekstensi file yang dapat di enkripsi lebih banyak lagi, karena dalam penelitian ini file yang dienkripsi hanya teks dalam ekstensi .docx dan .txt
2. MD5 cenderung rentan terhadap serangan collision yaitu suatu peristiwa di mana dua nilai yang berbeda dapat memiliki nilai hash yang sama, cara untuk memanfaatkan collision ini untuk memalsukan sertifikat SSL jejaring palsu, sehingga algoritma MD5 tidak cocok untuk dipakai sebagai fungsi enkripsi yang membutuhkan ketahanan dari serangan *collision*. Untuk alternatif dari MD5, dianjurkan penggunaan algoritma SHA1 (*Secure Hash Algorithm-1*) atas ketahanan algoritma ini terhadap serangan *collision* yang relatif lebih mumpuni dibandingkan dengan MD5.
3. File yang dienkripsi sebaiknya merupakan file yang benar-benar perlu untuk dilindungi, seperti file informasi pribadi, file yang berhubungan dengan pekerjaan atau proyek, file data-data transaksi keuangan, serta file sejenis agar penggunaan aplikasi ini tepat sasaran.

DAFTAR PUSTAKA

[1]Amri, Khoirul. 2014. Implementasi *Hybrid Cryptosystem* Menggunakan Algoritma RSA, 3DES Dan MD5 Pada Aplikasi Sms Berbasis Android. Tesis. STMIK AMIKOM YOGYAKARTA

[2]Ariyus, Dony. 2006. *Computer Security*. Penerbit Andi : Yogyakarta

[3]Ariyus, Dony. 2008. Pengantar Ilmu Kriptografi : Teori, Analisis Dan Implementasi. Penerbit Andi : Yogyakarta

[4]Diffie, W., Hellman, M.E.,New directions in cryptography, IEEE Transactions on Information Theory 22 (1976), 644–654.

[5]Fauziah, Yuli. 2008. Pengamanan Pesan Dalam Editor Teks Menggunakan *Hybrid Cryptosystem.* Seminar nasional

Informatika 2008 (SemNasIF 2008) ISSN: 1979-2328 Yogyakarta, 24 Mei 2008

[6]Hasan, Alwi, dkk. 2000. Tata Bahasa Baku Bahasa Indonesia (Edisi Ketiga). Balai Pustaka : Jakarta.

[7]Kristanto, Andri. 2003. Keamanan Data Pada Jaringan Komputer. Penerbit Gava Media : Yogyakarta

[8]Kurniawan, Yusuf. 2004. Kriptografi Keamanan Internet dan Jaringan Komunikasi. Cetakan Pertama.Informatika : Bandung

[9]Listiyono, Hersatoto. 2009. *Implementasi Algoritma Kunci Public Pada Algoritma RSA*. Dinamika informatika – Vol I No 2, September 2009 ISSN : 2085-3343

[10]Supriyanto, Aji. 2009. Pemakaian Kriptografi Kunci Publik Untuk Proses Enkripsi Dan Tandatangan Digital Pada Dokumen E-Mail, DINAMIKA INFORMATIKA – Vol I No 1, Maret 2009 ISSN : 2085-3343

[11]Wahyuni, Ana. 2011. *A*plikasi Kriptografi Untuk Pengamanan E-Dokumen Dengan Metode *Hybrid :* Biometrik Tandatangan Dan Dsa *(Digital Signature Algorithm)*. Tesis. Universitas Diponegoro Semarang

[12]Zain, R.H. 2012. Perancangan Dan Implementasi Cryptography Dengan Metode Algoritma Rc4 Pada Type File Document Menggunakan Bahasa Pemrograman Visual Basic 6.0. Jurnal Momentum ISSN : 1693-752X - Vol.12.No.1. Februari 2012.