ISSN : 1978-6603

JURNAL Ilmiah SAINTIKOM Sains dan Komputer

APLIKASI KEAMANAN EMAIL MENGGUNAKAN ALGORITMA RC4

Nurcahyo Budi Nugroho[#1], Zulfian Azmi[#2], Saiful Nur Arif[#3]
[#1, 2, 3] Program Studi Sistem Komputer, STMIK Triguna Dharma
email: nurcahyobn@gmail.com

Abstrak

Tujuan mendasar dari ilmu kriptografi adalah aspek keamanan informasi yaitu kerahasiaan, integritas data, autentikasi, non-repudiasi atau nirpenyangkalan.E-mail adalah aktivitas pertama yang paling banyak digunakan di internet, baik untuk berkirim pesan dari teman, rekan bisnis, transaksi bank dan lain sebagainya. Keempat aspek tersebut harus dimiliki pada pesan berbasis jaringan global ini. Kebebasan mengakses layanan dimanapun mengakibatkan peluang yang besar bagi para penyusup di internet untuk mengambil informasi baik yang bersifat umum maupun pribadi. Hal inilah yang mendasari penelitian untuk membuat aplikasi yang dapat membantu menjaga keamanan pesan pada email. Algoritma kriptografi Rivest Code 4 (RC4) merupakan salah satu algoritma kunci simetris dibuat oleh RSA Data Security Inc (RSADSI) yang berbentuk stream chipper yang memproses unit atau input data, pesan atau informasi pada satu saat.

Kata kunci : Keamanan, Kriptografi, Email, RC4

Abstract

The fundamental purpose of the science of cryptography is the security aspect of information, namely confidentiality, data integrity, authentication, non-repudiation or nirpenyangkalan.E-mail is the first activity of the most widely used on the Internet, either to send messages from friends, business associates, and other bank transactions etc. The fourth aspect of the must-have in this global network-based message. Freedom of access to services wherever resulted in a great opportunity for the intruders on the internet to retrieve information both public and private. It is the background research to create applications that can help maintain the security of the email message. Cryptographic algorithms Rivest Code 4 (RC4) is a symmetric key algorithm created by RSA Data Security Inc. (RSADSI) shaped stream cipher that processes or data input unit, messages or information at one time.

Keywords: *Security, Cryptography, Email, RC4*

A. PENDAHULUAN

1. Latar Belakang

Masalah keamanan dan kerahasiaan merupakan salah satu aspek penting dari suatu pesan, data, atau informasi. Di mana kebenaran dan keaslian suatu informasi sangat penting baik pada saat pengiriman ataupun pada saat informasi tersebut diterima. Pesan, data, atau informasi tidak akan berguna lagi apabila pada saat pengiriman informasi tersebut disadap atau dibajak oleh orang yang tidak berhak atau berkepentingan.

Masalah keamanan dan kerahasiaan merupakan salah satu aspek penting dari suatu pesan, data, atau informasi. Di mana kebenaran dan keaslian suatu informasi sangat penting baik pada saat pengiriman ataupun pada saat informasi tersebut diterima. Pesan, data, atau informasi tidak akan berguna lagi apabila pada saat pengiriman informasi tersebut disadap atau dibajak oleh orang yang tidak berhak atau berkepentingan.

2. Tujuan Penelitian

Tujuan dari penulisan penelitian ini adalah sebagai berikut:

a. Mengetahui secara jelas bagaimana algoritma kriptografi RC4 dapat mengenkripsipesan yang dikirim oleh pengirim.
b. Memproteksi data e-mail sehingga yang bisa membaca e-mail hanya orang tertentu menggunakan *password* dari metode RC4.
c. Membangun aplikasi kriptografi untuk mengamankan email dalam bentuk teks dengan panjang kunci minimal 5 karakter.

3. Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut

a. Aplikasi ini dapat membantu pengguna e-mail untuk tidak salah dalam penyalahgunaan pesan/data tersebut dan menjaga kerahasiaan pesan.
b. Untuk pengamanan e-mail agar data tidak boleh dibuka oleh sembarang orang, kecuali yang berhak.
c. Memudahkan para pengguna untuk mengirimkan pesan rahasia ke *client* tanpa harus membuka *browser*.

B. LANDASAN TEORI

1. Keamanan Data

Masalah keamanan dan kerahasiaan data merupakan salah satu aspek penting dari suatu informasi. Secara umum keamanan komputer mencakup beberapa aspek (Stiawan, 2005), yaitu :

a. Privacy / Confidentiality

Privacy lebih kearah data- data yang sifatnya rahasia, sedangkan *confidentiality* berhubungan dengan data yang diberikan ke pihak lain untuk keperluan tertentu dan hanya diperbolehkan untuk keperluan tertentu.

b. Integrity

Aspek ini menekankan bahwa informasi tidak boleh diubah tanpa seizin pemilik informasi. Informasi yang diterima harus sesuai dan sama persis seperti saat informasi dikirimkan. Jika terdapat perbedaan antara informasi atau data yang dikirim dengan yang diterima maka aspek *integrity* tidak tercapai.

c. Authenticity

Aspek ini berhubungan dengan metode atau cara untuk menyatakan bahwa informasi betul-betul asli, orang yang mengakses atau memberikan informasi adalah betul-betul orang yang dimaksud.

d. Availability

Aspek ini behubungan dengan ketersediaan data dan informasi. Data dan informasi yang berbeda dalam

suatu sistem komputer tersedia dan dapat dimanfaatkan oleh orang yang berhak.

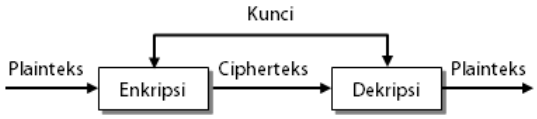
e. *Access Control*

Aspek ini berhubungan dengan cara pengaturan akses kepada informasi. Hal ini biasanya berhubungan dengan klasifikasi data, mekanisme authentication dan juga *privacy*. *Access control* seringkali dilakukan dengan menggunakan kombinasi *user id/password* atau dengan menggunakan mekanisme lain.

2. Algoritma Kriptografi Simetris

Kriptografi kunci-simetris merujuk pada metode enkripsi di mana kedua pengirim dan penerima membagi kunci yang sama. Algoritma kriptografi simetris atau disebut juga algoritma kriptografi konvensional adalah algoritma yang menggunakan kunci yang sama untuk proses enkripsi dan proses dekripsi.

Algoritma kriptografi simetris dibagi menjadi 2 kategori yaitu algoritma aliran (*stream ciphers*) dan algoritma blok (*block ciphers*). Pada algoritma aliran, proses penyandian berorientasi pada satu bit atau satu byte data. Sedangkan pada algoritma blok, proses penyandiannya berorientasi pada sekumpulan bit atau byte data (per blok).



Gambar 2.1 Algoritma Kriptografi Simetris

3. Algoritma RC4

Algoritma kriptografi Rivest Code 4 (RC4) merupakan salah satu algoritma kunci simetris dibuat oleh RSA Data *Security Inc* (RSADSI) yang berbentuk *stream chipper*. Algoritma ini ditemukan pada tahun 1987 oleh Ronald Rivest dan menjadi simbol keamanan RSA(merupakan singkatan dari tiga nama penemu: Rivest Shamir Adleman). RC4 menggunakan panjang kunci dari 1 sampai 256 *byte* yang digunakan untuk menginisialisasikan tabel sepanjang 256 *byte*. Tabel ini digunakan untuk generasi yang berikut dari *pseudo random* yang menggunakan XOR dengan *plainteks* untuk menghasilkan *cipherteks*. Masing-masing elemen dalam tabel saling ditukarkan minimal sekali.

RC4 merupakan merupakan salah satu jenis *stream cipher*, yaitu memproses unit atau input data pada satu saat. Dengan cara ini enkripsi atau dekripsi dapat dilaksanakan pada panjang yang variabel. Algoritma ini tidak harus menunggu sejumlah input data tertentu sebelum diproses, atau menambahkan *byte* tambahan untuk mengenkrip. Metode enkripsi RC4 sangat cepat kurang lebih 10 kali lebih cepat dari DES.

Algoritma RC4 Stream Cipher Untuk melakukan enkripsi-dekripsi,

a. Proses inisialisasi *S-Box* (*Array* S)

```
for i = 0 to 255, S[i] = i
```

b. Proses inisialisasi *S-Box* (*Array* K)

```
for i = 0 to 255, S[i] = i
```

c. Kemudian lakukan langkah pengacakan S-Box

```
i=0; j=0
for i= 0 to 255{
        j=(j+S[i]+[k] mod 256
        swap S[i] dan S[j]
}
```

d. Membuat *pseudorandom byte*

i= (i+1) mod 256
j=(j+S[i]) mod 256
swap S[i] dan S[j]
t=(S[i] + S[j]) mod 256
K=S[t]

4. Email

Electronic mail (surat elektronik, *e-mail*) adalah sebuah metode menggubah, mengirim, menyimpan, dan menerima pesan melalui sistem komunikasi elektronik. Istilah *e-mail* meliputi sistem yang berdasar pada *Simple Mail Transfer Protocol* (SMTP) dan sistem intranet yang memungkinkan pengguna dalam satu organisasi mengirimkan pesan kepada satu sama lain. Seringkali kelompok organisasi tersebut menggunakan *internet protocol* sebagai layanan *e-mail* internal.

Format dari sebuah pesan *e-mail* dari internet didefinisikan di RFC 2822 dan seri dari RFC yang secara keseluruhan disebut sebagai *Multipurpose Internet MailExtensions* (MIME). Sebuah pesan e-mail terdiri dari dua bagian besar:

a. Header

Disusun menjadi beberapa *field*, umumnya nama *field* dimulai pada karakter pertama pada suatu baris, diikuti oleh tanda ':', diikuti oleh nilai non-null, bukan spasi atau bukan tab pada karakter pertamanya. Nama field dan nilainya masuk dalam karakter ASCII sebesar 7 bit. Bagian header dan body dipisahkan oleh satu baris kosong. Pesan pada umumnya paling sedikit memiliki 4 *field* berikut :

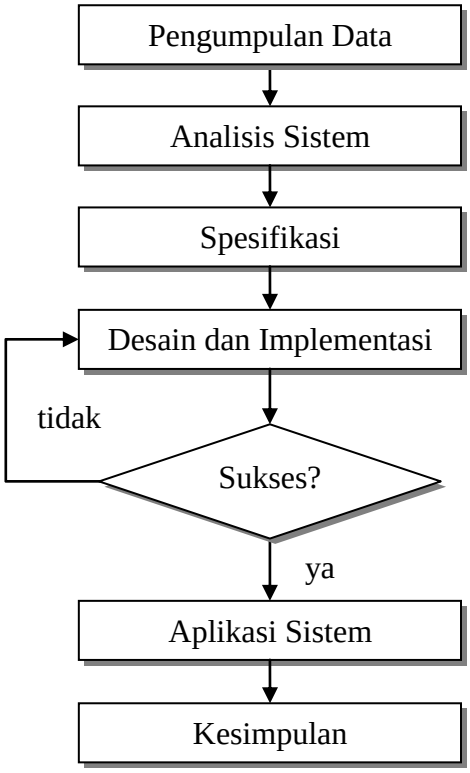
1. From : Alamat e-mail dan terkadang diikuti nama pengirim pesan.
2. To : Alamat e-mail dan terkadang diikuti nama penerima pesan.
3. Subject : Rangkuman isi pesan.
4. Date : Waktu dan tanggal setempat saat pesan dikirim

b. Body

Pesan yang diterima berupa teks tanpa struktur, terkadang mengandung tanda pengenal di bagian akhir. Pada awalnya didesain menggunakan 7-bit ASCII, tapi sekarang sebagian besar menggunakan 8-bit, namun belum bersifat universal.

C. METODE PENELITIAN

Langkah-langkah yang dilakukan dalam penelitian ini dapat dilihat pada gambar berikut ini.



Gambar 3.1 Bagan Alir Penelitian

D. ANALISA SISTEM

RC4 merupakan sistem sandi stream berorientasi byte, kemudian dilakukan operasi XOR dengan sebuah byte kunci, dan menghasilkan sebuah byte sandi.

Byte K di-XOR-kan dengan *plainteks* untuk menghasilkan *cipherteks* atau di-XOR-kan dengan *cipherteks* untuk menghasilkan *plainteks*.

Untuk mencoba kasus yang akan dibahas dan disimulasikan pembelajarannya dalam penulisan skripsi ini adalah dengan menyajikan data sebagai berikut:

Plainteks : "PDP2016"

Kunci : "Email"

Analisis kriptografi menggunakan algoritma RC4 adalah sebagai berikut:

1. Inisialisasi S-Box dengan panjang 256 byte, dengan S[0]=0, S[1]=1, S[2]=2 dan S[3]=3,….., S[255]=255 sehingga array S menjadi:

S[i]	0	1	2	3	4		255
Nilai	0	1	2	3	4		255

2. Inisialisasi 7 byte kunci array, Ki. Misalkan kunci terdiri dari 8 byte yaitu "Email" maka kalimat akan diubah kedalam bentuk Desimal "69, 109, 97, 105, 108". Ulang kunci sampai memenuhi seluruh array K sehingga array K menjadi:

i	0	1	2	3	4		255
Char	E	m	a	i	l	…	E
Dec	69	109	97	105	108		69

3. Berikutnya mencampur operasi dimana akan menggunakan variabel i dan j ke index array S[i] dan K[i]. pertama kita beri nilai inisial untuk i dan j dengan 0. Operasi Pencampuran adalah pengulangan rumusan (j + S[i]+K[i]) mod 256 yang diikuti dengan penukaran S[i] dengan S[j]. Untuk contoh ini, karena kita menggunakan array dengan panjang 256 byte maka algoritma menjadi:

```
For i = 0 to 256
    j = (j +S[i] + K[i]) mod 256
    swap S[i] dan S[j]
```

Dengan algoritma seperti di atas maka dengan nilai awal i = 0 sampai i = 255 akan menghasilkan array S seperti di bawah ini:

45	179	22	130	10	48	80	23	66	253	76	196	7	167	33	117
4	100	223	94	183	237	156	49	41	238	208	112	181	255	218	25
111	230	64	220	109	243	3	68	38	52	24	13	121	240	36	154
228	44	61	182	40	17	105	221	126	50	151	62	8	244	234	142
116	16	247	14	147	173	124	178	231	159	110	18	106	152	187	95
175	128	91	213	74	141	77	189	12	67	190	203	102	207	122	113

134	222	101	144	115	39	210	162	161	27	73	45	93	87	235	79
241	59	172	70	75	249	28	9	107	149	123	171	138	227	5	63
226	229	201	155	11	26	118	245	209	192	6	135	82	191	174	166
163	69	0	204	186	214	37	251	21	143	239	84	88	31	232	170
177	137	47	15	202	180	200	81	233	176	34	99	108	29	104	199
20	215	206	193	46	96	83	43	212	92	119	85	158	72	56	127
224	165	157	168	216	58	252	236	86	19	57	169	164	53	150	194
54	225	35	198	30	148	160	139	55	120	1	51	250	254	78	131
97	125	132	2	60	71	136	188	195	185	246	133	90	184	140	248
65	42	219	153	242	103	129	89	114	211	98	205	146	197	217	32

4. Membuat *pseudorandom byte*
   i= (i+1) mod 256
   j=(j+S[i]) mod 256
   swap S[i] dan S[j]
   t=(S[i] + S[j]) mod 256
   K=S[t]

Iterasi ke-1 :
i = (i + 1) mod 256
= (0 + 1) mod 256 = 1
j = (j + S[i]) mod 256
= (0 + S[1]) mod 256
= (179 + 179) mod 256 = 179
Swap(S[1],S[179])
t = (S[i] + S[j]) mod 256
= (S[1] + S[179]) mod 256
= (193 + 179) mod 256 = 116
Key[0] = S[116] = 75
Key XOR Plaintext : 01001011 XOR 01010000 = 00011011 (27) ->

Iterasi ke-2 :
i = (i + 1) mod 256
= (1 + 1) mod 256 = 2
j = (j + S[i]) mod 256
= (179 + S[2]) mod 256
= (201 + 22) mod 256 = 201
Swap(S[2],S[201])
t = (S[i] + S[j]) mod 256
= (S[2] + S[201]) mod 256
= (19 + 22) mod 256 = 41
Key[1] = S[41] = 52
Key XOR Plaintext : 00110100 XOR 01000100 = 01110000 (112) -> p

Iterasi ke-3 :
i = (i + 1) mod 256
= (2 + 1) mod 256 = 3
j = (j + S[i]) mod 256
= (201 + S[3]) mod 256
= (75 + 130) mod 256 = 75
Swap(S[3],S[75])
t = (S[i] + S[j]) mod 256
= (S[3] + S[75]) mod 256
= (18 + 130) mod 256 = 148
Key[2] = S[148] = 186
Key XOR Plaintext : 10111010 XOR 01010000 = 11101010 (234) -> ê

Iterasi ke-4 :
i = (i + 1) mod 256
= (3 + 1) mod 256 = 4
j = (j + S[i]) mod 256
= (75 + S[4]) mod 256

= (85 + 10) mod 256 = 85
Swap(S[4],S[85])
t = (S[i] + S[j]) mod 256
= (S[4] + S[85]) mod 256
= (141 + 10) mod 256 = 151
Key[3] = S[151] = 251
Key XOR Plaintext : 11111011 XOR 00110010 = 11001001 (201) -> É

Iterasi ke-5 :
i = (i + 1) mod 256
= (4 + 1) mod 256 = 5
j = (j + S[i]) mod 256
= (85 + S[5]) mod 256
= (133 + 48) mod 256 = 133
Swap(S[5],S[133])
t = (S[i] + S[j]) mod 256
= (S[5] + S[133]) mod 256
= (26 + 48) mod 256 = 74
Key[4] = S[74] = 110
Key XOR Plaintext : 01101110 XOR 00110000 = 01011110 (94) -> ^

Iterasi ke-6 :
i = (i + 1) mod 256
= (5 + 1) mod 256 = 6
j = (j + S[i]) mod 256
= (133 + S[6]) mod 256
= (213 + 80) mod 256 = 213
Swap(S[6],S[213])
t = (S[i] + S[j]) mod 256
= (S[6] + S[213]) mod 256
= (148 + 80) mod 256 = 228
Key[5] = S[228] = 60
Key XOR Plaintext : 00111100 XOR 00110001 = 00001101 (13) ->

Iterasi ke-7 :
i = (i + 1) mod 256
= (6 + 1) mod 256 = 7
j = (j + S[i]) mod 256
= (213 + S[7]) mod 256
= (236 + 23) mod 256 = 236
Swap(S[7],S[236])
t = (S[i] + S[j]) mod 256
= (S[7] + S[236]) mod 256
= (90 + 23) mod 256 = 113
Key[6] = S[113] = 59
Key XOR Plaintext : 00111011 XOR 00110110 = 00001101 (13)

E. HASIL DAN PEMBAHASAN

Setelah tahap analisa dan perancangan pada metode penelitian selesai, langkah selanjutnya adalah implementasi dan pengujian aplikasi keamanan email dengan spesifikasi perangkat sistem lunak sebagai berikut:

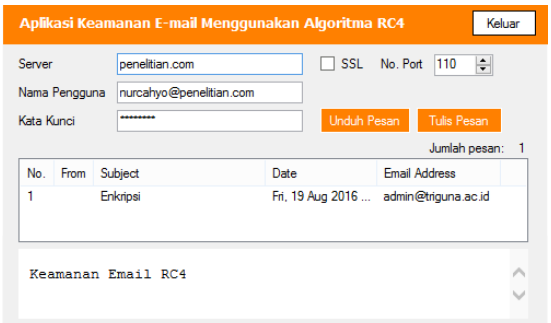
1. Microsoft Visual Studio 2010
2. Mercury Main Server
3. Mozilla Thunderbird atau Microsoft Outlook

1. Implementasi

Implementasi sistem merupakan prosedur yang dilakukan untuk mnyelesaikan desain sistem dan dapat melakukan pengujian sistem yang dibuat.

1. *Form* Utama

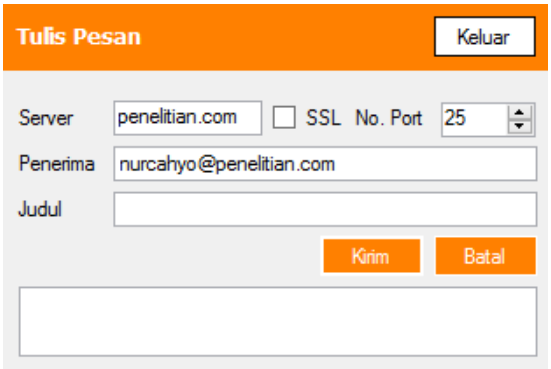
Form utama berfungsi sebagai *form* untuk melakukan koneksi keserver e-mail. Sehingga e-mail yang ada di server bisa terlihat dan tampil di *form* ini



Gambar 4.1 Form Utama Email

2. *Form* Tulis Pesan

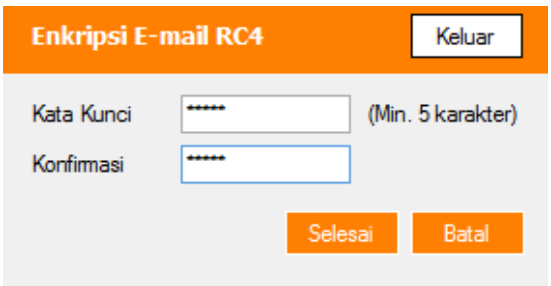
Form tulis pesan berfungsi sebagai *form* untuk melakukan pengiriman e-mail ke *client*.



Gambar 4.2Form Kirim Email

3. *Form* Enkripsi dan Dekripsi E-mail

Form ini berfungsi sebagai *form* untuk melakukan pembentukan kunci. Sehingga e-mail yang ada di server tidak bisa dibaca. dan minimal pembentukan kunci adalah 5 karakter termasuk dekripsi.



Gambar 4.3 Form Enkripsi Email

F. SIMPULAN

Kesimpulan yang dapat diambil setelah melakukan penelitian dan hasil pengujian terhadap program aplikasi, maka dapat ditarik kesimpulan sebagai berikut:

1. Semua karakter pada *body* e-mail bisa di enkripsi dan di dekripsi dengan valid dan berjalan baik menggunakan algoritma RC4.
2. Sistem yang dibuat sudah mampu memenuhi kebutuhan aplikasi e-mail *client* yang menerapkan kriptografi RC4.
3. Aplikasi *E-mail Client* ini langsung menampilkan Form Koneksi sehingga memudahkan dan cara kerja aplikasi ini sangat cepat dalam mengirimkan E-mail tanpa harus membuka *browser*.

G. DAFTAR PUSTAKA

Endang, Vantonny, & Reza. *RC4 Stream Chiper*. Ditemukenali 17 September 2013, dari http://www.achtung.com/crypto/rc4.html

http://library.binus.ac.id/eColls/eThesis/Bab2/2007-2-00194-IF-Bab%202.pdf

Ihsan. *KriptografiRC4*. Ditemukenali 17 September 2013, dari http://catatanichan.blogspot.com

Kurniawan. 2004. *Kriptografi-Keamanan Internet dan Jaringan Telekomunikasi,* Edisi 1. Bandung: INFORMATIKA.

Munir, Rinaldi. 2006. *Kriptografi.* Bandung: INFORMATIKA.

Priyanto, Rahmat. 2009. *Visual Basic 2008,* Edisi 1, Yogyakarta: ANDI.

Safira, Rika. *Studi Perbandingan Sistem Penyandian Pesan Dengan Algoritma Kriptografi RC2, RC4, RC5 dan RC6*. Ditemukenali 24 September 2013