Seminar Nasional Teknologi Komputer & Sains (SAINTEKS)
SAINTEKS 2019
ISBN: 978-602-52720-1-1
Januari 2019
Hal: 916 - 920

Implementasi Algoritma RC4+ Untuk Mengamankan Pesan Teks Pada Aplikasi Chatting

Dani Iqbal[1], Asep Roy Panggabean[1], Indra Williamsyah Sinaga[1], Taronisokhi Zebua[2]

[1]Prodi Teknik Informatika STMIK Budi Darma, Medan, Indonesia
[2]AMIK STIEKOM Sumatera Utara, Rantauprapat, Indonesia

Abstrak

Keamanan sebuah data pada era perkembangan teknologi saat ini sangat perlu diperhatikan. Pesan teks merupakan salah satu jenis pesan teks yang membutuhkan pengamanan terutama pesan teks yang bersifat rahasa dan pribadi. Aplikasi chatting merupakan salah satu media yang sering digunakan sebagai media berkomunikasi dalam bertukar pesan. Salah satu teknik yang dapat digunakan untuk mengoptimalkan keamanan terhadap sebuah data adalah memanfaatkan teknik kriptografi. Teknik kriptografi mengamankan pesan teks dengan cara menyandikan pesan asli menjadi pesan yang berbeda dengan pesan aslinya. Penelitian ini menguraikan tentang pengamanan pesan teks pada aplikasi chatting berdasarkan algoritma RC4+. Algoritma ini akan melakukan proses *key schedulling algorithm* dan proses *pseudo random generation algorithm* serta melakukan proses pemutaran *state* dan proses pemindahan sejumlah bit pada posisi tertentu untuk memperoleh kunci yang lebih acak yang dibutuhkan pada proses enkripsi dan dekripsi sehingga diperoleh *cipher* pesan asli yang sulit untuk dipecahkan oleh penyerang.

Kata Kunci: Kriptografi, Pesan, Aplikasi Chatting, Algoritma RC4+, Stream Cipher

1. PENDAHULUAN

Era reolusi industri 4.0 saat ini sangatlah berdampak terhadap peningkatan keamanan terhadap data-data yang didistribusikan dalam kegiatan berkomunikasi. Pesan teks adalah salah satunya, karena hingga saat ini pesan teks masih sangat umum dimanfaatkan sebagai media yang digunakan untuk mengkomunikasikan informasi yang bersifat rahasia maupun tidak rahasia. Salah satu hal penting yang perlu di perhatikan dalam pengiriman teks adalah keamanan dan kerahasian dalam pesan itu sendiri. Berdasarkan penelitian sebelumnya disebutkan bahwa pesan teks yang didistribusikan sangat beresiko terhadap adanya penyadapan pesan, karena tingkat keamanan informasi yang kurang diperhatikan[1].

Teknik kriptografi dapat dimanfaatkan untuk meminimalisir resiko dari pesan rahasia yang belum diamankan. Banyak media yang bisa digunakan untuk saling bertukar informasi atau pesan, salah satunya aplikasi *chatting*. Namun dalam aplikasi chatting masih ada kekurangan termasuk privasi atau kerahasian sebuah pesan. Pesan yang dikirim dan tidak diacak dapat beresiko terhadap informasi yang dimiliki oleh sebuah pesan dicuri dimana dapat dimanipulasi oleh pihak yang tidak berwenang sehingga merugikan pihak pengirim dan penerima [2].

Pengamanan suatu informasi dapat dilakukan dengan teknik kriptografi. Teknik kriptografi pada umumnya memiliki dua tahap yaitu tahap enkripsi dan dekripsi. Tahap enkripsi adalah suatu proses yang dilakukan untuk mengubah pesan asli menjadi *ciphertext* atau pesan tersandi sedangkan dekripsi adalah proses yang dilakukan untuk mengubah pesan tersandi menjadi pesan yang dapat dibaca dan dimengerti. Saat ini banyak algoritma kriptografi yang digunakan untuk mengoptimasi algoritma-algoritma sebelumnya salah satunya adalah RC4+ yang merupakan salah satu variasi dari algorima RC4, khususnya untuk mewujudkan prinsip dari teknik kriptografi yaitu *diffusion* (mengaburkan) dan *confusion* (membingungkan)[3]. Algoritma ini menggunakan panjang kunci dari 1 sampai 256 putaran untuk mengenali sebanyak 256 array. Algorima ini juga melakukan pengacakan dengan proses *Key Schedulling Algorithm* (KSA) dan proses *Pseudo Random Generation Algorithm* (PRGA) serta melakukan pemindahan sejumlah bit-bit kunci untuk memperoleh kunci yang lebih acak. Nilai-nilai kunci inilah yang akan digunakan pada proses enkripsi dan dekripsi.

Tabel 1. Penelitian Terkait

No	Penulis	Judul	Kesimpulan
1	Hendri	Pengamanan Aplikasi Chatting Menggunakan Metode Kriptografi Government Standard[4]	Berdasarkan penelitian terdahulu disimpulkan bahwa aplikasi chatting dapat mengamankan pesan dengan melakukan proses enkripsi pesan sebelum dikirim dan melakukan proses dekripsi pada saat menerima pesan
2	Tasya Asprina, Muhammad Yamin, Sutardi	Pembangunan Aplikasi Keamanan Pesan Chatting Dengan Menerapkan Algoritma Tiny Encryption Algorithm (TEA) Berbasis Client Server[5]	Berdasarkan penelitian terdahulu disimpulkan bahwa pemanfaatan algoritma kriptografi dapat mengoptimalkan pengamanan terhadap pesan yang didistribuskan melalu aplikasi chatting
3	M A Budiman, Amalia dan N I Chayanie	An Implementation of RC4+ Algorithm and Zig-zag Algorithm in a Super Encryption Scheme for Text Security[6]	Berdasarkan penelitian terdahulu disimpulkan bahwa algoritma RC4+ memiliki kelebihan untuk menghasilkan kunci yang cukup acak

ISBN: 978-602-52720-1-1
Januari 2019
Hal: 916 - 920

2. METODOLOGI PENELITIAN

2.1 Kriptografi

Kriptografi adalah seni dan ilmu pengetahuan untuk memproteksi pengiriman data dengan mengubahnya menjadi kode tertentu dan hanya ditujukan kepada pengguna (*user*) yang hanya memiliki sebuah kunci untuk mengubah kode tersebut[7][8]. Dalam kriptografi terdapat dua konsep utama yakni enkripsi dan dekripsi. Enkripsi adalah proses dimana informasi atau data yang akan dikirim diubah menjadi bentuk yang hampir tidak dapat dikenali sebagai informasi awal dengan menggunakan algoritma tertentu sedangkan dekripsi merupakan proses pengembalian sandi menjadi teks asli.

2.2 Pesan Teks

Pesan teks merupakan suatu layanan yang memungkinkan pengguna telepon seluler untuk mengirimkan pesan singkat kepada pengguna telepon seluler lainnya dengan cepat dan dengan biaya yang kecil [9]. Pesan singkat yang akan di kirimkan juga jauh lebih cepat dari pada pengiriman pesan suara karena pesan teks hanya terdiri dari karakter teks dan angka.

2.3 Aplikasi *Chatting*

Aplikasi *Chatting* merupakan suatu perangkat lunak (*software*) yang memfasilitasi pengiriman pesan singkat antara dua *user* atau lebih. Salah satu kelebihan dari aplikasi *chatting* adalah kemampuannya memungkinkan pengguna untuk bertukar pesan baik dalam jenis teks, video, audio atau gambar[10]. Banyak fitur dari Aplikasi *Chatting* yang di kembangkan pada saat ini, tidak hanya berfungsi sebagai pengiriman teks saja tetapi sudah memiliki fitur seperti, *voice call* dan *video call* yang bisa dilakukan secara bersama-sama *(multiuser*). Fitur dar Aplikasi *Chatting* juga memungkinkan *user* (pengguna) untuk menyimpan daftar orang yang dapat diajak berkomunikasi secara mudah dan cepat.

2.4 Algoritma RC4+

Salah satu algoritma kunci simetris dalam bentuk *stream cipher* adalah algoritma RC4+. Dalam hal ini *stream cipher* sangat penting dan tidak dapat diabaikan pada aplikasi komputer. Oleh karena itu, saat ini model untuk desain *stream cipher* sangat diperlukan[3],[6]. Agar sandi pada algoritma RC4+ lebih kuat, maka ditambahkan beberapa operasi menggunakan struktur seperti RC4. Keamanan yang lebih baik dipeoleh dengan memanfaatkan poin yang ada pada RC4 kemudian memberikan beberapa fitur tambahan. Algoritma RC4+ memiliki dua bagian utama yaitu *Key Scheduling Algorithm* (KSA) dan *Pseudo Random Generation Algorithm* (PRGA).

Proses KSA (Key Schedulling Algorithm) :

```
F or i = 0 to 255
      S[i] := i
endfor

j := 0
for i from 0 to 255
      j := (j + S[i] + key[i mod keylength]) mod 256
      swap values of S[i] and S[j]
      j :=j
endfor
```

Proses PRGA (Pseudo Random Generation Algorithm)

```
i=0; j=0
For i = 0 to PanjangPlain-1
     i := i + 1
     a := S[i]
     j := (j + a) mod 256
     Swap S[i] and S[j]
     b := S[j]; S[i] := b; S[j] := a
     c := S[i<<5 ⊕ j>>3] + S[j<<5 ⊕ i>>3]
     z :=(S[a+b] + S[c⊕0xAA]) ⊕ S[j+b]
endFor
```

i dan j adalah indeks array 8-bit, << pergeseran ke kiri dan >> pergeseran ke kanan, ⊕ adalah eksklusif OR. Adapun proses enkripsi dan dekripsi pada algoritma RC4+ ini adalah sebagai berikut:

Formulasi Enkripsi :

$$Ci = Pi \oplus Ki \quad (1)$$

Formulasi Dekripsi :

$$Pi = Ci \oplus Ki \quad (2)$$

ISBN: 978-602-52720-1-1
Januari 2019
Hal: 916 - 920

3. ANALISA DAN PEMBAHASAN

Banyak permasalah yang ditemukan dalam proses pendistribusian pesan penting, salah satunya adalah aspek keamanan pesan yang sangat rawan terhadap penyadapan pesan oleh penyerang. Untuk mengatasi masalah tersebut di perlukan teknik keamanan dengan algoritma kriptografi salah satunya RC4+. Langkah yang tepat dalam mengamankan pesan teks yang akan dikirim yaitu dengan melakukan proses enkripsi untuk menyandikan pesan kemudian hasil enkripsi (*chipper*) dikirim ke penerima (*receiver*) untuk didekripsi agar *chiper* dari pesan kembali menjadi pesan asli (*plaintext*).

3.1 Proses Enkripsi Pesan

Proses enkripsi berdasarkan algoritma RC4+, meliputi :

1. Proses *Key Schedulling Algorithm* (KSA)
   Pembentukan Tabel S-Box, dilakukan berdasaran *pseudocode* dan menghasilkan array dengan nilai 0 sampai dengan 255, sehingga tabel S-Box :

Tabel 2. Tabel S-Box

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Tabel 3. Karakter kunci dibuat dalam bentuk array :

Index 0	Index 1	Index 2	Index 3	Index 4
S	T	M	I	K

Kemudian proses swap isi Tabel sesuai dengan *pseudocode*
Nilai mulai dari 0 hingga 255
Pada saat nilai i=0, maka

j= (j + S[i] + Key [i mod keylength]) Mod 256

j=(0+ S [0] + Key [0 mod 5]) mod 256
j=(0+0+Key [0]) Mod 256
j=(0 + 0 + 83) Mod 256
j=83 Mod 256
j=83
Nilai S[i] Swap dengan Nilai S[j]

Nilai S [0] ditukar (swap) dengan Nilai S [83]
Maka:
S[0]=83 dan S [83]=0

Lakukan proses swap hingga i=255

Setelah proses permutasi ini dilakukan hingga nilai i=255, maka dapat menyebabkan nilai array S-Box dapat di tukar secara berulang atau lebih dari satu kali. Hasil proses permutasi Tabel S-Box keseluruhan adalah:

Tabel 4. Hasil Permutasi S-Box

83	21	109	67	223	62	219	233	128	41	123	163	83	2	139	61
137	37	149	86	170	184	146	131	229	64	125	107	232	251	84	120
161	149	242	129	242	96	209	196	199	15	176	121	165	38	161	81
148	156	158	30	155	27	164	193	180	143	187	73	217	192	234	234
205	77	43	183	136	222	31	217	157	49	206	200	221	156	53	215
104	144	180	82	249	162	69	229	6	52	226	138	152	215	165	59
232	22	254	245	173	253	14	192	32	56	239	11	116	207	220	178
81	1	210	159	191	213	217	195	103	45	240	154	160	158	119	6
243	122	240	104	237	189	150	181	70	199	237	198	71	133	148	54
25	254	76	68	92	64	106	177	47	19	182	239	117	245	167	153

ISBN: 978-602-52720-1-1
Januari 2019
Hal: 916 - 920

141	10	102	84	172	147	55	39	174	97	20	239	135	249	179	101
185	190	209	176	255	75	0	13	230	250	33	37	44	60	244	90
99	111	252	246	164	186	203	24	113	23	42	226	95	8	228	168
206	241	180	212	130	216	57	100	16	171	208	94	46	88	127	249
220	17	18	108	155	211	183	65	114	166	227	34	91	145	202	12
80	142	201	7	78	151	218	26	93	169	247	63	132	204	29	112

2. Proses *Pseudo Random Generation Algorithm* (PRGA)
Proses PRGA menggunakan hasil permutasi S-Box pada Tabel 2 di atas. Proses ini dilakukan untuk menghasilkan *key stream* yang akan digunakan pada proses enkripsi ataupun dekripsi.
Proses iterasi pada PRGA dilakukan sebanyak jumlah karakter plainteks. Bila *plaintext* sampel yang digunakan di atas memiliki 9 nilai, maka iterasi proses PRGA akan di lakukan sebanyak 9 kali.
Nilai Biner dari hasil proses PRGA, akan di XOR-kan dengan nilai biner masing-masing plain sehingga dihasilkan *cipher*.

Palinteks = Budidarma
Cari Nilai i, a dan j :
Nilai i dan j dimulai dari 0
i = (i+j) Mod 256 →(0+1) Mod 256=1
a = S[i] →S [i]=21
j = (j+a) Mod 256 →(0+21) Mod 256=21
j = j → Nilai j selanjutnya adalah nilai j terakhir
Tukarkan nilai i, a dan j
b = S[i] →S [21] =184
S [i] = b →S [1] = 184
S [j] = a →S [21] = 21

Selanjutnya adalah menghitung nilai c dan z
Berdasarkan proses sebelumnya telah di tetapkan nilai i dan j adalah :
i = 1 dan j = 21
S [i] = b →S [1] = 184
S [j] = a →S [21] = 21

Cari nilai c dan z
$c = (S[((i<<5) \oplus (j>>3)) \bmod 2560 + S[((j<<5) \oplus (i>>3)) \bmod 256]) \bmod 256$
$= (S[((1<<5) \oplus (21>>3)) \text{ MOD } 256) + S[((21<<5) \oplus (1>>3)) \bmod 256]) \bmod 256$
$= (S[(32 \oplus 2) \bmod 256] + S[(160 \oplus 0) \bmod 256]) \bmod 256$
$= (S[34 \bmod 256] + S[160 \bmod 256]) \bmod 256$
$= (S[34] + S[160] \bmod 256$
$= (242 + 141) \bmod 256$
$= 383 \bmod 256$
$c = 127$
Berdasarkan proses sebelumnya telah di tetapkan nilai i dan j adalah :
i = 1 dan j = 21
a = S [i] →S [1]=21
j = (j+a) mod 256 →(0+21)mod 256=21
S [i] = b →S [1]=184
S [j] = a →S [21]=21
Nilai c = 127
Selanjutnya mencari nilai z :
$c = ((S[a+b] \bmod 256 + S[((c \oplus 00AA) \bmod 2560)] \oplus S[((j+b) \bmod 256]) \bmod 256$
$= ((S[21+184] \bmod 256 + S[((127 \oplus 170) \bmod 2560)] \oplus S[((21 + 184) \bmod 256]) \bmod 256$
$= ((S[205 \bmod 256 + S[((127 \oplus 170) \bmod 256)]) \oplus S[205 \bmod 256]) \bmod 256$
$= ((S[205] + S[((213 \bmod 256)]) \oplus S[205 \bmod 256]) \bmod 256$
$= ((S[205] + S[213]) \oplus S[205]) \bmod 256$
$= ((8 + 216) \oplus 8) \bmod 256$
$= (224 \oplus 8) \bmod 256$
$= 232 \bmod 256$
$z = 232$
Lakukan proses PRGA sebanyak karakter plainteks untuk mendapatkan kunci yang digunakan untuk mengenkripsi karakter plainteks lainnya.
Ssetelah didapatkan seluruh nilai kunci, maka proses enkripsi dilakukan berdasarkan persamaan 1.
Plainteks : Budidarma

ISBN: 978-602-52720-1-1
Januari 2019
Hal: 916 - 920

P [0] B = 66 → 01000010
K [0] = 232 → 11101000 ⊕
C[0] = 170 → 10101010 → a

3.2 Proses Dekripsi Pesan

Proses dekripsi meliputi proses KSA, proses PRGA untuk menghasilkan kunci dan proses dekripsi proses KSA dan PRGA pada saat dilakukan proses dekripsi sama seperti proses KSA dan PRGA pada saat melakukan enkripsi untuk memperoleh kunci dekripsi. Proses dekripsi ekripsi dilakukan dengan mengikuti persamaan 2.

4. KESIMPULAN

Berdasarkan pembahasan di atas, maka disimpulkan bahwa proses pengacakan KSA dan pergeseran bit pada proses enkripsi maupun dekripsi pada algoritma RC4+ sangat efektif untuk mempersulit bagi para penyadap baik dengan cara *know plain attack* maupun *cipher- only attack*. Proses yang dilakukan pada algoritma RC4+ ini sangat efektif dalam menyandikan pesan asli sehingga *ciphertext* yang dihasilkan sangat berbeda dengan *plaintext*.

REFERENCES

[1] I. Algoritma *et al.*, "ALGORITMA DES UNTUK ENKRIPSI DAN DESKRIPSI PESAN TEKS BERBASIS ANDROID," pp. 1–10.
[2] Gratia Vintana and Mardi Hardjianto, "Security Chatting Berbasis Desktop dengan Enkripsi Caesar Cipher Key Random," *J. TICOM*, vol. 5, no. 1, pp. 25–30, 2016.
[3] F. Akbar, H. Mawengkang, and S. Efendi, "Comparative analysis of RC4 + algorithm , RC4 NGG algorithm and RC4 GGHN algorithm on image file security," *2nd Nommensen Int. Conf. Technol. Eng.*, pp. 1–7, 2018.
[4] Hendri, "PENGAMANAN APLIKASI CHATTING MENGGUNAKAN METODE KRIPTOGRAFI GOVERNMENT STANDARD," *Maj. Ilm. INTI*, vol. 12, no. 3, pp. 295–300, 2017.
[5] T. Asprina, M. Yamin, and Sutardi, "PEMBANGUNAN APLIKASI KEAMANAN PESAN CHATTINGDENGAN MENERAPKAN ALGORITMA TINY ENCRYPTION ALGORITHM(TEA) BERBASIS CLIENT SERVER," *semanTIK*, vol. 4, no. 2, pp. 57–63, 2018.
[6] M. A. Budiman, Amalia, and N. I. Chayanie, "An Implementation of RC4+Algorithm and Zig-zag Algorithm in a Super Encryption Scheme for Text Security," *J. Phys. Conf. Ser.*, vol. 978, no. 1, 2018.
[7] D. Lombu, S. D. Tarihoran, and I. Gulo, "Kombinasi Mode Cipher Block Chaining Dengan Algoritma Triangle Chain Cipher Pada Penyandian Login Website," no. 1, pp. 1–11, 2018.
[8] T. Zebua, "Encoding the Record Database of Computer Based Test Exam Based on Spritz Algorithm," *Lontar Komput. J. Ilm. Teknol. Inf.*, vol. 9, no. 1, p. 52, May 2018.
[9] Y. Bin Pairin, "Kode Autentikasi Hash pada Pesan Teks Berbasis Android," *Eksplora Inform.*, vol. 8, no. 1, p. 6, 2018.
[10] T. Zebua, R. K. Hondro, and E. Ndruru, "Message Security on Chat App based on Massey Omura Algorithm," *IJISTECH (International J. Inf. Syst. Technol.*, vol. 1, no. 2, p. 16, 2018.