Perancangan Aplikasi Enkripsi Pada File Teks Menggunakan Algoritma Spritz

Syariah Kabeaken, Imam Saputra

Program Studi Teknik Informatika, STMIK Budi Darma, Medan, Indonesia

Email: syariah1995@gmail.com

Submitted 21-04-2020; Accepted 27-05-2020; Published 14-06-2020

Abstrak

Keamanan dari suatu file merupakan hal yang perlu diperhatikan dalam menjaga kerahasiaan informasi terutama yang berisi informasi yang hanya boleh diketahui isinya oleh pihak yang berhak saja. Pengiriman file atau informasi tanpa dilakukan pengamanan akan beresiko terhadap penyadapan dan informasi yang ada di dalamnya, dapat mudah diketahui oleh pihak-pihak yang tidak berhak. Salah satu hal penting dalam menjaga kerahasiaan dan keamanan file adalah dengan proses enkripsi. Maka berdasarkan teknik dalam kriptografi maka salah satu metode yang akan digunakan dalam mengamankan file teks dengan algoritma Spritz. Kriptografi modern lebih dipercaya oleh para ilmuwan daripada kriptografi klasik karena memiliki tingkat keamanan yang baik. Kriptografi klasik hanya melakukan pengacakan dari A-Z sedangkan kriptografi modern beroperasi dalam mode bit. Algoritma modern yang digunakan yaitu algoritma Spritz. Algoritma Spritz merupakan varian dari algoritma RC4, tingkat keamanan dari algoritma Spritz lebih baik dari RC4.

Kata Kunci: File, Teks, Algoritma, Spritz.

Abstract

The security of a file is a matter of concern in maintaining the confidentiality of information, especially those containing information that only the authorized parties may know about. Sending files or information without security will risk tapping and the information contained therein, can be easily known by unauthorized parties. One important thing in maintaining the confidentiality and security of files is the encryption process. Then based on techniques in cryptography, one of the methods that will be used in securing text files with the Spritz algorithm. Modern cryptography is more trusted by scientists than classical cryptography because it has a good level of security. Classical cryptography only does randomization from A-Z while modern cryptography operates in bit mode. The modern algorithm used is the Spritz algorithm. The Spritz algorithm is a variant of the RC4 algorithm, the security level of the Spritz algorithm is better than RC4.

Keywords: File, Text, Algorithm, Spritz.

1. PENDAHULUAN

Kriptografi pada awalnya dijabarkan sebagai ilmu yang mempelajari bagaimana menyembunyikan pesan. Namun pada pengertian modern kriptografi adalah ilmu yang bersandarkan pada teknik matematika untuk berurusan dengan keamanan informasi seperti kerahasiaan, keutuhan data dan otentikasi entitas. Jadi pengertian kriptografi modern adalah tidak saja berurusan hanya dengan penyembunyian pesan namun lebih pada sekumpulan teknik yang menyediakan keamanan informasi. Pada perkembangan selanjutnya, kriptografi berkembang menjadi sebuah disiplin ilmu sendiri karena teknik-teknik kriptografi dapat diformulasikan secara matematik sehingga menjadi sebuah metode formal. Salah satu cara untuk mempertahankan kerahasiaan dari teks tersebut, maka teks tersebut disandikan menjadi kode-kode yang tidak dipahami, sehingga bila ada pihak ketiga yang ingin merubah akan kesulitan dalam menterjemahkan isi teks yang sebenarnya dan harus membongkar enkripsinya terlebih dahulu.

File teks merupakan salah satu bentuk informasi yang berupa data berbentuk teks banyak format teks yang ada seperti doc, txt, docx, pdf. File yang berisi informasi-informasi dalam bentuk teks. Data yang berasal dari dokumen pengolahan kata, angka yang digunakan dalam perhitungan, nama dan alamat dalam basis data merupakan contoh masukan data teks yang terdiri dari karakter, angka dan tanda baca. Hanya dirinya dan pihak tertentu yang boleh mengetahuinya. Bahkan para petugas yang bekerja sebagai agen-agen rahasia pun mempunyai data-data rahasia. Perlu diketahui oleh pihak yang mempunyai data rahasia yang tidak boleh dibajak dan diketahui oleh orang lain. Dalam perakteknya, pencurian data dapat berwujud dalam pembacaan suatu file teks oleh pihak yang tidak berwenang, Memanifulasi data file teks, kerusakan data akibat buruknya keamanan sebuah file teks tersebut[1].

Keamanan dilakukannya penyandian file teks, untuk menghindari masalah tersebut perlu dilakukan proses menyembunyikan atau menyamarkan suatu informasi sedemikian rupa sehingga tidak jatuh kepada pihak lain yang tidak berwenang misalnya seperti data yang bersifat rahasia, data keuangan, seperti data password, pin, maka dilakukan lah proses penyandian file teks tersebut ke dalam bentuk yang tidak dapat dimengerti yaitu berupa kriptografi, Salah satunya dengan menggunakan metode Spritz. Spritz termasuk algoritma stream cipher. Stream cipher merupakan algoritma yang mengenkripsi pesan satu per satu karakter dengan menggunakan transformasi enkripsi yang bergantung pada waktu yang singkat. RC4 menggunakan panjang kunci dari 1 sampai 256 byte yang digunakan untuk menginisialisasikan tabel sepanjang 256 byte. Proses dalam menginisilasi pada Algoritma Spritz menggunakan dua substitusi Key Scheduling Algorithm (KSA) dan Pseudo-Random Generation Algorithm (PRGA). Sedangkan RC4 menggunakan dua buah S-Box yaitu array sepanjang 256 yang berisi permutasi dari bilangan 0 sampai 255, dan S-Box kedua, berisi permutasi merupakan fungsi dari kunci dengan panjang yang variabel.

2. METODE PENELITIAN

2.1 Kriptografi

Kriptografi pada awalnya dijabarkan sebagai ilmu yang mempelajari bagaimana menyembunyikan pesan. Namun pada pengertian modern kriptografi adalah ilmu yang bersandarkan pada teknik matematika untuk berurusan dengan keamanan informasi seperti kerahasiaan, keutuhan data dan otentikasi entitas.

Jadi pengertian kriptografi modern adalah tidak saja berurusan hanya dengan penyembunyian pesan namun lebih pada sekumpulan teknik yang menyediakan keamanan informasi. Kriptografi klasik umumnya merupakan teknik penyandian dengan kunci simetrik dan menyembunyikan pesan yang memiliki arti ke sebuah pesan yang nampaknya tidak memiliki arti dengan metode pergantian huruf (substitusi) atau transpose (pertukaran tempat)[2].

2.2 Algoritma Kriptografi Spritz

Algoritma Spritz merupakan varian dari algoritma RC4 didesain oleh Ron Rivest yang berasal dari RSA Security pada tahun 1987. Spritz sendiri mempunyai singkatan resmi yaitu "Rivest Cipher", namun juga dikenal sebagai "Ron's Code". RC4 sebenarnya dirahasiakan dan tidak dipublikasikan kepada khalayak ramai, namun ternyata ada orang yang tidak dikenal menyebarkan RC4 ke mailing list Cypherpunks. Kemudian berita ini dengan cepat diposkan ke sci.crypt newsgroup, dan dari newsgroup ini kemudian menyebar luas di internet. Kode yang dibocorkan tersebut dipastikan keasliannya karena output yang dikeluarkan sama dengan software-software yang menggunakan RC4 yang berlisensi. Spritz termasuk algoritma stream cipher. Stream cipher merupakan algoritma yang mengenkripsi pesan satu per satu karakter dengan menggunakan transformasi enkripsi yang bergantung pada waktu yang singkat. Algoritma Spritz termasuk algoritma kriptografi kunci simetrik. Algoritma ini memiliki dua bagian utama: Key Scheduling Algorithm (KSA) dan Pseudo-Random Generation Algorithm (PRGA)[1].

Adapun langkah-langkah enkripsi

1. Inisialisasi S-Box
Pada tahap ini S-Box akan diisi dengan nilai sesuai indeksnya untuk mendapatkan S-Box awal.
2. Menyimpan kunci ke Key Byte Array
Pada tahapan ini, kunci (key) yang akan digunakan untuk mengenkripsi akan dimasukkan ke dalam array berukuran 256 secara berulang sampai seluruh array terisi.
3. Permutasi pada S-Box
Pada tahap ini akan dibangkitkan sebuah nilai yang akan dijadikan aturan untuk permutasi pada S-Box.
4. Stream generation/Pseudo Random Generation Algorithm (PRGA)
Pada tahapan ini akan dihasilkan nilai pseudo random yang akan dikenakan operasi XOR untuk menghasilkan ciphertext.

Adapun langkah-langkah dekripsi

1. Lakukan Inisialisasi S-Box
Pada tahap ini S-Box akan diisi dengan nilai sesuai indeksnya untuk mendapatkan S-Box awal.
2. Menyimpan kunci ke key byte array
Pada tahapan ini, kunci (key) yang akan digunakan untuk dekripsi akan dimasukkan ke dalam array berukuran 256 secara berulang sampai seluruh array terisi.
3. Permutasi pada S-Box
Pada tahap ini akan dibangkitkan sebuah nilai yang akan dijadikan aturan untuk permutasi pada S-Box.
4. Stream generation/Pseudo Random Generation Algorithm (PRGA)
Pada tahapan ini akan dihasilkan nilai pseudo random yang akan dikenakan operasi XOR untuk menghasilkan plaintext.

2.3 File Teks

File teks merupakan file yang berisi informasi-informasi dalam bentuk teks. Data yang berasal dari dokumen pengolahan kata, angka yang digunakan dalam perhitungan, nama dan alamat dalam basis data merupakan contoh masukan data teks yang terdiri dari karakter, angka dan tanda baca. Masukan dan keluaran data teks direpresentasikan sebagai set karakter atau sistem kode yang dikenal oleh sistem komputer. Ada tiga macam set karakter yang umum digunakan untuk masukan dan keluaran pada komputer, yaitu ASCII, EBCDIC, dan Unicode[3].

3. HASIL DAN PEMBAHASAN

Solusi dalam mengamankan file teks ini adalah menggunakan teknik kriptografi. Adapun untuk penerapan teknik kriptografi ini menggunakan kunci metode sekaligus diantaranya metode *Spritz*. Karena proses file dari kunci yang dibuat berdasarkan kode ASCII dibutuhkan input plaintext atau penyandian menggunakan (*stream cipher*) dengan kunci rahasia/kunci simetri (kunci yang sama digunakan untuk proses enkripsi dan dekripsi).

Alasan karena teknik kriptografi *Spritz* merupakan salah satu teknik yang digunakan untuk meningkatkan aspek keamanan suatu informasi. Berdasarkan teknik penyandian *Spritz* dalam mengamankan file teks, diantaranya untuk menjaga suatu file teks agar isi file tersebut aman. Hasil proses enkripsi file teks akan menjadi sebuah ciphertext, sehingga file text tersebut tidak dapat terbaca.

3.1 Penerapan Algoritma *Spritz*

Tahapan Enkripsi file teks dengan Algoritma *spritz* sebelum melakukan Enkripsi dimulai *key scheduling* dengan menginisialisasikan *state* awal berupa larik yang terdiri dari 256 elemen, jadi hasil dari KSA adalah permutasi awal setelah itu kunci dimasukkan kedalam larik berukuran 256 dengan mengulang kunci hingga larik terisi penuh kemudian Kunci file diubah ke dalam kode ASCII. selanjutnya *Pseudo-Random Generation Algorithm (PRGA)* untuk Enkripsi Setelah melakukan tahap *key scheduling* maka akan dilakukan proses enkripsi file teks. Tahap ini menghasilkan nilai *pseudo-random* yang kemudian di- *XOR*-kan dengan *plainteks* untuk proses enkripsi atau dengan *ciphertext* pada proses dekripsi. Dalam penerapan kriptografi yang sedang penulis bahas yaitu mengenai pengamanan file teks dengan menggunakan algoritma *Spritz* berikut langkah-langkah penjabarannya :

Proses Enkripsi :

Dimisalkan file teks dengan nama “SASA” dan kunci “itu”, Dalam algoritma *spritz* ubah terlebih dahulu *kunci* ke ASCII dari setiap huruf.

Plainteks = SASA

Kunci = itu

Nilai ASCII untuk plainteks “SASA” yaitu :

S = 83

A = 65

S = 83

A = 65

Nilai ASCII untuk kunci “itu” yaitu :

i = 105

t = 116

u = 117

Tahap *Key Scheduling Algorithm (KSA)* Langkah *key scheduling* dimulai dengan menginisialisasikan *state* awal berupa larik yang terdiri dari 256 elemen. larik *state awal* berbentuk seperti tabel 1.

Tabel 1. Larik *S* awal

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	32	33	34	35	36	37	38	39	40	41
42	43	44	45	46	47	48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79	80	81	82	83
84	85	86	87	88	89	90	91	92	93	94	95	96	97
98	99	100	101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122	123	124	125
126	127	128	129	130	131	132	133	134	135	136	137	138	139
140	141	142	143	144	145	146	147	148	149	150	151	152	153
154	155	156	157	158	159	160	161	162	163	164	165	166	167
168	169	170	171	172	173	174	175	176	177	178	179	180	181
182	183	184	185	186	187	188	189	190	191	192	193	194	195
196	197	198	199	200	201	202	203	204	205	206	207	208	209
210	211	212	213	214	215	216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231	232	233	234	235	236	237
238	239	240	241	242	243	244	245	246	247	248	249	250	251
252	253	254	255										

Kemudian dilakukan perhitungan nilai *j* yang pertama dengan nilai awal *i* = 0 dan *j* = 0 sebagai berikut :

$$j = (j + S[i] + \text{key}[i \bmod \text{keylength}]) \bmod 256$$

$$j = (0 + 0 + 105) \bmod 256 = 105$$

Tukarkan nilai *S*[0] dengan *S*[105]. Kemudian dilakukan perhitungan kembali untuk nilai *j* yang kedua dengan nilai *i* = 1 dan *j* = 105 sebagai berikut :

$$j = (j + S[i] + \text{key}[i \bmod \text{keylength}]) \bmod 256$$

$$j = (105 + 1 + 116) \bmod 256 = 222$$

Tukarkan nilai *S*[1] dengan *S*[222]. Langkah ini diulangi hingga *i* mencapai nilai 255 dan hasil dari tahap *key scheduling* dapat dilihat pada tabel 2. dimana nilai *i* berada pada baris yang berwarna biru.

Tabel 2. Hasil *Key Scheduling Algorithm (KSA)*

0	1	2	3	4	5	6	7	8	9	10	11	12	13
105	220	155	14	202	122	50	157	125	64	215	138	33	100

14	15	16	17	18	19	20	21	22	23	24	25	26	27
223	7	83	22	166	169	12	142	197	243	35	62	88	177
28	29	30	31	32	33	34	35	36	37	38	39	40	41
198	11	59	175	227	42	194	56	117	159	207	98	234	156
42	43	44	45	46	47	48	49	50	51	52	53	54	55
119	206	6	224	32	225	186	27	170	162	74	99	147	141
56	57	58	59	60	61	62	63	64	65	66	67	68	69
217	70	249	23	73	250	91	46	140	75	69	252	203	107
70	71	72	73	74	75	76	77	78	79	80	81	82	83
171	148	150	52	36	137	105	136	97	104	112	19	9	151
84	85	86	87	88	89	90	91	92	93	94	95	96	97
20	195	21	199	45	232	17	173	123	115	167	212	94	81
98	99	100	101	102	103	104	105	106	107	108	109	110	111
153	244	13	161	172	145	110	40	208	189	103	114	163	214
112	113	114	115	116	117	118	119	120	121	122	123	124	125
48	160	218	213	216	158	253	131	38	204	174	144	24	26
126	127	128	129	130	131	132	133	134	135	136	137	138	139
239	226	101	102	240	89	54	44	132	168	49	82	121	8
140	141	142	143	144	145	146	147	148	149	150	151	152	153
154	111	5	109	39	1	229	93	106	245	96	84	248	0
154	155	156	157	158	159	160	161	162	163	164	165	166	167
242	85	3	190	184	192	165	183	79	139	90	66	68	146
168	169	170	171	172	173	174	175	176	177	178	179	180	181
126	80	34	116	182	25	67	237	176	95	221	196	238	43
182	183	184	185	186	187	188	189	190	191	192	193	194	195
219	187	149	241	135	28	72	77	30	211	55	2	29	210
196	197	198	199	200	201	202	203	204	205	206	207	208	209
209	58	201	178	127	65	57	181	191	129	180	15	220	228
210	211	212	213	214	215	216	217	218	219	220	221	222	223
31	251	188	233	92	10	113	133	118	78	200	230	231	193
224	225	226	227	228	229	230	231	232	233	234	235	236	237
87	37	41	130	222	152	51	128	236	225	254	61	4	124
238	239	240	241	242	243	244	245	246	247	248	249	250	251
76	246	71	47	247	164	16	179	235	108	185	53	205	134
252	253	254	255										
143	60	63	120										

Tahap *Pseudo-Random Generation Algorithm (PRGA)* untuk Enkripsi

Setelah melakukan tahap *key scheduling* maka akan dilakukan proses enkripsi pesan “SASA” dengan proses per karakter. Pada tahap awal inialisasikan nilai $i = 0$, $j = 0$, $k = 0$, $z = 0$, dan w adalah bilangan acak yang merupakan bilangan GCD atau relatif prima dengan panjang S yaitu 256. Selanjutnya lakukan proses seperti berikut:

Karakter “S”

Tahap awal dilakukan inialisasi $i = 0$, $j = 0$, $k = 0$ dan $z = 0$ dan setiap baris di modulo dengan 256. Kemudian dilakukan perhitungan dengan cara sebagai berikut :

$$i = i + w = 0 + 221 \text{ mod } 256 = 221$$

$$j = k + S[j + S[i]] = 0 + S[0 + S[221]] \text{ mod } 256$$

$$= 0 + S[0 + 230] \text{ mod } 256$$

$$= S[230] = 51$$

$$k = i + k + S[j] = 221 + 0 + S[51] \text{ mod } 256$$

$$= 221 + 162 \text{ mod } 256$$

$$= 127$$

$$S[i], S[j] = S[i] = S[221] = 230, S[j] = S[51] = 162$$

$$\text{Swap } S[i] \text{ with } S[j] = S[i] = S[221] = 162, S[j] = S[51] = 230$$

$$z = S[j + S[i + S[z + k]]] = S[51 + S[221 + S[0 + 127]]] \text{ mod } 256$$

$$= S[51 + S[221 + 226]] \text{ mod } 256$$

$$= S[51 + S[191]] \text{ mod } 256$$

$$= S[51 + 211] \text{ mod } 256$$

$$= S[6] = 50$$

Kemudian lakukan operasi XOR pada karakter pertama *ciphertext* yang dihasilkan sebelumnya yaitu karakter “S” yang bernilai 83 dengan output z yang dihasilkan adalah 50. Dengan cara sebagai berikut :

$$83 = 1010011$$

$50 = 0110010 \oplus$
 $= 1100001 = 97_{10}$ dalam tabel ASCII merupakan karakter “a”.

Dan seterusnya sehingga hasil chipertext dari proses enkripsi dengan Plainteks “SASA” adalah 97, 71, 36, 75 dalam tabel ASCII yaitu : aG\$K

Proses Dekripsi :

Ciphertext yang akan didekripsi adalah “aG\$K ” dimana akan dilakukan operasi XOR pada setiap karakter dengan cara sebagai berikut:

Karakter “a”

Tahap awal dilakukan inisialisasi $i = 0$, $j = 0$, $k = 0$, $z = 0$ dan w adalah bilangan acak yang merupakan bilangan GCD atau relatif prima dengan 256 dan setiap baris di modulo dengan 256.

$$\begin{aligned} i &= i + w = 0 + 221 \bmod 256 = 221 \\ j &= k + S[j + S[i]] = 0 + S[0 + S[221]] \bmod 256 \\ &= 0 + S[0 + 230] \bmod 256 \\ &= S[230] = 51 \\ k &= i + k + S[j] = 221 + 0 + S[51] \bmod 256 \\ &= 221 + 162 \bmod 256 \\ &= 127 \\ S[i], S[j] &= S[i] = S[221] = 230, S[j] = S[51] = 162 \\ \text{Swap}[S[i] \text{ with } S[j]] &= S[221] = 162, S[j] = S[51] = 230 \\ z &= S[j + S[i + S[z + k]]] = S[51 + S[221 + S[0 + 127]]] \bmod 256 \\ &= S[51 + S[221 + 226]] \bmod 256 \\ &= S[51 + S[191]] \bmod 256 \\ &= S[51 + 211] \bmod 256 = S[6] = 50 \end{aligned}$$

Kemudian lakukan operasi XOR pada karakter pertama *ciphertext* yang dihasilkan sebelumnya yaitu karakter “a” yang bernilai 97 dengan output z yang dihasilkan adalah 50. Dengan cara sebagai berikut :

$97 = 1100001$
 $50 = 0110010 \oplus$
 $= 1010011 = 83_{10}$ dalam tabel ASCII merupakan karakter “S”.

Dalam proses pendekripsian sama dengan langkah-langkah proses enkripsi. Seperti contoh diatas *ciphertext* yang didapat akan diproses kembali menjadi teks aslinya. Yaitu dalam tabel ASCII : 83, 65, 83, 65 menjadi “SASA”.

4. KESIMPULAN

Berdasarkan hasil studi literatur, analisis, perancangan, implementasi, dan pengujian sistem ini, maka kesimpulan yang didapat adalah sebagai berikut :

1. Teknik yang dilakukan dalam penyandian file teks dengan ekstensi (*.txt) yaitu dengan cara menerapkan Algoritma Spritz kedalamnya yang bisa mengubah data asli ke dalam bentuk data rahasia.
2. Pada penyandian file teks dengan menggunakan kode Algoritma spritz menggunakan array panjang kunci menjadi 256 byte dalam mengubah karakter menjadi chiperteks. Algoritma Spritz termasuk ke dalam kode aliran (stream cipher) dengan kunci rahasia/kunci simetri (kunci yang sama digunakan untuk proses enkripsi dan dekripsi).
3. Perancangan aplikasi yang digunakan pada perangkat lunak ini adalah dengan menggunakan program Visual Basic.Net 2008.

REFERENCES

- [1] R.L. Rivest and J.C.N. Schuld, 2014. Spritz A Spongy RC4-Like Stream Cipher And Hash Function. Schuld Research Institute For Secure System AIST, Japan.
- [2] Yuri ariyanto, 2009. Algoritma RC4 Dalam Proteksi Transmisi Dan Hasil Query Untuk ORDBMS POSTGRESQL. Jurnal Informatika
- [3] Muhammad Fauzan, E. P. 2012. Implementasi Algoritma Kriptografi Rc4 Pada Dsp Tms320c6713 Sebagai Pendukung Skuritas Jaringan Komunikasi Voice Over Internet Protocol (Voip). Jurnal EECCIS
- [4] Aryus, D. 2008, Teori Analisis Dan Implementasi, STMIK AMIKOM, Yogyakarta.
- [5] Novi Dian Natasham & Eko Wicaksono, 2011. Penerapan Teknik Kriptografi Stream Chiper Untuk Pengamanan Basis Data. Jurnal Basis Data, Ict Research Ceneter UNAS.
- [6] Sutardi, 2014. Implementasi Dan Analisis Kinerja Algoritma Shannnonfano Untuk Kompresi File Text. Jurnal Ilmiah Teknik Mesin. Universitas Halu Oleo Kampus Hijau Bumi Tridarma Andounohu Kendari.