SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Jurnal & Penelitian Teknik Informatika
Volume 3 Nomor 1, Oktober 2018
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Kombinasi Kriptografi RSA dengan *Linear Congruential Generator*

Muhammad Khoiruddin Harahap
Teknik Informatika
Politeknik Ganesha Medan
Choir.harahap@yahoo.com

Rina
Teknik Informatika
Politeknik Ganesha Medan
qweenarynna@gmail.com

***Abstract*—** Kriptografi menjadi sebuah teknik keamanan yang terus berkembang dan tak pernah selesai dalam pembahasannya. Keamanan terhadap data juga selalu menjadi perhatian yang sangat penting demi untuk menjaga kerahasiaan dan keamanan dari data tersebut. Dalam pembahasan ini bertujuan untuk menggabungkan dua buah algoritma yaitu pembangkit bilangan acak menggunakan metode Linear Congruential Generator untuk membuat tabel baru index bilangan ASCII yang kemudian dilanjutkan dengan Enkripsi / dekripsi menggunakan RSA. Melalui pembahasan tersebut didapatkanlah keamanan data menjadi 2 *layer*, yaitu layer pertama berdasarkan pengacakan code ASCII menjadi tabel baru dan Layer Kedua Enkripsi dan Dekripsi menggunakan metode RSA.

Keywords—*LCG; RSA; Kriptografi; Sekuriti.*

I. PENDAHULUAN

Keamanan data merupakan aspek yang sangat penting dalam pengiriman pesan terutama untuk pesan yang bersifat rahasia. Hal tersebut dapat kita lihat dari aktivitas sehari-hari seperti penggunaan internet untuk mengirimkan e-mail, sosial media, jual beli secara online, dan lain-lain. Untuk itu diperlukan suatu kode agar pesan tersebut masih bersifat rahasia dan aman, karena keamanan data yang dioperasionalkan pada jaringan publik rentan terhadap serangan oleh siapapun. Layanan keamanan data diwujudkan dengan menggunakan mekanisme keamanan data. Mekanisme keamanan data pada implementasinya menggunakan teknik-teknik penyandian, yaitu kriptografi.

Penelitian oleh Martha Monica yang membahas tentang perbandingan dari berbagai algoritma kriptografi yang dapat dimanfaatkan dalam menjaga keamanan informasi dalam sebuah smart card. Terdapat 3 algoritma yang dibandingkan dalam penelitian ini, yaitu algoritma El Gamal, RSA, dan DES. Hasilnya algoritma RSA yang paling cocok digunakan pada sebuah smart card dibandingkan algoritma El Gamal dan DES. Walaupun tingkat keamanan pada RSA tidak setinggi algoritma El Gamal, namun masih lebih aman dibanding DES . Resource yang dibutuhkan juga tidak sebesar algoritma El Gamal sehingga algoritma RSA dapat menjadi pilihan yang tepat untuk penjagaan keamanan informasi yang tersimpan pada smart card(Monica 2013). I Made Divya Biantara, I Made Sudana, Alfa Faridh Suni, suryono dan Arimaz Hangga dari jurusan Teknik Electro Universitas Negeri Semarang membahas tentang penerapan pengacakan soal pada ujian komputerisasi yang bertujuan untuk memberikan soal acak yang berbeda kepada setiap siswa dengan menggunakan metode Linear Congruential (LCG) dan bantuan matrik yang menjadi Coupe Linear Congruential Generator (CLCG). Kelebihan dari sistem tersebut adalah Metode modifikasi CLCG menghasilkan pengacakan yang lebih baik dan pola yang lebih rumit dibandingkan dengan metode LCG.(Suni et al. 2015)

Penelitian ini menggunakan algoritma kriptografi Rivest, Shamir, Adleman (RSA) yang mengimplementasikan sistem kriptografi kunci publik dan dikombinasikan dengan algoritma pembangkit bilangan acak linear congruental generator (LCG), dalam hal ini menggunakan Big Prime Number (bilangan prima yang besar) untuk pengiriman pesan rahasia. Berdasarkan latar belakang diatas, maka penulis tertarik untuk melakukan penelitian dengan judul "Implementasi Penyandian Data Hibridisasi Rivest Shamir Adleman (RSA) *Linear Congruental Generator* (LCG) dan Mersenne Big Prime Number**".**

Jurnal & Penelitian Teknik Informatika
Volume 3 Nomor 1, Oktober 2018
e-ISSN : 2541-2019
p-ISSN : 2541-044X

II. TINJAUAN PUSTAKA

2.1. Algoritma Kriptografi Asimetris

Algoritma asimetris (*asymmetric algorithm*) adalah suatu algoritma dimana kunci enkripsi yang digunakan tidak sama dengan kunci dekripsi. Pada algoritma ini menggunakan dua kunci yaitu kunci publik (public key) dan kunci privat (private key). Disebut kunci publik karena kunci yang digunakan pada proses enkripsi dapat diketahui oleh orang banyak atau disebarkan secara umum. Sedangkan kunci privat yang digunakan untuk proses dekripsi disimpan secara rahasia oleh si pengguna. Mengetahui kunci publik semata tidak cukup untuk menentukan kunci rahasia. walau kunci publik telah diketahui namun akan sangat sukar mengetahui kunci privat yang digunakan. Dalam sistem kriptografi kunci publik ini, proses enkripsi dan dekripsi menggunakan kunci yang berbeda, namun kedua kunci tersebut memiliki hubungan matematis karena itu disebut juga sistem asimetris.

2.2. Kriptografi Rivest Shamir Adleman (RSA)

Dalam kriptografi, RSA adalah algoritma untuk enkripsi kunci publik (*public-key encryption*). RSA merupakan algoritma kriptografi yang paling sering digunakan karena sangat sulit untuk dipecahkan. RSA adalah singkatan dari huruf depan 3 orang yang menemukan algoritmanya, pada tahun 1977 di MIT (Massachussets Institute of Technology) yaitu Ron Rivest, Adi Shamir dan Len Adleman. Algoritma ini adalah kriptografi kunci simetri (kunci – publik) yang kuat sampai saat ini. (Stallings 1995) Pada tahun 1977, Rivest, Shamir, Adleman merumuskan algoritma praktis yang mengimplementasikan sistem kriptografi kunci publik disebut dengan sistem kriptografi RSA. RSA adalah salah satu teknik kriptografi dimana kunci untuk melakukan enkripsi berbeda dengan kunci untuk melakukan dekripsi. Kunci untuk melakukan enkripsi disebut sebagai kunci publik, sedangkan kunci untuk melakukan dekripsi disebut sebagai kunci privat. Orang yang mempunyai kunci publik dapat melakukan enkripsi tetapi yang dalam melakukan dekripsi hanya orang yang memiliki kunci privat. Kunci publik dapat dimiliki oleh sembarang orang, tetapi kunci privat hanya dimiliki oleh orang tertentu saja.

Algoritma enkripsi dan dekripsi sistem kriptografi RSA bersandar pada asumsi fungsi satu arah (*one-way function*) yang dibangun oleh fungsi eksponensial modular pada grup perkalian (Z*n, x) dan grup perkalian (Z*ø(n), x) dengan

n = p x q. Dimana p, q adalah bilangan prima dan ø(n) = (p – 1) (q – 1).

Terdapat 3 algoritma pada sistem kriptografi RSA, yaitu algoritma pembangkitan kunci, algoritma enkripsi, dan algoritma dekripsi.

Untuk pembangkitan sepasang kunci RSA digunakan algoritma sebagai berikut :

1. Pilih dua bilangan prima yang besar, p dan q
2. Hitung n = p . q nilai n tidak dirahasiakan
3. Hitung Ø(n) = (p -1) (q-1)
4. Pilih sebuah bilangan bulat sebagai kunci publik, sebut *e*. Dimana *e* relatif prima terhadap Ø(n), artinya faktor pembagi terbesar keduanya adalah 1. secara matematis disebut gcd (e, Ø(n)) = 1.
5. Hitung kunci privat, sebut d, dengan persamaan e.d ≡ 1 (mod Ø(n)) atau d ≡ e-1 mod (Ø(n))

Hasil dari algoritma diatas adalah : kunci publik adalah pasangan (*e, n*) dan kunci privat adalah pasangan (*d, n*).

2.3. Algoritma Enkripsi dan Dekripsi RSA

a. Algoritma Enkripsi

Untuk algoritma enkripsi menggunakan RSA adalah sebagai berikut :

1. Ambil kunci publik penerima pesan e dan modulus n.
2. Nyatakan pesan menjadi blok – blok plainteks : m1, m2, m3, ... (syarat : 0 < mi < n – 1)
3. Hitung blok cipherteks C, dengan persamaan Ci = m . i. e mod n . dalam hal ini e adalah kunci publik.

b. Algoritma Dekripsi

Untuk algoritma dekripsi menggunakan RSA adalah sebagai berikut :

Setiap blok cipherteks Ci didekripsi kembali menjadi blok mi dengan persamaan mi = c . i . d mod n. dalam hal ini d adalah kunci privat.

2.4. Properti Algoritma RSA

Besaran-besaran yang digunakan pada algoritma RSA :

- p dan q bilangan prima (rahasia)
- *n* = p . q (tidak rahasia)
- Ø(*n*) = (p – 1)(q – 1) (rahasia)
- e (kunci enkripsi) (tidak rahasia)
- d (kunci dekripsi) (rahasia)
- m (plainteks) (rahasia)
- c (cipherteks) (tidak rahasia)

2.5. Linear Congruential Generator (LCG)

Linear congruential generator merupakan algoritma pembangkit bilangan acak kongruen. Algoritma *pseudo random number* yang paling populer(Clawdia, Khairina, and Harahap 2017). Algoritma ini mudah dipahami dan dapat diimplementasikan secara cepat. Keuntungan dari LCG adalah operasinya yang sangat cepat. LCG dapat diterapkan untuk menghasilkan nilai acak atau digunakan untuk mengacak posisi dari sekumpulan nilai, dimana bilangan acak tersebut muncul berdasarkan rumus aritmatika yang sudah ditetapkan. LCG didefinisikan dalam relasi rekurens (Schneier 1996) dengan rumus :

$Z_i = (a * Z_{i-1} + b) \bmod m$

Keterangan :

Z_i = bilangan acak ke i dari deretnya
Z_{i-1} = bilangan acak sebelumnya
a = faktor pengali
b = penambah (increment)
m = modulus (a, b, dan m semuanya konstans)(Apdilah et al. 2018)

LCG mempunyai periode tidak lebih besar dari m, dan kebanyakan kasus periodenya kurang dari m, maksudnya adalah deret bilangan acak yang dihasilkan tidak lebih banyak dari modulonya. LCG mempuyai periode penuh (m – 1) jika memenuhi syarat berikut (Munir 2006):

- b relative prima terhadap m
- a – 1 dapat dibagi dengan semua faktor prima dari m
- a – 1 adalah kelipatan 4, jika m adalah kelipatan 4
- m > maks (a, b, X_0)
- a > 0, b > 0

X_0 adalah kunci pembangkit atau disebut juga umpan (*seed*). secara teori LCG mampu menghasilkan bilangan acak, namun sensitif terhadap pemilihan nilai-nilai a, b, dan m. Pemilihan nilai-nilai yang tidak sesuai dapat mempengaruhi implementasi pada LCG. LCG memiliki kelebihan pada kecepatannya karena sedikit membutuhkan operasi bit. namun kemunculan bilangan acaknya mudah diprediksi sehingga tidak aman secara kriptografi, namun demikian LCG tetap berguna untuk latihan awal penerapan enkripsi dengan metode stream cipher menggunakan kunci yang dibangkitkan oleh algoritma LCG.

III. PEMBAHASAN

Langkah yang dilakukan pada pembahasan ini adalah yang pertama dilakukan yaitu membentuk sebuah bilangan urutan bilangan acak yang dikombinasi dengan membentuk deret yang acak dari ASCII code sebanyak 256 huruf. Dengan menggunakan CLG maka dibentuklah deret yang baru seperti tabel berikut ini dengan nilai nilai a = 1, b = 7, m = 256 dan Z_0 = 12.

Index	asc index	Char	z1	Index	asc	Char	z1
1	0	#value!	19	69	68	D	239
2	1		26	70	69	E	246
3	2		33	71	70	F	253
4	3		40	72	71	G	4
5	4		47	73	72	H	11
6	5		54	74	73	I	18
7	6		61	75	74	J	25
8	7		68	76	75	K	32
9	8		75	77	76	L	39
10	9		82	78	77	M	46
11	10		89	79	78	N	53
12	11		96	80	79	O	60
13	12		103	81	80	P	67
14	13		110	82	81	Q	74
15	14		117	83	82	R	81
16	15		124	84	83	S	88
17	16		131	85	84	T	95
18	17		138	86	85	U	102
19	18		145	87	86	V	109
20	19		152	88	87	W	116
21	20		159	89	88	X	123
22	21		166	90	89	Y	130
23	22		173	91	90	Z	137
24	23		180	92	91	[	144
25	24		187	93	92	\	151
26	25		194	94	93	]	158
27	26		201	95	94	^	165
28	27		208	96	95	_	172
29	28		215	97	96	`	179
30	29		222	98	97	A	186
31	30	-	229	99	98	B	193
32	31		236	100	99	C	200
33	32		243	101	100	D	207
34	33	!	250	102	101	E	214
35	34	"	1	103	102	F	221
36	35	#	8	104	103	G	228
37	36	$	15	105	104	H	235
38	37	%	22	106	105	I	242
39	38	&	29	107	106	J	249
40	39	'	36	108	107	K	0
41	40	(	43	109	108	L	7
42	41	)	50	110	109	M	14
43	42	*	57	111	110	N	21
44	43	+	64	112	111	O	28
45	44	,	71	113	112	P	35
46	45	-	78	114	113	Q	42
47	46	.	85	115	114	R	49
48	47	/	92	116	115	S	56
49	48	0	99	117	116	T	63
50	49	1	106	118	117	U	70
51	50	2	113	119	118	V	77
52	51	3	120	120	119	W	84
53	52	4	127	121	120	X	91
54	53	5	134	122	121	Y	98
55	54	6	141	123	122	Z	105
56	55	7	148	124	123	{	112
57	56	8	155	125	124	|	119
58	57	9	162	126	125	}	126
59	58	:	169	127	126	~	133
60	59	;	176	128	127	□	140
61	60	<	183	129	128	€	147
62	61	=	190	130	129	•	154
63	62	>	197	131	130	‚	161
64	63	?	204	132	131	Ƒ	168

65	64	@	211	133	132	„	175
66	65	A	218	134	133	…	182
67	66	B	225	135	134	†	189
68	67	C	232	136	135	‡	196
137	136	ˆ	203	214	213	Õ	230
138	137	‰	210	215	214	Ö	237
139	138	Š	217	216	215	×	244
140	139	‹	224	217	216	Ø	251
141	140	Œ	231	218	217	Ù	2
142	141	•	238	219	218	Ú	9
143	142	Ž	245	220	219	Û	16
144	143	•	252	221	220	Ü	23
145	144	•	3	222	221	Ý	30
146	145	‘	10	223	222	Þ	37
147	146	’	17	224	223	ß	44
148	147	“	24	225	224	À	51
149	148	”	31	226	225	Á	58
150	149	•	38	227	226	Â	65
151	150	–	45	228	227	Ã	72
152	151	—	52	229	228	Ä	79
153	152	˜	59	230	229	Å	86
154	153	™	66	231	230	Æ	93
155	154	Š	73	232	231	Ç	100
156	155	›	80	233	232	È	107
157	156	Œ	87	234	233	É	114
158	157	•	94	235	234	Ê	121
159	158	Ž	101	236	235	Ë	128
160	159	Ÿ	108	237	236	Ì	135
161	160		115	238	237	Í	142
162	161	¡	122	239	238	Î	149
163	162	¢	129	240	239	Ï	156
164	163	£	136	241	240	Ð	163
165	164	¤	143	242	241	Ñ	170
166	165	¥	150	243	242	Ò	177
167	166	¦	157	244	243	Ó	184
168	167	§	164	245	244	Ô	191
169	168	¨	171	246	245	Õ	198
170	169	©	178	247	246	Ö	205
171	170	ª	185	248	247	÷	212
172	171	«	192	249	248	Ø	219
173	172	¬	199	250	249	Ù	226
174	173		206	251	250	Ú	233
175	174	®	213	252	251	Û	240
176	175	¯	220	253	252	Ü	247
177	176	°	227	254	253	Ý	254
178	177	±	234	255	254	Þ	5
179	178	²	241	256	255	Ÿ	12
180	179	³	248				
181	180	´	255				
182	181	µ	6				
183	182	¶	13				
184	183	·	20				
185	184	¸	27				
186	185	¹	34				
187	186	º	41				
188	187	»	48				
189	188	¼	55				
190	189	½	62				
191	190	¾	69				
192	191	¿	76				
193	192	À	83				
194	193	Á	90				
195	194	Â	97				
196	195	Ã	104				
197	196	Ä	111				
198	197	Å	118				
199	198	Æ	125				
200	199	Ç	132				
201	200	È	139				
202	201	É	146				
203	202	Ê	153				
204	203	Ë	160				
205	204	Ì	167				
206	205	Í	174				
207	206	Î	181				
208	207	Ï	188				
209	208	Ð	195				
210	209	Ñ	202				
211	210	Ò	209				
212	211	Ó	216				
213	212	Ô	223				

Berdasarkan tabel di atas, maka didapatkanlah index yang baru untuk urutan ascii sebagai berikut :

CHR	INDX	CHR	INDX	CHR	INDX	CHR	INDX	CHR	INDX
K	0		68	£	136	7	148	Ó	216
"	1	¾	69	Z	137	Î	149	Š	217
Ù	2	U	70		138	¥	150	A	218
•	3	,	71	È	139	\	151	Ø	219
G	4	Ã	72	□	140		152	¯	220
Þ	5	Š	73	6	141	Ê	153	F	221
µ	6	Q	74	Í	142	•	154		222
L	7		75	¤	143	8	155	Ô	223
#	8	¿	76	[	144	Ï	156	‹	224
Ú	9	V	77		145	¦	157	B	225
‘	10	-	78	É	146	]	158	Ù	226
H	11	Ä	79	€	147		159	°	227
Ÿ	12	›	80	7	148	Ë	160	G	228
¶	13	R	81	Î	149	¸	161	-	229
M	14		82	¥	150	9	162	Õ	230
Ş	15	À	83	\	151	Ð	163	Œ	231
Û	16	W	84		152	§	164	C	232
’	17	.	85	Ê	153	^	165	Ú	233
I	18	Å	86	•	154		166	±	234
#value!	19	Œ	87	8	155	Ì	167	H	235
·	20	S	88	Ï	156	Ƒ	168		236
N	21		89	¦	157	:	169	Ö	237
%	22	Á	90	]	158	Ñ	170	•	238
Ü	23	X	91		159	¨	171	D	239
“	24	/	92	Ë	160	_	172	Û	240
J	25	Æ	93	,	161		173	²	241
	26	•	94	9	162	Í	174	I	242
¸	27	T	95	Đ	163	„	175		243
O	28		96	§	164	;	176	×	244
&	29	Â	97	^	165	Ò	177	Ž	245
Ý	30	Y	98		166	©	178	E	246
”	31	0	99	Ì	167	`	179	Ü	247
K	32	Ç	100	Ƒ	168		180	³	248
	33	Ž	101	:	169	Î	181	J	249
¹	34	U	102	Ñ	170	…	182	!	250
P	35		103	¨	171	<	183	Ø	251
'	36	Ã	104	_	172	Ó	184	•	252
Þ	37	Z	105		173	ª	185	F	253
•	38	1	106	Í	174	A	186	Ý	254
L	39	È	107	„	175		187	´	255
	40	Ÿ	108	;	176	Ï	188		
º	41	V	109	Ò	177	†	189		
Q	42		110	©	178	=	190		
(	43	Ä	111	`	179	Ô	191		
ß	44	{	112		180	«	192		
–	45	2	113	Î	181	B	193		
M	46	É	114	…	182		194		
	47		115	<	183	Đ	195		
»	48	W	116	Ó	184	‡	196		
R	49		117	ª	185	>	197		
)	50	Å	118	A	186	Õ	198		
À	51	|	119		187	¬	199		
—	52	3	120	Ï	188	C	200		
N	53	Ê	121	†	189		201		
	54	¡	122	=	190	Ñ	202		
¼	55	X	123	Ô	191	ˆ	203		
S	56		124	£	136	?	204		

*	57	Æ	125	Z	137	Ö	205
Á	58	}	126		138		206
~	59	4	127	È	139	D	207
O	60	Ë	128	□	140		208
	61	¢	129	6	141	Ò	209
½	62	Y	130	Í	142	‰	210
T	63		131	¤	143	@	211
+	64	Ç	132	[	144	÷	212
Â	65	~	133		145	®	213
™	66	5	134	É	146	E	214
P	67	ì	135	€	147		215

Berdasarkan tabel yang baru terbentuk maka dilakukanlah proses enkripsi dan dekripsi menggunakan original RSA dan contoh dapat dilihat sebagai berikut :

Sebagai contoh dalam mengimplementasikan diatas adalah sebagai berikut :

Plaintext : RIN
Plaintext di LCG menghasilkan : , š , - dengan ascii code masing masing = 81, 18, 53

Hasil Enkripsi :
Ambil nilai p = 11 dan q = 19, maka didapatkanlah Enkripsi = 16, 94, 70

Dan pada proses dekripsi akan mengembalikan ke nilai Dekripsi = 81, 18, 53

IV. KESIMPULAN

Berdasarkan pembahasan di atas, maka dapat ditarik kesimpulan bahwa penggabungan LCG dengan RSA dapat membuat sekuritas dari data menjadi 2 *layer*, dimana layer pertama bahwa plaintext terlebih dahulu diacak kemudian dilanjutkan hasil acak tersebut di lakukan proses Enkripsi, dan juga sebaliknya, hasil enkripsi di dekripsi untuk kemudian di lakukan proses LCG lagi untuk mengembalikan ke plaintext semua.

REFERENCES

Apdilah, D., M.K. Harahap, N. Khairina, A.M. Husein, and M. Harahap. 2018. "A Comparison of One Time Pad Random Key Generation Using Linear Congruential Generator and Quadratic Congruential Generator." *Journal of Physics: Conference Series* 1007 (1). https://doi.org/10.1088/1742-6596/1007/1/012006.

Clawdia, Jhessica, Nurul Khairina, and Muhammad Khoiruddin Harahap. 2017. "Implementasi Algoritma Kriptografi One Time Pad (Otp) Dengan Dynamic Key Linear Congruential Generator (Lcg)." *KOMIK (Konferensi Nasional Teknologi Informasi Dan Komputer)* I: 12–14.

Monica, Martha. 2013. "Pemanfaatan Algoritma Kriptografi Dalam Pembuatan Smart Card." *Makalah IF3058 Kriptografi– Semester II Tahun 2012/2013*, no. 13510080.

Munir, Rinaldi. 2006. *Kriptografi*.

Schneier, Bruce. 1996. *Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C (Cloth)*.

Stallings, William. 1995. *Cryptography and Network Security (4th Edition)*.

Suni, Alfa Faridh, Arimaz Hangga, I Made Sudana, and I Made Diyya Biyantara. 2015. "Modifikasi Metode Linear Congruential Generator." *SemNaSIF* 2015 (November): 182–86.