JURIKOM (Jurnal Riset Komputer), Vol. 8 No. 1, Februari 2021
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v7i5.2541
Hal 20−27

Perancangan Aplikasi Keamanan Data Dengan Kombinasi Algoritma Kriptografi RC4 dan One Time Pad

Irfan Anas, Guidio Leonarde Ginting, Eferoni Ndruru, Abdul Sani Sembiring, Taronisokhi Zebua

Teknik Informatika, STMIK Budi Darma, Medan, Indonesia
Email: uzumakirfan09@gmail.com
Submitted **19-10-2020**; Accepted **27-10-2020**; Published **25-02-2021**

Abstrak

Data merupakan aspek penting yang memiliki informasi yang penting di dalamnya, yang terdapat berupa angka, karakter, simbol, gambar, suara, atau tanda tanda lain yang dapat untuk dijadikan informasi. Data merupakan suatu fakta-fakta tertentu, sehingga menghasilkan suatu informasi. Kriptografi sering digunakan untuk mengamankan dan menyandikan data teks, pesan, atau pun informasi lainnya, maka dengan menggunakan algoritma kriptografi pada data teks mampu mengamankan data yang sangant penting bagi setiap orang. Maka hasil yang diharapkan dari algoritma Kriptografi RC4 dan One Time Pad, yang digunakan pada data teks dapat membantu dalam mengamankan atau menyandikan data teks yang sangat penting, sehinnga teks terebut tidak mudah untuk dibaca atau di ketahui pesan aslinya.

Kata Kunci: Kriptografi; Keamanan; RC4; OTP

Abstract

Data is an important aspect that has important information in it, which includes numbers, characters, symbols, pictures, sounds, or other signs that can be used as information. Data are certain facts, thus producing information. Cryptography is often used to secure and encode text data, messages, or other information, so using cryptographic algorithms on text data is able to secure data that is very important to everyone. So the expected results from the RC4 Cryptography algorithm and One Time Pad, which are used in text data can help in securing or encoding very important text data, so that the text is not easy to read or know the original message.

Keywords: Cryptography; Security; RC4; OTP

1. PENDAHULUAN

Data merupakan aspek penting yang memiliki informasi penting di dalamnya. Data juga merupakan fakta mentah atau rincian peristiwa yang belum diolah, sehingga data harus diolah terlebih dahulu menjadi informasi untuk dapat diterima oleh penerima. Data bisa berupa angka, karakter, simbol, gambar, suara, atau tanda tanda lain yang dapat untuk dijadikan informasi. Data juga merupakan catatan atas kumpulan fakta, data adalah hasil pengukuran atau pengamatan yang bentuknya dapat berupa angka, kata-kata, atau citra. Sehingga disimpulkan data merupakan suatu fakta-fakta tertentu, sehingga menghasilkan suatu data dalam menarik suatu keputusan.

Keamanan dan integritas data adalah sesuatu yang harus diperhatikan. Keamanan dan kerahasian suatu data menjadi isu yang penting dan terus berkembang. Beberapa kasus yang melibatkan keamanan data sekarang merupakan perkerjaan yang membutuhkan biaya untuk penanganannya dan keamanan yang sangat banyak untuk menjaganya. Untuk menjaga informasi pada data tersebut tidak terjerumus kepada tangan orang-orang yang tidak berwewenang maka diperlukannya mekanisme keamanan yang baik. Teknologi juga bisa digunakan dalam hal kejahatan, seperti halnya para peretas yang suka mengubah, mengunci, menyadap, menghapus dan juga memodifikasi isi data seseorang. Dalam menjaga keamanan data dan juga kerahasiaan data terutama bagi perusahaan, atau suatu organisasi yang mempunyai dokumen-dokumen rahasia dan penting. Untuk menjaga keamanan dan kerahsiaan pesan, data atau informasi supaya tidak bisa dibaca atau dipahami oleh siapapun, kecuali untuk penerima yang berhak atas data tersebut, dengan begitu penerapan sistem keselamatan dirancang dengan algoritma kriptografi [1]. Dengan teknologi yang sekarang, apapun dapat dilakukan untuk melakukan peretasan terhadap suatu data. Setiap pengguna memiliki data privasi yang tidak semua orang boleh mengetahuinya. Terlebih lagi di era globalisasi zaman sekarang, untuk mendapatkan sebuah informasi sangat mudah, mudah bagi setiap orang untuk mendapatkan data yang diinginkan. Kemudahan dalam mendapatkan informasi juga memberikan beberapa ancaman seperti pada password, data kerahasiaan perusahaan atau instansi[2]. Banyak cara yang dilakukan untuk mengamankan data dari ancaman pihak luar yang tidak memiliki hak untuk mengolah data dokumen tersebut, dan tidak sedikit juga sesorang yang datanya di baca yang bukan merupakan haknya. Maka dibutuhkan cara untuk meningkatkan keamanan pada data terhadap dokumen teresbut yaitu dengan kriptografi.

Ada banyak algoritma kriptografi yang telah diciptakan oleh para peneliti kriptografi, namun dalam penelitian berikut yang akan membahas mengenai Algoritma One Time Pad (OTP) dan Rivest Cipher 4 (RC4). Algoritma RC4 disebut sebagai stream cipher, algoritma RC4 melakukan proses enkripsi dan deskripsi secara satu persatu berdasarkan kunci yang telah dibangkitkan sebelumnya. Ada tiga proses utama pada algoritma RC4 yaitu proses Key Schedulling Algorithm (KSA), Pseudo Random Generation Algorithm (PRGA) dan juga proses enkripsi dan deskripsi[3]. Algoritma One Time Pad juga disebut sebagai stream cipher yang melakukan proses enkripsi dan deskrpsi dengan satu karakter setiap kali proses. One Time Pad merupakan perbaikan dari algoritma vernam cipher untuk menghasilkan keamanan yang sempurna[4]. Jadi kedua algoritma tersebut memiliki kesamaan, yaitu jenis stream cipher dan proses pada satu karakter tiap enkripsi dan deskripsi. Kedua algoritma tersebut memiliki kelebihan dan kekurangan dan juga tingkat kerumitan yang berbeda.

Penelitian tentang kombinasi algoritma kriptografi RC4 dan OTP diketahui belum pernah dilakukan tetapi kedua algoritma tersebut dengan algoritma yang lainnya pernah dilakukan oleh Wahyu Hari Haji dan Slamet Mulyono yang

JURIKOM (Jurnal Riset Komputer), Vol. 8 No. 1, Februari 2021
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v7i5.2541
Hal 20−27

berjudul "Implementasi RC4 Stream Cipher untuk keamanan basis data" dengan tujuannya yaitu "menerapkan algoritma RC4 pada basis data untuk menjaga kerahasiaan dan keamanan pada basisi data tersebut dengan kesimpulan Data yang diinput akan tersimpan pada database dalam keadaan terenkripsi sehingga keamanan datanya dapat terjaga." Sedangkan algoritma yang dilakukan oleh Hasrul Hasrul dan Lamro Herianto Siregar dengan judul "Penerapan Teknik Kriptografi Pada Database Menggunakan Algoritma One Time Pad" dengan tujuannya yaitu "untuk mengamankan database dengan cara menggunakan algoritma One Time Pad agar pihak yang tidak bertanggung jawab tidak dapat mengubah, mengambil, atau menyalahgunakan data dan informasi tersebut". Dan kedua penelitian tersebut, penulis menyimpulkan bahwa algoritma kriptografi RC4 dan OTP memiliki tingkat keamanan yang kuat dalam mengamankan data. Algoritma One Time Pad berisi deretan karakter kunci yang dibangkitkan secara acak. Sistem OTP menggunakan barisan kunci acak ditambah teks-asli yang tidak acak maka menghasilkan teks-kode yang seluruhnya acak. Sedangkan algoritma RC4 merupakan jenis aliran kode yang berarti operasi enkripsinya dilakukan perkarakter 1 byte untuk sekali operasi [5]. Algoritma RC4 digunakan pembangkit kunci untuk OTP, lalu hasil pembangkit kunci digunakan algoritma OTP untuk menghasilkan sebuah ciphertext pada plainteks.

2. METODOLOGI PENELITIAN

2.1 Aplikasi

Aplikasi merupakan sebuah *software* (perangkat lunak) yang bertugas sebagai *front end* pada sebuah sistem yang dipakai untuk mengelolah berbagai macam data sehingga menjadi sebuah informasi yang bermanfaat untuk penggunanya dan juga sistem yang berkaitan [7]. Aplikasi merupakan alat bantu yang dapat dijalankan pada sebuah komputer atau smartphone yang membantu memudahkan penggunanya dalam mengelolah data.

2.2 Keamanan Data

Keamanan merupakan salah satu aspek tepenting dari sebuah sistem informasi. Masalah keamanan sering kuranf mendapat perhatian dari para perancang dan pengelola dari sistem informasi. Masalah keamanan ering berada diurutan setelah tampilan, atau bahkan diurutan terakhir dalam daftar hal-hal yand dianggap penting. Beberapa ancaman dan serangan yang harus diperhatikan dalam sistem keamanan data adalah sebagai berikut :

1. *Interruption* merupakan ancaman terhadap Avability informasi, data yang ada pada komputer dirusak atau dihapus sehingga jika data atau informasi tersebut dibutuhkan maka pemiliknya akan mengalami kesulitan untuk mengaksesnya, bahkan mungkin informasi itu hilang.
2. *Interception* merupakan ancaman terhadap kerahasian (*secrecy*). Informasi disadap sehingga orang yang tidak berhak dapat mengakses komputer dimana informasi tersebut disimpan.
3. *Modification* merupakan ancaman terhadap integritas. Oran tidak berhak berhasil menyadap lalu lintas informasi yang sedang dikirim dan kemudian mengubahnya sesuai keinginan orang tersebut.
4. *Febrication* merupakan ancaman terhadap integritas. Orang yang tidak berhak berhasil meniru atau memalsukan informasi sehingga orang yang menerima informasi tersebut menyangka bahwa informasi tersebut berasal dari orang yang dikehendaki oleh si penerima informasi [5].

2.3 Kriptografi

Kriptografi berasal dari bahasa Yunani, *crypto* dan *graphia*. Crypto berarti secret (rahasia) dan graphia berarti writing (tulisan). Menurut terminologinya, kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat lain [5].
Secara umum, istilah yang sering digunakan didalam kriptografi adalah:

1. *Enkripsi*: merupakan hal yang sangat penting dalam kriptografi, merupakan pengamanan data yang dikirimkan agar terjaga kerahasiannya. Pesan asli disebut plainteks, yang diubah menjadi kode-kode yang tidak dimengerti. Enkripsi bisa diartikan dengan cipher atau kode. Sama halnya dengan kita tidak mengerti akan sebuah kata maka kita akan melihatnya di dalam kamus atau daftar istilah. Beda halnya dengan enkripsi, untuk mengubah teks-asli ke bentuk teks-kode kita menggunakan algoritma yang dapat mengkodekan data yang kita inginkan.
2. *Deskripsi*: merupakan kebalikan dari enkripsi. Pesan yang telah dienkripsi dikembalikan kedalam bentuk asalnya (teks-asli), disebut dengan deskripsi pesan. Algoritma yang digunakan untuk deskripsi tentu berbeda dengan algoritma yang digunakan untuk enkripsi.
3. *Kunci*: yang dimaksud disini adalah kunci yang dipakai untuk melakukan enkripsi dan deskripsi. Kunci terbagi menjadi dua bagian, kunci rahasia (*private key*) dan kunci umum (*puclic key*) [6].

2.4 Algoritma RC4

RC4 merupakan jenis dari aliran kode yang yang berarti operasi enkripsinya dilakukan per karakter 1 byte untuk sekali operasi. Secara resmi RC4 tidak pernah dipublikasikan, tetapi pada bulan september tahun 1994 seseorang yang tidak dikenal mengirim detil algoritmanya ke internet sehingga di ketahui publik. RC4 memiliki panjang kunci 2048 bit (256 byte), namun yang biasa digunakan hanya 40 bit atau 128 bit, sisanya digunakan untuk perulangan kunci yang dipakai. Algoritma RC4

JURIKOM (Jurnal Riset Komputer), Vol. 8 No. 1, Februari 2021
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v7i5.2541
Hal 20−27

memakai kotak-S (kotak subtitusi) dengan larik 256 byte dengan ukuran 8 x 8. Walaupun RC4 menggunakan kotak–S, tetapi operasi yang terjadi didalam kotak –S adalah operasi permutasi.
Proses inisialisasi S-box (*array* S):
For r = 0 to 255
S[r] = r
Lakukan permutasi terhadap nilai-nilai didalam *array*-S dengan cara menukarkan isi *array* S[i] dengan S[j]. Prosesnya adalah sebagai berikut:
j = 0
For i = 0 to 255
j = (j + S[i] + K [i]) mod 256
isi S[i] dan isi S[j] ditukar
bangkitkan keystream dan lakukan enkripsi.
Proses untuk membangkitkan kunci enkripsi adalah sebagai berikut:
i = j = 0
i = (i + 1) mod 256
j = (j+S [i]) mod 256
isi S[i] dan S[j] di tukar
t = (S[i] + S[j]) mod 256
K = S[t]
Proses pembangkitan *keystream* K dipilih dengan mengambil nilai S[i], S[j] dan menjumlahkannya dalam modulo 256. Hasil penjumlahan adalah nilai indeks t sedemikian sehingga S[t] menjadi kunci aliran K.
Keystream K kemudian digunakan untuk mengenkripsikan plainteks ke-idx sehingga didapatkan cipherteks, sedangkan untuk mendapatkan plainteks, XOR-kan cipherteks dengan kunci yang sama dengan proses enkripsi [8].

2.5 Algoritma One Time Pad

One Time pad pertama kali ditemukan pada tahun 1917 oleh Major Joseph Mauborgne. *One time pad* (pad = kertas blaknot) berisi barisan karakter-karakter kunci yang dibangkitkan secara acak. Aslinya, satu buah *One Time Pad* adalah sebuah pita yang berisi barisan karakter-karakter kunci. Satu *Pad* hanya digunakan sekali (*One time*) saja untuk mengenkripsi pesan, setelah itu pad yang digunakan dihancurkan supaya tidak dipakai kembali untuk mengenkripsi pesan yang lain. Pengirim pesan menggunakan setiap karakter kunci untuk mengenkripsikan satu karakter plainteks. Enkripsi dapat digambarkan sebagai penjumlahan modulo 256 dari satu karakter plainteks dengan satu karakter kunci *One Time Pad*[9] :
Ci = (Pi + Ki) mod 256
Yang dalam hal ini,
Pi : karakter plainteks ke-i
Ci : karakter ciperteks ke-i
Ki : karakter kunci ke-i
Adapun syarat-syarat yang harus dipenuhi untuk dapat mengenkripsi atau mendeskripsikan pesan menggunakan algoritma kriptografi *One Time Pad* tersebut, yaitu:
1. Panjang kunci harus sama dengan panjang plainteks, sehingga tidak ada kebutuhan mengulang penggunaan kunci selama proses enkripsi.
2. Jika panjang kunci tidak sama dengan panjang plainteks, maka karakter-karakter kunci harus diulang sebanyak jumlah karakter palinteks agar proses enkripsi maupun deskripsi dapat dilakukan.

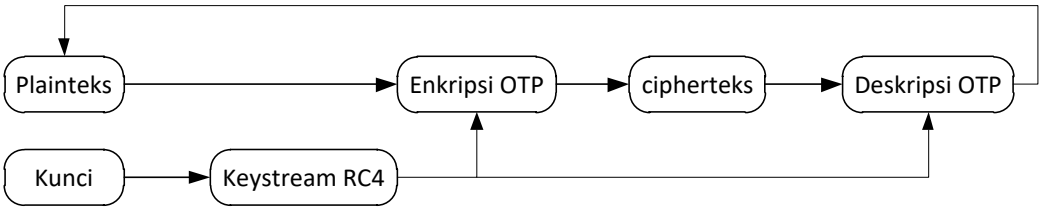
Penerimaan pesan menggunakan *One Time Pad* yang sama untuk mendeskripsikan karakter-karakter cipherteks menjadi karakter-karakter plainteks dengan persamaan:
Pi = (Ci - Ki) mod 256
Sistem cipher *One Time Pad* ini tidak dapat dipecahkan karena: Barisan kunci acak yang ditambahkan ke plainteks menghasilkan ciperteks yang seluruhnya acak.
Beberapa barisan kunci yang digunakan untuk mendeskripsi cipherteks mungkin menghasilkan pesan-pesan plainteks yang mempunyai makna, sehingga kriptanalis tidak punya cara untuk menentukan plainteks mana yang benar [13].

3. HASIL DAN PEMBAHASAN

Analisa masalah yang dilakukan pada penelitian ini adalah pada bagaimana cara untuk meningkat keamanan data, dimana data tersebut tidak boleh diketahui teks aslinya oleh yang bukan haknya. Teks yang sering dicuri informasinya dikarenakan teks tersebut memiliki arti yang sangat penting dan berguna bagi si pelaku.

Cara untuk mengamankannya adalah dengan menggunakan algoritma-algoritma dari kriptografi, yaitu algoritma *One Time Pad,* yang mana tingkat keamanan dari algoritma *One Time Pad* masih bisa di anggap cukup akurat untuk keamanan, sehingga untuk lebih meningkatkan keamanan dari algoritma *One Time Pad*, Maka akan ditinggkatkan lagi keamanan dari algoritma One Time Pad pada kuncinya dengan diperkuat lagi dengan tambahan dari algoritma kriptografi yaitu *RC4*. *RC4* memiliki pembangkit kunci yang mana akan menutup kelemahan atau kekurangan dari algoritma *One Time Pad*. Sehingga teks akan tidak tersusun dengan benar atau secara acak.

JURIKOM (Jurnal Riset Komputer), Vol. 8 No. 1, Februari 2021
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v7i5.2541
Hal 20−27



Gambar 1. skema proses plinteks dan kunci

Dengan digunakannya kedua algoritma kriptografi tersebut pada teks, akan mampu menyulikan pihak ketiga dalam upaya mendapatkan teks asli tersebut, sehingga pihak ketiga tidak akan mudah dalam mendapatkan teks asli yang sebenarnya pada ciperteks tersebut.

3.1 Penerapan Algoritma RC4

Untuk dapat melakukan opersi perhitungan pada proses enkripsi maka terlebih dahulu teks dan kunci yang digunakan sesuai dengan yang telah digunakan. Sebagai contoh kunci yang akan digunakan adalah “STMIKBDARCOP” maka akan diubah dalam bentuk bilangan karakter dan desimal.

Kunci (char)	S	T	M	I	K	B	D	A	R	C	O	P
Kunci (dec)	83	84	77	73	75	66	68	65	82	67	79	80

Untuk tahap pertama yang dilakukan adalah seperti berikut:

1. Inisiasi *vector* S
   Tiap-tiap elemen dari *vector* S diberi nilai awal secara berurutan, sedangkan *vector* T tiap-tiap elemennya diberi nilai berdasarkan nilai dari kunci. Inisiasi awal *vector* S dan *vector* T dijabarkan sebagai berikut.
   - Iterasi ke – 1
     i = 0
     S[i] = i
     S[0] = 0
     T[i] = kunci [i mod panjangkunci]
     T[0] = kunci [0 mod 12]
     = kunci [0]
     = 83
   - Iterasi ke – 2
     i = 1
     S[i] = i
     S[0] = 1
     T[i] = kunci [i mod panjangkunci]
     = kunci [1 mod 12]
     = kunci [1]
     = 84
   - Dan seterusnya hingga iterasi ke – 256

Maka didapatlah nilai awal *vector* S yang ditunjukkan pada tabel 2. dan untuk *vector* T ditunjukkan pada tabel 3.

Vector S []															
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

Gambar 2. Nilai Awal *Vector* S

Vector T []															
83	84	77	73	75	66	68	65	82	67	79	80	83	84	77	73
75	66	68	65	82	67	79	80	83	84	77	73	75	66	68	65
82	67	79	80	83	84	77	73	75	66	68	65	82	67	79	80
83	84	77	73	75	66	68	65	82	67	79	80	83	84	77	73
75	66	68	65	82	67	79	80	83	84	77	73	75	66	68	65
82	67	79	80	83	84	77	73	75	66	68	65	82	67	79	80
83	84	77	73	75	66	68	65	82	67	79	80	83	84	77	73
75	66	68	65	82	67	79	80	83	84	77	73	75	66	68	65
82	67	79	80	83	84	77	73	75	66	68	65	82	67	79	80
83	84	77	73	75	66	68	65	82	67	79	80	83	84	77	73
75	66	68	65	82	67	79	80	83	84	77	73	75	66	68	65
82	67	79	80	83	84	77	73	75	66	68	65	82	67	79	80
83	84	77	73	75	66	68	65	82	67	79	80	83	84	77	73
75	66	68	65	82	67	79	80	83	84	77	73	75	66	68	65
82	67	79	80	83	84	77	73	75	66	68	65	82	67	79	80
83	84	77	73	75	66	68	65	82	67	79	80	83	84	77	73

Gambar 3. Nilai Awal *Vector* T

Kemudian dilakukan permutasi dalam *vector* S sebagai berikut:

- Iterasi ke -1
  i = 0 , j = 0
  j = (j + S[i] + T[i]) mod 256
  j = (0 + S[0] + T[0]) mod 256
  j = (0 + 0 + 83) mod 256
  j = 83
  swap (S[0], S[83])

hasil permutasi iterasi pertama vector S ditunjukkan pada tabel 4.

Vector S []															
83	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	0	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

Gambar 4. hasil permutasi iterasi pertama *Vector* S

Maka didapatlah hasil akhir inisiasi *vector* S seperti ditunjukkan pada tabel 3.6.

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

Gambar 5. Hasil akhir inisiasi *Vector* S

JURIKOM (Jurnal Riset Komputer), Vol. 8 No. 1, Februari 2021
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v7i5.2541
Hal 20−27

2. *Stream Generation* (Pembangkit Kunci)
   Pembangkit kunci dilakukan sebanyak panjang teks karena setiap karakter teks dienkripsi oleh kunci yang berbeda. Pembangkit kunci dijabarkan sebagai berikut.
   - Iterasi ke – 1
     i = 0, j = 0
     i = (i + 1) mod 256
       = (0 + 1) mod 256
       = 1
     J = (j + S[i]) mod 256
       = (0 + S[1]) mod 256
       = (0 +164) mod 256
       = 164
     Swap (S[i], S[j])
     Swap (S[1], S[164])
     t = (S[i] + S[j]) mod 256
       = (S[1] + S[164]) mod 556
       = (168 + 164) mod 256
       = 76
     k = S[t]
       = S[76]
       = 18

   Dan seterusnya hingga iterasi ke – 12.
   - Iterasi ke – 12
     i = 11, j = 196
     i = (i + 1) mod 256
       = (11 + 1) mod 256
       = 12
     J = (j + S[i]) mod 256
       = (196 + S[12]) mod 256
       = (196 + 76) mod 256
       = 272 mod 256
       = 16
     Swap (S[i], S[j])
     Swap (S[12], S[16])
     t = (S[i] + S[j]) mod 256
       = (S[12] + S[16]) mod 256
       = (147 + 76) mod 256
       = 223
     k = S[t]
       = S[223]
       = 14

Maka didapatlah aliran kunci =

18	248	63	242	190	71	190	194	35	57	13	14

Setelah aliran kunci dihasilkan maka untuk menghasilkan sebuah cipherteks akan dilakukan dengan algoritma One Time Pad.

3.1 Penerapan Algoritma One Time Pad

Enkripsi One Time Pad tergantung dari banyaknya jumlah karakter plainteks dan kunci yang memiliki jumlah karakter yang sama dengan Plainteks agar menghasilkan sebuah cipertext. Setiap karakter plainteks dikonversi ke desimal sesuai dengan ketentuan karakter dan nilai desimal semua karakter plainteks dapat dilihat pada gambar 3.2. Seperti yang penulis sampaikan dalam analisa masalah kalau dalam enkripsi algoritma *One Time Pad* panjang kunci harus sama dengan panjang plainteks. Panjang kunci tidak boleh melebihi panjang plainteks dan jika panjang kunci tidak sama dengan panjang plainteks maka tambah satu atau lebih huruf hingga menyamai panjang plainteks. Pada gambar 3.2 terlihat gambar karakter yang penulis buat sehingga terdapat jumlah karakter. Proses enkripsi per karakter plainteks dan kunci dijumlahkan dan hasilnya dilakukan mod m yaitu mod 256 dari jumlah karakter.
Kunci awal yang digunakan yaitu hasil dari

S	T	M	I	K	B	D	A	R	C	O	P
83	84	77	73	75	66	68	65	82	67	79	80

Yang menjadi setelah di gunakan *RC4 keystream*

18	248	63	242	190	71	190	194	35	57	13	14

JURIKOM (Jurnal Riset Komputer), Vol. 8 No. 1, Februari 2021
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v7i5.2541
Hal 20−27

maka setelah dilakukan pembangkit kunci menggunakan algoritma RC4, jumlah karakter pada plainteks sama dengan jumlah kunci yang digunakan yaitu berjumlah 12 karakter. Dikarenakan setelah kunci dibangkitkan dengan RC4 maka hasilnya berbeda jauh dari kunci yang sebelumnya, walaupun mungkin menghasilkan beberapa karakter yang sama. Dengan begitu kunci yang dihasilkan berupa acak yang berfariasi.

Berikut proses enkripsi dan deskripsi pada plainteks dengan kunci yang sudah dibangkitkan:

Contoh yang digunakan adalah plainteks yang memiliki 12 karakter, dengan kunci 12 karakter.

Plainteks :	I	R	F	A	N	S	T	M	I	K	B	D
Desimal :	73	82	70	65	78	83	84	77	73	75	66	68
Kunci :	18	248	63	242	190	71	190	194	35	57	13	14
char		ø	?	ò	¾	G	¾	Â	#	9		

Maka operasinya :

(I +) mod 256	= (73 + 18)	mod 256	= 91	= [
(R + ø) mod 256	= (82 + 248)	mod 256	= 74	= J
(F + ?) mod 256	= (70 + 63)	mod 256	= 133	= ...
(A + ò) mod 256	= (65 + 242)	mod 256	= 51	= 3
(N + ¾) mod 256	= (78 + 190)	mod 256	= 12	=
(S + G) mod 256	= (83 + 71)	mod 256	= 154	= š
(T + ¾) mod 256	= (84 + 190)	mod 256	= 152	= ˜
(M + Â) mod 256	= (77 + 194)	mod 256	= 142	= Ž
(I + #) mod 256	= (73 + 35)	mod 256	= 155	= ›
(K + 9) mod 256	= (75 + 57)	mod 256	= 142	= Ž
(B +) mod 256	= (66 + 13)	mod 256	= 145	= ‘
(D +) mod 256	= (68 + 14)	mod 256	= 148	= ”

Maka semua hasil jika diurutkan menjadi cipherteks, berikut plainteks, kunci dan cipherteks:

Plainteks :	I	R	F	A	N	S	T	M	I	K	B	D
Desimal :	73	82	70	65	78	83	84	77	73	75	66	68
Kunci :	18	248	63	242	190	71	190	194	35	57	13	14
Char		Ø	?	ò	¾	G	¾	Â	#	9		
Desimal	91	74	133	51	12	154	152	142	155	142	145	148
Ciperteks	[	J	…	3		š	˜	Ž	›	Ž	‘	”

Hasil cipherteks inilah yang akan digunakan untuk di jadikan informasi, sehingga hasilnya menjadi acak yang merupakan bukan teks aslinya lagi, selama kunci atau hasil kunci tidak ada yang tau, maka ciperteks tersebut aman dan tidak mudah untuk dibaca data aslinya.

3.3 Penerapan Deskripsi *One Time Pad*

Untuk mengetahui isi data teks yang sudah di amankan, maka pertama kali dilakukan adalah deskripsi, ketentuannya sama dengan enkripsi. Yang membedakan adalah pada enkripsi satu karakter plainteks ditambah dengan satu karakter kunci, sedangkan deskripsi satu karakter cipherteks dikurangi satu karakter kunci. Selanjutnya digunakan Modulus yang sama yaitu mod 256.

Berikut proses deskripsi cipherteks dan kunci yang digunakan yaitu:

Desimal	91	74	133	51	12	154	152	142	155	142	145	148
Ciperteks	[	J	…	3		Š	˜	Ž	›	Ž	‘	”
Kunci :	18	248	63	242	190	71	190	194	35	57	13	14
Char		Ø	?	ò	¾	G	¾	Â	#	9		

Maka operasinya :

([-)	mod 256	= (91 - 18)	mod 256	= 73 = I
(J - ø)	mod 256	= (74 - 248)	mod 256	= 82 = R
(... - ?)	mod 256	= (133 - 63)	mod 256	= 70 = F
(3 - ò)	mod 256	= (51 - 242)	mod 256	= 65 = A
(- ¾)	mod 256	= (12 - 190)	mod 256	= 78 = N
(Š - G)	mod 256	= (154 - 71)	mod 256	= 83 = S
(˜ - ¾)	mod 256	= (152 - 190)	mod 256	= 84 = T
(Ž - Â)	mod 256	= (142 - 194)	mod 256	= 77 = M
(› - #)	mod 256	= (155 - 35)	mod 256	= 73 = I
(Ž - 9)	mod 256	= (142 - 57)	mod 256	= 75 = K
(‘ -)	mod 256	= (145 - 13)	mod 256	= 66 = B
(” -)	mod 256	= (148 - 14)	mod 256	= 68 = D

Dan hasilnya jika diurutkan menjadi plainteks: “IRFANSTMIKBD”.

4. KESIMPULAN

Berdasarkan hasil analisa yang dilakukan, maka kesimpulan dari penelitian ini proses algoritma kriptografi RC4 dan One Time Pad memiliki cara yang berbeda dalam mengamankan data teks, yang memiliki tingkat kesulitan masing masing dalam mengamankannya, maksudnya memiliki tingkat keamanan yang berbeda dari kedua algoritma tersebut. Untuk menerapkan kedua algoritma kriptografi RC4 dan One Time Pad pada data teks, yaitu dengan cara enkripsi dan deskripsi pada teks yang ingin diamankan. Dengan mengkombinasikan pembangkit kunci yang dimiliki RC4 untuk hasil kunci yang acak. Sehingga akan dilakukan enkripsi pada plainteks bersamaan dengan kunci acak, maka dienkripsi dan deskripsi berdasarkan dari algoritma One Time Pad. Dengan berkembang pesatnya teknologi maka data bukan hanya huruf A sampai Z saja, malainkan data juga memiliki variasi dari jenis karakter yaitu berupa huruf, angka, simbol dan lainnya..

REFERENCES

[1] E. Helmud, "KOMBINASI KRIPTOGRAFI RC4 DAN STEGANOGRAFI LSB PADA CITRA DIGITAL DENGAN FORMAT BITMAP UNTUKMENJAGA KEAMANAN PESAN," vol. 2, no. 2, pp. 20–27, 2017.
[2] E. L. Hakim, Khairil, and F. H. Utami, "Aplikasi Enkripsi Dan Deskripsi Data Menggunakan Algoritma Rc4 Dengan Menggunakan Bahasa Pemrograman Php," J. Media Infotama, vol. 10, no. 1, pp. 1–7, 2014.
[3] M. Diana and T. Zebua, "Optimalisasi Beaufort Cipher Menggunakan Pembangkit Kunci RC4 Dalam Penyandian SMS," J. Sains Komput. Inform., vol. 2, no. 1, pp. 12–22, 2018.
[4] M. K. Harahap, "Analisis Algoritma One Time Pad Dengan Algoritma Cipher Transposisi Sebagai Pengamanan Pesan Teks," vol. 1, no. April 2017, pp. 58–62, 2019.
[5] D. Ariyus, "Pengantar ilmu Kriptografi teori, analisis, dan implementasi", yogyakarta : ANDI. 2008
[6] S. Rizky, "konsep perancangan", dalam konsep dasar rekayasa perangkat lunak, jakarta, prestasi pustaka, 2011, pp.140-141
[7] S. Widianti, "Pengantar Basis Data", Fajar, Jakarta: 2000
[8] E. Setyaningsih, "Kriptografi & Implementasinya menggunakan Matlab", yogyakarta : ANDI. 2015
[9] T. Y. Nitti, A. Fanggidae, and D. M. Sihotang, "PENGGUNAAN METODE BLUM BLUM SHUB UNTUK ENKRIPSI DAN DEKRIPSI DATA TEKS DENGAN ALGORITMA ONE-TIME PAD," vol. 3, no. 2, 2015.