Vigenère Cipher dengan Pembangkit Bilangan Acak-Semu

Aditya Pratama - 13507084
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
if17084@students.if.itb.ac.id

Vigenère cipher adalah salah satu algoritma kriptografi klasik yang menggunakan konsep substitusi. Teknik enkripsi dari vigenère cipher ini adalah substitusi setiap karakter dari pesan menjadi karakter lain berdasarkan kunci yang digunakan. Jika panjang kunci yang digunakan lebih pendek dari panjang pesan maka kunci yang digunakan akan diulang sampai panjang kunci menjadi sama dengan panjang pesan. Vigenère cipher ini menggunakan konsep substitusi abjad-majemuk sehingga satu huruf bisa dienkripsi menjadi dua huruf yang berbeda.

Kelemahan dari algoritma ini adalah karena kunci yang digunakan akan berulang jika panjang kunci lebih pendek dari panjang pesan maka ada kemungkinan akan didapatkan kriptogram yang berulang. Hal inilah yang dimanfaatkan pada metode Kasiski. Metode Kasiski ini dapat digunakan untuk menemukan panjang dari kunci yang digunakan pada vigenère cipher dan dengan menggunakan metode frekuensi analisis kriptanalis dapat mendapatkan kunci yang digunakan.

Dalam makalah ini penulis akan memberikan ide modifikasi dari vigenère cipher ini sehingga menjadi lebih kuat dan tahan dari metode Kasiski. Ide penulis dalam memodifikasi vigenère cipher ini adalah dengan menggunakan pembangkit bilangan acak-semu yang juga digunakan pada teknik steganografi untuk membangkitkan bilangan acak [2]. Dengan digunakannya pembangkit bilangan acak-semu ini dapat dihasilkan kunci acak yang panjangnya sesuai dengan panjang pesan. Dengan kunci yang acak ini diharapkan ciphertext yang dihasilkan lebih kuat dan tahan dari serangan menggunakan metode Kasiski.

Kata kunci: **Kriptografi, metode Kasiski, pembangkit bilangan acak-semu, Vigenère cipher**

I. PENDAHULUAN

Informasi merupakan hal yang sangat penting bagi semua orang. Dengan semakin pesatnya perkembangan teknologi khususnya teknologi informasi dan komunikasi maka informasi menjadi makin berharga. Dengan semakin berharganya suatu informasi maka akan terdapat usaha-usaha perahasiaan yang membatasi akses semua orang kepada informasi tersebut. Salah satu permasalahan dalam pembatasan ini adalah saat informasi tersebut akan disampaikan kepada pihak lain, terutama pada saat informasi masih dikirimkan dengan cara konvensional dalam bentuk pesan tertulis.

Informasi yang berbentuk pesan tertulis akan sangat sulit dijaga kerahasiaannya. Akan dibutuhkan biaya yang sangat tinggi jika ingin menjaga kerahasiaan dari sebuah pesan tertulis karena kemudahan dari pihak-pihak lain yang ingin mengetahuinya. Yang perlu dilakukan oleh pihak lain tersebut hanya mendapatkan media dari pesan tersebut dan membacanya. Oleh karena itu untuk melindungi pesan tertulis tersebut dikembangkanlah sistem penyandian pesan tertulis tersebut dan muncul istilah kriptografi.

Istilah kriptografi berasal dari bahasa Yunani yang berarti tulisan rahasia. Menurut definisi yang lama, kriptografi adalah ilmu dan seni untuk menjaga kerahasian pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Dalam perkembangannya, kriptografi sekarang bukan hanya sebatas untuk mengenkripsi pesan, melainkan juga untuk memberikan aspek keamanan. Oleh karena itu sekarang terdapat definisi baru untuk kriptografi, yaitu ilmu dan seni untuk menjaga keamanan pesan.

Saat ini kriptografi terbagi menjadi dua kategori, yaitu kriptografi klasik dan kriptografi modern. Kriptografi klasik adalah kriptografi yang menerapkan algoritma penyandian yang berbasis karakter. Algoritma kriptografi klasik ini hanya menggunakan operasi substitusi dan transposisi atau permutasi. Beberapa contoh dari algoritma kriptografi klasik antara lain Caesar cipher, Vigenère cipher, dan Playfair cipher. Kriptografi modern adalah kriptografi yang menerapkan algoritma penyandian yang berbasiskan bit. Kriptografi modern muncul dan berkembang sampai saat ini karena perkembangan dan penggunaan komputer digital yang memungkinkan suatu informasi untuk direpresentasikan dalam bentuk bit biner. Algoritma kriptografi modern ini banyak menggunakan operasi XOR karena karakteristiknya. Contoh dari algoritma kriptografi modern ini antara lain Blowfish cipher, Rijndael cipher, dan Twofish cipher.

II. DASAR TEORI

2.1 Vigenère Cipher

Vigenère cipher adalah salah satu algoritma kriptografi klasik yang diperkenalkan pada abad 16 atau kira-kira

pada tahun 1986. Algoritma kriptografi ini dipublikasikan oleh seorang diplomat dan juga kriptologis yang berasal dari Prancis, yaitu Blaise de Vigenère, namun sebenarnya algoritma ini telah digambarkan sebelumnya pada buku *La Cifra del Sig. Giovan Batista Belaso*, sebuah buku yang ditulis oleh Giovan Batista Belaso, pada tahun 1553. Karena yang memperkenalkan algoritma ini kepada public adalah Blaise de Vigenère maka algoritma ini dinamakan Vigenère Cipher.

Gambar 1 Blaise de Vigènere

Cara kerja dari Vigenère cipher ini mirip dengan Caesar cipher, yaitu mengenkripsi plainteks pada pesan dengan cara menggeser huruf pada pesan tersebut sejauh nilai kunci pada deret alphabet. Vigenère cipher adalah salah satu algoritma kriptografi klasik yang menggunakan metode substitusi abjad-majemuk. Substitusi abjad-majemuk mengenkripsi setiap huruf yang ada menggunakan kunci yang berbeda, tidak seperti Caesar cipher yang menerapkan metode substitusi abjad-tunggal yang semua huruf di suatu pesan dienkripsi menggunakan kunci yang sama.

Sebagai contoh Caesar cipher jika terdapat plainteks:

MAKALAH KRIPTOGRAFI

maka jika dienkripsi dengan dengan nilai kunci 2 akan didapat cipherteks:

OCMCNCJ MTKRVQITCHK

dari cipherteks yang didapat dapat kita lihat bahwa huruf M dienkripsi menjadi O, huruf A dienkripsi menjadi huruf C, dan seterusnya dimana huruf pada pesan digeser sejauh nilai kunci. Algoritma Caesar cipher sangat sederhana sehingga sangat berisiko untuk dipecahkan karena hanya dibutuhkan pengetahuan satu huruf dari plainteks untuk mengetahui kunci yang digunakan.

Vigenère cipher yang menerapkan metode substitusi abjad-majemuk tidak memiliki permasalahan tersebut karena setiap huruf pada pesan yang dienkripsi dengan Vigenère cipher ini akan digeser dengan nilai yang berbeda tergantung dengan kunci yang diberikan. Kunci yang digunakan pada Vigenère cipher berbeda dengan yang digunakan pada Caesar cipher. Jika pada Caesar cipher kuncinya hanya satu nilai saja, maka pada Vigenère cipher kunci yang digunakan berbentuk deretan huruf. Kunci yang berbetuk deretan kata tersebut akan memungkinkan setiap huruf plainteks untuk dienkripsi dengan kunci yang berbeda. Jika panjang kunci yang digunakan lebih pendek dari panjang plainteks maka kunci akan diulang sampai panjang kunci sama dengan panjang plainteks. Algoritma ini akan meminimalkan kemungkinan dipecahkannya cipherteks jika satu huruf plainteks diketahui.

Model matematika dari enkripsi pada algoritma Vigenère cipher ini adalah seperti berikut

$$C_i = E_K(M_i) = (M_i + K_i) \mod 26$$

dan model matematika untuk dekripsinya adalah

$$M_i = D_K(C_i) = (C_i - K_i) \mod 26$$

Dengan C memodelkan cipherteks, M memodelkan plainteks, dan K memodelkan kunci

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Gambar 2 Tabel pemetaan Vigenère Cipher

Contoh dari penerapan algoritma Vigenère cipher adalah jika kita memiliki sebuah plainteks yang ingin dienkripsi:

MAKALAH KRIPTOGRAFI

dan kita menggunakan kunci:

TUGAS

maka plainteks akan dienkripsi dengan cara:

plainteks	: MAKALAH KRIPTOGRAFI
kunci	: TUGASTU GASTUGASTUG
cipherteks	: FUQADTB QRAINUGJTZO

Huruf pada kunci akan dikonversi menjadi sebuah nilai, misalnya A = 0, B = 1, sampai dengan Z = 25. Setelah itu prosesnya sama seperti pada Caesar cipher dimana setiap huruf pada plainteks akan digeser sejauh nilai kunci yang posisinya bersesuaian. Pergeseran huruf-huruf ini bisa dipetakan dalam bentuk tabel 26x26 yang memetakan antara huruf pada plainteks dengan huruf pada kunci seperti yang diperlihatkan pada gambar 2.

2.2 Metode Kasiski

Metode Kasiski adalah salah satu metode kriptanalisis. Metode Kasiski dikembangkan oleh Friedrich Kasiski, yaitu orang yang pertama kali memecahkan Vigenère cipher pada tahun 1863. Metode Kasiski ini tidak hanya bisa digunakan untuk memecahkan Vigenère cipher, melainkan seluruh algoritma kriptografi klasik yang menggunakan metode substitusi abjad-majemuk. Metode ini membantu para kriptanalis untuk mengetahui panjang kunci yang digunakan dalam algoritma kriptografi tersebut.

Ide yang mendasari pengembangan metode ini adalah pada bahasa Inggris terdapat kelemahan bahwa adanya perulangan huruf, bahkan pasangan huruf atau tripel huruf. Bahkan sudah ada penelitian yang menghasilkan urutan frekuensi kemunculan huruf-huruf dalam teks bahasa Inggris. Dengan adanya pengulangan huruf-huruf tersebut dan juga kemungkinan pengulangan kunci maka akan muncul kemungkinan bahwa huruf-huruf yang sama akan dienkripsikan dengan kunci yang sama, sehingga menghasilkan kriptogram yang berulang. Jika para kriptanalis berhasil mendapatkan dua atau lebih kriptogram yang berulang pada sebuah cipherteks maka sang kriptanalis bisa menemukan panjang kunci yang digunakan.

Secara umum langkah-langkah penerapan metode Kasiski untuk menemukan panjang kunci yang digunakan adalah sebagai berikut:

1. Temukan semua kriptogram yang berulang di dalam cipherteks.
2. Hitung jarak antara kriptogram yang berulang.
3. Hitung semua faktor pembagi dari jarak tersebut (faktor pembagi menyatakan panjang kunci yang mungkin).
4. Tentukan irisan dari himpunan faktor pembagi tersebut. Nilai yang muncul di dalam irisan menyatakan angka yang muncul pada semua faktor pembagi dari jarak-jarak tersebut . Nilai tersebut mungkin adalah panjang kunci.

Jika setelah melakukan langkah-langkah tersebut sang kriptanalis hanya mendapatkan satu nilai maka nilai tersebut adalah panjang dari kunci yang digunakan pada algoritma kriptografi tersebut. Dengan didapatkannya panjang kunci tersebut maka kriptanalis sudah bisa memulai pencarian kata kunci yang digunakan, misalnya dengan menggunakan *exhaustive key serach*, yaitu dengan mencoba seluruh kombinasi huruf yang mungkin. Cara tersebut tidak efisien karena jika panjang kunci adalah N maka jumlah kunci yang mungkin adalah 26^N sehingga membutuhkan waktu yang cukup lama. Untuk mengefisienkan usaha yang digunakan biasanya kriptanalis akan menggunakan teknik analisis frekuensi sebagai pelengkap metode Kasiski.

Sudah banyak penelitian yang memberikan urutan frekuensi kemunculan huruf, pasangan huruf, bahkan tripel huruf untuk berbagai bahasa. Dengan dimilikinya frekuensi kemunculan huruf tersebut maka kriptanalis dapat menemukan kunci yang digunakan dengan lebih mudah. Langkah-langkah untuk mengetahui kunci yang digunakan menggunakan teknik analisis frekuensi jika kita mengetahui panjang kunci yang digunakan adalah sebagai berikut:

1. Kelompokkan setiap huruf kelipatan N dengan N adalah panjang kunci yang digunakan.
2. Tiap-tiap kelompok pesan tersebut dapat dipecahkan menggunakan teknik analisis frekuensi karena setiap huruf di kelompok tersebut dienkripsi secara substitusi abjad-tunggal.
3. Jika sudah ditemukan beberapa huruf yang digunakan sebagai kunci maka kriptanalis bisa langsung menyusun urutan huruf tersebut jika sudah sesuai dengan panjang kuncinya atau jika ada huruf dari kunci yang belum diketahuii maka kriptanalis bisa menerka dari plainteks yang sudah ada huruf kunci yang tersisa.

2.3 Pembangkit Bilangan Acak-Semu

Pembangkit Bilangan Acak-Semu atau yang biasa dikenal dengan singkatan PRNG (*Pseudo-Random Number Generator*) adalah sebuah algoritma untuk menghasilkan suatu urutan bilangan yang terlihat acak, namun sebenarnya urutan tersebut tidak benar-benar acak karena urutan tersebut ditentukan oleh suatu nilai awal. Urutan bilangan yang terlihat acak ini sangat penting karena bisa dimanfaatkan untuk suatu parameter bagi percobaan atau simulasi dan juga menjadi pusat pake praktik kriptografi.

Sebuah pembangkit bilangan acak-semu bisa dimulai dengan memberikan nilai umpan. Pembangkit bilangan acak-semu ini akan selalu memberikan urutan bilangan yang sama jika diberikan nilai umpan yang sama, dengan jumlah bilangan yang dihasilkan bergantung kepada besar nilai umpan yang diukur dengan satuan bit.

Contoh dari urutan bilangan hasil penggunaan pembangkit bilangan acak-semu yang digunakan dua kali dengan nilai umpan 63 bisa dilhat pada tabel 1.

Percobaan Pertama	Percobaan Kedua
1372323792	1372323792
650039580	650039580
1727148275	1727148275
10878762	10878762
1395080022	1395080022
1319514059	1319514059

286901698	286901698
103713261	103713261
1657217166	1657217166
1084529942	1084529942
1515972154	1515972154
1400856726	1400856726

Tabel 1 Hasil Penggunaan PRNG

Keuntungan dari penggunaan pembangkit bilangan acak-semu ini adalah efisien, algoritma ini mampu menghasilkan banyak angka dalam waktu singkat, dan tertentu, urutan yang digunakan bisa dimunculkan kembali dengan mudah jika nilai awalnya diketahui. Efisien adalah karakteristik yang sangat baik jika aplikasi kita membutuhkan banyak angka. Tertentu juga akan berguna jika kita perlu mengulang suatu urutan bilangan.

Sampai saat ini belum ada teori dan praktik dari kriptografi untuk mengetahui atau mencari perbedaan dari urutan bilangan acak hasil pembangkit bilangan acak-semu ini dan urutan bilangan yang benar-benar acak tanpa mengetahui algoritma dan nilai awal yang digunakan. Keamanan dari beberapa algoritma kriptografi dan protokol yang menerapkan pembangkit bilangan acak-semu ini bergantung pada asumsi bahwa tidak mungkin membedakan antara bilangan acak hasil pembangkit bilangan acak-semu ini dengan bilangan yang benar-benar acak.

III. MODIFIKASI VIGENÈRE CIPHER DENGAN PEMBANGKIT BILANGAN ACAK-SEMU

Vigenère cipher sebenarnya merupakan algoritma yang cukup kuat. Hal ini terbukti dengan lamanya waktu yang dibutuhkan dari sejak algoritma ini diperkenalkan sampai muncul metode pemecahannya, yaitu sekitar dua ratus tahun. setelah muncul metode pemecahannya dan disebarluaskan pada abad ke 19 maka algoritma Vigenère cipher mulai kehilangan kekuatannya, meskipun masih bisa digunakan pada situasi tertentu. Keunggulan lain dari algoritma ini selain kekuatannya adalah kesederhanaannya. Kesederhanaan dari algoritma ini menjadikan penggunaan algoritma ini tidak membutuhkan sumber daya dan usaha yang cukup besar dan mudah untuk dimodifikasi.

Kelemahan Vigenère cipher akan terlihat saat panjang kunci yang digunakan oleh pengguna lebih pendek dari pesan yang ingin dienkripsi atau plainteks, karena saat panjang kunci lebih pendek dari panjang plainteks maka akan terjadi pembuatan kunci baru yang dibuat dari pengulangan kunci lama yang panjangnya sama dengan panjang plainteks. Pengulangan kunci ini mengakibatkan munculnya kemungkinan pengulangan suatu pola huruf pada plainteks dienkripsi oleh kunci yang sama sehingga menghasilkan kriptogram yang sama dan berulang. Kelemahan inilah yang dimanfaatkan pada metode Kasiski.

Kelemahan Vigenère cipher tersebut sebenarnya bisa diminimalkan dengan menggunakan kunci sepanjang mungkin, bahkan bisa dihilangkan jika kunci yang digunakan sama panjangnya dengan panjang plainteks. Namun hal ini akan sangat menyulitkan pengguna jika plainteks berukuran sangat panjang. Selain menyulitkan pengguna juga menghilangkan nilai kesederhanaan dari algoritma Vigenère cipher ini, karena kunci yang panjangnya bernilai sama dengan panjang plainteks yang bisa digunakan untuk satu plainteks itu saja. Oleh karena itu solusi menggunakan kunci yang panjangnya bernilai sama dengan panjang plainteks sangat tidak disarankan untuk memperkuat Vigenère cipher ini.

Sudah banyak ide-ide yang telah diajukan oleh orang lain dalam makalah-makalah atau tulisan-tulisan lain yang ada untuk memodifikasi Vigenère cipher agar memperkuat algoritma ini dan hampir semuanya berusaha untuk menciptakan kunci baru yang panjangnya sama dengan panjang plainteks dan tidak dihasilkan dari pengulangan unsur-unsur tertentu. Beberapa ide modifikasi yang telah diajukan antara lain menggunakan posisi huruf kunci sebagai variabel tambahan dalam pembuatan kunci baru, menggunakan plainteks sebagai tambahan pada kunci, dan melakukan pembuat kunci baru lebih dari satu kali.

Pada tulisan ini penulis berusaha menuangkan ide untuk memodifikasi Vigenère cipher agar lebih kuat dalam menghadapi metode kriptanalisis, khususnya metode Kasiski. Ide yang diajukan oleh penulis adalah memanfaatkan pembangkit bilangan acak-semu sebagai penghasil kunci baru.

Seperti yang sudah dijelaskan sebelumnya, pembangkit bilangan acak-semu adalah suatu algoritma yang menghasilkan urutan bilangan yang terlihat semu, namun sebenarnya urutan bilangan tersebut dihasilkan dengan memproses sebuah nilai awal. Pemanfaatan pembangkit bilangan acak-semu ini dimulai terletak pada pembuatan kunci baru dari algoritma Vigenère cipher ini. Saat pengguna memberikan kunci yang digunakan pada Vigenère cipher ini, kunci tersebut tidak akan diulang jika ternyata panjang kunci lebih pendek dari panjang plainteks, tetapi kunci tersebut akan dikonversi menjadi sebuah bilangan. Bilangan hasil konversi kunci yang diberikan oleh pengguna itu akan dijadikan sebagai nilai awal untuk pembangkit bilangan acak-semu tersebut. Akan didapatkan sebuah urutan bilangan yang terlihat acak yang panjangnya disesuaikan dengan panjang plainteks. Tiap bilangan pada urutan tersebut kemudian dijadikan nilai kunci untuk menggeser huruf plainteks dan menghasilkan cipherteks. Dengan memanfaatkan pembangkit bilangan acak-semu ini maka kelemahan dari Vigenère cipher yaitu kunci yang berulang bisa dihilangkan.

IV. PENGUJIAN DAN ANALISIS

4.1 Pengujian

Pada bagian akan diuji dan dibandingkan hasil dari algoritma Vigenère cipher biasa dengan hasil dari algoritma Vigenère cipher yang sudah dimodifikasi menggunakan pembangkit bilangan acak-semu menggunakan sebuah program sederhana

Plainteks yang digunakan adalah:

```
CORPORATIONSANDORGANIZATIONSROUTINEL
YUSEINTERNETVOTINGTOELECTOFFICERSANDB
OARDMEMBERSANDFOROTHERPROXYELECTIO
NSINTERNETVOTINGSYSTEMSHAVEBEENUSEDP
RIVATELYINMANYMODERNNATIONSANDPUBLI
CLYINTHEUNITEDSTATESTHEUKSWITZERLAND
ANDESTONIAINSWITZERLANDWHEREITISALREA
DYANESTABLISHEDPARTOFLOCALREFERENDU
MSVOTERSGETTHEIRPASSWORDSTOACCESSTHE
BALLOTTHROUGHTHEPOSTALSERVICEMOSTVO
TERSINESTONIACANCASTTHEIRVOTEINLOCALA
NDPARLIAMENTARYELECTIONSIFTHEYWANTTO
VIATHEINTERNETASMOSTOFTHOSEONTHEELEC
TORALROLLHAVEACCESSTOANEVOTINGSYSTE
MTHELARGESTRUNBYANYEUROPEANUNIONCO
UNTRY
```

dengan kunci yang digunakan adalah TUGAS
jika menggunakan algoritma Vigenère cipher yang biasa maka kunci baru yang akan dihasilkan adalah:

```
TUGASTUGASTUGASTUGASTUGASTUGASTUGAS
TUGASTUGASTUGASTUGASTUGASTUGASTUGAS
TUGASTUGASTUGASTUGASTUGASTUGASTUGAS
TUGASTUGASTUGASTUGASTUGASTUGASTUGAS
TUGASTUGASTUGASTUGASTUGASTUGASTUGAS
TUGASTUGASTUGASTUGASTUGASTUGASTUGAS
TUGASTUGASTUGASTUGASTUGASTUGAS
TUGASTUGASTUGASTUGASTUGASTUGASTUGAS
TUGASTUGASTUGASTUGASTUGASTUGASTUGAS
TUGASTUGASTUGASTUGASTUGASTUGASTUGAS
TUGASTUGASTUGASTUGASTUGASTUGASTUGAS
TUGASTUGASTUGASTUGASTUGASTUGASTUGAS
TUGASTUGASTUGASTUGASTUGASTUGASTUGAS
TUGASTUGASTUGASTUGASTUGASTUGASTUGAS
TUGASTUGASTUGASTUGASTUGASTUGASTUGAS
TUGASTUGASTUGAS
```

sedangkan jika menggunakan pembangkit bilangan acak-semu maka kunci barunya adalah:

```
IIVYYTSPUCIWPHRPBOEOWKAXBOVWAYXVZY
HYQOLNJEZBQIGVCJBSWMHYFYUKYTMTFNXR
XQZNAOCPLZETFXURXFWNOJQPUYPQSDWALP
VJKOPFHPNGRYFYWSWXRHAOXHQCZMXFAEIN
ZIHGQODWWKVTTORXEWJBMHJAVKGVPFDNQP
ZOLQHABBSNMXJLSABEVEBSWEIDIIEOEIDBPDJ
LPJXLPWOYBVAPBZSJOEPHFEPTUBGBRFTPONPJ
HQAANHZNIPNFNUNILCDIREOOQKEWZIYLKGC
YBPRYCCIFOYOXSLEMHEPAZMKDADLAOZVHPP
EWIFGPTGXPWHEOEMEDOZWMFOVWOIOMDEF
AEFBZYAEQPSETZXTZSSQQQPOUKHBLSIJPZBH
WICFOSDXKKZXOMMULOPKMOHPOZTYEGMUY
QXRQKZVKPEHLHKWDOYBKOSFBNOPZFTBVFZ
YPVBYMRXUKXBEMXVNILKHVPNDTKWFWAM
DIMEFNAMQOVVBJHDZHGTKVPDAIVCCVSYHL
YWAQQXFZRTXUJUMHUYQECITG
```

Dapat dilihat bahwa kunci yang dihasilkan sangat jauh berbeda, dimana kunci yang dihasilkan melalui algoritma Vigenère cipher biasa kunci barunya hanya pengulangan dari kunci lama dan kunci yang dihasilkan melalui modifikasi Vigenère cipher dengan memanfaatkan pembangkit bilangan acak-semu menghasilkan kunci yang "acak".

Hasil dari enkripsinya pun pasti berbeda, dimana hasil dari algoritma Vigenère cipher biasa adalah:

```
VIXPGKUZIGGMGNVHLMAFBTGTAHHYRGNN
ONWESASWBHZEJGYZVGMCTGLHYREUMILFAV
YXSSGXHOSKXSEEUYXSSGXLOJHNNEJILUXQXF
KCLBITSAGNKRFXNBOLBHMSQLNKMKAUBETX
YTUKXXVRAOUZEDRCTMSGSSOVXLTNSMCUNK
THJPMUFOCDRCTTZXOTILXXYTSMYYTZXOQSO
BNFEJEUTDSGXKSLHHOAAGMCILSYXLSGXCHW
KYOTALURRWTXEAFXMZATECYHWWJGRLHZR
OUTFXEXXLKNVNGYVGMYXSYXNZHWBLVAKL
QURVLNUAUVYYSLAYHADEIZTZKIAGZMBKPGL
NGLKXLBIUXGUSLOIZEJLCTEKMITISVUTCSLNZ
HWBLBOLXCTLGVURAFWJGRDBUSEFMUXYWE
YITAHHYIXMBKYOTHZTGOCGTZXCTTWKHKTS
LGUSLHZZHGLYUNLAYKLWVNURSELULDAUBE
SVWKSKMIGNWOIZIFZMESLXGZHWEUXGWLNX
UFUSGNQXOXOHXUTUFBITCGNHZRQ
```

Sedangkan hasil enkripsi dari algoritma Vigenère cipher yang telah dimodifikasi menggunakan pembangkit bilangan acak-semu adalah:

```
KWMNMKSICQVOPUUDSUEBEJAQJCIORMROHLL
JOIDRRRSFHVKOXXUAJSAMJJYMRHRYNPBIPQM
QBCCGOLIFGBLJXSZSCAEIBCGFJRTYPAZLDWDS
ZXAZVPSCPNKQZUGGVZJGLEEFVIJRDVBYURSN
EFVMXZPFRIJOKTXDZBTIPYLBDHZBOFBBMJUY
KFWEZEVRDXFVOXIWMAXVICNTLLCKTAILCZO
LEZSIPMAJWRHDNXOXLMGOUBATFRXCPIALEL
ZQGPAJFNNJWKVPVHDOHDSBKWALVXGDFGVL
FWUXJMHBJDKQAXWEHDZDSVHOFCNADPGYM
XYIAKYPHSSHXTVRIFDFMSKKGBOXVIWVMHFL
MSXLDLIKRFGDKLEDYPQUXJBDLBQTHSCKBGN
BDRZKVZKRBMLWWPWQBAPFXXJIIFCMDPZVD
GZILEUEAYRLORIOWFYFEBTSFLNJXSMUOIMEV
LHDEFIXBXGWCKSMDYOAKRJWCOHAEXTNNQL
CODOPZBRAKFDCTOAZBGUOJSUMWWNOURWN
GXXHDHUVHAEYPBKE
```

Dapat dilihat dari hasil percobaan tersebut terdapat perbedaan baik pada pembuatan kunci baru mau pun pada ciphertext hasil enkripsi antara algorita Vigenère cipher normal dengan algoritma Vigenère cipher yang telah dimodifikasi. Algoritma Vigenère cipher yang memanfaatkan pembangkit bilangan acak-semu berhasil menghasilkan kunci yang memiliki panjang sama dengan plainteks, terlihat acak, dan tidak membutuhkan sumber daya dan usaha yang cukup besar. Dalam proses dekripsi pun tidak ada masalah karena dengan menggunakan kunci yang sama akan menghasilkan bilangan konversi yang sama yang akan digunakan sebagai umpan untuk membangkitkan urutan bilangan acak tersebut, sehingga tidak ada masalah dalam mengembalikan kembali kunci dan plainteks.

4.1 Analisis

Setelah melakukan pengujian baik enkripsi mau pun dekripsi didapatkan bahwa tidak ada masalah dalam penerapan pembangkit bilangan acak-semu dalam modifikasi algoritma Vigenère cipher. Modifikasi Vigenère cipher dengan memanfaatkan pembangkit bilangan acak-semu ini berhasil menghasilkan kunci baru yang panjangnya sama dengan plainteks, "acak", dan tidak membutuhkan usaha dan sumber daya yang cukup berat karena pengguna bisa menggunakan kunci sepanjang apa pun.

Proses dekripsi cipherteks menjadi plainteks pun tidak mengalami masalah selama kunci yang digunakan untuk dekripsi sama dengan kunci yang digunakan untuk enkripsi karena kunci tersebut akan dikonversi menjadi sebuah bilangan yang akan digunakan menjadi umpan pembangkit bilangan acak-semu tersebut. Selama nilai umpan tersebut sama maka urutan bilangan yang dijadikan kunci untuk enkripsi dan dekripsi akan sama.

Permasalahan mungkin akan muncul ketika algoritma dari pembangkit bilangan acak-semu yang digunakan tersebar ke publik. Dengan tersebarnya algoritma dari pembangkit bilangan acak-semu tersebut maka para kriptanalisis bisa menemukan kunci yang digunakan dan akhirnya memecahkan Vigenère cipher hasil modifikasi ini.

Jika memang hal tersebut terjadi maka solusi yang bisa dilakukan adalah mengganti algoritma pembangkit bilangan acak-semu yang sudah diketahui oleh publik tersebut dengan algoritma yang lain karena banyak sekali algoritma yang fungsinya untuk membangkit urutan bilangan acak-semu.

V. KESIMPULAN

Berdasarkan pengujian dan analisis dari modifikasi Vigenère cipher dengan memanfaatkan pembangkit bilangan acak-semu, penulis mendapatkan beberapa kesimpulan, yaitu:

1. Kelemahan dari Vigenère cipher terletak pembuatan kunci baru yang hanya merupakan pengulangan dari kunci lama.
2. Pembangkit bilangan acak-semu bisa dimanfaatkan pada modifikasi Vigenère cipher.
3. Pembuatan kunci baru dengan memanfaatkan pembangkitan bilangan acak-semu berhasil menghasilkan kunci baru yang memiliki panjang yang sama dengan plainteks dan terlihat acak.
4. Pengguna tidak perlu mengeluarkan sumber daya dan usaha yang besar dalam membuat kunci baru yang kuat (sama panjang dengan plainteks dan acak).
5. Algoritma ini akan relatif aman selama algoritma dari pembangkit bilangan acak-semu yang digunakan tidak diketahui.

DAFTAR PUSTAKA

[1] http://bytes.com/topic/c-sharp/answers/588374-how-random-random
Tanggal akses: 21 Maret 2011

[2] http://en.wikipedia.org/wiki/Pseudorandom_number_generator
Tanggal akses: 18 Maret 2011

[3] http://en.wikipedia.org/wiki/Vigen%C3%A8re_cipher
Tanggal akses: 2 Maret 2011

[4] http://home.cogeco.ca/~cipher/cyproc.htm
Tanggal akses: 21 Maret 2011

[5] http://www.random.org/randomness/
Tanggal akses: 20 Maret 2011

[6] Munir, Rinaldi. (2004). Bahan Kuliah IF5054 Kriptografi. Departemen Teknik Informatika, Institut Teknologi Bandung

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 23 Maret 2011

Aditya Pratama, 13507084