Modifikasi *Vigenere Cipher* Dengan Menggunakan Teknik Pengenkripsian Pada Kuncinya

Muchamad Surya Prasetyo – NIM : 13505065

Program Studi Teknik Informatika, Institut Teknologi Bandung

Jl. Ganesha 10, Bandung

E-mail : if15065@students.if.itb.ac.id

Abstrak

Makalah ini membahas tentang modifikasi *Vigenere Cipher* untuk menyandikan suatu pesan. *Vigenere Cipher* adalah suatu algoritma kriptografi klasik yang menerapkan suatu metode cipher subtitusi abjad majemuk. Algoritma ini sudah dapat dipecahkan melalui metode Kasiski, maka dari oleh sebab itu dilakukanlah modifikasi agar algoritma yang baru lebih sulit untuk dipecahkan dengan metode tersebut. Algoritma baru ini memanfaatkan penyandian kunci yang digunakan untuk menyandikan pesan sebenarnya.

Kata kunci : *Vigenere Cipher, Caesar Cipher, metode Kasiski*

1. Pendahuluan

Kriptografi adalah suatu ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara merubah atau menyandikan pesan tersebut ke dalam bentuk yang tidak dapat dimengerti maknanya. Kriptografi dapat dibagi ke dalam 2 jenis, yaitu kriptografi klasik dan modern.

Kriptografi klasik sudah mulai dipelajari manusia sejak tahun 400 SM, yaitu pada zaman Yunani kuno. Pada kriptografi klasik ini lebih menekankan pada perubahan tiap karakter dalam upaya menjaga kerahasiaan pesan. Contoh kriptografi klasik adalah *Vigenere Cipher* dan *Caesar Cipher*. Sedangkan pada kriptografi modern lebih menekankan pada pengoperasian bit-bit. Contohnya adalah *DES* dan *GOST*.

Pada kriptografi klasik terdapat 2 algoritma, yaitu cipher substitusi dan cipher transposisi. Pada cipher substitusi, setiap huruf dalam plainteks akan tepat berkorespondensi satu-satu dengan huruf dalam cipherteksnya, sedangkan pada cipher transposisi huruf-huruf pada plainteks hanya dimanipulasi letak posisinya (transpose) untuk menjadi cipherteks.

Vigenere Cipher merupakan salah satu algoritma kriptografi klasik yang menggunakan teknik subtitusi. *Vigenere Cipher* menggunakan Bujursangkar *Vigenere* untuk melakukan enkripsi. Bila panjang kunci lebih pendek dari plainteksnya, maka akan dilakukan pengulangan pada kuncinya tersebut.

Pada awalnya, teknik ini cukup sulit dpecahkan sebelum ditemukannya Metode Kasiski dan Analisis Frekuensi. Dengan adanya kedua teknik tersebut, maka pemecahan cipherteks yang menggunakan *Vigenere Cipher* akan menjadi lebih mudah. Namun bisa dilakukan beberapa manipulasi dengan cara melakukan beberapa modifikasi pada kuncinya, salah satunya adalah dengan mengenkripsi kunci pada saat pengulangan.

Dalam makalah ini akan dijelaskan suatu pengembangan algoritma *Vigenere Cipher* agar dapat menanggulangi kelemahan tersebut, yaitu dengan melakukan pengenkripsian ulang kunci dengan kunci tersebut, sehingga pola-pola yang dimanfaatkan dalam metode kasiski dan analisis frekuensi untuk memecahkannya dapat diminimalisir.

2. *Vigenere Cipher* dan Metode Kasiski

Algoritma enkripsi *Vigenere Cipher* diperkenalkan oleh Bastia Belaso pada tahun 1553 dan disempurnakan oleh Blaise de Vigenere pada tahun 1586. Karena mayoritas orang mengira Vigenere

adalah penemunya, maka algoritma ini disebut *Vigenere Cipher*.

Algoritma ini menjadi terkenal karena cukup sulit dipecahkan. Matematikawan Charles Lutwidge Dodgson menyatakan bahwa algoritma ini tidak terpecahkan. Pada tahun 1917, ilmuwan Amerika menyebutkan bahwa *Vigenere Cipher* adalah sesuatu yang tidak mungkin untuk ditranslasikan. Namun hal ini terbantahkan sejak Kasiski berhasil memecahkan algoritma ppada abad ke-19.

Pada dasarnya *Vigenere Cipher* serupa dengan *Caesar Cipher*, perbedaannya adalah pada *Vigenere Cipher* setiap huruf pesan aslinya digeser sebanyak satu huruf pada kuncinya sedangkan pada *Caesar Cipher* setiap huruf pesannya digeser sebanyak 1 huruf yang sama.

Algoritma *Vigenere Cipher* ini menggunakan Bujursangkar Vigenere untuk melakukan enkripsi. Setiap baris di dalam bujursangkar menyatakan huruf-huruf cipherteks yang diperoleh dengan *Caesar Cipher*. Untuk lebih jelasnya dapat melihat gambar 1 di bawah ini. Deretan huruf kuning mendatar merepresentasikan plainteks, sedangkan deretan huruf kuning menurun merepresentasikan kunci.

Jika panjang kunci lebih pendek daripada panjang plainteks, maka kunci tersebut akan diulang secara periodik. Bila panjang kunci adalah x, maka periodenya adalah x. Contohnya sebagai berikut :

Kunci : abcd
Plainteks : aaaa aaaaaa

Maka proses yang dilakukan adalah setiap huruf dicek pada Bujursangkar Vigenere dan hasilnya berupa cipherteks.

Plainteks : aaaa aaaaaa
Kunci : abcd abcdab
Cipherteks : abcd abcdab

Hal diatas merupakan karakteristik dari cipher abjad majemuk. Pada cipher substitusi sederhana, setiap huruf cipherteks selalu menggantikan huruf plainteks tertentu, sedangkan pada cipher abjad majemuk setiap huruf cipherteks dapat memiliki kemungkinan banyak huruf plainteks. Sehingga kita dapat mencegah frekuensi huruf-huruf di dalam cipherteks yang mempunyai pola tertentu yang sama sebagaimana yang diperlihatkan pada cipher substitusi sederhana atau abjad tunggal.

	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 1 Bujursangkar Vigenere

3. Metode Kasiski

Metode Kasiski merupakan sutau metode dimana dapat membantu menemukan panjang kunci dari *Vigenere Cipher*. Metode Kasiski memanfaatkan keuntungan bahasa inggris tidak hanya pengulangan 1 huruf tapi juga pengulangan pasangan huruf atau tripel. Perulangan seperti ini memungkinkan memberikan hasil cipherteks yang berulang.

Sebagai contohnya :
Plainteks :
CRYPTO IS SHORT FOR CRYPTOGRAPHY

Kunci :
abcdab cd abcda bcd abcdabcdabc

Cipherteks :
CSASTP KV SIQUT GQU CSATPIUAQJB

Pada contoh diatas, kata CRYPTO secara tidak sengaja memiliki enkripsi yang sama yaitu CSASTP. Hal ini dikarenakan jarak antara kedua kata yang berulang tersebut merupakan kelipatan dari panjang kunci. Tujuan utama dari metode Kasiski adalah mencari dua atau lebih cipherteks yang berulang untuk menentukan panjang kunci.
Langkah-langkah metode Kasiski adalah sebagai berikut :

- Cari pasangan huruf atau triple yang berulang pada cipherteks.
- Hitung jarak antara pasangan huruf yang berulang
- Hitung semua factor pembagi dari jarak tersebut
- Tentukan irisan dari semua himpunan faktor pembagi tersebut. Nilai yang muncul merupakan nilai yang memungkinkan menjadi panjang kunci.

Setelah panjang kunci diketahui, langkah selanjutnya adalah menentukan kata kunci. Kata kunci ini dapat ditentukan dengan melalui frekuensi analisis

Langkah-langkahnya adalah sebagai berikut :

- Misalkan panjang kunci yang sudah berhasil dideduksi adalah n. Maka setiap huruf kelipatan ke-n pasti dienkripsi dengan kunci huruf yang sama. Lalu kelompokkan setiap huruf tersebut, masing-masing kelompok ini dienkripsi dengan substitusi alphabet tunggal (*Caesar Cipher*).
- Tiap-tiap kelompok tersebut dapat dipecahkan melalui teknik analisis frekuensi.
- Setelah itu dapat disusun huruf-huruf kunci yang memungkinkan. Lalu diuji satu per satu kunci-kunci yang memungkinkan tersebut.

4. Modifikasi *Vigenere Cipher*

Dari penjelasan diatas sudah dapat terlihat kelemahan dari *Vigenere Cipher*. Dari hasil cipherteksnya bias ditemukan kuncinya. Maka untuk mengurangi kemungkinan tersebut dilakukan modifikasi dalam *Vigenere Cipher* tersebut.

Konsepnya adalah kunci yang digunakan pada saat mengenkripsi dienkripsi pada setiap pengulangannya, sehingga hasil cipherteks yang didapat apabila terdapat pasangan huruf yang berulang memiliki kemungkinan yang sangat kecil bahwa pasangan huruf tersebut merupakan pengulangan dari kunci.

Prosesnya adalah dengan menentukan kunci ke-i dengan mengenkripsi kunci pengulangan ke-(i-1) dengan kunci ke-1.

Plainteks :
CRYPTO IS SHORT FOR CRYPTOGRAPHY
Kunci : abcd
Kunci pengulangan ke-1 : abcd
Kunci pengulangan ke-2 :
abcd dienkripsi dengan abcd : aceg
Kunci pengulangan ke-3 :
aceg dienkripsi dengan abcd : adgj
dst.

5. Implementasi dan pengujian

***Vigenere Cipher* :**

```
string keyword;
string msg;
int crypted;
int real_difference;
int i=0;
```

```
    int k;
    int l;
    cout<<"Enter the keyword : ";
    getline(cin, keyword);

    cout<<"Enter the message you
want to crypt : ";
    getline(cin, msg);

    k = keyword.length();
    l = msg.length();

    loop1:
    if(k < l);
    {
       keyword = keyword +
keyword;
       k = keyword.length();
       if(k < l)
          goto loop1;
    }

    cout<<"\n\n";
    while (i<= msg.length())
    {
       keyword[i] -= 'a'- 1;
       if((msg[i] + keyword[i]) >
'z')
          crypted = 'a' +
(keyword[i]+msg[i] - 'z') - 1;
       else if(msg[i] == 32)
          crypted = msg[i];
       else
          crypted = msg[i] +
keyword[i];

       cout<<(char)crypted;
       i++;

       if(i>= msg.length())
       break;
    }
return 0;
```

***Vigenere Cipher* Modifikasi :**

```
    string keyword;
    string keyword_encrypt;
    string msg;
    int crypted;
    int real_difference;
    int i=0;
    int k;
    int l;
    cout<<"Enter the keyword : ";
    getline(cin, keyword);

    cout<<"Enter the message you
want to crypt : ";
    getline(cin, msg);

    k = keyword.length();
    l = msg.length();
    keyword_encrypt = keyword;
    loop1:
    if(k < l);
    {
       while (i<=
keyword_encrypt.length())
       {
          keyword[i] -= 'a'- 1;
          if((keyword_encrypt[i] +
keyword[i]) > 'z')
             crypted = 'a' +
(keyword[i]+ keyword_encrypt[i]
- 'z') - 1;
          else if(msg[i] == 32)
             crypted =
keyword_encrypt[i];
          else
             crypted =
keyword_encrypt[i] +
keyword[i];

          cout<<(char)crypted;
           i++;

           if(i>=
keyword_encrypt.length())
           break;
       }

       keyword = keyword +
crypted;
       k = keyword.length();
       if(k < l)
          keyword_encrypt =
crypted;
          goto loop1;
    }

    cout<<"\n\n";
    crypted = “”;
    while (i<= msg.length())
    {
       keyword[i] -= 'a'- 1;
       if((msg[i] + keyword[i]) >
'z')
          crypted = 'a' +
(keyword[i]+msg[i] - 'z') - 1;
       else if(msg[i] == 32)
          crypted = msg[i];
       else
          crypted = msg[i] +
keyword[i];

       cout<<(char)crypted;
       i++;
```

```
            if(i>= msg.length())
            break;
        }
    return 0;
```

Contohnya sebagai berikut :

Vigenere Cipher biasa :
Plainteks :
CRYPTO IS SHORT FOR CRYPTOGRAPHY

Kunci :
abcdab cd abcda bcd abcdabcdabcd

Cipherteks :
CSASTP KV SIQUT GQU CSASTPIUAQJB

Vigenere Cipher modifikasi :
Plainteks :
CRYPTO IS SHORT FOR CRYPTOGRAPHY

Kunci :
abcdac eg adgja eim afkpagmsahov

Cipherteks :
CSASTQ MY SKUAT JWD CWIETUSJAWVY

Bila dilakukan metode Kasiski dan frekuensi analisis pada algoritma VIgenere Cipher biasa maka akan didapat bahwa :

Jarak CS – CS = 16
Faktor pembagi 16 : 4
Kesimpulan panjang kunci = 4

Setelah didapat panjang kunci maka dilakukan frekuensi analisis untuk cipherteks tersebut.
Dengan dibagi menjadi 4 kelompok maka akan didapat kelompok-kelompok seperti berikut :

- Kelompok 1

  CTSTCTA

- Kelompok 2

  SPIGSPQ

- Kelompok 3

  AKQQAIJ

- Kelompok 4

  SVUUSUB

Dari kelompok-kelompok tersebut dilakukan frekuensi analisis.
Kelompok 1 :

C = 2
T = 3
S = 1
A = 1

Kelompok 2 :

S = 2
P = 2
I = 1
G = 1
Q = 1

Kelompok 3 :

A = 2
K = 1
Q = 2
I = 1
J = 1

Kelompok 4 :

S = 2
V = 1
U = 3
B = 1

Dengan dicoba-coba dengan segala kemungkinan akan ditemukan kuncinya : abcd. Namun membutuhkan usaha yang besar.

Sedangkan bila dilakukan metode Kasiski dan frekuensi analisis maka hasil yang didapat dari algoritma Vigenere Cipher modifikasi tidak akan ditemukan kuncinya. Karena pada dasarnya metode Kasiski mencari pasangan huruf yang berulang, sedangkan bila menggunakan cara yang telah dimodifikasi munculnya pengulangan akan jauh berkurang daripada *Vigenere Cipher* biasa.

6. Kesimpulan

Bahwa algoritma *Vigenere Cipher* merupakan algoritma yang sangat kuat sebelum dipecahkan oleh Kasiski. Sehingga diperlukan beberapa modifikasi pada algoritma tersebut agar metode

Kasiski menjadi tidak efektif karena kesulitan mencari panjang kata kunci.

Algoritma *Vigenere Cipher* dengan modifikasi pengenkripsian pada kuncinya dapat mengurangi atau meminimalis kegunaan dari metode Kasiski. Karena metode Kasiski yang pada dasarnya mencari pasangan huruf yang berulang hamper tidak dapat menemukan pasangan huruf yang berulang. Walaupun ada pun kecil kemungkinannya bila pasangan huruf tersebut pada plainteksnya adalah sama.

7. Daftar Referensi

[MUN06] Munir, Rinaldi, *Kriptografi*, Institut Teknologi Bandung, 2006.

[WIK09] http://en.wikipedia.org , diakses pada bulan April 2009.