Rancangan Algoritma Kriptografi Variasi Vigenere Cipher dan Spiral

Muhammad Ihsan (13504120)[1)]
1) Program Studi Teknik Informatika ITB, Bandung 40132, email: if14120@students.if.itb.ac.id

Abstract *– Makalah ini membahas satu jenis algoritma enkripsi dan dekripsi yang diusulkan yang merupakan variasi dari algoritma cipher. Pada prinsipnya algoritma ini terdiri atas dua tahap inti yang pertama, mengaplikasikan variasi algoritma vigenere cipher dan kedua, menyusun hasil enkripsi pertama menjadi bentuk spiral.*

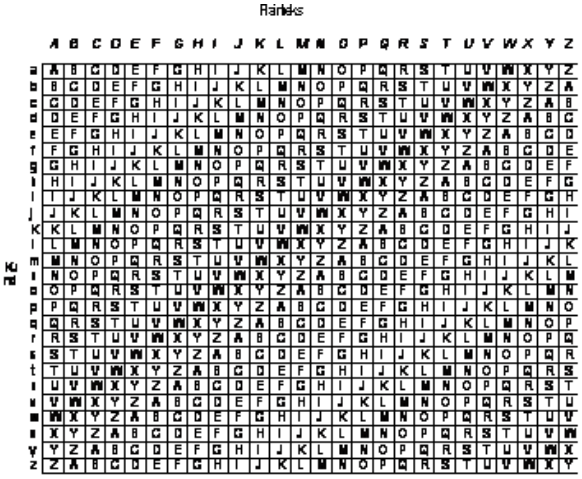
Sebelum diaplikasikan variasi algoritma vigenere cipher, setiap karakter dan kunci yang akan dienkripsi direpresentasikan dalam bentuk 8 bit. Karena dalam variasi ini yang diusulkan adalah vigenere cipher dalam operasi biner XOR setiap bit karakter dan bit kunci. Setelah dihasilkan bit operasi biner ini direpresentasikan lagi dalam bentuk karakter.
Setelah itu baru disusun hasil enkripsi secara spiral. Dengan alasan agar lebih aman, hasil enkripsi dalam bentuk spiral ini ditulis ulang dengan pembacaan secara biasa.

Untuk proses dekripsi merupakan proses sebaliknya. Pertama cipherteks hasil enkripsi yang akan didekripsi ditulis ulang dahulu dalam bentuk spiral. Kemudian setelah terbentuk, dilakukan pencarian titik pusat spiral, dan selanjutnya dibaca secara spiral. Hasil pembacaan ini direpresentasikan lagi dalam bentuk 8 bit. Kemudian kunci yang digunakan juga direpresentasikan dalam bentuk 8 bit. Ciphertеks dalam bentuk 8 bit tadi dioperasi XOR kan dengan kunci dalam bentuk 8 bit. Dan diperolehlah plainteks.

Kata Kunci*:operasi biner, vigenere cipher,spiral.*

1. PENDAHULUAN

Vigenere cipher yang merupakan chiperabjad-majemuk dipublikasikan oleh Blaise de Vigenere pada tahun 1586. Namun pada pertengahan abad 19 chiper ini dipecahkan oleh Babbage dan Kasiski. *Vigenere chiper* menggunakan Bujursangkar *Vigenere* untuk melakukan enkripsi. Setiap baris di dalam bujursangkar yang diperoleh dengan *Caesar Cipher*. Berikut bujursangkar *Vigenere* :



Gambar 1.1 Bujursangkar Vigenere

Jika panjang kunci lebih pendek daripada panjang plainteks, maka kunci diulang secara periodik. Sebagai contoh ;

plainteks : `INI TUGAS MAKALAH`
kunci : `ihs anihs anihsan`
cipherteks : `TUA THOHK MNSHDAU`

Huruf yang sama tidak selalu dienkripsi menjadi huruf yang sama pula. Inilah yang menjadi karakteristik dari *cipher* abjad majemuk. Pada cipher substitusi sederhana setiap huruf selalu mengantikan huruf tertentu. Vigenere cipher sendiri dapat mencegah frekuensi huruf-huruf dalam cipherteks seperti pada *cipher* abjad tunggal.
Namun jika periode kunci diketahui, kunci dapat dipecahkan dengan melakukan *exhaustive key search*. Orang yang pertama kali memecahkan *Vigenere cipher* adalah Friedrich Kasiski, dialah yang menemukan metode membantu menemukan panjang kunci.
Metode Kasiski memanfaatkan keuntungan bahwa bahasa Inggris juga mengandung perulangan pasangan huruf. Perulangan ini memungkinkan adanya kriptogram yang berulang, walaupun demikian tidak selalu terjadi kriptogram yang sama dari pasangan huruf yang dienkripsi. Ini karena adanya faktor kelipatan dari jarak dua buah pasangan huruf dengan panjang kunci.

Dengan mempertimbangkan itu semua tujuan yang diharapkan dari metode Kasiski adalah mencari dua atau lebih kriptogram yang berulang untuk menentukan panjang kunci. Baru setelah itu ditentukan kata kunci, bisa menggunakan *exhaustrive key search* atau yang lebih mangkus teknik analisis frekuensi.

Oleh karena itu semua penulis mencoba membuat suatu algoritma yang bisa menghasilkan cipherteks dengan syarat-syarat ;

- Menghindari munculnya enkripsi huruf maupun pasangan huruf yang menjadi cipherteks yang sama.
- Menghindari faktor kelipatan jarak dua buah pasangan huruf.

2. Algoritma Variasi Vigenere Cipher dan Spiral

Sebagaimana yang telah disebutkan pada pendahuluan di atas, agar mempersulit pemecahan cipherteks menjadi plainteks dan menjaga agar kunci tidak terpecahkan, perlu dibangun suatu algoritma yang tidak terpecahkan. Ada 2 hal yang akan dibahas pada bagian ini, yang pertama bagaimana proses dalam mengenkripsi plainteks menjadi cipherteks. Yang kedua, bagaimana mengubah cipherteks menjadi plainteks lagi.

2.1. Enkripsi

Untuk mengenkripsi plainteks menjadi cipherteks, ada beberapa tahap yang harus dilakukan. Pertama adalah mengaplikasikan variasi vigenere cipher. Sebelum dienkripsi setiap karakter plainteks diubah menjadi bentuk 8 bit. Begitu juga dengan kunci diubah menjadi bentuk 8 bit. Sebenarnya ini bisa diibaratkan dengan tabel ASCII yang memuat 255 karakter, dimana setiap nomor baris tabel merujuk ke satu karakter, dan nomor baris tersebut ditulis ulang dalam bentuk biner yang terdiri dari 8 bit.

Sebagai contoh;

Plainteks : INI TUGAS MAKALAH

Berikut nilai ASCII karakter ;

I : 01001001
N : 01001110
T : 01010100
U : 01010101
G : 01000111
A : 01000001
S : 01010011
M : 01001101
K : 01001011
L : 01001100
H : 01001000

Representasi 8 bit plainteks

01001001 01001110 01001001 01010100
01010101 01000111 01000001 01010011
01001101 01000001 01001011 01000001
01001100 01000001 01001000

Kunci : ihsan

i : 01101001
h : 01101000
s : 01110011
a : 01100001
n : 01101110

representasi 8 bit kunci :

01101001 01101000 01110011 01100001
01101110

Bit per bit plainteks dan kunci dioperasi binerkan dengan operasi XOR. Untuk kunci yang lebih pendek dari plainteks, kunci ditulis berulang-ulang. Setiap karakter pada kunci akan berpadanan dengan satu karakter pada plainteks. Walaupun nantinya bentuk 8bit, setiap satu karakter plainteks yang terdiri dari 8 bit akan berpadanan dengan satu karakter kunci yang terdiri dari 8 bit. Dan operasi yang dilakukan bukan untuk satu blok karakter 8 bit, namun per satu bit. Sehingga karakter pengganti yang akan dihasilkan dari suatu plainteks akan sangat susah untuk dicari frekuensinya. Karena kemungkinan menghasilkan pasangan karakter yang sama sangat sulit terjadi.

Sehingga operasi biner XOR plainteks dan kunci :

01001001 01001110 01001001 01010100
01101001 01101000 01110011 01100001 XOR
00100000 00100110 00111010 00110101

01010101 01000111 01000001 01010011
01101110 01101001 01101000 01110011 XOR
00111011 00101110 00101001 00100000

01001101 01000001 01001011 01000001
01100001 01101110 01101001 01101000 XOR
00101100 00101111 00100010 00101001

01001100 01000001 01001000
01110011 01100001 01101110 XOR
00111111 00100000 00100110

Gambar 2.1 Deskripsi operasi biner XOR

Dengan menggunakan tabel ASCII, berikut karakter yang bersesuaian dengan representasi di atas :

00100000 : ~
00100110 : &
00111010 : :
00110101 : 5
00111011 : ;
00101110 : .
00101001 :)
00100000 : ~
00101100 : ,
00101111 : /
00100010 : “

00101001 :)
00111111 : ?
00100000 : ~
00100110 : &

Hasil operasi biner dalam bentuk bit seperti contoh gambar diatas, diubah menjadi karakter lagi seperti dalam gambar berikut ini :

~&:5;.)~,/“)?~&

Gambar 2.2 Hasil representasi karakter dari hasil operasi biner XOR 8 bit

Walaupun sudah sulit untuk bisa memecahkan suatu huruf, penulis mengembangkan lebih jauh hasil enkripsi pertama ini, dengan masuk ke tahap kedua,yaitu menyusun rangkaian karakter tersebut dengan pola spiral. Algoritmanya sebagai berikut :

a. Karakter pertama akan menjadi pusat spiral
b. Karakter berikutnya mulai menyusun ke kiri, setelah itu ke bawah, ke kanan dan begitu seterusnya mengikuti pola putaran jarum jam.

Untuk lebih jelasnya, pola penyusunan karakter sebagaimana contoh tadi dapat digambarkan pada pengisian tabel berikut ini :

Tabel 1 Langkah Pertama Penyusunan

~	⇨ &

Tabel 2 Langkah Kedua

~	& ⇩
	:

Tabel 3 Langkah Ketiga

~	&
5 ⇦	:

Tabel 4 Bentuk Final Dalam Pola Spiral

)	~	,	/
.	~	&	“
;	5	:	)
	&	~	?

Sebagai tahap akhir hasil enkripsi dengan pola spiral ini dituliskan lagi dengan pembacaan secara biasa, mulai dari baris pertama kolom sebelah kanan.
Sehingga hasil pembacaan cipherteks diatas adalah :

)~,?.~&”;5:)&~?

Dengan pola ini, walaupun mencurigakan, namun tidaklah mudah untuk mendapatkan plainteks walaupun sudah memperoleh kunci. Karena pola penulisan dengan spiral.

2.2. Dekripsi

Untuk proses dekripsi sebagai kebalikan, untuk memperoleh plainteks dari cipherteks tersebut. Tentu saja harus mengubah susunan karakter yang ada pada cipherteks dari struktur spiral yang ditulis ulang menjadi struktur linear biasa.

Setelah diperoleh kembali dalam pola biasa, barulah diubah karakter tersebut dalam bentuk 8 bit. Demikian juga dengan kunci yang diberikan, juga diubah dalam bentuk 8 bit.

Salah satu keistimewaan operasi biner XOR, adalah jika A XOR B = C, maka C XOR B = A. Karena itu, untuk mengubah cipherteks tadi, cukup dengan mengoperasi binerkan cipherteks dalam bentuk 8 bit dengan kunci dalam bentuk 8 bit, dengan operasi XOR. Dari hasil operasi ini diperolehlah plainteks yang masih dalam bentuk 8bit-8bit. Langkah terakhir adalah mengubah 8bit tersebut menjadi karakter-karakter.

3. HASIL DAN PEMBAHASAN

Dari proses-proses yang ditulis pada bagian 2, tergambar dengan jelas bagaimana menjaga keamanan data-data plainteks menjadi cipherteks yang sangat sulit dipecahkan.

Penulis yakin sulit untuk dipecahkan, karena operasi dilakukan pada level bit, dimana setiap bit-bit yang bersesuaian antara bit plainteks dengan bit kunci dioperasi biner kan. Untuk sekedar menebak oleh para pencuri data, ada 4 operasi biner yang mungkin, namun tidaklah mudah menemukan pola operasi biner yang digunakan dalam pengenkripsian plainteks ini.

Selain itu, penulis menyarankan menggunakan operasi biner, untuk membingungkan pencuri data menemukan karakter yang mungkin menggantika karakter yang ada pada plainteks.

Untuk tahap kedua, agar menjaga kalau seandainya pencuri data berhasil memecahkan bahwa ini merupakan operasi biner, penulis mengembangkan pola penyusunan karakter dalam bentuk spiral. Sehingga metode kasiski dan analisis kemunculan huruf tidak berhasil memecahkan cipherteks hasil enkripsi algoritma variasi *vigenere cipher* ini.

Setelah dituliskan secara spiral, untuk membuat kemunculan huruf atau karakter benar-benar acak

adalah dengan membaca ulang secara biasa dari bagian terluar dari hasil penulisan spiral. Akan terjadi bermacam-macam pola pasangan huruf dan muncul seperti kalimat terputus, karena prinsip pembacaan bertentangan dengan pola penyusunan huruf secara spiral.

Dengan mempertimbangkan semuanya sangat sulit untuk menemukan kunci suatu cipherteks. Namun jika pencuri data berhasil mencuri kunci dari penulis pesan, dan tidak mengerti pola pembangkitan karakter pengganti dari plainteks, tetap tidak akan terpecahkan. Walaupun demikian, dengan menuliskan program dengan kinerja super cepat, mungkin suatu saat nanti sampai tahap ini berhasil dipecahkan, penulis mengantisipasinya dengan melakukan operasi bit per bit setiap karakter yang diwakilinya. Sebenarnya operasi biner yang diterapkan tidak harus operasi XOR, namun karena operasi ini "cukup unik" karena dengan kunci dan operasi yang sama bersifat bolak-balik, dari plainteks dan kunci dengan operasi ini bisa menghasilkan cipherteks, dan begitu juga sebaliknya. Dengan cipherteks dan kunci dengan operasi yang sama XOR menghasilkan plainteks.
Meskipun dengan pola yang unik ini bisa memudahkan untuk mendapatkan data, namun penulis yakin cipherteks dalam bentuk terakhir yang tinggal dioperasikan XOR ini sulit untuk diperoleh dengan pertimbangan penyusunan dengan spiral tadi.

4. KESIMPULAN

Untuk memperoleh suatu algoritma yang tidak terpecahkan, amat sulit untuk diimplementasikan. Banyak pembuat algoritma yang mengklaim bahwa algoritma yang ditulisnya merupakan algoritma yang mangkus, yang tahan dari serangan. Padahal untuk membuat suatu algoritma yang tidak terpecahkan tidak semudah itu. Amat sulit dan kadang membutuhkan banyak analisa. Sebagai contoh, beberapa serangan yang mungkin muncul; dengan analisis kemunculan huruf atau pasangan huruf. Baik itu frekuensi kemunculan dibandingkan dengan hasil penelitian huruf yang paling sering banyak dipakai, maupun menganalisis jarak dua pasangan huruf yang sering muncul. Selama ini, huruf yang paling sering muncul di plainteks, paling sering juga muncul pada cipherteks.
Sebagaimana yang terjadi pada vigenere cipher, setelah 3 abad berhasil dipecahkan, apalagi dengan teknologi di zaman sekarang ini, dengan menuliskan program dan diproses oleh komputer, pencarian yang jika dilakukan oleh tangan manusia berbulan-bulan, bisa diselesaikan hanya dalam hitungan detik. Sehingga tidak tertutup kemungkinan bisa dipecahkan. Untuk itulah penulis mengusulkan suatu algoritma yang sulit dipecahkan, yang memiliki angka kemungkinan yang sangat besar.

Walaupun dilakukan dengan komputer dengan prosesor dan memori yang tinggi, membutuhkan waktu yang lama untuk dapat memecahkan kunci dan mencuri data yang dienkripsi. Sehingga disaat terpecahkan, informasi ini tidak lagi berguna.

Diharapkan untuk pengembangan selanjutnya, pola penyusunan karakter dapat lebih bagus dari seperti yang diajukan sekarang.

DAFTAR REFERENSI

[1] R. Munir, "Catatan Kuliah Kriptografi", 2006.