Vigènere Chiper dengan Modifikasi Fibonacci

Anggriawan Sugianto / 13504018
Teknik Informatika - STEI - ITB, Bandung 40132, email: if14018@students.if.itb.ac.id

Abstrak *- Vigènere chiper merupakan salah satu algoritma kriptografi klasik yang menggunakan chiper substitusi. Algoritma ini termasuk algoritma substitusi abjad-majemuk. Algoritma ini cukup populer karena mudah dimengerti dan diimplementasikan.*

Namun, Vigènere chiper relatif mudah untuk dipecahkan dengan kriptanalisis, yakni dengan memanfaatkan perulangan huruf ataupun perulangan pasangan huruf. Perulangan huruf ini mungkin untuk membangkitkan perulangan pada chiperteks. Teknik ini digunakan pada metode Kasiski yang biasa dipakai untuk memecahkan Vigènere chiper. Karena itu, Vigènere chiper biasa merupakan breakable chiper (chiper yang dapat dipecahkan).

Untuk membuat unbreakable chiper (chiper yang tidak dapat dipecahkan), hal yang harus dilakukan adalah menggunakan kunci yang benar-benar acak dan panjang kunci sama dengan panjang plainteks. Salah satu algoritma kriptografi yang tidak dapat dipecahkan adalah one-time pad, yang menggunakan deretan karakter kunci yang dibangkitkan secara acak. Sayangnya, algoritma ini tidak efisien karena bermasalah saat menyimpan dan mendistribusikan kunci yang sangat panjang.

Dengan memanfaatkan sifat algoritma untuk unbreakable chiper, seperti pada one-time pad, Vigènere chiper bisa dikembangkan menjadi lebih susah untuk dipecahkan. Caranya adalah dengan membangkitkan deretan karakter kunci yang "acak" sepanjang plainteks dengan memanfaatkan sifat bilangan Fibonacci terhadap kunci masukan dari pengguna. Kunci masukan ini pun tidak perlu sangat panjang, sehingga memudahkan untuk disimpan dan didistribusikan.

Kata Kunci*: Vigènere chiper, Fibonacci*

1. PENDAHULUAN

Dewasa ini perkembangan teknologi informasi dan komunikasi sudah berkembang sangat pesat. Hampir di setiap bidang kehidupan telah menggunakan teknologi ini sebagai saran pendukung maupun sarana utama. Sehubungan dengan hal ini, aspek keamanan dalam teknologi informasi dan komunikasi tentunya tidak bisa diabaikan. Dalam kegiatan kirim-terima pesan, aspek keamanan yang perlu diperhatikan antara lain kerahasiaan, integritas data, otentikasi, dan nirpenyangkalan [1]. Aspek keamanan tersebut bisa dijaga dengan memanfaatkan kriptografi.

Pada umumnya, algoritma kriptografi bisa dibagi menjadi dua, yakni kriptografi klasik dan kriptografi modern. Kriptografi klasik biasanya menggunakan algoritma yang sederhana dan berbasiskan karakter. Sedangkan kriptografi modern biasanya menggunakan algoritma yang kompleks dan beroperasi dalam mode bit, sehingga lebih susah untuk dipecahkan.

Algoritma kriptografi klasik terdiri dari dua macam, yaitu *chiper* substitusi dan *chiper* transposisi. *Chiper* substitusi menyandikan plainteks dengan cara mengganti setiap karakter dengan karakter lain dalam susunan abjad. Jenis-jenis *chiper* substitusi ini antara lain *chiper* abjad-tunggal, *chiper* abjad-majemuk, chiper substitusi homofonik, dan *chiper* substitusi poligram. Sedangkan *chiper* transposisi menyandikan plainteks dengan cara melakukan *transpose* terhadap rangkaian karakter di dalam plainteks.

Algoritma kriptografi klasik ini menarik untuk dipelajari karena mudah dipahami dan mudah diimplementasikan. Selain itu, kriptografi klasik merupakan dasar dari algoritma kriptografi modern.

2. VIGÈNERE CHIPER

2.1. Konsep Dasar

Vigènere *chiper* merupakan salah satu contoh *chiper* abjad-majemuk (*polyalphabetic substitution chiper*). *Chiper* abjad-majemuk akan mengganti setiap karakter pada plainteks dengan karakter lain yang mungkin berbeda-beda pada chiperteksnya. Vigènere *chiper* menggunakan bujur sangkar Vigènere untuk melakukan enkripsi. Setiap baris di dalam bujur sangkar menyatakan huruf-huruf chiperteks yang diperoleh dengan *Caesar chiper*, di mana jauh pergeseran huruf plainteks ditentukan oleh nilai desimal dari huruf kunci tersebut (A = 0, B = 1, C = 3, ... , Z = 25).

Untuk melakukan enkripsi dengan Vigènere *chiper*, lakukan pada bujur sangkar Vigènere sebagai berikut: tarik garis vertikal dari huruf plainteks ke bawah, lalu tarik garis mendatar dari huruf kunci ke kanan. Perpotongan kedua garis tersebut menyatakan huruf chiperteksnya.

Pada Vigènere *chiper*, jika panjang kunci lebih pendek daripada panjang plainteks, maka kunci tersebut akan diulang penggunaannya.

Contoh penggunaan Vigènere *chiper*:

```
P : SAYASUKAKRIPTOGRAFI
K : MUSIKMUSIKMUSIKMUSI
C : EUQICGESSBUJLWQDUXQ
```

Pada contoh di atas, plainteks "SAYASUKAKRIPTOGRAFI" dienkripsi dengan kunci "MUSIK" menghasilkan chiperteks "EUQICGESSBUJLWQDUXQ".

Perhatikan bahwa huruf A pada plainteks disubstitusi dengan huruf yang berbeda-beda pada chiperteks, yakni U, I, S. Hal inilah yang menyebabkan Vigènere *chiper* termasuk *chiper* abjad-majemuk.

Aturan enkripsi pada Vigènere *chiper* bisa dinyatakan juga sebagai penjumlahan modulo 26 dari satu karakter plainteks dengan satu karakter kunci.

$$C_i = (P_i + K_i) \bmod 26 \quad (1)$$

di mana

P_i : karakter plainteks
K_i : karakter kunci
C_i : karakter chiperteks

Dekripsi pada Vigènere *chiper* dilakukan dengan cara yang berkebalikan, yaitu dengan cara menarik garis horizontal dari huruf kunci sampai ke huruf chiperteks yang dituju, lalu dari huruf cipherteks tarik garis vertikal ke atas sampai ke huruf plainteks. Atau, bisa juga dinyatakan dalam persamaan:

$$P_i = (C_i - K_i) \bmod 26 \quad (2)$$

2.2. Kekuatan

Kekuatan algoritma Vigènere *chiper* ini adalah dapat mencegah frekuensi huruf-huruf di dalam chiperteks yang memiliki pola tertentu yang sama, seperti yang terjadi pada *chiper* abjad-tunggal. Pada *chiper* abjad-tunggal, huruf yang paling sering muncul di chiperteks merupakan substitusi dari huruf yang paling sering muncul di plainteks. Karena itu, dengan teknik analisis frekuensi, kriptanalis bisa dengan mudah menebak huruf tersebut. Namun, pada Vigènere *chiper* hal tersebut tidak bisa dilakukan karena satu macam huruf pada plainteks mungkin dienkripsi menjadi beberapa macam huruf pada chiperteks, seperti pada contoh sebelumnya.

2.3. Kelemahan

Vigènere *chiper* memungkinkan perulangan huruf atau pasangan huruf pada plainteks terjadi juga pada chiperteksnya. Hal ini dikarenakan kunci yang digunakan untuk melakukan enkripsi juga diulang. Akibatnya, bagian plainteks dan bagian kunci tertentu bisa "berpasangan" lebih dari satu kali. Contoh:

```
P : SAYACINTAPACARSAYA
K : KASIHKUKASIHKUKASI
C : CAQIJSHDAHIJKLCAQI
```

Terlihat pada contoh di atas, SAYA dienkripsi menjadi kriptogram yang sama, yaitu CAQI. Namun, perlu diperhatikan bahwa kasus seperti ini tidak selalu demikian, misalnya pada contoh berikut ini:

```
P : SAYACINTAPACARSAYA
K : SAYANGKUSAYANGKUSA
C : KAWAPOXNSPYCNXCUQA
```

Pada contoh di atas, SAYA tidak dienkripsi menjadi kriptogram yang sama.

Sifatnya yang mungkin untuk menghasilkan kriptogram yang sama terhadap bagian plainteks yang sama ini menjadi kelemahan Vigènere *chiper*.

2.4. Metode Kasiski

Metode Kasiski memanfaatkan kelemahan Vigènere *chiper* yang mungkin menghasilkan kriptogram yang sama untuk bagian plainteks yang sama. Pada contoh di atas, di mana SAYA dienkripsi menjadi kriptogram yang sama, yaitu CAQI, secara intuitif menunjukkan bahwa jika jarak antara dua buah *string* yang berulang pada plainteks merupakan kelipatan dari panjang kunci, maka *string* yang sama tersebut akan muncul menjadi kriptogram yang sama pula pada chiperteks. Pada contoh pertama di atas, jarak antara *string* SAYA adalah 14, dan panjang kunci KASIHKU adalah 7. Sedangkan pada contoh kedua, jarak antara *string* SAYA adalah 14, dan panjang kunci SAYANGKU adalah 8.

Dengan metode Kasiski, kriptanalis bisa memperoleh panjang kunci dari suatu Vigènere *chiper*. Caranya adalah:

1. Mencari semua jarak kriptogram yang berulang pada chiperteks.
2. Mencari faktor pembagi terbesar dari jarak-jarak tersebut. Faktor pembagi ini menyatakan panjang kunci yang mungkin.

Contoh:

```
QWERTYUIOPASDQWERASDJKLZXCVN
```

Kriptogram yang berulang adalah QWER dan ASD. Jarak antara dua perulangan QWER adalah 14. Jarak antara dua perulangan ASD adalah 7. Faktor pembagi

terbesar 14 dan 7 adalah 7. Dengan demikian, kemungkinan besar panjang kunci adalah 7.

Jika panjang kunci telah diketahui, maka kunci dapat ditentukan dengan beberapa cara, antara lain:

1. *Exhaustive key search*, yakni dengan membangkitkan semua kemungkinan kunci. Jika panjang kunci adalah *p*, maka cara ini membutuhkan 26^p kali percobaan.
2. Mengelompokkan huruf-huruf pada chiperteks ke sejumlah *p* kelompok (*p* = panjang kunci). Lalu, melakukan teknik analisis frekuensi pada tiap-tiap kelompok tersebut.

Dengan demikian, Vigènere *chiper* merupakan *chiper* yang bisa terpecahkan (*breakable chiper*).

3. ONE-TIME PAD

3.1. Konsep Dasar

Untuk membuat suatu *unbreakable chiper* (*chiper* yang tidak bisa dipecahkan), ada dua syarat yang harus dipenuhi, yaitu:

1. Kunci yang digunakan harus benar-benar acak
2. Panjang kunci harus sama dengan panjang plainteks

Satu-satunya algoritma kriptografi yang tidak dapat dipecahkan adalah *one-time pad*. *One-time pad* berisi deretan karakter-karakter kunci yang dibangkitkan secara acak. Satu *pad* hanya digunakan sekali saja untuk mengenkripsi pesan, setelah itu *pad* yang telah digunakan dihancurkan supaya tidak dipakai kembali untuk mengenkripsi pesan yang lain.

Aturan enkripsi dan dekripsi pada *one-time pad* sama seperti pada Vigènere *chiper*. Contohnya:

```
P : SELAMATPAGIDUNIA
K : QOIJFMKVMVZLATIM
C : ISTJRMDKMBHOUGQM
```

3.2. Kekuatan

Kekuatan utama dari algoritma *one-time pad* adalah tidak bisa dipecahkan. Hal ini dikarenakan:

1. Barisan kunci acak yang ditambahkan ke pesan plainteks tidak acak akan menghasilkan chiperteks yang seluruhnya acak.
2. Beberapa barisan kunci yang digunakan untuk mendekripsikan chiperteks bisa menghasilkan beberapa pesan plainteks yang memiliki makna, sehingga kriptanalis bingung menentukan plainteks mana yang benar. Contohnya:

   Untuk chiperteks seperti pada contoh sebelumnya
   `ISTJRMDKMBHOUGQM`
   Jika kriptanalis mencoba barisan kunci
   `WOGDKMOQUSDFUWES`
   Maka plainteks yang dihasilkan
   `MENGHAPUSJEJAKMU`

   Sedangkan jika kriptanalisis mencoba barisan kunci
   `HYTQRCJRIKPKHIWA`
   Maka plainteks yang dihasilkan
   `BUATAKUTERSENYUM`

3.3. Kelemahan

Meskipun *one-time pad* merupakan *chiper* yang tidak bisa dipecahkan, algoritma ini jarang digunakan. Hal ini dikarenakan untuk pesan yang sangat panjang, dibutuhkan kunci yang sangat panjang juga.

Pada aplikasi kriptografi untuk data tersimpan, timbul masalah dalam penyimpanan kunci, yakni bagaimana menyimpan kunci dengan aman, yang panjangnya sama dengan panjang pesan yang dienkripsi.

Sedangkan pada aplikasi kriptografi untuk komunikasi pesan, timbul masalah dalam pendistribusian kunci, yakni bagaimana kunci dapat dikirim secara aman, apakah melalui saluran komunikasi yang sama dengan saluran untuk pesan (hal ini tentu mengganggu *traffic* pesan yang padat dan memerlukan kunci yang berlapis untuk melindungi kunci yang dikirim), atau melalui saluran komunikasi kedua yang umumnya lambat dan mahal.

4. VIGÈNERE CHIPER DENGAN MODIFIKASI FIBONACCI

4.1. Konsep Dasar

Vigènere *chiper* dengan modifikasi Fibonacci dirancang oleh penulis untuk mengurangi kelemahan pada Vigènere *chiper* biasa. Pada dasarnya, algoritma baru ini mengadopsi sifat *one-time pad* yang menggunakan kunci yang acak dan panjangnya sama dengan panjang plainteksnya. Dengan memanfaatkan modifikasi sifat bilangan Fibonacci terhadap kunci masukan, kriptografer bisa membangkitkan barisan kunci yang "acak" yang panjangnya sama dengan panjang plainteks.

Adapun bilangan Fibonacci (0,1,1,2,3,5,8,13,...) memiliki sifat sebagai berikut:

$$U_n = U_{n-1} + U_{n-2} \qquad (3)$$

di mana

U_n = karakter kunci ke-*n*

Pada algoritma baru ini, sifat tersebut dimodifikasi menjadi:

$$U_n = (U_{n-k} + U_{n-k+m}) \bmod 26 \qquad (4)$$

di mana

k = panjang kunci masukan

$m = 1 + (\sum$ (karakter_tiap_kunci) mod (k-1))

Contoh:

Plainteks (P) : SATUDALAMNADACINTA
Kunci Masukan (K) : KLASIK

$k = 6$

$m = 1 + ((10+11+0+18+8+10) \bmod 5) = 3$

$U_7 = U_1 + U_4 = K + S = C$
$U_8 = U_2 + U_5 = L + I = T$
$U_9 = U_3 + U_6 = A + K = K$
$U_{10} = U_4 + U_7 = S + C = U$
...

P : SATUDALAMNADACINTA
K' : KLASIKCTKUBUWUEQVY
C : CLTMLKNTWHBXWWMDOY

4.2. Implementasi

Contoh program kecil yang mengimplementasikan algoritma di atas adalah sebagai berikut (ditulis dengan bahasa C++) :

```
// File   : VF.cpp

#include <cstdlib>
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

int main(int argc, char *argv[])
{
  if (argc < 5) {
    cout << "VF [E|D] [IN] [OUT] [KEY]";
    exit(1);
  }
  else {
    ifstream Fin (argv[2]);
    ofstream Fout(argv[3]);

    string key = argv[4];
    int len = key.length();

    if (len < 3) {
      cout << "Key at least 3 chars" << endl;
      exit(1);
    }

    char cc;

    /* Calculate m */
    int m = 0;
    for (int j=0; j<len; j++) {
      cc = key[j];
      if (cc >= 97 && cc <= 122 ) {
        cc -= 32;
      }
      m += (int) cc;
    }
    m = 1 + (m % (len-1));

    int i = 0;
    cc = Fin.get();
    while (cc != EOF) {
      if (cc >= 97 && cc <= 122 ) {
        cc -= 32;
      }
      if (cc >= 65 && cc <= 90) {
        int delta = key[0];

        if (argv[1][0] == 'E') {
          cc = ((cc + delta) % 26) + 65;
        }
        else { // 'D'
          cc = ((cc - delta + 26) % 26) + 65;
        }

        /* Generate key */
        char ccc = ((key[0]+key[m])%26)+65;
        for (int j=0; j<len-1; j++) {
          key[j] = key[j+1];
        }
        key[len-1] = ccc;

        i++;
      }
      Fout << cc;
      cc = Fin.get();
    }
    Fin.close();
    Fout.close();
  }
  return 0;
}
```

Cara menjalankan program di atas adalah sebagai berikut:

```
VF [E|D] [IN] [OUT] [KEY]
```

di mana

VF : nama program (Vigènere Fibonacci)
[E|D] : mode (E=enkripsi, D=dekripsi)
[IN] : file input (jika mode E, maka IN=plainteks; jika mode D, maka IN=chiperteks)
[OUT] : file output (jika mode E, maka OUT=chiperteks; jika mode D, maka OUT=plainteks
[KEY] : kunci masukan dari pengguna

4.3. Kekuatan

Seperti pada Vigènere *chiper* biasa, algoritma ini tidak dapat dipecahkan dengan teknik analisis frekuensi biasa karena termasuk *chiper* abjad-majemuk. Selain itu, berbeda dengan Vigènere *chiper* biasa, pada algoritma ini perulangan *string* di chiperteks yang sesuai dengan perulangan *string* di plainteks akan sangat jarang terjadi, bahkan mungkin tidak akan terjadi. Hal ini dikarenakan kunci yang digunakan tidak berulang. Karena itu, metode Kasiski, yang digunakan untuk menentukan panjang kunci pada Vigènere *chiper*, tidak bisa digunakan untuk menyerang algoritma ini.

4.4. Kelemahan

Kunci yang digunakan pada Vigènere *chiper* dengan modifikasi Fibonacci ini tidak benar-benar acak, seperti pada *one-time pad*, tetapi dibangkitkan dari kunci masukan. Hal ini merupakan kelemahan algoritma ini. Jika kriptanalis berhasil melakukan *known-plaintext attack*, maka ia akan mendapatkan suatu potongan kunci.

Jika panjang potongan kunci itu lebih dari panjang kunci masukan, maka kriptanalis bisa mencari nilai *m* pada persamaan (4) dengan mencari kombinasi 3 karakter yang tepat pada potongan kunci tersebut. Hal ini dikarenakan setidaknya ada 1 karakter (misalnya Q) pada potongan kunci itu yang merupakan hasil pembangkitan dari "penjumlahan" 2 karakter lainnya pada potongan kunci tersebut. Jika Q ditemukan dan untuk karakter-karakter di sebelah kanannya pun sama dengan "penjumlahan" 2 karakter sebelah kanan dari 2 karakter pembangkit Q, maka nilai *m* telah ditemukan. Jika nilai *m* telah ditemukan, kriptanalis bisa membangkitkan barisan kunci sisanya.

5. KRIPTANALISIS PADA VIGÈNERE CHIPER DENGAN MODIFIKASI FIBONACCI

5.1. *Chipertext-only attack*

Algoritma ini sangat kuat terhadap *chipertext-only attack*. Penggunaan sifat bilangan Fibonacci yang dimodifikasi menyebabkan tidak adanya bagian kunci yang berulang, sehingga tidak ada perulangan *string* pada chiperteks yang bersesuain dengan perulangan *string* pada plainteks. Karena itu, *chipertext-only attack* seperti teknik analisis frekuensi ataupun metode Kasiski tidak mungkin bisa menyerang algoritma ini.

5.2. *Known-plaintext attack*

Algoritma ini sangat rawan terhadap *known-plaintext attack*, sebagaimana dijelaskan di bab 4.4 tentang kelemahan algoritma ini. Contoh:

Diberikan chiperteks:
SINAREVRECIQWNYTNPJXMYRMQEOOBLRDOGZ

Dan diketahui bahwa *string* PJXMYRMQ ternyata kriptogram untuk *string* PERASAAN. Maka dari persamaan (1) atau

$$K_i = (C_i - P_i) \bmod 26 \qquad (5)$$

Didapatkan potongan kunci AFGMGRMD. Dari potongan kunci ini, kriptanalis mencari kombinasi 3 karakter yang tepat sesuai sifat Vigènere *chiper*, dan didapatkanlah A+G=G, F+M=R, G+G=M, M+R=D. Jadi, nilai *m* = 2.

Setelah itu, kriptanalis bisa membangkitkan barisan kunci yang lengkap, yaitu:
DEWAZEVEUIPMJUYGH**AFGMGRMD**SUEXWRAOWF

Lalu, pesan plainteksnya pun terungkap:
PERASAANKUTENTANG**PERASAAN**MUKEPADAKU

Dari barisan kunci yang lengkap itu, bisa diketahui bahwa kunci masukan adalah DEWA, yang panjangnya 4. Sedangkan panjang potongan kunci AFGMGRMD adalah 8. Karena itu, *known-plaintext attack* ini bisa berhasil memecahkan algoritma ini. Jika panjang potongan kunci tidak lebih dari panjang kunci masukan, serangan ini tidak akan berhasil.

5.3. *Chosen-plaintext attack*

Seperti halnya terhadap *known-plaintext attack*, algoritma ini pun sangat rawan terhadap *chosen-plaintext attack*. Bahkan, kriptanalisis akan lebih mudah karena kriptanalis bisa memilih pasangan plainteks-chiperteks tertentu yang langsung mengarah ke kunci masukan, yaitu pasangan paling kiri (awal).

6. KESIMPULAN DAN SARAN

6.1. Kesimpulan

Berdasarkan percobaan dan analisis yang sudah dituliskan di atas, penulis bisa menarik beberapa kesimpulan terkait Vigènere *chiper* dengan modifikasi Fibonacci.

- Vigènere *chiper* dengan modifikasi Fibonacci lebih baik daripada Vigènere *chiper* biasa karena menghilangkan perulangan kunci yang digunakan.
- Vigènere *chiper* dengan modifikasi Fibonacci sangat kuat terhadap serangan *chipertext-only*, berupa teknik analisis frekuensi ataupun metode Kasiski.
- Vigènere *chiper* dengan modifikasi Fibonacci sangat rawan terhadap serangan *known-plaintext* dan *chosen-plaintext*.

6.2. Saran

Saran yang penulis sampaikan terkait Vigènere *chiper* dengan modifikasi Fibonacci, antara lain:

- Bagi pengguna algoritma ini, sebaiknya menggunakan kunci masukan yang panjang.
- Bagi yang ingin mengembangkan algoritma ini, sebaiknya pembangkitan kunci lebih kompleks lagi, misalnya digabungkan dengan metode transposisi.

DAFTAR REFERENSI

[1] Munir, Rinaldi, *Diktat Kuliah IF5054 Kriptografi*, STEI ITB, 2007