Penyamaran Plainteks pada Algoritma Vigenere Chiper "StegaVig" Makalah IF5054 Kriptografi

Dyah Saptanti Perwitasari
Teknik Informatika ITB, Bandung 40135, email: if14017@students.if.itb.ac.id

Abstract – *Permasalahan umum yang sering muncul pada Vigènere Chiper adalah mengenai kemunculan kelompok huruf chiperteks yang berulang, yang mungkin berkorespondensi dengan kelompok plainteks yang sama, yang justru dapat menjadi celah bagi kriptanalis untuk memecahkannya. Makalah yang akan diajukan ini berisi solusi untuk mengatasi perulangan huruf beserta analisisnya. Solusi yang akan dibahas adalah penyamaran plainteks yang bersangkutan sebelum dilakukan enkripsi sehingga memperkecil kemungkinan pengulangan suatu kata pada plainteks. Penyamaran plainteks dilakukan dengan menyisipkan suatu partisi plainteks ke dalam partisi plainteks yang lain.*

Kata Kunci*: vigenere, penyamaran, teks.*

1. LATAR BELAKANG

Karakteristik dari algoritma kriptografi klasik adalah dalam hal kesederhanaannya sehingga algoritma kriptografi klasik sering disebut juga algoritma yang mudah dipecahkan. Terkait dengan karakteristik dalam hal kesederhanaan tersebut, Vigènere Chiper digolongkan sebagai salah satu jenis algoritma kriptografi klasik. Seperti halnya algoritma kriptografi klasik lainnya, Vigènere Chiper mudah dipecahkan. Penerapan algoritma Vigènere Chiper untuk kunci yang tak acak atau berulang, terutama untuk kalimat yang panjang memungkinkan terjadinya pengulangan kata pada plainteks yang kemudian dapat menjadi celah untuk melakukan kriptanalisis, misalnya dengan metode Kasiski. Solusi untuk mengurangi kemungkinan terjadinya pengulangan kata pada chiperteks ada dua pilihan, yaitu mengurangi pengulangan kata pada plainteks atau membangkitkan kunci seacak mungkin.

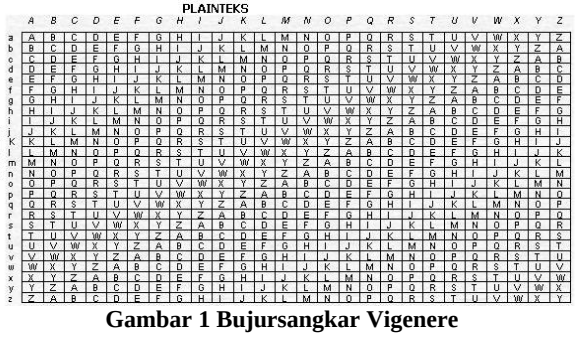
Kenyataannya, pembangkitan dua kunci acak yang sama merupakan suatu hal yang mustahil, kemungkinan untuk menghasilkan kunci yang sama persis sangatlah kecil. Karena itu pada makalah ini akan dibahas mengenai pendekatan untuk mengurangi kelemahan Vigènere Chiper dalam hal pengulangan kata dari segi plainteks, yaitu dengan mengurangi kemungkinan kemunculan huruf-huruf yang sama pada plainteks, dengan cara menyamarkan plainteks.

2. LANDASAN TEORI

2.1. Vigènere Chiper

Vigènere Chiper adalah algoritma yang dipublikasikan oleh Blaise de Vigènere, seorang diplomat Perancis. Vigènere Chiper merupakan *cipher* abjad-majemuk (*polyalpabetic substitution cipher*) sehingga huruf yang sama tidak selalu dienkripsi menjadi huruf cipheteks yang sama, karena setiap huruf chiperteks mempunyai kemungkinan banyak huruf plainteks dan sebaliknya.

Vigènere Cipher yang mempunyai beberapa varian ini menggunakan "Bujursangkar *Vigènere*" untuk melakukan enkripsi yang dapat dilihar pada Gambar 1 Bujursangkar Vigenere. Baris bagian atas bujursangkar menyatakan huruf plainteks, baris bagian kiri merupakan huruf kunci, sedangkan baris-baris di dalam bujursangkar menyatakan huruf-huruf cipherteks yang diperoleh dengan *Caesar Cipher.*



Gambar 1 Bujursangkar Vigenere

Pada algoritma ini, jika panjang kunci lebih pendek daripada panjang plainteks, misalnya untuk plainteks yang cukup panjang, maka kunci akan diulang secara periodik. Sedangkan rumus untuk melakukan enkripsi sama seperti Caesar Chiper, yaitu

$$C(P) = (P + K) \bmod 26 \quad \text{..... (1)}$$

dengan:
P : huruf plainteks
K : huruf kunci
C(P) : huruf chiperteks untuk huruf plainteks P

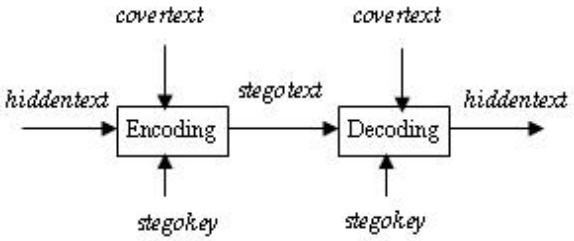
Meskipun algoritma ini dirancang untuk mengurangi pengulangan chiperteks yang sama pada chiper abjad tunggal pada umumnya, terjadinya pengulangan chiperteks tetap merupakan kelemahan terbesar Vigenere Chiper. Dengan metode Kasiski Vigenere

Chiper menjadi dapat dipecahkan dengan mudah, terutama untuk kalimat yang cukup panjang. Metode Kasiski digunakan untuk mencari panjang kunci yang mungkin dengan menganalisis pengulangan chiperteks.

2.2. Steganografi

Steganografi adalah ilmu dan seni menyembunyikan (*embedded*) informasi dengan cara menyisipkan pesan di dalam pesan lain (menyamarkan). Pesan dapat disamarkan melalui teks, gambar, audio, maupun video. Proses *encoding* dan *decoding* digambarkan seperti pada Gambar 2 Encoding dan Decoding pada Steganografi. Properti steganografi antara lain sebagai berikut:

1. *Embedded message* (*hiddentext)*, yaitu pesan rahasia yang disembunyikan.
2. *Cover-object* (*covertext)*, yaitu pesan asli yang digunakan untuk menyembunyikan *embedded message*.
3. *Stego-object* (*stegotext)*, yaitu pesan yang sudah berisi pesan *embedded message*.
4. *Stego-key*, yaitu kunci yang digunakan untuk menyisipkan pesan dan mengekstraksi pesan dari *stegotext*.

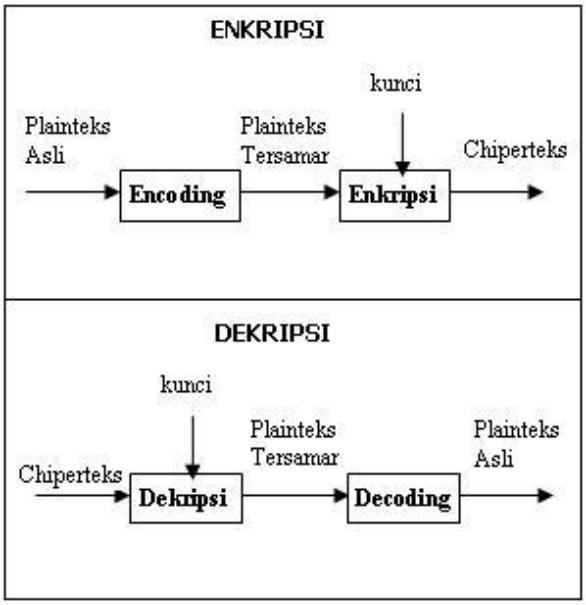


Gambar 2 Encoding dan Decoding pada Steganografi

3. HASIL DAN PEMBAHASAN

Chiperteks yang dihasilkan dengan menggunakan algoritma Vigenere Chiper mungkin memiliki pengulangan kata yang berkorespondensi dengan kata yang sama pada plainteks. Dengan metode Kasiski panjang kunci dapat diketahui dengan mudah apabila teks cukup panjang dan panjang kunci cukup pendek. Pada plainteks dalam bahasa inggris misalnya, terdapat beberapa pengulangan 2 hingga 3 huruf yang sering muncul seperti THE, TH, IS, OF, TO, dan sebagainya. Akibatnya chiperteks yang dihasilkan mungkin berulang.

Vigenere Chiper memiliki kelemahan dalam hal perulangan chiperteks, yang mungkin akan menghasilkan plainteks yang sama. Di sisi lain, konsep Steganografi adalah menyamarkan pesan sehingga membuat pesan asli tidak terbaca. Berdasarkan kedua hal tersebut maka dilakukan kombinasi antara algoritma Vigenere Chiper dengan konsep Steganografi (StegaVig). Apabila digambarkan algoritma kombinasi ini akan menjadi seperti berikut:



Gambar 3 Enkripsi dan Dekripsi pada StegaVig

Keterangan gambar:
Plainteks Asli : plainteks semula (sebelum enkripsi),
Plainteks Tersamar : plainteks setelah disamarkan,
kunci : kunci simetri untuk proses enkripsi dan dekripsi pada algoritma Vigenere Chiper,
Chiperteks : teks hasil enkripsi.

Pada proses enkripsi, plainteks awal disamarkan terlebih dahulu dengan menggunakan algoritma StegaVig (*encoding*), kemudian hasil dari plainteks yang telah tersamar inilah yang akan dikenakan enkripsi dengan algoritma Vigenere Chiper. Pada proses dekripsi, chiperteks didekripsi terlebih dahulu dengan menggunakan kunci simetri, kemudian hasil dari dekripsi chiperteks ini adalah sebuah plainteks Tersamar. Pada Plainteks Tersamar dilakukan *decoding* sehingga menghasilkan Plainteks Asli.

3.1. Encoding

Encoding pada StegaVig adalah sebagai berikut:
Plainteks dibagi tepat menjadi dua bagian, misalnya B1, dan B2. B1 adalah bagian awal plainteks semula, sedangkan B2 adalah bagian akhir. Kedua bagian ini kemudian digabungkan kembali, dengan saling menyisipkan terlebih dahulu. Sebagai gambaran, plainteks tersamar hasil encoding adalah sebagai berikut:

Indeks plainteks tersamar	1	2	3	4	2n-1	2n
Plainteks Tersamar	B^1_i	B^2_i	B^1_{i+1}	B^2_{i+1}	B^1_n	$B^2_{n,}$

dengan B^X_Y adalah plainteks bagian X pada huruf ke Y, dan n adalah panjang maksimal plainteks yang terbagi

(B1 atau B2).
Proses encoding dilakukan sebanyak putaran tertentu untuk memperkuat penyamaran plainteks.

Berikut ini adalah contoh implementasi encoding pada file StegaVig.java:

```
public String plainSamarPutaran(String plain)
{
    // Deklarasi semua variabel di sini
[DELETED]

    // Lakukan encoding berulang-ulang hingga
    // putaran tertentu [DELETED]

        // Panggil fungsi untuk menyamarkan
        str = getPlainTersamar(str);
   return str;
}

public String getPlainTersamar(String plain)
{
    // Deklarasi semua variabel [DELETED]
    // Bagi plainteks menjadi 2 bagian
[DELETED]

    // Sisipkan keduanya, str2 adalah
    // plainteks hasil sisipan
    return str2;
}
```

3.2. Decoding

Decoding pada StegaVig merupakan proses balikan dari encoding. Berikut ini adalah contoh implementasi decoding pada file StegaVig.java:

```
public String plainAsliPutaran(String
plainteks)
{
    // Deklarasi semua variabel [DELETED]
    // Lakukan decoding berulang-ulang hingga
    // putaran tertentu
        // Panggil fungsi untuk mengembalikan
        // ke plainteks asli
        str = getPlainAsli(str);
    return str;
}

public String getPlainAsli(String plainteks)
{
    // Deklarasi semua variabel [DELETED]
    // Pisahkan sisipan plainteks
    // str adalah plainteks sebelum
    // disisipkan

    return str;
}
```

3.3. Penerapan algoritma StegaVig

Berikut ini adalah contoh implementasi enkripsi dan dekripsi pada file StegaVig.java yang disertai encoding dan decoding:

```
// Proses Enkripsi
public void convertToChiper(String key,
String plain, File file) throws IOException
{
    // Deklarasi semua variabel di sini

    // File untuk menyimpan hasil enkripsi
    FileWriter outputStream = new
FileWriter(file);

    // StrBaru berisi string hasil
    // penghilangan spasi [DELETED]

    // str adalah plainteks tersamar, hasil
    // encoding
    String str = plainSamarPutaran(strBaru);

    try{
        while(k<str.length())
        {
          if(str.charAt(k)!=' '){
        // Enkripsi, LOOK-UP bujursangkar
        // Vigenere[DELETED]

        // Tulis chiperteks ke file
        outputStream.write(chiper);
        }
        k++;
      }
    }catch(EOFException e){}
    finally{
        outputStream.close();
    }

}

// Proses Dekripsi
public void convertToPlain(String key, String
chiper, File file) throws IOException
{
    // Deklasari semua variabel di sini

    // File untuk menyimpan hasil dekripsi
    FileWriter outputStream = new
FileWriter(file);

    try{
        while(j<chiper.length())
        {
        // Dekripsi, LOOK-UP bujursangkar
        // Vigenere [DELETED]
              j++;
        }
        // plain asli hasil decoding
        plainAsli =
        plainAsliPutaran(plainTersamar);

     for(int i=0;i<plainAsli.length();i++){
        // Tulis plainteks ke file
     outputStream.write(plainAsli.charAt(i));
        }
    }catch(EOFException e){}
    finally{
        outputStream.close();
    }
}
```

3.4. Implementasi

Misalnya diberikan plainteks seperti di bawah ini:

INI CONTOH AJA KOK INI AJA

Dengan melakukan penghilangan spasi untuk menghilangkan perkiraan kata, maka dihasilkan plainteks tanpa spasi seperti berikut:

INICONTOHAJAKOKINIAJA

Misalnya akan dilakukan enkripsi dengan menggunakan kunci yang kemungkinan besar dapat memberikan chiperteks yang sama, panjang plainteks adalah kelipatan 3 dengan letak kata berulang yang

mempunyai kelipatan yang sama, karena itu kunci yang akan digunakan adalah MOM.

Enkripsi tanpa StegaVig

Dengan menggunakan algoritma Vigenere Chiper, chiperteks hasil enkripsi adalah seperti berikut:

UBUOCZFCTMXMWCWUBUMXM

Masih terdapat beberapa pengulangan kata seperti UBU dan MXM, yang berkorespondensi dengan INI dan AJA. Dengan menggunakan metode Kasiski panjang kunci dapat ditemukan dengan sangat mudah dengan mencoba kombinasi huruf sebanyak 26^P, dengan p = 2 apabila huruf pertama dan ketiga kunci diasumsikan sama.

Enkripsi dengan StegaVig.

Plainteks disamarkan sebanyak n iterasi, misalnya dalam hal ini n=3, maka pada iterasi pertama plainteks disamarkan menjadi:

IANKIOCKOINNTIOAHJAAJ

Plainteks tersamar hasil terasi ke-2:

INATNIKOIAOHCJKAOAIJN

Plainteks tersamar hasil terasi ke-3:

IHNCAJTKNAIOKAOIIJANO

Chiperteks hasil enkripsi:

UVZOOVFYZMWAWOAUWVMBA

Pada chiperteks di atas tidak ada kelompok huruf chiperteks yang berulang meski kunci sangat lemah.

3.5. Kriptanalisis

Untuk mencoba kekuatan algoritma ini dilakukan serangkaian tes, diantaranya adalah dengan menggunakan file "*astronomi.txt*". Enkripsi dilakukan dengan menggunakan kunci 'LANGIT' dan putaran penyamaran plainteks dilakukan sebanyak 3 kali. Hasil enkripsi disimpan pada file "*enkripsi astronomi.txt*", sedangkan dekripsi disimpan pada file "*dekripsi astronomi.txt*". Hasil *print screen* proses enkripsi pada Gambar 4 Enkripsi file *astronomi.txt* dan proses dekripsi pada Gambar 5 Dekripsi file *enkripsi astronomi.txt*

Mula-mula gunakan metode Kasiski untuk mencari panjang huruf. Kelompok huruf pada chiperteks yang sering muncul dikelompokkan, karena mungkin berasosiasi dengan plainteks yang sama. Berikut adalah huruf chipeteks yang sering muncul:

Tabel 1 Frekuensi kemunculan kelompok huruf

Level Digit Huruf	Huruf	Frekuensi kemunculan rata-rata
2	EN, IB, KM, PE,.. dst	6
3	UML,FUM,WGP, .. dst	2
4	EGQK, MCTR, LTQO	2
5	(tidak ada)	0

Gambar 4 Enkripsi file *astronomi.txt*

Gambar 5 Dekripsi file *enkripsi astronomi.txt*

Huruf chiperteks berulang yang kemungkinan besar bersesuaian dengan kelompok huruf plainteks berulang yang sama yaitu kelompok huruf chiperteks

yang cukup panjang, dalam hal ini ambil kelompok huruf chiperteks terpanjang, yaitu 4 huruf. Dengan menggunakan metode Kasiski, temukan faktor pembagi terbesar untuk menemukan panjang kunci. Faktor pembagi terbesar dari jarak huruf EGQK, MCTR, dan LTQO adalah 2. Untuk 3 digit huruf yang berulang, faktor pembagi terbesar adalah 1. Untuk 2 digit huruf yang berulang, faktor pembagi terbesar adalah 1. Panjang kunci 2 sangat lemah dan chiperteks percobaan dekripsi dengan panjang kunci 2 tidak memberikan plainteks yang diinginkan. Panjang kunci 1 tidak mungkin, karena kemunculan kelompok huruf chiperteks berulang yang cukup panjang seharusnya banyak, misalnya BANDUNG, VANALBADA, dan sebagainya. Dengan demikian kriptanalisis terhadap chiperteks dengan plainteks yang telah disamarkan sebelumnya cukup sulit, padahal jumlah putaran penyamaran plainteks yang digunakan hanya sebanyak tiga kali.

4. KESIMPULAN

Penggunaan teknik penyamaran plainteks pada Vigenere Chiper dapat mengurangi peluang kemunculan kelompok huruf chiperteks yang berulang karena plainteks seolah-olah acak dan menjadi tidak bermakna. Dengan demikian kemungkinan keberhasilan metode Kasiski untuk menemukan panjang huruf yang tepat semakin kecil karena berkurangnya pengulangan chiperteks.

DAFTAR REFERENSI

[1] Munir. Rinadi, *Diktat Kuliah IF5054 Kriptografi*, Institut Teknologi Bandung, 2006.

[2] Menezes, Alfred J., Paul C van Oorschot, dan Scott A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.