Tugas Pengganti UTS

Cryptanalysis

Nama : Ramadhan Kalih Sewu

NPM: 18068148826

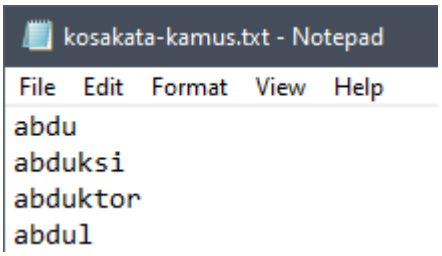
1. **Multiplicative Decipher**

   Karena multiplicative cipher memperlakukan tiap karakter dalam teks secara sama, maka ini mudah sekali diprediksi dengan melihat kesamaan huruf. Misalnya disini kita ambil sebuah kata NKN, dalam bahasa Indonesia sangat mungkin untuk membentuk kata yang terdiri dari pengulangan huruf seperti layaknya N pada decipher. Oleh karena itu kita dapat menebak dengan kata {ada, ini, non, apa} dll. Dari situ kita dapat menentukan nilai kunci dengan mudah sehingga dapat membentuk kata yang diharapkan. Kemudian kunci tersebut digunakan untuk men-decipher seluruh teks dan melihat hasilnya apakah masuk akal atau tidak. Dalam hal ini kita dapat menemukan kunci dari kata ‘non’.

   Dalam program, saya menggunakan approach berbeda:

   Karena kunci cipher ini cuman satu dan berupa angka, ini dapat mudah ditebak dengan cara coba-coba. Berikut langkah yang saya lakukan:

   - Saya mengunduh sebuah teks berisi seluruh kosa kata dalam Bahasa Indonesia.

   - Kemudian kita ambil satu kata dari sebuah teks cipher, dalam hal ini saya ambil kata pertama. Ini dilakukan untuk mempercepat waktu komputasi.
   - Selanjutnya melakukan iterasi untuk mencoba key. Gunakan key tersebut untuk memanggil fungsi decipher, hasil decipher akan dicek apakah terdapat dalam kamus kosa kata.
   - Apabila tidak ada dalam kamus, lakukan seperti langkah sebelumnya sampai iterasi yang ditentukan atau sampai menemukan kata yang terdapat di kamus.

- Dalam percobaan yang saya lakukan, terdapat kecocokan kata dengan kamus pada iterasi ke 20. Namun setelah di cek melakukan decipher dengan key 20 kepada keseluruhan teks, hasilnya tidak masuk akal.
- Lanjutkan lagi mulai dari iterasi 21. Ternyata fungsi menemukan kecocokan pada iterasi ke 23. Saat dicek dengan decipher pada teks secara keseluruhan, hasilnya membentuk sebuah kalimat Bahasa Indonesia yang masuk akal.

Key: 23

Teks:

cetak biru pengembangan sistem logistik nasional sebagaimana dimaksud dalam pasal tersebut berfungsi sebagai acuan bagi menteri pimpinan lembaga non kementerian gubernur dan bupati serta walikota dalam rangka penyusunan kebijakan dan rencana kerja yang terkait pengembangan sistem logistik nasional di bidang tugas masing masing yang dituangkan dalam dokumen rencana strategis masing masing kementerian atau lembaga pemerintah non kementerian dan pemerintah daerah sebagai bagian dari dokumen perencanaan pembangunan

2. Affine Decipher

Cara Brute-Force ini sama persis seperti yang saya lakukan pada Multiplicative Cipher, yaitu menggunakan sebuah kamus untuk melakukan cross-check. Karena pada fungsi ini kita menggunakan dua kunci, maka iterasi dibentuk dengan nested loop. Kompleksitas waktu dari fungsi yang saya buat tentunya membentuk $O(N^2)$ dengan N adalah banyaknya iterasi yang dicoba.

Key: 19, 20

Teks:

bahwa dalam rangka mendorong penggunaan informasi geospasial guna pelaksanaan pembangunan nasional dan untuk mendukung terwujudnya agenda prioritas nawacita diperlukan kebijakan satu peta yang mengacu pada satu referensi geospasial satu standar satu basis data dan satu geoportal bahwa berdasarkan pertimbangan sebagaimana dimaksud pada tersebut perlu menetapkan peraturan presiden tentang percepatan pelaksanaan kebijakan satu peta pada tingkat ketelitian peta skala tertentu kebijakan satu peta yang selanjutnya disebut ksp adalah arahan strategis dalam terpenuhinya satu peta yang mengacu pada satu referensi geospasial satu standar satu

basis data dan satu geoportal pada tingkat ketelitian peta skala dimaksud peta adalah suatu gambaran dari unsurunsur alam atau buatan manusia yang berada di atas maupun di bawah permukaan bumi yang digambarkan pada suatu bidang datar dengan skala tertentu geospasial atau ruang kebumian adalah aspek keruangan yang menunjukkan lokasi letak dan posisi

3. Hill Decipher

Pada operasi cipher, kita melakukan perkalian matriks antara plain teks dengan kunci untuk membentuk sebuah cipher teks.

$$[C] = [P][K]$$

Karena kita mengetahui nilai kedua teks tersebut, kita dapat mencari nilai key dengan operasi aljabar sederahana. Kita pindah matriks plain untuk dikalikan dengan matriks cipher untuk mendapat key. Karena ia pindah ruas, maka pada matriks plain teks akan dicari modular multiplicative inverse nya terlebih dahulu.

$$[T]^{-1}[C] = [K]$$

Selain itu kita tahu bahwa key memiliki matriks dengan ukuran 3x3. Oleh karena itu kita bentuk sebuah matriks plain teks dan cipher teks dengan ukuran yang sama. Untuk mencapai ini, saya mengambil 9 karakter pertama dari kedua teks tersebut.

Key:

```
key =

     3     7    10
     5    11    21
     4     9    17
```

Teks:

bahwanegarakesatuanrepublikindonesiaadalahsebuahnegarakepulauanyangbercirinusantaradengannsegalakekayaansumberdayaalamdansumberdayalainnyasebagaianugerahtuhanyangmahaesayangharusdikeloladenganbaikdanpenuhrasatanggungjawabuntukmenjadisumberkemakmuranbagiseluruhrakyatindonesiabaikdimasakinimaupundimasamendatangbahwadalammengelolasumberdayaalamdansumberdayalainnyasertapenanggulanganbencanadalamwilayahnegarakesatuanrepublikindonesia

4. Vigenere Decipher

Dalam memecah kunci cipher ini, kita menggunakan sebuah metode cryptanalysis menggunakan analisa frekuensi munculnya huruf dalam cipher teks. Dengan teknik ini

terdapat permasalahan yang masih sulit untuk dipecahkan, yaitu menentukan panjangnya teks kunci. Dengan bantuan metode Kasiski, kita dapat menentukan panjang key dengan lebih mudah.

Mari kita analisis algoritma Viginere:

1. **Pengulangan cipherteks**

   Ini menjadi kelemahan dan kelebihan tersendiri bagi algoritma ini, terutama pada cipher polyalphabetic. Mari lihat test case dibawah:

```
keyword: water
keystream:  w a t e r w a t e r w a t e r w a t e r w
plaintext:  o n a p l a n e t h e p l a n e i s d u e
ciphertext: K N T T C W N X X Y A P E E E A I L H L A
```

   kata pengulangan plane menghasilkan sebuah teks cipher yang berbeda yaitu TCWNX dan PEEEA. Tetapi, lihatlah contoh dibawah:

```
keyword: milk
keystream:  m i l k m i l k m i l k m i l k m i l k m
plaintext:  o n a p l a n e t h e p l a n e i s d u e
ciphertext: A V L Z X I Y O F P D Z X I Y O U A O E Q
```

   Dengan key yang berbeda, kita dapat melihat sebuah seri yang berulang yaitu ZXIYO. Ini dapat terjadi juga untuk key lainnya.

2. **Pecahan yang muncul berulang kali**

   Apabila kita memiliki cipher teks dengan pecahan yang cukup panjang, ini merupakan sebuah tanda bahwa sangat mungkin untuk memiliki key dengan faktor jarak antara pecahan cipherteks yang terulang tersebut.

Probable Key Lengths

Key Length	Distances Matched
4	93.65 %
62	3.17 %
97	1.59 %
183	1.59 %
245	1.59 %

Dari perhitungan berdasarkan situs https://planetcalc.com/8550/ probabilitas key dengan panjang 4 memiliki nilai yang tinggi. Ini dilihat dari jarak pengulangan kata.

SIT	298	4
ITA	299	4

Terdapat dua seri kata yang berulang diambil dari potongan 'KWSITASITALV'. Ini juga dapat kita anggap dengan pengulangan kata 'SITA' dengan jarak 4.

3. **Pengelompokan huruf**

Mari kita coba melakukan pengelompokan huruf dengan jumlah anggota sebanyak probabilitas yang kita dapatkan pada analisa sebelumnya. Mari kita awali dengan pengelompokan 4 huruf (karena ini probabilitas terbesar). Contoh:

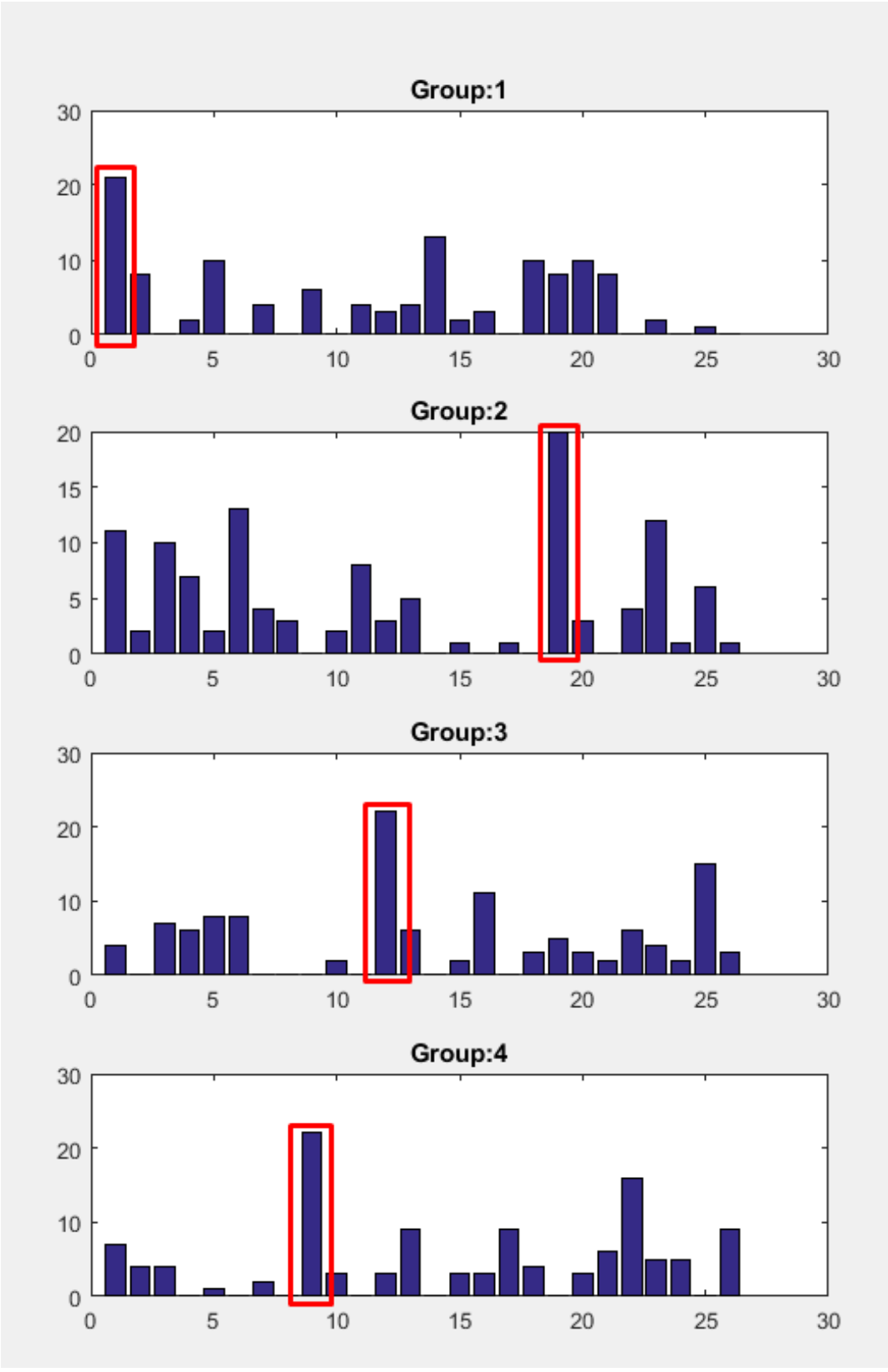
```
keyword letter: 12341 23412 34123 41234 12341
     plaintext: JAKXQ SWECW MMJBK TQMCM LWCXJ
```

Saya menggunakan Matlab untuk melihat plot bar dan menganalisa nilai yang akan diambil. Kemudian saya menyoba memutuskan untuk mengambil nilai dengan frekuensi terbanyak, nilai tengah, atau dari chi-squared scoring. Pertama saya mencoba dengan frekuensi terbanyak pada tiap grup. Hal tersebut membentuk sebuah kata 'ASLI'. Setelah saya cek kunci tersebut, ternyata membentuk sebuah kalimat Bahasa Indonesia yang masuk akal:

Key: ASLI

Teks:

bahwa sistem jaminan sosial nasional merupakan program negara yang bertujuan memberikan kepastian perlindungan dan kesejahteraan sosial bagi seluruh rakyat bahwa untuk mewujudkan tujuan sistem jaminan sosial nasional perlu dibentuk badan penyelenggara yang berbentuk badan hukum berdasarkan prinsip kegotongroyongan nirlaba keterbukaan kehatihatian akuntabilitas portabilitas kepesertaan bersifat wajib dana amanat dan hasil pengelolaan dana jaminan sosial seluruhnya untuk pengembangan program dan untuk sebesarbesar kepentingan peserta


Group:1
Group:2
Group:3
Group:4

Lampiran

Source Code

```
function key = DigestMultiplicative(cipher, dicts, start, stop)
    firstword = strtok(cipher);
    for i = start:stop
        res = DecipherMultiplicative(firstword, i);
        idx = find(ismember(dicts, res));
        if (~isempty(idx))
            key = i;
            return;
        end
    end
end
```

```
function [key1,key2] = DigestAffine(cipher, dicts, start, stop)
    key1 = [];
    key2 = [];
    firstword = strtok(cipher);
    for i = start:stop
        for j = start:stop
            res = DecipherAffine(firstword, i, j);
            idx = find(ismember(dicts, res));
            if (~isempty(idx))
                key1 = i;
                key2 = j;
                return;
            end
        end
    end
end
```

```
function key = DigestHill(cipher, expected, keyRowSize)
    cipher = cipher(1:keyRowSize^2);
    cipher = upper(cipher)-65;
    expected = expected(1:keyRowSize^2);
    expected = upper(expected)-65;
    C = reshape(cipher,[],keyRowSize)';
    T = reshape(expected,[],keyRowSize)';
    n = 26;
    m = round(mod(det(T),n));
    [d,s,t] = gcd(n,m);
    if d == 1
        Tinverse = round(t*det(T)*inv(T));
        Tinverse = mod(Tinverse,n);
    else
        Tinverse = NaN;
    end
    key = mod(Tinverse*C,26);
end
```

```
function PlotGrouping(text, size)
    text = text(regexp(text, '[a-zA-Z]'));
    text = upper(text) - 65;
    count = zeros(size,26);
    g = 1;
    for i = 1:length(text)
        c = text(i);
        count(g,c+1) = count(g,c+1) + 1;
        if (g == size) g = 0; end;
        g = g + 1;
    end
    for i = 1:size
        subplot(size,1,i);
        y = count(i,:);
        bar(y);
        title(sprintf('Group:%d', i));
    end
end
```