Homework 2

Suyi Liu

February 24, 2017

1 P1

- (a)
  Eve can get $c1 = DES_K(m1) + L$, $m1$, $c2 = DES_K(m2) + L$ and $m2$ at the same time.
  Using these information, Eve can compute
  $c1 + c2 = DES_K(m1) + L + DES_K(m2) + L$
  $= DES_K(m1) + DES_K(m2)$ since base n is 2.
  In this way, Eve can permute 2^{64} DES keys(each time use $m1$ and $m2$ to compute $DES_K(m1)+DES_K(m2)$ and see if the sum equals corresponding $c1 + c2$) to find the corresponding K.
  As long as eve get the K, she can compute L immediately by using formula:
  $c1 - DES_K(m1) = L$

- (b)
  Eve can get $c1 = DES_K(m1 + L)$, $m1$, $c2 = DES_K(m2 + L)$ and $m2$ at the same time.
  Since $DES_{\hat{K}}(c1) = m1 + L$, $DES_{\hat{K}}(c2) = m2 + L$,
  $m1 + L + m2 + L = DES_{\hat{K}}(c1) + DES_{\hat{K}}(c2) = m1 + m2$
  These functions are in the same form as what is given in part (a), except that K is replaced by $\hat{K}$ and $m1$, $m2$ swapped with $c1$, $c2$.
  So we can use exactly the same strategy(permute 2^{64} $\hat{K}$, and use known $c1$ and $c2$ to compute $DES_{\hat{K}}(c1) + DES_{\hat{K}}(c2)$ and see if the sum equals corresponding $m1+m2$) described in part (a) to get $\hat{K}$, and L respectively.

2 P2

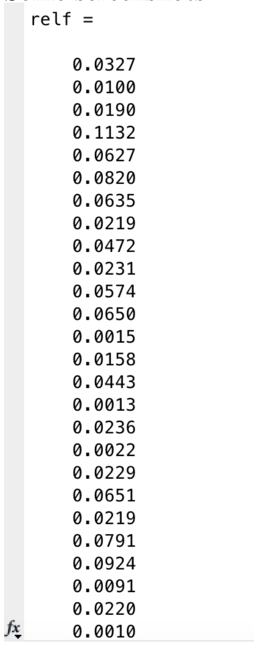
I selected the invertible matrix m bu shifting one "position" each time. compute $A = CM^{-1}$, and $M = A^{-1}C$
See crackhill.m and the detailed comments in it.

3 P3

I first get the relative frequencies of each letter appeared in the ciphertext and sorted them(shown in first screenshot). I then got the relatice frequencies of English letters and sorted them. Then I get the mapping of four most frequent letters that are accurate:
d - e
w - t
f - a
v - o
Then I get the rest of the mappings as my guess:
t - i
l - n
g - s
e - h
k - r
i - d
o - l
a - u
q - c
j - m
s - w
y - f
u - y
h - g
c - p
n - b
b - v
x - k
r - j
m - z
p - x
z - q
Then I used the bigram hint:
Since wg, and gd are among the top 2 most frequent bigram, and they match th, he well, so we can assume g - h here.
Since we are sure that d - e, so the fourth frequent bigram is messed up. So this further confirms our assumption of g - h pair since no more than two pairs are messed up. Also the 5th bigram is messed up since we already know f - a. So the third bigram is correct, which means l - i, t - n. Then we substitute using new evidences.
Then I looked into the trigrams, we are sure the wgd - the pair is correct, we assume tha is among the other two, but it is impossible since t and a doesn't find any match in those ciphertexts using our clue. So 'and' is among those two. And using f - a pair, fti means and. So we get i - d. We substitute the ciphertext again.

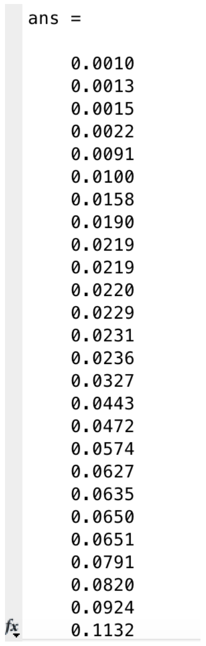
Since a letter cannot be misordered by more than one, e - s is for sure, otherwise k will be misordered by 2 indices. And hence k - r for by the same logic.
I noticed ”theho*sestood” as a phrase. So ho*se is a noun, and since r is already assigned to k, the * is probably u, and we try a - u pair. Thus o - l pair should be valid according to at most 1 misordering rule. By substituting again, I noticed words such as ”thursda*”, thou*ht, so j - y, y - g. And words like ”loo*edat”, ”the*illage”, so b - k, x - v.
For the rest, ”theedgeo*thevillage”, ”stoodonitso*n”, ”a**ro*riate”, ”*ountry”, ”lookedovera*road”, ”depart*ent”, ”e*actly”, *uiteatease”, ”bulldo*er”, s - f, q - w, h - c, n - b, u - m, r - x, z - q, m - z, and the last one must be p - j.
Eventually, the text makes sense as I substitute using command:
substitute(’ukpesahcdyrizbljwxfnmotvgq’,ciphertext).

So the mapping function f is : abcdefghijklmnopqrstuvwxyz(char i in plaintext) to fnhidsyglpboutvczkewaxqrjm(char as f(i), $1 \le i \le 26$)
Some screenshots:



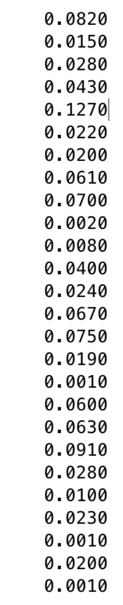
```
relf =

    0.0327
    0.0100
    0.0190
    0.1132
    0.0627
    0.0820
    0.0635
    0.0219
    0.0472
    0.0231
    0.0574
    0.0650
    0.0015
    0.0158
    0.0443
    0.0013
    0.0236
    0.0022
    0.0229
    0.0651
    0.0219
    0.0791
    0.0924
    0.0091
    0.0220
    0.0010
```

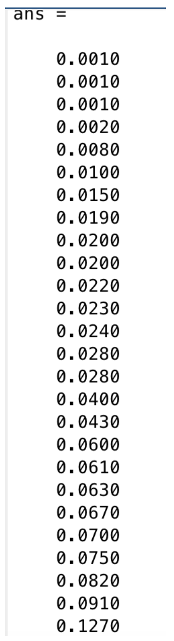


```
ans =

    0.0010
    0.0013
    0.0015
    0.0022
    0.0091
    0.0100
    0.0158
    0.0190
    0.0219
    0.0219
    0.0220
    0.0229
    0.0231
    0.0236
    0.0327
    0.0443
    0.0472
    0.0574
    0.0627
    0.0635
    0.0650
    0.0651
    0.0791
    0.0820
    0.0924
    0.1132
```

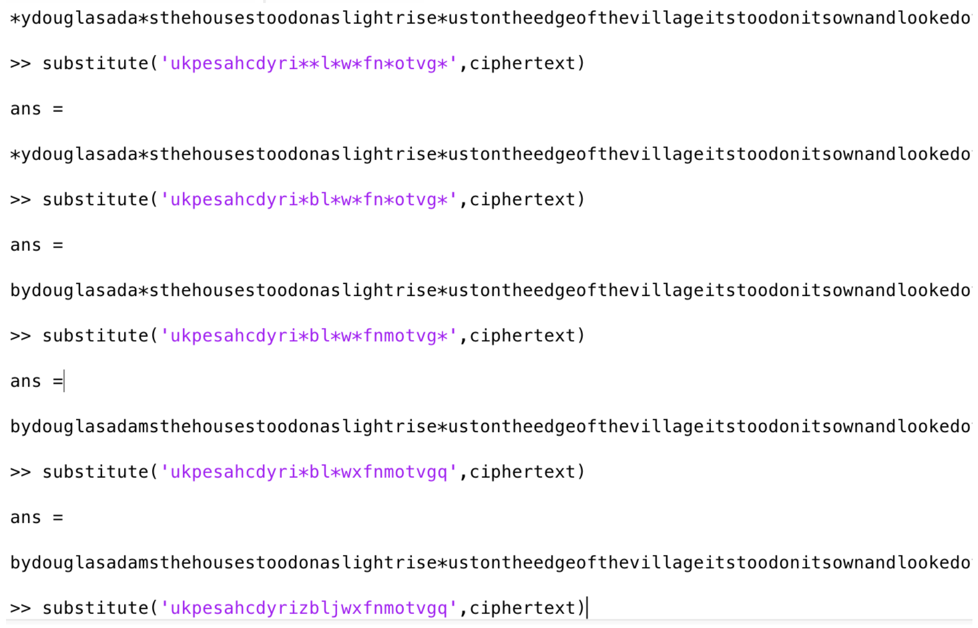


```
    0.0820
    0.0150
    0.0280
    0.0430
    0.1270
    0.0220
    0.0200
    0.0610
    0.0700
    0.0020
    0.0080
    0.0400
    0.0240
    0.0670
    0.0750
    0.0190
    0.0010
    0.0600
    0.0630
    0.0910
    0.0280
    0.0100
    0.0230
    0.0010
    0.0200
    0.0010
```



```
ans =

    0.0010
    0.0010
    0.0010
    0.0020
    0.0080
    0.0100
    0.0150
    0.0190
    0.0200
    0.0200
    0.0220
    0.0230
    0.0240
    0.0280
    0.0280
    0.0400
    0.0430
    0.0600
    0.0610
    0.0630
    0.0670
    0.0700
    0.0750
    0.0820
    0.0910
    0.1270
```



```
*ydouglasada*sthehousestoodonaslightrise*ustontheedgeofthevillageitstoodonitsownandlookedo

>> substitute('ukpesahcdyri**l*w*fn*otvg*',ciphertext)

ans =

*ydouglasada*sthehousestoodonaslightrise*ustontheedgeofthevillageitstoodonitsownandlookedo

>> substitute('ukpesahcdyri*bl*w*fn*otvg*',ciphertext)

ans =

bydouglasada*sthehousestoodonaslightrise*ustontheedgeofthevillageitstoodonitsownandlookedo

>> substitute('ukpesahcdyri*bl*w*fnmotvg*',ciphertext)

ans =

bydouglasadamsthehousestoodonaslightrise*ustontheedgeofthevillageitstoodonitsownandlookedo

>> substitute('ukpesahcdyri*bl*wxfnmotvgq',ciphertext)

ans =

bydouglasadamsthehousestoodonaslightrise*ustontheedgeofthevillageitstoodonitsownandlookedo

>> substitute('ukpesahcdyrizbljwxfnmotvgq',ciphertext)
```

4 Extra Credit

Plaintext is The Constitution of the United States The Bill of Rights All Amendments