CS641A: Modern Cryptology Assignment 2

objectStrongly: Parv Jain (170464)
Nikhil Naidu(170758)
Dewansh Singh (170241)

January 23^{rd}, 2020

1 Solution to Chapter 2: The Caveman

- We entered the command "go" to go to the next page.
- The page asked us to count the horizontal lines in the face. It came out to be {1, 2, 2, 5, 3, 2, 6, 2, 10}. however it even said to bow and look up , which possibly means that the order is reversed {10,2,6,2,3,5,2,2,1}
- We entered the command "go" again which lead us to the first page and then we entered the command "read" to read the pattern on the boulder.
- Here we got presented with a cipher text -
"Lg ccud qh urg tgay ejbwdkt, wmgtf su bgud nkudnk lrd vjfbg. Yrhfm qvd vng sfuuxytj "vkj_ecwo_ogp_ej_rnfkukf" wt iq urtuwjm. Ocz iqa jdag vio uzthsivi pqx vkj pgyd encpggt. Uy hopg yjg fhkz arz hkscv ckoq pgfn vu wwygt nkioe zttft djkth"
- We perform a frequency analysis(see Table 1) on the cipher text and found that all the letters appear several times , and the frequency does not vary much. This suggests a poly alphabetic substitution cipher .
- We are using the first crypt analysis technique of index of coincidence. Which came out to be 0.04236 . (Notice Index of coincidence is low (close to 0.0385)) which again suggested that the text might have been encrypted using a poly alphabetic Vigenère Cipher.
- Now we perform the "Kasiski examination" on the cipher text . We perform shifts in the entire cipher text one by one starting from 1 and to 12 shifts and notice that the max no of coincidence occured at 9 shifts, indicating the key lenght to be 9.

Shift	1	2	3	4	5	6	7	8	9	10	11	12
Coincidence	5	4	12	19	14	12	8	16	20	11	11	16

- The nine numbers on the face {10,2,6,2,3,5,2,2,1} correspondingly matching to the letters {j,b,f,b,c,e,b,b,a} hints that the 9 length key might be "jbfbcebba" .We used this key to solve the Vigenère cipher and got the result
  "Cf xbsz pg uif ofyu dibncfs, uifsf jt wfsz mjuumf kpz uifsf. Tqfbl pvu uif qbttxpse "uif_dbwf_nbo_cf_qmfbtfe" up hp uispvhi. Nbz zpv ibwf uif tusfohui gps uif ofyu dibncfs. Up gjoe uif fyju zpv gjstu xjmm offe up vuufs nbhjd xpset uifsf"

- We Further examined this above description which even turned out to be simply a Caesar cipher with shift index 1. Or we can say that the key itself was ciphered using Caesar cipher with index 1 , thus making our key for Vigenère cipher "kcgcdfccb". Now we may use the Vigenère square or Vigenère Decryption to finally get the ciphered text decoded into -
  "Be wary of the next chamber, there is very little joy there. Speak out the password "the_cave_man_be_pleased" to go through. May you have the strength for the next chamber. To find the exit you first will need to utter magic words there"

- Hence we got our password to the next level which was "the_cave_man_be_pleased" (without the quotes)

Table 1: Frequency Analysis

Alphabet	Frequency	Alphabet	Frequency
a	4	n	7
b	3	o	7
c	7	p	6
d	9	q	6
e	5	r	6
f	9	s	4
g	16	t	13
h	7	u	13
i	6	v	9
j	10	w	7
k	12	x	2
l	2	y	7
m	3	z	5

2 Code Repository

2.1 Vigenère Decryption Algorithm

```
Decrypt(String S,string k){
j=0
for(i = 0 to len(S))
    {
        if(S[i]>='A' && S[i]<='Z' || S[i]>='a' && S[i]<='z')
        S[i] = S[i] - k[j]%26
        j=(j+1)%len(k)
    }
return S
}
```

2.2 Caesar Cipher Decryption Algorithm

```
Decrypt(String S ,int index){
for( i= 0 to len(S) )
  {
    if(S[i] >= 'a' && S[i] <= 'z'){
      S[i] = S[i] - index
      if(S[i] < 'a'){
        S[i] = S[i] + 'z' - 'a' + 1;
      }
      }
    if(S[i] >= 'A' && S[i] <= 'Z'){
      S[i] = S[i] - index
      if(S[i] < 'A'){
        S[i] = S[i] + 'Z' - 'A' + 1;
      }
      }
  }
}
```