Q1 Team Name
0 Points

Noobs

Q2 Commands
10 Points

List the commands used in the game to reach the ciphertext.

The commands we entered were:
1. go
2. back
3. read

Q3 CryptoSystem
10 Points

What cryptosystem was used in this level?

Playfair cipher was used in this level to decrypt the code and the key was "CRYPTANALYSIS". The key was decoded using morse code.

Q4 Analysis
20 Points

What tools and observations were used to figure out the cryptosystem? (Explain in less than 300 words)

First we just used the command read directly and got the cipher text. We analyzed the frequency count of all letters in the cipher text. But, there was no such uneven distribution observed as in Substitution cipher for English. We also observed that there are no numbers in the cipher text. So, the chances of it being a substitution or a caesar cipher were very low. We restarted the game and entered the command go. We got a morse code and a message from spirit that mentioned "PLAY FAIR". So, we searched for it and found out the cipher text to be playfair cipher. All letters of cipher text were in upper case too. Playfair cipher does not supports numeric characters and contains code all in either upper cases or lower cases. We figured out that such properties are posessed by our cipher. So, most probably digraphs substitution must be there and it must be a playfair cipher.
So, we tried to figure out the key as "CRYPTANALYSIS". We found code written using "." and "-" which was basically the word "CRYPTANALYSIS" written using morse code. We decoded this morse code with help from morse table and got the decrypted answer as our key.
After figuring out the cryptosystem, we did the following steps:
1. We built a 5*5 matrix and entered the key along with other alphabets. We got the following key matrix:
[[C,R,Y,P,T],
[A,N,L,S,I/J],
[B,D,E,F,G],
[H,K,M,O,Q],
[U,V,W,X,Z]]
2. Then we carried out our deciphering process. We removed all the commas, full stops, under scores, spaces etc. from cipher text as playfair cipher doesn't supports it.
3. We iterated the cipher text then pairwise forming digraphs. To decrypt each digraph we took help of the key table.
4. If both the alphabets lied on same row of the key table, then we replace each letter of digraph with their immediately left letter in the key table. Eg. DF : -> BE
5. If both alphabets lied on same column of the key table, then we replace each letter of the digraph with their immediately upper letter in the key table. Eg. XO: -> OF
6. If both alphabets lied on different rows and different column, then we'll form a rectangle with the 2 letters as diagonally opposite elements. Then for each letter, replace it with their diagonally opposite element(the same row) in that rectangle of key table. Eg. UL : -> WA
We carried out above process for all digraphs in the cipher text. The final decrypted text that we got contained only digraph words. We formed the correct words by using common sense and replaced characters X at certain places as it was jut acting as a filler element for some words to make them digraph.

Q5 Decryption Algorithm
15 Points

Briefly describe the decryption algorithm used. Also mention the plaintext you deciphered. (Use less than 350 words)

The decryption algorithm that we used here is described as follows:
1. We ask the user to input the key for the playfair cipher and store it as a variable named "key". We entered the key as "CRYPTANALYSIS".
2. We remove spaces(if any) from the key and convert it all to upper case(If key entered by user is a mix of capital and small letters).
3. We then create a 5*5 matrix using a list data structure and name the list as "result". All it's values are initialized to 0.
4. We start storing key in the result list. We iterate the key letter by letter using for loop. If the letter is not in the list, we add it. If the letter is 'I', then add 'J' too along with 'I'. It's so because there are 25 cells and 26 alphabets. So, to fit all, I and J are written together by convention.
5. After adding only unique letters of key, we add remaining left over letters in the matrix. We use a for loop and iterate between ASCII values 65('A') to 90('Z'). We'll add the letter in the list only if it's not present earlier. We'll handle the case of I/J also and make them store it together.
6. We ask the user then to input the cipher text and we store it in a variable named "msg". We convert msg to upper case. We remove blank spaces, commas, full stops, double quotes and under scores from the msg. All the given tasks are done because Playfair cipher only works for English alphabets only.
Our key matrix will look as below:
[[C,R,Y,P,T],
[A,N,L,S,I/J],
[B,D,E,F,G],
[H,K,M,O,Q],
[U,V,W,X,Z]]
7. Then we iterate the cipher text using a while loop. We increment the counter by 2 so that we can only work with digraphs.
8. We create 2 empty lists named "loc" and "loc1". loc contains position of 1st letter in the matrix key and loc1 contains position of 2nd letter in the matrix key. We get the row number(loc[1]) and column number(loc[0]) of that letter in the key matrix as it's location.
9. If both letters are in the same row(loc[1]==loc1[1]), go 1 place left in the matrix. For corner cases we do modulo 5 to get round about. Eg. DF : -> BE
10. If both letters are in the same column(loc[0]==loc1[0]), go 1 place up in the matrix. For corner cases we do modulo 5 to get round about. Eg. XO: -> OF
11. If both letters are in different rows and columns, go to diagonally opposite pair of the matrix(row order must be maintained i.e., upper row's diagonal must be replaced by upper row's opposite diagonal element only). Eg. UL : -> WA
We get the final output using this algorithm as:
BE WA RY OF TH EN EX TC HA MB ER TH ER EI SV ER YL IT TL EI OY TH ER ES PE AK OU TX TH EP AS SW OR DA BR AC AD AB RA TO GO TH RO UG HM AY YO UH AV ET HE ST RE NG TH FO RT HE NE XT CH AM BE RT OF IN DT HE EX IT YO UF IR ST WI LX LN EX ED TO UT TE RM AG IC WO RD ST HE RE
Applying common sense, we try figuring out the actual message. At certain places X was added as a filler element just to make the word a digraph or as a seperator between 2 consecutive letters. So, we handle those cases too. There is a word IOY, that doesn't makes sense. So, we replaced I with J(as I/J are together in key table) to make it JOY. After manually figuring out, we come up with the following output finally:
BEWARY OF THE NEXT CHAMBER THERE IS VERY LITTLE JOY THERE SPEAK OUT THE PASSWORD ABRACADABRA 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
We tried with both the passwords ABRA_CA_DABRA and ABRACADABRA. Both of them worked. But, by observing the cipher text given in double scores and underscores involved, we preferred ABRA_CA_DABRA.

Q6 Password
10 Points

What was the final command used to clear this level?

The final command used to clear this level was either of ABRA_CA_DABRA or ARACADABRA. We cleared it using ABRA_CA_DABRA command.

Q7 Code
0 Points

Upload any code that you have used to solve this level

ModCryptA2.ipynb Download

In [ ]:

```
key=input("Please enter the key for playfair cipher:")
key=key.replace(" ", "") #Removing any spaces if any in the key
key=key.upper() #Converting all characters of key to the same case, i.e., upper case
def matrix(x,y,initial): #Defined a matrix with dimensions x*y and containing all values='initial'
    return [[initial for i in range(x)] for j in range(y)]

result=list() #Creating a list that stores the results.

for c in key: #storing key in the results list
    if c not in result:
        if c=='J':
            result.append('I') #I and J are addedtogether as there are only 25 cells and the no. of alphabets are 26.
        else:
            result.append(c)
flag=0
for i in range(65,91): #storing other characters on the basis of ASCII values
    if chr(i) not in result:
        if i==73 and chr(74) not in result: #73->I and 74->J in ASCII values
            result.append("I")
            flag=1
        elif flag==0 and i==73 or i==74:
            pass
        else:
            result.append(chr(i))
k=0
my_matrix=matrix(5,5,0) #initialize matrix with all 0s
for i in range(0,5): #making matrix
    for j in range(0,5):
        my_matrix[i][j]=result[k]
        k+=1

def locindex(c): #get location of each character
    loc=list()
    if c=='J':
        c='I'
    for i,j in enumerate(my_matrix):
        for k,l in enumerate(j):
            if c==l:
                loc.append(i)
                loc.append(k)
                return loc


msg=str(input("ENTER CIPHER TEXT:"))
msg=msg.upper()
msg=msg.replace(" ", "") #Removing blank spaces
msg=msg.replace(",", "") #Removing commas
msg=msg.replace(".", "") #Removing full stops
msg=msg.replace('"', "") #Removing double quotes
msg=msg.replace('_', "") #Removing under scores
#All the above tasks are done because Playfair cipher only works for English alphabets only.
print("The plain text is:",end=' ')
i=0
while i<len(msg): #Iterating the cipher code letter by letter:
    loc=list() #Creating an empty list
    loc=locindex(msg[i]) #Gets location of ith letter in the matrix of Key, and appends it to the list
    loc1=list() #Creating a new empty list
    loc1=locindex(msg[i+1]) #Gets location of (i+1)th letter in the matrix of Key, and appends it to the list
    if loc[1]==loc1[1]: #If both letters of digraph are in same row:
        print("{}{}".format(my_matrix[(loc[0]-1)%5][loc[1]],my_matrix[(loc1[0]-1)%5][loc1[1]]),end=' ') #Go 1 place left in the matrix. For corner cases we do modulo 5 to get round about.
    elif loc[0]==loc1[0]: #If both letters of digraph are in same column:
        print("{}{}".format(my_matrix[loc[0]][(loc[1]-1)%5],my_matrix[loc1[0]][(loc1[1]-1)%5]),end=' ')  #Go 1 place up in the matrix. For corner cases we do modulo 5 to get round about
    else: #If both letters of digraph are in different rows: and columns:
        print("{}{}".format(my_matrix[loc[0]][loc1[1]],my_matrix[loc1[0]][loc[1]]),end=' ')  #Go to leftmost of the matrix for diagonal elements.
    i=i+2          # Increment by 2 because we are working with digraphs.
```

```
Please enter the key for playfair cipher:CRYPTANALYSIS
ENTER CIPHER TEXT:DF ULYP XO CQD LFWC RUBHEDY, CQDYG LN XDYL EGIYIG LMP CQ
The plain text is: BE WA RY OF TH EN EX TC HA MB ER TH ER EI SV ER YL IT TI
```

In [ ]:

Assignment 2

GRADED

GROUP
Manu Shukla
Rishabh Lakhwani
Divyansh Bisht
View or edit group

TOTAL POINTS
65 / 65 pts

QUESTION 1	Team Name	**0** / 0 pts
QUESTION 2	Commands	**10** / 10 pts
QUESTION 3	CryptoSystem	**10** / 10 pts
QUESTION 4	Analysis	**20** / 20 pts
QUESTION 5	Decryption Algorithm	**15** / 15 pts
QUESTION 6	Password	**10** / 10 pts
QUESTION 7	Code	**0** / 0 pts