Q1 Team Name

0 Points

Enciphered

Q2 Commands

10 Points

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

go
back
read

Q3 CryptoSystem

10 Points

What cryptosystem was used in this level?

(i) Playfair Cipher (symmetric encryption technique as well as polyalphabetic substitution cipher in nature that encodes bigrams.)
(ii) International Morse Code (not a cryptosystem but rather a means of text encoding)

Q4 Analysis

20 Points

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

Tools used:
1. International morse code chart (https://morsecode.world/) and python script to
 break the morse code.

2. Python script to decrypt the cipher (Reference:https://github.com/themrshubh/custom-playfair-cipher/)

Observation:

1. At the start of level 2, we used the go command to navigate to a distant boulder and inspect its patterns. We came upon an unusual curved design comprised of dots and dashes. Based on our knowledge, we assumed that it was morse code.

2. We used an international morse code chart and the associated Python script to generate something usable; thankfully, it was just morse code that decoded to the term 'CRYPTANALYSIS.'

3. The spirit, i.e. the chamber keeper, blessed the occupants with the words " you must always trust in yourself and PLAY FAIR". This served as a cue for the subsequent actions. We were previously aware of the existence of a symmetric cryptographic system called " PLAY FAIR," and because the morse code decoded as a word, we reasoned that the Playfair cipher with the key CRYPTANALYSIS would be required at this level.

4.   After accumulating this amount of data, we used the back command, followed by the read command, to reveal the ciphertext.

5. The ciphertext had an even number of letters, and the ciphertext space included a subset of upper case English alphabets, white spaces, and punctuation.

6. In the Playfair cipher, the letters I and J are treated identically in the 5x5 matrix, which is used to insert in the ciphertext according to the encoder's preference. As the ciphertext did not contain any "J," it is possible that the encoder replaced the element I/J with I alone. Furthermore, the "J"-less ciphertext strengthened the case for decryption using the Playfair cipher.

7. When the ciphertext was broken into pairs of two consecutive letters, no pair included the same letter. If there are no double letter bigrams in the ciphertext and the length of the message is long enough to make this statistically significant, it is very likely that the method of encryption is Playfair.

All of the foregoing facts influenced our decision to proceed with the decryption of the ciphertext using the playfair cipher..

Q5 Decryption Algorithm

15 Points

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

Decryption Algorithm

1. The Playfair algorithm is based on the use of a 5 * 5 matrix of letters constructed using keywords. Because the keyword is CRYPTANALYSIS, the matrix will be constructed as follows.

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

The matrix is built by filling in the letters of the keyword (minus duplicates) from left to right and from top to bottom and then filling in the rest of the matrix in alphabetical order with the remaining letters. I and J are counted as one letter.

2. The ciphertext is stripped of special characters and white spaces.

3. We now divide the ciphertext into pairs of two consecutive letters. If the ciphertext has an odd number of alphabets, we will add an extra filling letter, say Z, with the last alphabets. We didn't need that Z in our case because the ciphertext had an even number of alphabets.

4. The following are the decryption rules:

(i) Two plaintext letters that fall in the same row of the matrix are each replaced by the letter to the left, with the last element of the row circularly following the first. For example, B and F of BF lies in the same row, hence got decoded as BE

(ii) Two plaintext letters that fall in the same column are each replaced by the letter above, with the bottom element of the column circularly following the first. For example, L and E of LE lie in the same column, hence got decoded as YL.

(iii) Otherwise, each ciphertext letter in a pair is replaced by the letter that lies in its own row and the column occupied by the other ciphertext letter. For example, UL does not satisfy the above two and got replaced by WA.

5. Continuing with the above description rules (used the attached python script), the decrypted message is as follows:

BE WARY OF THE NEXT CHAMBER, THERE IS VERY LITTLE IOY THERE.

SPEAK OUTX THE PASSWORD "ABRA_CA_DABRA" TO GO
THROUGH. MAY YOU
HAVE THE STRENGTH FOR THE NEXT CHAMBER. TO FIND
THE EXIT YOU
FIRST WILXL NEXED TO UTTER MAGIC WORDS THERE.

6. The decrypted text featured an additional X in the terms OUTX THE, WILXL, and NEXED following decoding with the Playfair Cipher. When we use the play fair cipher to encrypt any message, we employ a filler character, such as X, to separate repeated plaintext letters in the same pair. As a result, we must remove the X to revert to the plaintext's original condition.
7. There was the word "IOY" that had no sense after the decryption. This word is more likely to be "JOY" since "J" is replaced by "I" in the Playfair cipher.
8. Spaces and special characters remain unaltered.
9. After removing the unnecessary X and substituting I and J as the accordingly, the final plaintext is as follows:

BE WARY OF THE NEXT CHAMBER, THERE IS VERY LITTLE
JOY THERE.
SPEAK OUT THE PASSWORD "ABRA_CA_DABRA" 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.

Q6 Password

10 Points

What was the final command used to clear this level?

ABRA_CA_DABRA

Q7 Code

0 Points

Upload any code that you have used to solve this level

▾ code.ipynb Download

Morse Code Conversion

In [1]:

```
## Convert Morse code to english
dict = {'A':'.-', 'B':'-...',
                    'C':'-.-.',
'D':'-..', 'E':'.',
                    'F':'..-.', 'G':'-
-.', 'H':'....',
                    'I':'..', 'J':'.---',
'K':'-.-',
                    'L':'.-..', 'M':'--',
'N':'-.',
                    'O':'---', 'P':'.-
-.', 'Q':'--.-',
                    'R':'.-.', 'S':'...',
'T':'-',
                    'U':'..-', 'V':'...-
', 'W':'.--',
                    'X':'-..-', 'Y':'-.--
', 'Z':'--..',
                    '1':'.----', '2':'..-
--', '3':'...--',
                    '4':'....-',
'5':'.....', '6':'-....',
                    '7':'--...', '8':'--
-..', '9':'----.',
                    '0':'-----', ', ':'-
-..--', '.':'.-.-.-',
                    '?':'..--..',
'/':'-..-.', '-':'-....-',
                    '(':'-.--.', ')':'-.-
-.-'}
new_dict = {v:k for k,v in dict.items()}
morse_code = '-.-. .-. -.-- .--. - .- -.
.- .-.. -.-- ... .. ...'
morse_code = morse_code.split(' ')
english_word = ''
for i in morse_code:
    english_word +=new_dict[i]
english_word
```

Out [1]:

```
'CRYPTANALYSIS'
```

Play fair cipher decryption

In [5]:

```
key = 'CRYPTANALYSIS'
matrix = [['C','R','Y','P','T'],
               ['A','N','L','S','I'],
               ['B','D','E','F','G'],
               ['H','K','M','O','Q'],
               ['U','V','W','X','Z'],]
```

In [25]:

```
def decryption(matrix, ciphertext):
    for i in matrix:
        if ciphertext[0] in i:
            row1 = matrix.index(i)
            col1 = i.index(ciphertext[0])
    for i in matrix:
        if ciphertext[1] in i:

            row2 = matrix.index(i)
            col2 = i.index(ciphertext[1])
    #Case1: Same row
    if row1 == row2:
        if col1 == 0 and col2 != 0:
            ciphertext = matrix[row1]
[4]+matrix[row1][col2-1]
        elif col1 != 0 and col2 == 0:
            ciphertext = matrix[row1]
[col1-1]+matrix[row1][4]
        elif col1 == 0 and col2 == 0:
            ciphertext = matrix[row1]
[4]+matrix[row1][4]
        else:
            ciphertext = matrix[row1]
[col1-1]+matrix[row1][col2-1]

    #Case 2: Same column
    elif col1 == col2:
        if row1 == 0 and row2 != 0:
            ciphertext = matrix[4]
[col1]+matrix[row2-1][col1]
        elif row1 != 0 and row2 == 0:
            ciphertext = matrix[row1-1]
[col1]+matrix[4][col1]
        elif row1 == 0 and row2 == 0:
            ciphertext = matrix[4]
[col1]+matrix[4][col1]
        else:
            ciphertext = matrix[row1-1]
[col1]+matrix[row2-1][col2]

    #Case3: Rectangle
    else:
        ciphertext = matrix[row1][col2] +
matrix[row2][col1]
    return ciphertext
```

In [26]:

```
ciphertext = 'DF ULYP XO CQD LFWC RUBHEDY, CQDY
EGIYIG LMP CQDYF.LYFNH HXPZ CQF YNILXKPB "NDCB_
PQ FQ CQPKZBK. OLC PMCUNUG YMB IPYDIDCQ OXY CME
AULHDFY. CX OALG RMB FWGI PMXBNTIP ZLSWS LFWFE
KIBAT XMNKI PMBYD.'
ciphertext_modified = ciphertext.replace("
","").replace(",","").replace(".","").replace(

ans = ''
for i in range(0,len(ciphertext_modified),2):
    bigraph =
ciphertext_modified[i]+ciphertext_modified[i+1]
    converted_bigraph = decryption(matrix,bigra
    ans+=converted_bigraph[0]+converted_bigraph
ans
```

Out [26]:

```
'BEWARYOFTHENEXTCHAMBERTHEREISVERYLITTLEIOYTHER
```

In [ ]:

Assignment 2

● GRADED

GROUP

Gargi Sarkar
Anindya Ganguly
Utkarsh Srivastava
✎ 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