Q1 Team Name

0 Points

Enciphered

Q2 Commands

5 Points

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

enter, enter, enter, enter, enter, give, read

Q3 Analysis

30 Points

Give a detailed description of the cryptanalysis used to figure out the password. (Use Latex wherever required. If your solution is not readable, you will lose marks. If necessary the file upload option in this question must be used TO SHARE IMAGES ONLY.)

From the TA's comment in the forum " you just have to implement the most simple attack that comes to your mind when you try to break a cryptosystem " straightly directed to implement Brute-Force to figure out the password.

In the panel, the given hash values of the password were:
20 16 61 116 89 59 96 62 35 58 56 42 27 39 119 122 12 24 40 70 38 71 14 55 121 2 74 120 94 74 4 57
The comments related to the hash to help the decryption showed in the panel were :

1. the above hash values of your password is made of letters between 'f' and 'u'.
2. For hashing, the password is viewed as a sequence of numbers $x_1, x_2, ..., x_m$ in the field F_{127}
3. The ith number of the hashed sequence equals $x_1^{i-1} + x_2^{i-1} + ... + x_m^{i-1}$

Comment no 3 leads to the following equations:
1. When $i = 0,\ \ 1 + 1 + 1.....up\ to\ mth\ time = 20$
this implies m=20
2. When $i = 1,\ x_1 + x_2 + ... + x_{20} = 16$

As there are 32 hash values, similarly the 32th equation is :
for $i = 31, x_1^{31} + x_2^{31} + ...x_{20}^{31} = 57$

Now as it is mentioned the password is made of letters between f and u i.e. between 102 to 117.
As we are using brute force, which leads to 16^20 possibilities, a manual count of which is not possible we have written code to help us(please find it in the code section)
The explanation of the code is straightforward as its starts by printing letters between f to u, mapping letter to their corresponding ASCII values, then importing the iterative tools library and using combination with replacement functions which returns array of the desired length using number 102 to 117.
We are checking if the returned array satisfies the above 32 equations (as $x_1, x_2, ...x_m$ are from the field F_{127}, before equating left-hand side of the above equations a mod by 127 will be done with sum). If it satisfies all the above 32 equations, the function will break the loop and will print the desired list of numbers.
next, we will map these numbers from ASCII values to letter, which will lead us to our final password.

No files uploaded

Q4 Password
15 Points

What was the final command used to clear this level?

fhhiijklmmmmnnoppqrt

Q5 Codes

0 Points

It is MANDATORY that you upload the codes used in the cryptanalysis. If you fail to do so, you will be given 0 for the entire assignment.

▾ ASSIGNTMENT7.ipynb Download

In [9]:
```
from itertools import
combinations_with_replacement
```

In [10]:
```
def range_char(start, stop):
    return [chr(n) for n in range(ord(start),
ord(stop) + 1)]
ans = range_char('f','u')
ans
```

Out [10]:
```
['f',
 'g',
 'h',
 'i',
 'j',
 'k',
 'l',
 'm',
 'n',
 'o',
 'p',
 'q',
 'r',
 's',
 't',
 'u']
```

In [11]:
```
def get_ascii(ans):
    return [ord(n) for n in ans]
res = get_ascii(ans)
```

In [12]:
```
res
```

Out [12]:
```
[102,
 103,
 104,
 105,
 106,
 107,
 108,
 109,
 110,
 111,
 112,
 113,
 114,
 115,

 116,
 117]
```

In [13]:
```
hash_key =
[20,16,61,116,89,59,96,62,35,58,56,42,27,39,119,122,12,
```

In [14]:
```
def Satisfy_eqn(ans,hash_key):
    for i in range(32):
        if(check(ans,hash_key,i)==False):
            return False
    return True
```

In [15]:
```
def check(ans,hash_key,i):
    res = 0
    for val in ans:
        res = (res + pow(val,i,127))%127
    return res==hash_key[i]
```

In [ ]:
```
x = combinations_with_replacement(res,20)
ctr = 0
for i in x:
    if(Satisfy_eqn(i,hash_key)==True):
        print(i)
        break
    # print(i)
    ctr+=1
    if(ctr%100000000==0):
        print("ctr=",ctr)
```

In [ ]:
```

```

Assignment 7

GRADED

GROUP

Anindya Ganguly
Utkarsh Srivastava
Gargi Sarkar

View or edit group

TOTAL POINTS

45 / 50 pts

QUESTION 1

Team Name — **0** / 0 pts

QUESTION 2

Commands — **5** / 5 pts

QUESTION 3

Analysis — **25** / 30 pts

QUESTION 4

Password — **15** / 15 pts

QUESTION 5

Codes — **0** / 0 pts