Q1 Team Name

0 Points

Turing

Q2 Commands

5 Points

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

```
enter
enter
enter
enter
enter
give
read
c
hjkkkklllnnoopppppqrrtuu
```

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.)

The hash values of the password is:
24 115 119 13 17 54 82 123 35 64 86 123 45 93 77 108 65 89 77 55 93 113 12 93 73 14 6 99 122 38 33 16
We know these values are made of letters between 'f' and 'u' and the letters in the password are in alphabetical order.
Given that password is a sequence of numbers
$x_1, x_2, ..., x_m$ in the field F_{127}.
The ith number of hashed sequence equals :

$$x_1^{i-1} + x_2^{i-1} + ... + x_m^{i-1}$$

where, i is between 1 to 32.
when i=1:

$$x_1^0 + x_2^0 + ... + x_m^0 = 24$$
$1 + 1 + ...m$ times = 24

It implies that we have 24 letters in our password, therefore value of m is 24.

Similarly when i=2 to i=32:

$\left(x_1^1 + x_2^1 + ... + x_m^1\right)$ ....eq(1)
$\left(x_1^2 + x_2^2 + ... + x_m^2\right)$ ....eq(2)

. . . .
. . . .
. . . .
. . . .

$\left(x_1^{30} + x_2^{30} + ... + x_m^{30}\right)$ ...eq(30)
$\left(x_1^{31} + x_2^{31} + ... + x_m^{31}\right)$ ...eq(31)

from equation 1:
we get to know that modular sum of ascii values of all letters in password is 115.
we know that letters in password are 'f'(102) to 'u'(117).
password length is 24 letters.

minimum sum of password is 2448:
$102 + 102 + ...24$ times = 2448
maximum sum of password is 2808:
$117 + 117 + ...24$ times = 2808

Therefore the value of $\left(x_1^1 + x_2^1 + ... + x_{24}^1\right)$ lies between 2440 to 2808.
The possilbe sums for which we get modular sum 115 are 2528, 2655, 2782

So, now we have to find all the combinations whose sum is 2528, 2655 or 2788.

Recursive approach is used to find all the combinations with given sum. The code is given in next section.
It is found that combinations with sum 2655 and 2788 does not satisfy above 31 equations.

But, one of the the combinations whose sum is 2655 completely satisfies all the equations.

The combination is:

[105,105,106,106,106,106,106,106,106,107,108,108,109,112,112,114,116,116,116,117,117,117,117,117]

The text corresponding to it is:
"hjkkkklllnnoopppppqrrtuu"

The passowrd is "hjkkkklllnnoopppppqrrtuu".

Algorithm used to find all the combinations with given sum:
1. Used recursion and backtracking to solve the problem.
    a. if at any time target is 0 then means that combination is our possible candidate. Check if it satisfies all equations.
    b. if current element is greater than target than ignore the subproblem and return.
    c. else, insert the present index value in that array to the current list and call the function with
        target = target-a [index] and index = index, then pop that element from current index (backtrack)
2. other optimizations are also used to get to the result faster.

No files uploaded

Q4 Password
15 Points

What was the final command used to clear this level?

hjkkkklllnnoopppppqrrtuu

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.

code.py Download

```python
def compute_pow_sum(a,i):
    s = 0
    for j in a:
        s += pow(j,i)
    return s;

def check(a):
    result = [24, 115, 119, 13, 17, 54, 82, 123, 35, 64, 86, 123, 45, 93, 77, 108, 65, 89, 77, 55, 93, 113, 12, 93, 73, 14, 6, 99, 122, 38, 33, 16]
    for i in range(1,32):
        if compute_pow_sum(a,i)%127!=result[i]:
            return False
    return True

def find_combinations(a, index, target, temp, length):
    if(target == 0 and length==0):
        if check(temp):
            print("Caught you")
            print("password combinations is",temp)
        return
    # if length is 0 and target is still non-zero, return
    if(length==0):
        return
    # condition when target is so large that even if we use largest possible value in remaining places, target sum still will not be achieved
    if(a[-1]*length<target):
        return
    start_index = a.index(temp[-1]) if len(temp)>0 else 0
    # condition when target is so small that even if we use smallest possile value in remaining places, target sum still will not be achieved
    if(a[start_index]*length>target):
        return
    for i in range(start_index,len(a)):
        # if current index value is greater than target, return
        if(a[i]>target):
            return
        temp.append(a[i])
        find_combinations(a, index, target-a[i],temp, length-1)
```

```
            temp.remove(a[i])

a = [i for i in range(102,118)]
temp = []
find_combinations(a, 0, 2528, temp, 24)
find_combinations(a, 0, 2655, temp, 24)
find_combinations(a, 0, 2782, temp, 24)

```

Assignment 7

● GRADED

GROUP

Dinkar Tewari
Rohit kushwah
Deepak Raj
View or edit group

TOTAL POINTS

50 / 50 pts

QUESTION 1

Team Name **0** / 0 pts

QUESTION 2

Commands **5** / 5 pts

QUESTION 3

Analysis **30** / 30 pts

QUESTION 4

Password **15** / 15 pts

QUESTION 5

Codes **0** / 0 pts