Q1 Teamname
0 Points

Cryvengers

Q2 Commands
5 Points

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

go, go, go, go, go, give, read

Q3 Analysis
50 Points

Give a detailed description of the cryptanalysis used to figure out the password. (Explain in less than 100 lines and 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.)

By going through the above commands, finally we have got that:
You see the following written on the panel:
6E6262640000000000188808065E16CE26FEEEBEC65E16CE2018C898865E16CE200040
90800000000018C898865E16CE26E666B6C000000006E62626400000000
As you wonder what do these numbers mean, you hear a whisper in your ears ...
"I am so happy that he went away without noticing me. He is the one who bound me to the hole. Oh, I was so scared that he will notice me!
You must be wondering about these numbers. This is the hash value of your password which is created by a toy version of SHA-3. This version of SHA-3 has only three step mappings:Theta, Pi, Chi. Also, the password is no more than 16 characters.

This has been seen. It has been given that password is not more than 128 bits. As our state matrix has 1600 bits, we would have have only first 128 bits filled and remaining bits are considered to be padding and are 0's. Our state matrix is of order $5 \times 5 \times 64$. So this state matrix has 2 blocks(3rd dimensional) filled and remaining are 0's.
It can be seen that the operations done by the theta,chi,pi are just along first 2 dimensions and there is no operation done along the 3rd dimension. So if we assume 1st 64 bits as m_1 and next 64 bits as m_2(in binary)(remaining all are 0 blocks), then the operations will be in terms of m_1, m_2.(can be seen from the code provided in the resources section) and so after each round we get each block among 1600 bits to be in terms of m_1, m_2 and after final round too.
So changing any bit of m_1 doesn't affect the other bits of that block. So first we can brute force on first four bits of m_1, m_2(making remaining 0) and find the pair for which we get the output O such that the first four bits of every block of O match with the first four bits of every block of the state matrix of the hash value given.
Now repeat above process by fixing the first four bits that we have got and brute forcing on next four bits of m_1, m_2. This process will be continued till we get the final block and we can obtain the exact value of m_1, m_2.

No files uploaded

Q4 Password
25 Points

What was the final command used to clear this level?

outrjxctgfmc

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 marks for the entire assignment.

▾ Assignment7_CS641.ipynb Download

In [ ]:

```
import numpy as np

hexa = ['0', '1', '2', '3', '4', '5', '6', '7',
        '8', '9', 'A', 'B', 'C', 'D', 'E', 'F']
output =
"6E6262640000000000188808065E16CE26FEEEBEC65E16CE2018C898865E16CE2000


rem = 15 #substitute values from 0-15 to get m1, m2 in sequence
b = 1600
l = 512
c = 1024
r = 576
rounds = 24

state = np.zeros((5,5,64))
tempstate = np.zeros((5,5,64))
four_bits = ['0000', '0001', '0010', '0011',
             '0100', '0101', '0110', '0111',
             '1000', '1001', '1010', '1011',
             '1100', '1101', '1110', '1111']

given_str = ''
for i in range(rem,len(output),16):
    given_str += output[i]
print(given_str)

for m2 in four_bits:
    for m1 in four_bits:

        message = '0000' * rem + m1 + '0' * (60 - rem*4) +
        '0000' * rem + m2 + '0' * (60 - rem*4) + '0' * 448

        for i in range(5):
            for j in range(5):
                for k in range(64):
                    state[i,j,k] = 0

        for k in range(r):
            state[k//(64*5), (k//64)%5, k%64] = message[k]


        current_round = 0
        column_parity = np.zeros((5,64))


        while (current_round < rounds):

            #theta operation
            for i in range(5):
                for k in range(64):
                    column_parity[i, k] = 0
                    for j in range(5):
                        column_parity[i, k] = int(column_parity[i, k
int(state[i, j, k])

            for i in range(5):
                for j in range(5):
                    for k in range(64):
                        state[i, j, k] = int(state[i, j, k]) ^
(int(column_parity[(i+4)%5, k]) ^ int(column_parity[(i+1)%5, k]))
                        tempstate[i, j, k] = state[i, j, k]


            #pi operation
            for i in range(5):
                for j in range(5):
                    for k in range(64):
                        state[j, ((2 * i) + (3 * j)) % 5, k] = temps
j, k]


            #chi operation
            for i in range(5):
                for j in range(5):
                    for k in range(64):
                        tempstate[i, j, k] = state[i, j, k]

            for i in range(5):
                for j in range(5):
                    for k in range(64):
                        state[i, j, k] = int(tempstate[i, j, k]) ^
int(~int(tempstate[i, (j+1)%5, k]) & int(tempstate[i, (j+2)%5, k]))

            current_round += 1


        k = 0
        out_str = ''
        while (k < l):


            index = 0
```

```
                for j in [3,2,1,0]:
                    index = index * 2 + state[k//(64*5), (k//64) % 5, (k
j)]
                if k%64 == 4*rem:
                    out_str += hexa[int(index)]
                k += 4

            if out_str == given_str:
                print(out_str)
                print(m1)
                print(m2)
                break
        else:
            continue
        break
```

Assignment 7

UNGRADED

GROUP

KARPURAPU MANOJ KUMAR
VANDANAPU PRANAY
VANGALA KRISHNA SAI
View or edit group

TOTAL POINTS

- / 80 pts

QUESTION 1

Teamname — 0 pts

QUESTION 2

Commands — 5 pts

QUESTION 3

Analysis — 50 pts

QUESTION 4

Password — 25 pts

QUESTION 5

Codes — 0 pts