

## Q1 Team Name

0 Points

The\_Kryptonians

## Q2 Commands

5 Points

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

go → wave → dive → go → read → password

## Q3 Analysis

50 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.)

### Observations :

After observing ciphertexts of various inputs, we noticed that they consisted of letters from 'f' to 'u' only. So a total of 16 letters were used. We represented each letter with 4 bits. We mapped the letters such that f represented 0000 to u being 1111. But as the given Field is of size 128, we figured out that each byte of input was from 'ff' (which is mapped to 0) to 'mu' (which is mapped to 127). We generated plaintexts such that they consist of these 16 letters only. We also observed that on changing  $i^{th}$  block of input, all the output blocks from  $i^{th}$  block onwards (including  $i^{th}$  one) remained the same. All the output bytes before  $i^{th}$  block got changed. From this, we concluded that linear transformation matrix A is an lower triangular matrix. Therefore A will be :

$$A = \begin{bmatrix} a_{00} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{10} & a_{11} & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{20} & a_{21} & a_{22} & 0 & 0 & 0 & 0 & 0 \\ a_{30} & a_{31} & a_{32} & a_{33} & 0 & 0 & 0 & 0 \\ a_{40} & a_{41} & a_{42} & a_{43} & a_{44} & 0 & 0 & 0 \\ a_{50} & a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & 0 & 0 \\ a_{60} & a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} & 0 \\ a_{70} & a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & a_{77} \end{bmatrix}$$

Since it was given that a block is of size 8 bytes or 64 bits. So we divided our password of 32 letters into two 16 letters passwords with sizes of 8 bytes each.

### Generation of plaintexts and ciphertexts :

We used the input format  $C^{i-1}PC^{8-i}$  to generate plaintexts and generated corresponding ciphertexts. Here C is ff and P can be any pair from ff to mu. We generated our plaintexts and ciphertexts using our file gen\_inp\_out.py and stored them in the inputs.txt and outputs.txt file.

### Decryption :

As in our input format only one block is non-zero, we iterated over all possible values of diagonal elements and elements of exponent vector. If the non-zero  $i^{th}$  byte value is x then corresponding output value in that byte would be

$$O = (a_{ii}(a_{ii} * x^{e_i})^{e_i})^{e_i}.$$

Using  $x^7 + x + 1$  as a generator, operations were implemented over the Field  $F_{128}$  and this we used to carry out operations in code file decryptpion.py.

Now we iterated through all possible values of  $a_{ii}$  &  $e_i$  for each pair of plaintext and ciphertext and compared the output. We stored all possible value pairs of

$a_{ii}$  &  $e_i$ . We found 3 pairs per block.

BlockNumber	Possible Values of $a_{i,i}$	Possible Values of $e_i$
Block0	[73, 84, 20]	[18, 21, 88]
Block1	[86, 37, 70]	[33, 102, 119]
Block2	[43, 17, 15]	[39, 106, 109]
Block3	[12, 52, 100]	[71, 79, 104]
Block4	[28, 112, 62]	[58, 86, 110]
Block5	[17, 110, 11]	[29, 43, 55]
Block6	[52, 27, 27]	[1, 19, 107]
Block7	[62, 38, 21]	[24, 28, 75]

Now to find other non-diagonal elements of the matrix and to eliminate pairs, we used some more plaintexts and ciphertexts pairs. We iterated

over the above pairs to find elements between 0 to 127 such that equation O is valid. We did this in triangular manner to find  $a_{ij}$  with the help of  $a_{ii}$  and  $a_{jj}$ . We then obtained all elements next to diagonal elements. Also possible pairs of  $a_{ii}$  &  $e_i$  reduced to 1.

<i>BlockNumber</i>	<i>Final Value of <math>a_{i,i}</math></i>	<i>Final Value of <math>e_i</math></i>
<i>Block0</i>	84	21
<i>Block1</i>	70	119
<i>Block2</i>	43	39
<i>Block3</i>	12	71
<i>Block4</i>	112	86
<i>Block5</i>	11	55
<i>Block6</i>	27	19
<i>Block7</i>	38	28

Then we found the remaining elements by iterating over all possible values from 0 to 127 using the final values of the Exponent vector and already found values of Matrix A for which equation O holds.

Thus our Linear Transformation Matrix A obtained is :

$$\begin{bmatrix} 84 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 113 & 70 & 0 & 0 & 0 & 0 & 0 & 0 \\ 9 & 31 & 43 & 0 & 0 & 0 & 0 & 0 \\ 100 & 26 & 31 & 12 & 0 & 0 & 0 & 0 \\ 110 & 56 & 6 & 122 & 112 & 0 & 0 & 0 \\ 31 & 42 & 24 & 50 & 99 & 11 & 0 & 0 \\ 14 & 122 & 19 & 102 & 27 & 82 & 27 & 0 \\ 95 & 2 & 76 & 28 & 24 & 68 & 6 & 38 \end{bmatrix}$$

And the Exponent Vector E obtained is :

$$[21 \quad 119 \quad 39 \quad 71 \quad 86 \quad 55 \quad 19 \quad 28]$$

Now to decrypt our password we iterated over all possible values for a block and checked if the output after applying  $EAEAE$  function on the input is the same as our current password. We did this in two parts. Thus the obtained decrypted password is :


**Password : *uxpmblyga000000***

We figured that the '0's were used for padding thus the final password

obtained is :

**Final Password :** *uxpmxblyga*



 No files uploaded

## Q4 Password

5 Points

What was the final commands used to clear this level?

uxpmxblyga

## Q5 Codes

0 Points

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

▼ The\_Kryptonians.zip

 Download

1	Binary file hidden. You can download it using the button above.
---	---

## GROUP

Pranshu Gaur

Maryam Raza Khan

Maulik Singhal

 [View or edit group](#)

## TOTAL POINTS

**50 / 60 pts**

### QUESTION 1

Team Name

**0** / 0 pts

### QUESTION 2

Commands

**5** / 5 pts

### QUESTION 3

Analysis

**R** **40** / 50 pts

### QUESTION 4

Password

**5** / 5 pts

### QUESTION 5

Codes

**0** / 0 pts