Q1 Team Name
0 Points

INFINITY

Q2 Commands
5 Points

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

go, wave, dive, go, read

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.)

On analyzing the output produced by the server, we observed that all the outputs ranges from $[f-u]$. The odd positions contains $[f-m]$ and the even position contains $[f-u]$. The server accepts multiple blocks and produces an output block corresponding to each block. As $[ff-mu]$ comprised of 128 different characters and $[ff-mu]$ matches ASCII 128 different characters. So we mapped **ff** to **00** in incremental fashion all the way up to **mu** to **7f** in hex base system.
On observing various inputs and outputs, we observed that if we change the i^{th} pair bit of the input data, the output from the corresponding i^{th} pair onwards starts changing. This gave us the hint that the $matrix(A)$ is of the **Lower Triangular Form**.
Then we generated inputs of the form C^iPC^{7-i} (P is the non-zero pair). The constant bytes across multiset is taken as $ff\ or\ 00$. So our input has only one pair which is non zero at a time. Inputs are generated and stored in the $input.txt$ and their corresponding output in the $output.txt$. The inputs are generated for all 8 blocks where in i^{th} block, only i^{th} pair is changed and as we know a single pair has 128 possibilities $[ff-mu]$. thus in total we need $8*128$ inputs text. (if not taking into consideration zero input($fffffffffffffffff$) then need **8*127** inputs for the attack).
With the above observation in hand, it becomes fairly simple to crack the Exponential transformation box E and diagonal elements of the Linear Transform A.
We denote the elements of A by $a_{i,j}$. i being the row and j being the column. Also, e_i denotes the i_{th} element of vector $\mathbf{E}$.

$$A=\begin{bmatrix} a_{0,0} & 0 & 0 & .... & 0 \\ a_{1,0} & a_{1,1} & 0 & .... & 0 \\ a_{2,0} & a_{2,1} & a_{2,2} & .... & 0 \\ . & . & . & .... & . \\ . & . & . & .... & . \\ . & . & . & .... & . \\ a_{7,0} & a_{7,1} & a_{7,2} & .... & a_{7,7} \end{bmatrix}$$

$$E=[e_0,e_1,e_2,e_3,...,e_7]$$

Given the linear transformation is lower triangular matrix, any i^{th} block of output, is dependent on the j^{th} block of input iff $i \geq j$.
In our chosen pattern of the input if x is the value of non-zero input block(say i), then the corresponding block of output has the value

$$(a_{i,i}(a_{i,i}\times x^{e_i})^{e_i})^{e_i} \qquad \mathbf{(1)}$$

Iterating over values of $a_{i,i}$ (from 0 to 127) and e_i (from 1 to 126), the values which satisfy the eqn-1, we get 3 tuples of values for each block. Each tuple consists of $(a_{i,i}, e_i)$.

$Block\ Number$	$Pair\ of\ a_{i,i}\ and\ e_i$
0	$[(84,18),(28,21),(113,88)]$
1	$[(122,26),(62,113),(70,115)]$
2	$[(43,40),(14,89),(72,125)]$
3	$[(68,17),(95,41),(12,69)]$
4	$[(109,59),(112,90),(67,105)]$
5	$[(51,17),(11,41),(53,69)]$
6	$[(16,8),(27,25),(28,94)]$
7	$[(38,18),(35,21),(39,88)]$

Now, these values will be used to get the other elements of the matrix and eliminate some pairs among these.
Any element $a_{i,j}$ can be obtained by looking at the i_{th} block of output when the j_{th} block of input is non-zero. The ith byte of output when jth byte of input is the only non-zero byte in the input is given by $X_i=(\Sigma_{k=j}^{k=i}A_{ik}(A_{kj}B_j^{E_j})^{E_k})^{E_i}$.
To calculate this, besides the diagonal elements, some other elements of A are also needed. To discover $a_{i,j}$, all elements of the following type should be known,
$a_{n,m}, where\ n>m, j\leq n\ and\ m\leq i$ and also, $a_{n,n}\ where,\ j\leq n\leq i$.
These elements form a right-angled triangle, with corners $\{a_{j,j}, a_{i,j}\ and\ a_{i,i}\}$. We search for the non-diagonal elements iteratively, first on the elements of the form $a_{i+1,i}$, followed by $a_{i+2,i}$ and so on. While searching for the $a_{i+1,i}$, we could eliminate the extra pairs from the above matrix. For implementing alternate Multiply and Exponentiation is used. The addition in F_{128} is similar to the XOR operation in integers. For finding the matrix A and vector E, code is given in **break_cipher.py**
Finally,

$$A=\begin{bmatrix} 84 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 124 & 70 & 0 & 0 & 0 & 0 & 0 & 0 \\ 13 & 28 & 43 & 0 & 0 & 0 & 0 & 0 \\ 102 & 22 & 13 & 12 & 0 & 0 & 0 & 0 \\ 100 & 38 & 0 & 117 & 112 & 0 & 0 & 0 \\ 24 & 38 & 28 & 32 & 100 & 11 & 0 & 0 \\ 10 & 119 & 23 & 101 & 25 & 94 & 27 & 0 \\ 77 & 8 & 82 & 26 & 23 & 67 & 10 & 38 \end{bmatrix}$$

and $E=[18,115,40,69,90,41,25,18]$

Finding the Password

Using Matrix A and Vector E
The above matrix and vector E were also used to decrypt the actual password as shown in the **decrypt2.py**. Here, for both the block of the encrypted password, we processed each byte sequentially from first to eight and tried all possible values(0 to 127) for that byte and selected the values for which the output bytes after encryption (EAEAE) match with the given value up to that byte. Since ith byte of output depends only on bytes in the input, we would end up with correct values using the above procedure.

BruteForce Way
The encrypted password has 2 blocks $ijmrgggglsihgsml$ and $mtknhgfhhpgohqmu$ respectively. First, we select the first pair(other bytes don't care) such that on encrypting it, the first byte of this ciphertext and actual ciphertext match. This is the first byte of the password. For this, we brute-force each of the possible inputs, find its encrypted text using server and stop when the above condition holds. This is because the ciphertext byte at an index i depends only on plaintext bytes with $index\ \leq i$ (due to the property of lower-triangular matrix). This is shown in **decrypt1.py**
We get $rtqkbnroeb000000$ as the padded password from both the methods. Thus confirming our matrix A and Vector E are correct. On removing the padding of 0's, we get the required password i.e. **rtqkbnroeb**.

No files uploaded

Q4 Password
5 Points

What was the final commands used to clear this level?

rtqkbnroeb

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.

INFINITY.zip Download

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