Q1 Team Name

0 Points

Goldfish

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.)

Since the method of encryption was mentioned in the final page where password was present hence we performed structural analysis method to decipher the password.
we got hints that element of plaintext belong to GF(128) and it has 128 elements. Further from password , we got that character in password ranges from f to u. As output has 128 element , we concluded that input could be from 'ff' to 'mu' as 16 * 8 = 128. Further we used ASCII codes to convert all character to integers .
As each character takes half byte , so for 8 byte input we need to send 16 characters.
we used hit and trial method where we randomly sent some plaintext to generate cipher text. We found out that:
- A character present at i_{th} position in plaintext can only affect the output of character present after i_{th} position in the plaintext. Thus first character affects corresponding cipher text character of every other character , while the last character does not affect any other cipher text character. This gives us a hint of matrix being lower triangular.
- If plain text consist of only 'ff' character then corresponding cipher text also has only 'ff' characters .

We generated 128 different plain text by only varying first byte value of plain text from 'ff' to 'mu' and keeping all other byte value to 'ff'. as first byte of cipher text only dependent on first byte of plain text , we get that particular plain text value for the corresponding first byte of cipher text. we fixed that pair of character as first byte of the plain text.

Again we generated 128 different plain text by only varying second byte

value of plain text from 'ff' to 'mu' and keeping bytes from 3 to 8 as 'ff'. when for a particular pair of character as second byte of input , we got the corresponding cipher text byte which matches with the encrypted second byte . again we fixed the second byte of the plain text. This way we generated the first two byte of the plain text .
we followed the same brute force method to generate remaining bytes of the plain text.

Steps:

1. We generated 128* 8 plaintext and stored them in plaintext.text file
2. Ciphertext corresponding to the given plaintext were generated using server.py
3. We also find out about the Linear Transformation Matrix A that it had elements from GF(128) and also it is lower triangular matrix and the E box has elements between 1 and 126. If X is the value of a non zero input block(for instance i) and since we know that the matrix is lower triangular then the corrsponding block of output has value $O = a_{ij}(a_{ij} * x^{ei})^{ei}$. Then we performed operations over a finite field of 128 with generator $x^7 + x + 1$. (Here element of matrix A are shown as $a_{i,j}$ and element of exponent matrix are shown as e_i)

We compared the values of e_i and ai,j for each pair of plaintext-ciphertext and compared the output.
Possible values of $a_{i,i}$ as per block

$$a_{i,i} = \begin{bmatrix} 1 & [27, 84, 84] \\ 2 & [84, 35, 70] \\ 3 & [43, 14, 72] \\ 4 & [36, 24, 12] \\ 5 & [28, 112, 62] \\ 6 & [31, 11, 12] \\ 7 & [123, 27, 63] \\ 8 & [108, 38, 9] \end{bmatrix}$$

Similarly , Possible values of e_i as per block

$$e_i = \begin{bmatrix} 1 & [1, 19, 107] \\ 2 & [64, 73, 117] \\ 3 & [40, 89, 125] \\ 4 & [24, 28, 75] \\ 5 & [58, 86, 110] \\ 6 & [9, 44, 74] \\ 7 & [18, 21, 88] \\ 8 & [12, 14, 101] \end{bmatrix}$$

we found final value of diagonal elements of tranformation matrix A . we also found non diagonal value of matrix.

$$\begin{bmatrix} & a_{i,i} & e_i \\ 1 & 84 & 19 \\ 2 & 70 & 117 \\ 3 & 43 & 40 \\ 4 & 12 & 75 \\ 5 & 112 & 86 \\ 6 & 11 & 44 \\ 7 & 27 & 21 \\ 8 & 38 & 14 \end{bmatrix}$$

For each pair of paintext and cipher text, we determined the a_{ij} and corresponding e_i , and compared the output.
The final Linear Transformation matrix and Exponential matrix determined as

$$A = \begin{bmatrix} 84 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 115 & 70 & 0 & 0 & 0 & 0 & 0 & 0 \\ 15 & 28 & 43 & 0 & 0 & 0 & 0 & 0 \\ 96 & 26 & 0 & 12 & 0 & 0 & 0 & 0 \\ 96 & 42 & 6 & 118 & 112 & 0 & 0 & 0 \\ 19 & 39 & 31 & 51 & 111 & 11 & 0 & 0 \\ 10 & 120 & 21 & 101 & 31 & 82 & 27 & 0 \\ 64 & 12 & 94 & 28 & 20 & 72 & 11 & 38 \end{bmatrix}$$

Exponent matrix is = [19, 117, 40, 75, 86, 44, 21, 14]

Result

our password is "gsiiiplhgsjpltljiujnkglrkilqguig" .
corresponding to the given password , our password was generated in form of numbers . when we changed those numbers back to their character form using ASCII list , we got "svqqxqnkge000000" .
After removing zeros from the password , we reached level 5.

No files uploaded

Q4 Password

5 Points

What was the final commands used to clear this level?

svqqxqnkge

Q5 Codes