Q1 Team Name

0 Points

Enigma

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.)

We got the following information about the problem statement from the screen:

1) The password has been encrypted using the Block Cipher.

2) Each block in Block Cipher is 8 bytes long.

3) We are given a vector that is constructed using the

polynomial equation $x^7 + x + 1$ over F_2. This is a 8 x 1 vector over F_{128}.

4) EAEAE is the transformation that will be used, where A is a linear transformation by matrix multiplication and E is the element-wise exponentiation of an 8-byte vector.

5) The matrix A has the elements from F_{128} and it has a non-zero determinant value i.e. it is invertible.

6) Elements of the E matrix are numbers that range from 1 to 126.

7) In order to get the encrypted password from the spirit in the cave, we whispered “password” and got the text as "lhmfhrjrlpmhfmgrluilgmhomninmoim" which is our encrypted password.

8) We could see that the input was given in string format so therefore we tried to observe some kind of mapping with which we could convert the input from string to binary format. We noticed that only ‘f’ to ‘u’ characters were used giving us a total of 16 characters. So we mapped them from 0 to 15 as shown below.

'f' : '0000',
'g' : '0001',
'h' : '0010',
'i' : '0011',
'j' : '0100',
'k' : '0101',
'l' : '0110',
'm' : '0111',
'n' : '1000',
'o' : '1001',
'p' : '1010',
'q' : '1011',
'r' : '1100',
's' : '1101',

't' : '1110',
'u' : '1111'

9) The field available with us is F_{128} which consists of 128 elements and the input is of 8 Byte therefore we can have the input pairs as “ff” to “mu” which we mapped from ‘0’ to ‘127’ respectively.

10) We experimented with several input formats and discovered that changing the i^{th} bit of plain text changes all the bits in the output from the i^{th} bit forward. The presence of 0s at the end of each row in the matrix indicates that the transformation matrix (A) is a Lower Triangular matrix.

Using the above observations we decrypt the password.

11) We begin with generating plaintext inputs of the form $C^{i-1}PC^{8-i}$ using the file "plain_cipher_text_generation.py" . Only one block per input is non-zero due to the input format we utilized. As a result, we can iterate over all possible values of the transformation matrix A's diagonal members and vector E's elements. Since the matrix A is lower triangular, if x is the value of the non-zero input block i.e. $x \neq 0$ and if this block is the i^{th} block, then the corresponding i^{th} block of output has the value $\left(a_{i,i}\left(a_{i,i} * x^{e_i}\right)^{e_i}\right)^{e_i}$ where $a_{i,j}$ denotes element of matrix A where i denotes row and j denotes column, and e_i denotes the i-th element of the 8*1 vector E. We generate ciphertexts corresponding to each plaintext present in file “simpleplaintexts.txt” after this and which are stored in the file “ciphertexts.txt”. This is done in the program "plain_cipher_text_generation.py" as well.

12) The generator polynomial $x^7 + x + 1$ was then used to implement operations over F_{128}. We have used “plaintexts” to run a brute force attack to see if our ciphertexts matched the encrypted output

the encrypted output.

13) Now, for each combination of plaintext and ciphertext, we use the above processes to loop through each conceivable value of $a_{i,i}$ (diagonal elements of matrix A) and e_i and compare the results. If they're the same, we'll add them to a list that we have maintained for all potential values. We kept track of the potential value pairings of $a_{i,i}$ and e_i and discovered that each block has three pairs.

Block Number	Possible values of $a_{i,i}$	Possible values of e_i
Block 0	[84, 67]	[20, 108]
Block 1	[18, 70, 10]	[39, 106, 109]
Block 2	[33, 43, 98]	[22, 37, 68]
Block 3	[6, 9, 12]	[11, 34, 8]
Block 4	[109, 112, 67]	[59, 90, 10]
Block 5	[38, 11, 58]	[29, 43, 55]
Block 6	[27, 90, 127]	[24, 28, 75]
Block 7	[38, 61, 125]	[17, 41, 69]

14) We now need to filter these pairings and look for non-diagonal elements, as we only detected probable diagonal elements in the previous phase. To do so, we used utilized some additional plaintext-ciphertext pairings and iterated over the above $(a_{i,i}, e_i$ pairs, looking for elements in the range 0-127 (because matrix A has members from $F128$) that satisfy eqn 1.

Block Number	Final $a_{i,i}$	Final e_i

Block 0	84	20
Block 1	70	106
Block 2	43	37
Block 3	12	82
Block 4	112	90
Block 5	11	43
Block 6	27	24
Block 7	38	17

15) Analyzing the i_{th} output block where the j_{th} input block is non-zero to find an element $a_{i,j}$. To find $a_{i,j}$, we use $a_{i,i}$ and $a_{j,j}$. These three form a triangle in the matrix. Using this, we found every element next to every diagonal element. Through this, the number of possible $(a_{i,i}, e_i$ pairs (given in point #13) also reduces as for certain pairs we could not produce elements next to each diagonal element. Each list now contains only 1 element for which we can find each element next to each diagonal element and this is our final values of $a_{i,i} and e_i$

16) To find the remaining values of matrix A, we iterated over all the possible values (0-127). In order to do so we have used the final diagonal elements of A ($a_{i,i}$) and E (e_i). For each possible value between 0 and 127, we check the validity of eqn 1 and discard those that don't satisfy the equation.

The Exponentiation vector E is:
[20, 106, 37, 82, 90, 43, 24, 17]

And the Linear Transformation Matrix A is:
[[84, 114, 26, 123, 99, 24, 13, 67],
[0, 70, 17, 22, 37, 42, 121, 15],
[0, 0, 43, 3, 0, 19, 0, 81],
[0, 0, 0, 12, 122, 41, 104, 26],
[0, 0, 0, 0, 112, 100, 4, 24],
[0, 0, 0, 0, 0, 11, 94, 67],
[0, 0, 0, 0, 0, 0, 27, 6],

[0, 0, 0, 0, 0, 0, 0, 38]]

The code for the above explained procedure is given in the "Cryptanalysis_decoder.py"

17) Using the A and E matrix that we have found, we decrypted the password using "Cryptanalysis_decoder.py". The Encrypted Password: "lhmfhrjrlpmhfmgrluilgmhomninmoim" contained 32 characters. Each character is represented using 4 bits. The password is of 16 Bytes (128 bits) and the block size is 8 bytes therefore the password contains 2 blocks.
We apply the reverse transformation on each block. As we know the values of E and A we used:
$E^{-1}(A^{-1}(E^{-1}(A^{-1}(E^{-1}(EncryptedPassword)))))$

After doing so, we got the following results:
Inverse EAEAE values for block: 0: [116, 107, 110, 120, 98, 112, 113, 110]
Inverse EAEAE values for block: 1: [101, 97, 48, 48, 48, 48, 48, 48]

We tried decrypting this and found that these are ASCII codes and so we mapped them accordingly and got the password as tknxbpqnea000000.

We ignored the zeroes in the end and got the final password as “tknxbpqnea”.

No files uploaded

Q4 Password

5 Points

What was the final commands used to clear this level?

tknxbpqnea

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.

Enigma_ass5 2.zip Download

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

Assignment 5

GRADED

2 DAYS, 21 HOURS LATE

GROUP

Kajal Sethi
Pranshu Sahijwani
Gajender Sharma
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 **40** / 50 pts

QUESTION 4

Password **5** / 5 pts

QUESTION 5

Codes **0** / 0 pts