Q1 Team Name
0 Points

Group Name

force_de_fem

Q2 Commands
5 Points

List all the commands in sequence used from the start screen of this level to the end of the level

go -> wave -> dive -> go -> read -> password -> c -> qtqmyaskbz -> c

Q3 Cryptosystem
5 Points

What cryptosystem was used at this level?

The crypto system used at this level is EAEAE cipher (a substitution-affline block cipher with 5 layers) which is a variant of AES cipher. Here, E is a element wise exponential vector of 8 bytes and A is a linear transformation matrix over F_128.

Q4 Analysis
80 Points

Knowing which cryptosystem has been used at this level, give a detailed description of the cryptanalysis used to figure out the password.

The hints provided by the spirit gave the following information about the cryptosystem to be used in this level
-

- The encryption used is a block cipher with 8 bytes as block size.
- $x^7 + x + 1$ is the degree 7 irreducible polynomial used to construct the 8x1 vector over F_{128}.
- Matrix A has elements from F_{128} and elements of vector E range between 1 and 126.
- We obtained the encrypted password $"immsjrfjilmsmpksmsjikmhljrjffgumq"$ by typing "password" as told by the spirit.

Our Cryptanalysis Approach

Finding the encoding of the ciphertexts

We analyzed a few ciphertexts obtained from the server and found that similar to last level this time also the ciphertexts contained 16 characters in the range 'f' to 'u'. These characters mapped to 0-15 respectively, similar to hexadecimal base system:
{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}
Here, each character is represented using 4 bits. Therefore,, a byte consists of 2 characters. The field F_{128} consists of 128 elements, therefore we analyzed all the 128 pairs from ff-mu, and found that all these pairs are uniquely encrypted strings. We assumed that encryption is somehow related to ASCII values and thus, mapped ff-mu to 0-127 in decimal.

Observations about the cryptosystem used

The cryptosystem used in this level is an EAEAE cryptosystem and we used structural cryptanalysis to break it. As mentioned above, the input has a block size of 8 bytes as a 8*1 vector over the field F_{128}. We observer the following patterns -
- Providing all 0s as input plaintext, gives all 0s as ciphertext.
- If the first byte of input plaintext is 0, then also ciphertext comes out to be all 0s.
- If the input plaintext is changed at the i^{th}-bit, the output ciphertext also starts changing from the i^{th}-bit onwards.
On considering all the above observations we could say that matrix A could be a lower triangular matrix.

More precisely, let p_0,...,p_7 be the 8 byte input plaintext and c_0,...,c_7 be the 8 byte output ciphertext.
- If the input plaintext has p_j as non-zero byte and $p_i = 0$, $\forall i \neq j$, we get ciphertext of the following form:
$c_i = 0$ for $i < j$
- If the input is changed from p_0,...,p_k,p_{k+1},...,p_7 to p_0,...,p_k, p'_{k+1},...,p_7, ciphertext also changes after the k^{th} byte.
This indicates that all the 0s present in each row are present at the end of that row. Hence, A could be a lower triangular matrix.

Randomly generate plaintexts and their correspo

In order to crack the exponentiation transformation E and linear transformation A, we generated the input plaintexts using file $Generate - inputs.ipynb$ in the form $C^{i-1}PC^{8-i}$. That is, atleast one non-zero block is present in each input at a time. For every such block choice, 128 plaintext values from ff to mu are possible. We generated 8 sets of 128 input plaintexts containing one non-zero byte in each set. Hence, a total of 128*8 plaintexts (stored in plaintexts.txt) were generated to carry forward this attack.
The elements of 8x8 key matrix A are referred as $a_{i,j}$, where i=row and j=column of element. e_i is the i^{th} element of the 8x1 vector E.

Similar to the last level, we used $server.py$ (which uses Python's pexpect to connect to the game server) to establish connection to the game server and generate the ciphertexts (stored in $ciphertexts.txt$) corresponding to the plaintexts in the $plaintexts.txt$ file.

Calculating matrices A and E for transformation

The values of matrices A and E is calculated using $Cryptanalysis.ipynb$.Using the information specified above we infer that A is a lower triangular matrix. Let's say $\mathbf{'m'}$ $(m_i \neq 0)$ is the i^{th} non-zero input block, and the output block $\mathbf{'n'}$ is calculated as:
$n = (a_{i,i} * (a_{i,i} * m^{e_i})^{e_i})^{e_i} - - - - - - - - - > (eq)$

The above equation can be used to find the diagonal elements of A and all the elements of vector E.
To compute these values, we use the property that i^{th}

block output is dependent on j^{th} block input for a lower triangular matrix given $i \geq j$. The blocks are defined over the finite field F_{128}, which is constructed using degree 7 irreducible polynomial, $x^7 + x + 1$ over F_2. The operations addition is performed using XOR of integers over the filed F_{128}. The operations multiply and exponent are used to encrypt the plaintext and check if the output matches their corresponding ciphertext.

Now, we use the plaintext-ciphertext pairs, to predict all the possible values for the elements e_i
of the matrix E and the diagonal elements $a_{i,i}$ of matrix A, by checking if the input matches to the output. If the match happens, the values are appended to the corresponding list of each block. The given range of values for matrix E is [1-126] and the range of values for matrix A is [1-127] as it is defined over F_{128}. Each block has 3 tuple values. The possible values of e_i and $a_{i,i}$ for each block is as follows:

Block	Possible $a_{i,i}$ values	Possible e_i values
Block0	[84, 8, 109]	[17, 41, 69]
Block1	[122, 62, 70]	[26, 113, 115]
Block2	[43, 14, 72]	[40, 89, 125]
Block3	[12, 52, 100]	[71, 79, 104]
Block4	[103, 8, 112]	[2, 38, 87]
Block5	[106, 11, 70]	[10, 54, 63]
Block6	[27, 20, 124]	[26, 113, 115]
Block7	[108, 38, 9]	[12, 14, 101]

Next, we need to compute the remaining elements of matrix A. To find them we use some more plaintext-ciphertext pairs, then iterate over ($a_{i,i}$, e_i) pairs given above and check if the (eq) holds, also the elements should be in the range [1-127]. If the condition fails, we eliminate such pairs. The value $a_{i,j}$ is known using i^{th} block output iff j^{th} block input is non-zero.
We can get all these elements using the following set:
$$S_{i,j} = \{a_{n,m} \middle| n > m, j \leq n, m \leq i\} \cap \{a_{n,n} \middle| j \leq n \leq i\}$$
The elements of the set $S_{i,j}$ form a right-angled triangle with corner elements as {}. The non-diagonal elements and diagonal elements of matrix A, and the elements of vector E are known from the set.

Block	Final $a_{i,i}$ values	Final e_i values
Block0	84	17
Block1	70	115
Block2	43	40
Block3	12	71
Block4	112	87
Block5	11	54
Block6	27	26
Block7	38	14

The remaining non-diagonal element can be found by iterating over the possible values from 0-127, also using the diagonal elements of matrix A and elements of E, and thus, eliminating the values that don't hold the equation (eq). Hence we find the elements starting from $a_{i+1,i}$ and so on, discard other values.

The final values of A and E are

$$A = \begin{bmatrix} 84 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 125 & 70 & 0 & 0 & 0 & 0 & 0 & 0 \\ 13 & 28 & 43 & 0 & 0 & 0 & 0 & 0 \\ 100 & 27 & 30 & 12 & 0 & 0 & 0 & 0 \\ 101 & 42 & 13 & 123 & 112 & 0 & 0 & 0 \\ 31 & 41 & 28 & 44 & 101 & 11 & 0 & 0 \\ 16 & 118 & 23 & 101 & 27 & 83 & 27 & 0 \\ 94 & 9 & 94 & 23 & 25 & 68 & 5 & 38 \end{bmatrix}$$

$$E = \begin{bmatrix} 17 & 115 & 40 & 71 & 87 & 54 & 26 & 14 \end{bmatrix}$$

Decryption of password

The values of matrices A and E that are calculated in the above section are used to decrypt the ciphertext we obtained from level 5. The ciphertext we got **"immsjrfjilmsmpksmsjikmhljrjfgumq"**. We used $Decrypt.ipynb$ to get the plaintext here. The given ciphertext is divided into two blocks and decryption using inverse of matrices A and E is applied on each block is defined as follows:

$$(E^{-1}(A^{-1}(E^{-1}(A^{-1}(E^{-1}(ciphertext))))))$$

The output of the decryption is decrypted values for each block is [113, 116, 113, 109, 121, 97, 115, 107] and [98, 122, 48, 48, 48, 48, 48, 48] respectively. To get the password from these blocks we used the encoding ff-mu: 0-127, but the resulting value was a wrong password. Then, we tried to decode the values using ASCII encoding and obtained the password as **qtqmyaskbz000000**. We removed the trailing zeroes as they might have been used as padding bits. On entering the password **qtqmyaskbz** we cleared level 5.

References -
https://link.springer.com/content/pdf/10.1007/3-540-44987-6_24.pdf

Q5 Password
10 Points

What was the password used to clear this level?

qtqmyaskbz

Q6 Code
0 Points

Please add your code here. It is MANDATORY.

CS641_Assignment5_force_de_fem.zip Download

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

Assignment 5

Graded

Group
ASHEE JAIN
Kriti Majumdar

Srujana Sabbani

View or edit group

Total Points

75 / 100 pts

Question 1

Team Name	**0** / 0 pts

Question 2

Commands	**5** / 5 pts

Question 3

Cryptosystem	**5** / 5 pts

Question 4

Analysis	**55** / 80 pts

Question 5

Password	**10** / 10 pts

Question 6

Code	**0** / 0 pts