

Q1 Team Name

0 Points

Pasta_Sandwich

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.)

The basic idea here is that, for encrypting a plaintext block, we are multiplying it with matrices namely E and A in the order EAEAE. So, if we are able to find the elements of these matrices then we can get back the plaintext from a ciphertext by multiplying it with matrices Ainverse and Einverse in order. Also, these are additional observations required in this question:-

- 1) Block cipher is used with block size of 8 bytes.
- 2) Irreducible polynomial of degree 7 which is used is $(x^7 + x + 1)$ over F_2 .
- 3) Matrix **A** is invertible and its elements belong to F_{128} .
- 4) Elements of **E** are numbers between 1 and 126. **E** is applied on a block by taking *i*th element of block and raising it to the power equal to *i*th element of **E**.
- 5) These transformations are applied in the sequence EAEAE.
- 6) The encrypted password is `lhiqilqjugsjlnifmgtmrkrlhqlsqgiklg`.

It was also observed by seeing several ciphertexts that the characters in the cipher are only between 'f' and 'u'(including 'f' and 'u'). Similar to the previous assignment, we mapped these alphabets to the numbers 0-15. So, the encoding is f=0000, g=0001, h=0010, i=0011,....., u=1111 . Note that we have used 4 bits for encoding because we have in total 16 alphabets to encode(hence 4 bits).

In this cryptosystem, we are encrypting a plaintext block by multiplying it with matrices E and A in order EAE. Let a_{ij} be an element of A and e_i be an element of E . Also, each byte contains two alphabets as one alphabet occupies 4 bits (so 8 bits=2alphabets). Block length is given to be of 8 bytes. Therefore a block consists of 16 alphabets. But note that E is 8×1 vector and according to the rules of exponentiation as given above (i.e. Each element of block is raised to power equal to the corresponding element of E), it clues to an idea that two alphabets should be considered as one entity. In that way, there will be 8 entities in both E and plaintext block. If we don't consider this then there will be 16 entities in plaintext block while 8 entities in E causing mismatch. Also, F_{128} consists of 128 elements, so all the pairs will be within \mathbb{F}_μ . These pairs should be mapped to numbers so that multiplication can happen with E and A matrices. We tried mapping them to 0-127.

- 1) If input plaintext is all ffffffffffff then the corresponding ciphertext is also ffffffffffff
 - 2) If we change kth bit of input plaintext the ciphertext changes only after kth bit.
- For example, ffffffffffff encodes to ffffffff and fffffffffff encodes to ffffffffkngl ,
mmmmmmmmmmmmmmmmmmmm encodes to khhgkfgkpgfjglt and
mmmmmmmmmmmmmmmmmmmm encodes to khhgkfgjgmfgtrkl

If the element a_{ij} in A such that $i < j$ is non zero then the second point cannot hold. Hence for all a_{ij} in A such that $i < j$ then $a_{ij} = 0$. Hence A is lower triangular matrix.

Now, we can find the diagonal elements of A by fixing all bytes as zero except one single byte which will be non zero. We wrote python script to generate such plaintexts and corresponding ciphertexts.

After multiplying with matrices E and A in order EAEAE, we get

$$c_i = (a_{ii} * (a_{ii} * (p_i)^{e_i})^{e_i})^{e_i}, \text{ where } c_i = \text{output cipher's } i\text{th byte } p_i = \text{ith byte of plaintext}$$

a_{ii} =matrix A's ith diagonal element , e_i =ith element of E.

For every choice of block that we chose, we will get 128 possible plaintext value of an entity (there are 8 entities in a block of plaintext). Finite field F_{128} with generator $x^7 + x + 1$ is used for computations. There are multiple possible values of elements of E and A.

a00 = [84, 67]
a11 = [72, 101, 70]
A11 = [43, 14, 72],
A33 = [6, 9, 12]
A44 = [5, 31, 112]
A55 = [11, 92, 126]
A66 = [27, 66, 70]
A77 = [104, 38, 38]

E = [[20, 108], [53, 83, 118], [40, 89, 125], [11, 34, 82], [78, 85, 91], [51, 80, 123], [22, 37, 68], [1, 19, 107]]

Now, we will narrow down on the possibilities of A and E and find other non diagonal elements of A.
Firstly we have found out elements next to diagonal entries,
For all possible combinations of $(a_{i,i}, e_i)$, $(a_{i+1,i+1}, e_{i+1})$ and $a_{i+1,i}(0, 127)$ we iterate over all plaintext having non zero as ith byte and checking whether (i+1)th byte matches with cipher text.
After doing this, we get to know the accurate possible values of diagonals and exponents and we get to know elements on right of each diagonals.

Possible Diagonal Elements are:
a00 = 84
a11 = 70
A11 = 43
A33 = 12
A44 = 112
A55 = 11
A66 = 27
A77 = 38
And possible Exponents = [20, 118, 40, 82, 91, 51, 22, 19]

Similarly, we can find other non diagonal elements, and obtain a final linear transformation matrix.

Final Exponent Matrix we get :
[20, 118, 40, 82, 91, 51, 22, 19]

Final Linear Transformation matrix:
[84, 0, 0, 0, 0, 0, 0, 0]
[114, 70, 0, 0, 0, 0, 0, 0]
[14, 28, 43, 0, 0, 0, 0, 0]
[123, 23, 27, 12, 0, 0, 0, 0]
[97, 39, 6, 104, 112, 0, 0, 0]
[28, 40, 19, 40, 110, 11, 0, 0]
[17, 121, 20, 101, 4, 95, 27, 0]
[91, 13, 83, 28, 10, 71, 4, 38]

Now our encrypted password was : lhiqigjugsinjjifmgmtmrkhiqksgqikig
We processed it in two halves of 8 bytes because block size is 8 bytes.
Processing both halves one by one, for each byte of the half, we iterate over every value of byte from 0 to 127, i.e ff to mu and perform EAEAE and check if corresponding byte is equal to password.
Using this we get password from each half, and we concatenate it.
Password we get : ['t', 'w', 'q', 'x', 'c', 'x', 'o', 'p', 'g', 'o', 0, 0, 0, 0, 0, 0]
We remove the extra 0 padding at the end and presenting it as string, the final password we get is
Password: twqxcxopgo

No files uploaded

Q4 Password

5 Points

What was the final commands used to clear this level?

twqxcxopgo

Q5 Codes


0 Points

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

▼ Pasta_Sandwich.zip		Download
1	Binary file hidden. You can download it using the button above.	



Select a question.

 Group Members

 Submission History

Next Question 