Q1 Team Name

0 Points

Lazarus

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.)

The EAEAE is a weak form of SASAS attack. After inputting several plaintexts, we observed that ciphertext contains only 16 letters from 'f' to 'u.' We decided to represent each letter by 4 bits, i.e., 0000 to 1111 for 'f' to 'u.' Therefore, each byte is made up of 2 letters. Also, we know each byte is an element of the field F_{128} in range 0 to 127, so the MSB of each byte is 0. Therefore possible letter pairs are from 'ff' to 'mu.'

Cryptanalysis

Observations

By inputting multiple plaintexts we observed that -

i) If input plain text is ffffffffffffffff then output is also ffffffffffffffff.
ii) If first i bytes of plaintext is f's then first i bytes of ciphertext is also f's.
iii) If we change i^{th} byte of plaintext then output ciphertext also changes from the i^{th} byte. Let plaintext P be $p_0, p_1, \ldots, p_7$ where p_i is 1 byte. Then if we change input from $p_0, p_1, \ldots, p_k, p_{k+1}, \ldots, p_7$ to $p_0, p_1, \ldots, p_k, p'_{k+1}, \ldots, p_7$ then resulting ciphertexts differ after k^{th} byte.
The above observation hinted that matrix A is a lower triangular matrix.

To calculate transformation matrix $A \mathrm{and} E$

The matrix A is of dimension 8×8 and E is of dimension 8×1.
Let $a_{i,j} \in A$ where i is row index and j is column index and let $e_i \in E$.

For generating the plaintext set to be used in the attack we use plaingen.py. We generate plaintext using formula $C^{i-1}PC^{8-i}$ where $C = \text{'ff'}$ and $P \in [\text{ff,mu}]$ and $i \in [1, 8]$. Using this 8 sets of plaintext containing 128 plaintexts each were obtained where all plaintext in set i differed only at i^{th} byte value. These plaintexts are stored in plaintexts.txt

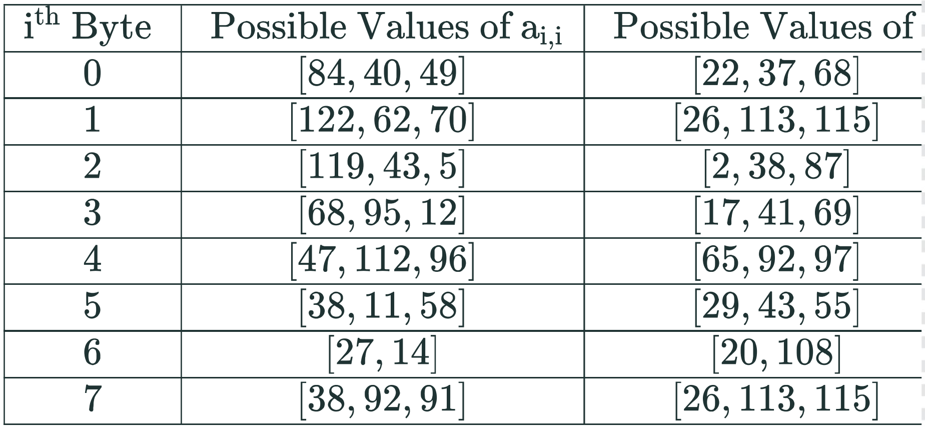
Ciphertext corresponding to each plaintext in plaintexts.txt was obtained by running robot.py, a python script using python library 'pexpect' to establish a connection to the game server, input commands in order then pass plaintext to get the corresponding ciphertext. The obtained ciphertext is stored in ciphertexts.txt

We know that matrix A is lower triangular matrix and

$$C = (A * (A * (P)^E)^E)^E \ldots\ldots 1$$

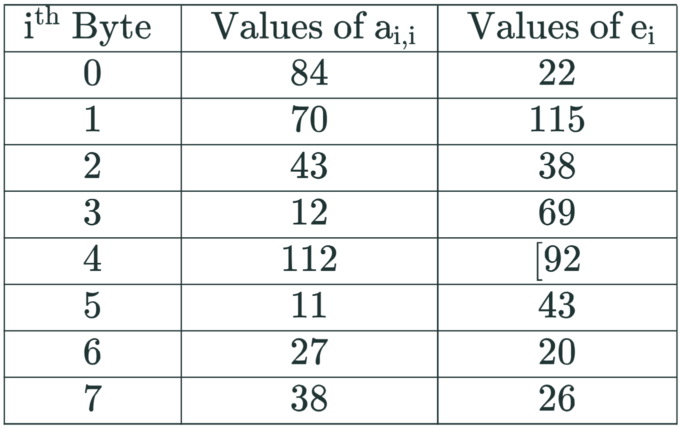
where P is plaintext and C is ciphertext. We first try to find the possible diagonal elements of matrix A and elements of E using a brute-force method.

The encryption process is performing exponentiation, linear transformation, exponentiation, linear transformation, exponentiation over Field F_{128} with modulo $x^7 + x + 1$ which is irreducible polynomial over F_2 is used to perform operations. Addition is performed as XOR of integers since the field is F_{128}. To find diagonal elements of A and elements of E, for each plaintext, ciphertext pair we iterate over values $[0, 127]$ for A and $[1, 126]$ for E to check whether plaintexts on encryption map to ciphertext or not. We store those values where plaintexts map to ciphertexts.



i^{th} Byte	Possible Values of $\text{a}_{\text{i,i}}$	Possible Values of
0	$[84, 40, 49]$	$[22, 37, 68]$
1	$[122, 62, 70]$	$[26, 113, 115]$
2	$[119, 43, 5]$	$[2, 38, 87]$
3	$[68, 95, 12]$	$[17, 41, 69]$
4	$[47, 112, 96]$	$[65, 92, 97]$
5	$[38, 11, 58]$	$[29, 43, 55]$
6	$[27, 14]$	$[20, 108]$
7	$[38, 92, 91]$	$[26, 113, 115]$

Next we needed to find non diagonal elements of A and eliminate some pairs of $(a_{i,i}, e_i)$. We iterate over plaintext-ciphertext pairs with $(a_{i,i}, e_i)$ and try to find values which satisfy equation 1 above.



i^{th} Byte	Values of $\text{a}_{\text{i,i}}$	Values of e_{i}
0	84	22
1	70	115
2	43	38
3	12	69
4	112	[92
5	11	43
6	27	20
7	38	26

To find $a_{i,j}$ we have to know all values of set

$$Z_{i,j} = \{a_{n,m} \mid n > m, j <= n, m <= i\} \cap \{a_{n,n} \mid j <= r$$

Final Linear Transformation Matrix A is,

$$A = \begin{bmatrix} 84 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 112 & 70 & 0 & 0 & 0 & 0 & 0 & 0 \\ 13 & 30 & 43 & 0 & 0 & 0 & 0 & 0 \\ 102 & 16 & 3 & 12 & 0 & 0 & 0 & 0 \\ 111 & 62 & 0 & 104 & 112 & 0 & 0 & 0 \\ 25 & 50 & 25 & 51 & 99 & 11 & 0 & 0 \\ 8 & 123 & 23 & 103 & 25 & 88 & 27 & 0 \\ 67 & 3 & 74 & 26 & 10 & 66 & 30 & 38 \end{bmatrix}$$

Final Exponent Vector E is

$$E = [22, 115, 38, 69, 92, 43, 20, 26]$$

To decrypt the Password

Using above A transformation matrix and E exponent vector , password can be decrypted by reversing the applied transformation ie for each 8 byte block of encrypted password (p) we perform following operation to get 8 byte decrypted password:

$$E^{-1}\left(A^{-1}\left(E^{-1}\left(A^{-1}\left(E^{-1}(\text{ p })\right)\right)\right)\right)$$

Our encrypted password is 'gsfojqmrimffismjfkjtkpkujlmjhjkp'

Encrypted Block 1 = 'gsfojqmrimffismj'
Encrypted Block 2 = 'fkjtkpkujlmjhjkp'

Decrypted Block 1 ASCII = [118, 116, 111, 107, 100, 112, 109, 119]
Decrypted Password 1 = 'vtokdpmw'

Decrypted Block 1 ASCII = [101, 111, 48, 48, 48, 48, 48, 48]
Decrypted Password 2 = 'eo000000'

Decrypted Password :

$$'vtokdpmweo000000'$$

We assumed '000000' at end to be padding and tried

$$'vtokdpmweo'$$

as password for level and successfully cleared it

No files uploaded

Q4 Password

5 Points

What was the final commands used to clear this level?

vtokdpmweo

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.

▾ Lazarus.zip Download

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

```
above.
```

Assignment 5

● GRADED

1 DAY, 23 HOURS LATE

GROUP

Varun Vankudre
Aditya Loth
Harsh Agarwal
View or edit group

TOTAL POINTS

55 / 60 pts

QUESTION 1

Team Name

0 / 0 pts

QUESTION 2

Commands

5 / 5 pts

QUESTION 3

Analysis

45 / 50 pts

✓ **+ 10 pts** Encoding used in the cryptosystem, i.e., odd positions contains $[f-m]$ whereas even positions contains $[f-u]$

Solution 1: Computing A and E

✓ **+ 5 pts** Correctly reason why A seems to be a lower triangular matrix.
Reason: For i^{th} plaintext byte, changing any byte at $j>i$ does not change the corresponding i^{th} ciphertext byte. However, changing any byte at $j<i$ changes the corresponding i^{th} ciphertext byte.

✓ **+ 5 pts** Compute diagonal elements of A: Brute force each a_{ii} independtly

+ 5 pts Compute non-diagonal elements of A: Order is important. Explain what elements are required beforehand to brute force a_{ij}

✓ **+ 5 pts** Correct A: A is a lower triangular matrix with correct values.

✓ **+ 5 pts** Correct E

Solution 2: Brute forcing the plaintext vector

+ 25 pts Correctly reasoning why efficient brute force attack works.

✓ **+ 10 pts** Finding the password by either method:
1. Computing inverses of A and E
2. Brute-forcing the bytes of plaintext one by one starting from the first byte.

✓ **+ 3 pts** Plaintext password blocks or their ASCII representation with padding

✓ **+ 2 pts** Final plaintext password without padding

+ 0 pts Wrong answer or NA

QUESTION 4

Password

5 / 5 pts

QUESTION 5

Codes

0 / 0 pts