Q1 Team Name
0 Points

Group Name

the_boys

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

Q3 Cryptosystem
5 Points

What cryptosystem was used at this level?

The cryptosystem used in the passage is a block cipher, which is The EAEAE (Encrypted Alphabet with Alphabet Encryption) attack is considered a weak form of SASAS (Substitution and Symmetric Algorithm Substitution) attack in cryptography, where a block of 8 bytes is transformed using two key-dependent operations: a linear transformation using a key matrix A and an exponentiation transformation using a key vector E. The input block is transformed using the sequence EAEAE, where E is applied first, followed by A, and then E again. Both E and A are considered part of the key in this cryptosystem. The coded password can be obtained by applying these transformations to the input block and decoding the resulting output block.

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 EAEAE (Encrypted Alphabet with Alphabet Encryption) attack is considered a weak form of SASAS (Substitution and Symmetric Algorithm Substitution) attack in cryptography. During our analysis, we discovered that the ciphertext resulting from this attack consists of only 16 letters, specifically ranging from 'f' to 'u'. To facilitate further analysis, we decided to represent each letter using 4 bits, assigning '0000' to 'f' and '1111' to 'u'. As a result, each byte in the ciphertext is composed of 2 letters.

Additionally, we know that each byte in the ciphertext belongs to the field F128, which has a range of 0 to 127. This means that the Most Significant Bit (MSB) of each byte must be 0, as it cannot exceed 127 in decimal representation. Consequently, the possible letter pairs in the ciphertext range from 'ff' to 'mu', considering the 4-bit representation we assigned to each letter. These observations provide valuable insights for cryptanalysis and may aid in decrypting the ciphertext and recovering the original plaintext.

During our analysis, we made several observations by inputting multiple plaintexts into the encryption algorithm:

i) When the input plaintext is set to `ffffffffffffffff`, the output ciphertext remains unchanged as `ffffffffffffffff`.

ii) If the first i bytes of the plaintext are all fs, then the first i bytes of the ciphertext also consist of fs.

iii) When we change the kth byte of the plaintext from P_k to P_o, P_1, ..., P_7, and P_k, P_{k+1}, ..., P_7, P_o, P_1, the resulting ciphertexts show differences starting from the kth byte.

This observation suggests that the transformation matrix A used in the encryption algorithm is a lower triangular matrix. 'A' is an 8 x 8 matrix, where a_{ij} represents the

element at row i and column j, and E is an 8 x 1 matrix, where e_i represents the element in the ith row.

Let $a_{ij} \in$ A, where i is the row index and j is the column index, and let $e_i \in$ E.

To generate the set of plaintexts for our attack, we used a Python script called 'generate_plain.py'. We generated plaintexts using the formula $C^{-1}PC^{(8-i)}$, where 'C' is set to 'ff' and 'PE' is a range from 'ff' to 'mu', and 'i' ranges from 1 to 8. This resulted in 8 sets of plaintexts, each containing 128 plaintexts, where all plaintexts in a set differed only at the ith byte value. These generated plaintexts were stored in 'plaintexts.txt'.

To obtain the corresponding ciphertexts for each plaintext in 'plain_texts.txt', we ran another Python script called 'run_script.pynb', which used the 'paramiko' library to establish a connection with the game server and input commands in a specific order, including passing the plaintexts as input to obtain the ciphertexts. The obtained ciphertexts were then stored in 'cipher_texts.txt'.

After this, further cryptanalysis is performed in decryptCipher.py file.

As we know so far, matrix A is a lower-triangular matrix and

$$C = (A * (A * (P)^E)^E)^E .....1$$

In order to determine the possible diagonal elements of matrix A and the elements of matrix E, we used a brute-force method. The encryption process involves multiple steps, including exponentiation, linear transformation, and modular arithmetic, over a field denoted as F_{128}, where addition is performed as XOR of integers.

The encryption process is performing exponentiation, linear transformation, exponentiation, linear transformation, and exponentiation over Field F_{128} with modulo $x^7 + x + 1$ which is an irreducible polynomial

over F_2 is used to perform operations.

We iterated over values from 0 to 127 for the possible diagonal elements of A and values from 1 to 126 for the elements of E. For each plaintext-ciphertext pair, we checked whether the plaintext on encryption maps to the ciphertext using the selected values of A and E. We stored the values of A and E where the plaintexts correctly mapped to the corresponding ciphertexts.

The table below shows the possible values of A for each byte position (ith byte) and the possible values of E obtained from our brute-force method:

ith Byte	Possible Values of A	Possible Values of E
0	[84, 67]	[20, 108]
1	[29, 52, 70]	[6, 7, 114]
2	[105, 43, 107]	[17, 41, 69]
3	[6, 9, 12]	[11, 34, 82]
4	[64, 100, 112]	[18, 21, 88]
5	[11, 41, 127]	[53, 83, 118]
6	[27, 66, 70]	[22, 37, 68]
7	[38, 61, 125]	[17, 41, 69]

Note: The values of A and E shown in the table are the possible values obtained from the brute-force method for each byte position. Further analysis and refinement may be required to determine the exact values of A and E to be used in the attack.

To accomplish the next step in our task, we required identifying the non-diagonal elements of matrix A and eliminating certain pairs of ($a_{i,i}$, e_i). Our approach involved iterating over plaintext-ciphertext pairs with ($a_{i,i}$, e_i) and attempting to identify values that satisfy the equation

$$C = (A * (A * (P)^E)^E)^E$$

, where P represents plaintext and E represents the corresponding ciphertext exponent.

ith Byte	Possible Values of A	Possible Values of E
0	84	20
1	70	114

2	43	41
3	12	82
4	112	88
5	11	53
6	27	22
7	38	17

We know,

$$Z_{i,j} = (a_{n,m} | n > m, j <= n, m <= i) \cap (a_{n,n} | j <= n)$$

From all this we get a final Linear Transformation matrix A which will be lower-triangular matrix.

$$A = \begin{pmatrix} 84 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 114 & 70 & 0 & 0 & 0 & 0 & 0 & 0 \\ 18 & 29 & 43 & 0 & 0 & 0 & 0 & 0 \\ 123 & 20 & 25 & 12 & 0 & 0 & 0 & 0 \\ 97 & 37 & 12 & 109 & 112 & 0 & 0 & 0 \\ 30 & 46 & 31 & 44 & 111 & 11 & 0 & 0 \\ 21 & 121 & 8 & 100 & 4 & 92 & 27 & 8 \\ 89 & 13 & 81 & 22 & 15 & 69 & 2 & 38 \end{pmatrix}$$

The final Exponent vector will be,

$$E = \begin{pmatrix} 20 & 114 & 41 & 82 & 88 & 53 & 22 & 17 \end{pmatrix}$$

By reversing the applied transformation, we can decrypt the encrypted password for each 8-byte block using A transformation matrix and E exponent vector.

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

The encrypted password: 'lhhofnjohghrhjkpfnfijklpfulhfull'
Encrypted Block 1: 'lhhofnjohghrhjkp'
Encrypted Block 2: 'fnfijklpfulhfull'

Decryption Process:

Decrypting Block 1:
Encrypted Block 1: 'lhhofnjohghrhjkp'
Decrypted Block 1 ASCII: [116, 115, 114, 120, 122, 122, 111, 110]
Decrypted Password 1: 'tsrxzzon'

Decrypting Block 2:
Encrypted Block 2: 'fnfijklpfulhfull'
Decrypted Block 2 ASCII: [97, 106, 48, 48, 48, 48, 48, 48]
Decrypted Password 2: 'aj000000'

Concatenating Decrypted Passwords:
Decrypted Password 1: 'tsrxzzon'
Decrypted Password 2: 'aj000000'
Final Decrypted Password: 'tsrxzzonaj000000'

Padding Assumption:
It is assumed that '000000' at the end of the decrypted password is padding.
The assumed padding is: '000000'

Result:
The password attempt 'tsrxzzonaj' was used to clear the level successfully.

Q5 Password
10 Points

What was the password used to clear this level?

tsrxzzonaj

Q6 Code
0 Points

Please add your code here. It is MANDATORY.

▾ the_boys_assignment5.zip Download

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

Assignment 5

● Graded

Group

SANKET SANJAY KALE
PRATIK MAHIPAL PATIL
ADITYA SUNILKUMAR KANKRIYA
View or edit group

Total Points

100 / 100 pts

Question 1

Team Name — **0** / 0 pts

Question 2

Commands — **5** / 5 pts

Question 3

Cryptosystem — **5** / 5 pts

Question 4

Analysis — **80** / 80 pts

Question 5

Password — **10** / 10 pts

Question 6

Code — **0** / 0 pts