Q1 Team Name
0 Points

The_Kryptonians

Q2 Commands
10 Points

List the commands used in the game to reach the ciphertext.

exit1 -> exit3 -> exit4 -> exit4 -> exit1 -> exit3 -> exit4 -> exit1 -> exit3 -> exit2 -> read

Q3 Analysis
60 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.)

Reaching the ciphertext:

To reach the ciphertext, we have to enter a series of exit commands by choosing out of the four open exits. To decide the sequence, we translated the hex codes given at each exit, by randomly trying, to get a set of phrases. We arranged these phrases to form a meaningful sentence namely 'You see a Gold-Bug in one corner. It is the key to a treasure found by' corresponding to '59 6f 75 20 73 65 65 20 61 20 47 6f 6c 64 2d 42 75 67 20 69 6e 20 6f 6e 65 20 63 6f 72 6e 65 72 2e 20 49 74 20 69 73 20 74 68 65 20 6b 65 79 20 74 6f 20 61 20 74 72 65 61 73 75 72 65 20 66 6f 75 6e 64 20 62 79' which was displayed on the screen in order when we entered the exit commands in correct order as in Ans 2 followed by 'read'.

Problem given:

On the following screen, the given problem was displayed-
N =
84364443735725034864402554533826279174703893439763343343863260342756678609216895093779263028809246505955647572176682669445270008816481771701417554768871285020442403001649254405058303439906229201909599348669565697534331652019516409514800265887388539283381053937433496994442146419682027649079704982600857517093

C =
52173497528501369942466416338781595307789995517422616782058440900593036515497800636336647821038408421981333281914863230175507615562959143163713038085557420855479590434066049717503887110897811594460482588313198119932668270597880283058143514494571977773823495602656343842623682111479031494111851017835799924 8

where C is the encrypted password.

RSA description:

RSA is given as-
1. Encryption: $C = M^e$ mod N
2. Decryption: $M = C^d$ mod N
where C is the encrypted message and M is the decrypted message or the message to be encrypted.

Deciding on the decryption method to be used:

To decrypt this, one can-
1. Find factors of N, find $\phi(N)$, and compute d. But since length of N is very large, finding factors of N isn't possible, hence, d cannot be computed in an efficient manner.
2. Its given that the exponent e = 5. Since the public exponent is small, we decided to use Coppersmith's Algorithm which uses a low-exponent attack, which doesn't require d to be computed prior.

Requirements of Coppersmith's Algorithm:

- Polynomial form as input.
Before using the message, we need to check for padding. Thus, we checked if $C^{1/e}$ is an integer or not. Using our given C, we concluded that its not an integer, hence there is padding to our message.

Assuming 'p' to be the padding added to our message M, the equation of RSA can be modified and written as: $(p + M)^e = C$ mod N

We tried to find the padding p by trying the following:
1. 'You see a Gold-Bug in one corner. It is the key to a treasure found by'
2. 'You see a Gold-Bug in one corner. It is the key to a treasure found by
'
3. 'The_Kryptonians: This door has RSA encryption with exponent 5 and the password is'
4. 'The_Kryptonians: This door has RSA encryption with exponent 5 and the password is '
and different similar combinations.

Finally after many attempts were made and the padding after converting to binary and adding to the message, we ran the code using Coppersmith's algorithm, we got the root in binary form which was then converted to text and used as password. The 4th attempt generated a password that was successfully accepted. It gave us the desired binary length of 79 and the leftmost bit would be 0. The explanations are given after the description of the algorithm.

Coppersmith's Algorithm:

Let f be a polynomial of degree δ and N be an integer. Given f and N, we
can recover all x_0 such that-
f(x_0) = 0 mod N AND $x_0 < N^{1/\delta}$.
This can be done in polynomial time.

Thus, our problem can be formulated as: f(M) = $(p + M)^e$ mod N.

We took reference from 'Lattice Reduction Techniques to Attack RSA by David Wong' mentioned in references at the end, and wrote our code. To compute the polynomial input mod N, we first converted p to its binary form namely padding_in_binary. Then we estimated the length of password M: The maximum length of our message can be around 200 bits as according to our assumption in the Coppersmith algorithm-
$x_0 < N^{1/e}$

Thus after adding padding our final polynomial becomes:
$f(M) = ((padding\_in\_binary << length\_of\_message) + M)^e - C$

The required password is the root of the above polynomial which can be
calculated using Coppersmith's Algorithm and LLL (Lattice reduction).

Getting the password:

Finally, we used the modified Coppersmith's Algorithm to calculate the root which came to be:
100001100111000010110010101000000110111011011110100110001101111001011001011001.
As length of above root is 79 , we added additional zeroes from the left side to make its length a multiple of 8 .

We then divided it into groups of 8 representing 1 byte each, and then converted the password in binary to text (using ASCII value of 8 bits integer) :
01000011 00111000 01011001 01010000 00110111 01101111 01001100 01101111 00110110 01011001

The final decrypted password was: **C8YP7oLo6Y**

References :

1. https://github.com/mimoo/RSA-and-LLL-attacks

No files uploaded

Q4 Password

10 Points

What was the final command used to clear this level?

C8YP7oLo6Y

Q5 Codes

0 Points

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

The_Kryptonians.zip Download

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

Assignment 6

GRADED

GROUP

Pranshu Gaur
Maulik Singhal
Maryam Raza Khan
View or edit group

TOTAL POINTS

80 / 80 pts

QUESTION 1

Team Name — **0** / 0 pts

QUESTION 2

Commands — **10** / 10 pts

QUESTION 3

Analysis — **60** / 60 pts

QUESTION 4

Password — **10** / 10 pts

QUESTION 5

Codes — **0** / 0 pts