Q1 Team Name

0 Points

flash

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.)

After using the read command at the end we obtained the screen with the following message:
"You see the following written on the panel:
n =
84364443735725034864402554533826279174703893439763343343863260342756678609216895093779263028809246505955647572176682669445270008816481771701417554768871285020442403001649254405058303439906229201909599348669565697534331652019516409514800265887388539283381053937433496994442146419682027649079704982600857517093

flash: This door has RSA encryption with exponent 5 and the password is
20246814261631107295898476034132329896369527825079116685612857757892232359486103941729994361167407980

438612333848331322879182789162456328153352994772895860371820367948003594656128111706995320347616444145949194970320274123036963426831242388832336757391049066416301167450591044081183258069135034000912275455680 4"

From this hint it is clear that RSA is used for encryption. For decrypting the ciphertext and obtaining the original message we have the function of the form here we don't know d: $decrypt(C, N, d) = (C)^d mod N$

#In order to get the original message we have following approach:

1.One way is to factorize N into two primes p and q as N can be written as $p * q$ but it is not possible as N is too large.

2.Other way way is to find d, for which we need to compute $\phi(N)$, but we cannot compute it without computing p and q.

Since, the value of e(exponent) is small, so we can make use of Coppersmith's theorem. Coppersmith's Theorem states that: -

If N is an integer and $f \in \mathbb{Z}[x]$ is a monic polynomial of degree d over the integers, we set $x = N^{1/d-\epsilon}$ for $1/d > \epsilon > 0$, where a monic polynomial is a single-variable polynomial (i.e., a univariate polynomial) in which the leading coefficient (the nonzero coefficient of highest degree) is equal to 1.Given $< N, f >$ an attacker can find out all integers $x_0 < X$ satisfying $f(x_0) \equiv 0(mod N)$.Lenstra-Lenstra-Lovasz (LLL) is lattice basis reduction algorithm which is used to find out
the LLL-reduced lattice basis. The LLL-reduced basis so obtained, is nearly as short as possible as there are absolute bounds
corresponding to each of the basis vectors.

#We tried to figure out whether some padding was used or not during enryption. For this we calculated $C^{1/e}$. It is not an integer. Therefore we concluded that some padding was used.

#Since, the value of e(exponent) is small, so we used

Coppersmith's theorem to break RSA algorithm.

#We can write our problem as finding the roots of the polynomial :
$(p+x)^e mod N$ where e and N are known. In this we only need to search p(padding) after which we can apply Coppersmith's Theorem and LLL to get the root.

#We tried various paddings some of those are listed below:

1. We made use of the hexadecimal numbers that appeared on our screen while using the different commands in reaching the ciphertext. The hexadecimals are :
"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"

As these numbers occurs in pairs so we got a hint that 8 bits (two hexadecimal digits represented by 4 bits each) form a group. Also we converted ASCII values to corresponding characters where each character is represented by 8 bit binary value.

We obtained "You see a Gold-Bug in one corner. It is the key to a treasure found by" as a padding. When we used it as a padding we were unable to roots of the polynomial. So we tried 2nd approach.

2. We also tried some other paddings like "IIT Kanpur", "Cryptology" but it didn't worked.

3.Finally we assumed "flash: This door has RSA encryption with exponent 5 and the password is" as a padding. This padding gives us the root

#We followed the following steps to get the original message(password):

1.Convert padding to binary
2.We don't know the length of password x but we assumed that $x < N^{1/e} \approx 4 * 10^{61}$ which can be represented by

using 205 bits. Thus, length of x cannot be greater than 205.
3.Final polynomial comes out to be $((binarypadding << x\_length) + x)^e - C$.
4.After applying Coppersmith's algorithm we obtained the root as
"1000000100001100111000010110010101000000110111011011101001100011011110011011001011001"

#In order to discover the plaintext corresponding to root, we divided the binary number in the groups of 8 bit, calculated their decimal values and converted them into characters as follows:

Binary	Decimal(ASCII VALUES)	character
01000011	67	C
00111000	56	8
01011001	89	Y
01010000	80	P
00110111	55	7
01101111	111	o
01001100	76	L
01101111	111	o
00110110	54	6
01011001	89	Y

Thus we got the password as "C8YP7oLo6Y ". After using it as final command we cleared this level.

$References$:
1.https://doc.sagemath.org/html/en/reference/polynomial_rings/sage/rings/polynomial/polynomial_modn_dense_ntl.html
2.https://github.com/mimoo/RSA-and-LLL-attack

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.

Flash.zip Download

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

Assignment 6

GRADED

GROUP

Nimit Jain
Ravi Shankar Das
Prashant Mishra
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