Q1 Team Name
0 Points

Noobs

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.)

While reaching the ciphertext (using above commands), we see several hexadecimal values on the screen. So we used ASCII table to convert it to the characters and by joining these characters, we get the string "You see a Gold-Bug in one corner. It is the key to a treasure found by". We thought that this might be the padding required but we were wrong. Now by looking at the ciphertext, we found that the cryptosystem used is RSA and the values of N, C and e is given. Since the value of $e = 5$, we can use the low exponent attack to break the cryptosystem. For low-exponent attack, $R(x) = (M + x)^5 - C$. We thought that we can find the padding M by converting the string "You see a Gold-Bug in one corner. It is the key to a treasure found by" into integer but we were wrong. However ,when we tried the same on the statement "Noobs: This door has RSA encryption with exponent 5 and the password is " the program worked perfectly. So the padding M is the integer value of the above statement. Also we know the value of C, but length of x is unknown. So we took an assumption that x is upper bounded by the value of $N^{\frac{1}{e}}$ to efficiently break the cryptosystem. So length of x should be less than $\frac{len(N)}{e}$. Since N is 1024 bit number, length of x would be less than 210. Since ASCII characters have been translated into binary, the length of x should be a multiple of 8. So we run a loop to consider all possible length of x. Now, we built the lattice using these 7 polynomials : $N^2 * K^i * x^i \; \forall 0 \leq i \leq 4, N * R(Kx), K * x * N * R(Kx)$. Then we used the LLL algorithm to find the shortest vector (in polynomial time) in the lattice. We converted shortest vector into a polynomial and find the root of that polynomial. Since the value of $x <= N^{\frac{1}{e}}$, the root that we have find should be the encrypted value of x. The root comes out to be: "317438007725212639901273." Then we convert this root into binary and then use ASCII table to find the corresponding string. The final string comes out to be: "C8YP7oLo6Y".

References :
(i) https://github.com/mimoo/RSA-and-LLL-attacks
(ii) http:/www.crypto-uni.lu/jscoron/cours/mics3crypto/cop.pdt

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.

solution.sage Download

```
def convert_to_binary(x):
        bin = ''
        for i in range(8):
                if (x >= 2**(7-i)):
                        bin += '1'
                        x = x-2**(7-i)
                else:
                        bin += '0'
        return bin

def encode(padding):
        enc_padding = ''
        for i in padding:
                str = convert_to_binary(ord(i))
                enc_padding += str
        return int(enc_padding, 2)

def decode(x):
        binary_x = bin(x).replace("0b", "")
        no_of_char = ceil(len(binary_x)/8)
        rem = no_of_char*8 - len(binary_x)
        password = ''
        for i in range(rem):
                password += '0'
        password += binary_x
        answer = ''
        for i in range(no_of_char):
                c = password[i*8:(i+1)*8]
                answer += chr(int(c, 2))
        return answer

N = 
84364443735725034864402554533826279174703893439376334343863260342756678609216895093779240
C = 
14273633006182869531348702746940453424568817097018649956391523706920736690290588678934848
K = -1
e = 5

padding = "Noobs: This door has RSA encryption with exponent 5 and the password is "
a = encode(padding)

for len_x in range(0, 210, 8):

        R.<x> = PolynomialRing(ZZ)
        pol = ((a<<len_x) + x)^e - C
        if K == -1:
            K = ceil(N**((1**2/pol.degree()) - (1/7)))
        lattice = []
        for i in range(5):
                lattice.append((K**i)*(x**i)*(N**2))
        lattice.append(N*pol(K*x))
        lattice.append(K*x*N*pol(K*x))

        mat = Matrix(ZZ, 7)

        for i in range(7):
                for j in range(7):
                        mat[i, j] = lattice[i][j]

        mat = mat.LLL()

        new_pol = 0
        for i in range(7):
                new_pol += x**i * mat[0, i] / K**i

        roots = new_pol.roots()
        if roots:
            print(roots[0][0], decode(roots[0][0]))
```

Assignment 6

GRADED

GROUP
Manu Shukla
Divyansh Bisht
Rishabh Lakhwani
View or edit group

TOTAL POINTS
75 / 80 pts

QUESTION 1
Team Name **0** / 0 pts

QUESTION 2
Commands **10** / 10 pts

QUESTION 3
Analysis R **55** / 60 pts

QUESTION 4
Password **10** / 10 pts

QUESTION 5
Codes **0** / 0 pts