Q1 Team Name
0 Points

Turing

Q2 Commands
10 Points

List the commands used in the game to reach the ciphertext

enter
enter
pluck
c
back
give
back
back
thrnxxtzy
read

Q3 Analysis
50 Points

Give a detailed analysis of how you figured out the password? (Explain in less than 500 words)

Given, the password is the element of the multiplicative group Z_p^* where p = 455470209427676832372575348833 is a prime and * is binary operation(modulo multiplication). We have three pairs of numbers of the form (a, password * g^a) where g is an element in Z_p^* and a is an integer. The g in each pair is the same.

(429, 431955503618234519808008749742)
(1973, 176325509039323911968355873643)

(7596, 98486971404861992487294722613)

Let

a_1 = 429

a_2 = 1973

a_3 = 7596

n_1 = 431955503618234519808008749742

n_2 = 176325509039323911968355873643

n_3 = 98486971404861992487294722613

Let password be pass

$pass * g^{a_1} = n_1$ .... eq(1)

$pass * g^{a_2} = n_2$ .... eq(2)

$pass * g^{a_3} = n_3$ .... eq(3)

Dividing eq(2) by eq(1), we get

$g^{a_2 - a_1} = (n_2 * n_1^{-1})$ mod p

$g^{1544} = 111590994894663139264552154672$ .... eq(4)

where n^{-1} is the inverse of n under modulo p.

Dividing eq(3) by eq(2), we get

$g^{a_3 - a_2} = (n_3 * n_2^{-1})$ mod p

$g^{5623} = 420413074251022028027270785553$ .... eq(5)

Dividing eq(3) by eq(1), we get

$g^{a_3 - a_1} = (n_3 * n_1^{-1})$ mod p

$g^{7167} = 110411376670918912626907526185$ .... eq(6)

Dividing eq(5) by eq(1)^3, we get

$g^{991} = 161798558270556961732424822635$ .... eq(7)

Dividing eq(6) by eq(7)^7, we get

$g^{230} = 263509268584013168241508095725$ .... eq(8)

Dividing eq(7) by eq(8)^4, we get

$g^{71} = 200335025748509210338477331839$ .... eq(9)

Dividing eq(2) by eq(9)^{79}, we get

$g^{14} = 21644501854425994119977888301$ .... eq(10)

Dividing eq(9) by eq(10)^5, we get

$g^1 = 52565085417963311027694339$ .... eq(11)

The value of g is 52565085417963311027694339

Using eq(1) to calculate value of password,

$$pass * g^{a_1} = n_1$$
$$pass = n_1 * (g^{a_1})^{-1}$$
$$pass = 431955503618234519808008749742 * (52565085417963311027694339^{429})^{-1}$$
$$pass = 134721542097659029845273957$$

The final password we got is 134721542097659029845273957.

The procedure used to calculate modulo inverse:

Since p is a prime number, we can use Fermat's Little Theorem to calculate inverse modulo.
According to this theorem, the inverse of a under modulo 'p' is a^{p-2} % p.
We know that a is the element of multiplicative group Z_p^*,
therefore a^{p-1} = identity element
a^{p-1} % p = 1 (identity element of multiplicative group is always 1)
multiplying both side by a^{-1}, we get
a^{p-2} % p = a^{-1}
$a^{-1} = a^{p-2}$ % p

Q4 Password
10 Points

What was the final command used to clear this level?

134721542097659029845273957

Q5 Codes
0 Points

Upload any code that you have used to solve this level

crypto.py
Download

```python
def modInverse(a, m):

        g = gcd(a, m)

        if (g != 1):
                print("Inverse doesn't exist")

        else:
                return power(a, m - 2, m)


def power(x, y, m):

        if (y == 0):
                return 1

        p = power(x, y // 2, m) % m
        p = (p * p) % m

        if(y % 2 == 0):
                return p
        else:
                return ((x * p) % m)


def gcd(a, b):
        if (a == 0):
                return b

        return gcd(b % a, a)

# This function is used to get equation without pass(which includes only g)
def get_equation(a1, a2, n1, n2, p):
    x = a1-a2
    inv = modInverse(n2, p)
    y = (n1*inv)%p
    return x, y

# This function is used to reduce power of g and get its corresponding values
# returns reduced power and its corresponding value
def solve_equation(a1, a2, n1, n2, p):
    x = a1-(a1//a2)*a2
    inv = modInverse(n2, p)
    z = power(inv, a1//a2, p)
    y = (n1*z)%p
    return x, y

```

```
p = 455470209427676832372575348833

a1, b1 = get_equation(1973, 429,
176325509039323911968355873643,
431955503618234519808008749742, p)
a2, b2 = get_equation(7596, 1973,
984869714048619924872947226l3,
176325509039323911968355873643, p)
a3, b3 = get_equation(7596, 429,
984869714048619924872947226l3,
431955503618234519808008749742, p)


print(a1, b1, a2, b2, a3, b3)
a4, b4 = solve_equation(a2, a1, b2, b1, p)
print(a1, b1, a2, b2, a3, b3, a4, b4)
a5, b5 = solve_equation(a3, a4, b3, b4, p)
print(a5, b5)
a6, b6 = solve_equation(a4, a5, b4, b5, p)
print(a6, b6)
a7, b7 = solve_equation(a2, a6, b2, b6, p)
print(a7, b7)
a8, b8 = solve_equation(a6, a7, b6, b7, p)
print(a8, b8)

temp = (176325509039323911968355873643 *
modInverse(power(b8, 1973, p), p))%p
print("password is ", temp)
```

Assignment 3

GRADED

GROUP

Dinkar Tewari

Rohit kushwah

Deepak Raj

View or edit group

TOTAL POINTS

70 / 70 pts

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

QUESTION 2
Commands **10** / 10 pts

QUESTION 3
Analysis **50** / 50 pts

QUESTION 4
Password **10** / 10 pts

QUESTION 5
Codes **0** / 0 pts