Q1 Team Name

0 Points

ANV

Q2 Commands

10 Points

List the commands used in the game to reach the ciphertext

Commands Used : –

- *to enter the small chamber* : **go**
- *to get into the underground chamber* : **enter**
- *to pick Mushrooms* : **pick**
- *to return to small Chamber* : **back**
- *to put some Mushrooms into small hole after returning to small chamber* **give**
- *to go back from the creature* : **back**
- *to reach the hidden door in the main Chamber* : **back**
- *to enter the hidden door* : **thrnxxtzy**
- *to read the cipher text* : **read**

After executing the above commands in sequence, we reached the ci

Q3 Analysis

50 Points

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

Given that : $-$

Password is an element of the multiplicative group Z_p^*.
where p = 455470209427676832372575348833 is a prime.

After looking around, we found three pairs of numbers of the form ($\mathbf{a}$, $\mathbf{password} * \mathbf{g^a}$) where $\mathbf{g}$ is a element in Z_p^* and $\mathbf{a}$ is an integer. The $\mathbf{g}$ in each pair is the same.

a_1, x = (429, 431955503618234519808008749742)
a_2, y = (1973, 176325509039323911968355873643)
a_3, z = (7596, 98486971404861992487294722613)

After looking for more numbers, we find

5____50___4_______31_______94___9

We thought this was $\mathbf{g}$ with many digits missing.
We need to figure out the password to complete the level.

■**ANALYSIS** : –

Let us assume

$$x = \left(password * g^{a_1}\right) \mod p \qquad \rightarrow (1)$$
$$y = \left(password * g^{a_2}\right) \mod p \qquad \rightarrow (2)$$
$$z = \left(password * g^{a_3}\right) \mod p \qquad \rightarrow (3)$$

- Here, we do not know the password. So we try to eliminate the password term.
  We eliminate the password term by dividing equations (3) and (2) ,

we get $\Rightarrow z * modinv(y, p) = \left(g^{a_3} * modinv(g^{a_2}, p)\right) \mod p$

$$\Rightarrow z * modinv(y, p) = g^{a_3 - a_2} \mod p \qquad \rightarrow (4)$$

here, $modinv(y, p)$ means $\mathbf{modulo\ multiplicative\ inverse}$ of $"y"$ under $"p"$.

- We need to eliminate the power $\left(a_3 - a_2\right)$ on the right side of the equation (4) to find $"g"$.

Now, we have to find some value $"d"$ such that

=> $\quad (g^{a_3 - a_2} \mod p)^d \mod p = g \qquad ==> \bigstar$

$\{$If we recall $\mathbf{RSAalgorithm}$, we have a message (M) ,encryption key (e) , decryption key (d) and a number (n) such that n is product of two prime numbers, where M and n are relatively prime and e is coprime to n.

In RSA, we find a number (d) such that $e * d * mod(\Phi(n)) = 1$

where,

Φ function is $\mathbf{Euler's\ totient\ function}$, such that $\Phi(n)$ is the count of numbers less than n and relatively prime to n. All of this can be derived from Euler's Theorem. Link to the resources of the proof are provided at the end.

Here, the encrypted message is $M^e \mod n$
and decrypted message is $(M^e \mod n)^d \mod n.$

That means applying the power of d will get rid of the exponent $"e"$ on the message M and we will only be left with M. We can use similar technique in our problem. $\}$

- In our problem, e is analogous to "$a_3 - a_2$", d is analogous to $"d"$ and M is analogous to g. Instead of n, we are using our given prime number p. g is relatively prime to p.

- So, we need to find a $"d"$ such that

=> $$(a_3 - a_2) * d \mod \Phi(p) = 1$$

where ($\Phi(p) = p - 1$, as p is a prime number.)

=> $$(a_3 - a_2) * d \mod (p - 1) = 1$$

=> $$\mathbf{d} = modinv(a_3 - a_2, p - 1) \qquad ==> \bigstar$$

On applying the power of $"d"$ on both sides of $\mathbf{equation4}$. We get

$$((z * modinv(y,p))^d) \mod p = ((g^{(a_3-a_2)} \mod p)^d) \mod p$$

- Now, we can replace $\left((g^{a_3-a_2} \mod p)^d \mod p\right)$ *with* $\mathbf{g}$

$$=> \quad ((z * modinv(y,p))^d) \mod p = \mathbf{g}$$

$$=> \quad \mathbf{g} = ((z * modinv(y,p))^d) \mod p \qquad ==> \bigstar$$

We get the value of $\mathbf{g}$ to be $\mathbf{525650854179633110276943339}$

- Now, substituting $\mathbf{g}$ in $\mathbf{equation(3)}$ to find out the value of $password$.

$$=> z = \left(password * g^{a_3}\right) \mod p$$

$$=> password = (z * modinv(g^{a_3}, p)) \mod p$$

We get the password to be $\mathbf{134721542097659029845273957}$

- We can convert the password to $\mathbf{hexadecimal}$ and convert it to $\mathbf{binary\ string}$ to get the password as $\mathbf{"open\_sesame".}$

■Resources :

1.) https://leimao.github.io/article/RSA-Algorithm/

2.) https://www.educative.io/edpresso/what-is-the-rsa-algorithm

Q4 Password

10 Points

What was the final command used to clear this level?

The final command used to clear this level is **134721542097659029845273957**
The number mentioned above is the password that we have derived.
The binary String representation of the password is => **"open_sesame".**

Q5 Codes

0 Points

Upload any code that you have used to solve this level

a.py

Download

```python
import gmpy2  ## We used gmpy2 module since it supports fast multiple precision arithmetic.
import binascii  ## We used binascii module to convert between binary and various ACII encoded binary representations.
p=455470209427676832372575348833  ## Value of p given in the cipher text.

## Below are the three pairs of numbers of the form(a,password*g^a)
a1,x=(429, 431955503618234519808008749742)
a2,y=(1973, 176325509039323911968355873643)
a3,z=(7596, 98486971404861992487294722613)


## Finding the value of d.
d = gmpy2.invert(a3-a2,p-1) # 5119281030736115206463944877 5

## Finding the value of g^(a3-a2)
g_pow_a3_minus_a2 = z*gmpy2.invert(y,p) # 225483113639385718013207578673275482155632088243572192316 25
```

```
## Finding the value of g
g = pow(g_pow_a3_minus_a2,d,p) # 52565085417963311027694339

## Finding the value of password
password = (gmpy2.invert(pow(g,a3,p),p)*z)%p #134721542097659029845273957
print("The password is ",password)

## Finding the binary string representation of the password
password_string = binascii.unhexlify(hex(password).replace("0x",""))
print("The binary string represented by the password is",password_string.decode("ascii"))
```

Assignment 3

GRADED

GROUP

Vikas

Idamakanti Venkata Nagarjun Reddy

Dibbu Amar Raja

View or edit group

TOTAL POINTS

65 / 70 pts

QUESTION 1

Team Name	**0** / 0 pts
QUESTION 2	
Commands	**10** / 10 pts
QUESTION 3	
Analysis	R **45** / 50 pts
QUESTION 4	
Password	**10** / 10 pts
QUESTION 5	
Codes	**0** / 0 pts