Q1 Team name
0 Points

Cipherberg

Q2 Commands
10 Points

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

go, enter, pluck, c, 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)

After giving the mushrooms, we were presented a screen which had some hints and equations related to multiplicative groups. This gave us the idea to use Modular Arithmetic in further analysis.

Given, the prime modulus p = 19807040628566084398385987581

Using the equations given in the hint,
$wg^{324} \equiv 11226815350263531814963336315(say, y_1)$ (mod p) -------eqn 1
$wg^{2345} \equiv 9190548667900274300830391220(say, y_2)$ (mod p) -------eqn 2
$wg^{9513} \equiv 4138652629655613570819000497(say, y_3)$ (mod p) -------eqn 3
where w is the password.

Divide eqn. 2 with eqn. 1,
$g^{2021} \equiv y_2 * y_1^{-1}$ (mod p) --------eqn 4

Divide eqn. 3 with eqn. 1,
$g^{9189} \equiv y_3 * y_1^{-1}$ (mod p) --------eqn 5

Divide eqn. 3 with eqn. 2,
$g^{7168} \equiv y_3 * y_2^{-1}$ (mod p) --------eqn 6

To perform modular division, we need to find the modular inverse of the denominators ($y_1 \text{ and } y_2$), if it exists. From properties of Multiplicative group of integers modulo n, we know that, gcd(y_1, p) = 1 and gcd(y_2, p) = 1 i.e. y_1 and y_2 are coprime to p, therefore, modular inverse of y_1 and y_2 exists under modulo p.
Modular inverse of y_1 is a number x such that $(y_1 * x)\%p = 1$.

Since p is a prime number, therefore we can use Fermat's little theorem.
Therefore, Using Fermat's little theorem, $a^{p-1} \equiv 1(\text{mod p})$
Multiplying both sides with a^{-1} and rearranging,
$\implies a^{-1} \equiv a^{p-2}(\text{mod p})$,
Using the above equation, we get
$y_1^{-1} = x \equiv y_1^{p-2}(\text{mod p}) \equiv$
$17983774594023309985368857902 \ (\text{mod p})$.

Therefore using eqn 4,
$g^{2021} \equiv (y_2 * x)$ (mod p)
$g^{2021} \equiv$
$1652807556389197278518852346006354965107111466632$

$g^{2021} \equiv 70212843693016386405770066679$ (mod p) --------eqn 7

Similarly, by solving equation 5 and 6, we get,
$g^{9189} \equiv 34263473851449952258250016781$ (mod p) --------eqn 8
$g^{7168} \equiv 63392488517373275089240592257$ (mod p) --------eqn 9

Multiplying both sides of equation 9 by inverse of $\left(g^{2021}\right)^3$ gives,

$g^{7168} * \left(\left(g^{2021}\right)^3\right)^{-1} \text{ (mod p)} \equiv$
$6339248851737327508924059257 *$
$\left(\left(g^{2021}\right)^3\right)^{-1} \text{ (mod p)}$
$g^{1105} \equiv 1332524359715193692493602650 \text{ (mod p)}$

Similarly the following calculations can be carried out:

$g^{349} \equiv g^{9189} * \left(\left(g^{1105}\right)^8\right)^{-1} \text{ (mod p)} \equiv$
$905484678554451261017569922 6\text{(mod p)}$
$g^{73} \equiv \left(g^{349}\right)^6 * \left(g^{2021}\right)^{-1} \text{ (mod p)} \equiv$
$2748579083294760009905704356\text{(mod p)}$
$g^{16} \equiv \left(g^{73}\right)^5 * \left(g^{349}\right)^{-1} \text{ (mod p)} \equiv$
$10610366411880988999637482966\text{(mod p)}$
$g^3 \equiv \left(g^{16}\right)^{22} * \left(g^{349}\right)^{-1} \text{ (mod p)} \equiv$
$5924011030095759455963670302\text{(mod p)}$
$g \equiv g^{16} * \left(\left(g^3\right)^5\right)^{-1} \text{ (mod p)} \equiv$
$192847283928500239481729\text{(mod p)}$

Also, in the hints mentioned on the panel, it is written that g is 1___4_2_______0___94____9 with some values missing. The result obtained for g from our computation agrees with this hint. Therefore, we can surely say that g is 192847283928500239481729.

To compute the password put value of g in eqn. 1,
$wg^{324} \equiv 1122681535026353181496333631 5$ (mod p)
Multiply both sides by inverse of g^{324},

$w * g^{324} * (g^{324})^{-1} \equiv$
$(11226815350263531814963336315 * (g^{324})^{-1})$ (mod p)
From properties of Multiplicative group of integers modulo p, we know that $a * a^{-1} = 1$,
Therefore,
$w \equiv (11226815350263531814963336315 *$
$7280920143223660694435112264)$ (mod p)
where the modular inverse$(g^{324})^{-1}$ is again obtained using

Fermat's little theorem. Therefore,
$w = 360852885036840078603672 5$

Hence the password is 3608528850368400786036725

The python code for above computations and for finding the modular inverse using Fermat's little theorem is attached.

Q4 Password

10 Points

What was the final command used to clear this level?

3608528850368400786036725

Q5 Codes

0 Points

Upload any code that you have used to solve this level.

▾ crypto_ass3.ipynb Download

In [15]:

```
def gcd(x, y):
    if (x == 0):
        return y
    return gcd(y % x, x)

def power(a, b, m):
    if (b == 0):
        return 1

    p = power(a, b // 2, m) % m
    p = (p * p) % m

    if (b % 2 == 0):
        return p
    else:
        return ((a * p) % m)


def modInverse(a, m):
    g = gcd(a, m)
    if(g==1):
        z=power(a, m - 2, m)
        print("The modular multiplicative
inverse is ",z)
        return z
    else:
        print("Oops! The inverse does not
```

```
exist")

y1 = 11226815350263531814963336315
y2 = 9190548667900274300830391220
y3= 4138652629655613570819000497
p = 19807040628566084398385987581

y1_inverse = modInverse(y1, p)
y2_inverse= modInverse(y2,p)
```

```
The modular multiplicative inverse is  17983774594
The modular multiplicative inverse is  14487011570
```

In [16]:

```
g_2021= (y2 * y1_inverse)%p
g_2021
```

Out [16]:

```
7021284369301638640577066679
```

In [17]:

```
g_7168= (y3 * y2_inverse)%p
g_7168
```

Out [17]:

```
6339248851737327508924059257
```

In [18]:

```
g_9189= (y3 * y1_inverse) % p
g_9189
```

Out [18]:

```
3426347385144995225825016781
```

In [19]:

```
g_1105= (g_7168 *
modInverse(power(g_2021,3,p),p)) % p
g_1105
```

```
The modular multiplicative inverse is  37593003101
```

Out [19]:

```
1332524359715193692493602650
```

In [20]:

```
g_349= (g_9189 *
modInverse(power(g_1105,8,p),p)) % p
g_349
```

```
The modular multiplicative inverse is  14402582163
```

Out [20]:

```
9054846785544512610175699226
```

In [21]:

```
g_73= (power(g_349,6,p) *
modInverse(g_2021,p)) % p
```

```
g_73
```

The modular multiplicative inverse is 16586880129

Out [21]: 2748579083294760009905704356

In [22]:

```
g_16= (power(g_73,5,p) * modInverse(g_349,p))
% p
g_16
```

The modular multiplicative inverse is 98360082593

Out [22]: 10610366411880988999637482966

In [23]:

```
g_3= (power(g_16,22,p) * modInverse(g_349,p))
% p
g_3
```

The modular multiplicative inverse is 98360082593

Out [23]: 5924011030095759455963670302

In [24]:

```
g= (g_16 * modInverse(power(g_3,5,p),p)) % p
g
```

The modular multiplicative inverse is 16621723109

Out [24]: 192847283928500239481729

In [25]:

```
password= (y1 * modInverse(power(g, 324,
p),p))%p
password
```

The modular multiplicative inverse is 72809201432

Out [25]: 3608528850368400786036725

Assignment 3

● GRADED

GROUP
SAMBHRANT MAURYA
DEEKSHA ARORA
SHRUTI SHARMA
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