Q1 Team Name

0 Points

flash

Q2 Commands

10 Points

List the commands used in the game to reach the ciphertext

go
enter
pick
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)

We were presented with a screen that had some hints and equations related to multiplicative groups.It is wriitten that "password for this level is an element of the multiplicative group Z_p^* " where p is a prime. g is also an element of this group. This gave us the idea to use Modular arithmetic for the analysis.

Given, the prime p = 455470209427676832372575348833.Considering the multiplicative group under modulo p.

Using the equations of the form (a, password * g^a) given in the hint:

$$password * g^{429} = 43195550361823451980800874 9742(say = x_1)modp --> (1)$$
$$password * g^{1973} = 176325509039323911968355873643(say = x_2)modp --> (2)$$
$$password * g^{7596} = 98486971404861992487294722613(say = x_3)modp --> (3)$$

Dividing (2) by (1), we get,
$$g^{1544} = x_2 * x_1^{-1} modp ----> (4)$$

Dividing (3) by (1), we gt,
$$g^{7167} = x_3 * x_1^{-1} modp ----> (5)$$

Dividing (3) by (2), we get,
$$g^{5623} = x_3 * x_2^{-1} modp ----> (6)$$

In order to perform modular division, we need to find the modular inverse of the denominators ($x_1$and x_2), if it exists.

From properties of the multiplicative group of integers modulo n, we know that gcd(x_1,p) = 1 and gcd(x_2,p) = 1 that means x_1 and x_2 are co-prime to p. Therefore, the modular inverse of x_1 and x_2 exists under modulo p.

Modular inverse of $x_1$is a number y_1 such that ($x_1 * y_1$)mod p = 1 (identity of the multiplicative group).

Since it is given that p is a prime number, therefore we can use Fermat's little theorem to compute the modular inverse.

Therefore using Fermat's little theorem which states that,
$$a^{p-1} \equiv 1(modp)$$

Multiplying both sides by a^{-1} we get,
$a^{p-2} \equiv a^{-1}(modp)$

After rearranging the above equation we get,

$a^{-1} \equiv a^{p-2}(modp)$

Using the above equation to find inverse of $x_1 \; and \; x_2$ we get
,

$x_1^{-1} = y_1 = x_1^{p-2}(modp) =$
$707499967902234717329046816 40(modp)$

$x_2^{-1} = y_2 = x_2^{p-2}(modp) =$
$228947149478752602606353685125(modp)$

Substituting the value of $x_2 \; and \; x_1^{-1} \; in \; (4)$ we get,

$g^{1544} = 176325509039323911968355873643 *$
$707499967902234717329046816 40(modp)$

$g^{1544} = 111590994894663139264552154672(modp) -$
$-- > (7)$

In the similar way solving (5) and (6) we get,

$g^{7167} = 110411376670918912626907526185(modp) -$
$-- > (8)$

$g^{5623} = 420413074251022028027270785553(modp) -$
$-- > (9)$

Multiplying both sides of (9) by inverse of $(g^{1544})^3$ gives,

$g^{5623} * ((g^{1544})^3)^{-1} =$
$420413074251022028027270785553 *$
$((g^{1544})^3)^{-1}(modp)$

$g^{991} = 161798558270556961732424822635(modp)$

In the similar way, the following calculations can be carried out

$g^{230} = g^{7167} * ((g^{991})^7)^{-1}(modp) =$

$$263509268584013168241508095725(modp)$$

$$g^{164} = g^{1544} * ((g^{230})^6)^{-1}(modp) = 365054265331735818534782577890(modp)$$

$$g^7 = g^{991} * ((g^{164})^6)^{-1}(modp) = 296995893405083794209776051694(modp)$$

$$g^6 = g^{991} * ((g^7)^{32})^{-1}(modp) = 313908172363422710658462184069(modp)$$

$$g = g^7 * (g^6)^{-1}(modp) = 525650854179633110276943 39(modp)$$

In the hint we were given that g looks like 5___50__4_____31_____94__9. The g we computed looks similar to it. Thus it gives us confirmation that we have computed g correctly.

Now for computing password

Multiplying $(g^{429})^{-1}$ on both sides of (1) we have,

$$password * g^{429} * (g^{429})^{-1} = 43195550361823451980800874 9742 * (g^{429})^{-1}(modp)$$

For the multiplicative group $a * a^{-1} = 1$

So,$password = 43195550361823451980800874 9742 * (g^{429})^{-1}(modp)$

Computing $(g^{429})^{-1}$ using fermat's little theorem and substituting it we get,

$$password = 43195550361823451980800874 9742 * 442956820316148690889301696615(modp)$$

Therefore,$password = 134721542097659029845273957$

We then used this password to clear the level.

We used python to implement Fermat's little theorem to find the modular inverse and do all the computation required.

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_Assignment3.ipynb Download

Implementation of Fermat's little theorem

In [76]:
```
# Function to compute gcd of two
number a and b

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

In [77]:
```
# Function to compute x^y under modulo
p

def power_under_modulo(x,y,p):
        if(y==0):
            return 1

a=power_under_modulo(x,y//2,p)%p
        a=(a*a)%p
        if(y%2==0):
            return a
        else:
            return (x*a)%p
```

In [78]:
```
# Function to find modular inverse of
a under modulo p

def modInverse(a, p):
    if(gcd(a,p)!= 1):
        print("Inverse of",a,"doesn't
exist !!")
    else:

        q=power_under_modulo(a,p-2, p)
        print("Modular multiplicative
```

```
inverse of",a,"is :",q)
        return q
```

In [79]:
```
x1=43195550361823451980800874 9742
x2=17632550903932391196835587 3643
x3=98486971404861992487294722613
p=45547020942767683237257534 8833
```

In [80]:
```
x1_inv=modInverse(x1,p)
x2_inv=modInverse(x2,p)
```

```
Modular multiplicative inverse of 431955503
Modular multiplicative inverse of 176325509
```

In [81]:
```
g_1544=(x2*x1_inv)%p
print("g_1544 :",g_1544)
```

```
g_1544 : 11159099489466313926455215 4672
```

In [82]:
```
g_7167=(x3*x1_inv)%p
print("g_7167 :",g_7167)
```

```
g_7167 : 11041137667091891262690752 6185
```

In [83]:
```
g_5623=(x3*x2_inv)%p
print("g_5623 :",g_5623)
```

```
g_5623 : 42041307425102202802727078 5553
```

In [84]:
```
g_991=
(g_5623*modInverse(power_under_modulo(g_154

print("g_991 :",g_991)
```

```
Modular multiplicative inverse of 345360603
g_991 : 16179855827055696173242482 2635
```

In [85]:
```
g_230=
(g_7167*modInverse(power_under_modulo(g_991

print("g_230 :",g_230)
```

```
Modular multiplicative inverse of 451843559
g_230 : 26350926858401316824150809 5725
```

In [86]:
```
g_164=
(g_1544*modInverse(power_under_modulo(g_230
```

```
print("g_164 :",g_164)
```

```
Modular multiplicative inverse of 199249906
g_164 : 365054265331735818534782577890
```

In [87]:
```
g_7=
(g_991*modInverse(power_under_modulo(g_164,

print("g_7 :",g_7)
```

```
Modular multiplicative inverse of 398578713
g_7 : 296995893405083794209776051694
```

In [88]:
```
g_6=
(g_230*modInverse(power_under_modulo(g_7,32

print("g_6 :",g_6)
```

```
Modular multiplicative inverse of 303462035
g_6 : 313908172363422710658462184069
```

In [89]:
```
g=(g_7*modInverse(g_6,p))%p
print("The value of g is :",g)
```

```
Modular multiplicative inverse of 313908172
The value of g is : 52565085417963311027694
```

In [90]:
```
password=
(x1*modInverse(power_under_modulo(g,429,p),
print("The password is :",password)
```

```
Modular multiplicative inverse of 190938394
The password is : 13472154209765902984527395
```

Assignment 3

GRADED

GROUP

Ravi Shankar Das
Nimit Jain
Prashant Mishra
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