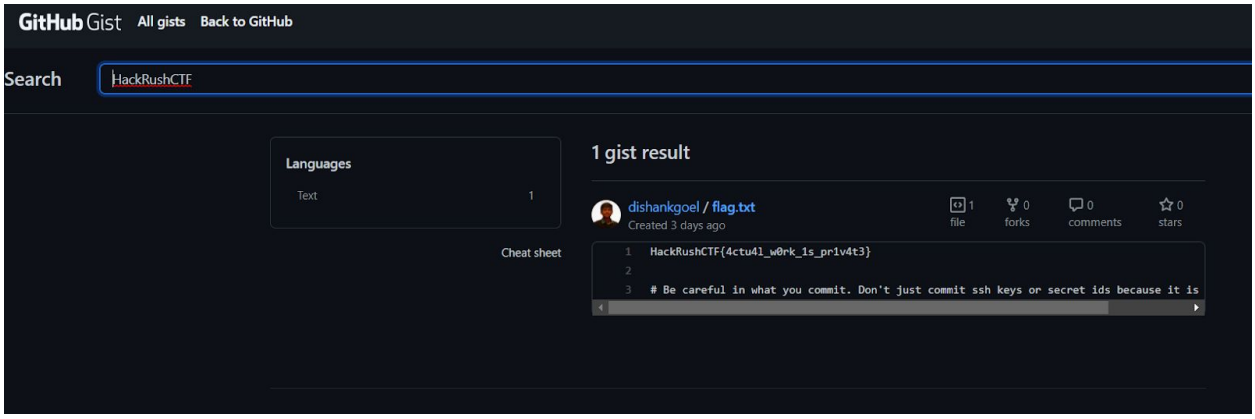
OSINT:

1. Discord problem: You know how we did it
2. Codesharing platform:
   Searched “HackRushCTF” on Github Gist and found this:



3. On the internet #2:
   https://www.linkedin.com/in/dishankgoel/

Cryptography

1. Ancient: Translated using Brahmi script. It was written Ashoka in Brahmi Script.

2. Prime Magic: GIven big number is the product of two prime numbers. The problem is of the form a^^b%c == d. We know b,c,d and we have to find the value of a. This is actually only possible if the big number is a multiplication of some prime number. This is the code for the problem.

```python
import functools
import math
import binascii
big_number = 25992347861099219061069221843214518860756327486173319027118759091795941826930677
exponent = 0x10001

def egcd2(a, b):
    x,y, u,v = 0,1, 1,0
    while a != 0:
        q, r = b//a, b%a
        m, n = x-u*q, y-v*q
        b,a, x,y, u,v = a,r, u,v, m,n
    gcd = b
    return gcd, x, y
P, Q = 375716079290975467394539222629547559486 3, 691808237490131385512539766532597713557 9
gcd, x, y = egcd2(exponent, (P-1)*(Q-1))
x=x + (P-1)*(Q-1)
c = 230269636125531384539942413418585456691619544980189231582104875209429373288994 63
ans = pow(c, x, big_number)
print(ans)
flag = b"---REDACTED---" # Who knows what was here?
flag = int(binascii.hexlify(flag), 16)
print(binascii.unhexlify('4861636B527573684354467B5253415F31735F6330306C7D'))
```

3. Prime Number 2: This is exactly the same as the above. Just the big number is the multiplication of multiple prime numbers. We just have to subtract all prime numbers with -1 and multiply them to solve the equation. We used this code for this question.

```python

import functools
import math
import binascii
big_number = 13269353506569762322866448443179444023604712744966341096534397703952746262066379915270
exponent = 0x10001
# t = 2 x 3 x 5 x 7 x 11 x 13 x 17
def egcd2(a, b):
    x,y, u,v = 0,1, 1,0
    while a != 0:
        q, r = b//a, b%a
        m, n = x-u*q, y-v*q
        b,a, x,y, u,v = a,r, u,v, m,n
    gcd = b
    return gcd, x, y
P, Q = 375716079290975467394539222629547559486 3, 691808237490131385512539766532597713557 9
R, S, T, U, V, W, X =2, 3, 5, 7, 11, 13, 17
gcd, x, y = egcd2(exponent, ((P-1)*(Q-1)*(R-1)*(S-1)*(T-1)*(U-1)*(V-1)*(W-1)*(X-1)))
x=x +  ((P-1)*(Q-1)*(R-1)*(S-1)*(T-1)*(U-1)*(V-1)*(W-1)*(X-1))
print(x)
c = 119018096473324513738497229746180211321063379102749269506790371907782514443117657629 9
ans = pow(c, x, big_number)

print(ans)
flag = b"---REDACTED---" # Who knows what was here?
flag = int(binascii.hexlify(flag), 16)
print(binascii.unhexlify('4861636B527573684354467B31305F31735F6233747433725F7468346E5F323F7D'))

```

4. Double the Trouble: This question clearly explained why doubling the encryption is a bad idea. We just encrypted the plaintext and decrypted the encrypted flag and saved both in dictionaries with the keys. We just have to check the common between the above two to get both the keys.

Reverse Engineering

1. Simple Check: The C code has provided and it is clear from those multiple if statements what exactly the code will be.