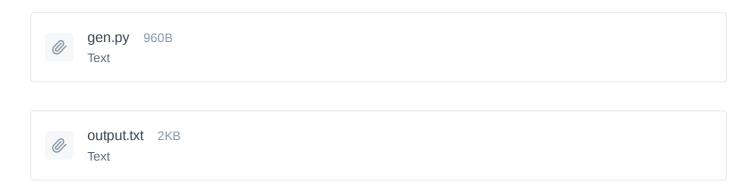


Crypto/RSA-Intro

What is RSA? Really Spicy Applesauce? Ridiculously Smart Alpaca? Random Squirrel Alliance? Nope, not at all. Just some dudes who made a cool public-key cryptosystem!



The flag is divided into three parts.

PART 1:

Part 1 is standard RSA, we have the values of p,q,ct,e

We just need to find d (private key) to decrypt the ciphertext.

```
rsa_solve.py 1KB
Text
```

PART 2:

Here we only have the values of e,n,ct

public key (e) is 3, which makes the ciphertext vulnerable to cube root attack.

If you the cubed root of ct, and if that is smaller than the cubed root of n, then your plaintext message m is just the cubed root of c.

So no need to find the private key in this case.

```
rsa_solve_2.py 1KB
Text
```

Part 3:

e: 65537

n: 107710970774233

ct: [18128889449669, 12202311999558, 10705744036504, 23864757944740]

Here we have a list of cipherText, the idea is to find the private key and iterate it through the list to get the plaintext.

using http://factordb.com/ found the factors of n .

using p,q calculated phi and found the private key.

use the d to decrypt the ct one by one

```
from Crypto.Util.number import *

e= 65537
n= 107710970774233
ct=[18128889449669, 12202311999558, 10705744036504, 23864757944740]
p=8885719
q=12121807

phi=(p-1)*(q-1)
d=pow(e,-1,phi)
flag=''
for i in ct:
    f=pow(i,d,n)
    flag += long_to_bytes(f).decode()
    f=''
print(flag)
```

Flag:

After combining all the output we get the final flag as flag{361862d054e2a9abe41cc315517cfa31}