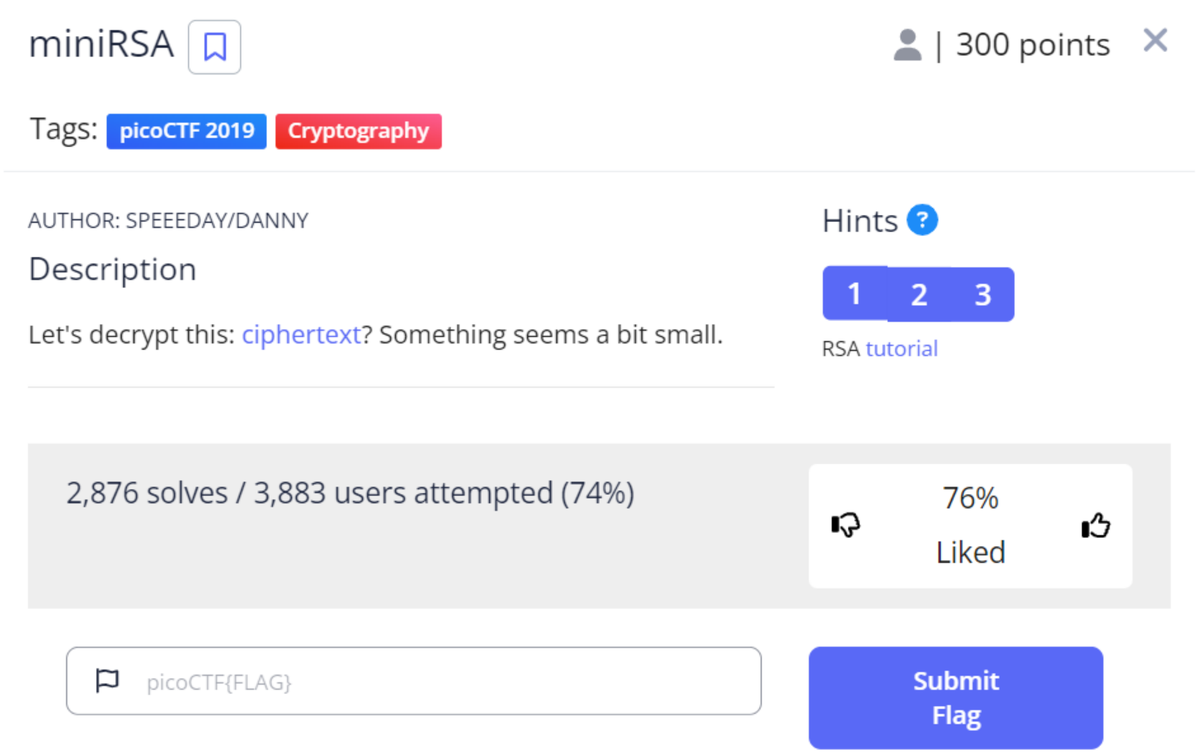
vi) miniRSA



miniRSA

| 300 points

Tags: picoCTF 2019 Cryptography

AUTHOR: SPEEEDAY/DANNY

Description

Let's decrypt this: ciphertext? Something seems a bit small.

Hints

RSA tutorial

2,876 solves / 3,883 users attempted (74%)

76%

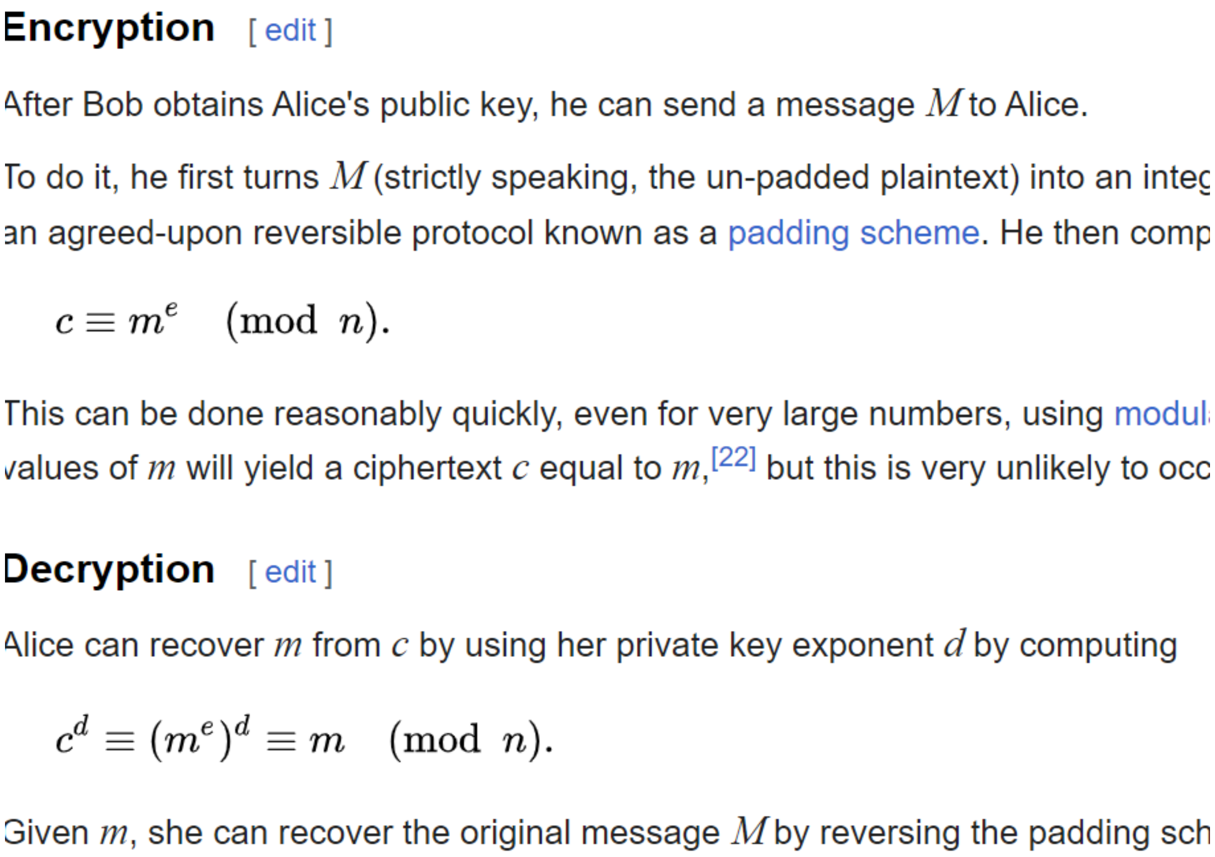
Liked

The following is our ciphertext:

2205316413931134031074603746928247799030155221252519872650
073010782049179856976080512716237308882294226369300412719
995904064931819531456392957957122459640736424089744772221
933500860936331459280832211445548332429338572369823704784
625368933

And the value of e is given to be 3.

In the wiki page:



Encryption [edit]

After Bob obtains Alice's public key, he can send a message M to Alice.

To do it, he first turns M (strictly speaking, the un-padded plaintext) into an integ
an agreed-upon reversible protocol known as a padding scheme. He then comp

$$c \equiv m^e \pmod{n}.$$

This can be done reasonably quickly, even for very large numbers, using modul
values of m will yield a ciphertext c equal to m,[22] but this is very unlikely to occ

Decryption [edit]

Alice can recover m from c by using her private key exponent d by computing

$$c^d \equiv (m^e)^d \equiv m \pmod{n}.$$

Given m, she can recover the original message M by reversing the padding sch

We observe that (mod n) remains in ciphered and deciphered text. So essentially we have to calculate the e-th root of ‘c’ to get our message.

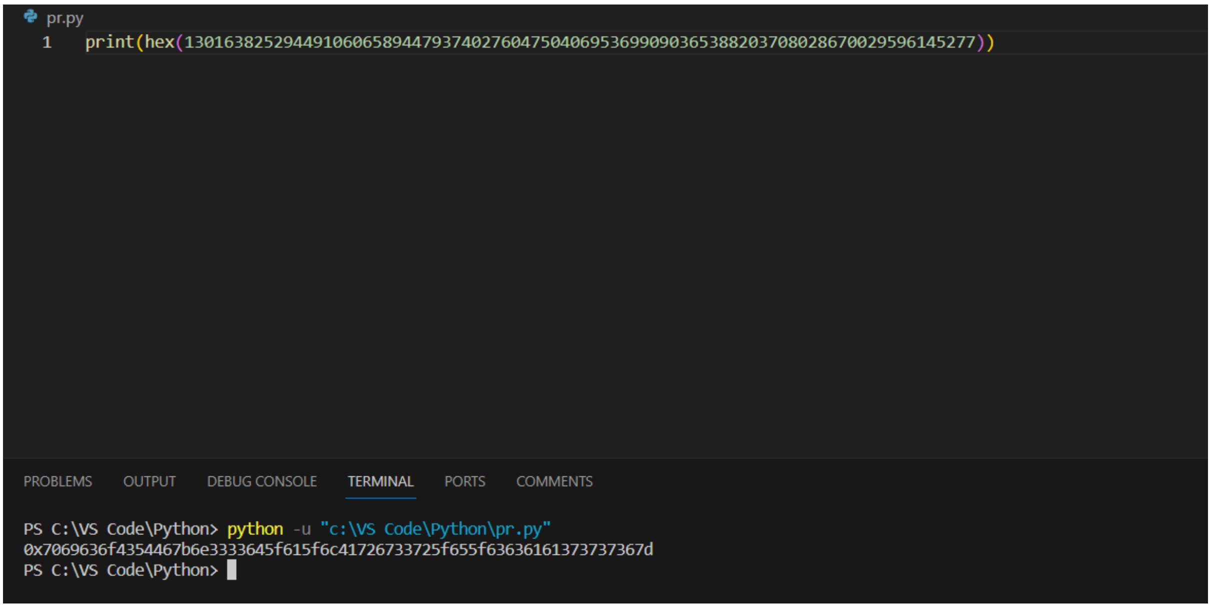
That is what we do.

The cube root (e=3) of ciphertext is:

13016382529449106065894479374027604750406953699090365388203708028670029596145277

(I had to use an online calculator for this as python was not returning the exact answer)

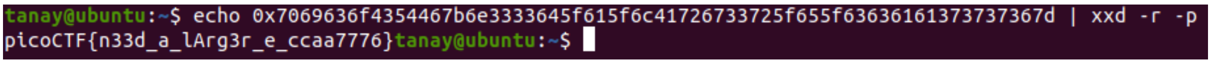
I converted it to hex. This was what was returned:



```python
print(hex(13016382529449106065894479374027604750406953699090365388203708028670029596145277))
```

```
PS C:\VS Code\Python> python -u "c:\VS Code\Python\pr.py"
0x7069636f4354467b6e3333645f615f6c41726733725f655f63636161373737367d
PS C:\VS Code\Python> 
```

Upon converting this to ASCII:



```
tanay@ubuntu:~$ echo 0x7069636f4354467b6e3333645f615f6c41726733725f655f63636161373737367d | xxd -r -p
picoCTF{n33d_a_lArg3r_e_ccaa7776}tanay@ubuntu:~$ 
```

This is our flag
[Source: dCode]