# IceCTF RSA 1

### Solved By: Nullp0inter

# RSA? Cryptography 50pts

John was messing with RSA again... he encrypted our flag! I have a strong feeling he had no idea what he was doing however, can you get the flag for us? flag.txt

# Solution

We are provided a file, `flag.txt` which contains three variables for RSA encryption:

```
N=0x180be86dc898a3c3a710e52b31de460f8f350610bf63e6b2203c08fddad44601d96eb454a34dab7684589bc32b19eb27cffff8c07179e349c

e=0x1

c=0x4963654354467b66616c6c735f61706172745f736f5f656173696c795f616e645f7265617373656d626c65645f736f5f63727564656c797d
```

`N` is known as the public modulus and is defined as `N = p * q` where `p` and `q` are two prime numbers. `e` is known as the public exponent and is used in both encryption and decrypting. `c` is what is called the cipher text and is the encrypted message we are trying to break, containing of course the flag.

This challenge has a trick to it that is pretty obvious if you know two things: firstly how RSA encryption is decrypted and secondly how the modulus operator works when we have `A mod B` where `B` is larger than `A`, in which case the result is just `A`.

Looking at the wikipedia page for RSA we can see that decryption is done with the private key exponent, `d`. With a little more reading we can also see that `ed = 1 mod phi` and that `phi = p * q`. Based on that math we know that phi is going to pretty darn big considering how large `N` is. Now you could actually bother factoring `N` to get `p` and `q` (for which I recommend using yafu)but if you see the trick here you don't even need that!

Remember what I mentioned about modulus when the right number is way bigger than the left? Well here we have `ed = 1 mod phi`. We were already *told* that `e = 0x1` which is just `1` so we sub that in and now we have `d = 1 mod phi`. Well `phi` is *way* bigger than 1, so the result is just `d = 1`. Great we have `d` so now what? Again there is an easy way and a hard way here. Sure you can read the math on the wikipedia page (or elsewhere) and decrypt it manually...*OR* you can simply use this online RSA calculator and put `d` and `c` in the correct places (making sure to tick the C has a hex string option under the text box and remove the 0x from the fron), then click 'decrypt'. 'WAIT A SECOND' I can hear some of you scream, 'it just returned a bunch of hex!'.

```
0x49 0x63 0x65 0x43 0x54 0x46 0x7b 0x66 0x61 0x6c 0x6c 0x73 0x5f 0x61 0x70 0x61 0x72 0x74 0x5f 0x73 0x6f 0x5f 0x65 0x
```

Well fret not, even though the tool is nice, it isn't nice enough to give us our string directly but the hex is all we need. Simply copy and paste it into an online hex-to-ascii conversion tool and claim your prize!

flag is: `IceCTF{falls_apart_so_easily_and_reassembled_so_crudely}`