Chall Name: It is not what you think

Description: It is not what you think

Solution:

So analaysing the out.txt file which is given is definitely not binary and the clues were given!

So it is a variant of the base2!

So the cipher is Negative Binary (Base -2)
You can decode it here
Link: https://www.dcode.fr/negabinary-system

After decoding that , you will get ascii values which will lead you to
this link
Link: https://pastebin.com/PnDXM4yP

Reading that paste carefully and analysing it , this is Fernet Cipher!

You can use any online tools or you can do it by coding as well :)

After decoding that it will lead you to this link
Link: https://ghostbin.com/bF3KC

Analysing this paste , it is definitely a Classic Diffie Hellman Key Exchange Problem

Feel free to use the code to solve

```
p = 124438483745558435263748494994847736266261238844
a = 8
b = 5
A = 40868925635114561319347724945514777605438721O1
B = 659785914328175216892615385317344443336499672221

secret = ((pow(B,a)) % p)
print(secret)
```

So we get the output as a number
If you remember the challenge correctly there is a netcat server that asks for secret key , giving this number as secret key , you will get a link
Link: https://pastebin.com/TvQiPYti

Analysing this paste carefully , this is a popular Stream Symmetric Cipher
So this one is RC4!

Feel free to use this code to solve

```
MOD = 256


def KSA(key):

    key_length = len(key)
    S = list(range(MOD))
    j = 0
    for i in range(MOD):
        j = (j + S[i] + key[i % key_length]) % MOD
        S[i], S[j] = S[j], S[i]

    return S


def PRGA(S):

    i = 0
    j = 0
    while True:
        i = (i + 1) % MOD
        j = (j + S[i]) % MOD

        S[i], S[j] = S[j], S[i]
        K = S[(S[i] + S[j]) % MOD]
        yield K


def get_keystream(key):

    S = KSA(key)
    return PRGA(S)
```

```
def logic(key, text):

    #if key is in hex
    key = codecs.decode(key, 'hex_codec')
    key = [c for c in key]
    keystream = get_keystream(key)

    res = []
    for c in text:
        val = ("%02X" % (c ^ next(keystream)))
        res.append(val)
    return ''.join(res)


def encrypt(key, plaintext):

    plaintext = [ord(c) for c in plaintext]
    return logic(key, plaintext)


def decrypt(key, ciphertext):

    ciphertext = codecs.decode(ciphertext, 'hex_codec')
    res = logic(key, ciphertext)
    return codecs.decode(res, 'hex_codec').decode('utf-8')




key = '4974206973206e6f74207768617420796f75207468696e6b'
ciphertext = '4F8A3320DF694F1C929BBF171D9C96B829CAA39F4FE686CE88FEE60E768A01'
dec= decrypt(key, ciphertext)
print('Decrypted Text:', dec)
```

So decrypting it , you will get another link (Last one :))
Link: https://hastebin.com/zitinuyomu

Checking the paste , tells us this is a variant of previous part
So it is not RC4 variant , it is RC Variant

So finally the cipher is RC6!
Decoding it using any tools!!
You will finally get the flag :)

Flag:
TamilCTF{1t_w4s_at_thiss_m0ment_th4t_3v3ry0ne_kn3w_th4t_th3_4uth0r_suck_at_Cryptography!!}

Some people might say this challenge is guessy , I dont give a damn about their opinion but I definitely know beginners will learn something new in this challenge!!

Teams who solved this challenge found it very interesting :)

LET HATERS KEEP ON BARKING