Here is the Python code to decrypt the flag :

solve.py_____
```
from Crypto.Util.number import long_to_bytes
from sympy.ntheory.residue_ntheory import sqrt_mod

# Known values from "output.txt"
p = 139944126…3
g = 2
public = 896210892…6
flag_1 = 150404492…6
flag_2 = 570520863…0
flag_3 = 467170431…3

# 1st step : We find the secret calculating the square root
roots = sqrt_mod(public, p, all_roots=True)
possible_secrets = [(root * pow(2, -1, p)) % p for root in roots]  # We divide it to 2 mode P
print(possible_secrets)

# We assume that secret is the first root (it must be checked manually if it does not work)
secret = possible_secrets[0]

# second step : We reverse the LCG to find Flag_enc
# We calculate a and c of flag_1, flag_2, flag_3
denominator = (flag_1 - flag_2) % p
inv_denominator = pow(denominator, -1, p)
a = ((flag_2 - flag_3) * inv_denominator) % p
c = (flag_2 - a * flag_1) % p

# We reverse LCG to get flag_enc
inv_a = pow(a, -1, p)
flag_enc = ((flag_1 - c) * inv_a) % p

# Step 3: We decrypt xor
secret_mod_p = secret % p
flag_original = flag_enc ^ secret_mod_p
flag = long_to_bytes(flag_original)
```
_____

THE FLAG :
CTF{f5f987eb589797a98c0234cf4dbad05351e979ef41cf6beed2c46eb029e731e7}
~Z4que