# Walkthrough: Exfiltrated Entropy 2.0

This walkthrough explains how to decrypt the given message and retrieve the original flag using Python.

## Step 1: Given Data

From the challenge file, we are given an encrypted message in Base64 format.

## Step 2: Convert Base64 to Bytes

First, we convert the Base64-encoded ciphertext back to bytes.

import base64  
  
encrypted\_b64 = "PUT\_THE\_BASE64\_STRING\_HERE"  
ciphertext = base64.b64decode(encrypted\_b64)

## Step 3: Reverse the LCG XOR Encryption

Since an LCG (Linear Congruential Generator) was used for key generation, we need to recover the keystream and decrypt the message.

# LCG Parameters  
a = 0xa1d41ebef9c575ac113fcfd5ac8dbda9  
b = 0x8dcf3cf766e0b6c30e753416a70e2367  
m = 0x100000000000000000000000000000000  
  
# Final LCG State (UUID)  
state = 0x46eda76067c4e7f37d3902ee995fcd05  
  
# Decrypt  
plaintext = bytearray()  
for byte in ciphertext:  
 state = (a \* state + b) % m # LCG Update  
 keystream\_byte = state & 0xFF # Extract a byte  
 plaintext.append(byte ^ keystream\_byte)  
  
# Convert to string  
decrypted\_flag = plaintext.decode()  
print("Decrypted Flag:", decrypted\_flag)

## Step 4: Expected Output

The final output should be:

G8KEY{A$#B0rn3\_3nTr0Py}