# ISCC2024 WriteUp

Whooops+李卓航+925798691@qq.com

### reverse+ Badcode

### 解题思路

### EXP

# 导入Crypto库中的Util模块中的number子模块

from Crypto.Util.number import \*

# 定义shift函数，用于加密算法中的位移操作

def shift(z, y, x, k, p, e):

return ((((z >> 5) ^ (y << 2)) + ((y >> 3) ^ (z << 4))) ^ ((x ^ y) + (k[(p & 3) ^ e] ^ z)))

# 定义加密函数，使用指定的密钥加密输入的数据

def encrypt(v, k):

delta = 0x61C88647

n = len(v)

rounds = 6 + 52 // n

x = 0

z = v[n - 1]

for i in range(rounds):

x = (x - delta) & 0xFFFFFFFF

e = (x >> 2) & 3

for p in range(n - 1):

y = v[p + 1]

v[p] = (v[p] + shift(z, y, x, k, p, e)) & 0xFFFFFFFF

z = v[p]

p += 1

y = v[0]

v[n - 1] = (v[n - 1] + shift(z, y, x, k, p, e)) & 0xFFFFFFFF

z = v[n - 1]

return v

# 定义解密函数，使用指定的密钥解密输入的数据

def decrypt(v, k):

delta = 0x61C88647

n = len(v)

rounds = 6 + 52 // n

x = (0 - rounds \* delta) & 0xFFFFFFFF

y = v[0]

for i in range(rounds):

e = (x >> 2) & 3

for p in range(n - 1, 0, -1):

z = v[p - 1]

v[p] = (v[p] - shift(z, y, x, k, p, e)) & 0xFFFFFFFF

y = v[p]

p -= 1

z = v[n - 1]

v[0] = (v[0] - shift(z, y, x, k, p, e)) & 0xFFFFFFFF

y = v[0]

x = (x + delta) & 0xFFFFFFFF

return v

# 主程序入口

if \_\_name\_\_ == '\_\_main\_\_':

# 初始化明文列表

plain = [0x154470BD, 0xCBAC1CA8, 0x7AFCD5E2, 0x4F666267, 0x5A76A6AE, 0x88FEB27D]

# 初始化密钥列表

key = [0x12345678, 0x9ABCDEF0, 0xFEDCBA98, 0x76543210]

# 解密数据

decrypted = decrypt(plain, key)

# 初始化标志位列表

flag = []

# 初始化16进制字符串

v16 = '674094872038771148666737'

# 遍历解密后的数据列表

for i in range(len(plain)):

# 将每个解密后的数据转换为字节序列

x = long\_to\_bytes(decrypted[i])

# 将每个字节加入到标志位列表中

for j in range(3, -1, -1):

flag.append(x[j])

# 对标志位列表中的每个元素进行异或和调整操作

for i in range(len(flag)):

flag[i] ^= ord(v16[i]) - 0x30

if i % 2:

flag[i] -= 2

else:

flag[i] += 3

# 打印解密后的字节序列

print(bytes(flag))

得到ISCC{@WEyhFC49ihdRPxPwl}