# ISCC2024 WriteUp

Whooops+李卓航+925798691@qq.com

### reverse+ CrypticConundrum

### 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 \*

# 将十六进制数0x00000034FD81BBF2F9D74983B8191BA899E3486E9B2E056DDF789CBB63转换为字节序列，并将其反转

cmp = long\_to\_bytes(0x00000034FD81BBF2F9D74983B8191BA899E3486E9B2E056DDF789CBB63)[::-1]

# 将字节序列转换为列表

cmp = list(cmp)

# 设置第一个密钥

key1 = 'So--this-is-the-right-flag'

key1\_ = []

# 设置第二个密钥

key2 = 'ISCC'

# 减去10并将结果限制在0到255之间

for i in range(len(cmp)):

cmp[i] -= 10

cmp[i] &= 0xff

# 将每个元素与其后一个元素相加，并将结果限制在0到255之间

for i in range(len(cmp)-1):

cmp[i] += cmp[i+1]

cmp[i] &= 0xff

# 将每个元素与key2的第三个字符的ASCII码值进行异或运算

for i in range(len(cmp)-1):

cmp[i] ^= ord(key2[2])

# 将偶数索引位置的元素与key2中对应位置的字符的ASCII码值进行异或运算

for i in range(0, len(cmp), 2):

cmp[i] ^= ord(key2[i%4])

# 反转列表中前半部分与后半部分的元素

for i in range(len(cmp)//2):

cmp[i], cmp[26-i-1] = cmp[26-i-1], cmp[i]

# 再次反转列表中前半部分与后半部分的元素

for i in range(len(cmp)//2):

cmp[i], cmp[26-i-1] = cmp[26-i-1], cmp[i]

# 将每个元素与key2中对应位置的字符的ASCII码值相加，并将结果限制在0到255之间

for i in range(len(cmp)):

cmp[i] += ord(key2[i%4])

cmp[i] &= 0xff

# 将列表转换为字节序列并打印结果

print(bytes(cmp))

得到ISCC{pb9H4Znp?cAvNOB&-kl`}