南邮ctf nctf CG-CTF 密码学题writeup

原创

XQin9T1an 于 2019-07-19 11:20:26 发布 1777 收藏 2

CG-CTF密码学链接：https://cgctf.nuptsast.com/challenges#Crypto

0x01 easy!

密文

```
bmN0Znt0aGlzX2lzX2Jhc2U2NF9lbmNvZGV9
```

直接base64解密
nctf{this_is_base64_encode}

0x02 Keyboard

密文

```
ytfvbhn tgbgy hjuygbn yhnmki tgvhn uygbnjm uygbn yhnijm
```

题目提示看键盘
ytfvbhn的形状跟a很像，这道题应该是根据密文在键盘上的位置组成的答案

```
ytfvbhn -------->a
tgbgy ---------->r
hjuygbn -------->e
yhnmki --------->u/v
tgvhn ---------->h
uygbnjm -------->a
uygbn ---------->c
yhnijm---------->k
```

所以flag可能是nctf{areuhack}或者是nctf{arevhack}根据意思来判断，前者可能性要大一点，都提交一下，前者是flag
flag:nctf{areuhack}

0x03 异性相吸

题目要求解密压缩文件的内容
下载链接：https://pan.baidu.com/s/1kVssDb5
提取码：assm
题目给出TIPS: 1.xor 2.hex2binary 3.len(bin(miwen))==len(bin(mingwen))

```
明文
lovelovelovelovelovelovelove
密文

V

0
0
0
```

根据题目的提示，明文和密文的二进制长度是相同的，先从16进制转化成2进制，再异或就可以得到答案
将明文密文丢进WinHex
得知16进制如下

```
明文
6C 6F 76 65 6C 6F 76 65 6C 6F 76 65 6C 6F 76 65 6C 6F 76 65 6C 6F 76 65 6C 6F 76 65
密文
0A 03 17 02 56 01 15 11 0A 14 0E 0A 1E 30 0E 0A 1E 30 0E 0A 1E 30 14 0C 19 0D 1F 10 0E 06 03 18
```

转化成二进制

```
明文
01101100 01101111 01110110 01100101 01101100 01101111 01110110 01100101
01101100 01101111 01110110 01100101 01101100 01101111 01110110 01100101
01101100 01101111 01110110 01100101 01101100 01101111 01110110 01100101
01101100 01101111 01110110 01100101 01101100 01101111 01110110 01100101
密文
00001010 00000011 00010111 00000010 01010110 00000001 00010101 00010001
00001010 00010100 00001110 00001010 00011110 00110000 00001110 00001010
00011110 00110000 00001110 00001010 00011110 00110000 00010100 00001100
00011001 00001101 00011111 00010000 00001110 00000110 00000011 00011000
```

这里可以编写python代码来实现异或

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
a='01101100 01101111 01110110 01100101 01101100 01101111 01110110 01100101 01101100 01101111 01110110 01100101 0
1101100 01101111 01110110 01100101 01101100 01101111 01110110 01100101 01101100 01101111 01110110 01100101 01101
100 01101111 01110110 01100101 01101100 01101111 01110110 01100101'
b='00001010 00000011 00010111 00000010 01010110 00000001 00010101 00010001 00001010 00010100 00001110 00001010 0
0011110 00110000 00001110 00001010 00011110 00110000 00001110 00001010 00011110 00110000 00010100 00001100 00011
001 00001101 00011111 00010000 00001110 00000110 00000011 00011000'
c=[]
for i in range(len(a)):
 if a[i]== ' ':
  c.append(' ')
 if a[i]==b[i]:
  c.append('0')
 else:
  c.append('1')
print ''.join(c)
```

异或后的结果为

```
01100110 001101100 001100001 001100111 000111010 001101110 001100011 001110100
001100110 001111011 001111000 001101111 001110010 001011111 001111000 001101111
001110010 001011111 001111000 001101111 001110010 001011111 001100010 001101001
001110101 001100010 001101001 001110101 001100010 001101001 001110101 001111101
```

转换为ascii码为flag:nctf{xor_xor_xor_biubiubiu}
flag:nctf{xor_xor_xor_biubiubiu}

0x04 注意！！

flag{zhaowomen}

0x05 Wiener Wiener Chicken Dinner

0x06 Baby RSA

0x07 Classical

密文

```
nk gqsanez h yhxe ulj dklapdn e xhoaeu loylpneawyiyw
```

题目告诉是古典密码

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
a='nk gqsanez h yhxe ulj dklapdn e xhoaeu loylpneawyiyw'
import string

lowercase = string.ascii_lowercase

def substitution(text, key_table):
    text = text.lower()
    result = ''
    for l in text:
        i = lowercase.find(l)
        if i < 0:
            result += l
        else:
            result += key_table[i]
    return result

def caesar_cypher_encrypt(text, shift):
    key_table = lowercase[shift:] + lowercase[:shift]
    return substitution(text, key_table)

def caesar_cypher_decrypt(text, shift):
    return caesar_cypher_encrypt(text, -shift)

for i in range(0,25):
 print caesar_cypher_decrypt(a,i)
```