

BugKu CTF(解密篇Crypto)---道友不来算一算凶吉?

原创

肖萧然  于 2021-12-28 16:22:55 发布  856  收藏 2

分类专栏: [MyCTF # CRYPTO](#) 文章标签: [python](#) [crypto](#) [base64](#) [编码](#) [加密](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/qq_52549196/article/details/122195743

版权



[MyCTF 同时被 2 个专栏收录](#)

44 篇文章 1 订阅

订阅专栏



[CRYPTO](#)

13 篇文章 0 订阅

订阅专栏

BugKu CTF(解密篇Crypto)—道友不来算一算凶吉?

文章目录

[BugKu CTF\(解密篇Crypto\)---道友不来算一算凶吉?](#)

[题目](#)

[题解](#)

[编码方式](#)

[二进制转字符串](#)

[bsae64 解密](#)

[加密脚本4逆回](#)

[加密脚本5逆回](#)

[解出flag](#)

[总结](#)

题目

半仙我夜观天象，掐指一算，卜出卦象如下，不知道的有无道友可解此卦。

密文:升益艮归妹井萃旅离旅困未济屯未济中孚未济升困噬嗑鼎震巽噬嗑解节井萃离未济蒙归妹大畜无妄解兑临睽升睽未济无妄遁涣归妹

嗯? 为什么还有a和b呢?

a=5

b=7

```
# -- coding:UTF-8 --
from secret import flag

def encrypt5():
    enc=''
    for i in flag:
        enc+=chr((a*(ord(i)-97)+b)%26+97)
    return(enc)

def encrypt4():
    temp=''
    offset=5
    for i in range(len(enc)):
        temp+=chr(ord(enc[i])-offset-i)
    return(temp)
```

题解

编码方式

```
s = '升益艮归妹井萃旅离旅困未济屯未济中孚未济升困噬嗑鼎震巽噬嗑解节井萃离未济蒙归妹大畜无妄解兑临睽升睽未济无妄遁涣归妹'
dic = {'坤': '000000', '剥': '000001', '比': '000010', '观': '000011', '豫': '000100', '晋': '000101', '萃': '000110', '否': '000111', '谦': '001000', '艮': '001001', '蹇': '001010', '真': '100000', '颐': '100001', '屯': '100010', '益': '100011', '震': '100100', '噬嗑': '100101', '随': '100110', '无妄': '100111', '明夷': '101000', '贲': '101001'}
```

```
l = []
k = 0 # 两个字符的标志位
for i in range(len(s)):
    if k == 1:
        k = 0
        continue
    try:
        l.append(dic[s[i]])
    except:
        l.append(dic[s[i]+s[i+1]])
        k = 1

ss = ''.join(l)
print(ss)
```

```
0110001000110010011101000110100001100011011010011010110010101100101011000101101001010111001101010110000101101001010110110010011010001101011010101000
011000011010110001101010101100111001111001111010011110100
```

CSDN @肖萧然

易经有64卦 采用编码 000000 -> 1111111

坤: '000000'

剥: '000001'

比: '000010',

观: '000011'

...

‘乾’: ‘111111’

利用python的字典 依次替换密文

二进制转字符串

```
enc = ''
for i in range(0, len(ss), 8):
    enc += chr(eval('0b'+ss[i:i+8]))
print(enc)
[30] ✓ 0.3s
... b2thcm5ebW5XZlWdnU2hkUGNgS15cUg==
```

每8位->10进制->ascii字符

bsae64 解密

```
import base64
s=base64.b64decode(enc).decode()
print(s)
[4] ✓ 0.3s
... okarn^mnWeggShdPc`J^\R
```

加密脚本4逆回

```
def encrypt4(enc):
    temp = ''
    offset = 5
    for i in range(len(enc)):
        temp += chr(ord(enc[i])-offset-i)
    return(temp)
def decrypt4(enc):
    temp = ''
    offset = 5
    for i in range(len(enc)):
        temp += chr(ord(enc[i])+offset+i)
    return(temp)

str="qwert"
print(decrypt4(encrypt4(str)))
[32] ✓ 0.4s
... qwert
```

加密脚本5逆回

```
a,b=5,7
def encrypt5(flag):
    enc = ''
    for i in flag:
        enc += chr((a*(ord(i)-97)+b) % 26+97)
    return(enc)
def decrypt5(flag):
    enc = ''
    for i in flag:
        for k in range(20):
            if (ord(i) - 97 - b+26*k) % a == 0:
                enc += chr((ord(i) - 97 - b + 26 * k) // a + 97)
                break
    return(enc)

str = "qwert"
print(decrypt5(encrypt5(str)))
```

33] ✓ 0.3s
... qwert

CSDN @肖萧然

%的不确定,需用加 26*k 判断

解出flag

```
print(decrypt5(decrypt4(s)))
```

[35] ✓ 0.8s
... shaodayouxiduoduyijing

总结

```
import base64
s = '升益艮归妹井萃旅离困未济屯未济中孚未济升困噬嗑鼎震巽噬嗑解节井萃离未济蒙归妹大畜无妄解兑临睽升睽未济无妄遁涣归妹'
dic = {'坤': '000000', '剥': '000001', '比': '000010', '观': '000011', '豫': '000100', '晋': '000101', '萃': '000110', '否': '000111', '谦': '001000', '艮': '001001', '蹇': '001010', '渐': '001011', '小过': '001100', '旅': '001101', '咸': '001110', '遁': '001111', '师': '010000', '蒙': '010001', '坎': '010010', '涣': '010011', '解': '010100', '未济': '010101', '困': '010110', '讼': '010111', '升': '011000', '蛊': '011001', '井': '011010', '巽': '011011', '恒': '011100', '鼎': '011101', '大过': '011110', '姤': '011111', '复': '100000', '颐': '100001', '屯': '100010', '益': '100011', '震': '100100', '噬嗑': '100101', '随': '100110', '无妄': '100111', '明夷': '101000', '贲': '101001', '既济': '101010', '家人': '101011', '丰': '101100', '离': '101101', '革': '101110', '同人': '101111', '临': '110000', '损': '110001', '节': '110010', '中孚': '110011', '归妹': '110100', '睽': '110101', '兑': '110110', '履': '110111', '泰': '111000', '大畜': '111001', '需': '111010', '小畜': '111011', '大壮': '111100', '大有': '111101', '夬': '111110', '乾': '111111'}
l = []
k = 0 # 两个字符的标志位
for i in range(len(s)):
    if k == 1:
        k = 0
        continue
```

```

try:
    l.append(dic[s[i]])
except:
    l.append(dic[s[i]+s[i+1]])
    k = 1

ss = ''.join(l)

# print(ss)

enc = ''
for i in range(0, len(ss), 8):
    enc += chr(eval('0b'+ss[i:i+8]))

# print(enc)

s = base64.b64decode(enc).decode()

# print(s)

def encrypt4(enc):
    temp = ''
    offset = 5
    for i in range(len(enc)):
        temp += chr(ord(enc[i])-offset-i)
    return(temp)

def decrypt4(enc):
    temp = ''
    offset = 5
    for i in range(len(enc)):
        temp += chr(ord(enc[i])+offset+i)
    return(temp)

a, b = 5, 7

def encrpyt5(flag):
    enc = ''
    for i in flag:
        enc += chr((a*(ord(i)-97)+b) % 26+97)
    return(enc)

def decrypt5(flag):
    enc = ''
    for i in flag:
        for k in range(20):
            if (ord(i) - 97 - b+26*k) % a == 0:
                enc += chr((ord(i) - 97 - b + 26 * k) // a + 97)
                break
    return(enc)

print(decrypt5(decrypt4(s)))

```



