x-ctf新手区-crypto

原创

endinggy0 于 2021-07-29 17:35:52 发布 31 收藏

分类专栏： CTF 文章标签： CTF 加密

本文链接：https://blog.csdn.net/weixin_43856210/article/details/119216691

easy_RSA

```
p = 473398607161
q = 4511491
e = 17

def ex_gcd(a, b):
    if b == 0: return 1, 0, a
    else:
        x, y, gcd = ex_gcd(b, a % b) #递归直至余数等于0(需多递归一层用来判断)
        x, y = y, (x - (a // b) * y) #辗转相除法反向推导每层a、b的因子使得gcd(a,b)=ax+by成立
        return x, y, gcd

_, d, _ = ex_gcd((p-1) * (q - 1), e)
print(d)
```

转轮机加密

```
s = '''
< ZWAXJGDLUBVIQHKYPNTCRMOSFE <
< KPBELNACZDTRXMJQOYHGVSFUWI <
< BDMAIZVRNSJUWFHTEQGYXPLOCK <
< RPLNDVHGFCUKTEBSXQYIZMJWAO <
< IHFRLABEUOTSGJVDKCPMNZQWXY <
< AMKGHIWPNYCJBFZDRUSLOQXVET <
< GWTHSPYBXIZULVKMRAFDCEONJQ <
< NOZUTWDCVRJLXKISEFAPMYGHBQ <
< XPLTDSRFHENYVUBMCQWAOIKZGJ <
< UDNAJFBOWTGVRSCZQKELMXYIHP <
< MNBVCXZQWERTPOIUYALSKDJFHG <
< LVNCMXZPQOWEIURYTASBKJDFHG <
< JZQAWSXCDERFVBGTYHNUMKILOP <
'''
order = [2,3,7,5,13,12,9,1,8,10,4,11,6]
start = 'NFQKSEVOQOFNP'
s = s.replace('<', '').replace('>','').split()
t = []
for i in range(13):
    st = s[order[i]-1].find(start[i])
    t.append(s[order[i]-1][st:] + s[order[i]-1][:st])
ans = []
for j in range(len(t[0])):
    ans.append('')
    for i in range(len(t)):
        ans[j] += t[i][j].lower()
print(*ans)
```

在答案备选区找出那个 有意义字符串即可

easyChallenge

附件下载下来为pyc文件，在线反编译
得到如下代码

```python
#!/usr/bin/env python
# visit https://tool.lu/pyc/ for more information
import base64

def encode1(ans):
    s = ''
    for i in ans:
        x = ord(i) ^ 36
        x = x + 25
        s += chr(x)

    return s


def encode2(ans):
    s = ''
    for i in ans:
        x = ord(i) + 36
        x = x ^ 36
        s += chr(x)

    return s


def encode3(ans):
    return base64.b32encode(ans)

flag = ' '
print 'Please Input your flag:'
flag = raw_input()
final = 'UC7KOWVXWVNKNIC2XCXKHKK2W5NLBKNOUOSK3LNNVWW3E==='
if encode3(encode2(encode1(flag))) == final:
    print 'correct'
else:
    print 'wrong'
```

写出对于解码程序，注意注释部分

```python
import base64
def decode1(ans):
    s = ''
    for i in ans:
        x = ord(i) - 25
        x = x ^ 36
        s += chr(x)
    return s

def decode2(ans):
    s = ''
    for i in ans:
        x = i ^ 36 # 这里没有加ord，因为base32解码出来后已经为asc数字码
        x = x - 36
        s += chr(x)
    return s

def decode3(ans):
    return base64.b32decode(ans)

final = 'UC7KOWVXWVNKNIC2XCXKHKK2W5NLBKNOUOSK3LNNVWW3E==='
print(decode1(decode2(decode3(final))))
```

easy_ECC

椭圆曲线求公钥

```python
def ex_gcd(a, b):
    if b == 0:
        return 1, 0, a
    else:
        k = a // b
        remainder = a % b
        x1, y1, gcd = ex_gcd(b, remainder)
        x, y = y1, x1 - k * y1
        return x, y, gcd

def get_inverse(a, p):
    x, y, gcd = ex_gcd(a, p)
    return (x + p) % p

def ecc_add(P, Q):
    px, py = P
    qx, qy = Q
    flag = 0
    if P == Q:
        dy = 3 * px ** 2 + a
        dx = 2 * py
    else:
        dy = qy - py
        dx = qx - px
        if dy * dx < 0:
            flag = 1
            dy, dx = abs(dy), abs(dx)
    _, _, d = ex_gcd(dx, dy)
    dx, dy = dx // d, dy // d
    dx_inverse, _, _ = ex_gcd(dx, p)
    k = dy * dx_inverse
    if flag:
        k = (-k + p) % p
    rx = (k ** 2 - px - qx) % p
    ry = (k * (px - rx) - py) % p
    return [rx, ry]

def mul_k(P, k):
    if k == 1:
        return P
    mid = k // 2
    midP = mul_k(P, mid)
    R = ecc_add(midP, midP)
    if k & 1:
        R = ecc_add(R, P)
    return R


p, a, b = 15424654874903, 16546484, 4548674875
G = [6478678675,5636379357093]
k = 546768
print(sum(mul_k(G, k)))
```