20211030-第四届浙江省大学生网络与信息安全竞赛（决赛）-Crypto方向WP

原创

4XWi11 于 2021-11-02 21:50:25 发布 262 收藏 2

分类专栏： 树哥让我天天写之Crypto 文章标签： 网络 python 开发语言

本文链接：https://blog.csdn.net/m0_49109277/article/details/121110182

树哥让我天天写之Crypto 专栏收录该内容

21 篇文章 5 订阅

订阅专栏

Crypto

decode_and_decode

前不久buu刷题刚做过，写了个脚本

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
from base64 import *

with open("decode_and_decode.txt", "r+") as fp:
    cipher = fp.read()
    while 1:
        print(cipher)
        try:
            cipher = b16decode(cipher)
        except:
            try:
                cipher = b32decode(cipher)
                assert str(cipher).isprintable() and str(cipher).isascii()
            except:
                try:
                    cipher = b64decode(cipher)
                except:
                    break
    print(cipher)
```

> DASCTF{9f128fe14dfa30d90fd922393e6bbdfe}

dssssa1

这个类型的dsa在ctfshow一次比赛上遇到过

但这个明显太垃了，这种题有200分啊，大跌眼镜，就是简单推导。。。

```
s = (h + x*r) * invert(k, q) % q
```

啥都知道了，求个x还不简单

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
from Crypto.Util.number import *
from gmpy2 import *

p, q, g, y, h, r, s, c, k = 945150402202630978758725416680714706194357073582117165622199173317977674880220530872675665867099447853297085715591266403396093751663859041471899500356309104045346426221148046358563149284385315445532364582442256986948466073332267044679320797125159716156438682092814604296298809205504691704499352954546292933993, 100153551413699469552963612831121230125032676786 9, 892887002251716765997597741841467983211917487397032463955290019799884013038000660446744498341840956677478989743754317005038001428408991944921820578856751476816002179794667196929848633302983477426574729365590419304897021162559994124489967149231128242442679108087827944428955186858641748175010400606809624479419, 938875286953602925248138142401903287322836632554268061281979577206744962600607035959336760828822047245010856334249425823047073954492220433288958528125435764185677167818701796060498995404497290367712905506457709786670758210437975692557872719325562180149203734628823298025976720268065524177356605531443446506422, 77559352130513427596747225421840126470316613881 7, 75084117510316201869105133948164969652170742276, 5994170044542088258848655292814537743240931348 27, 94203926294365722030261882520165826558476099177297861176153811285238289485953276649563642144753132730431066372867407177248194182778827143183520415437355921352580608448713381897280433120409711633310458263502217605470824497215111936036532237050330222480782799188409969149722885261258984444311562364318406725475829089368796269160936194172040318140462371217663, 20867245776787730389532722202098296393177912381 9
n = p * q
phi = (p - 1) * (q - 1)

x = ((s * k - h) * invert(r, q)) % q
print(x.bit_length())
d = invert(x, phi)
print(long_to_bytes(pow(c, d, n)))
```

easyNTRU

用la佬博客上的脚本跑，果然不行，报错多项式没有逆元

似乎可以爆破。。。

```
#!/usr/bin/env sage
# -*- coding: utf-8 -*-
from Crypto.Hash import SHA3_256
from Crypto.Cipher import AES
import sys
from Crypto.Util.Padding import unpad

N = 10
p = 3
q = 512
d = 3

R.<x> = ZZ[]

c = b'\xb9W\x8c\x8b\x0cG\xde\x7fl\xf7\x03\xbb9m\x0c\xc4L\xfe\xe9Q\xad\xfd\xda!\x1a\xea@}U\x9ay4\x8a\xe3y\xdf\xd5
BV\xa7\x06\xf9\x08\x96="f\xc1\x1b\xd7\xdb\xc1j\x82F\x0b\x16\x06\xbcJMB\xc8\x80'

table = [-1, 0, 1]
for i1 in table:
    for i2 in table:
        for i3 in table:
            for i4 in table:
                for i5 in table:
                    for i6 in table:
                        for i7 in table:
                            for i8 in table:
                                for i9 in table:
                                    for i10 in table:
                                        result = [i1, i2, i3, i4, i5, i6, i7, i8, i9, i10]
                                        m = R(result)
                                        sha3 = SHA3_256.new()
                                        key = sha3.update(bytes(str(m).encode('utf-8'))).digest()
                                        dypher = AES.new(key, AES.MODE_ECB)
                                        try:
                                            flag = unpad(dypher.decrypt(c), 32)
                                            if flag.startswith(b'flag') or flag.startswith(b'DASCTF'):
                                                print(flag)
                                                sys.exit(0)
                                        except:
                                            pass
```

Re

RE人三项

看主逻辑

```
__int64 __fastcall main(int a1, char **a2, char **a3)
{
  v11 = __readfsqword(0x28u);
  memset(plaintext, 0, sizeof(plaintext));
  memset(cipher1, 0, sizeof(cipher1));
  memset(cipher2, 0, sizeof(cipher2));
  memset(cipher3, 0, 64uLL);
  printf("Input >");
  scanf("%64s", plaintext);
  if ( divide(plaintext, cipher1, cipher2, cipher3) < 0 )// 按_分隔成三段cipher
    quit();
  length_cipher = strlen(cipher1);
  if ( encode1(cipher1, length_cipher) < 0 )
    quit();
  v4 = strlen(cipher2);
  if ( encode2(cipher2, v4) < 0 )
    quit();
  v5 = strlen(cipher3);
  if ( encode3(cipher3, v5) < 0 )
    quit();
  puts("Check completed:)\nYou got it!");
  return 0LL;
}
```

按照下划线把flag大括号里的分成三份分别加密

第一个直接看出来，出来是`R0t13`，哦ROT13

第三个很神奇的东西，不知道是什么加密，按照他的做的写逆就好，不会存在不确定的路径

第二个，分析了半天没看出什么名堂，猜应该是一个对称加密，然后和misc兼逆向手讨论，还好他之前记过常见加密的sbox

SM4的沙盒

```
sbox = [0xD6, 0x90, 0xE9, 0xFE, 0xCC, 0xE1, 0x3D, 0xB7, 0x16, 0xB6,
        0x14, 0xC2, 0x28, 0xFB, 0x2C, 0x05, 0x2B, 0x67, 0x9A, 0x76,
        0x2A, 0xBE, 0x04, 0xC3, 0xAA, 0x44, 0x13, 0x26, 0x49, 0x86,
        0x06, 0x99, 0x9C, 0x42, 0x50, 0xF4, 0x91, 0xEF, 0x98, 0x7A,
        0x33, 0x54, 0x0B, 0x43, 0xED, 0xCF, 0xAC, 0x62, 0xE4, 0xB3,
        0x1C, 0xA9, 0xC9, 0x08, 0xE8, 0x95, 0x80, 0xDF, 0x94, 0xFA,
        0x75, 0x8F, 0x3F, 0xA6, 0x47, 0x07, 0xA7, 0xFC, 0xF3, 0x73,
        0x17, 0xBA, 0x83, 0x59, 0x3C, 0x19, 0xE6, 0x85, 0x4F, 0xA8,
        0x68, 0x6B, 0x81, 0xB2, 0x71, 0x64, 0xDA, 0x8B, 0xF8, 0xEB,
        0x0F, 0x4B, 0x70, 0x56, 0x9D, 0x35, 0x1E, 0x24, 0x0E, 0x5E,
        0x63, 0x58, 0xD1, 0xA2, 0x25, 0x22, 0x7C, 0x3B, 0x01, 0x21,
        0x78, 0x87, 0xD4, 0x00, 0x46, 0x57, 0x9F, 0xD3, 0x27, 0x52,
        0x4C, 0x36, 0x02, 0xE7, 0xA0, 0xC4, 0xC8, 0x9E, 0xEA, 0xBF,
        0x8A, 0xD2, 0x40, 0xC7, 0x38, 0xB5, 0xA3, 0xF7, 0xF2, 0xCE,
        0xF9, 0x61, 0x15, 0xA1, 0xE0, 0xAE, 0x5D, 0xA4, 0x9B, 0x34,
        0x1A, 0x55, 0xAD, 0x93, 0x32, 0x30, 0xF5, 0x8C, 0xB1, 0xE3,
        0x1D, 0xF6, 0xE2, 0x2E, 0x82, 0x66, 0xCA, 0x60, 0xC0, 0x29,
        0x23, 0xAB, 0x0D, 0x53, 0x4E, 0x6F, 0xD5, 0xDB, 0x37, 0x45,
        0xDE, 0xFD, 0x8E, 0x2F, 0x03, 0xFF, 0x6A, 0x72, 0x6D, 0x6C,
        0x5B, 0x51, 0x8D, 0x1B, 0xAF, 0x92, 0xBB, 0xDD, 0xBC, 0x7F,
        0x11, 0xD9, 0x5C, 0x41, 0x1F, 0x10, 0x5A, 0xD8, 0x0A, 0xC1,
        0x31, 0x88, 0xA5, 0xCD, 0x7B, 0xBD, 0x2D, 0x74, 0xD0, 0x12,
        0xB8, 0xE5, 0xB4, 0xB0, 0x89, 0x69, 0x97, 0x4A, 0x0C, 0x96,
        0x77, 0x7E, 0x65, 0xB9, 0xF1, 0x09, 0xC5, 0x6E, 0xC6, 0x84,
        0x18, 0xF0, 0x7D, 0xEC, 0x3A, 0xDC, 0x4D, 0x20, 0x79, 0xEE,
        0x5F, 0x3E, 0xD7, 0xCB, 0x39, 0x48]
```

然后预赛就把la佬的博客给保存在本地了，拿脚本直接跑

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
from SM4Cipher import SM4Cipher

cipher1 = 'E0gy3'
space1 = 'NOPQRSTUVWXYZABCDEFGHIJKLMnopqrstuvwxyzabcdefghijklm'
plain1 = ''
for i in cipher1:
    if 'A' <= i < 'Z':
        plain1 += space1[ord(i) - 65]
    elif 'a' <= i <= 'z':
        plain1 += space1[ord(i) - 71]
    else:
        plain1 += i

cipher2 = [0xF2, 0x73, 0x52, 0xFB, 0x8D, 0xF4, 0x1D, 0x6D, 0xC2, 0x33,
           0xB5, 0xA5, 0xEE, 0xC1, 0x60, 0xDA]
key = [b'' for _ in range(16)]
key[0] = b'\x01'
key[1] = b'#'
key[2] = b'E'
key[3] = b'g'
key[4] = b'\x89'
key[5] = b'\xAB'
key[6] = b'\xCD'
key[7] = b'\xEF'
key[8] = b'\xFE'
key[9] = b'\xDC'
key[10] = b'\xBA'
key[11] = b'\x98'
key[12] = b'v'
```

```
key[12] = b'v'
key[13] = b'T'
key[14] = b'2'
key[15] = b'\x10'

key = b''.join(key)
plaintext = bytes(cipher2)
sm4 = SM4Cipher(key)
plain2 = sm4.decrypt(plaintext).decode()

check_box = [7856030,
             7856049,
             7856004,
             7856004,
             7856854,
             7856004,
             7856045,
             7856004,
             7856050,
             7856004,
             7856004,
             7856004,
             7856024,
             7856004,
             7856004,
             7856060,
             7856004,
             7856004,
             7856234,
             7856004,
             7856004,
             7856004,
             7856100,
             7856151,
             7856230,
             7856233,
             7856004,
             7856004,
             7856235,
             7856004,
             7856004,
             7856004,
             7856345,
             7856004,
             7856004,
             7856004,
             7856004,
             7856104,
             7856004,
             7856565,
             7856666,
             7856785,
             7856004,
             7856004,
             7856004,
             7856004,
             7856004,
             7856004,
             7857004,
             0,
```

```
            0,
            0]
trans = [1, 7, -1, -7]
start = 7856030  # 1
target = 7857004
assert 48 == check_box.index(target)
c1 = 'a'
c2 = 'b'
mid = check_box[1]
index = 1
plain3 = c1
while 1:
    if check_box[index + 1] > mid >= check_box[index + 7]:
        index = index + 1
        mid = check_box[index]
        plain3 += c1
    elif check_box[index + 7] > mid >= check_box[index + 1]:
        index = index + 7
        mid = check_box[index]
        plain3 += c2
    else:
        print('this seems not to happen')
    if mid == target:
        break

flag = 'ZJCTF{' + plain1 + '_' + plain2 + '_' + plain3 + '}'
print(flag)
# ZJCTF{R0tl3_Sm34@and_abbbaaabbaab}
```

把奇怪的东西去掉