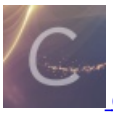
2021 蓝帽杯半决赛 write up

原创

ljahum 于 2021-06-06 17:45:54 发布

分类专栏： ctf

本文链接：https://blog.csdn.net/a_touhouer/article/details/117632700



蓝□ber 2021半决赛

> 很气，为什么这傻逼玩意3点结束
> 中文12点钟还在上机做实验 □□□
> 笑死 吃个饭根本没有时间写脚本

very smooth

> 白给题

连上后有一个 Hints 是一个特殊素数的题

可以对任意明文用 E，N 加密,上传4,8,16,0x20利用选择明文的方式得到 n

把n分解掉得到pq

```
from gmpy2 import gcd
from Crypto.Util.number import inverse, long_to_bytes, sieve_base as primes
c2 = 0x42a03c9a532106552a1517f833746c75951b9daebc0fd66b616f54622b44f3aec6227a3b4f02f4e77ae98209fa4c52c43596a44468a336d956d69be9588e6544c98313c9e8b4083cc3165102c7704834c66e165971419e17717f98eb0e494c71c498e7c7432a4753329912ec2a60a128e04c93e28c5c253e6ee17c3376f0712fb3b7ae72715f76366eeeb207a5367db8e2f43a5ad33459014bef0d3c225bd11768124a75a54e7c910151cd314ae7bbc4f86fbf6a5f2df7cda7467596bb415d2165ab282c38b2dccd736c3c5f2fea9cf64f417b6340f7d59069f739aa538b6e074fd149a016fc446439a8c2b59d306f748e9abd21a15947f3b31d9cac9c932e9c4d9e639247eaa2345651bf8d741943904c33316b5ebff740f609605ccb0d0069a0e45d230a6af1d12904d998d540387f89529cf8a5c6b491c7bd190ce98120adeab6c6400a735369234655d20dbd1f86213da6bd2ffc07f8c4debc39b33954dc38ac1d897bd58f585f892d35fa9367478f44bc25ac809d3a336aa0a9af3bc7f2661c06cbd414b1a99e1058593a7415135464a10fd0cffa2a5d50bbff9478825075371833996f5f780389b7e466a9545cfbb76fb4cff56addab40c4c043295752a4dbf085cd57e32a44d88c36bd8c71ab55f835a0f54d47db7b1cd9b38dffe4cb2076c7fa38d47ad791e97220c75d795009f933de26d18f19439cb89b1c6eee9b921155
c4 = 0x4e0fc3b860cc8b893eb63d23cfb68ac9750e0519e54949f45e5c230d53443758a7f5187fa111399edad5b82d01546cafc756738c4ba8c80f7f5ccd3f0401ce0d494ee6988fbf9995434a35e70ad5271d18f0790dfb75930229271l6a0c2dcdea61e25eeabc67da1b22bc0c2f44be8d35da70af6f78696302d5e78e838668eb22240bc8916c2ed9b90c48ba9bc2d610e9019efba6c8a1f63ed204fbed82954582671b7ca8be91411c0ef4dbc918d913ed80a234ca448b298366fcb6dc241ed3d7744cb3bf5d24811a34875a346bfb37e35f3a77cc5f96aa65863ff7aac97142b2d1eb2b73a72a7918b87d99cfd5f3e7adac18c5016e4e41c96d0110570f7a937b3a21d7e94ea0b407b58d87893903b10e6a3088e3a07426e8d41dd197af71ac855e9a6a6e060556b853e2b40aad8d2581732163ca76c938e687f37c30ac48b3dda19e8814c10f852e71421cae9bdd795e0c7a9afb34261900236c586e59469f0cecd9cdfc81ebeeb0bc6e2f19f6dfb949ee00bd4ed8a1a553dc4a145ae415e39aec7f56ad2c5ef1a33f943b126bc8b67fbb632b01eda6e06ce0f847e32218dcfa5f70687e55e8d7d844308d61311077c5cd4b022c42d6bbf0b436542093bf60cd0ce87b7c9bcb3517ae9d0e51d1a2e05cd40a97b469d127714f3c6dd0183956357db25be4c5443ea01aac72ba583d0cd0a72ad4f6f
```

```
a50d8ace2bbe87f9ff11dd293539
c8 = 0x187adec8ba464a1a8f911caaabd95a519fc641729d0bbaebd6733362313d718d783961d4db0772de7e1f53c0971b630007554d3f82ca74a6879a08b1572688306dc5759182f20009c4c8cd645eba15e8e1ecb90c7b332685eaeeea0fc5e0f358880c78fcee3aed48ffc28798c553ad43ffc09adcf8797a0177b98ee692d32fd5564dadee8756192685a325012aa460802197bfb04118e1444708403d75fec3888097c3289de53b2491323d4c0f11b0cd6242192f8a40fef3dae1cbdc0a70a2cbca2c746ccf40aa261e87c9a22f2205375c88098dc20d9d5093d9a1b79e4dcf3ff8029a62b4391e16c7c106b05def5ccf1b22565d0ce2deb3eb1b4943ca28999b86e815566f55d2a7a0c8c341adcfea7d608690c217c9128dc41c708c0b3830159947618a1860d11e5bb376b14e20017972116da12364cc4ba87a0fc33a02f55fab863ac5f841915c6ec6301cc56ca8c4fe7a37031df7cea4bedec32ca4d24832772aff2363d4ba6c943ce67bad23b4a643ac24e20cfaaada364d1b389409b2f9fd52632cd8b80a9264ef9e637947f80521d95a4c173649e05f6fb5c5ad350293e08e2215cfab6a7dbc7901a112f59209dd030219665438b708f7fcd7add7d9b1ebc4fc1f9abb6bcb3bfcaa757dd6d6853990371ec788ca5a5885fc979e0339f6d63fddff99472d09a1ff2a055add58ff8b3b4302ba0cc0f330c107e084c7bae53af298
c16 = 0x96e4bd1a084c122aee1d8547b5eb8cff24a3354ea768dfcc99457a62a7ef2b192bef7b032d046cc95a7171f24273bb84bf82f3640eec73f36a1089c5dd24006c96767f1c3753c686a53e37d9f60708649c9ff46d03117acd5318cd1fe266f26aa8a239c99c7339b5dc3c73c075feacfd7cc70b2324fc3d911a4839ff70f6ad900b57d30115750f6e1604e584f8308ed9f130d5be6e731db98cb3568fea1d941fc1b721888053f632ac39b14a1bdeb273a92ce3b2ade717a25126747ec5dd37397b39e9e7b275d343c52f772eac1a6b1b11b3afd4f39cb061946090c65debeb9c67fc395f07229f7747801252d3190afa627a7b9dd07e01b95d9b18e0cacdb7d6bdd855449139e56ae28b8fe38a37adef071811f14a4670c940bc8ea64a58bac4bb0693989501181fbd0ec225d288c8ea21849f2969b16d585a271d3cf9ad4e15ec720c3e24954ef2228e2d91dcd3c5e3765da779344ddc6798ed7d46ed29969fb8e1218830a2dda369adae57531d29c65a4a6b6d232034b047e2af76ccfd966e5106123000b248d376e44da1dc32a359d9d8ecc39d2ad48034dbdd5b5a35270da5e42749b9c15ae67e0c46d325efc1c0afb22df74910bad5388a21fe25c31137c6a62cb68e5ddd6754aa9f5961567f89f20ac240d755dab069acdfe3d16b2e83c13f5b58e6b77e132c957ca5eba6b1f9ac83f8f3d81278e4581c17973b8b2e8a85da96
c32 = 0x35ca1593e142b2847125b555abf2a7671d6453932a90752e4a5e14740b0b2d9150ff2a1e26245ba869916304ac8ff448647f03012a0d0ef3530cfbe381bae2c756a3191100f327c9e1254395a79ae38e180a0f06865976c30ad39e8ea0dd0279ef6e3d2e4585271cd1f60befdb53d7e93e368d17453962b60a81fddd5a0842513fabd3553cd64c9a2c6ab7a590c7dbe62ffbad7a7732b344dddfd599cc76249c9c9cd9fc5535d0e5f843516675a90e19c78cac4150035192c3b672f1a46ff1cfb707dc0c9faa49f43526ae52dfaef2cd7248b28367407828 56f1b591e45c165fc5ac198809a6622712a560b605a4ec3d9e6d82a383dd05e5fdbf4b79a6cc2c63a651311876a62304cc184e481548b05ef0a7f29b675355c4217c3f33e8a1904d76a8d89f9461feae3412fc6c16430627ff0709e1b44b754bcd1fb823a74cae92775fdc472755dc50c252685d6c8a9c1358f354e705dd69fbc7fb2bdea38e80cb6d89218918272f4b61daf155d8a7ab06a70be70188669c905ea9ff7b1edf8a1042b9fc8dfc21075e1c24b4300b80e17b8f61cb10a6b9de72dc1bdb45b47e90eb58ae89997f3f53373b8ddad181987b6be6f8945396eb9aec245ca5c13258e2c7d1edfe983d9e613036a010c8c85ac2d369ad9021b3eefd67f88482505625d907f0db8ac3970ea1c5a4cbc8194ac6cd0ba669b5635c3e7cc85eb89db38a86f7b569a7d7
e = 0xb0c9850f9011fff3

tmp1 = c2**2-c4
tmp2 = c2**3 - c8
print(gcd(tmp1,tmp2)) #n
n = 1069266467989373906134662496375623832690983298691174667143217797710680374783480848266474506650376765934229798539847367293478421415408398270137889283362495994938159321191942094900371349033471196470187617407549099823863960387593259193402721807127852699519668853280981628261316487943905406439579310354041849260102881975081789823121551549078134236266721752161100648646047002177435536398666029700548614023534767114735734728842955157080237390231435131520369196387124838023859155921311405661437123537163206073379935370503240829440588955485947613948332227393196459287107734320191811629691504913439770999752444370901983288422080248662214812798727634184210085923357959776683107806159173515534814166502315198062692151238851367629977909120419333922098779960273196827432805695810804644448419704458787657157664816190509952640675761876938451649406354980509238332895321401819512233216473056374066424570572281850681500875169156946555095186266069125873937021378187854248743589166656651441416054328239659696943069185393588780282804122490908644746220820624405856931133646945569355076346202323269661710103100230278384651780655834096829188599605881363959479501946474618663484908895626195051934397657098132392615982188560852333473406889360550952399280652020005101 3
c2 = pow(2,e,n)
print(hex(c2))

# ========================================
q = 2296069003095403164374287394075638780506363237434979818448615867905679396832608235374481085968115700438676488821622659005532126272533352501816294632936330561913873163062288816595597754987621171427244188617015168465430856901506578709435239884734778566364317799667136801466713079834365598781155920574708036473634111489003229336842258022360650135982205859675908609351198782471552494015517629418751155817239988867885813008667319940572463221182372626067125346430549598365895607131659411030293541450719845300640987644827479731356448902777744101545473782565956652067922354638486501746853731055164466358221991032884289460121898 7
p = 4656943961822845909947156251312307141614170417259526998787367813450120321742290774331771275992123188393528606983575473411084102178734538600418777071769880348421631293409372094513883764853119611839114557656141629257346772
```

```
675859394123230190399336184862639319722473439361147124807710590705557371468353610523561666346841797509798332791303691938188630856419454697594136024786170233057930420621273742702383985931522538257716450138083733168242287677389105139830826257506954799789107796369173384246933719603462395092527898523947674801184286663111192751334650467959345730381847683969111873522625097447124658228716050903353222099990
print(n-p*q)

c = 0x17bb4730ef3ba7591d5ac5aba56596ea7bb5b3cac908375fed01f827a4ad246457bcf4f67be416126bc421b2ec813aca9ebed52c36734be35fd39fd450bd11831f833053743c7822094bf295d17984380d062764f31c8e8725255eec779d375ec82b76b6d8956107264fd9550d0f7407ad6531bbbc79af22cb85191171e0a79c137f180d9376cce93e9a17fb1d7ba985150f73b8d67be7e47c72bf607aa274369aaaa230078eca1bf4daa0326d46b682b1d94c651946417f7f254c1b6531df833deca73b81d9cf026c5038c8021d31fee10efb4a1f041d9180b7b489a9ce0ea12f82c67529c9088f72e84de35aa48ebfd67e7a8fcaf01b07a937bbeeab578710495abcc80f8a2a57a84d5b76e54cd58d160f6e289bf4848b9b87c7b94164fd11b3dca467074a71e02ed9ea72db34915e1276c48339ef6f4c4a9e971e69041117b6a1f2502a475dfbba83b1c61d1b7e427c0de5132f82b73b0ba839bfe4ad93cb8768596d67aa2dfb873a7589d38a89cae70203e9fdbcd2497adf563480ad03163c96be553ba5f72ad2d57358acd9ebeeb00b2a023f5a9406b5549082a618e43773b903d501bd6aab795362dc49024c2a6a17ba33d13731aea574941766dd1a52c3f3d48f9e26901d4e1c3031ba1cd63c5078bc51d3b0f310236d3d2b9e8a9fa4fa01567293e02ef22dea9d3744924f46e5f16f25ad800f3fa93a5f4c4b58c68bbac1
e = 0xb0c9850f9011fff3
phi = (p-1)*(q-1)
d = inverse(e,phi)
m = pow(c,d,n)
print(long_to_bytes(m))
#  flag{984af69b-5497-471b-9fa3-cda517490ad2}
```

sharing_system

简单数学题，做不出来的建议重读初一.□□□□

用 keys1 给的 y 互相相减得到没有secret的方程

构造矩阵得到ts,带入公式得到k1 k2

exp:

```
from itertools import product
from pwn import *
from icecream import *
from hashlib import sha256
r = remote('0.0.0.0',20001)
from time import *

def gopow():
    # print(r.recv(1024))
    r.recvuntil('sha256(XXXX+')
    s1 = r.recvuntil(') == ')[:-5]
    hashstr = r.recvline()[:-1]
    print(r.recvuntil('Give me XXXX >'))
    print(s1,hashstr)
    tab = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'
    for i in product(tab,repeat=4):
        tmp = ''.join(i)
        s0 =  bytes(tmp,encoding='utf-8')
        s = s0+s1
        hash_value = sha256(s).hexdigest()
        hash_value = bytes(hash_value,encoding='utf-8')
        if hash_value == hashstr:
            print(hash_value)
            ic("XXXX=",s0)
            ic(hash_value,hashstr)
            print(r.recv(2048))
            r.sendline(s0)
```

```
            r.sendline(s0)
            break

gopow()


buf = r.recv(1024)
print(buf)
buf = r.recvuntil('Enter option > ')
print(buf)
r.sendline('1')

buf = r.recvline() # p
print(buf)
p = int(buf[3:-1])
print(p)

buf = r.recvuntil('key = (') # key1
x1 = r.recvuntil(', ')
y1 = r.recvuntil(')\n')
print(buf,x1,y1)

buf = r.recvuntil('Enter option > ')
print(buf)
r.sendline('2')

buf = r.recvline() # p
# print(buf)
p = int(buf[3:-1])
# print(p)

buf = r.recvuntil('key = (') # key1
x2 = r.recvuntil(', ')
y2 = r.recvuntil(')\n')
print(buf,x1,y1)

x1 = int(x1[:-2].decode())
y1 = int(y1[:-2].decode())
x2 = int(x2[:-2].decode())
y2 = int(y2[:-2].decode())
ic(x1)
ic(x2)
ic(y1)
ic(y2)
ic(p)
keys1 = [[x1,y1]]
keys2 = [[x2,y2]]
def getXY():
    for i in range(1,50):
        # print(r.recvuntil('Enter option > ').decode())
        r.sendline('3')
        r.recvuntil('umber (1-49) > ')
        r.sendline(str(i))
        r.recvuntil('key = (')
        x = int(r.recvuntil(', ')[:-2])
        y = int(r.recvuntil(')\n')[:-2])
        # print(x,y)
        keys1.append([x,y])
```

```python
def getts(key):
    keys_1 = key
    XS=[]
    for i in range(49):
        xi = keys_1[i][0]
        xi_1 = keys_1[i+1][0]
        xs=[]
        for j in range(49):
            x1 = pow(xi,j+1,p)
            x2 = pow(xi_1,j+1,p)
            tmp = (x1-x2)%p
            xs.append(tmp)
        XS.append(xs)

    YS =[]
    for i in range(49):
        yi = keys_1[i][1]
        yi_1 = keys_1[i+1][1]
        YS.append([(yi-yi_1)%p])
    # print(YS)
    X = Matrix(Zmod(p),XS)
    Y = Matrix(Zmod(p),YS)
    invx = X.inverse()
    T = invx*Y
    # print(ts)

    TS = [i[0] for i in T]

    return TS


getXY()
print(keys1)
print(len(keys1))

# t = 58510083877094693891040277851267905853617844771064576929521712771940231668984770883328421254238895212042598002659174059244201616561130950785238401399837559644941651285130352373844849078590789578028528768860166902952495931639692465443534927973027133719331474369633030490640443060779278904144461746302422307614
# ts = [ t for _ in range(50 - 1)]
ts = getts(keys1)

tmp1=0
for i in range(0,49):
    XS = pow(x1,i+1,p)*ts[i]
    tmp1 = (tmp1 + XS)%p
k1 = (y1 - tmp1)%p

tmp2=0
for i in range(0,49):
    XS = pow(x2,i+1,p)*ts[i]
    tmp2 = (tmp2 + XS)%p
k2 = (y2 - tmp2)%p

ic(k1,k2)
# print(r.recv(1024))
print(r.recvuntil('Enter option > '))
r.sendline('5')
```

```
print(r.recvuntil('Please enter secret 1 > '))
r.sendline(str(k1))
print(r.recvuntil('Please enter secret 2 > '))
r.sendline(str(k2))
sleep(1)
print(r.recv(1024))
```

```
b'Please enter secret 1 > '
b'Please enter secret 2 > '
b'Wow! How smart you are! Here is your flag: \nflag{0c10bc45-cb3b-4648-b11d-d8aa85f5e63b}\nBye!\n\n'
[*] Closed connection to 118.190.62.234 port 52863
```