CTF-RSA-低指数广播攻击-给出几组N和c，求m

原创

Flemington、 于 2020-04-13 14:51:25 发布 2831 收藏 2

分类专栏： Python 文章标签： CTF RSA Crypto

本文链接：https://blog.csdn.net/Everywhere_wwx/article/details/105488703

crypto 同时被 2 个专栏收录

5 篇文章 0 订阅

订阅专栏

Python

23 篇文章 0 订阅

订阅专栏

RSA-低指数广播攻击-给出几组N和c，求m

低指数广播攻击， 即用相同的公钥加密相同的消息，但每一组的n不同，e是一个很小的数，例如3或者10这个题目给出的是5进制数字，需要先处理一下

题目打开是RSA.txt，三组N与C，且都为8进制。

```
import gmpy2
import gmpy
import libnum
from Crypto.Util.number import long_to_bytes

n1 = 13373530262553241406216361271702271204405557676245503750760573165333235422300235267660763331704251025201543
34467517732411315170476671165724742354160402251270554440057761705537572156141533555772422640756663126004200515740
60004120450161126774437214316201343616247763272033777642316604135227334334401650503563600242255561155566770157200
37222426225
c1 = 12025126641660246361576105403421512244715435470364744600164021475461052014474170454442246537023733627523413
63322451753100151750503067342720065033313445131063222541322153325635131007242515762532551275707560660526675655055
56774631730565575734000633611067034462143366466652354215414725034403527364351071334764741116435071467341675513131
46071223345

n2 = 11563273742153302172253730146037041367361227730351231645455155774420706757040435124714570746364331061175106
72665023657734404674203526003646400117767742641676755531544546252161055274731457260043033712072713376714062731204
14111014321027447361362117156404152641754742754330100724657420750151372057371002012171735642344740456273100550200
35015207117
c2 = 40402352220647352767163676715126257415777254577110016277763447116300617514640010212232326414016431434003045
62322307351170002202303413440565454215372516114524047430232633761424161727700613410424012531666764205056413336616
66435612030462431754232214251737333673565636400737530145002577173035167575113163200526057400454467455546530615642
446134465

n3 = 13454356570615751313225762774515127535070515141673570055217071740253555515743574050255606767022773567711630
71245132443022416732571717711522571502404322121602724320217107211260063650452020733301133761220634216130773505477
75363334263563654673414676675713703535540215541151145527265461016014152667552227545734321754465462732603337574775
54152473413
c3 = 50547551057315710375065243304631043730311304536720407603265753772355343153271112143375506113574650653545136
26440516372271170646600111734220124344045372274502165205634534117244010273013102756227700511075455347574261554022
```

```
334430245162064362615751742036161156117552372612601673341773402670005256205023137146742517362253764247306723407127031772
```

```python
def boradcast_fuzz(question, e):
    N = 1
    for i in range(len(question)):
        N *= question[i]['n']
    N_list = []
    for i in range(len(question)):
        N_list.append(N / question[i]['n'])
    t_list = []
    for i in range(len(question)):
        t_list.append(int(gmpy2.invert(N_list[i], question[i]['n'])))
    sum = 0
    for i in range(len(question)):
        sum = (sum + question[i]['c'] * t_list[i] * N_list[i]) % N
    sum = gmpy.root(sum, e)[0]
    # return libnum.n2s(sum)
    return long_to_bytes(sum)


n1 = int(str(n1), 8)
n2 = int(str(n2), 8)
n3 = int(str(n3), 8)
c1 = int(str(c1), 8)
c2 = int(str(c2), 8)
c3 = int(str(c3), 8)

question = [
    {'n': n1, 'c': c1},
    {'n': n2, 'c': c2},
    {'n': n3, 'c': c3},
]

for i in range(2, 20):
    res = boradcast_fuzz(question, i)
    if 'noxCTF' in res:
        print res
        print 'e=%d' % (i)
        break
```

运行，即可得flag及e。