Crypto

RSA Attack

百度的大数分解网站分解 n，再百度一个脚本~~【失去百度的话我大概就无了】~~。

```
from libnum import n2s

e = 65537
n = 700612512827159827368074182577656505408114629807
c = 122622425510870177715177368049049966519567512708
p = 715800347513314032483037
q = 978782023871716954857211

def egcd(a, b):
    if a == 0:
        return (b, 0, 1)
    else:
        g, y, x = egcd(b % a, a)
        return (g, x - (b // a) * y, y)


def modinv(a, m):
    g, x, y = egcd(a, m)
    if g != 1:
        raise Exception('modular inverse does not exist')
    else:
        return x % m


d = modinv(e, (p - 1) * (q - 1))
m = pow(c, int(d), n)
flag = n2s(m).decode('ascii', errors='ignore')
print(flag)
```

Chinese Character Encryption

得到第一个 hint 的时候就猜测是用拼音的 ascii 的和对128取余然后再根据 ascii 转换成字符，但是没有考虑到声调，后来把声调考虑进来就对了。

```
import pypinyin


def s2n(c):
    a = 0
 for i in c:
        a = a + ord(i)
    return a


def n2s(c):
    a = c % 128
 return chr(a)

text = '陉莕倲蘭貑謠祥彐朔髯篢凰葴秉僦笆猒雔耿睽渺仦殣樠鄽偟壮褙劳充迪蝔镁蘮蒕懱萑踺猳钌
緲蟓蛹醢徣纆褺'
content = ''
for i in text:
    content = content + n2s(s2n(pypinyin.slug(i, separator=' ', style=8)))

print(content)
```

RSA Attack 2

第一个用的 RSA Attack 的方法，剩下的根据 https://www.dazhuanlan.com/wyr-1995/topics/981091 写的。

```
import gmpy2 as gmpy2
from libnum import n2s


def egcd(a, b):
    if a == 0:
        return (b, 0, 1)
    else:
        g, y, x = egcd(b % a, a)
        return (g, x - (b // a) * y, y)


def modinv(a, m):
    g, x, y = egcd(a, m)
    if g != 1:
        raise Exception('modular inverse does not exist')
    else:
        return x % m
```

```
# task1
e = 65537
n1 =
14611545605107950827581005165327694782823188603151768169731431418361306231114
98503777591746143392530805439697080969080407398583537646462986060971029218136
86006186265904984918504045034434142414554873044483448923378774224657157091542
38653505141605904184985311873763495761345722155289457889686019746663293720106
87422732369928827779429220895717244652342059639111489155953781102947315012364
16241081036765167544494928051266425527512783096348467776360421141359905162459
07517377320190091400729277307636724890592155256437996566160995456743018225013
85193759388608612913135158295881100359644580606149295251385193223856362719455
3
c1 =
96507580355493298866427181643918380232881201369420374132076310537603691258499
50316476723484681113104236808581019906700670653062375961216648843536799876895
32305437801346923070145524106271337770666947677115752724993307387122132705797
01272623707355066941911004630825740848453506351567806677768101721151098142927
33469280229711494110645562250012873991413061360817224710750324230796929083802
67160214143720516748000734987068685104675254411687005690312116824966036851568
22382888433511214463726809039715853293714112265407595273005233157398070113637
82120029567192951927339556733152342740645199576701998951005086235618385104 79
n2 =
20937478725109983803079185450449616567464596961348727453817249035110047585580
14282355128957714595812712158679287850938608517845217111245589042947445779721
92028270308842622730613347524934967979353466315098066855891796183674539927497
53318273834113016237120686880514110415113673431170488958730203963489455418967
54412861923439491582039290842297407593275183801218554296884269182420320651779
56938938639451006619409884556959235117773065664193733940919073494316866464855
16325575494902682337518438042711296437513221448397034813099279203955535025939
12013968060449548698076591089243828494545073337515693386315080836979683089236
3
c2 =
11536506945313747180442473461658912307154460869003392732178457643224057969838
22460105983686088371845998600310697037577844372574860708562093878771408132131
58171444141155899522374924484834389103788653592395751693261166680304632758176
09827626048962304593324479546453471881099976644410889657248346038986836461779
78018341168626075677671172057705331950469137355010752529656093646743528381249
33964866781780202924333658980325970273388760451827434928318141756738341983453
37514065596396477709839868387265840430322983945906464646824470437783271607499
08979186939859055731471309467420826176129989470577251344094813942901142594809
0
p =
11810617170951861319033738012072163909610943387175855148175055962860784152519
99333964010458573138419626670876810000779085753498562031979892801375181196104
47265022793158335778819939567162786340083036604758380394175830091289942677310
94096270635401836263248840410297634444690374827621466828546811921494039272512
3
```

```
q =
12371534352197068400012879987607104283057072321811693115146722024476505588941
76268065548681145255669784363239750834987038327945614932913120796913966712748
37322036085911028636844643698862533724625315331567014898932701977758733187411
73877161788515363911817406277396649961220155557592341204564402885798901660341
1
r =
16923914309296392221334368692467736308896348563302709164550115138848256549023
33237968896916242726649851735258120023555304847414328471705113481770657043389
78754457533424010842217007432554862861949141613925946472939183705336155629494
10705047095247481664708043200218930927283558114874021120867801241696013644183
3
d1 = modinv(e, (p - 1) * (q - 1))
m1 = pow(c1, d1, n1)
print(n2s(m1))

# task2
e = 7
n =
14157878492255346300993349653813018105991884577529909522555551468374307942096
21496460417273438191305127374522829393083231448346692252924095899489769747593
98670255613480427259196635469490150246939526419364818415527514846041230971480
71800416608762258562797116583678332832015617217745966495992049762530373531163
82197962736120092154422357817071874134824201216411559377770090395440910311009
29215788210489333468932128050716822355758137241139783415928859577673775874922
02740185970828629767501662195356276862585025913615910839679860669917255271734
41386521134012654419976062844505413166148418487667962694636075300951263434953
7
c =
10262871020519116406312674685238364023536657841034751572844570983750295909492
14910150086980641860373218135008257644759476658757235024667544550893157767015
82955586412195827293455816974482311163180804561125167007179847316559007263881
85866905989088504004805024490513718243036445638662260558477697146032055765285
26344608425981456019754901804409993515835193188515761652723528322906614539096
40949290070569463320513644745284539709042510506056315148690078906 25
from Crypto.Util.number import long_to_bytes

k = 0
while 1:
    if gmpy2.iroot(c + k * n, e)[1] == 1:
        print(long_to_bytes(gmpy2.iroot(c + k * n, e)[0]))
        break
 k += 1

# task3
n =
18819509188106230363444813350468162056164434642729404632983082518225388069544
77737454414231761285844834534413737222298803336652808623663521375622781661086
50459243572321887689136421584486033463304625356961217396227022005403441054641
```

```
266954320117391815312175829498049395557207004573505128983223765918131353119219045803383402035695826818892434524953638495589559471249752937365094264004600839810788461387400506349068244386897127483243368787916226769743418146910412622806042773578898922117171243193296660528100291311722299307234779814687613695167717202505717130279720649749998021680179462747363831480018659297192481590757295e1 = 2519901323c1 =
```


```
323077972622554487253144116900930707207375457876188838798340320636454845149673651390546038190792810731003008634658935110580902859965030353960758140762781979794433739860140051056099246245504845132659399359508980015034299902187473474806669296236265054003600207374876650934764981813930436391408387991892987357770632359962803161864179307401830452124346048755136482329968505251885268570668780020950527742686914005105699624288213261625669518887078263431036297315376669828625894689686639667087245180311428084670957277978055848222339375947599910360770451061833225371050385756102561363259268293155222815017142384620387534487e2 = 3676335737c2 =
```

```
940818595622279161439836719641707846790294650888799822335007385854166736459283129434769062995122371073636785371800857633841379139761091890426137981113087519934854663776695944489430385663011713917022574342380155718317794204988626116362865144125136624722782309455452257758808172415884403909840651554485364309237853885251876941477098008690389600544398998669635962495989736021020715396415375890720335697504837045188626103142204474942751410819466379437091569610294575687793060945525108986660851277475079994466474859114092643797418927645726430175928247476884879817034346652560116597965191204061051401916282814886688467861gcd, s, t = gmpy2.gcdext(e1, e2)m3 = gmpy2.powmod(c1, s, n) * gmpy2.powmod(c2, t, n) % nprint(n2s(int(m3)))
```


MISC

奇妙小游戏

一开始还是挺迷茫的，玩了两遍发现是一个找出口的小游戏~~【废话】~~，最下面的一行是入口，从左起第一个是零，每一条路都要走，找到出口就行了。

这个小游戏规则很简单，但我做出来之后血压就升高了。我手打到了 4-4 ，之后开始写脚本，写了1 个小时左右 ~~【 recv 不太熟练，折腾了好久】~~ ，打完发现最后就是 5-5 ，着实是感觉不如手打 ~~【虽然学到了一些新知识，但是感觉真的好慢】~~ 。

```
#!/usr/bin/env python
# coding=utf-8
from pwn import *
```

```python
from pwnlib.util.iters import mbruteforce
import itertools
import base64

context.log_level = "debug"


def line2dots(line):
    i = len(line)
    i = int(i / 5)

    if line[1] == ' ':
        path = [0]
    else:
        path = [1]

    for k in range(1, i):
        j = k * 5
 if line[j + 1] == ' ' and line[j - 1] == ' ':
            path.append(k)
        elif line[j + 1] == ' ':
            path.append(k - 1)
        elif line[j - 1] == ' ':
            path.append(k + 1)

    if line[i * 5 - 1] == ' ':
        path.append(i)
    else:
        path.append(i - 1)

    return path


def go(entry, path, n):
    k = entry
    for i in range(0, n + 1):
        k = path[n - i][k]
    return k


sh = remote("chuj.top", 50891)

sh.recvuntil(') == ')
hash_code = sh.recvuntil('\n', drop=True).decode().strip()
log.success('hash_code={},'.format(hash_code))

charset = string.printable
proof = mbruteforce(lambda x: hashlib.sha256((x).encode()).hexdigest() ==
hash_code, charset, 4, method='fixed')
```

```
sh.sendlineafter('????> ', proof)

sh.sendlineafter('任意输入开始', 'a')
while 1:
    sh.recvuntil('level')
    sh.recvline()
    path = [0]
    n = -1
 while 1:
        line = sh.recvline()
        if line[3] == '-':
            break
 n = n + 1
 path.append(line2dots(line))
        print(line)

    del path[0]

    sh.recvuntil('is ')

    entry = sh.recv()
    print('entry=', entry)

    ans = go(entry, path, n)
    sh.sendlineafter('answer', ans)

sh.interactive()
```

~~我这周真的好水啊。虽然上周也没多高，但是这周真的好低，好处是要写的 wp 减少了。^-^~~