picoCTF2019-rsa-pop-qui Writeup

原创

FSky-晓 于 2020-07-12 22:13:22 发布 239 收藏

文章标签： 密码学 加密解密 CTF

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。

本文链接：https://blog.csdn.net/weixin_43659784/article/details/107305471

版权

picoCTF2019-rsa-pop-qui Writeup

目录

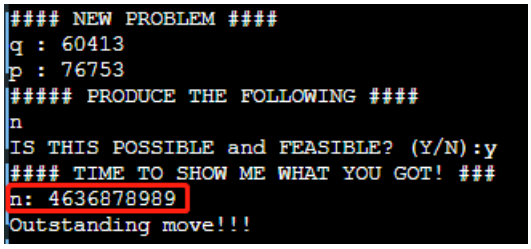
目录

题目

- Class, take your seats! It's PRIME-time for a quiz… nc 2019shell1.picoctf.com 30962
  开始没整明白怎么回事（我的垃圾英语，，，）查了几个题解：
  参考题解1
  参考题解2
  RSA算法
  后来逐渐明白了，就是rsa加密的那几个参数来回搞，不需要破解，解不出来直接选“N”就可以了
- RSA参数：
  - p，q为两个质数
  - n = p*q
  - r = (p-1)*(q-1)
  - ed ≡ 1 (mod n)

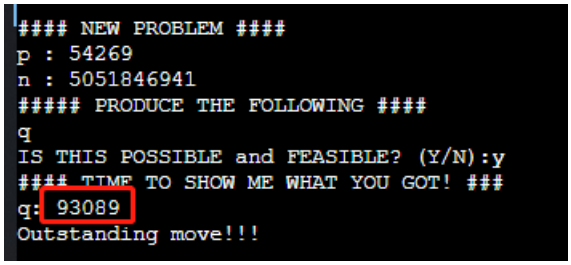
第一题：已知p，q 求 n

- 直接相乘
- n = p*q = **4636878989**



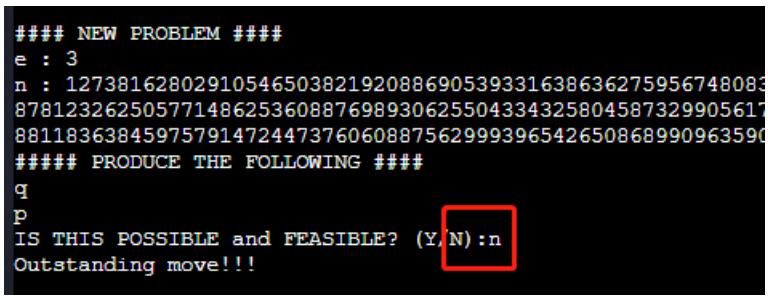
第二题：已知p，n 求 q

- 直接除
- q = n/p = **93089**



第三题：已知n，e 求 p，q

- 这个相当于知道公钥，求原来的p，q，做不到（需要对 n 进行大整数的素因子分解，很难解，这是RSA加密的基础）
- 直接选**“N”**



第四题：已知p，q 求 r

- totient（）：欧拉函数
- r = (p-1)*(q-1) = **836623060**

```
#### NEW PROBLEM ####
q : 66347
p : 12611
##### PRODUCE THE FOLLOWING ####
totient(n)
IS THIS POSSIBLE and FEASIBLE? (Y/N):y
#### TIME TO SHOW ME WHAT YOU GOT! ###
totient(n): 836623060
Outstanding move!!!
```

第五题：已知e，n 和 明文 求 密文

- e，n就是公钥，可以用来对明文加密

```
import gmpy2
import rsa

n = 29129463609326322559521123136222078780585451208149138547799121083622333250646678767769126248182207478527881025116332742616201890576280859777513414460842754045651093593251726785499360828237897586278068419875517543013545369871704159718105354690802726645710699029936754265654381929650494383622583174075805797766685192325859982797796060391271817578087472948205626257717479858369754502615173773514087437504532994142632207906501079835037052797306690891600559321673928943158514646572885986881016569647357891598545880304236145548059520898133142087545369179876065657214225826997676844000054327141666320553082128424707948750331
e = 3
msg = 635729417148931154719098761554457513358196788649948409135266140641404444047520534288284123635766597343146249135508941371039227338020303879324156430477427152910872971 7

c = pow(msg,e,n)
print(c)
```

运行结果为：

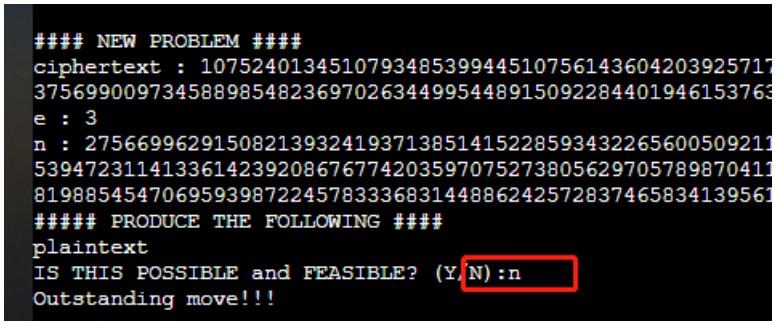
256931246631782714357241556582441991993437399854161372646318659020994329843524306570818293602492485385337029697819837182169818816821461486018802894936801257629375428544752970630870631166355711254848465862207765051226282541748174535990314552471546936536330397892907207943448897073772015986097770443616540466471245438117157152783246654401668267323136450122287983612851171545784168132230208726238881861407976917850248110805724300421712827401063963117423718797887144760360749619552577176382615108244813

```
#### NEW PROBLEM ####
plaintext : 635729417148931154719098761554457513358196788649948407
e : 3
n : 291294636093263225595211231362220787805854512081491385477991210836223332506466787677691262481822074785278810
053546908027266457106990299367542656543819296504943836225831740758057977666851923258599827977960603912718175780874729482056262577174798583697545026151737735140874375045329941426322079065010798350370527973066908
657288598688101656964735789159854588030423614554805952089813314208754536917987606565
#### PRODUCE THE FOLLOWING ####
ciphertext
IS THIS POSSIBLE and FEASIBLE? (Y/N):y
#### TIME TO SHOW ME WHAT YOU GOT! ###
ciphertext: 256931246631782714357241556582441991993437399854161374535990314552471546936536330397892907207943448897073772015986097036074961955257717638261510824481 3
Outstanding move!!!
```

https://blog.csdn.net/weixin_43659784

第六题：已知e，n 和 密文 求 明文

- 这个就相当于知道公钥（e,n）想要解密，所以需要破解RSA，所以办不到
- 直接选“N”



第七题：已知p，q，e 求 d

- 根据 p，q 可以很快求出 n 和 r 所以这个是可解的（甚至不需要算n）

```
import gmpy2
import rsa

p = 92092076805892533739724722602668675840671093008520241548191914215399824020372076186460768206814914423802230398410980218741906960527104568970225804374404612617736579286959865287226538692911376507934256844456333236362669879347073756238894784951597211105734179388300051579994253565459304743059533646753003894559
q = 97846775312392801037224396977012615848433199640105786119757047098757998273009741128821931277074555731813289423891389911801250326299324018557072727051765547115514791337578758859803890173153277252326496062476389498019821358465433398338364421624871010292162533041884897182597065662521825095949253625730631876637
e = 65537

# n = p*q
r = (p-1)*(q-1)
d = int(gmpy2.invert(e,r))
print(d)
```

运行结果为：

14050462695032074691407915484036395331274164162142106949720850791717875804637768204259658981742728704860157395161257861828216370066007421406825523216455037432806708398190787490927301105498818912713173964501580216882539897671455787234582527694655455041421396634767474792259239331924214054644145747862729637416526769455504142139663476747479225923933192421405464414574786272963741652684484355489763878614603686957265320473565084318672273273688891878937905405012220525316570508553874365125840039058097104314464498465491485672 9

```
#### NEW PROBLEM ####
q : 92092076805892533739724722602668675840671093008520241548191914215399
94784951597211105734179388300051579994253565459304743059533646753003894551
p : 97846775312392801037224396977012615848433199640105786119757047098757
64421624871010292162533041884897182597065662521825095949253625730631876631
e : 65537
##### PRODUCE THE FOLLOWING ####
d
IS THIS POSSIBLE and FEASIBLE? (Y/N):y
#### TIME TO SHOW ME WHAT YOU GOT! ###
d: 14050462695032074691407915484036395331274164162142106949720850791717875
25276946554550414213966347674747922592393319242140546441457478627292637416
26844843554897638786146036869572653204735650843186722732736888918789379054
Outstanding move!!!
```

https://blog.csdn.net/weixin_43659784

第八题：已知p，n，e和 密文 求 明文

- 要解密需要私钥（d,n）
- q = n/p
- 有p，q，e 可以参考第七题求出 d，就可以得到私钥（d,n）

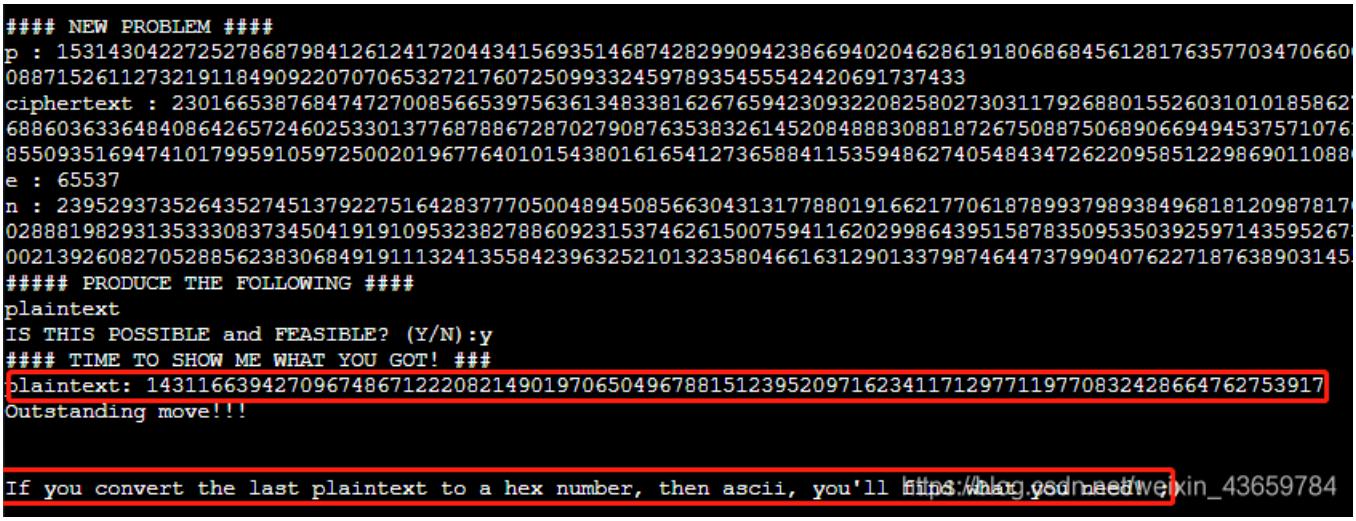
```
import gmpy2
import rsa
p = gmpy2.mpz(153143042272527868798412612417204434156935146874282990942386694020462861918068684561281763577034706600608387699148071015194725533394126069826857182428660427818277378724977554365910231524827258160904493774748749088477328204812171935987088715261127321911849092207070653272176072509933245978935455542420691737433)
n = gmpy2.mpz(23952937352643527451379227516428377705004894508566304313177880191662177061878993798938496818120987817049538365206671401938265663712351239785237507341311858383628932183083145614696585411921662992078376103990806989257289472590902167457302888198293135333083734504191910953238278860923153746261500759411620299864395158783509535039259714359526738924736952759753503357614939203434092075676169179112452620687731670534906069845965633455748606649062394293289967059348143206600765820021392608270528856238306849191113241355842396325210132358046616312901337987464473799040762271876389031455051640937681745409057246190498795697239)
e = 65537
ciphertext = 230166538768474727008566539756361348338162676594230932208258027303117926880155260310101858627968204640007405386633678682736281905585746697699261459099667999407431383305425044003188144448777639313232399800994043033230646011161688603633648408642657246025330137768788672870279087635383261452084888308818726750887506890669494537571076110764779993698480618931668201314772559596644589095764625978782137326626060212768500393448876190971927242710610260309007968081574796585509351694741017995910597250020196776401015438016165412736588411535948627405484347262209585122986901108801424861394120150476365956009191469715176537347 39

q = n//p
r = (p-1)*(q-1)
d = gmpy2.invert(e,r)

plaintext = pow(ciphertext,d,n)
print(plaintext)
```

运行结果：

1431166394270967486712220821490197065049678815123952097162341171297711977083242866476275391 7



最终结果处理

- 根据第八题最后的提示可以知道，先将结果转化为16进制，然后再转化为字符串就可以得到结果

```
import binascii

txt = 1431166394270967486712220821490197065049678815123952097162341171297711977083242866476275391 7

htxt = hex(txt)
s = binascii.a2b_hex(htxt[2:])
print(s)
```

- 输出结果：**b’picoCTF{wA8_th4t$_ill3aGal…o8227181c}’** 即可得到最终的 flag