Nebu1ea 韩磊 [1504895347@qq.com](mailto:1504895347@qq.com)

Misc: RSA\_KU

解题思路：

n = 129699330328568350681562198986490514508637584957167129897472522138320202321246467459276731970410463464391857177528123417751603910462751346700627325019668100946205876629688057506460903842119543114630198205843883677412125928979399310306206497958051030594098963939139480261500434508726394139839879752553022623977

e = 65537

c = 107408576731048313356159300235660481727026022478524939098218582730629791243612611009209057771018646216077107737673659302299864198944390279252388338856195035517355259953749317105146843664196466560211410994799040419513708164311599731714025431317725068605314383138029956400160699540819467907441462359917872006620

#(p-2)\*(q-1) = 129699330328568350681562198986490514508637584957167129897472522138320202321246467459276731970410463464391857177528123417751603910462751346700627325019668067056973833292274532016607871906443481233958300928276492550916101187841666991944275728863657788124666879987399045804435273107746626297122522298113586003834

#(p-1)\*(q-2) = 129699330328568350681562198986490514508637584957167129897472522138320202321246467459276731970410463464391857177528123417751603910462751346700627325019668066482326285878341068180156082719320570801770055174426452966817548862938770659420487687194933539128855877517847711670959794869291907075654200433400668220458

附件内容是上方的字符串，一道已知n，(p-2)(q-1)和(p-1)(q-2)的RSA，具体细节看这篇文章：<https://blog.csdn.net/qq_38798840/article/details/130383064>

用文章的Exp跑一遍就行了。



Exp:

import gmpy2

from Crypto.Util.number import \*

n = 129699330328568350681562198986490514508637584957167129897472522138320202321246467459276731970410463464391857177528123417751603910462751346700627325019668100946205876629688057506460903842119543114630198205843883677412125928979399310306206497958051030594098963939139480261500434508726394139839879752553022623977

e = 65537

c = 107408576731048313356159300235660481727026022478524939098218582730629791243612611009209057771018646216077107737673659302299864198944390279252388338856195035517355259953749317105146843664196466560211410994799040419513708164311599731714025431317725068605314383138029956400160699540819467907441462359917872006620

n1=129699330328568350681562198986490514508637584957167129897472522138320202321246467459276731970410463464391857177528123417751603910462751346700627325019668067056973833292274532016607871906443481233958300928276492550916101187841666991944275728863657788124666879987399045804435273107746626297122522298113586003834

n2=129699330328568350681562198986490514508637584957167129897472522138320202321246467459276731970410463464391857177528123417751603910462751346700627325019668066482326285878341068180156082719320570801770055174426452966817548862938770659420487687194933539128855877517847711670959794869291907075654200433400668220458

ppq=(n-n1+n-n2+4)//3 #p+q

phi=n-ppq+1 #phi=(p-1)\*(q-1)=pq-(p+q)+1

d=gmpy2.invert(e,phi)

flag=long\_to\_bytes((pow(c,d,n)))

print(flag)