攻防世界 Crypto高手进阶区 5分题 RSA_gcd

原创

思源湖的鱼 于 2021-01-01 18:26:14 发布 376 收藏

分类专栏： ctf 文章标签： ctf 攻防世界 crypto rsa

本文链接：https://blog.csdn.net/weixin_44604541/article/details/112062355

ctf 专栏收录该内容

200 篇文章 23 订阅

订阅专栏

前言

继续ctf的旅程

攻防世界Crypto高手进阶区的5分题

本篇是RSA_gcd的writeup

发现攻防世界的题目分数是动态的

就仅以做题时的分数为准了

解题过程

得到两个nec

```
n: 232206198396426241272088043293290792892734979273515640119852920262549143948336915425528908105117512396563616860736282733093903148816045802044297084615875125006361581613034199162592710781738648002670635405269431811737081083244718157829856267231981446432564327749848848806985943645839494857495754673181730344678461433805741454551951527937426117171696022379692865800286627210654953801928151750579454201827423667916614168226239155238685907103876359351798762751470563960185272604884593330511327205589531429840386352237939926516377081504949647854750654045688440399833814039093413020987735333250809100578455738989843142460895
e: 65537
c: 970061474841350329126096623186356211750209628461621670744527635527486908661979652761847321342250999684343029652659411357267584055934507734441909890081870957764232490040558249968360478698114409987802178456754065404083391206314170991365341639488876628146520068285237879447880132925122480100682092585850727313050423656382212083852074627028073112144283941225839719196303604055353969784653503884154120905050306100107090972580657423009024604189148650698093929424553725261094479957392084423522109695639109571611162999859407576250734543094552349277591579082807800045370532078348674473499421302847644692281587005331197384496
```

n：22642739016943309717184794898017950186520467348317322177556419830195164079827782890660385734113396507640392461790899249329899658620250506845740531699023854206947331021605746078358967885852989786535093914459120629747240179425838485974008209140597947135295304382318570454491064938082423309363452665886141604328435366646426917928023608108470382196753292656828513681562077468846105122812084765257799070754405638149508107463233633350462138751758913036373169668828888213323429656344812014480962916088695910177763839393954730732312224100718431146133548897031060554005592930347226526561939922660855047026581292571487960929911

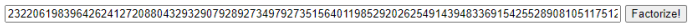
e：65537

c：20513108670823938405207629835395350087127287494963553421797351726233221750526355985253069487753150978011340115173042210284965521215128799369083065796356395285905154260709263197195828765397189267866348946188652752076472172155755940282615212228370367042435203584159326078238921502151083768908742480756781277358357734545694917591921150127540286087770229112383605858821811640935475859936319249757754722093551370392083736485637225052738864742947137890363135709796410008845576985297696922681043649916650599349320818901512835007050425460872675857974069927846620905981374869166202896905600343223640296138423898703137236463508

就是经典的RSA

素数分解

得到pq



Result:		
status (?)	digits	number
FF	617 (show)	2322061983...89_{<617>} = 1383766045...39_{<309>} · 1678074116...51_{<309>}

然后解码

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

p1 = 13837660453353041240023955834042470031241270269902248111935779905471587782929163529083271983503314058069005
38656770793162419191691663751236919176752359794627721036813987252858085510419246248828409015838928581742707144 71
36653175897524186847093813823830700578265129617957996186980184139568278264591684852377 1439
q1 = 16780741164967646254666111964411308191554237875577832705715619128445315088766234341490891695315489718361354
80835589194103596424500013436448140211598287248447308811119550509923985350634098281694621809706295377926769986 47
88011046352755503404087148596436141808048111305995941061644677221803814115705100709 1689351
n1 = 22642739016943309717184794898017950186520467348317322177556419830195164079827782890660385734113396507640392
46179089924932989965862025050684574053169902385420694733102160574607835896788585298978653509391445912062974724 01
79425838485974008209140597947135295304382318570454491064938082423309363452665886141604328435366646426917928023 60
81084703821967532926568285136815620774688461051228120847652577990707544056381495081074632336333504621387517589 13
03637316966882888213323429656344812014480962916088695910177763839393954730732312224100718431146133548897031060 5
54005592930347226526561939922660855047026581292571487960929911

p2 =  13837660453353041240023955834042470031241270269902248111935779905471587782929163529083271983503314058069 00
53865677079316241919169166375123691917675235979462772103681398725285808551041924624882840901583892858174270714 47
13665317589752418684709381382383070057826512961795799618698018413956827826459168485237714 39
q2 =  16363126623371283748182308837833713415102187106027588787133825027435992221805354333353257972877781350995 62
61662615493179160669715503833949420308311581736674332268131534602581626817039237393599222688271607325131529790 64
03757656978327466147004836816589117539365204823835924787430932332613714517188442757620946 49
n2 = 22642739016943309717184794898017950186520467348317322177556419830195164079827782890660385734113396507640392
46179089924932989965862025050684574053169902385420694733102160574607835896788585298978653509391445912062974724 01
79425838485974008209140597947135295304382318570454491064938082423309363452665886141604328435366646426917928023 60
81084703821967532926568285136815620774688461051228120847652577990707544056381495081074632336333504621387517589 13
03637316966882888213323429656344812014480962916088695910177763839393954730732312224100718431146133548897031060 5
54005592930347226526561939922660855047026581292571487960929911

e = 65537

c1 = 97006147484135032912609662318635621175020962846162167074452763552748690866197965276184732134225099968434302
96526594113572675840559345077344419098900818709577642324900405582499683604786981144099878021784567540654040833 91
20631417099136534163948887662814652006828523787944788013292512248010068209258585072731305042365638221208385207 46
27028073112144283941225839719196303604055353969784653503884154120905050306100107090972580657423009024604189148 65
06980939294245537252610944799573920844235221096956391095716111629998594075762507345430945523492775915790828078 00
04537053207834867447349942130284764469228158700533119738449 61
c2 = 20513108670823938405207629835395350087127287494963553421797351726233221750526355985253069487753150978011340
11517304221028496552121512879936908306579635395285905154260709263197195828765397189267866348946188652752076472 1
72155755940282615212228370367042435203584159326078238921502151083768908742480756781277358357734545694917591921 15
01275402860877702291123836058588218116409354758599363192497577547220935513703920837364856372250527388647429471 37
89036313570979641000884557698529769692268104364991665059934932081890151283500705042546087267585797406992784662 09
05981374869166202896905600343223640296138423898703137236463508
```

```
n = q*p

d1 = libnum.invmod(e, (p1 - 1) * (q1 - 1))
d2 = libnum.invmod(e, (p2 - 1) * (q2 - 1))
m1 = pow(c1, d1, n1)
m2 = pow(c2, d2, n2)
string = long_to_bytes(m1) + long_to_bytes(m2)
print(string)
```

得到flag：`flag{336BB5172ADE227FE68BAA44FDA73F3B}`

结语

简单的RSA