[BJDCTF2020]easyrsa writeup

原创

_bestkasscn 于 2021-10-21 20:28:46 发布 196 收藏

文章标签： python 密码学

本文链接：https://blog.csdn.net/bestkasscn/article/details/120894231

[BJDCTF2020]easyrsa

题目描述

```
from Crypto.Util.number import getPrime,bytes_to_long
from sympy import Derivative
from fractions import Fraction
from secret import flag

p=getPrime(1024)
q=getPrime(1024)
e=65537
n=p*q
z=Fraction(1,Derivative(arctan(p),p))-Fraction(1,Derivative(arth(q),q))
m=bytes_to_long(flag)
c=pow(m,e,n)
print(c,z,n)
'''
output:
79225478668577614598074915026542162830127761777895115493506729581018102813484022840983101477965494306892538035109948774201355372685494106526544796208586913241103671820256487884070415999430913862275431821577462029470995723896760843927064060843076570001046656966544091550063132039572928857437917151987819742055786547921231915849576652932083904537483691823331528098823124533597061478081989229167627737217266815889771038774541190437448891645293831880774991949329096439186966468769073273647513809531825178831345918108008489717191848087136943429854581030066760134519122210802527359489936926748993998260848486221458154610353211574867762320966747162287218527507025792476601502007280526735983905939328431659588293337228973212727407643458751933330014247301034469480388516855754880120249593322621543776332928024211355652449845755956287290081160205694442396740377762330696188075761324632872961664303262896407293127208586692804597379937471184682515778105696516417850523252424580917923560757156717422882256169788864596855934360837533198809715714526435762673814164655635350099492411587574819831803629689860409700093827219590305673356588015054027536923963779397592332959871600335030825932143675257929100035556043154222969975995514115291470836249448215310745161336895413406690009324766200789179248896951942047235448901612351128459309145825547569298479821101249094161867207686537607047447968708758990950136380924747359052570549594098569970632854351825950729752563502284849263730127586382522703959893392329333760927637353052250274195821469023401443841395096410231843592101426591882573405934188675124326997277775238287928403743324297705151732524641213516306585297722190780088180705070359469719869343939106529204798285957516860774384001892777525916167743272419958572055332232056095979448155082465977781482598371994798871917514767508394730447974770329967681767625495394441

'''
```

c,e,n已知，求出p,q即可

题目中关于p,q的线索在变量z，分析z，推导公式如下：

$$\frac{1}{(\arctan p)'}-\frac{1}{(ar\,\mathrm{th}\,q)'}=z$$

其中Derivative()是求导函数,Fraction(a,b)相等于a/b，经过计算可知z = p^2 + q^2
又因为n = p * q,所以

```
p + q = sqrt(z + 2 * n)
p - q = sqrt(z - 2 * n)
```

连立方程组即可解出p,q

exp如下：

```
import gmpy2
from Crypto.Util.number import *

c = 79225478668577614598074915026542162830127761777895115493506729581018102813484022840983101477965494306892538035109948774201355372685494106526544796208586913241103671820256487884070415999430913862275431821577462029470995723896760843927064060843076570001046656966544091550063132039572928857437917151987819742055786547921231915849576652932083904537483691823331528098823124533597061478081989229167627737217266815889771038774541190437448891645293831880774991949329096439186966468769073273647513809531825178831345918108008489717191848087136943429854581030066760134519122210802527359489936926748993998260848486221458154610 35
z = 32115748677623209667471622872185275070257924766015020072805267359839059393284316595882933372289732127274076434587519333300142473010344694803885168557548801202495933226215437763329280242113556524498457559562872900811602056944423967403777623306961880757613246328729616643032628964072931272085866928045973799374711846825157781056965164178505232524245809179235607571567174228822561697888645968559343608375331988097157145264357626738141646556353500994924115875748198318036296898604097000938272195903056733565880150540275369239637793975923329598716003350308259321436752579291000355560431542229699759955141152914708362494482
n = 15310745161336895413406690009324766200789179248896951942047235448901612351128459309145825547569298479821101249094161867207686537607047447968708758990950136380924747359052570549594098569970632854351825950729752563502284849263730127586382522703959893392329333760927637353052250274195821469023401443841395096410231843592101426591882573405934188675124326997277775238287928403743324297705151732524641213516306585297722190780088180705070359469719869343939106529204798285957516860774384001892777525916167743272419958572055332232056095979448155082465977781482598371994798871917514767508394730447974770329967681767625495394441

x = gmpy2.iroot(z + 2 * n, 2)[0]
y = gmpy2.iroot(z - 2 * n, 2)[0]
e = 65537
p = (x + y) // 2
q = x - p
d = gmpy2.invert(e, (p - 1) * (q - 1))
print(long_to_bytes(pow(c, d, n)))

) * (q - 1))
print(long_to_bytes(pow(c, d, n)))
```