```
from Crypto.Util.number import *
m=bytes_to_long(b'rOOt2023{*****}')
e=65537
p=getPrime(256)
q=getPrime(512)
r=getPrime(512)
n=p*q*r
P=pow(p,2,n)
Q=pow(q,2,n)
c=pow(m,e,n)
print(f"P = {P}")
print(f"Q = {Q}")
print(f"n = {n}")
print(f"c = {c}")
'''
P =
590262762941440903285280188171904578629499599677014994500307517516567833606601926526530056043656619929516121549968231498113945813100767622471589254977672 1
Q =
114835830930775259980589040661622778314062162886181458325573003516068194820210343509561878607897119092584119052965553213784766045043093559663974212258201396286990987782389235318468303024496790444337117821808939974013154622033250305778820459964439279014357509293010532632103484167089585068019300433903279247129
n =
109855065266951861635480469955699139631501043451300653852988242333617642967177896743043630421275426180507292201151045384462144211845198950677345590865535785840965872169859178548855059767149863446208977523409373137381826941362731638314469561391540681938576179623030701201695950280405585649808478881426941390012764675310511183780321064887350668863303949206748115979448599883241310092454471183780321064887350668863303949206748115979448599883241310092454472842949318111356437098282635826127637115573735230012872381578789738954069800934187491519330951237250482932870245148518876922500118638471379820201500955328491175605603665554183899384309297952797978806266606138574242982978643739385703722572375828714225787581669804909753468078962955668555742384799661096290372093651108166816145895127560474145585820009681747796045930635272790246185888411'''
```

根据题目可以得到

$$P = p^2 modn \quad Q = q^2 modn$$

由因为有

```
p=getPrime(256)
q=getPrime(512)
r=getPrime(512)
n=p*q*r
```

可以认为 $n > p^2$ 且 $n > q^2$ ，那么此时 $P = p^2 modn$ 与 $Q = q^2 modn$ 可以化简为$P = p^2$ 与 $Q = q^2$，此时只需要将P与Q分别开方即可求得p与q，又有$n = p * q * r$，则可以通过 $r = n/(p * q)$ 求得 r 。至此 p, q, r 计算完毕。

接着计算 $phi(n) = (p - 1) * (q - 1) * (r - 1)$ 即可通过逆元求得私钥d以得到m。

```
import gmpy2
from Crypto.Util.number import *
P =
590262762941440903285280188171904578629499599677014994500307517516567833606601926526530056043656619929516121549968231498113945813100767622471589254977672 1
Q =
114835830930775259980589040661622778314062162886181458325573003516068194820210343509561878607897119092584119052965553213784766045043093559663974212258201396286990987782389235318468303024496790444337117821808939974013154622033250305778820459964439279014357509293010532632103484167089585068019300433903279247129
n =
109855065266951861635480469955699139631501043451300653852988242333617642967177896743043630421275426180507292201151045384462144211845198950677345590865535785840965872169859178548855059767149863446208977523409373137381826941362731638314469561391540681938576179623030701201695950280405585649808478881426941390012764675310511837803210648873506688633039492067481159794485998832413100924544 7
c =
428429493181113564370982826358261276371155737352300128723815787897389540698009341874915193309512372504829328702451485188769225001186384713798202015009553284911756056036655541838993843092979527979788062666061385742429829786437393857037225723758287142257875816698049097534680789629556685557423847996610962903720936511081668161458951275604741455858200096817477960459306352727902461858884 1
e = 0x10001
p = gmpy2.iroot(P,2)[0]
q = gmpy2.iroot(Q,2)[0]
r = n // p // q
phi = (p-1)*(q-1)*(r-1)
d = gmpy2.invert(e,phi)
print(long_to_bytes(gmpy2.powmod(c,d,n)))
```