CTF-RSA解密脚本

原创

1stPeak 于 2021-08-25 13:24:54 发布 679 收藏 4

分类专栏： CTF储备知识 文章标签： RSA

本文链接：https://blog.csdn.net/qq_41617034/article/details/119909235

版权

CTF储备知识 专栏收录该内容

17 篇文章 12 订阅

订阅专栏

cipher.txt内容：

```
0x1055f45639da0fe7ebb355a6f140425fb2d10b74a61860e60cdadc0d205d4905833a20dac9c2e2a873872f1b5d6ac045c996b9136414
b8648b6fc821aea718fd4175324fc254bcaeb2cb2065bcf76d1351b483c2caf87a89ed3a3b0933d82466f6d022b728b85db547596509fb8
a2d46ee920d306f5d591f50a6ad2c505c9c32e30dcd18faf1500cc1d79b1505dab7ab05943ff9bb4a0de531e9c609a4f5cdb850c611e673f
18d89557e46a48bb35e09e10a521c7b45eb3aac85ddb4684b7695815c79b8300f991582b99299a5c28361247e4902da13f267ec7a925
de0cd4d2d0ba2a2f47914d8c3b10a1dbc045f74a4d8617dfe0657aec8045deafbe60dd6d7aL
```

Pubkey.txt内容：

```
65537
337741676001996910724704248988429281685705599403627707860606993209895468516951064669241638168437298283999846497709007930148960378847740396605625469370904128442761855603849649835082911748678080821823865668133931570542594641088581589037395781197603942283415646962255139544009955436296242099425653699725556799803599929555148265897812867381006161492268853024035050624154926796332172753791534218301050216734175446083982498663980427864216304959688108540367820251205099990227738060695910801901669200796882173349685286417477392412343539188920292635443881611604276685189916669602513811067888994519123170012475375764281862916891689
```

做题时，没有用到题目给的encrypt，用的某大佬的脚本

解题过程如下：

（1）用脚本分解n得到p，q，脚本如下

```
def gcd(a, b):
  while b:
    a, b = b, a%b
  return a

def mapx(x):
  x=(pow(x,n-1,n)+3)%n
  return x

n=337741676001996910724704248988429281685705599403627707860606993209895468516951064669241638168437298283999846497709007930148960378847740396605625469370904128442761855603849649835082911748678080821823865668133931570542594641088581589037395781197603942283415646962255139544009955436296242099425653699725556799803599929555148265897812867381006161492268853024035050624154926796332172753791534218301050216734175446083982498663980427864216304959688108540367820251205099990227738060695910801901669200796882173349685286417477392412343539188920292635443881611604276685189916669602513811067888994519123170012475375764281862916891
x1=x2=1

while True:
  x1=mapx(x1)
  x2=mapx(mapx(x2))
  p=gcd(x1-x2,n)
  if (p!=1):
    print(p)
    print(n/p)
 break
```

得到p、q

> p：
> 17799346181607540824086675222721031931682557429100037672752399131508609760506383756334228656081982384961014671338337038338626029556510897392094459314167702461211451711983167666545675423523317234436261068493854277438695689406667510384024463320246966172505094817799567100907031148625364642043506117507866044118 3
> q（n/p）：
> 18974948436644986163073648262203020422960007493673339722966873858660589597981182399402950072544858133274686046828954004112576872614861457925506299417753172778460519409483699828267671243528627349784295636899711603617016539391202256093579193466269545387084602431291560404980521941014042046916379777912964445458 3

（2）使用脚本求出d

```
# coding = utf-8

def computeD(fn, e):

    (x, y, r) = extendedGCD(fn, e)

    #y maybe < 0, so convert it

    if y < 0:

        return fn + y

    return y



def extendedGCD(a, b):
```

```
    #a*xi + b*yi = ri

    if b == 0:

        return (1, 0, a)

    #a*x1 + b*y1 = a

    x1 = 1

    y1 = 0

    #a*x2 + b*y2 = b

    x2 = 0

    y2 = 1

    while b != 0:

        q = a / b

        #ri = r(i-2) % r(i-1)

        r = a % b

        a = b

        b = r

        #xi = x(i-2) - q*x(i-1)

        x = x1 - q*x2

        x1 = x2

        x2 = x

        #yi = y(i-2) - q*y(i-1)

        y = y1 - q*y2

        y1 = y2

        y2 = y

    return(x1, y1, a)



p = 17799346181607540824086675222721031931682557429100037672752399131508609760506383756334228656081982384961014671338337038338626029556510897392094459314167702461211451711983167666545675423523317234436261068493854277438695689406667510384024463320246966172505094817799567100907031148625364642043506117507866044118 3

q = 18974948436644986163073648262203020422960007493673339722966873858660589597981182399402950072544858133274686046828954004112576872614861457925506299417753172778460519409483699828267671243528627349784295636899711603617016539391202256093579193466269545387084602431291560404980521941014042046916379779129644454583
e = 65537
```

```
n = p * q

fn = (p - 1) * (q - 1)



d = computeD(fn, e)

print d
```

得出d：

> 112644117888393555924448563016144883639564719040610562558816358050900943754574002037631928942211307595582169533956749551203075758134395983780242634077074361650699431462974283932667688252477310383499794860522628076795093361992679194549329340455273005636982150713352348141553162426210587298918869319141877420846647023926024298068356174682098443387113155484553154526927006164644655631087679216937211504529396507251538746448986365437328542506411294112061096428194888787822495751306726571826657800115600501592812123592229802104726021787464119823287399350938835906708119662436613826997770787476771581082505579645769896020891126644117888393555924448563016144883639564719040610562558816358050900943754574002037631928942211307595582169533956749551203075758134395983780242634077074361650699431462974283932667688252477310383499794860522628076795093361992679194549329340455273005636982150713352348141553162426210587298918869319141877420846647023926024298068356174682098443387113155484553154526927006164644655631087679216937211504529396507251538746448986365437328542506411294112061096428194888787822495751306726571826657800115600501592812123592229802104726021787464119823287399350938835906708119662436613826997770787476771581082505579645769896020891126644117888393555924448563016144883639564719040610562558816358050900943754574002037631928942211307595582169533956749551203075758134395983780242634077074361650699431462974283932667688252477310383499794860522628076795093361992679194549329340455273005636982150713352348141553162426210587298918869319141877420846647023926024298068356174682098443387113155484553154526927006164644655631087679216937211504529396507251538746448986365437328542506411294112061096428194888787822495751306726571826657800115600501592812123592229802104726021787464119823287399350938835906708119662436613826997770787476771581082505579645769896020892
```
n = 337741676001996910724704248988429281685705599403627707860606993209895468516951064669241638168437298283999846497709007930148960378847740396605625469370904128442761855603849649835082911748678080821823865668133931570542594641088581589037395781197603942283415646962255139544009955436296242099425653699725556799803599929555148265897812867381006161492268853024035050624154926796332172753791534218301050216734175446083982498663980427864216304959688108540367820251205099990227738060695910801901669200796882173349685286417477392412343539188920292635443881611604276685189916669602513811067888994519123170012475375764281862916891c = eval('0x1055f45639da0fe7ebb355a6f140425fb2d10b74a61860e60cdadc0d205d4905833a20dac9c2e2a873872f1b5d6ac045c996b9136414b8648b6fc821aea718fd4175324fc254bcaeb2cb2065bcf76d1351b483c2caf87a89ed3a3b0933d82466f6d022b728b85db547596509fb8a2d46ee920d306f5d591f50a6ad2c505c9c32e30dcd18faf1500cc1d79b1505dab7ab05943ff9bb4a0de531e9c609a4f5cdb850c611e673f18d89557e46a48bb35e09e10a521c7b45eb3aac85ddb4684b7695815c79b8300f991582b99299a5c28361247e4902da13f267ec7a925de0cd4d2d0ba2a2f47914d8c3b10a1dbc045f74a4d8617dfe0657aec8045deafbe60dd6d7aL')
d = 11264411788839355592444856301614488363956471904061056255881635805090094375457400203763192894221130759558216953395674955120307575813439598378024263407707436165069943146297428393266768825247731038349979486052262807679509336199267919454932934045527300563698215071335234814155316242621058729891886931914187742084664702392602429806835617468209844338711315548455315452692700616464465563108767921693721150452939650725153874644898636543732854250641129411206109642819488878782249575130672657182665780011560050159281212359222980210472602178746411982328739935093883590670811966243661382699777078747677158108250557964576989602089
m = pow(c, d, n)
print hex(m)
```

注：环境使用python2
可以参考例题：RSABD