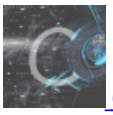
非套路的RSA题

原创

贮藏的仓鼠 于 2019-11-12 17:12:07 发布 383 收藏

分类专栏： CTF 文章标签： RSA 非套路

本文链接：https://blog.csdn.net/qq_42391153/article/details/103031816



CTF 非套路的RSA题

做了很多RSA类型的题目

除了典型的RSA简单解密，还有各种套路题

举几个栗子

e很大的wiener攻击

e很小像3和2的直接开放

低加密指数广播攻击

共模攻击

…

这里介绍一下前两天在湖湘杯踩得坑的非套路RSA

嗯，下面是这道题

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

flag = open("flag.txt","rb").read()
m=libnum.s2n(flag)
p=getPrime(1024)
q=getPrime(1024)
n=p*q
e=65537
c=pow(m,e,n)
phi=(p-1)*(q-1)
d=gmpy2.invert(e,phi)
dp=d%(p-1)
print dp,n,e,c

#
dp:84373069210173690047629226878686144017052129353931011112880892379361035492516066159394115482289291025932915787077633999791002846189004408043685986856359812230222233165493645074459765748901898518115384084258143483508823079115319711227124403284267559950883054402576935436305927705016459382628196407373896831725
n:22000596569856085362623019573995240143720890380678581299411213688857584612953014122879995808816872221032805734151343458921719334360194024890377075521680399678533655114261000716106870610083356478621445541840124447459943322577740268407217950081217130055057926816065068275999620502766866379465521042298370686053823448099778572878765782711260673185703889168702746195779250373642505375725925213796848495518878490786035363094086520257020021547827073768598600151928787434153003675096254792245014217044607440890694190989162318846104385311646123343795149489946251221774030484424581846841141819601874562109228016707364220840611
e:65537
c:14874271064669918581178066047207495551570421575260298116038863877424499500626920855863261194264169850678206604144314318171829367575688726593323863145664241189167820996601561389159819873734368810449011761054668595565217970516125181240869998009561140277444653698278073509852288720276008438965069627886972839146199102497874818473454932012374251932864118784065064885987416408142362577322906063320726241313252172382519793691513360909796645028353257317044086708114163313328952830378067342164675055195428728335222242094290731292113709866489975077052604333805889421889678354330267704176247030117181203474154603851824297957 35
```

可以看出这里基本上把能给的给了，为了题目难度给了个dp，不是传送那个dp，dp=d mod (p-1)

试过很多方法，先试着解n，发现factor没有这个n的记录
用工具解，边试边试着推到dp的公式

emmm，工具解着就要几个小时了。
推导dp的公式，果然没有数论的基础，理解这个东西真的麻烦

Two days later…

今天看了writeup，原来这已经是一道老题了，只是没做到，难怪是easyRSA

今年1月份飘零大佬发在合天智汇，下面是链接
https://zhuanlan.zhihu.com/p/43033684

现在我们可以知道的是

$c \equiv me \bmod n$

$m \equiv cd \bmod n$

$\phi(n)=(p-1)*(q-1)$

$d*e \equiv 1 \bmod \phi(n)$

$dp \equiv d \bmod (p-1)$

由式5*e可以得到

$dp*e \equiv d*e \bmod (p-1)$

因此可以得到

$d*e=k*(p-1)+dp*e$

$d*e \equiv 1 \bmod \phi(n)$

我们将式1带入式2可以得到

$k*(p-1)+dp*e \equiv 1 \bmod (p-1)*(q-1)$

故此可以得到

$k2*(p-1)*(q-1)+1=k1*(p-1)+dp*e$

变换一下

$(p-1)*[k2*(q-1)-k1]+1=dp*e$

因为

$dp<p-1$

可以得到

$e>k2*(q-1)-k1$

我们假设

$x=k2*(q-1)-k1$

可以得到x的范围为

$(0,e)$

因此有

$x*(p-1)+1=dp*e$

那么我们可以遍历

$x \in (0,e)$

求出p-1，求的方法也很简单，遍历65537种可能，其中肯定有一个p可以被n整除那么求出p和q，即可利用

$\phi(n)=(p-1)*(q-1)$

$d*e \equiv 1 \bmod \phi(n)$

推出

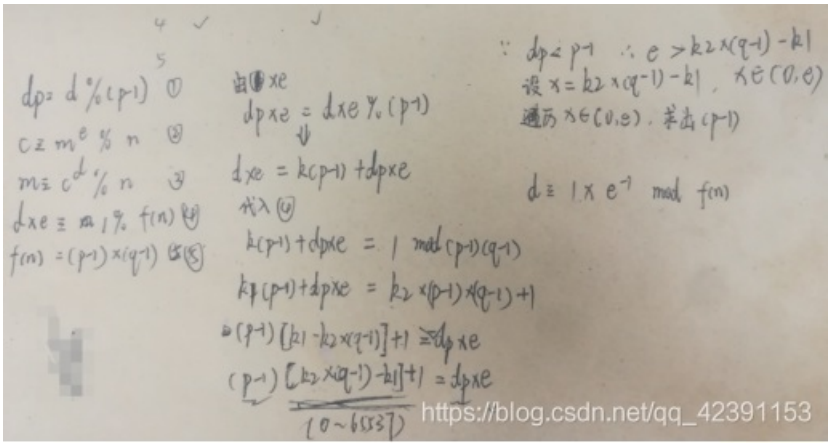
$d \equiv 1*e-1 \bmod \phi(n)$

注：这里的-1为逆元，不是倒数的那个-1

```
#coding:utf-8
import gmpy2
import libnum
n = 220005965698560853626230195739952401437208903806785812994112136888575846129530141228799958088168722210328057
341513434589217193343601940248903770755216803996785336551142610007161068706100833564786214455418401244474599433 2
257774026840721795008121713005505792681606506827599962050276686637946552104229837068605382344809977857287876578 2
711260673185703889168702746195779250373642505375725925213796848495518878490786035363094086520257020021547827073 7
685986001519287874341530036750962547922450142170446074408906941909891623188461043853116461233437951494899462512 2
177403048442458184684114181960187456210922801670736422084061 1
e = 65537
dp = 843730692101736900476292268786861440170521293539310111128808923793610354925160661593941154822892910259329157
787077633999791002846189004408043685986856359812230222233165493645074459765748901898518115384084258143483508823 0
791153197112271244032842675599508830544025769354363059277050164593826281964073738968317 25
c = 148742710646699185811780660472074955515704215752602981160388638774244995006269208558632611942641698506782066
041443143181718293675756887265933238631456642411891678209966015613891598198737343688104490117610546685955652179 7
051612518124086999800956114027744465369827807350985228872027600843896506962788697283914619910249787481847345493 2
012374251932864118784065064885987416408142362577322906063320726241313252172382519793691513360909796645028353257 3
170440867081141633133289528303780673421646750551954287283352222420942907312921137098664899750770526043338058894 2
188996783543302677041762470301171812034741546038518242979573 5

for i in range(1,65538):
    if (dp*e-1)%i == 0: #由(p-1)[k2*(q-1)-k1]+1 = dp*e，也就是说算模 k2*(q-1)-k1等不等于0，就是求k2*(q-1)-k1是不是其一个因子
        if n%(((dp*e-1)/i)+1)==0:  #这里的判断是这样的 由(p-1)[k2*(q-1)-k1]+1 = dp*e 所以这里的 (dp*e-1)/i求的是 p-1，再加1就是p，n%p==0就是n=p*q
            p=((dp*e-1)/i)+1
            q=n/(((dp*e-1)/i)+1)
            phi = (p-1)*(q-1)
            d = gmpy2.invert(e,phi)%phi
            print libnum.n2s(pow(c,d,n))
```

推导理解了一下



最后，感谢飘零大佬！！