# CTF CRYPTO RSA入门刷题

原创

北冥有鱼-其名为咸 于 2021-04-22 20:23:42 发布 247 收藏 1

分类专栏： CTF 文章标签： rsa 密码学



本文链接：https://blog.csdn.net/weixin_46286993/article/details/116028185





## CTF CRYPTO RSA入门刷题

起因：因为学院要办一个什么极客挑战赛，对CTF一窍不通的我被三个学弟拉进来做密码手

这周六比赛，比赛前只能临时抱佛脚刷刷题

靶场（题库）

顺着CRYPTO专题往下做的，本篇只记录一些RSA相关的题目解答，顺序是按照point从过的人多到少

### RSA

```
在一次RSA密钥对生成中，假设p=473398607161，q=4511491，e=17
求解出d作为flga提交
```

rsa入门题，知道原理即可，不解释

代码：

```
import gmpy2

p = 473398607161
q = 4511491
e = 17

phi = (p - 1) * (q - 1)
d = gmpy2.invert(e, phi)
print(d)
```

得到答案flag{125631357777427553}

### RSA1

```
p = 863763376725700856709965348654109117132049150943361544753916243791124417588566780639841179052408355344515811350222774520620532769093950403299469990205322 9
q = 126406749739964727691760479371708834209270508214800105815931371353724738805956137373376306297525773461470392840300825934907766305725849599542052336880228469
dp = 650079570221683462110904235119326153065004384105625293093094966335862501688183284072806602615026469307610935487409984138045488171609777830726811691058292 9
dq = 783472263673553449019532580386470672380574033551303889137911760438881683674556098098256795673512201963002175438762767516968043599582527539160811120550041
c = 247223054038873820735673164676490806626315529059602293990791079956021544181760563358006388875276141640735304376570850796761573502053519452229893513160764865735995760419783398722659250627643185360890073102702785261596789374319038628924007479155251189839599706079341429747366757843259934459420313721073421038 52
```

已知p、q、dp、dp、c，求明文m
其中dp、dq含义如下：
$dp \equiv d \pmod{p-1}$
$dq \equiv d \pmod{q-1}$
原理费马小定理+中国剩余定理
具体见代码

```
import gmpy2
p = 863763376725700856709965348654109117132049150943361544753916243791124417588566780639841179052408355344515811350222774520620532769093950403299469990205322 9
q = 126406749739964727691760479371708834209270508214800105815931371353724738805956137373376306297525773461470392840300825934907766305725849599542052336880228469
dp = 650079570221683462110904235119326153065004384105625293093094966335862501688183284072806602615026469307610935487409984138045488171609777830726811691058292 9
dq = 783472263673553449019532580386470672380574033551303889137911760438881683674556098098256795673512201963002175438762767516968043599582527539160811120550041
c = 247223054038873820735673164676490806626315529059602293990791079956021544181760563358006388875276141640735304376570850796761573502053519452229893513160764865735995760419783398722659250627643185360890073102702785261596789374319038628924007479155251189839599706079341429747366757843259934459420313721073421038 52

n=p*q
m1 = gmpy2.powmod(c,dp,p)
m2 = gmpy2.powmod(c,dq,q)
p_inv = gmpy2.invert(p,q)
q_inv = gmpy2.invert(q,p)
m = gmpy2.powmod((q*q_inv*m1+p*p_inv*m2),1,n)
print(m)
print(hex(m))
print(bytes.fromhex(hex(m)[2:]))
```

## RSA3

```
c1=22322035275663237041646893770451933509324701913484303338076210603542612758956262869640822486470121149424485571361007421293675516338822195280313794991136048140918842471219840263536338886250492682739436410013436651161720725855484866690084788721349555662019879081501113222996123305533009325964377798892703161521852805956811219563883312896330156298621674684353919547558127920925706842808914762199011054955816534977675267395009575347820387073483928425066536361482774892370969520740304287456555508933372782327506569010772537497541764311429052216291198932092617792645253901478910801592878203564861118912045464959832566051361
n=22708078815885011462462049064339185898712439277226831073457888403129378547350292420267016551819052430779004755846649044001024141485283286483130702616057274698473611149508798869706347501931583117632710700787228016480127677393649929530416598686027354216422565934459015161927613607902831542857977859612596282353679327773303727004407262197231586324599181983572622404590354084541788062262164510140605868122410388090174420147752408554129789760902300898046273909007852818474030770699647647363015102118956737673941354217692696044969695308506436573142565573487583507037356944848039864382339216266670673567488871508925311154801
e1=11187289
c2=18702010045187015556548691642394982835669262147230212731309938675226458555210425972429418449273410535387985931036711854265623905066805665751803269106880746769003478900791099590239513925449748814075904017471585572848473556490565450062664706449128415834787961947266259789785962922238701134079720414228414066193071495304612341052987455615930023536823801499269773357186087452747500840640419365011554421183037505653461286732740983702740822671148045619497667184586123657285604061875653909567822328914065337797733444640351518775487649819978262363617265797982843179630888729407238496650987720428708217115257989007867331698397
e2=9647291
```

题目：知道同一明文m在不同密钥e1、e2下加密的结果，求明文m

原理：共模攻击，e1、e2是互质的，利用扩展欧几里得求e1*x*+*e*2*y*=1，然后把m的幂次变为1即可，别忘了x或y小于零的时候要取模逆

代码：

```
import gmpy2

c1 = 22322035275663237041646893770451933509324701913484303338076210603542612758956262869640822486470121149424485571361007421293675516338822195280313794991136048140918842471219840263536338886250492682739436410013436651161720725855484866690084788721349555662019879081501113222996123305533009325964377798892703161521852805956811219563883312896330156298621674684353919547558127920925706842808914762199011054955816534977675267395009575347820387073483928425066536361482774892370969520740304287456555508933372782327506569010772537497541764311429052216291198932092617792645253901478910801592878203564861118912045464959832566051361
n = 22708078815885011462462049064339185898712439277226831073457888403129378547350292420267016551819052430779004755846649044001024141485283286483130702616057274698473611149508798869706347501931583117632710700787228016480127677393649929530416598686027354216422565934459015161927613607902831542857977859612596282353679327773303727004407262197231586324599181983572622404590354084541788062262164510140605868122410388090174420147752408554129789760902300898046273909007852818474030770699647647363015102118956737673941354217692696044969695308506436573142565573487583507037356944848039864382339216266670673567488871508925311154801
e1 = 11187289
c2 = 18702010045187015556548691642394982835669262147230212731309938675226458555210425972429418449273410535387985931036711854265623905066805665751803269106880746769003478900791099590239513925449748814075904017471585572848473556490565450062664706449128415834787961947266259789785962922238701134079720414228414066193071495304612341052987455615930023536823801499269773357186087452747500840640419365011554421183037505653461286732740983702740822671148045619497667184586123657285604061875653909567822328914065337797733444640351518775487649819978262363617265797982843179630888729407238496650987720428708217115257989007867331698397
e2 = 9647291

gcd, x, y = gmpy2.gcdext(e1, e2)
if x < 0:
    x = -x
    c1 = gmpy2.invert(c1, n)
if y < 0:
    y = -y
    c2 = gmpy2.invert(c2, n)

ans = gmpy2.powmod(c1, x, n) * gmpy2.powmod(c2, y, n) % n
print(ans)
print(hex(ans))
print(bytes.fromhex(hex(ans)[2:]))
```

## RSA2

题目：

```
e = 65537
n = 24825400785152624117772152669890180298583276617622160961225887737162058006043310153832803030521991869764361981420093067961210988553380133534844502375167047843707305554472428068473329805159916766030364518314616149748535863368149212966880240206579778990555048954764511878726660192942972413316776846530966590611 3
dp = 90507449805234690464302513287951833069192517457305400462187725331868267505542197094355201669552856036483444630319693920705664292714809329037444021050365 7
c = 14042367097625269680753367358620940057566428210068411978420352712452118899640382659743688376604187906749428095741020195893573736038080184545382929399743341418883872575179626170262202858721156035336284719106030657851051138096516213347269871306359262102895916707278148256267368309059052121421807116028766518075 1
```

题目：已知e、n、dp、c，求明文

思路：先放链接，以后再补详细：题解

代码：

```
import gmpy2

e = 65537
n = 24825400785152624117772152669890180298583276617622160961225887737162058006043310153832803030521991869764361981420093067961210988553380133534844502375167047843707305554472428068473329805159916766030364518314616149748535863368149212966880240206579778990555048954764511878726660192942972413316776846530966590611 3
dp = 90507449805234690464302513287951833069192517457305400462187725331868267505542197094355201669552856036483444630319693920705664292714809329037444021050365 7
c = 14042367097625269680753367358620940057566428210068411978420352712452118899640382659743688376604187906749428095741020195893573736038080184545382929399743341418883872575179626170262202858721156035336284719106030657851051138096516213347269871306359262102895916707278148256267368309059052121421807116028766518075 1

p = q = 0
for x in range(1, e):
    if (dp * e - 1) % x == 0 and n % (((dp * e - 1) // x) + 1) == 0:
        p = ((dp * e - 1) // x) + 1
        q = n // p

phi = (p - 1) * (q - 1)
d = gmpy2.invert(e, phi)
m = gmpy2.powmod(c, d, n)
print(m)
print(hex(m))
print(bytes.fromhex(hex(m)[2:]))
```

得到答案为：flag{wow_leaking_dp_breaks_rsa?_98924743502}