CTF中关于RSA的常见题型

原创

abtgu 于 2020-09-11 18:10:15 发布 1257 收藏 21

分类专栏： 密码学 CTF 文章标签： 密码学 信息安全

本文链接：https://blog.csdn.net/weixin_43790779/article/details/108539518


密码学 同时被 2 个专栏收录

10 篇文章 2 订阅

订阅专栏


CTF

22 篇文章 1 订阅

订阅专栏

CTF中关于RSA的常见题型

> 关RSA算法原理的描述请看https://blog.csdn.net/weixin_43790779/article/details/105622327

1.已知(p, q, e)，求d。

示例：【CTF秀-crypto4】 p=447685307 q=2037 e=17，提交flag{d}即可。

解题思路：

```
import gmpy2
p = 447685307
q = 2037
e = 17
phi = (p-1)*(q-1)
d = gmpy2.invert(e,phi)
print(d)
```

2.已知(p, q, e,c)，求m。

示例：【CTF秀-crypto5】 p=447685307 q=2037 e=17 c=704796792，提交flag{m}。

解题思路：

```
import gmpy2
p=447685307
q=2037
e=17
c=704796792

phi = (p-1)*(q-1)
d = gmpy2.invert(e,phi)
m = gmpy2.powmod(c,d,p*q)

print(m)
```

3.已知(p, q, dp,dq,c)，求m。

示例：【BUUCTF-RSA1】

解题思路：

$m = c \mod n$

由③得，

$c =$ $n+$ ③

脚本代码如下：

```
p = 86376337672570085670996534865410911713204915094336154475391624379112441758856678063984117905240835534451581135022277452062053276909395040329946999020532 29
q = 12640674973996472769176047937170883420927050821480010581593137135372473880595613737337630629752577346147039284030082593490776630572584959954205336880228469
dp = 65007957022168346211090423511932615306500438410562529309309496633586250168818328407280660261502646930761093548740998413804548817160977783072681169105829 29
dq = 78347226367355344901953258038647067238057403355130388913791176043888168367455609809825679567351220196300217543876276751696804359958252753916081112055004 1
c = 24722305403887382073567316467649080662631552905960229399079107995602154418176056335800638887527614164073530437657085079676157350205351945222989351316076486573599576041978339872265925062764318536089007310270278526159678937431903862892400747915525118983959970607934142974736675784325993445942031372107342103852
```

```
import gmpy2
import binascii
l = gmpy2.invert(p,q)
mp = gmpy2.powmod(c,dp,p)
mq = gmpy2.powmod(c,dq,q)

m = ((l*(mp-mq))%q)*p+mp
print(binascii.unhexlify(hex(m)[2:]))
```

4.已知(e,dp, n, c)，求m。

示例：【BUUCTF-RSA2】

解题思路：

$e * d =$ $-1)($

脚本代码如下：

```
import gmpy2
import binascii
e = 65537
n = 24825400785152624117772152669890180298583276617622160961225887737162058006043310153832803030521991869764361981420093067961210988553380133534844502375167047843707305554472428068473329805159916766030364518314616149748535863368149212966880240206579778990555048954764511878726660192942972413316776846530966590611 3
dp = 90507449805234690464302513287951833069192517457305400462187725331868267505542197094355201669552856036483444630319693920705664292714809329037444021050365 7

c = 14042367097625269680753367358620940057566428210068411978420352712452118899640382659743688376604187906749428095741020195893573736038080184545382929399743341418883872575179626170262202858721156035336284719106030657851051138096516213347269871306359262102895916707278148256267368309059052121421807116028766518075 1

for i in range(1,e):
    if (e*dp-1)%i == 0 and n%((e*dp-1)//i+1)==0:
        q = n//((e*dp-1)//i+1)
        phi = (q-1)*((e*dp-1)//i)
        d = gmpy2.invert(e,phi)
        m = gmpy2.powmod(c,d,n)

print(binascii.unhexlify(hex(m)[2:]))
```

5.已知(n, e1, e2,c1,c2)，求m。

示例：【BUUCTF-RSA3】

解题思路： 涉及知识点为**共模攻击**

$$c1\equiv m \mod n$$

$$c1 * c2 \mod n$$

脚本代码如下：

```
import gmpy2
import binascii

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

s = gmpy2.gcdext(e1,e2)
a = s[1]
b = s[2]

if a<0:
    a = -a
    c1 = gmpy2.invert(c1,n)
else:
    b = -b
    c2 = gmpy2.invert(c2,n)

m = (gmpy2.powmod(c1,a,n)*gmpy2.powmod(c2,b,n))%n

print(binascii.unhexlify(hex(m)[2:]))
```

6.已知(e,n1,c1,n2,c2)，求m。

示例：【CTF秀-easyrsa2】

解题思路： 两组数中e相同，n，c不同，求出n1与n2的最大公因数即为p，之后就可以得到q和d，从而求解m。

```
import gmpy2
import binascii

e = 65537
n1 = 23686563925537577753047229040754282953352221724154495390687358877775380147605152455537988563490716943872517593212858326146811511103311865753018329109314623702207073882884251372553225986112006827111351501044972239272200616871716325265416115038890805114829315111950319183189591283821793237999044427887934536835813526748759612963103377803089900662509399569819785571492828112437312659229879806168758843603248823629821851053775458651933952183988482163950039248487270453888288427540305542824179951734412044985364866532124803746008139763081886781361488304666575456680411806505094963425401175510416864929601220556158569443747
c1 = 16274841422378976139446078282689811939114174080648245407119451920356490881041330381474002240705884103351906626822311899975800846804242094953030780612051228489046483192196465887209940192492798634629810153294837247478239915137141724788863067032900448717811583933041473010587060037933578469220869949527634859992827415952040086638479635394220963433914645270685990469462793090372128599313033355074551460013903265506685316654932452938390098324686683908202826649840663990514032279900680322263822221734780785058882387495832379806436984050056892479229013422041428338754095051808479432121263024823584457686626082787317500648 15

n2 = 22257605320525584078180889073523223973924192984353847137164605186956629675938929585386392327672065524338176402496414014083816446508860530887742583338880317478862512306633061601510404960095143941320847160562050524072860211772522478494742213643890027443992183362678970426046765630946644339093149139143388752794932806956589884503569175226850419271095336798456238899009883100793515744579945854481430194879360765346236418019384644095257242811629393164402498261066077339304875212250897918420427814000142751282805980632089867108525335488018940091698609890995252413007073725850396076272027183422297684667565712022199054289711
c2 = 27426006954418365594695537028310983759486419154091069761578403779781239120073987536234611126597962099188669854804719113933627977536244795376468025104204150394618321180188490305806752498175769268583635416831357772393220027418201459442861091720662598437667557952559131899024036447211385549359914398938505896778496392630805285991975957059275354309424631848916894100780590904746826948864200222306576611579938759316009327638246187734200772736171062976601951799220188753991743468634047104201664970171964245861165359157129651471417750265498706363281956907742599901892866658446412891084748349737107304261050473189593079950 62

p = gmpy2.gcd(n1,n2)
q = n1 // p
phi = (p-1)*(q-1)

d = gmpy2.invert(e,phi)
m = gmpy2.powmod(c1,d,n1)

print(binascii.unhexlify(hex(m)[2:]))
```

7.已知(p+q,p-q, e,c)，求m。

示例：[BJDCTF 2nd]rsa0

解题思路：

```
import gmpy2
import libnum
e=16300321

a=213504305120595601355365067258861927916529217823427578155371278098188960961678617774328629887216249471767301211279462500447131879443770408269780926757458896

b=155326276588878943320139654398713444218448710086505615240962817059108805133745303293690183411861114641439571476835079409025106495391344048567616700680937 0

c=5450514571643701723678366908952545856999647466774795243500703960975257928359894435344292368060236063234297507975744848471202579371935247580634316381888113370759031842650946802970689781074296506888143181072094703519663295002432404078880113319447415040638069162473911533401588674055792133930535716709839274490 5

p = (a+b)//2
q = (a-b)//2

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

m = gmpy2.powmod(c,d,p*q)
print(libnum.n2s(m))
```

7.已知(e,n,c)，求m。

示例：【CTF秀-easyrsa1】

解题思路：

可以分解n得到p，q，在线分解大整数网址http://www.factordb.com/index.php 。

脚本代码如下：

```
import gmpy2
import binascii

e = 65537
n = 14559255297343581054614065322599117908073476164649910653018 47
c = 6938037105791424619260676068615223322565950336631933206500 9
p = 12011470594385307868353651945 67
q = 12121126370778629171921919138 41

phi = (p-1)*(q-1)
d = gmpy2.invert(e,phi)
m = gmpy2.powmod(c,d,n)

print(binascii.unhexlify(hex(m)[2:]))
```

8.已知(e,n,c)，求m。（低加密指数攻击）

示例：【CTF秀-easyrsa4】

解题思路： 题中e=3相对于n，c来说极小，故可知是**低加密指数攻击**。

①

当$m < n$时，$c = m^e$，直接将c开三次方即可得到m。

脚本代码如下：

```
import gmpy2
import binascii

e = 3
n = 189700537286166093664582860677312887490222649591584037583579859153933831179636938275688099257706793537656248108049043822788455264989814223463194179384348615582913667385420791651697362325586878217099373465034807562814897758594392546144724250175540511777251430681221859615526706462752290095315286785482518734210766916508275078298592993002726832239592676612886016198459544663651340775476998197344653213457584169572656821758642272735062507073117757979798340909070208630994679071199579678941722227477621516745009373563920297414877818366750215020226517547121383368598844556881961208526891778071894547257376536558816394575476 1
c = 150409620528139732054476072280993764527079006992643377862720337847060335153837950368208902491767027770946661

i = 0
while True:
    if gmpy2.iroot((c+i*n),3)[1] == True:
        m = gmpy2.iroot((c+i*n),3)[0]
        break
    i += 1

print(binascii.unhexlify(hex(m)[2:]))
```

9.已知(e,n,c)，求m。（低解密指数攻击）

示例：【CTF秀-easyrsa5】

解题思路： 题中e很大，故可知是**低解密指数攻击**。

可以使用破解脚本：求出d的值，文件下载地址https://github.com/pablocelayes/rsa-wiener-attack

（注意，这里要将破解脚本和rsa-wiener-attack的py文件放在同一个目录下）

脚本代码如下：

```
import gmpy2
import binascii
import RSAwienerHacker

e = 284100478693161642327695712452505468891794410301906465434604643365855064101922252698327584524956955373553355814138784402605517536436009073372339264422522610010012877243630454889127160056358637599704871937659443985644871453345576728414422489075791739731547285138648307770775155312545928721094602949588237119345
n = 468459887279781789188886188573017406548524570309663876064881031936564733341508945283407498306248145591559137207097347130203582813352382018491852922849186827279111555223982032271701972642438224730082216672110316142528108239708171781850491578433309964093293907697072741538649347894863899103340030347858867705231
c = 350429162418561525458539070186062788413426454598897326594935655762503536409897624028778814302849485850451243934994919418665502401195173255808119461832488053305530748068788500746791135053620550583421369214031040191188956888321397450005528879987036183922578645840167009612661903399312419253694928377398939392827

d = RSAwienerHacker.hack_RSA(e,n)
m = gmpy2.powmod(c,d,n)

print(binascii.unhexlify(hex(m)[2:]))
```