Hgame-2019几道有趣的RSA

原创

Gard3nia 于 2019-03-03 15:28:12 发布 841 收藏 1

分类专栏： Writeup 文章标签： Crypto Writeup

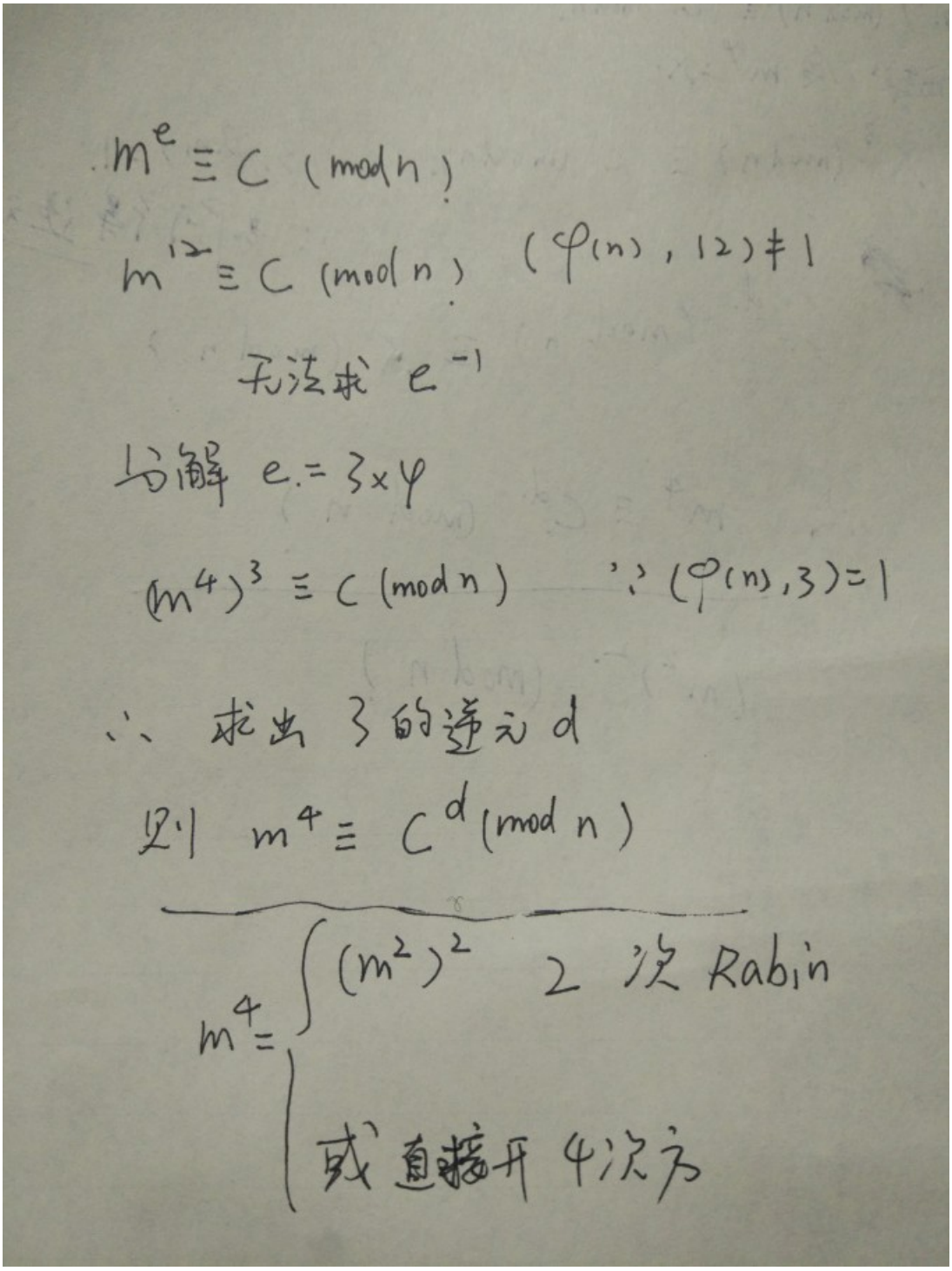
本文链接：https://blog.csdn.net/Gar_denia/article/details/88088708

babyRSA

```
e = 12
p = 583800044303078033678069964607731236037903057890983844889520562066615768274527
q = 818595269757200606493800981936716128012005050291270765394576801554876696228 67
ciphertext =
206087215323690202467878926681944491769659156726458690815919286163630886447291570510196171585626143608988384615185921752409380788006476576337410136447460
算出的m转化成字符串
```

说实话这题本来以为很简单的，想直接求私钥，直接求解，没想到算出来φ(n)和e=12不互素，无法求出e对应的公钥，所以直接解是没有办法的；

解法如下：


$m^e \equiv C \pmod n$
$m^{12} \equiv C \pmod n$ $(\varphi(n), 12) \neq 1$
无法求 e^{-1}
分解 $e = 3 \times 4$
$(m^4)^3 \equiv C \pmod n$ $\because (\varphi(n), 3) = 1$
∴ 求出 3 的逆元 d
则 $m^4 \equiv C^d \pmod n$
$m^4 =$
$(m^2)^2$ 2 次 Rabin
或直接开 4 次方

解题脚本如下：

```python
#coding:utf-8
import gmpy2
import math
def rabin_decrypt(c, p, q, e=2):
    n = p * q
    mp = pow(c, (p + 1) / 4, p)
    mq = pow(c, (q + 1) / 4, q)
    yp = gmpy2.invert(p, q)
    yq = gmpy2.invert(q, p)
    r = (yp * p * mq + yq * q * mp) % n
    rr = n - r
    s = (yp * p * mq - yq * q * mp) % n
    ss = n - s
    return (r, rr, s, ss)

def main():
 e = 12
 p = 583800044303078033678069964607731236037903057890983844889520562066157682745 27
 q = 818595269757200606493800981936716128012005050291270765394576801554876696228 67
 ciphertext = 206087215323690202467878926681944491769659156726458690815919286163630886447291570510196171585626143608988384615185921752409380788006476576337410136447460
 n=p*q
 ol=(p-1)*(q-1)
 d=gmpy2.invert(3,ol)
 ciphertext1=pow(ciphertext,d,n)
 ans=rabin_decrypt(ciphertext1,p,q,e=2)
 for i in ans:
  temp=rabin_decrypt(i,p,q,e=2)
  for x in temp:
   flag='{:x}'.format(x).decode('hex')
   if 'hgame' in flag:
    print flag


if __name__ == '__main__':
 main()
```

easy_rsa

m为17位十进制数，提交格式hgame{m}
e1:0x33240
e2:0x3e4f
n:0x9439682bf1b4ab48c43c524778c579cc844b60872275725c1dc893b5bcb358b9f136e4dab2a06318bb0c80e202a14bc54ea334519bec023934e01e9378abf329893f3870979e9f2f2be8fff4df931216a77007a2509f49f697bf286285e97fac5dc6e4a164b5c2cc430887b18136437ba67777bda05aafdeaf918221c812b4c7d1665238f84ab0fab7a77fcae92a0596e58343be7a8e6e75a5017c63a67eb11964970659cd6110e9ec6502288e9e443d86229ef2364dfecb63e2d90993a75356854eb874797340eece1b19974e86bee07019610467d44ec595e04af02b574a97fa98bdb2e779871c804219cab715f4a80fef7f8fb52251d86077560b39c1c2a1
c1:0x7c7f315a3ebbe305c1ad8bd2f73b1bb8e300912b6b8ba1b331ac2419d3da5a9a605fd62915c11f8921c450525d2efda7d48f1e503041498f4f0676760b43c770ff2968bd942c7ef95e401dd7facbd4e5404a0ed3ad96ae505f87c4e12439a2da636f047d84b1256c0e363f63373732cbaf24bda22d931d001dcca124f5a19f9e28608ebd90161e728b782eb67deeba4cc81b6df4e7ee29a156f51a0e5148618c6e81c31a91036c982debd1897e6f3c1e5e248789c933a4bf30d0721a18ab8708d827858b77c1a020764550a7fe2ebd48b6848d9c4d211fd853b7a02a859fa0c72160675d832c94e0e43355363a2166b3d41b8137100c18841e34ff52786867d
c2:0xf3a8b9b739196ba270c8896bd3806e9907fca2592d28385ef24afadc2a408b7942214dad5b9e14808ab988fb15fbd93e725edcc0509ab0dd1656557019ae93c38031d2a7c84895ee3da1150eda04cd2815ee3debaa7c2651b62639f785f6cabf83f93bf3cce7778ab369631ea6145438c3cd4d93d6f2759be3cc187651a33b3cc4c3b477604477143c32dfff62461fdfd9f8aa879257489bbf977417ce0fbe89e3f2464475624aafef57dd9ea60339793c69b53ca71d745d626f45e6a7beb9fcbd9d1a259433d36139345b7bb4f392e78f1b5be0d2c56ad50767ee851fac670946356b3c05d0605bf243b89c7e683cc75030b71633632fb95c84075201352d6
c1=pow(m, e1, n)
c2=pow(m, e2, n)

先转化为10进制，发现e1=209472，e2=15951，e1,e2不互素，最大公因数为3；本想用共模攻击，但是e1和e2不互素，没办法达到共模攻击的条件，但是仔细推来发现：

gcd(e1,e2)=3，则gcd(e1/3,e2/3)=1
所以令a1=e1/3,a2=e2/3，则gcd(a1,a2)=1
所以存在r*a1+s*a2=1，所以r*e1+s*e2=3 (扩展欧几里得算法)
则(c1^r) * (c2^s) =(m^(r*e1)) * (m^(s*e2)) =m^(r*e1+s*e2) = m^3 //共模攻击
所以到此为止：(m^3) mod n = (c1^r) * (c2^s) mod n 为已知数
所以直接低指数攻击即可

上解题脚本：

```python
#coding:utf-8
import gmpy2
def same_attack(e1,e2,mess1,mess2,n):
 gcd,s,t=gmpy2.gcdext(e1,e2)
 if s<0:
  s=-s
  mess1=gmpy2.invert(mess1,n)
 if t<0:
  t=-t
  mess2=gmpy2.invert(mess2,n)
 mess=gmpy2.powmod(mess1,s,n)*gmpy2.powmod(mess2,t,n)%n
 return mess

def small_attack(c,n,e):
 i=0
 while 1:
     if(gmpy2.iroot(c+i*n, e)[1] == 1):
         x = gmpy2.iroot(c+i*n, e)[0]
         print x
         break
  i += 1

def main():
 e1="0x33240"
 e2="0x3e4f"
 n="0x9439682bf1b4ab48c43c524778c579cc844b60872275725c1dc893b5bcb358b9f136e4dab2a06318bb0c80e202a14bc54ea334519b
ec023934e01e9378abf329893f3870979e9f2f2be8fff4df931216a77007a2509f49f697bf286285e97fac5dc6e4a164b5c2cc430887b181
36437ba67777bda05aafdeaf918221c812b4c7d1665238f84ab0fab7a77fcae92a0596e58343be7a8e6e75a5017c63a67eb11964970659cd
6110e9ec6502288e9e443d86229ef2364dfecb63e2d90993a75356854eb874797340eece1b19974e86bee07019610467d44ec595e04af02b
574a97fa98bdb2e779871c804219cab715f4a80fef7f8fb52251d86077560b39c1c2a1"
 c1="0x7c7f315a3ebbe305c1ad8bd2f73b1bb8e300912b6b8ba1b331ac2419d3da5a9a605fd62915c11f8921c450525d2efda7d48f1e503
041498f4f0676760b43c770ff2968bd942c7ef95e401dd7facbd4e5404a0ed3ad96ae505f87c4e12439a2da636f047d84b1256c0e363f633
73732cbaf24bda22d931d001dcca124f5a19f9e28608ebd90161e728b782eb67deeba4cc81b6df4e7ee29a156f51a0e5148618c6e81c31a9
1036c982debd1897e6f3c1e5e248789c933a4bf30d0721a18ab8708d827858b77c1a020764550a7fe2ebd48b6848d9c4d211fd853b7a02a8
59fa0c72160675d832c94e0e43355363a2166b3d41b8137100c18841e34ff52786867d"
 c2="0xf3a8b9b739196ba270c8896bd3806e9907fca2592d28385ef24afadc2a408b7942214dad5b9e14808ab988fb15fbd93e725edcc05
09ab0dd1656557019ae93c38031d2a7c84895ee3da1150eda04cd2815ee3debaa7c2651b62639f785f6cabf83f93bf3cce7778ab369631ea
6145438c3cd4d93d6f2759be3cc187651a33b3cc4c3b477604477143c32dfff62461fdfd9f8aa879257489bbf977417ce0fbe89e3f246447
5624aafef57dd9ea60339793c69b53ca71d745d626f45e6a7beb9fcbd9d1a259433d36139345b7bb4f392e78f1b5be0d2c56ad50767ee851
fac670946356b3c05d0605bf243b89c7e683cc75030b71633632fb95c84075201352d6"
 e1=int(e1,16)/3
 e2=int(e2,16)/3
 n=int(n,16)
 c1=int(c1,16)
 c2=int(c2,16)
 # print gmpy2.gcd(e1,e2)
 ans=same_attack(e1,e2,c1,c2,n)
 # print ans
 small_attack(ans,n,3)

if __name__ == '__main__':
 main()
```

加上“hgame”即可得到flag：

```
59594981651654789
[Finished in 0.2s]
```