Writeup challenge: RSA1,RSA2,AES, Chemistry Class

RSA1

Solver: cod201

Công cụ cần thiết:

- RsaCtfTool: công cụ phá mã RSA khi có weak publickey
  - Link: https://github.com/Ganapati/RsaCtfTool
- Python3,openssl,Kali

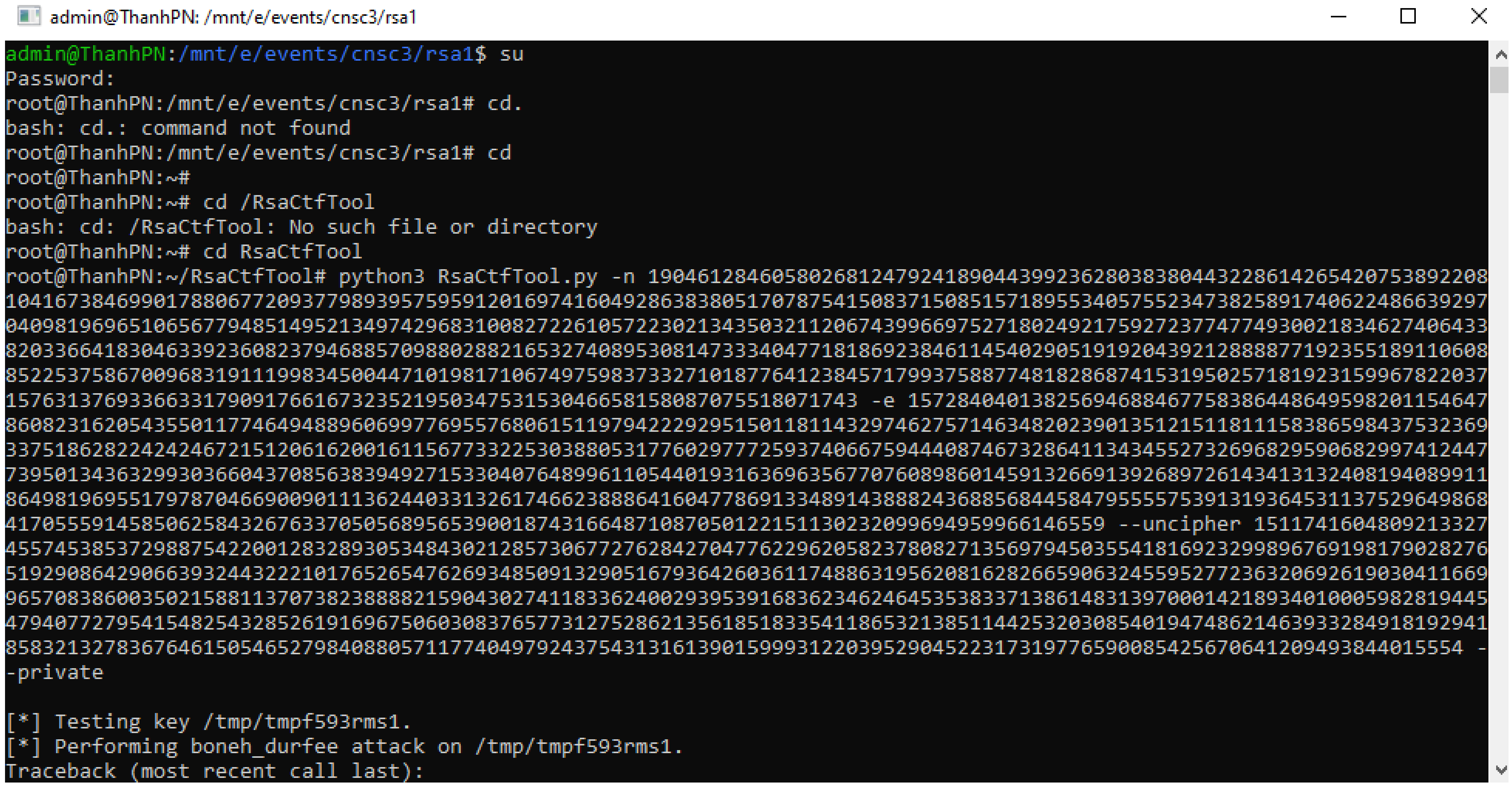
Cú pháp khi dùng RsaCtfTool:

```
python3  RsaCtfTool.py -n <n> -e <e> --uncipher <c> [--private] [--attack  <type>]
```

Với :

- <n> là modulus
- <e> là số mũ công khai
- <c> là văn bản bị mã hóa
- [--private] là tham số để yêu cầu hiện privatekey nếu giải được
- <type> là chỉ định attack khi biết được phương pháp crack

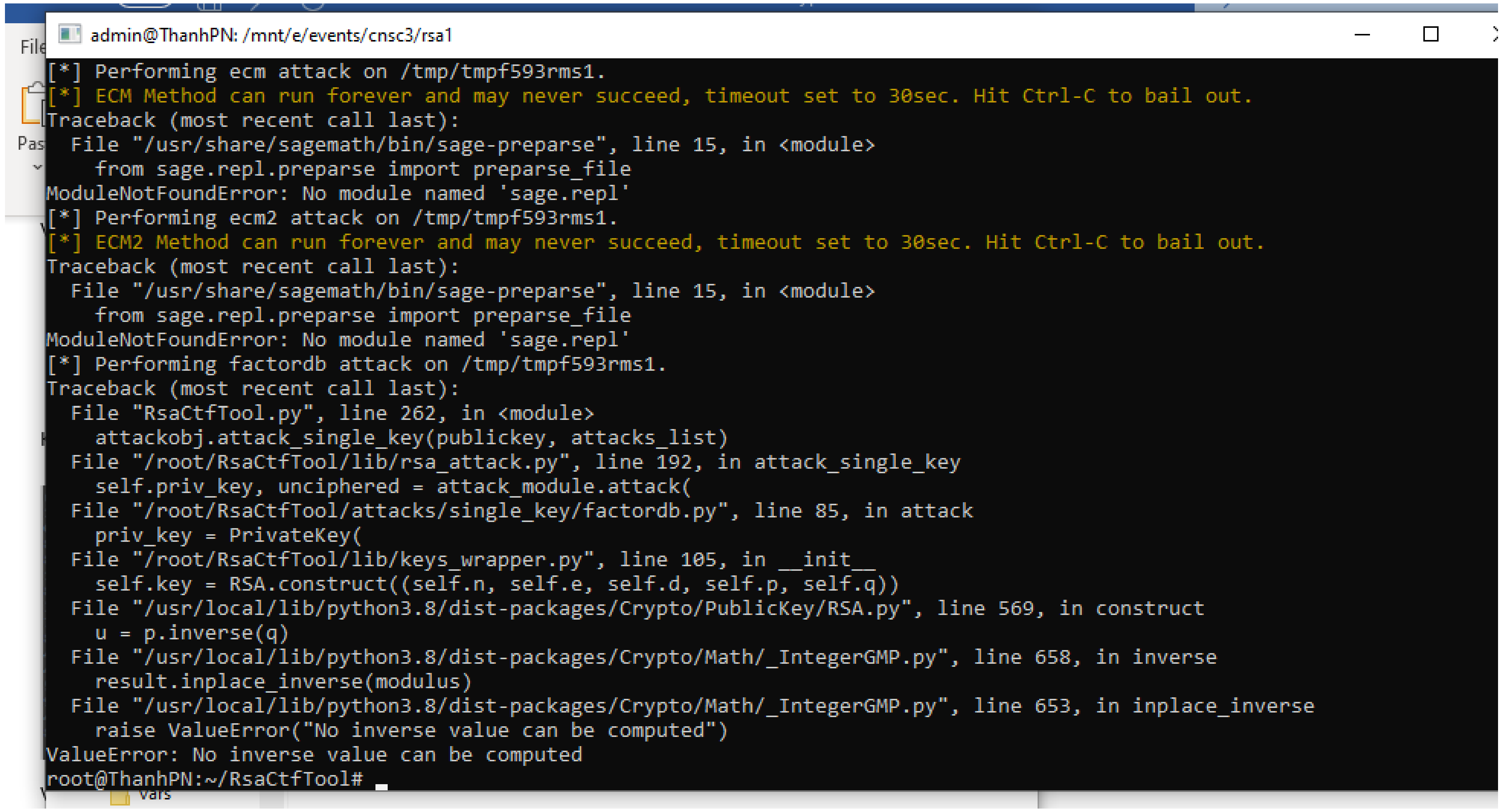
Khi mở file challenge ra, ta thấy có n,e1,c1. Thử dùng tool nào:



admin@ThanhPN: /mnt/e/events/cnsc3/rsa1

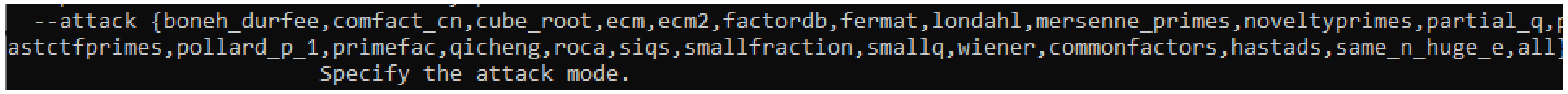
```
admin@ThanhPN:/mnt/e/events/cnsc3/rsa1$ su
Password:
root@ThanhPN:/mnt/e/events/cnsc3/rsa1# cd.
bash: cd.: command not found
root@ThanhPN:/mnt/e/events/cnsc3/rsa1# cd
root@ThanhPN:~#
root@ThanhPN:~# cd /RsaCtfTool
bash: cd: /RsaCtfTool: No such file or directory
root@ThanhPN:~# cd RsaCtfTool
root@ThanhPN:~/RsaCtfTool# python3 RsaCtfTool.py -n 19046128460580268124792418904439923628038380443228614265420753892208
104167384699017880677209377989395759591201697416049286383805170787541508371508515718955340575523473825891740622486639297
040981969651065677948514952134974296831008272261057223021343503211206743996697527180249217592723774774930021834627406433
820336641830463392360823794688570988028821653274089530814733340477181869238461145402905191920439212888877192355189110608
852253758670096831911199834500447101981710674975983733271018776412384571799375887748182868741531950257181923159967822037
157631376933663317909176616732352195034753153046658158087075518071743 -e 15728404013825694688467758386448649598201154647
860823162054355011774649488960699776955768061511979422292951501181143297462757146348202390135121511811158386598437532369
337518628224242467215120616200161156773322530388053177602977725937406675944408746732864113434552732696829590682997412447
739501343632993036604370856383949271533040764899611054401931636963567707608986014591326691392689726143413132408194089911
864981969551797870466900901113624403313261746623888641604778691334891438882436885684458479555575391319364531137529649868
41705559145850625843267633705056895653900187431664871087050122151130232099694959966146559 --uncipher 1511741604809213327
455745385372988754220012832893053484302128573067727628427047762296205823780827135697945035541816923299896769198179028276
519290864290663932443222101765265476269348509132905167936426036117488631956208162826659063245595277236320692619030411669
965708386003502158811370738238888215904302741183362400293953916836234624645353833713861483139700014218934010005982819445
479407727954154825432852619169675060308376577312752862135618518335411865321385114425320308540194748621463933284918192941
8583213278367646150546527984088057117740497924375431316139015999312203952904522317319776590085425670641209493844015554 -
-private

[*] Testing key /tmp/tmpf593rms1.
[*] Performing boneh_durfee attack on /tmp/tmpf593rms1.
Traceback (most recent call last):
```

Vì module factordb bị lỗi(như ảnh sau)



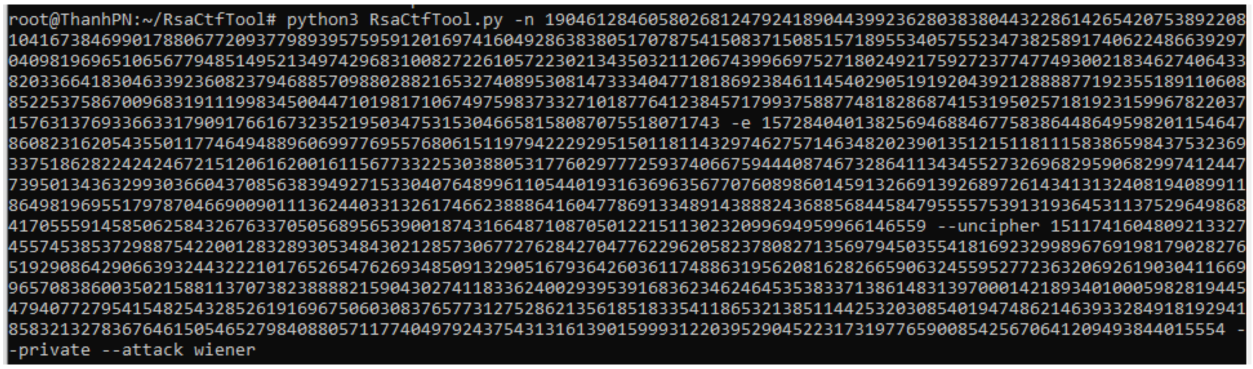
```
[*] Performing ecm attack on /tmp/tmpf593rms1.
[*] ECM Method can run forever and may never succeed, timeout set to 30sec. Hit Ctrl-C to bail out.
Traceback (most recent call last):
  File "/usr/share/sagemath/bin/sage-preparse", line 15, in <module>
    from sage.repl.preparse import preparse_file
ModuleNotFoundError: No module named 'sage.repl'
[*] Performing ecm2 attack on /tmp/tmpf593rms1.
[*] ECM2 Method can run forever and may never succeed, timeout set to 30sec. Hit Ctrl-C to bail out.
Traceback (most recent call last):
  File "/usr/share/sagemath/bin/sage-preparse", line 15, in <module>
    from sage.repl.preparse import preparse_file
ModuleNotFoundError: No module named 'sage.repl'
[*] Performing factordb attack on /tmp/tmpf593rms1.
Traceback (most recent call last):
  File "RsaCtfTool.py", line 262, in <module>
    attackobj.attack_single_key(publickey, attacks_list)
  File "/root/RsaCtfTool/lib/rsa_attack.py", line 192, in attack_single_key
    self.priv_key, unciphered = attack_module.attack(
  File "/root/RsaCtfTool/attacks/single_key/factordb.py", line 85, in attack
    priv_key = PrivateKey(
  File "/root/RsaCtfTool/lib/keys_wrapper.py", line 105, in __init__
    self.key = RSA.construct((self.n, self.e, self.d, self.p, self.q))
  File "/usr/local/lib/python3.8/dist-packages/Crypto/PublicKey/RSA.py", line 569, in construct
    u = p.inverse(q)
  File "/usr/local/lib/python3.8/dist-packages/Crypto/Math/_IntegerGMP.py", line 658, in inverse
    result.inplace_inverse(modulus)
  File "/usr/local/lib/python3.8/dist-packages/Crypto/Math/_IntegerGMP.py", line 653, in inplace_inverse
    raise ValueError("No inverse value can be computed")
ValueError: No inverse value can be computed
root@ThanhPN:~/RsaCtfTool# _
```

Mình sẽ thử tất cả loại attack dưới đây để tìm ra privatekey



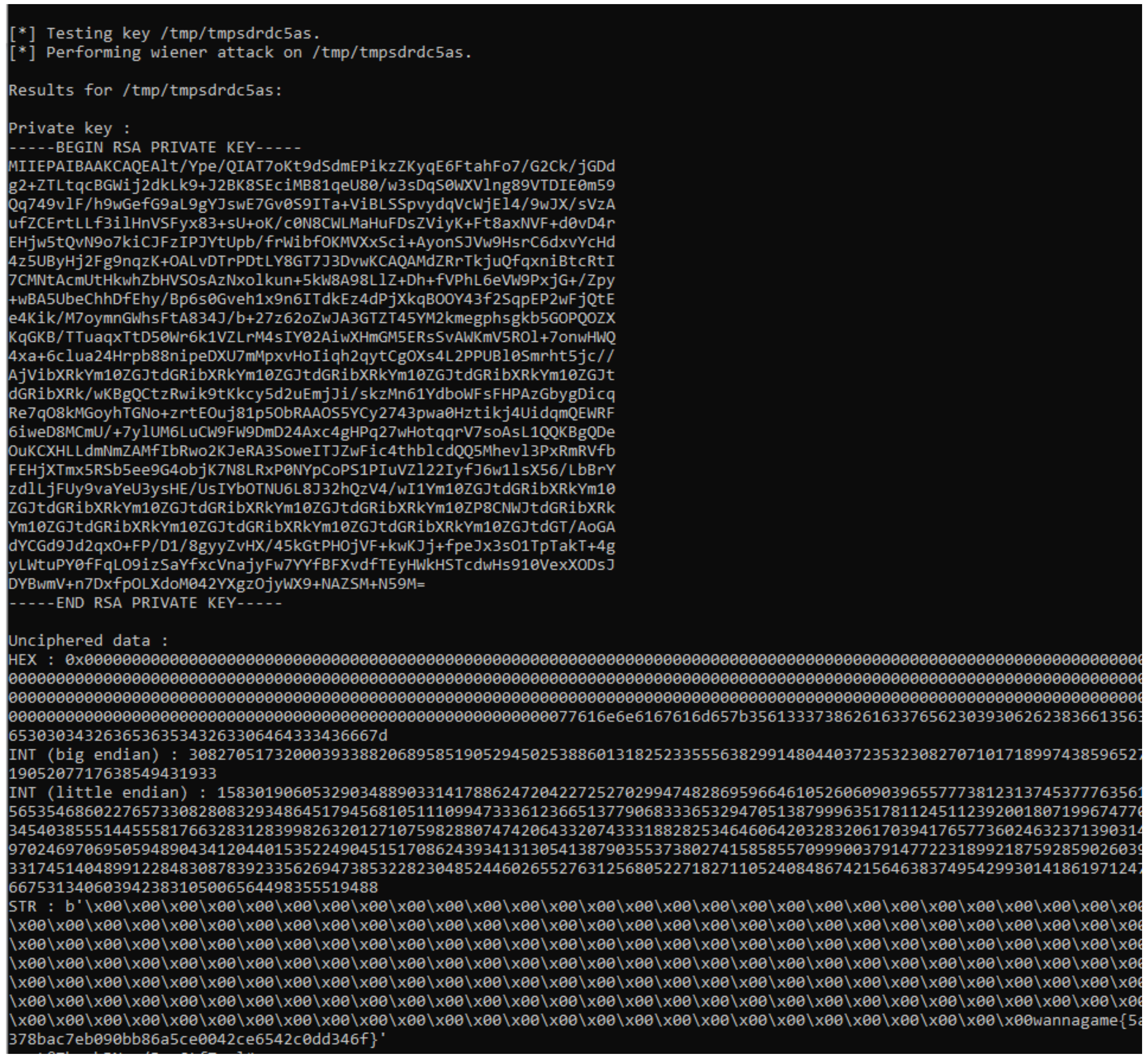
```
  --attack {boneh_durfee,comfact_cn,cube_root,ecm,ecm2,factordb,fermat,londahl,mersenne_primes,noveltyprimes,partial_q,p
astctfprimes,pollard_p_1,primefac,qicheng,roca,siqs,smallfraction,smallq,wiener,commonfactors,hastads,same_n_huge_e,all}
                        Specify the attack mode.
```

Thì attack wiener là có ra privatekey;



```
root@ThanhPN:~/RsaCtfTool# python3 RsaCtfTool.py -n 19046128460580268124792418904439923628038380443228614265420753892208
104167384699017880677209377989395759591201697416049286383805170787541508371508515718955340575523473825891740622486639297
040981969651065677948514952134974296831008272261057223021343503211206743996697527180249217592723774774930021834627406433
820336641830463392360823794688570988028821653274089530814733340477181869238461145402905191920439212888877192355189110608
852253758670096831911199834500447101981710674975983733271018776412384571799375887748182868741531950257181923159967822037
157631376933663317909176616732352195034753153046658158087075518071743 -e 15728404013825694688467758386448649598201154647
860823162054355011774649488960699776955768061511979422292951501181143297462757146348202390135121511811158386598437532369
337518628224242467215120616200161156773322530388053177602977725937406675944408746732864113434552732696829590682997412447
739501343632993036604370856383949271533040764899611054401931636963567707608986014591326691392689726143413132408194089911
864981969551797870466900901113624403313261746623888641604778691334891438882436885684458479555575391319364531137529649868
41705559145850625843267633705056895653900187431664871087050122151130232099694959966146559 --uncipher 1511741604809213327
455745385372988754220012832893053484302128573067727628427047762296205823780827135697945035541816923299896769198179028276
519290864290663932443222101765265476269348509132905167936426036117488631956208162826659063245595277236320692619030411669
965708386003502158811370738238888215904302741183362400293953916836234624645353833713861483139700014218934010005982819445
479407727954154825432852619169675060308376577312752862135618518335411865321385114425320308540194748621463933284918192941
8583213278367646150546527984088057117740497924375431316139015999312203952904522317319776590085425670641209493844015554 -
-private --attack wiener
```

Và đây là kết quả



```
[*] Testing key /tmp/tmpsdrdc5as.
[*] Performing wiener attack on /tmp/tmpsdrdc5as.

Results for /tmp/tmpsdrdc5as:

Private key :
-----BEGIN RSA PRIVATE KEY-----
MIIEPAIBAAKCAQEAlt/Ype/QIAT7oKt9dSdmEPikzZKyqE6FtahFo7/G2Ck/jGDd
g2+ZTLtqcBGWij2dkLk9+J2BK8SEciMB81qeU80/w3sDqS0WXVlng89VTDIE0m59
Qq749vlF/h9wGefG9aL9gYJswE7Gv0S9ITa+ViBLSSpvydqVcWjEl4/9wJX/sVzA
ufZCErtLLf3ilHnVSFyx83+sU+oK/c0N8CWLMaHuFDsZViyK+Ft8axNVF+d0vD4r
EHjw5tQvN9o7kiCJFzIPJYtUpb/frWibfOKMVXxSci+AyonSJVw9HsrC6dxvYcHd
4z5UByHj2Fg9nqzK+OALvDTrPDtLY8GT7J3DvwKCAQAMdZRrTkjuQfqxniBtcRtI
7CMNtAcmUtHkwhZbHVSOsAzNxolkun+5kW8A98LlZ+Dh+fVPhL6eVW9PxjG+/Zpy
+wBA5UbeChhDfEhy/Bp6s0Gveh1x9n6ITdkEz4dPjXkqBOOY43f2SqpEP2wFjQtE
e4Kik/M7oymnGWhsFtA834J/b+27z62oZwJA3GTZT45YM2kmegphsgkb5GOPQOZX
KqGKB/TTuaqxTtD50Wr6k1VZLrM4sIY02AiwXHmGM5ERsSvAWKmV5ROl+7onwHWQ
4xa+6clua24Hrpb88nipeDXU7mMpxvHoIiqh2qytCgOXs4L2PPUBl0Smrht5jc//
AjVibXRkYm10ZGJtdGRibXRkYm10ZGJtdGRibXRkYm10ZGJtdGRibXRkYm10ZGJt
dGRibXRk/wKBgQCtzRwik9tKkcy5d2uEmjJi/skzMn61YdboWFsFHPAzGbygDicq
Re7qO8kMGoyhTGNo+zrtEOuj81p5ObRAAOS5YCy2743pwa0Hztikj4UidqmQEWRF
6iweD8MCmU/+7ylUM6LuCW9FW9DmD24Axc4gHPq27wHotqqrV7soAsL1QQKBgQDe
OuKCXHLLdmNmZAMfIbRwo2KJeRA3SoweITJZwFic4thblcdQQ5Mhevl3PxRmRVfb
FEHjXTmx5RSb5ee9G4objK7N8LRxP0NYpCoPS1PIuVZl22IyfJ6w1lsX56/LbBrY
zdlLjFUy9vaYeU3ysHE/UsIYbOTNU6L8J32hQzV4/wI1Ym10ZGJtdGRibXRkYm10
ZGJtdGRibXRkYm10ZGJtdGRibXRkYm10ZP8CNWJtdGRibXRk
Ym10ZGJtdGRibXRkYm10ZGJtdGRibXRkYm10ZGJtdGRibXRkYm10ZGJtdGT/AoGA
dYCGd9Jd2qxO+FP/D1/8gyyZvHX/45kGtPHOjVF+kwKJj+fpeJx3sO1TpTakT+4g
yLWtuPY0fFqLO9izSaYfxcVnajyFw7YYfBFXvdfTEyHWkHSTcdwHs910VexXODsJ
DYBwmV+n7DxfpOLXdoM042YXgzOjyWX9+NAZSM+N59M=
-----END RSA PRIVATE KEY-----

Unciphered data :
HEX : 0x0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000077616e6e6167616d657b356133373862616337656230393062623836613563
653030343263653635343263306464333436667d
INT (big endian) : 30827051732000393388206895851905294502538860131825233555638299148044037235323082707101718997438596527
1905207717638549431933
INT (little endian) : 15830190605329034889033141788624720422725270299474828695966461052606090396557773812313745377763561
565354686022765733082808329348645179456810511109947333612366513779068333653294705138799963517811245112392001807199674776
345403855514455581766328312839982632012710759828807474206433207433318828253464606420328320617039417657736024632371390314
970246970695059489043412044015352249045151708624393413130541387903553738027415858557099900379147722318992187592859026039
331745140489912284830878392335626947385322823048524460265527631256805227182711052408486742156463837495429930141861971247
667531340603942383105006564498355519488
STR : b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00wannagame{5a
378bac7eb090bb86a5ce0042ce6542c0dd346f}'
```

Vậy là có flag rồi hihi, và nó xuất luôn cho mình private key nè. Lưu privatekey lại backup luôn.

RSA2

Solver: cod201, revirven

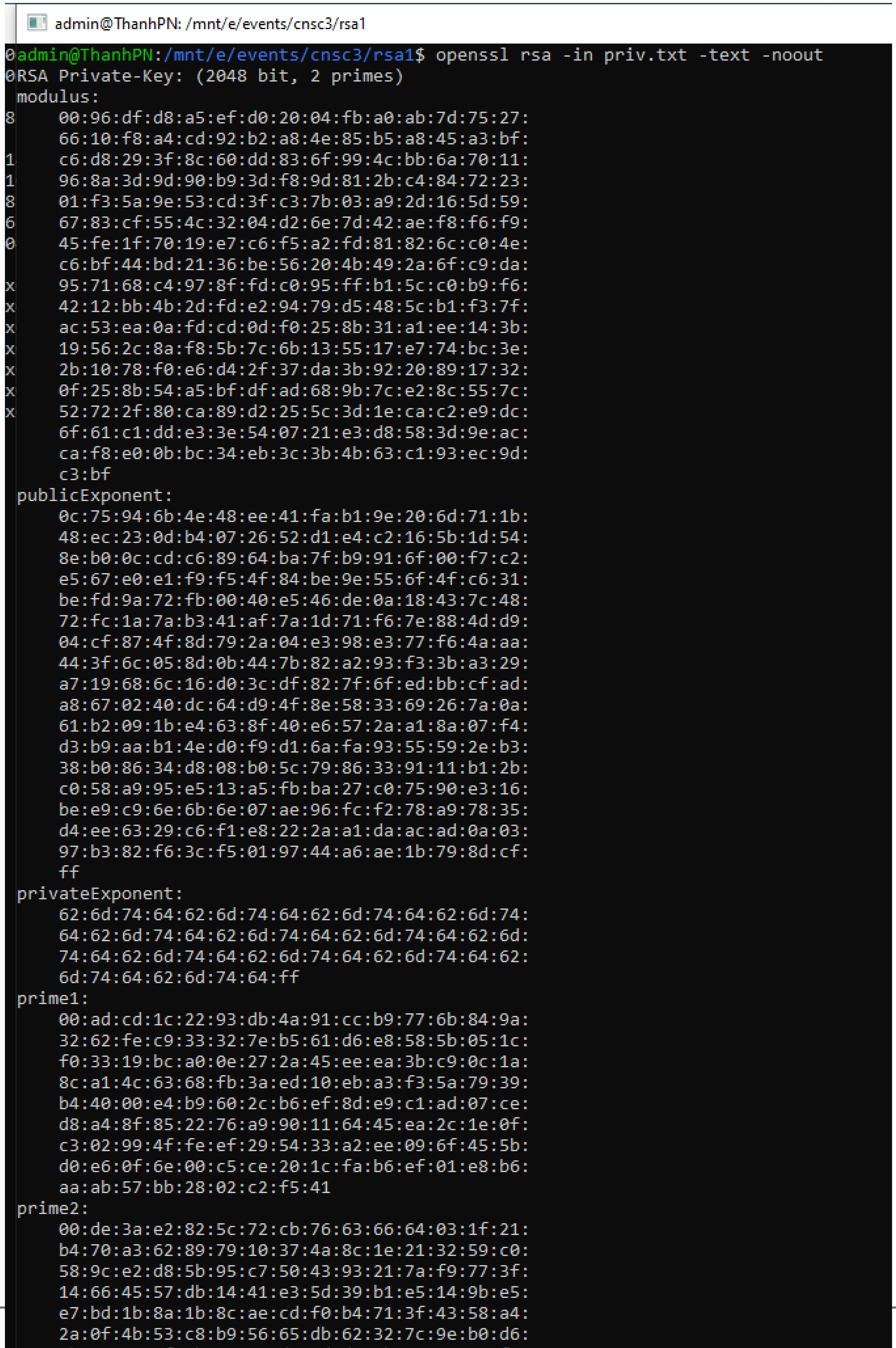
Các công cụ cần thiết: Như challenge trên

Khi mở file challenge ra, đập vào mắt mình chính là… Challenge này dùng lại n của bài RSA1.

Đầu tiên mình sẽ trích thông tin từ private key ra p với q để có thể giải mã bài này

openssl rsa -in priv.txt -text -noout

với priv.txt là tên file privatekey của bài RSA1.



admin@ThanhPN: /mnt/e/events/cnsc3/rsa1

```
0admin@ThanhPN:/mnt/e/events/cnsc3/rsa1$ openssl rsa -in priv.txt -text -noout
0RSA Private-Key: (2048 bit, 2 primes)
 modulus:
8    00:96:df:d8:a5:ef:d0:20:04:fb:a0:ab:7d:75:27:
     66:10:f8:a4:cd:92:b2:a8:4e:85:b5:a8:45:a3:bf:
1    c6:d8:29:3f:8c:60:dd:83:6f:99:4c:bb:6a:70:11:
1    96:8a:3d:9d:90:b9:3d:f8:9d:81:2b:c4:84:72:23:
8    01:f3:5a:9e:53:cd:3f:c3:7b:03:a9:2d:16:5d:59:
6    67:83:cf:55:4c:32:04:d2:6e:7d:42:ae:f8:f6:f9:
0    45:fe:1f:70:19:e7:c6:f5:a2:fd:81:82:6c:c0:4e:
     c6:bf:44:bd:21:36:be:56:20:4b:49:2a:6f:c9:da:
x    95:71:68:c4:97:8f:fd:c0:95:ff:b1:5c:c0:b9:f6:
x    42:12:bb:4b:2d:fd:e2:94:79:d5:48:5c:b1:f3:7f:
x    ac:53:ea:0a:fd:cd:0d:f0:25:8b:31:a1:ee:14:3b:
x    19:56:2c:8a:f8:5b:7c:6b:13:55:17:e7:74:bc:3e:
x    2b:10:78:f0:e6:d4:2f:37:da:3b:92:20:89:17:32:
x    0f:25:8b:54:a5:bf:df:ad:68:9b:7c:e2:8c:55:7c:
x    52:72:2f:80:ca:89:d2:25:5c:3d:1e:ca:c2:e9:dc:
     6f:61:c1:dd:e3:3e:54:07:21:e3:d8:58:3d:9e:ac:
     ca:f8:e0:0b:bc:34:eb:3c:3b:4b:63:c1:93:ec:9d:
     c3:bf
 publicExponent:
     0c:75:94:6b:4e:48:ee:41:fa:b1:9e:20:6d:71:1b:
     48:ec:23:0d:b4:07:26:52:d1:e4:c2:16:5b:1d:54:
     8e:b0:0c:cd:c6:89:64:ba:7f:b9:91:6f:00:f7:c2:
     e5:67:e0:e1:f9:f5:4f:84:be:9e:55:6f:4f:c6:31:
     be:fd:9a:72:fb:00:40:e5:46:de:0a:18:43:7c:48:
     72:fc:1a:7a:b3:41:af:7a:1d:71:f6:7e:88:4d:d9:
     04:cf:87:4f:8d:79:2a:04:e3:98:e3:77:f6:4a:aa:
     44:3f:6c:05:8d:0b:44:7b:82:a2:93:f3:3b:a3:29:
     a7:19:68:6c:16:d0:3c:df:82:7f:6f:ed:bb:cf:ad:
     a8:67:02:40:dc:64:d9:4f:8e:58:33:69:26:7a:0a:
     61:b2:09:1b:e4:63:8f:40:e6:57:2a:a1:8a:07:f4:
     d3:b9:aa:b1:4e:d0:f9:d1:6a:fa:93:55:59:2e:b3:
     38:b0:86:34:d8:08:b0:5c:79:86:33:91:11:b1:2b:
     c0:58:a9:95:e5:13:a5:fb:ba:27:c0:75:90:e3:16:
     be:e9:c9:6e:6b:6e:07:ae:96:fc:f2:78:a9:78:35:
     d4:ee:63:29:c6:f1:e8:22:2a:a1:da:ac:ad:0a:03:
     97:b3:82:f6:3c:f5:01:97:44:a6:ae:1b:79:8d:cf:
     ff
 privateExponent:
     62:6d:74:64:62:6d:74:64:62:6d:74:64:62:6d:74:
     64:62:6d:74:64:62:6d:74:64:62:6d:74:64:62:6d:
     74:64:62:6d:74:64:62:6d:74:64:62:6d:74:64:62:
     6d:74:64:62:6d:74:64:ff
 prime1:
     00:ad:cd:1c:22:93:db:4a:91:cc:b9:77:6b:84:9a:
     32:62:fe:c9:33:32:7e:b5:61:d6:e8:58:5b:05:1c:
     f0:33:19:bc:a0:0e:27:2a:45:ee:ea:3b:c9:0c:1a:
     8c:a1:4c:63:68:fb:3a:ed:10:eb:a3:f3:5a:79:39:
     b4:40:00:e4:b9:60:2c:b6:ef:8d:e9:c1:ad:07:ce:
     d8:a4:8f:85:22:76:a9:90:11:64:45:ea:2c:1e:0f:
     c3:02:99:4f:fe:ef:29:54:33:a2:ee:09:6f:45:5b:
     d0:e6:0f:6e:00:c5:ce:20:1c:fa:b6:ef:01:e8:b6:
     aa:ab:57:bb:28:02:c2:f5:41
 prime2:
     00:de:3a:e2:82:5c:72:cb:76:63:66:64:03:1f:21:
     b4:70:a3:62:89:79:10:37:4a:8c:1e:21:32:59:c0:
     58:9c:e2:d8:5b:95:c7:50:43:93:21:7a:f9:77:3f:
     14:66:45:57:db:14:41:e3:5d:39:b1:e5:14:9b:e5:
     e7:bd:1b:8a:1b:8c:ae:cd:f0:b4:71:3f:43:58:a4:
     2a:0f:4b:53:c8:b9:56:65:db:62:32:7c:9e:b0:d6:
```

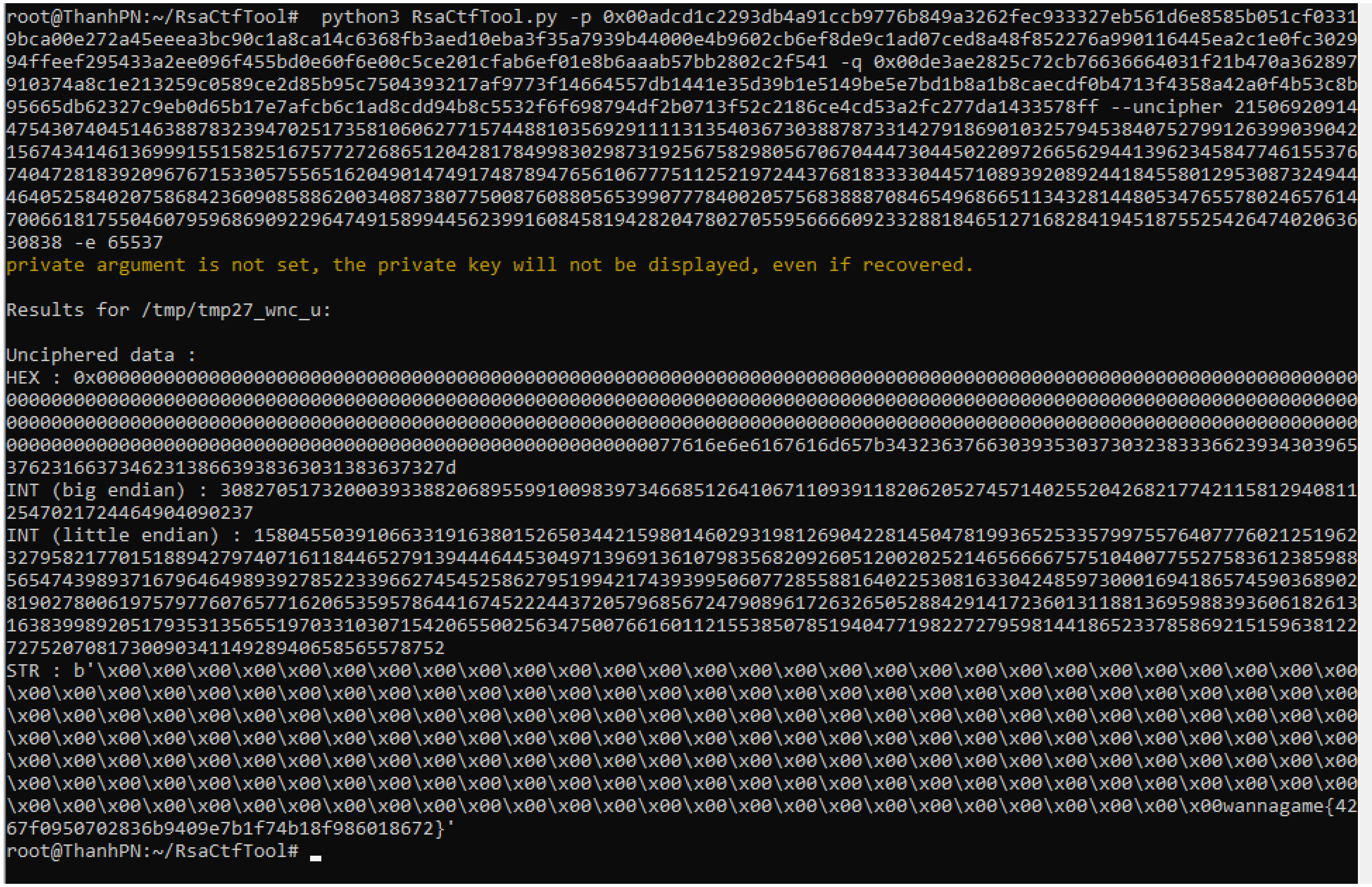
Với p và q là prime1 và prime2, ta đã có đủ dữ kiện để giải bài này

Cú pháp:

python3 RsaCtfTool.py -p <p> -q <q> -e <e> --uncipher <c>

Với p,q là prime1 và prime2 (sau khi bỏ dấu :)

Vì n = p*q, khi ta đã truyền p và q thì không cần n nữa



```
root@ThanhPN:~/RsaCtfTool#  python3 RsaCtfTool.py -p 0x00adcd1c2293db4a91ccb9776b849a3262fec933327eb561d6e8585b051cf0331
9bca00e272a45eeea3bc90c1a8ca14c6368fb3aed10eba3f35a7939b44000e4b9602cb6ef8de9c1ad07ced8a48f852276a990116445ea2c1e0fc3029
94ffeef295433a2ee096f455bd0e60f6e00c5ce201cfab6ef01e8b6aaab57bb2802c2f541 -q 0x00de3ae2825c72cb76636664031f21b470a362897
910374a8c1e213259c0589ce2d85b95c7504393217af9773f14664557db1441e35d39b1e5149be5e7bd1b8a1b8caecdf0b4713f4358a42a0f4b53c8b
95665db62327c9eb0d65b17e7afcb6c1ad8cdd94b8c5532f6f698794df2b0713f52c2186ce4cd53a2fc277da1433578ff --uncipher 21506920914
475430740451463887832394702517358106062771574488103569291111313540367303887873314279186901032579453840752799126399039042
156743414613699915515825167577272686512042817849983029873192567582980567067044473044502209726656294413962345847746155376
740472818392096767153305755651620490147491748789476561067775112521972443768183333044571089392089244184558012953087324944
464052584020758684236090858862003408738077500876088056539907778400205756838887084654968665113432814480534765578024657614
700661817550460795968690922964749158994456239916084581942820478027055956666092332881846512716828419451875525426474020636
30838 -e 65537
private argument is not set, the private key will not be displayed, even if recovered.

Results for /tmp/tmp27_wnc_u:

Unciphered data :
HEX : 0x0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000077616e6e6167616d657b343236376630393530373032383336623934303965
376231663734623138663938363031383637327d
INT (big endian) : 30827051732000393388206895599100983973466851264106711093911820620527457140255204268217742115812940811
2547021724464904090237
INT (little endian) : 1580455039106633191638015265034421598014602931981269042281450478199365253357997557640776021251962
327958217701518894279740716118446527913944464453049713969136107983568209260512002025214656666757510400775527583612385988
565474398937167964649893927852233966274545258627951994217439399506077285588164022530816330424859730001694186574590368902
819027800619757977607657716206535957864416745222443720579685672479089617263265052884291417236013118813695988393606182613
163839989205179353135655197033103071542065500256347500766160112155385078519404771982272795981441865233785869215159638122
727520708173009034114928940658565578752
STR : b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00wannagame{42
67f0950702836b9409e7b1f74b18f986018672}'
root@ThanhPN:~/RsaCtfTool# _
```

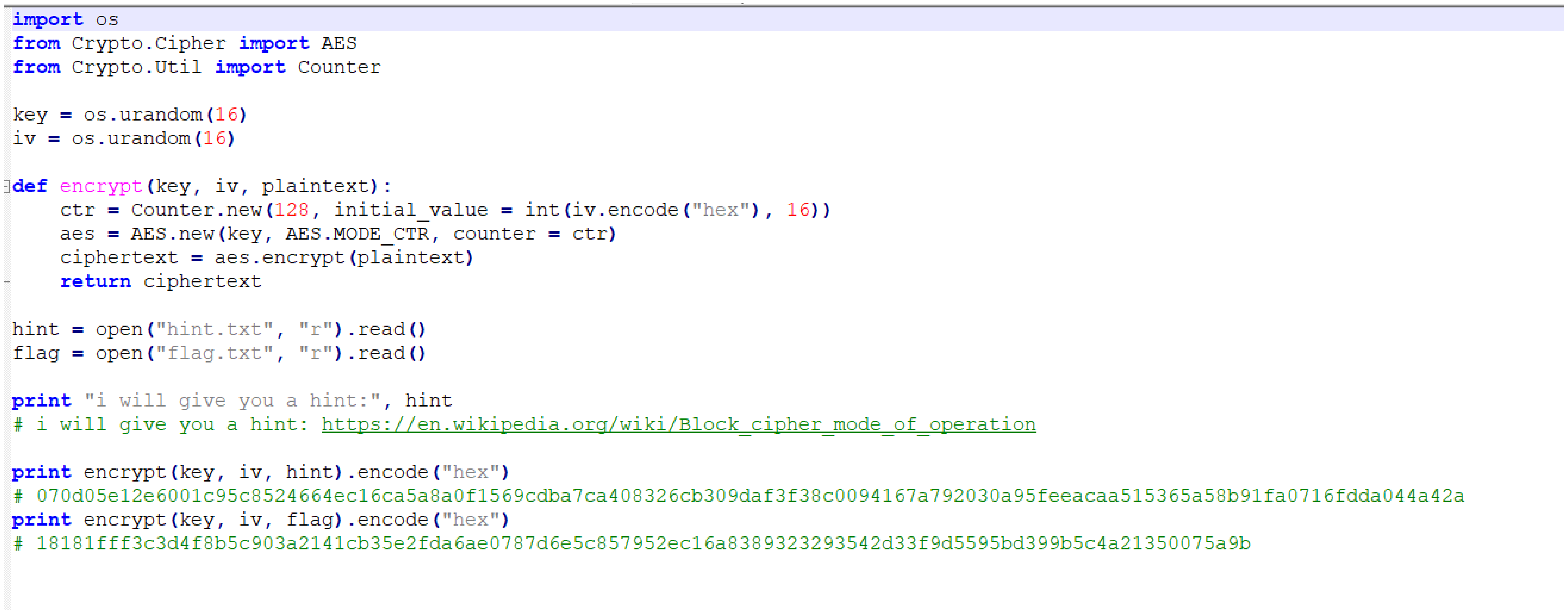
Vậy là ra.

P/s: Một thành viên của tụi mình, revirven đã thử tấn công RSA2 bằng RSACTFTOOL với sagemath nhưng sau 3 tiếng thì cả 2 tools đều bó tay. Không biết có cách nào để break nếu k có chall RSA1 không nữa

AES

Solver: cod201

Ở challenge này, thuật toán mã hóa là AES-CTR , là thuật toán mã hóa stream.



```
import os
from Crypto.Cipher import AES
from Crypto.Util import Counter

key = os.urandom(16)
iv = os.urandom(16)

def encrypt(key, iv, plaintext):
    ctr = Counter.new(128, initial_value = int(iv.encode("hex"), 16))
    aes = AES.new(key, AES.MODE_CTR, counter = ctr)
    ciphertext = aes.encrypt(plaintext)
    return ciphertext

hint = open("hint.txt", "r").read()
flag = open("flag.txt", "r").read()

print "i will give you a hint:", hint
# i will give you a hint: https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation

print encrypt(key, iv, hint).encode("hex")
# 070d05e12e6001c95c8524664ec16ca5a8a0f1569cdba7ca408326cb309daf3f38c0094167a792030a95feeacaa515365a58b91fa0716fdda044a42a
print encrypt(key, iv, flag).encode("hex")
# 18181fff3c3d4f8b5c903a2141cb35e2fda6ae0787d6e5c857952ec16a8389323293542d33f9d5595bd399b5c4a21350075a9b
```

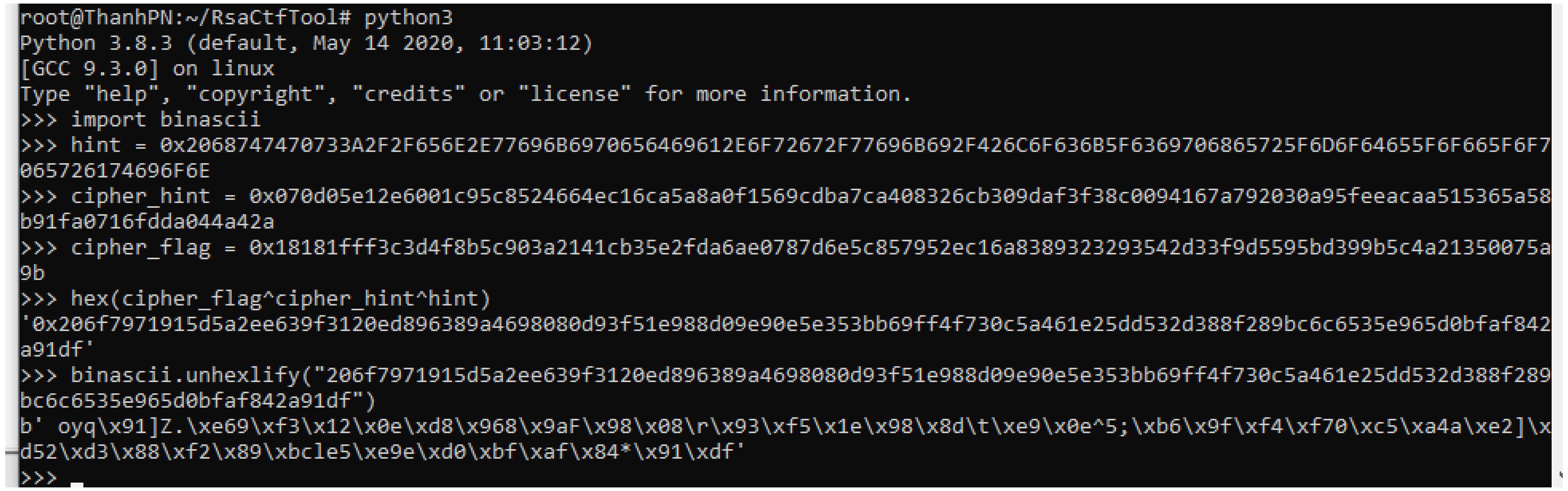
Ở challenge này BTC đã gợi ý vào wiki đọc thêm, và có 1 đoạn làm mình chú ý:

> If the IV/nonce is random, then they can be combined together with the counter using any invertible operation (concatenation, addition, or XOR) to produce the actual unique counter block for encryption. In case of a non-random nonce (such as a packet counter), the nonce and counter should be concatenated (e.g., storing the nonce in the upper 64 bits and the counter in the lower 64 bits of a 128-bit counter block). Simply adding or XORing the nonce and counter into a single value would break the security under a chosen-plaintext attack in many cases, since the attacker may be able to manipulate the entire IV–counter pair to cause a collision. Once an attacker controls the IV–counter pair and plaintext, XOR of the ciphertext with the known plaintext would yield a value that, when XORed with the ciphertext of the other block sharing the same IV–counter pair, would decrypt that block.[24]

Về cơ bản, vì key và VI không bị thay đổi khi tạo 2 ciphertext khác nhau, chỉ cần XOR ba cái cipher_hint, cipher_flag và hint là có thể lấy được flag

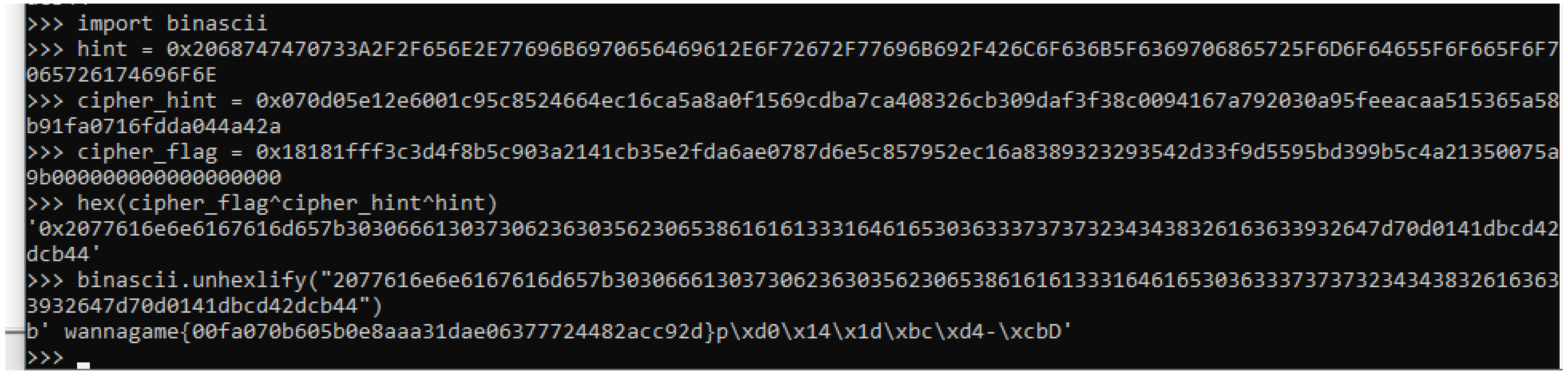
Nhưng đời đâu như mơ…

Chú ý: trong hint, có dấu cách trước https://



```
root@ThanhPN:~/RsaCtfTool# python3
Python 3.8.3 (default, May 14 2020, 11:03:12)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import binascii
>>> hint = 0x2068747470733A2F2F656E2E77696B6970656469612E6F72672F77696B692F426C6F636B5F6369706865725F6D6F64655F6F665F6F7
065726174696F6E
>>> cipher_hint = 0x070d05e12e6001c95c8524664ec16ca5a8a0f1569cdba7ca408326cb309daf3f38c0094167a792030a95feeacaa515365a58
b91fa0716fdda044a42a
>>> cipher_flag = 0x18181fff3c3d4f8b5c903a2141cb35e2fda6ae0787d6e5c857952ec16a8389323293542d33f9d5595bd399b5c4a21350075a
9b
>>> hex(cipher_flag^cipher_hint^hint)
'0x206f7971915d5a2ee639f3120ed896389a4698080d93f51e988d09e90e5e353bb69ff4f730c5a461e25dd532d388f289bc6c6535e965d0bfaf842
a91df'
>>> binascii.unhexlify("206f7971915d5a2ee639f3120ed896389a4698080d93f51e988d09e90e5e353bb69ff4f730c5a461e25dd532d388f289
bc6c6535e965d0bfaf842a91df")
b' oyq\x91]Z.\xe69\xf3\x12\x0e\xd8\x968\x9aF\x98\x08\r\x93\xf5\x1e\x98\x8d\t\xe9\x0e^5;\xb6\x9f\xf4\xf70\xc5\xa4a\xe2]\x
d52\xd3\x88\xf2\x89\xbcle5\xe9e\xd0\xbf\xaf\x84*\x91\xdf'
>>> 
```

Trong source code, khi mã hóa, mỗi vị trí trùng nhau chung một counter, còn khác nhau là khác counter, nên chúng ta phải cho độ dài 2 đoạn ciphertext bằng nhau bằng cách.. Thêm các số 0 vào sau cipher_flag để bằng độ dài cipher_int thì sẽ giải mã được



```
>>> import binascii
>>> hint = 0x2068747470733A2F2F656E2E77696B6970656469612E6F72672F77696B692F426C6F636B5F6369706865725F6D6F64655F6F665F6F7
065726174696F6E
>>> cipher_hint = 0x070d05e12e6001c95c8524664ec16ca5a8a0f1569cdba7ca408326cb309daf3f38c0094167a792030a95feeacaa515365a58
b91fa0716fdda044a42a
>>> cipher_flag = 0x18181fff3c3d4f8b5c903a2141cb35e2fda6ae0787d6e5c857952ec16a8389323293542d33f9d5595bd399b5c4a21350075a
9b000000000000000000
>>> hex(cipher_flag^cipher_hint^hint)
'0x2077616e6e6167616d657b30306661303730623630356230653861616133316461653036333737373234343832616363932647d70d0141dbcd42
dcb44'
>>> binascii.unhexlify("2077616e6e6167616d657b303066613037306236303562306538616161333164616530363337373732343438326163633
3932647d70d0141dbcd42dcb44")
b' wannagame{00fa070b605b0e8aaa31dae06377724482acc92d}p\xd0\x14\x1d\xbc\xd4-\xcbD'
>>> 
```

Chemistry Class

Unsolved - revirven

Chall này tụi mình có giải ra các PTHH, tuy nhiên thì tụi mình cũng bótay

Đáp án các PTHH là;

311

281

212

```
211
212
122
211
311
231
164
211
213
214
122
121
131
132
```