Network Security

Project 1
Hacking the Cipher
0556562 陳鴻君

Workflow

1. use openssl to translate public*.pub into Modulus 2048 bits format in Hex.
2. transform Hex into decimal.
3. find GCD as P, then we know each Q.
4. use rsatool to find private keys.

Step 1: openssl

toRSAkey.sh ?

```
1 #!/bin/bash
2 for i in $(seq 1 12)
3 do
4     openssl rsa -in "public$i.pub" -pubin -text > "key$i"
5 done
```

the translate results seem like:

key1 ?

```
1  Public-Key: (2048 bit)
2  Modulus:
3      00:c7:bf:53:3c:41:41:b0:16:22:bb:9d:cc:08:c4:
4      d9:7a:00:95:8a:23:41:8d:b0:36:1c:03:d2:4c:9e:
5      b8:a3:1b:98:4a:f1:65:54:07:95:14:e1:1d:5c:c8:
6      7a:0c:cd:dd:60:e8:43:1a:67:e4:66:17:81:5a:3d:
7      2e:81:b5:6b:56:ca:d2:c7:51:81:b9:91:d0:2a:cc:
8      d7:5e:57:0e:0d:c1:8f:01:b6:20:31:36:1a:51:bc:
9      21:4a:9f:91:ab:b2:cf:29:b3:3b:d2:24:0a:72:cd:
10     db:b8:03:aa:c8:8b:4d:7a:cc:3a:6e:f0:54:16:60:
11     35:ec:b2:7c:6a:0a:b5:47:c1:0d:38:91:0d:f3:06:
12     20:60:8a:31:b0:bf:a7:d6:03:b9:3c:f7:d2:ea:ba:
13     89:3e:d1:1c:07:49:81:b3:60:0c:51:2a:57:33:54:
14     35:7f:97:08:3c:ca:3d:63:56:01:81:37:6e:c9:3d:
15     89:6f:01:f7:40:22:62:4d:de:f9:d3:1e:f2:1e:f7:
16     84:3e:9e:80:0f:c2:15:2e:48:30:b5:08:30:68:41:
17     4a:af:85:18:ce:44:61:7a:e9:9b:0a:d8:cf:52:1e:
18     c3:65:47:34:56:bb:cc:ec:63:82:e4:51:ac:33:dd:
19     07:cb:ef:b1:e0:ee:8c:d4:80:97:e3:bd:69:09:10:
20     8d:6d
21 Exponent: 65537 (0x10001)
22 -----BEGIN PUBLIC KEY-----
23 MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAx79TPEFBsBYiu53MCMTZ
24 egCViiNBjbA2HAPSTJ64oxuYSvFlVAeVFOEdXMh6DM3dYOhDGmfkZheBWj0ugbVr
25 VsrSx1GBuZHQKszXXlcODcGPAbYgMTYaUbwhSp+Rq7LPKbM70iQKcs3buAOqyItN
26 esw6bvBUFmA17LJ8agq1R8ENOJEN8wYgYIoxsL+n1gO5PPfS6rqJPtEcB0mBs2AM
27 USpXM1Q1f5cIPMo9Y1YBgTduyT2JbwH3QCJiTd750x7yHveEPp6AD8IVLkgwtQgw
28 aEFKr4UYzkRheumbCtjPUh7DZUc0VrvM7GOC5FGsM90Hy++x4O6M1ICX471pCRCN
29 bQIDAQAB
30 -----END PUBLIC KEY----
```

Step 2: Hex to Decimal, and find GCD

findGCD.py

```python
from fractions import gcd

arr = [""]*12
for i in range(1, 13):
    filename = "key" + str(i)
    with open(filename, 'r') as f:
        for line in f:
            if "Public" in line or "Modulus" in line:
                continue
            elif "Exponent" in line:
                break
            else:
                line = line.replace('\n', '')
                line = line.replace(' ', '')
                line = line.replace(':', '')
                arr[i-1] += line

arr = map(lambda x: int(x, 16), arr)

with open('answer', 'w') as f:
    for i in range(len(arr)):
        for j in range(i):
            p = gcd(arr[i], arr[j])
            if p > 1:
                f.write("i="+str(i+1)+", j="+str(j+1)+"\np="+str(p)+"\nq1="+str(arr[i]/p)+"\nq2="+str(arr[j]/p)+"\n")
```

the output is:

answer

```
i=8, j=3
p=14656665144589336868890576345676445233783803276368267622102594568299164979334002689085447204937159234673045419122185037140840658147541857900888111571092173530748331667107582622861309727150160914480781841205155449584530166428770678446245420268299373990760393892275516496045323891286171163252445865368303271017
q1=15202267627983314042987791330678754823937798326693245382006781781581161294528297560813467269987953974649058941713121566472331884403957434009348342937482062143381687089719111519197113151388943626905241449645396658878293194764557298839319732260121099717666949664805945203137174210928305077042736209158172911 4989
q2=13591928925414424039837370917116630644641941004911771936349542880507716046500026193128068722178524319179545141848821767187968190076092373745098351287919441961214713691246398623260252096457890765634884088088126673634002841565038715431549291902966317164815250033102095030674377557386106688405799831761608826682
```

p is gcd, and q is the other prime for each.

Step 3: rsatool
https://github.com/ius/rsatool

the answer is:
private3.pem

private3.pem

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAnc57/Ghy7lohtMcrJgdFhJrvtHp/eoXczQ3fhEhHcxSAt+RkuPyrYlFXKZhU
r3lctZEOEx8tDYIkSuUp3U8lt+qJ13FanTVvRRU/rjwip3YYTaZmvHC+zM/y25Ud/2BbhY3NX/vS
uzI+M+I64+B47n6yfK6rCZUXb5hd/6hKX23Yr5Jjx0pY26Rd7Ft67AHfkXt/VTM0tVbpdwAAl4Pp
0+in/sdPE80rDbMwbMd/V/i1Nr7z22ii0GIX5vDOSD5ianrwXX3vwdR2GNza3amOzkOk14XXiQem
7z1X9AkUa2YcoazilGDGqu78C659qOJ1n0en6ofJXiBZuz/NiBHjlQIDAQABAoIBAFSqpkcEWzWa
0h7GBVZ/7EJ7RkSUvLSmAZNa8CNvDDaoN29ycL/3gJRdg+BjRHxpJt6bluuXfHqU6pfsYAvHHTuR
8KbQwIc8VhjAj8wRo54pdFyk/NX5v8TvhSNe3BtaPYBoVlo7KBMbn9gCdwowM3HyNB1FMasQfOR4
4WX7ivurghPCEipOZhEllH62cIFk/uQszFc1Pt5w5H3dXTZs/0qUnuYGNBhITRysiNOT1Anlo5Ow
rU5DWSjZ3uYSZbJlQnUuBuNSauG6POPbFrf3qeai1+TeAUsPGgPK4OtqgdtojEp9y5lLZUFUr049
OhTxLiZofxP0uOn7Judcb3gwWsECgYEA0LfEWRh1OkwKgZqi53eXEkwkrP+aDwk/SoZkHPtzzGO/
VkmI7ZqGsOb3FmFsEEKnJ360+BugeVJmCa50y7sckAuNKeKgtWjliBUWcE1u9a2Y3SOobkfDtyfM
W9B44CLvL2VswkC46OsfZgL1LHTYgFUKOsby9ET6pXWWhrCtfGkCgYEAwY40xXiPE9FvRZRu+ZaJ
o+xUNh8+j8Cu+juhEKYbz0bd1nyT/+VQ+uxY14N3r3XRXGA/cdzP5fS5NP9Z+3COXn1jI0wvpIbD
hLUo0QXOvpdhAmLgWfGCzdDe2AKwNwkmTb0/T2jg8SdtuvgTm9P79Pj/mnM3x7nYCwJg/S+4uE0C
gYA8Yr6TDGl2OWPcxXGWSFpT9+Wwi6715LYECMfUPsfs/XrxKwtLuyZ27gvU1NDgD3ivC5CuK8gM
8dMFwAyYmQUVg9IOqTvX4IZwDbHVQgH1YIlWiaeEaIzuY7xTYAhoyJKtAsoplXIqKUSHThnDB/dP
gcsY4FCQbZia4F9xjpJS6QKBgHcc13bqsUol6c8tAHpSXNPnHQ/NfIeksUnHJmCDiLuvRde4BLBG
4+X3Stkdy6rL25J4nLKqi7pdNxjZR/vHkaWujjLrsKoUUQA7KCox48BAkm+Qf7Z6POAB+RF/3geQ
duUCYJKy3TMq64lTH2YXN4fTMCFRZHNh+AUWaq+FDa1FAoGBAIFgSqppqwlmYDXta5NQwJrFSE32
hw4pqmU5vpCP1Iy/kMzQJwPNcs2VQPZJswkcuCl+Jt/TzuaxwNlsfSjBsPc/MKGBugzkkDUpIOnm
hh9y60j4GnMecUMIdTKS73mgynCt8jA3SgShaKVeAkVAFsJBv+ImHGE71r+Hz4vxWUUz
-----END RSA PRIVATE KEY-----
```

private8.pem