

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 egCViiNBjbA2HAPSTJ64oxuYSvFLVAeVF0EdXMh6DM3dY0hDGmfkZheBWj0ugbVr  
25 VsrSx1GBuZHQKszXXlc0DcGPAbYgMTYaUbwhSp+Rq7LPKbM70iQKcs3buAOqyItN  
26 esw6bvBUFA17LJ8agq1R8EN0JEN8wYgYIoxsL+n1g05PPfS6rqJPtEcB0mBs2AM  
27 USpXM1Q1f5cIPMo9Y1YBgTduyT2JbwH3QCJiTd750x7yHveEPp6AD8IVLkgwtQgw  
28 aEFKr4UYzkrRheumbCtjPUh7DZUC0VrvM7GOC5FGsM90Hy++x406M1ICX471pCRCN  
29 bQIDAQAB  
30 -----END PUBLIC KEY-----
```

Step 2: Hex to Decimal, and find GCD

```
findGCD.py ?
1 from fractions import gcd
2
3 arr = [""]*12
4 for i in range(1, 13):
5     filename = "key" + str(i)
6     with open(filename, 'r') as f:
7         for line in f:
8             if "Public" in line or "Modulus" in line:
9                 continue
10            elif "Exponent" in line:
11                break
12            else:
13                line = line.replace('\n', '')
14                line = line.replace(' ', '')
15                line = line.replace(':', '')
16                arr[i-1] += line
17
18 arr = map(lambda x: int(x, 16), arr)
19
20 with open('answer', 'w') as f:
21     for i in range(len(arr)):
22         for j in range(i):
23             p = gcd(arr[i], arr[j])
24             if p > 1:
25                 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 ? buffers
1 i=8, j=3
2 p=14656665144589336868890576345676445233783803276368267622102594568299164979334002689085447204937159234673045419122185037140840658147541857900888111157109
2173530748331667107582622861309727150160914480781841205155449584530166428770678446245420268299373990760393892275516496045323891286171163252445865368303271
017
3 q1=152022672798331404298779133067875482393779832669324538200678178158116129452829756081346726998795397464905894171312156647233188440395743400934834293748
2062143381687089719111519197113151388943626905241449645396658878293194764557298839319732260121099717666949664805945203137174210928305077042736209158172911
4989
4 q2=1359192892541442403983737091711663964464194100491177193634954288050771604650002619312806872217852431917954514184882176718796819007609237374509835128791
9441961214713691246398623260252096457890765634884088088126673634002841565038715431549291902966317164815250033102095030674377557386106688405799831761608826
682
```

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 ?
1 -----BEGIN RSA PRIVATE KEY-----
2 MIIIEowIBAAKCAQEAnC57/Ghy7lohtMcrJgdFhJrvtHp/eoXczQ3fhEhHcxSat+RkuPyrYLFXKZhU
3 r3lctZE0Ex8tDYIkSuUp3U8lt+qJ13FanTVvRRU/rjwip3YYTaZmvHC+zM/y25Ud/2BbhY3NX/vS
4 uzI+M+I64+B47n6yfk6rCZUXb5hd/6hKX23Yr5Jjx0pY26Rd7Ft67AHfkXt/VTM0tVbpdwAAL4Pp
5 0+in/sdPE80rDbMwbMd/V/i1Nr7z22ii0GIX5vD0SD5ianrwXX3vwdR2GNza3amOzk0k14XXiQem
6 7z1X9AkUa2YcoazilGDGqu78C659q0J1n0en6ofJXiBZuz/NiBHjlQIDAQABAoIBAFSqpkcEWzWa
7 0h7GBVZ/7EJ7RkSUVLSmAZNa8CNvDDaoN29yCL/3gJRdg+BjRHxpJt6bluuXfHqU6pfsYAvHHTuR
8 8KbQwIc8VhJAJ8wRo54pdFyk/NX5v8TvShSne3BtaPYBoVlo7KBMbn9gCdwomM3HyNB1FMasQf0R4
9 4Wx7ivurghPCEip0ZhEllH62cIFk/uQszFc1Pt5wH3dXTZs/0qUnuYGNBhITRysiN0T1Anlo50w
10 rU5DWSjZ3uYSZbJlQnUuBuNSauG6P0PbFr3qea1l+TeAUSPGgPK40tqgdtoJEp9y5LLZUFUro49
11 0hTxLiZofxP0uOn7Judcb3gwWsECgYEA0LfEWRh10kwKgZqi53eXEkwkRP+aDwk/SoZkHPTzzG0/
12 VkmI7ZqGs0b3FmFsEEKnJ360+BugeVJmCa50y7sckAuNKeKgtWjliBUWcE1u9a2Y3S0obkfDtyfM
13 W9B44CLvL2VswkC460sfZgLLHTYgFUK0sby9ET6pXWWhrCtfgkCgYEAwY40xXiPE9FvRZRu+ZaJ
14 o+xUNh8+j8Cu+juhEKYbz0bd1nyT/+vQ+uxY14N3r3XRXGA/cdzP5fS5NP9Z+3C0Xn1ji0wvpIbD
15 hLUo0XQ0vpdhamLgWfGCzdDe2AKwNwkmTb0/T2jg8Sdtuvgtm9P79Pj/mnM3x7nYcWJg/S+4uE0C
16 gYABYr6TDGL20WPcxXGWSFpT9+Wwi6715LYECMFUPsfs/XrxKwtLuyZ27gvU1NDgD3ivC5CuK8gM
17 8dMFwAyYmQUVg9I0qTvX4IZwDbHVQgH1YI1WiaEaIzuY7xTYAhoyJKtAsoplXIqKUSHTnDB/dP
18 gcsY4FCQbZia4F9xjpJS6QKBgHcc13bqsUol6c8tAHpSXNPNHQ/NfIeksUnHJmCDiLuvRde4BLBG
19 4+X3Stkdy6rL25J4nLKqi7pdNxjZR/vHkaWujjLrsKoUUQA7KCox48BAkm+Qf7Z6P0AB+RF/3geQ
20 duUCYJKy3TMq64LTH2YXN4fTMCFRZHNh+AUWaq+FDa1FAoGBAIFgSqppqWlmYDXta5NQwJrFSE32
21 hw4pqmU5vpCP1Iy/kMzQJwPNcs2VQPZJswkcUcl+Jt/TzuaxwNlsfSjBspc/MKGBugzkkDUPiOnm
22 hh9y60j4GnMecUMIDTKS73mgynCt8jA3SgShakVeAKVAFsJBv+ImHGE71r+Hz4vxWUUz
23 -----END RSA PRIVATE KEY-----
```

private8.pem

private8.pem ?

```
1 -----BEGIN RSA PRIVATE KEY-----
2 MIIIEogIBAAKCAQEAsIDMbcVpE90iK00mtkww8XXNbUwczKPXJi0+3NXVQHyrITITnVd/EIoPxqY
3 CloxebCowFx5xjnu+Vo+gM5CKYzb0znGMkYlCj5AmXV1+a6hMERL7+Qsth/12ghUhgMMMjshA+ZR
4 Pu6tJ1d7H7Hoge3z0PvkHnfNIxIcHQI43GYJURWNQ2Mij+h3CF25P1ictf+/Uijdfa1wk+3PLtTy
5 jRMcP7IOwdWUloPowu/vCUgKe/Qq0+WINNi/Uf1MaXBTq+4fUTPzXrVBGxQ6DYMjysHw9wxgZCYi
6 jLeIrJdIL7DPkaYVadv+pzLhKWAsigEdzbVI0mjjDnlc/QoL9g37tQIDAQABaoIBAF0fkjtnzLL0
7 8B1jXfb8viuITp6Ay7UkfTu3tfd0PCowU+IWrvhNLww3I+5R4Tr2ZH5tLmaE6cT66bgGA5rvHfWl
8 oLC8vKRRc3eA6wZJ2m1DSWch60FtT6myFr8IN1002hZqYUgBtPiGCE+daN4pYik83YFgBkJRLEWz
9 bNJpnRr+C3M4FLYoh12NHMKR5rD95jED3Y0vKSoj5VJY5E291naiEUe7BNnJ00AxRnfo2e5PZwJd
10 LJvF0XhNvHcH08rkgU95MytkcYe43Yn2cA1KiZeAxa3hqDF12ghI8+67VJGLnQhkWlmmcDV69l2z
11 7swCAuX3KdcYe9Y2GwYKXoHSXsECgYEA0LfEWRh10kwKgZqi53eXEkwkrP+aDwk/SoZkHPtzzG0/
12 VkmI7ZqGs0b3FmFsEEKnJ360+BugeVJmCa50y7sckAuNKeKgtWjliBUWcE1u9a2Y3S0obkfDtyfM
13 W9B44CLvL2VswkC460sfZgLLHTYgFUK0sby9ET6pXWWhrCtfGkCgYEA2HzLS8Me3l7n6mVIDeT0
14 QsaCjSgf7IEmxRepIuKL8A4uW/fwZi2VrGmqjAbcfH0bbQ7bLlvY6Njbk51V7Cigq2+/NSSB4y8X
15 wU+RXomgItF4csRfuUuTvm8zMyDi3lapMy1IyTFXo2Fj971PrEzBM6iA+aUj9KKYvmM2VwQri20C
16 gYA8Yr6TDGL20WPcxXGWSFpT9+Wwi6715LYECmfUPsfs/XrxKwtLuyZ27gvU1NDgD3ivC5CuK8gM
17 8dMFwAyYmQUVg9IOqTvX4IZwDbHVQgH1YilWiaeEaIzuY7xTYAhoyJKtAsoplXIqKUSHTnDB/dP
18 gcsY4FCQbZia4F9xjpJS6QKBgEauL92zfd+SUTunZ6gJnQIWN8eIYyvfWt4chH88Qf4FU0N5psZl
19 eamJ8kG3d8M5QXwQzqTLHf2f18jRJ1vAk4WtPHP0oxu8q2Noe0gwWS8yTuH0820jFuvjyLRS0DY
20 oHNyBIKTlo5kHLRk3Z9ei/GmwT1AmoWjliUMdVBhryndAoGAb6n53izpD61EyKl07Db8f6UEJPDB
21 NN3gTjiEYaAmg4Wq1Ug4R7w21mUIrxSj3LF09vqC9Mu3Lv7Kp/umm+1dSv1N+EaGq06WfXV/ePKa
22 tH1z4mfgqjJYItN0TKU4YKa/7tLxHrHnogmZ2DCIEQt3b5dnqnyPy686275oJ8y4ik=
23 -----END RSA PRIVATE KEY-----
```