

Viktor Chekhovoi

I named my file “homework.” Contents of “homework”:

```
1 -----BEGIN RSA PRIVATE KEY-----
2 MIIG4gIBAAKCAyEApDH5Wn5i7q3RjLkrgIiLMK9RU9SaUAPw32vrTzqfx0LLs5hf
3 jmrZgKCWACx9h0Y1sCUHmxZwi0Fl103VY4MEJZbafn9Bb6N5z2bouNLN3uaFXskx
4 kRtWU77/8U0Ov2pQIQcoXwSGqXX39R1AZWe0xkNEMmfzbBVKwxwks0/MP7TPXW
5 hqYf1uhqafHyIz0u0+NPJU5FwEVivGQJnTC00v1EGNmPY8wwf/OHMc6qc/Z9d/Kx
6 Rx7cvQxpRcmRS4wVt69McLSbVh/+ /sga80CxCkQY8e0VmlW+gWonQCxHvgX/4VuVVN
7 omWZt1tYqfQ8LMraDkqd0GFpDWFm1F02jYzQdhlLPQpXG6750s0AYHLxWt5+Ux3
8 TZuP+goZm5L9Jg5hW7EPqn7Hlv4iynjgTaG8nS/M90c8Mjdk7TbmX3X1ixPKvLd
9 38PImK4CYT00Isko5Zmd2Vmas8jK5LH2r1r6PBEUWLMEDIkd/TIqQ+gA1Zkww0zt
10 o0wN1qoWmgcTLHbtAgMBAACGgGAGaZs68QetGaW680yYwvXopNgwxNXX6gHGcB
11 hYkE+N3ocI/3LYpnm/56MGDC42UGchP20jVSM2ft90sF0Nvgw/aC/4vLrwj8uHe9
12 F1JnMtEUOLB5e8+rhomp9goxqZL0F1XBUwaFtn/970uDU9I307RwJBU7T6iiRW1W
13 EJqFptr+4RDuiG5fPvVj2LUfnHwmbY1vCcbmgpjUSdXe4p1/Kk/yWwDzTQXS9coX
14 bqJ4YLE5XImFRBsAJNBRIcNpXUHuyM/dDbAMZ5Q2ewEL/avq70z8ZDElyDXtxwZP
15 NWB5HEWqqUF5woujw0u810///c0uK5ZfhuA6Cfhy3xLjZK/vJ3ssQ6G7TZqxSSF
16 tMgC+NT+rBe9TmGF6C7jJndawsQXtyG2VCfvFU0a7tN5vHo0W/VDB0xgnqW/KKWF
17 pZM5STybfXvsPY8y7m45gQjHSFhA1tuyLKI8GP8k1r76qceH0becULMGKLD5BEAL
18 X7eELZpKBkl4JdjoL2IrAQUXcxXJAoHBANKSiezeBDWYef71ZPhNbIzHhtnR8oeR
19 v4EH5Hx77AcRgYnmghjM/4TTnBW54/wVIKQLSLoFgkLRkjdwC+rDfgKv9WfegfXx
20 vADDj7vct4SPHw3fXqRb49I+pBf9idJynBRPGdHyvJ/dwr98W+aibE/AynwODG+C
21 550KL/gqTnTLKI3vgLq7jdtVsVb0UXjzYrP971mz0udEiKAuATnriXRVBF140Fey
22 Zp1owVZw1rU/waCkUOm+2cjkWAwredkRIwKBwQDBo+yIz8llbWm2qwc5w8spmWdV
23 /24xw0MMNhrCkH+816bZbHsexevVHLDLQqbcj0z9vt0AL2+gGZD1bkf01igLtp
24 f0G5RTVc2Jrj05C0eiIwCHN50ZgCRH/VCX4TX+Waw2SSIHDa9CLFF7Vc8mPm+Ry4
25 +dq3uzHtFjHSC2sp8L+SDiw/gFc0Vt4vp9j1XIVac+XmtpcqFDDsYKKscPc5sEJY
26 b/Lva/ZOTeUnwS5/zmph8Wphzt2Hll+zIg0LQK8CgcAVfjFr45u1FtDVfsStlTTv
27 R35BqPkDlVnJ09c6wCZhmMkjt1AgVoPiSfWfB9S2sTzXIUrLKFB7I/L/QyTq70RY
28 LcX1r9Cq1jsGNFbr+guts2w5sck0KukG0y76NR9LV/W+Sqg8b/Vf0tziitSqgBn
29 ESa7W8S9cc+vC/J6Hu4/wDHWw+USGcn2kqiJfI1oLWire7QsRJLYVT/V5Fg4v0Qs
30 OyzAeIrgdAiXA11JSSjnJ1WDQ2Fbk4sKyhKxs0hQRTMcgcACQ2Y6WynNhehoNkuK
31 80zmmZJpo0iq0DtyZv8RC0Aptv20QQRXYbzV6trLMvGSy4urwNozNZp2t4LZPqKn
32 XQZxRkvqebkxrytmGmFck0wPSE1KoQfAmIidKaB2PCOTWqjmnEMfDCVfPEVOX03
33 OGVOhw7c4p650whMlxwN9nA0N0ol70B7hYZ/aByEtoFGqW8Xupq/etaZI4F1UNR
34 fKIHFhZQbk6KmdpxI9vv0QQ4Hj0EHYw+xGo2cX2A3H8GiKUCgcAhCo42WsmU9tDV
35 W1Vcw/hnQ1GYx4937qw+p54BDzz8wYsL3Zd9C/YTSZjWGCNTmqlOb+iqNXtCU08N
36 K04+TVwAlhvJXR6g++7IOA2pkYGBuWOWqPqB0d1vPA+qL9tuCs5DnxRTjcBz24PS
37 3272kDnqwxoz6FRyu+gEj32cbY3vkTrruXTyDEhOMG9Hn5LwFG+y/RXe9hUv8Yn3
38 NzyHcQtrK7Z7I+7ftTiu3uAW1BKDI1N1Ph0qovYV05sj7zPaF+Q=
39 -----END RSA PRIVATE KEY-----
```

Contents of “homework.pub”:

```
1 ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQCKMflafmLurdEmUquAiIswr1FT1JpQA/Dfa+tPop/
HqsuzmF+OatmAoJYALH2HRjWwJQebFnCLQWXTdVjgwQlltp+f0Fvo3nPZui40s3e5oVeyTGRG3BTvv/xTQ6/
aLaipyjHC5Iapdff1HUBLZ7TGQ0QyZ/NsFUrDHCSCw78w/tM9daGph/W6Gpp8fIjm67T408lTkXARWK8ZAMdMLQ6/UQY2Y9jzDB/
84cxzqpz9n138rFHHTy9DGLFyZFLjBW3r0xwtJtWH/7+yBrzQLEqpjx7RWZb6BaidALEe+Bf/
hW5VU2iZZm3W1ip9CryUyto0Sp3QYwKNZBzUU7aNhmp2Fss9Cmpcbvk6zQBgcVFa3n5THdNm4/6Chmbkv0mDmFbsQ+qfseW/
iLKeOBnobydL8z05zwyN2TtNuZfdWLE9eS8t3fw8iYrgJhNbQiySjlmZ3ZWZqzyMrUfavWvo8ERRYuYQMIR39MipD6ADVmtCT-
T02jTA3WqhYyBxMsdu0= kali@kali
```

Private Key:

I expect the contents of the private key file to contain information about the private key, encoded with DER and then converted to base64. The information should follow the standard ASN.1 syntax defined in RFC 8017:

```

RSAPrivateKey ::= SEQUENCE {
    version          Version,
    modulus           INTEGER,  -- n
    publicExponent    INTEGER,  -- e
    privateExponent   INTEGER,  -- d
    prime1            INTEGER,  -- p
    prime2            INTEGER,  -- q
    exponent1         INTEGER,  -- d mod (p-1)
    exponent2         INTEGER,  -- d mod (q-1)
    coefficient        INTEGER,  -- (inverse of q) mod p
    otherPrimeInfos    OtherPrimeInfos OPTIONAL
}

```

I used Lapo Luchini's ASN.1 decoder to decode the contents of the private key file. After reading the instructions, I copied the contents of the file and pasted them into the text box, then clicked "decode." I got the following output:

```

SEQUENCE (9 elem)
  INTEGER 0
  INTEGER (3072 bit) 372620891685503597322444680083102459791792054978738540127139768029752...
  INTEGER 65537
  INTEGER (3069 bit) 582097546283196641498266419685185922966929682297377843633620236673727...
  INTEGER (1536 bit) 204379822035784322427601689969659624317294686214920839044059986598593...
  INTEGER (1536 bit) 182317847218920853659694351363101294146234207175212773623294992593064...
  INTEGER (1533 bit) 202362126003662735284298543755912126309554178379941304081191579266409...
  INTEGER (1530 bit) 213094085737359619609267853493653342868937245672235507568921318226783...
  INTEGER (1534 bit) 311092551821162291661423747065501459379108454818747318706839103406389...

```

It has 9 integers in total:

1. The value of the version is 0, the offset is 4. The DER encoding is 0x020100." judging from the encoding of the other integers and the description of BER, 0x02 indicates that it's an integer, and 0x01 is the number of content octets
2. This is the modulus, also denoted as n. Its value is
37262089168550359732244468008310245979179205497873854012713976802975289974190
66407474197570761873655373903742326464038750995623872540533578250882955759203
83339057637707832436536963900273955333953322576713424485811067517576223130006
96372153375559482713062997249003356394276981688934206614912764986914791786372
15560983217883747746476291259144451371094164990786487346153786509811266598203
96637601432504350115381113087632518052755303265219283538399934204098919369443
84619824561856002820029295498248330541105537781027612851044216309233382245285
22623972521560200486496532284038262533195312214119407151227918350109856017388
12352849388066774205870383488251460919238212347556348069140330291349353064889
51740403492162431817536418564325087605781316930326921180863106158020729609427
23974584019038029591735364239087266235524239708638125397509377553254003578531
16862284280159388053991302975623049654305450270636569747983104898247219140785
3, the offset is 7. The DER encoding is 0x02820181", followed by 385 bytes. 0x02 indicates that it's an integer, "82" in binary means that it's stored in long form and it's going to be followed by 2 length octets describing its length, and 0x0181 is 385, which is the number of following content octets.

3. This is the public exponent e , its value is 65537 and its offset is 396. Its DER encoding is 0x0203010001. 0x02 means it's an integer, 0x03 means it's stored in short form and has 3 content octets, and the rest are the content octets storing 65537 in hexadecimal.
4. This is the private exponent d , its value is
58209754628319664149826641968518592296692968229737784363362023667372784649246
07760459104739225180048479183745660976063709460792713592319266083668721647730
11371480552184687807691141860476487283368801828034856628428782099416417047623
67740071449559483042607834633153235998688944951906313582171428038517728658449
14004813492331630899861798665047361511399981869123099539179771165516845054307
06436938441793114802519467749695854223432380919070910305039676271657954514909
46158153421931824403510420097744141440236448837960527293002831816761902168820
50900135985937535501517327806106259551920240806279133663985760597176240079301
72046085739245691277265857268523353146432987431322131835718463493329997602602
50390413987997714654508984557851541865055138605044456484075107608527542171179
65220367778069277169793810230598177705047613907516107314976454002965046255679
91422385751481833323422170850083014314725921513563751414821736240809073907145
, its offset is 401. Its DER encoding is 0x02820180. 0x02 means it's an integer, 0x82 means it's stored in long form and will be followed by 2 length octets, and 0x0180 means there will be 384 total content octets. It's then followed by those 384 octets which store the value of the exponent.
5. This is the first prime, p . Its value is
20437982203578432242760168996965962431729468621492083904405998659859316248050
23432756797066913470981943431387032965216025349254664921513934953745982247004
23101058191951745124153415925452120393248868916397811595584830656990953082230
62682234630166108601709461497553856263873962409136772870453021594083253513986
33945778211739845106768307481060910593836846240715696251821342653523254791690
41211210444547886865633808300997278623646767034824119484065750773207996203446
7, its offset is 789. Its DER encoding is 0x0281c1 followed by 193 more octets. 0x02 means it's an integer, 0x81 means it's stored in long form and has one additional length octet, and 0xc1 is that length octet, which tells us there will be 193 content octets. The following content octets store the value of p .
6. This is the second prime, q . Its value is
18231784721892085365969435136310129414623420717521277362329499259306427958184
49671366026208580391535871208298954394564588469464121181299890875782322236127
23248818729881928190545132362494447039392559906187267142460745366314172077972
81616905319005443890640833595703478858901702843530463366115980956396261685667
49858389289703671710237842643133673748195447065725875862804543338954032275232
09721703890359841581821854793771204619795304468138807204011394679699271026295
9, and its offset is 985. Its DER encoding is identical to the previous encoding: 0x0281c1 followed by 193 more octets. 0x02 means it's an integer, 0x81 means it's stored in long form and has one additional length octet, and 0xc1 is that length octet, which tells us there will be 193 content octets. The following content octets store the value of q .
7. This is one of the exponents used to encrypt and decrypt messages more efficiently using the Chinese Remainder Theorem (https://www.di-mgt.com.au/crt_rsa.html), and it's equal to

$d \bmod (p - 1)$. Its value is

20236212600366273528429854375591212630955417837994130408119157926640997167035
10544449525636999178052372083902292889709139644981235130644357220327094435328
20468249478550265363173705836437250596120040648565756663220770882587590910568
76839803548399816701480491273269599355521025080929630461627730461267105917612
54813945529392534735770564298732662288998168249386476796018843169310832101450
94895943447925788161053722639908958449249106808211195405969552583631641265875
, its offset is 1181. Its DER encoding is also very similar to the encoding of p and q : 0x0281c0,
followed by 192 more octets. 0x02 means it's an integer, 0x81 means its length is stored in long
form and it has one additional length octet. 0xc0 means there are 192 content octets. The rest
of the encoding are the content octets that store the value of the exponent.

8. This is the other exponent, equal to $d \bmod (q - 1)$. Its value is

21309408573735961960926785349365334286893724567223550756892131822678370715732
06659240392565684391895383288153255514040122019026682370074486795015424619487
40419297720508349472752142132948939426850025005934734014564186567277500971553
74396981055523093855731691310723507035596234765314761033377849783480379711035
45160025933309447380933194172208666998974186264775654837585303260202311248344
8837576510980917443837151650522413711149915806734261000392561198200961534117,
and its offset is 1376. Its DER encoding is identical to that of the first exponent: 0x0281c0,
followed by 192 more octets. 0x02 means it's an integer, 0x81 means its length is stored in long
form and it has one additional length octet. 0xc0 means there are 192 content octets. The rest
of the encoding are the content octets that store the value of the exponent.

9. This is another coefficient for the CRT calculation, equal to $q^{-1} \bmod p$. Its value is

31109255182116229166142374706550145937910845481874731870683910340638912209672
77577635671318563805948368413093592486435043577640953569775514460920135679667
89101716904485971987768208452289972674063820703489099535315682226208998894055
92924614677638148961202656078582819936044562253804548690165213415380464144603
62707991603459046993624302836469255904833960483301728625063620883884429023155
05053128111951912698528276528800943276908880962925478760918216965661568669668
, its offset is 1571. Its DER encoding is similar to the previous encodings: 0x0281c0, followed by
192 more octets. 0x02 means it's an integer, 0x81 means its length is stored in long form and it
has one additional length octet. 0xc0 means there are 192 content octets. The rest of the
encoding are the content octets that store the value of the coefficient.

Public key:

After reading the article by Leonardo Giordano, I have to find a way to decode the public key. I converted it to a regular PEM-encoded public key using the following command: `ssh-keygen -f key.pub -e -m pem > key.txt`. As a result, I had the key in the file `key.txt`. I expect that key to just contain the public exponent e and modulus n , following the RFC 8017 format:

```
RSAPublicKey ::= SEQUENCE {  
    modulus          INTEGER, -- n  
    publicExponent   INTEGER  -- e  
}
```

This is the result I get from decoding it:

```
SEQUENCE (2 elem)
  INTEGER (3072 bit) 372620891685503597322444680083102459791792054978738540127139768029752...
  INTEGER 65537
```

1. This is the modulus n . It's value is
37262089168550359732244468008310245979179205497873854012713976802975289974190
66407474197570761873655373903742326464038750995623872540533578250882955759203
83339057637707832436536963900273955333953322576713424485811067517576223130006
96372153375559482713062997249003356394276981688934206614912764986914791786372
15560983217883747746476291259144451371094164990786487346153786509811266598203
96637601432504350115381113087632518052755303265219283538399934204098919369443
84619824561856002820029295498248330541105537781027612851044216309233382245285
22623972521560200486496532284038262533195312214119407151227918350109856017388
12352849388066774205870383488251460919238212347556348069140330291349353064889
51740403492162431817536418564325087605781316930326921180863106158020729609427
23974584019038029591735364239087266235524239708638125397509377553254003578531
16862284280159388053991302975623049654305450270636569747983104898247219140785
3, it's offset is 4. Its DER encoding is 0x02810181, followed by additional 385 octets. 0x02 means
it's an integer, 0x81 means its length is stored in long form with two extra octets for storing the
length. 0x0181 means it has 385 content octets. The next 385 octets are those content octets,
which store the value of n .
2. This is the public exponent e . Its value is 65537, its offset is 393. Its DER encoding is
0x0203010001. 0x02 means it's an integer, 0x03 means its length is stored in short form and it
has 3 content octets. 0x010001 is the 3 content octets, which store 65537.

Sanity Check:

- The public exponents are the same in the public and private key, I tested it with Python.
- $e * d \bmod \lambda(n) = e * d \bmod \text{lcm}(p - 1, q - 1) = 1$ when tested with python. I'm not sure why this is failing because I copied all values directly from the ASN.1 decoder in the browser.