Alden Harcourt and Nathan Parker

Id_rsa_homework:

-----BEGIN RSA PRIVATE KEY-----
MIIG5QIBAAKCAYEAyjbZdj5OkIQysjVwxB/1hCXn5rkoGs0FfHByycvH5g8CDe9F
9078KkjjjgzIfZfstj+1YABA8fXMHcYr7rXE5gk/g+kepLy1lTb1lMpfJ2RWbfDy
Yiq4Wmy5WmrMi+s91pvn92uGCqOufxG5hPzVj/2o02u+kni3MoIM+BhrfKOfwEJs
6H6woIns+atpQIrUalzFAkljXBQmZBc0t2QcxSl1sLhBMZB9Vg64ynjbp9fPgokh
Hq9eYP1Aa3J5EVGMvyXcGOjn55Xq5GCvh//nUQ3BQtebPuKKU4Q4jO8TBvh6gsUO
K0Q2C7ScT8UxL2QYwhckKqaDKA3xHxIxO1MnUUpuqutv81c9XhnafRhDNeqL2s7/
jLZlmapp9b1+Y1I0xNYOSw8aE9iHRkjaO6ZUvhYcPTm88zO3F8cyJ8ietZN8kVcx
H7Y0WDZCDXHX7jskXsi45zbJFgdVN8vgfWxfEeYLi5kFEF38QGyHPUsk+12GeLrW
TC1/g2YHVtDth9C5AgMBAAECggGAJhIGhgr7+pxQ+RkzllEYBZ2nV9pjMQyJbGC1
U8WwaGFJ9zqllwaBViqr4NoKQw7/y14aNS1HDObEW5SsP8BsBg0WrqyMjuJSY3nZ
06cWHBH5bbBvyciWNbwDd4Dk6rDKzyVCGmRdc5JWb2j0XxPE11uf1dISqnvcrb8r
VuguEGSz1lwLKgh0E310Jps9cR/SFwZJNwF/Gd5XTf/Kdn58JiiEllVSPNUhq7qQ
0tnHLQXl9QMBP8gvgh4b6z69iWOrQKWgzJfQKwZR+squQRaErwykol0qO7idQ5BQ
/ZYdQ4q8XS/972cH3Ev4/jD6iuNNZeiko9gAaFzx3Jd9A7ZTowpgSezRx4vLqZYq
qpGx7kbgm8ptrat7gpoiO0ieY8eti2tBTDR6x3ITwAB3/AVfDrar3NSRfzUYidWL
kv56gTxyBYdyJqcAtbvHvgBkNFk4qFulw0N/wZcHsYt/jMOFq0nEaqTwJnks0/Aj
Dr/g728WmNR7CK7xSTFqEE+wP4LJAoHBAO+5GaAiDsXkHqEx6aJOSXt50CIW9eEU
/RWcBbzDNsPywPVTe3xpcBv1TsPqA1omL3EQt87lzV06hdHHGxzwlfBTKtg1rkHF
yJVLiaEUAVL2F0YdEow3VCE1sLWP70UCxnhRRhm+rcPhcxxp3tRt2KEDnDAAo27w
OTb/q9wFC9MdtI9dnxsPrwctIMc961U7yIDAMlM61bccMHEF4QwKiqKFp+w2TtT+
lHwJ0d7057Pg1C6eyKlkuZybjTbU+aop7QKBwQDX8cQofWJc2PW0UY5j/fQSfMiD
ceJg4x6hVICUNQ4NzbelMVrsvyw9CYs6FrCkVNz6Nt52cTVtQ/kFVO3dADRWOn4B
OZNAerOcqv9V3M4PoSTXsJPrB3ZYit2CeJyQtA1fT2naHVog1BqVXLnZk5vTQZtr
FradgxmlRa9i/YXsSQk47RjY8QUBb5hE10+aOSG4HskuYscVCgLKIXFlbgVM+kCn
UHqCrVGF2gUy9Di/6zMy7AjhlCq+o6bvos6ruH0CgcEAlriFcRYYvbk4vNa5809P
ii/Debt/6n2cxhprzQvcAgU95sEPUeClGR7539nhM6vwhiEhwBbiMOybuJJ77l0j
aI+Rz5CouDfXbm6o4LrIPlX1uiKLR9d9sMemC/GsWXJuQLWw4nztmcvE6Sdzb5KE
8m9noxKzrwugnYDQmCwgDCORR5KAd647uMJZ6ot2zAcjgDfXLFdAiblSh61PmpeC
JL7uHmji1a4ew4IVDx5iE8mW/pzcwwxOWzW96qyrMJ7dAoHBALgv68tJXwuotrIt
2hDpvDPEoVaUXa2cKzUaGW3QbwNREzHgjhhe20HYkRtj3RjdlXoKMOe/mf1vu8hT
b2tQUFO4ll+zFykpP2gC5jT7V/s2zHD4mMlgJE5Ta6psa8Z0/O7tknDLFmPn5iC9
7Xtqjr+7NvA5eFuTRd2VOYpqib9HcIlQmh/4O/fEkpEtQSVfU6ZzA8//yqTkXArC
SbFlDTpiPaE4YLZzVJShqEuUyY7Q82OctdqKgYcHmUzOhg8sFQKBwQC6yccGR+0e
pniKwxh/dnwgekHEvGSvJq4ZJTavIP5DPmdQCEhu58T9AQ4whnX9vxiqiu6GZXF1
0BlKBHYC+QF+D4RN5R0Sula1E486pzRZ5EDjfd5tasqg1vlbDJfKYyM4YG919GNQ
EDdzyZdhVD1leK3YFPL8d2UWACUDJnYMb6S3NRb/OnmJDR6fZ8RMlwl1SVy8DOlQ
JHnOGmpB8PWL6a/VZNFQK9dju459ws+Ki/Jr2/R0HPAto0SR/zTL2vE=
-----END RSA PRIVATE KEY-----

Id_rsa_homework.pub:
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQDKNtl2Pk6QhDKyNXDEH/WEJefmuSgazQV8cHLJ
y8fmDwIN70X3TvwqSOOODMh9l+y2P7VgAEDx9cwdxivutcTmCT+D6R6kvLWVNvUgyl8nZFZt
8PJiKrhabLlaasyL6z3Wm+f3a4YKo65/EbmE/NWP/ajTa76SeLcyggz4GGt8o5/AQmzofrCgiez5q
2lAitRqXMUCSWNcFCZkFzS3ZBzFKXWwuEExkH1WDrjKeNun18+CiSEer15g/UBrcnkRUYy/J
dwY6OfnlerkYK+H/+dRDcFC15s+4opThDiM7xMG+HqCxQ4rRDYLtJxPxTEvZBjCFyQqpoMoDf
EfEjE7UydRSm6q62/zVz1eGdp9GEM16ovazv+MtmWZqmn1vX5jUjTE1g5LDxoT2IdGSNo7plS
+Fhw9ObzzM7cXxzInyJ61k3yRVzEftjRYNkINcdfuOyReyLjnNskWB1U3y+B9bF8R5guLmQUQX
fxAbIc9SyT7XYZ4utZMLX+DZgdW0O2H0Lk= kali@kali

========Private Key=======
We expect these to be in the file:

- Version
- Modulus-n
- Public exponent-e
- Private exponent-d
- Prime1-p
- Prime2-q
- Exponent1-d mod(p-1)
- Exponent2-d mod(q-1)
- coefficient-(inverse of q) mod p
- otherPrimeInfos

To decode the private key we used cat id_rsa_homework and copy/pasted the key into [Lapo Luchini's ASN.1 decoder]

Integers in decoded file:

- Version, 0
    - This integer clarifies that the RSA key is using the two prime method and not the multi prime method
    - Found at offset 4. The DER encoding is 02 01 and then the version number
- Modulus
    - This is n, the product of primes p and q
    - Value:
      458900450114654612950077066674856086863663640214015092485387231023940672683357541869327145912839460192470604788279513217876105529722795880064859389499931776582097107566868824025233250316245931673564582702366606096152122514645514240325565384824873642140799192173020603182492952166937493602541020625556311971897340375426541926204125014000353251225620241380628766184270724525222644840424142876963266011149490130245157462840541029304104990012292836098031

449806401700036088808397038053954974863993411338055017811041796871289783711967924212255577146560703364917519017161459481420161581719035175647072919739292981245104020882441980419390614040106533145775665726786237479079021036533113189470204580982336326240458185240974180034951944473563911330409372257617647139446711049240289041062926893985670529998927653231594964524832939982487794694089650980311910849101241063827745815498556166099807917162722451706057577652312468872818993276731
- publicExponent
    - This is the public exponent, e
    - Value: 65537
- privateExponent
    - This is the private exponent, d
    - Value: 863961304872006033520686922985139970973238601825018725091314883194057095976869101477744326082300025876191854552418751825497707871657951533664993450622535090919359317898867965460021126848497860438267940807963496851759000815867933755627658956110548763836344482143951464419669696554886287870203454993122443087912985675603153570726398291841762453308408280555241776639856010277318095635072293038651640703506108606135446443269910964808993304433629119153394060509769100456125174841034691993243254509922487126837183525726831766988656132930032691779379252173668927947228304485017524872780708022039447452274742877760964016783966557171561621753842902240403476367985883842110081188264538205143728125521423044769634985114232529722352642868060899672822143528210564983113808155622603879422892455246166015485999682886304854508200876941572873334662473142199413350086041013320502195946055900155469962847188036163591240940909239785514107634377
- prime1
    - This is the prime p which is a factor of n
    - Value: 225706031187796847918700135699511141529954199418532446301841035714620370902759620086877651572798434523136318609142035072323139379403654574746744396595612407321168537919399645402397912610581272885363421508808743703250507137882255946794109130526759487163925217080941359835024205601396644520241447594076340792691715025534159371908793501489172567601354670389634678145001561966464871812155368419062411714467439205038248323256468629735808390761453679992864378653941194 9
- prime2
    - This is the prime q which is a factor of n
    - Value:

20331776147037482824278264395323262327413217783312302452027265277781755703842218569076258883689449268256734518093601472927276833120332593749271006685479238244079756680244054411141488953452897799059194740361392220704132998414846048999797308547447589897451642736774401196584411129922783759457497803853161857465591699934332716403501274088767729293434314542469346167074386517632835049929640103991879846595772379069054410056363490848010913762517874645193762711785572477

- exponent1
    - This is equal to d mod (p-1)
    - Value:
      14190788432630680559821229368903606492121645462930298075693669036759284653017700147519406797774020010872990842256584151174260277596361729637364546533595082539128659543416485937418261423499704516900986897890450103441471453707735105799245108447678600895050946137022122971759727164510961346196116465681852422849477573930962718646721449301709500965887695795359715752879731084468597956909626555544908486343028872916979755191392327827431809117484320130774946552993952 93

- exponent2
    - This is equal to d mod (q-1)
    - Value:(1536 bit)
      17341745194977619549175743495036010815876092297013509845825596254981496896242979931836882911017540086459300896087923291196756758694988657674145902356098142096919515978286500717585456932863322002374382818735393193846839655132085376093499393479923902950663737695392713314431115213567201571172233543457709304216993613296752437771630036777515408086633272023582006826575692081559391587286829610342891672564460994820957206886806884277781437484336888700912250212329175 25

- Coefficient
    - This is the modular inverse of q mod p
    - Value: (1536 bit)
    - 17586636839353586778680696833250751255611709331412428674957372329612343564947118438483031241213555166401569984708591603201115476908967120030786560554302587677890834251958034551747921844567550611014513398655046177685268212177264089715025562509531013397052326252330485077718635256193359863018731368896591072451265585843071577756595647786946394358619857247645457653080257488494523546582370570043913449931156838377618005100150122059561121750551913063552899160575741 29

====Public Key====

We decoded the file by using ssh-keygen -e -f id_rsa_homework.pub -m PKCS8 to convert from PEM to OpenSSH public. Then we used the [Lapo Luchini's ASN.1 decoder] to decode the key into ASN.1

What we expect to find

- Modulus
- publicExponent

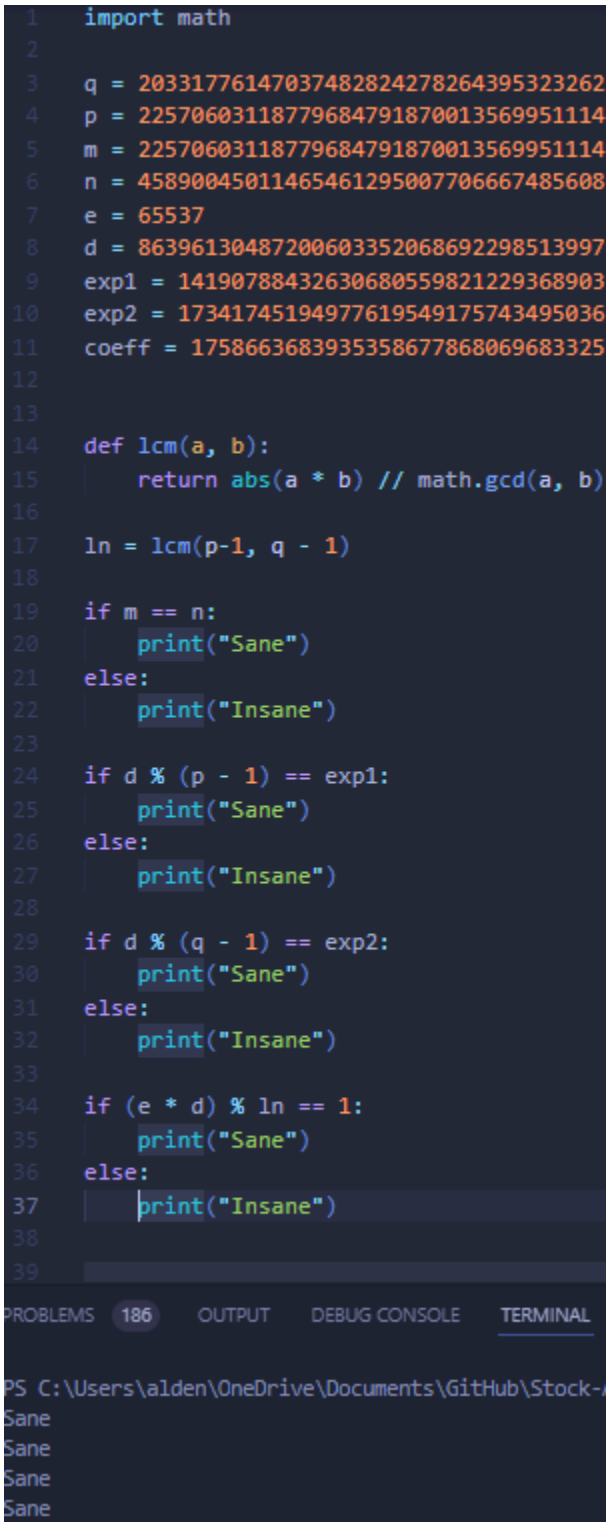
What we found:

- Sequence, contains the modulus and the publicExponent
    - Modulus
        - This is n
        - Value: (3072 bit)

          458900450114654612950077066674856086863663640214015092485387231023940672683357541869327145912839460192470604788279513217876105529722795880064859389499931776582097107566868824025233250316245931673564582702366606096152122514645514240325565384824873642140799192173020603182492952166937493602541020625556311971897340375426541926204125014000353251225620241380628766184270724525222644840424142876963266011149490130245157462840541029304104990012292836098034498064017000360888083970380539549748639934113380550178110417968712897837119679242122555771465607033649175190171614594814201615817190351756470729197392929812451040208824419804193906140401065331457756657267862374790790210365331131894702045809823363262404581852409741800349519444735639113304093722576176471394467110492402890410629268939856705299989276532315949645248329399824879469408965098031191084910124106382774581549855616609980791716272245170605757765231246887281899327673

    - publicExponent
        - This is e
        - 65537

======Sanity Check======
The following relationships were confirmed through the following python code:



```
import math

q = 20331776147037482824278264395323262
p = 22570603118779684791870013569951114
m = 22570603118779684791870013569951114
n = 45890045011465461295007706667485608
e = 65537
d = 86396130487200603352068692298513997
exp1 = 14190788432630680559821229368903
exp2 = 17341745194977619549175743495036
coeff = 17586636839353586778680696833325


def lcm(a, b):
    return abs(a * b) // math.gcd(a, b)

ln = lcm(p-1, q - 1)

if m == n:
    print("Sane")
else:
    print("Insane")

if d % (p - 1) == exp1:
    print("Sane")
else:
    print("Insane")

if d % (q - 1) == exp2:
    print("Sane")
else:
    print("Insane")

if (e * d) % ln == 1:
    print("Sane")
else:
    print("Insane")


PROBLEMS 186  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\alden\OneDrive\Documents\GitHub\Stock-
Sane
Sane
Sane
Sane
```