Alden Harcourt and Nathan Parker

Id_rsa_homework:

```
-----BEGIN RSA PRIVATE KEY-----
MIIG5QIBAAKCAYEAyjbZdj5OkIQysjVwxB/1hCXn5rkoGs0FfHByycvH5g8CDe9F
9078KkjjjgzIfZfstj+1YABA8fXMHcYr7rXE5gk/g+kepLy1ITb1IMpfJ2RWbfDy
Yiq4Wmy5WmrMi+s91pvn92uGCqOufxG5hPzVj/2o02u+kni3MoIM+BhrfKOfwEJs
6H6woIns+atpQIrUaIzFAkIjXBQmZBc0t2QcxSI1sLhBMZB9Vg64ynjbp9fPgokh
Hq9eYP1Aa3J5EVGMvyXcGOjn55Xq5GCvh//nUQ3BQtebPuKKU4Q4jO8TBvh6gsUO
K0Q2C7ScT8UxL2QYwhckKqaDKA3xHxIxO1MnUUpuqutv81c9XhnafRhDNeqL2s7/
jLZImapp9b1+Y1I0xNYOSw8aE9iHRkjaO6ZUvhYcPTm88zO3F8cyJ8ietZN8kVcx
H7Y0WDZCDXHX7jskXsi45zbJFgdVN8vgfWxfEeYLi5kFEF38QGyHPUsk+12GeLrW
TC1/g2YHVtDth9C5AgMBAAECggGAJhIGhgr7+pxQ+RkzIIEYBZ2nV9pjMQyJbGC1
U8WwaGFJ9zqIIwaBViqr4NoKQw7/y14aNS1HDObEW5SsP8BsBg0WrqyMjuJSY3nZ
06cWHBH5bbBvyciWNbwDd4Dk6rDKzyVCGmRdc5JWb2j0XxPE11uf1dISqnvcrb8r
VuguEGSz1IwLKgh0E310Jps9cR/SFwZJNwF/Gd5XTf/Kdn58JiiEIIVSPNUhq7qQ
0tnHLQXI9QMBP8gvgh4b6z69iWOrQKWgzJfQKwZR+squQRaErwykoI0qO7idQ5BQ
/ZYdQ4q8XS/972cH3Ev4/jD6iuNNZeiko9gAaFzx3Jd9A7ZTowpgSezRx4vLqZYq
qpGx7kbgm8ptrat7gpoiO0ieY8eti2tBTDR6x3ITwAB3/AVfDrar3NSRfzUYidWL
kv56gTxyBYdyJqcAtbvHvgBkNFk4qFuIw0N/wZcHsYt/jMOFq0nEaqTwJnks0/Aj
Dr/g728WmNR7CK7xSTFqEE+wP4LJAoHBAO+5GaAiDsXkHqEx6aJOSXt50CIW9eEU
/RWcBbzDNsPywPVTe3xpcBv1TsPqA1omL3EQt87IzV06hdHHGxzwIfBTKtg1rkHF
yJVLiaEUAVL2F0YdEow3VCE1sLWP70UCxnhRRhm+rcPhcxxp3tRt2KEDnDAAo27w
OTb/q9wFC9MdtI9dnxsPrwctIMc961U7yIDAMIM61bccMHEF4QwKiqKFp+w2TtT+
IHwJ0d7057Pg1C6eyKIkuZybjTbU+aop7QKBwQDX8cQofWJc2PW0UY5j/fQSfMiD
ceJg4x6hVICUNQ4NzbeIMVrsvyw9CYs6FrCkVNz6Nt52cTVtQ/kFVO3dADRWOn4B
OZNAerOcqv9V3M4PoSTXsJPrB3ZYit2CeJyQtA1fT2naHVog1BqVXLnZk5vTQZtr
FradgxmIRa9i/YXsSQk47RjY8QUBb5hE10+aOSG4HskuYscVCgLKIXFIbgVM+kCn
UHqCrVGF2gUy9Di/6zMy7AjhICq+o6bvos6ruH0CgcEAIriFcRYYvbk4vNa5809P
ii/Debt/6n2cxhprzQvcAgU95sEPUeCIGR7539nhM6vwhiEhwBbiMOybuJJ77I0j
aI+Rz5CouDfXbm6o4LrIPIX1uiKLR9d9sMemC/GsWXJuQLWw4nztmcvE6Sdzb5KE
8m9noxKzrwugnYDQmCwgDCORR5KAd647uMJZ6ot2zAcjgDfXLFdAibISh61PmpeC
JL7uHmji1a4ew4IVDx5iE8mW/pzcwwxOWzW96qyrMJ7dAoHBALgv68tJXwuotrIt
2hDpvDPEoVaUXa2cKzUaGW3QbwNREzHgjhhe20HYkRtj3RjdIXoKMOe/mf1vu8hT
b2tQUFO4II+zFykpP2gC5jT7V/s2zHD4mMIgJE5Ta6psa8Z0/O7tknDLFmPn5iC9
7Xtqjr+7NvA5eFuTRd2VOYpqib9HcIIQmh/4O/fEkpEtQSVfU6ZzA8//yqTkXArC
SbFIDTpiPaE4YLZzVJShqEuUyY7Q82OctdqKgYcHmUzOhg8sFQKBwQC6yccGR+0e
pniKwxh/dnwgekHEvGSvJq4ZJTavIP5DPmdQCEhu58T9AQ4whnX9vxiqiu6GZXF1
0BIKBHYC+QF+D4RN5R0SuIa1E486pzRZ5EDjfd5tasqg1vIbDJfKYyM4YG919GNQ
EDdzyZdhVD1IeK3YFPL8d2UWACUDJnYMb6S3NRb/OnmJDR6fZ8RMIwI1SVy8DOIQ
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
dwY6OfnIerkYK+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:
      458900450114654612950077066674856086863663640214015092485387231023940672683357541869327145912839460192470604788279513217876105529722795880064859389499931776582097107566868824025233250316245931673564582702366606096152122514645514240325565384824873642140799192173020603182492952166937493602541020625556311971897340375426541926204125014000353251225620241380628766184270724525222644840424142876963266011149490130245157462840541029304104990012292836098 03

    44980640170003608880839703805395497486399341133805501781104179687
    1289783711967924212255577146560703364917519017161459481420161581
    7190351756470729197392929812451040208824419804193906140401065331
    4577566572678623747907902103653311318947020458098233632624045818
    52409741800349519444735639113304093722576176471394467110492402890
    4106292689398567052999892765323159496452483293998248779469408965
    0980311910849101241063827745815498556166099807917162722451706057
    57765231246887281899327673

- publicExponent
    - This is the public exponent, e
    - Value: 65537
- privateExponent
    - This is the private exponent, d
    - Value:
      8639613048720060335206869229851399709732386018250187250913148831
      9405709597686910147774432608230002587619185455241875182549770787
      1657951533664993450622535090919359317898867965460021126848497860
      4382679408079634968517590008158679337556276589561105487638363444
      8214395146441966969655488628787020345499312244308791298567560315
      3570726398291841762453308408280555241776639856010277318095635072
      2930386516407035061086061354464432699109648089933044336291191533
      9406050976910045612517484103469199324325450992248712683718352572
      6831766988656132930032691779379252173668927947228304485017524872
      7807080220394474522747428777609640167839665571715616217538429022
      40403476367985883842110081188264538205143728125521423044769634985
      11423252972235264286806089967282214352821056498311380815562260387
      9422892455246166015485999682886304854508200876941572873334662473
      1421994133500860410133205021959460559001554699628471880361635912
      40940909239785514107634377
- prime1
    - This is the prime p which is a factor of n
    - Value:
      22570603118779684791870013569951114152995419941853244630184103571
      4620370902759620086877651572798434523136318609142035072323139379
      4036545747467443965956124073211685379193996454023979126105812728
      8536342150880874370325050713788225594679410913052675948716392521
      7080941359835024205601396644520241447594076340792691715025534159
      3719087935014891725676013546703896346781450015619664648718121553
      6841906241171446743920503824832325646862973580839076145367999286
      43786539411949
- prime2
    - This is the prime q which is a factor of n
    - Value:

    - 20331776147037482824278264395323262327413217783312302452027265277
      778175570384221856907625888368944926825673451809360147292727683
      3120332593749271006685479238244079756680244054411141488953452897799
      059194740361392220704132998414846048999797308547447589897451642
      736774401196584411129922783759457497803853161857465591699934332716
      403501274088767729293434314542469346167074386517632835049929640
      103991879846595772379069054410056363490848010913762517874645193
      76271785572477

- exponent1
    - This is equal to d mod (p-1)
    - Value:
      1419078843263068055982122936890360649212164546293029807569366903
      675928465301770014751940679777402001087299084225658415117426027
      7596361729637364546533595082539128659543416485937418261423499704
      5169009868978904501034414714537077351057992451084476786008950509
      4613702212297175972716451096134619611646568185242284947757393096
      2718646721449301709500965887695795359715752879731084468597956909
      6265555449084863430288729169797551913923278274318091174843201307
      7494655299395293

- exponent2
    - This is equal to d mod (q-1)
    - Value:(1536 bit)
      1734174519497761954917574349503601081587609229701350984582559625
      49814968962429799318368829110175400864593008960879232911967567586
      9498865767414590235609814209691951597828650071758545693286332200
      2374382818735393193846839655132085376093499393479923902950663737
      69539271331443111521356720157117223354345770930421699361329675243
      7771630036777515408086633272023582006826575692081559391587286829
      6103428916725644609948209572068868068842777814374843368887009122
      5021232917525

- Coefficient
    - This is the modular inverse of q mod p
    - Value: (1536 bit)
    - 1758663683935358677868069683325075125561170933141242867495737232
      96123435649471184384830312412135551664015699847085916032011154769
      0896712003078656055430258767789083425195803455174792184456755061
      1014513398655046177685268212177264089715025562509531013397052326
      2523304850777186352561933598630187313688965910724512655858430715
      7775659564778694639435861985724764545765308025748849452354658237
      05700439134499311568383776180051001501220595611217505519130635528
      9916057574129

====Public Key====

What we expect to find

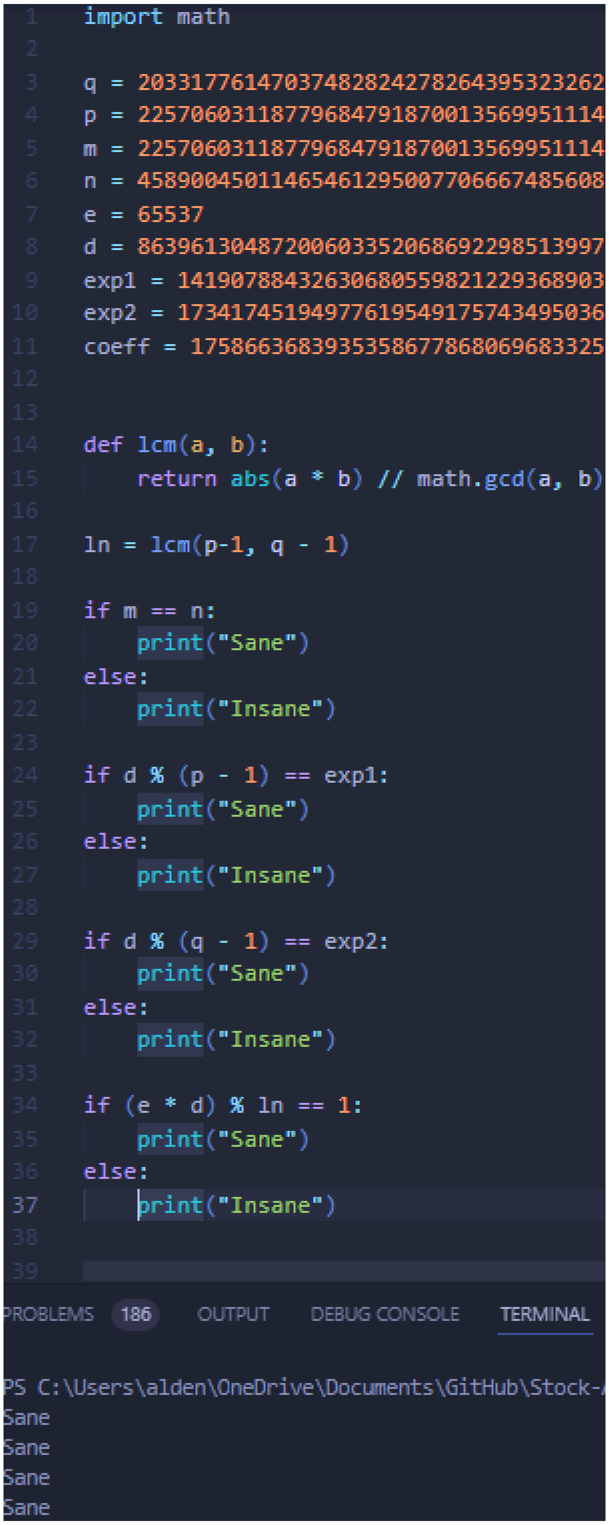
- Modulus
- publicExponent

What we found:

- Sequence, contains the modulus and the publicExponent
    - Modulus
        - This is n
        - Value: (3072 bit)
    4589004501146546129500770666748560868636636402140150924853872310239406726833575418693271459128394601924706047882795132178761055297227958800648593894999317765820971075668688240252332503162459316735645827023666060961521225146455142403255653848248736421407991921730206031824929521669374936025410206255563119718973403754265419262041250140003532512256202413806287661842707245252226448404241428769632660111494901302451574628405410293041049900122928360980344980640170003608880839703805395497486399341133805501781104179687128978371196792421225557714656070336491751901716145948142016158171903517564707291973929298124510402088244198041939061404010653314577566572678623747907902103653311318947020458098233632624045818524097418003495194447356391133040937225761764713944671104924028904106292689398567052999892765323159496452483293998248779469408965098031191084910124106382774581549855616609980791716272245170605757765231246887281899327673
    - publicExponent
        - This is e
        - 65537

======Sanity Check======

The following relationships were confirmed through the following python code:



```python
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")
```

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