

David Toledo, Angel Marti
Id_rsa_homework

-----BEGIN RSA PRIVATE KEY-----

MIIG4wIBAAKCAyEA8YwM+2qkOEblo1Js5xplUQ5DIStsKgWgtEPBcTPzbqsHVOD
qnl8tqb1vj2ffD38Wth36klKZsdr/MDfbO5Fe9gU3QNVJ8oEZDC/Bo6qXFHBGFDG
r3VGw3kjcM9Z3rR0M5/iHVh0uSRYzO1kOwvIBUz945NYHTyyZnbUYdq8gb2VYQv3
uxl2YEOSE9Y0TtPKIScEqG/Ea1BBFzaAr6LPMPuTaSmVA4b887D+sX9aMeyf3yi
iGlRh0xiBK2JxHmtL/uf3f6C/DGYDFglQAxz4SVSS9xfwIzElqFYvQlcuvXKZuA
GzvX0EE9TDRBfIFGIYiJQ5q3Dq3rnVt79MhJhRbf5xCv4z0iMasdb84yxGYaNVgU
q0u5ix+m3c1pGTd/n9SyDgt7bRVk3Rt5Fc4vwXrMX52PKVCplNADxJD3j1lumJG
5GEfd1dt8zDzP93K27jq6L3FlzyYLSGfboobzkm+yu4i4t5tm+ieZCXuGbScCMFs
Nzo9VeV0+5y/L73nAgMBAAECggGALH1W0symmLL+QvnAk0R1A3D0I7qOqEgddVTa
Rn68o7D5WKf0Q+cCglX6B2NER9cG7xi5aJAMPQUQMapoF+2rpvbl+YkPiLocifE
V4nZeQ2Z2keugMzCDuj2DSYz8GAecw/EzxDM1t8J+BzVdEa0Pb2zmcle3au7VEIY
1jk6tU7c/7X7mBuVgR+Q/VRb6BT8YEKCima7gZ7+tbYAdbbw4G1aLDTiaeL7ZQm1
cTAcf7sgdnjXWvbH1nDMMP5hvK2ITaxJSla9hel33laDsnx+fALAsbYWDqK7ijJL
6RbthWMZMMJIVYmdEH2hQ9/GTl2pnXn0ohTY+rVURGT0eyof9nxy8NILNp8eWdGk
syfnd3gv27V4EeNIZ4nYECdgC/MC4wKbhsnc53mrL/clj1oLFaaf60JIDXukITfr
aL8VWW6xvHijp+QBzDi2ups0Qw6xgVKYCqc5SBhqWtm4tAtMs2SFEKk+Xyd0qhNU
iK9K+33zchNd14/icradQA0lf96hAoHBAPwvgajhvC3CJm0Ky/sNBTpiECQXfqSG
EXc2C/8abX5QNzn88VRBSV08Tq/0hQmK5j/9CeO4XL2uFMadnupshbj/reemeYze
nDmKVeKcZ45fI0I4pNUDCfTzKBNnrBgUa1lUiCNO9seyzFPWKiwymaAN6vQt945v
TI/5zXM6h6zUKvHnsUpZED0GMSMcAfskFavetJF9DyygDZWP1PgK2m5z/X6np11/
uDG/iT0t3Ga3vZZZQIC+wdJB6LxsDwn79wKBwQD1M1nEXu1DoqqC2m2lNtPsS0my
H1eDBxtqaQlyYZsFhDvoKIYy0mBBofXPcaJnIQwCP+Ok/Yo7/z70PLBIOKRVQRfM
uSAXIkIVkl4ajaG95F4ZjC31NvfC+xF21ET2C0meq1jyKzY2VCOIcupKuhhm6+7D
CpKVb7O1kppRI2GqvWSM2wj2Dzv+I8iom4JTGvR99XTmFQC22IzQMv5D8ABxQ15m
4ncvpsbbhD4E+fwipgaTLHceoBZGxWAJGgwwcZECgcB56Hk0a5MtR5qNwddjkCLD
213UJKtq2wicaTsZYewmUJ6x4I3E3h6Z/KpJMtoRB91Qd4ENXTUv2HRgoxeMWVO3
X5nwzrAZog9BXZcxCcyhrWoAfVqpQpYXBGiOIOK3bT+IGTbKYsS3Or69IF0kqn1L
Ow8mZQY0C6iGitp/Zn7p1FLCR++TckSAS3vEh8iJYCM9x1XgsuTf/Ks0bqhUE+MN
2/JiBkfkChBXtdCiq4XM/eufP5zoLZ7Chmv8kfwfJncCgcEA10EeM/rVtYOOxtL0
Yl6SWqT5bEqBa8Cle9D7/xLaoWwlnznTdEglQljPTxRF6t0jCrKFqOj/Oo5n+Mdy
X1nrQgCj4RA/sRUyrHW7mmntrKNXTHw4OEXqGZyJB1VvuMPN/GIV8tXzSJxysqMA
NVvLYkcK0uRsLI3kKKkrmcFC0z3ykIkVA9X557AITqQ+M7C9l/qghf+4lcXScmd
JG0EJmpq7E0xLn7tofYk7/95Lf2sVfU6GYOWKsjl9xSL0NdRAoHAT+rgNIfr5yIK
yq0To8SswEul2YfxoLSJZ0lInUwPyooAVzirpyUCWXYVqVzXAUPRsbBapCT5RAJM
hqETfQUsgbwIO6okSK5Hu6IRDfIMkEqV8NkHPfb1bwOkDZPJUP2aDV953DNR6o1w
wdF6osEwkQwbLVnQbZEfM8uUgbhJ6t8bloYuOZ1RabLUkLZE3S98qpdNm5XGwpfm
S3x5R3tE7eEgIPUVENuRiBCg9NVUvbiHs06kEXmxNxc8UjBCP/

-----END RSA PRIVATE KEY-----

ld_rsa_homework.pub

ssh-rsa

```
AAAAB3NzaC1yc2EAAAADAQABAAQGDxjAz7aqQ4RuWjUmznGkhRDkOVK2uwqB
aC0Q8FxM/NuqwdU4OqeXy2pvW+PZ98Pfxa2HfqSUpmx2v8wN9s7kV72BTdA1UnygR
kML8GjqpcUcEYUMavdUbDeSNwz1netHQzn+ldWHS5JFjM7WQ7C8gFTP3jk1gdPLJm
dtRh2ryBvZVhC/e7GXZgQ5IT1jRO08qVJwSp8b8RrUEEXNoCvos8w+5NpKZUDhvzss
P6xf1ox7J/fKKlaWuHTGIErYnEea0v+5/d/oL8MZgMV+AhADHPhJVKz3F/AhkSWoVi9C
Vy+5cpm4AbO9fQQT1MNEF8gUaViIlDmrcOreudW3v0yEmFFt/nEK/jPSIqx1vzjLEZho
1WBSrS7mLH6bdzWkZN3+f1LIOC3ttFWTdG3kVzi/DGsxfnY8pUKmUs0APEkPePUI6Y
kbkYR93V23zMPM/3crbuOrovCujPJgtlZ9uihvOSb7K7iLi3m2b6J5kJJe4ZtJxwwWw3Oj1
V5XT7nL8vvec= angel@Angels-MacBook-Pro-2.local
```

After Conversion:

-----BEGIN RSA PUBLIC KEY-----

```
MIIBigKCAyEA8YwM+2qkOEblo1Js5xplUQ5DIstrsKgWgtEPBcTPzbqsHVODqnl8
tqb1vj2ffD38Wth36kIKZsdr/MDfbO5Fe9gU3QNVJ8oEZDC/Bo6qXFHBGFDGr3VG
w3kjcM9Z3rR0M5/iHVh0uSRYzO1kOwvlBUz945NYHTyyZnbUYdq8gb2VYQv3uxl2
YEOSE9Y0TtPKIScEqfG/Ea1BBFzaAr6LPMPuTaSmVA4b887D+sX9aMeyf3yiiGlr
h0xiBK2JxHmtL/uf3f6C/DGYDFglQAxz4SVSs9xfwIzElqFYvQlcvuXKZuAGzvX
0EE9TDRBfIFGIYiJQ5q3Dq3rnVt79MhJhRbf5xCv4z0iMasdb84yxGYaNVgUq0u5
ix+m3c1pGTd/n9SyDgt7bRVk3Rt5Fc4vwxrMX52PKVCplLNADxJD3j1lumJG5GEf
d1dt8zDzP93K27jq6L3FlzyYLSGfboobzkm+yu4i4t5tm+ieZCXuGbScCMFsNzo9
VeV0+5y/L73nAgMBAAE=
```

-----END RSA PUBLIC KEY-----

Private Key:

In a private key, we should expect to see a version, a modulus number, a public exponent, a private exponent, 2 different prime numbers, 2 different exponents derived from taking the $-d \bmod (p-1)$ and $-d \bmod (q-1)$ a coefficient number and other optional "PrimeInfos".

Decoding:

I went to the website and pasted the key into the decoder

Version INTEGER:

Our version was 0. This tells us that our key used 2 prime integers instead of multi-prime integers.

Modulus INTEGER:

Modulus is the RSA modulus n .

548161454685121424703432386096538644940814449999794752680136541495816
915620386152681552127600981793375711918645759995921803308124422804755
229092218710716408616971277930742064835009057965974097655144100077903
437054841673667182765939830784642949753057373489062471524056904112083
380480550978908952720821632334964101044733482762547688571840801637283
067790829490291014939324538271599607780034197625233518575352075393203
965293358451595098902658082269357678855516967760347841833216272279956
870048419957246078019794494332514568089716152646137006976773240695275
257614178263938986363547029552790630512024871185063088647049266964280
708282976208458617326136357066239380878256555063624499157509324497141
725565979133166420394181875409576748645061110828451710690535613064793
115814909979743388843727425468929114816462001915095312711273932134103
872143485962540543809191960625213139461318117313292345634266311214674
3708532247608440823191944679 (3072 bit)

publicExponent INTEGER: is the RSA public exponent e :

Our number was: 65537

privateExponent INTEGER is the RSA private exponent d .

(3070 bit)

100963683407908520646278168554729663901011203227909767911285658366968
368836743843157590608851052859115281727420738955258435505628422229827
431380322139509861122960469595831171164737390156816353095125569251573
498766940718110938296956828927192655240080497358537514591252130697727
367529498311891144976014128262132103448601215655686301612076999505068
036548867094577152468568694042700747143030544570764511691457877261262
570228361534235690355890347606289829279108676287127972161359573193358
113387520701692791140096158477461319132702296788878753571529183782130
683043611318968018554138012445999323879683728220967408829243036925685
99755427123355811983196561981860943411166701202139930031224196104834
665278273860118050749930296925222340719779219131751893496952464371387
190962200373882266714777297006838589844947790666892364863222529046757
366803624721471648995638828085400929947237059019477709091136287687052
8218307404452877672160157345

Prime1 INTEGER: the first prime factor (p) of n :

(1536 bit)

237439851001436079821640738294976777162479692191708917636998678981020
193900343829106044824239640367092203120110313868299249844343173351005
212573416785448110131629175203346043476279438461855844570833054751141
833139983390494539440465239965016017584440981722062719084111538107104
568377617885221077157021944628571196254341991294460326110560181071161
059195048430666273789328162888937961020039307329551139743192486686567
7568167650543024670230034554127942000938882563063

Prime2 INTEGER: the first prime factor (q) of n:

(1536 bit)

230863291218038223736366160597664849625261874986068433111026426913598
211611908620705147551624830811005121084380766677807806572839149548808
999626949085713394892743947945687061110773519962474730284873838031518
125996366038294780619870354824160181170308416633374337032957665934364
457946268758944923796947644639299101180257153270107502791956691554816
986584397909549835350783877113189678731770283779758281767907297577961
2168625859050013624818881234157639412794560377233

Exponent1 INTEGER: $d \bmod (p - 1)$:

114779924616270144267046099606682626261264921011421490450550912443317
526938321754885158095072036353050156202575870937967690530824359902546
594130619606936258557458289204834002218166972701070464193487068489111
866818862837698666463423398497515486718840879837903306579546479679740
907132937305975081944727592471089065238473055315314976143379420732036
155398604289667488898175161030935861316139971245365360913744620149185
2404551922377490037346197181886984917401540175479

Exponent2 INTEGER: $d \bmod (q - 1)$.

(1536 bit)

202668076562054917469892645645443822474177814876687598794218280049743
593827424793247009385349213284855236451892498119753979211058711735227
858698708542477512677788692756141014799062101164243994346394350114703
638295145143677824944733526467467349791283612831956386964877144120142
672978968834526729951169398035198364102777588218702335446078464611185
829213423942660339613297050550881819223872617554981662647863200200693
3898371172753168956111886324439499341384400295761

Coefficient is the CRT coefficient $q^{-1} \bmod p$.
(1535 bit)

```
752445718159952634010242682742315656047003121184041176304419502768807
636498727578929267322986674437294837943271681466241647627676872581956
366815260393554100393286322096625786740309813142564650269079450066832
66477822222992673612260882946147917367243741499271189259552145765977
457542500156772720481138767035542155873453191618511940751395533112818
808431127210248020561021487042325848536130863429710021346558000781950
285387228806971403017098340662984671858360853503
```

$p \cdot q$ should be n

Relationship between e and d and n

Public Key:

What I expect:

Before Conversion: Algorithm Key(Modulus and publicExponent) Comment

After Conversion: Modulus and publicExponent

Decoding:

I changed our pub file to a pem file and then used the decoder to view the file.

Modulus is the RSA modulus n .

Modulus INTEGER:

0x010001

publicExponent INTEGER: is the RSA public exponent e :

publicExponent INTEGER:

```
0x00f18c0cfb6aa43846e5a3526ce71a48510e43952b6bb0a81682d10f05c4cfcdbaac1
d5383aa797cb6a6f5be3d9f7c3dfc5ad877ea494a66c76bfcc0df6cee457bd814dd03552
7ca046430bf068eaa5c51c11850c6af7546c3792370cf59deb474339fe21d5874b92458
cced643b0bc8054cfde393581d3cb26676d461dabc81bd95610bf7bb197660439213d6
344ed3ca952704a9f1bf11ad41045cda02be8b3cc3ee4da4a6540e1bf3cec3fac5fd68c7
b27f7ca288696b874c6204ad89c479ad2ffb9fddfe82fc31980c57e0210031cf84954acf7
17f0219125a8562f42572fb97299b801b3bd7d0413d4c34417c8146958889439ab70ea
deb9d5b7bf4c8498516dfe710afe33d2231ab1d6fce32c4661a355814ab4bb98b1fa6ddc
```


d6919377f9fd4b20e0b7b6d1564dd1b7915ce2fc31acc5f9d8f2950a994b3400f1243de3
d48ba6246e4611f77576df330f33fddcadbb8eae8bdc5233c982d219f6e8a1bce49becae
e22e2de6d9be89e6425ee19b49c70c16c373a3d55e574fb9cbf2fbde7

Sanity Check:

$p * q = n$

```
>>> print(x)
23743985100143607982164073829497677716247969219170891763699867898102019390034382910604482423964036709220312011031
38682992498443431733510052125734167854481101316291752033460434762794384618558445708330547511418331399833904945394
40465239965016017584440981722062719084111538107104568377617885221077157021944628571196254341991294460326110560181
07116105919504843066627378932816288893796102003930732955113974319248668656775681676505430246702300345541279420009
38882563063
>>> y = 230863291218038223736366160597664849625261874986068433111026426913598211611908620705147551624830811005121
08438076667780780657283914954880899962694908571339489274394794568706111077351996247473028487383803151812599636603
82947806198703548241601811703084166333743370329576659343644579462687589449237969476446392991011802571532701075027
91956691554816986584397909549835350783877113189678731770283779758281767907297577961216862585905001362481888123415
7639412794560377233
>>> n = x*y
>>> print(n)
54816145468512142470343238609653864494081444999979475268013654149581691562038615268155212760098179337571191864575
99959218033081244228047552290922187107164086169712779307420648350090579659740976551441000779034370548416736671827
65939830784642949753057373489062471524056904112083380480550978908952720821632334964101044733482762547688571840801
63728306779082940929101493932453827159960778003419762523351857535207539320396529335845159509890265808226935767885
55169677603478418332162722799568700484199572460780197944943325145680897161526461370069767732406952752576141782639
38986363547029552790630512024871185063088647049266964280708282976208458617326136357066239380878256555063624499157
50932449714172556597913316642039418187540957674864506111082845171069053561306479311581490997974338884372742546892
91148164620019150953127112739321341038721434859625405438091919606252131394613181173132923456342663112146743708532
247608440823191944679
>>>
```

$\text{lcm}(p_B - 1, q_B - 1) = \lambda(n_B) =$

27408072734256071235171619304826932247040722499989737634006827074790845781019307634077606380049089
66878559593228799979609016540622114023776145461093553582043084856389653710324175045289829870488275
72050038951718527420836833591382969915392321474876528686744531235762028452056041690240275489454476
36041081616748205052236674138127384428592040081864153389541474514550746966226913579980389001709881
26167592876760376966019826466792257975494513290411346788394277584838799397693454983989881994315747
63657809645139113658277581910032305129013865747362161714751449696571500083307390936233241722867106
66485098568542544414058905272962816757991358767701513658896636783226258568188620183268140404026758
85406949855949784838328223917599862603157879285346797126934019457235319195731128210547229771222914
30450550106149794687968524766632466858012481880051741917450395999800425657895814180766926392159333
1416535036925847396371981013513544874502192

```
>>> from math import lcm
>>> lcm(x-1, y-1)
27408072734256071235171619304826932247040722499989737634006827074790845781019307634077606380049089668785595932287
99979609016540622114023776145461093553582043084856389653710324175045289829870488275720500389517185274208368335913
82969915392321474876528686744531235762028452056041690240275489454476360410816167482050522366741381273844285920400
81864153389541474514550746966226913579980389001709881261675928767603769660198264667922579754945132904113467883942
77584838799397693454983989881994315747636578096451391136582775819100323051290138657473621617147514496965715000833
07390936233241722867106664850985685425444140589052729628167579913587677015136588966367832262585681886201832681404
04026758854069498559497848383282239175998626031578792853467971269340194572353191957311282105472297712229143045055
01061497946879685247666324668580124818800517419174503959998004256578958141807669263921593331416535036925847396371
981013513544874502192
```

$\text{gcd}(e, \lambda(n_B)) = 1.$

```
[>>> from math import gcd
[>>> lmb = lcm(x-1, y-1)
[>>> e = 65537
[>>> gcd(e, lmb)
1
```

$$e_B d_B \bmod \lambda(n_B) = 1$$

```
>>> d = 1009636834079085206462781685547296639010112032279097679112856583669683688367438431575906088510528591
727420738955258435505628422229827431380322139509861122960469595831171164737390156816353095125569251573498766
811093829695682892719265524008049735853751459125213069772736752949831189114497601412826213210344860121565568
120769995050680365488670945771524685686940427007471430305445707645116914578772612625702283615342356903558903
289829279108676287127972161359573193358113387520701692791140096158477461319132702296788878753571529183782130
361131896801855413801244599932387968372822096740882924303692568599755427123355811983196561981860943411116670
399300312241961048346652782738601180507499302969252223407197792191317518934969524643713871909622003738822667
297006838589844947790666892364863222529046757366803624721471648995638828085400929947237059019477709091136287
28218307404452877672160157345
>>> e*d % lmb
1
>>>
```