Lane Maitland

>>> id_rsa_cs338 <<<

-----BEGIN RSA PRIVATE KEY-----
MIIG5QIBAAKCAYEA0/cJ1ailSDLOKZmknwypPwxnZU6ZbUWNT6K5zR7i58GQX4v7
Jf9W6XFXAh1WASbN/cknI57ArVOi9Sqq5iegKgOt9Xuz6PO3JDgyBC5h7xnmdUGC
8sNONwV3bvjeh8BuUciG6NXICjspCusoPsSdH5uUb3ZNEw7txkmfc/9crpt4Kuw/
h2vteP4+4yqRvyapb0q9gkklM1AWMz8awHgKxh+RABW5+QAVeJyxAvFqgHU8vlHo
FqZUK3HCJr8BEj4yP/bMKQ5piCq3VVAhrz5gVkaZdAauUOgFJIZFz2Ib3Hw7QQGF
fds4qA5tUkpfJ4DTz2fdZGlK/FGOhoyYDYh/1sSx2R/rDhEzTUG+Ly/Hx0tVIUrC
vV1KGayUtKh82I1KJx3o5PiuatdCUWJJk5opB9nrrpWszRLpf6qzQ+jQX4qrC/rn
OAOxCvsFppOcD0tUU5aRXBstNqZDr0ORYJT3afin/mIV9ZfqPIutJMgRRyitWvBW
2e/dvllzdkciw+wdAgMBAAECggGBALQPDxOpB36laex80BMsNjmQ1+R/OGZSxw/3
M87Dhg1JqNMBnn9QwQvddAQf14dy51nHHrBrk1Gp0qxhanvI/Y2zQntech5ZqiHi
etqCbD8oyVSiKhL/RdoHksV6M88t8IkYx0HTGPNX/i2ARdfMPY00JIvq+t5NuD7R
G9r1+k15VENiHTV0wW4ezF09NEIlHZmk7AdZC3jjd3hhCuClGMHk3tArpKeZIqQq
t1PAsLAe51LlUTqsYKPnJv6zEr3khk920qEFPQhOJ2Pegy25OAJDFDfI9usyyNd8
+ZEzOc/x47Sf5HW64X840Sl6B18XfHHKcmerb0fC5T0RyrJodsPBQN2wTRZkOApj
jMl6rsL2ZR9/7suApsqQ/M+o0wLbXnAICeUDSevIyIfBlNH6Fd2lA8qy71OTU+J8
N+yK6l4bLEqS9o/5ujpVsGVZsvSCNDWPGwhSHgoc++ra0xHoNyWzKhyk72KBsLTJ
/K2HyAtKHfJzjCGt7qqUdAo4cRbBIQKBwQDrO9fJ/9Nz7Oh/h/yV6YJqGcPawJ45
6J9BshFtpOh3jP3t/Qbh3jxel/JWtQf/5q+gnCk5Gm3wcnyW95Uhfjf1qptojs+B
6nIWQ4Eo9kMO8C3Hjg9eEJamE561Q+bUn5CpL4rTyHSMJAn9tdmgdIYqqkGA8MfO
tsMZE379y6HANddlr8+UE7ON1dH+i2aA8vdVIplhfocWXx6NM/7YZGXllixrADBO
SeyDeA9S22uj0r56FPHwwJIxbY3cU5UY41cCgcEA5q1Vc+EEKCDlFPRd/SHtbj1I
/G1k3U7Wz7W0E9bTzq4JUH/fcYCc0LezqSqB5Vqq7WTKouAaWv1u3BboMgf1MyL4
BZXNtmBxnUftIwgAjj+Gv6Z9PjGm7eDPACO36nrCc8jyQiC+YW2i2/WV3FgbyKVW
5OKv6r2HA2A7xmB4aYX/84yaxatyMeqf3pPQZnsyeCpkZH9pMAJAk6ofqVs8gx+R
dvzAWqwwBIOgNqBctlg1TeLxJ7/wiY88rprhuterAoHAfEJJU/gif7smea+g2qPV
8ywXDGfxbQ+XiMGCeVPCRNyiimO9jfSyGRnUtr68Mg5uub2j/PkcGAayD6FLEr00
3D26CDpgHPV2PJJ+GOA1Ph/bXu2I6pyCp8n3bTMoLSr45JYrlIHwJtMmv/Geuyay
dLkpDu6h7W0XlBh7XZ+CmyEV3i5HzaW2lHbdjcOfawBtbHhxY1HOFCA86EQOG8q1
FNkGL6OS4Ngl6rCyD4e+zXQWkDCeORnQX/HKHRe9y77TAoHBAI8P7xBX/GD6zbhq
OaRVF1CoV4yeN9D9JUJauZ0YPfJPg11WgPh+wEK9u4Ht5/ObgKiMOxQ6kn3d8ZQz
7LuirdKSHGoyVwuF0tCtnAd+3gFaqrJPihs5ZgLfyuGWRWj5y4FNndmfPxNjkH6E
V/X4+vo8eKCpalrWQla4pznH+MXRIkUk7ZQsWT1V+uJqn/P+8fUOrYaacl0g+HXc
K5i4CdZ0jJ3T8WQKqBkUnRdj6zv3hs+QMOnrIz3dBNBxJA/h0wKBwQDbwJPBnYFG
ub0GbtxmVCpLb1GY67RnP/AxNjuHYo3K7JTMQ/7Q6RGTyMyzwVa4n3/hnfiX0dYl
E25aO/+KFfuA1Je1Qj80XH+ysFLY8pn0+dvQRCo9tIMiUKz/YEGHUVJEKSHcP/a+
ksHuTkn8SG3ejjc57dGJNA1zL1RVmqc5LFIosX2HMerdQ4zl/E4YzJ4GcLwfpc+D
8VG2buNXtyeMVz5jF/H+8/46DhnBpSIx/m7eqCmSTANBrBwbcrHh//o=
-----END RSA PRIVATE KEY-----

>>> id_rsa_cs338.pub <<<

ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQDT9wnVqKVIMs4pmaSfDKk/DGdlTpltRY1Po
rnNHuLnwZBfi/sl/1bpcVcCHVYBJs39yScjnsCtU6L1KqrmJ6AqA631e7Po87ckODIELmHvGe
Z1QYLyw043BXdu+N6HwG5RyIbo1cgKOykK6yg+xJ0fm5Rvdk0TDu3GSZ9z/1yum3gq7D+
Ha+14/j7jKpG/JqlvSr2CSSUzUBYzPxrAeArGH5EAFbn5ABV4nLEC8WqAdTy+UegWplQrcc
ImvwESPjI/9swpDmmIKrdVUCGvPmBWRpl0Bq5Q6AUkhkXPYhvcfDtBAYV92zioDm1SSl8
ngNPPZ91kaUr8UY6GjJgNiH/WxLHZH+sOETNNQb4vL8fHS1UhSsK9XUoZrJS0qHzYjUon
Hejk+K5q10JRYkmTmikH2euulazNEul/qrND6NBfiqsL+uc4A7EK+wWmk5wPS1RTlpFcGy0
2pkOvQ5FglPdp+Kf+YhX1l+o8i60kyBFHKK1a8FbZ792+WXN2RyLD7B0=
lane-maitland@Lanes-MacBook-Air.local

>>> private key <<<

The private key is a sequence type that should contain the following:

name	**type**	**details**
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	q^(-1) mod p
otherPrimeInfos	OtherPrimeInfos OPTIONAL	

- The version number is 0 or 1.
- If the version is 0, then there is no instance of `OtherPrimeInfos`.
- If the version is 1, then there is an instance of `OtherPrimeInfos`, and it contains at least one instance of `OtherPrimeInfo`.
    - `OtherPrimeInfo` should contain the following:

name	**type**	**details**
prime	INTEGER	prime factor r_i of n
exponent	INTEGER	d_i = d mod (r_i - 1)
coefficient	INTEGER	t_i = (r_1 * r_2 * ... * r_(i-1))^(-1) mod r_i

    - The instances of `OtherPrimeInfo` are listed in order of the primes (r_i).

I used {https://holtstrom.com/michael/tools/asn1decoder.php} to decode the private key.

name	**value**
version	0x00 (0 decimal)
modulus	0x00d3f709d5a8a54832ce2999a49f0ca93f0c67654e996d458d4fa2b9cd1ee2e7c1905f8bfb25ff56e97157021d560126cdfdc927239ec0ad53a2f52aaae627a02a03adf57bb3e8f3b7243832042e61ef19e6754182f2c34e3705776ef8de87c06e51c886e8d5c80a3b290aeb283ec49d1f9b946f764d130eedc6499f73ff5cae9b782aec3f876bed78fe3ee32a91bf26a96f4abd824925335016333f1ac0780ac61f910015b9f90015789cb102f16a80753cbe51e816a6542b71c226bf01123e323ff6cc290e69882ab7555021af3e605646997406ae50e805248645cf621bdc7c3b4101857ddb38a80e6d524a5f2780d3cf67dd64694afc518e868c980d887fd6c4b1d91feb0e11334d41be2f2fc7c74b55214ac2bd5d4a19ac94b4a87cd88d4a271de8e4f8ae6ad742516249939a2907d9ebae95accd12e97faab343e8d05f8aab0bfae73803b10afb05a6939c0f4b545396915c1b2d36a643af43916094f769f8a7fe6215f597ea3c8bad24c8114728ad5af056d9efddbe5973764722c3ec1d
publicExponent	0x010001 (65537 decimal)
privateExponent	0x00b40f0f13a9077ea569ec7cd0132c363990d7e47f386652c70ff733cec3860d49a8d3019e7f50c10bdd74041fd78772e759c71eb06b9351a9d2ac616a7bc8fd8db3427b5e721e59aa21e27ada826c3f28c954a22a12ff45da0792c57a33cf2df08918c741d318f357fe2d8045d7cc3d8d34248beafade4db83ed11bdaf5fa4d795443621d3574c16e1ecc5d3d3442251d99a4ec07590b78e37778610ae0a518c1e4ded02ba4a79922a42ab753c0b0b01ee752e5513aac60a3e726feb312bde4864f76d2a1053d084e2763de832db93802431437c8f6eb32c8d77cf9913339cff1e3b49fe475bae17f38d1297a075f177c71ca7267ab6f47c2e53d11cab26876c3c140ddb04d1664380a638cc97aaec2f6651f7feecb80a6ca90fccfa8d302db5e700809e50349ebc8c887c194d1fa15dda503cab2ef539353e27c37ec8aea5e1b2c4a92f68ff9ba3a55b06559b2f48234358f1b08521e0a1cfbeadad311e83725b32a1ca4ef6281b0b4c9fcad87c80b4a1df2738c21adeeaa94740a387116c121
prime1	0x00eb3bd7c9ffd373ece87f87fc95e9826a19c3dac09e39e89f41b2116da4e8778cfdedfd06e1de3c5e97f256b507ffe6afa09c29391a6df0727c96f795217e37f5aa9b688ecf81ea7216438128f6430ef02dc78e0f5e1096a6139eb543e6d49f90a92f8ad3c8748c2409fdb5d9a074862aaa4180f0c7ceb6c319137efdcba1c035d765afcf9413b38dd5d1fe8b6680f2f7552299617e87165f1e8d33fed86465e5962c6b00304e49ec83780f52db6ba3d2be7a14f1f0c092316d8ddc539518e357
prime2	0x00e6ad5573e1042820e514f45dfd21ed6e3d48fc6d64dd4ed6cfb5b413d6d3ceae09507fdf71809cd0b7b3a92a81e55aaaed64caa2e01a5afd6edc16e83207f53322f80595cdb660719d47ed2308008e3f86bfa67d3e31a6ede0cf0023b7ea7ac273c8f24220be616da2dbf595dc581bc8a556e4e2afeabd8703603bc660786985fff38c9ac5ab7231ea9fde93d0667b32782a64647f6930024093aa1fa95b3c831f9176fcc05aac300483a036a05cb658354de2f127bff0898f3cae9ae1bad7ab
exponent1	0x7c424953f8227fbb2679afa0daa3d5f32c170c67f16d0f9788c1827953c244dca28a63bd8df4b21919d4b6bebc320e6eb9bda3fcf91c1806b20fa14b12bd34dc3dba083a601cf5763c927e18e0353e1fdb5eed88ea9c82a7c9f76d33282d2af8e4962b9481f026d326bff19ebb26b274b9290eeea1ed6d1794187b5d9f829b2115de2e47cda5b69476dd8dc39f6b006d6c78716351ce14203ce8440e1bcab514d9062fa392e0d825eab0b20f87becd741690309e3919d05ff1ca1d17bdcbbed3
exponent2	0x008f0fef1057fc60facdb86a39a4551750a8578c9e37d0fd25425ab99d183df24f835d5680f87ec042bdbb81ede7f39b80a88c3b143a927dddf19433ecbba2add2921c6a32570b85d2d0ad9c077ede015aaab24f8a1b396602dfcae1964568f9cb814d9dd99f3f1363907e8457f5f8fafa3c78a0a96a5ad64256b8a739c7f8c5d1224524ed942c593d55fae26a9ff3fef1f50ead869a725d20f875dc2b98b809d6748c9dd3f1640aa819149d1763eb3bf786cf9030e9eb233ddd04d071240fe1d3
coefficient	0x00dbc093c19d8146b9bd066edc66542a4b6f5198ebb4673ff031363b87628dcaec94cc43fed0e91193c8ccb3c156b89f7fe19df897d1d625136e5a3bff8a15fb80d497b5423f345c7fb2b052d8f299f4f9dbd0442a3db4832250acff6041875152442921dc3ff6be92c1ee4e49fc486dde8e3739edd189340d732f54559aa7392c5228b17d8731eadd438ce5fc4e18cc9e0670bc1fa5cf83f151b66ee357b7278c573e6317f1fef3fe3a0e19c1a52231fe6edea829924c0341ac1c1b72b1e1fffa

There is no `otherPrimeInfos` because the version is 0.

I used {https://lapo.it/asn1js/} to find out which bytes from the decoded base64 data represent each integer.

name	**number of bits**	**offset**	**DER encoding**
version		4	02 01 00
modulus	3072	7	02 82 01 81 …
publicExponent		396	02 03 …
privateExponent	3072	401	02 82 01 81 …
prime1	1536	790	02 81 C1 …
prime2	1536	986	02 81 C1 …
exponent1	1535	1182	02 81 C0 …
exponent2	1536	1377	02 81 C1 …
coefficient	1536	1573	02 81 C1 …

The type SEQUENCE is also encoded (30 82 06 E5) and has offset 0.

The first part of the DER encoding provides information on the type and size of the data. Note that all integers of 1536 bits begin with the same sequence (02 81 C1), and all integers of 3072 bits begin with the same sequence (02 82 01 81). The integer of 1535 bits begins with a sequence similar to integers of 1536 bits, but not identical. The others are unique.

In “02 03”, the “02” represents the type (integer), and the “03” represents the length (3 bytes). {https://docs.microsoft.com/en-us/windows/win32/seccertenroll/about-integer}

>>> public key <<<
The public key is a sequence type that should contain the following:

name	**type**	**details**
“ssh-rsa”	string	
modulus	INTEGER	n
publicExponent	INTEGER	e

I converted the public key from OpenSSH to PEM by running this command in my terminal:
ssh-keygen -f id_rsa_cs338.pub -e -m pem

-----BEGIN RSA PUBLIC KEY-----
MIIBigKCAYEA0/cJ1ailSDLOKZmknwypPwxnZU6ZbUWNT6K5zR7i58GQX4v7Jf9W
6XFXAh1WASbN/cknI57ArVOi9Sqq5iegKgOt9Xuz6PO3JDgyBC5h7xnmdUGC8sNO
NwV3bvjeh8BuUciG6NXICjspCusoPsSdH5uUb3ZNEw7txkmfc/9crpt4Kuw/h2vt
eP4+4yqRvyapb0q9gkklM1AWMz8awHgKxh+RABW5+QAVeJyxAvFqgHU8vlHoFqZU
K3HCJr8BEj4yP/bMKQ5piCq3VVAhrz5gVkaZdAauUOgFJIZFz2Ib3Hw7QQGFfds4
qA5tUkpfJ4DTz2fdZGlK/FGOhoyYDYh/1sSx2R/rDhEzTUG+Ly/Hx0tVIUrCvV1K
GayUtKh82I1KJx3o5PiuatdCUWJJk5opB9nrrpWszRLpf6qzQ+jQX4qrC/rnOAOx
CvsFppOcD0tUU5aRXBstNqZDr0ORYJT3afin/mIV9ZfqPIutJMgRRyitWvBW2e/d
vllzdkciw+wdAgMBAAE=
-----END RSA PUBLIC KEY-----

I used {https://holtstrom.com/michael/tools/asn1decoder.php} to decode the private key.

name	**value**
modulus	0x00d3f709d5a8a54832ce2999a49f0ca93f0c67654e996d458d4fa2b9cd1ee2e7c1905f8bfb25ff56e 97157021d560126cdfdc927239ec0ad53a2f52aaae627a02a03adf57bb3e8f3b7243832042e61ef19e 6754182f2c34e3705776ef8de87c06e51c886e8d5c80a3b290aeb283ec49d1f9b946f764d130eedc64 99f73ff5cae9b782aec3f876bed78fe3ee32a91bf26a96f4abd824925335016333f1ac0780ac61f9100 15b9f90015789cb102f16a80753cbe51e816a6542b71c226bf01123e323ff6cc290e69882ab7555021 af3e605646997406ae50e805248645cf621bdc7c3b4101857ddb38a80e6d524a5f2780d3cf67dd646 94afc518e868c980d887fd6c4b1d91feb0e11334d41be2f2fc7c74b55214ac2bd5d4a19ac94b4a87cd 88d4a271de8e4f8ae6ad742516249939a2907d9ebae95accd12e97faab343e8d05f8aab0bfae73803b 10afb05a6939c0f4b545396915c1b2d36a643af43916094f769f8a7fe6215f597ea3c8bad24c811472 8ad5af056d9efddbe5973764722c3ec1d
publicExponent	0x010001 (65537(decimal)

I used {https://lapo.it/asn1js/} to find out which bytes from the decoded base64 data represent each integer.

name	**number of bits**	**offset**	**DER encoding**
modulus		4	02 82 01 81 …
publicExponent	3072	393	02 03 …

The type SEQUENCE is also encoded (30 82 01 8A) and has offset 0. The encoding of the sequence type differs from the private key, but the modulus and publicExponent are the same.

>>> sanity check <<<
Given:

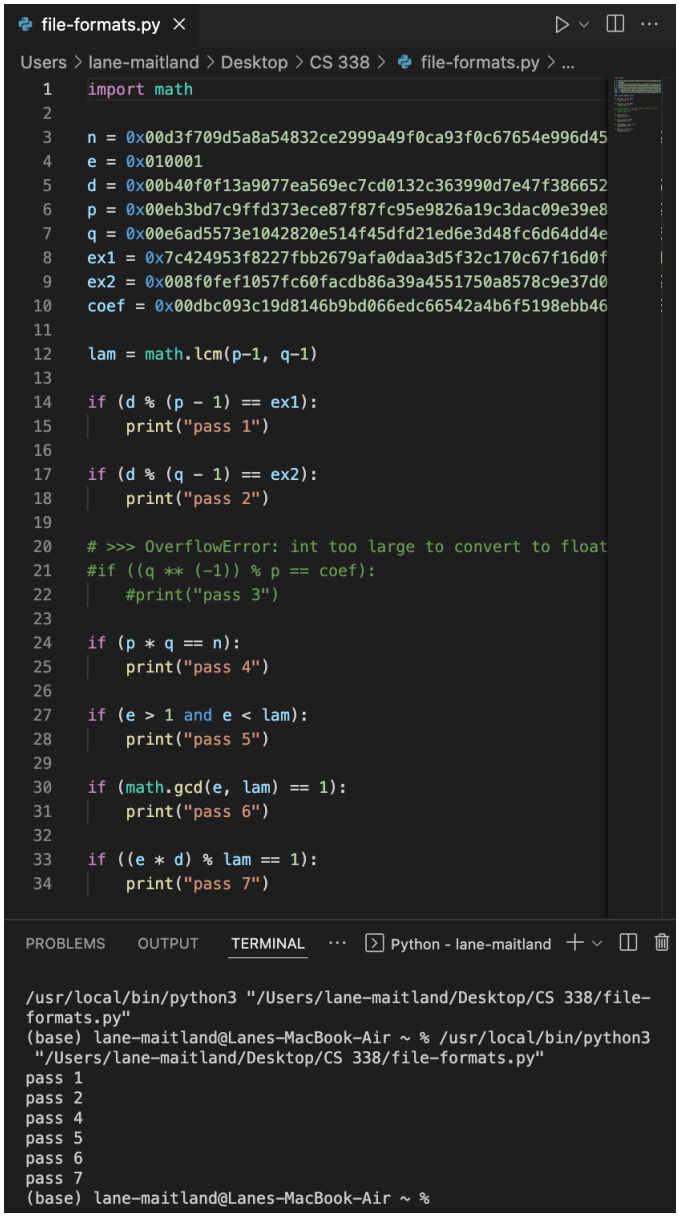
- n
- e
- d
- p
- q
- d mod (p-1)
- d mod (q-1)
- q^(-1) mod p

So:

- λ(n) = lcm (p-1, q-1) =

To confirm:

- d mod (p-1) = [exponent1]
- d mod (q-1) = [exponent2]
- q^(-1) mod p = [coefficient]
- n = p*q
- 1 < e < λ(n)
- gcd(e, λ(n)) = 1
- e*d mod λ(n) ≡ 1



Test 3 could not be confirmed, but all others passed.

Other sources:

- https://datatracker.ietf.org/doc/html/rfc8017#section-3
- https://datatracker.ietf.org/doc/html/rfc4253#section-6.6
- https://www.thedigitalcatonline.com/blog/2018/04/25/rsa-keys/
- https://blog.oddbit.com/post/2011-05-08-converting-openssh-public-keys/