Nicholas Vlahos-Sten

Jeff Ondich

Computer Security

30 September 2025

Key Formats

Contents of id_rsa_homework

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

MIIG4wIBAAKCAYEAyRghMGqdaD59jrglXcHY4POZITUQMUh0t/UkTDkk6Jwn98hc

mHHe0jPPunZiybBBYZjG1XrVzbbc/9RwnCQbgSgq5ndiwu97Y+nFIqblpkXdbbf+

PiF/juL8oa8wBMydvGN8ArK/awL91DqczH58gSSTBJprhg+P+cIoCeRjAP0vSb1M

qHcYtm8497r/JFtm9TtSoo4+ALzrehBB7oeprEpmcn2VzfwBHS6qx5SavbAku//T

BmPJ/BAxjG8n7W6HvLw9vmZhOyfS8eRNoWmiyX8PLMoVTG8nTqJFVt5G2dpBjXcZ

gzxUAg1QKoPaWw+nqaRzUJ3vpQLRq1ondapjRCjXowCQ9GI+dSgWOPZ7jJN5emtH

unyV8hylFWLlxA0xbr42vynCXBx38EEuDCcprxgyjuCv8mvdVdJmLQkuMR7f8Nuj

nOIISEMqLd81kAqSX5AzMMWLtILO7Vxt6rggyC3k55/KMqcpOvl1R4M+wN12tQwu

Mxe+395RilmlYOI1AgMBAAECggGAeLLq+MyFPhSD+C2+/89iBWMw/iWDWpFKvT9+

UeHarWNM+gRgz39JZpGClUCJetU36QRyE6Buw9KGQ2eXT6C8gYFCfqlZvTdQsF+X

23yhSnOUvE47Y79neLP1PnaQM9E8u95aiKQREIHzKHcltfgVcs4pEBI/h2+/0wEe

VxGl+un8gB4G7PbO9nE7mDW8/m1+FqNMrhUCUh8Sl2k6qZjz1l2xr/aA++nOIV3p

U3F2j3QyGEUsvCEZExoFI210YaZ88ORE8R5HsngGjztZJsuj9JIURvGsxX63ON3Z

MxpeWNF0GTycz6RqEGZfYVdQozy+NG08XVgX6O34xkgjfvLMVEnW/OC9aJu/dngt

kyECjDJ0mqhW0NR7Gq1TuxnjqvodnzXAsqxORIO0gmHBHUHThRBIIXTmSRmJ4nhf

VyfK0iTdejjZxk6/Pd1ZMZ7+O+ly6azwgTrWjpQESacEoZxAstVcKmM6g+066Lqn
4iTtroMxdob85sKxzr9m35fqdZYBAoHBAPh8DetheZfzbRRkmGYNpi+G3/NsGB9B
ejIFEDASYXPJDlok2GDcYmKhEpGw7iQ97uDty0M+D4hUnSyPZE/LQf94F1rc1xq4
H6ngMlTbmo4p5JxRZHlR6NJWPAK4H0QYGIwx9cYXIU4TLIhqk1pWaUjq37aXf9gm
a/SZaOdY+yGJH/34H1UDPWhxeYbccu4vcTE/Izbnx4wxyZJHSJwQYyp2/yv3va0O
nTGYOApT2bNO0W2dfOy7qNIIPknH5dJp8QKBwQDPLSU4XgiW3gQ/403sI21Rt5Ez
z0GyzgDiyi197cfOMm/j0IXiK+un9WEbTjKxSpsMN+1H6vT7r37eEE7/nY9JO2B9
7HiNif8FOw5anMtabfMfaNwP5HfQ18IZm2yoRQMPt7piUGxcRqvx+eGPfGcDEvH+
d+QnZIF+XSmHqMsgxKmD2Lah4tE/iREHsWKAsW8MedCFiG8t7DSyh3AemBs7y1An
bvRemfXYMeinyAxdeFy+CCfW/zMmCcpeaPf1WIUCgcEAp3ikwQGfPAQ6fkYWWKHW
Ya3bfmpaD6bHvrKe6Gda+viFDchLdWzTenno0/yVZCPBviNGnXlUaqPskr3J8+Wg
A5hi1UCGUWHLJWkPQxYHIs2lYJ2SxBmqeyjXpVWG7RP4Z3S8mte4ALD3kLGTwoz+
xMBYKu4MohezJtXGWDYme8G5qCf0pg1a8+uW5ofphhK/jVtqVnT8jsZ3fnlAvL5m
t5YgAPgmBF8XHvfivUMqNKloiSJdsggAjh40LPHnOVthAoHAe9CwQNq56WgwQ/pV
tH2LtN2vMrZ3viOtldxcibkUw1+TTSRvDzdOdCeK7GW0mA5NUWUPMv7xgIVmndJD
VDLW0l5hEYDEn1pMhdTAhFjWk2nIdgoWEo1pKe/4OZOQwZPU+4zDIESOLiuyUUAU
lMs1SToGVB1UxlIjreCBlGjFVQUThg97Po6unXkOK3nQs7bW9i7K9zMNtvVGnTei
QzaGeBrPAXInK2555cYWFEfTU/099/gpzyzfZalOP1QedIt9AoHAHLsgxlUv85+q
jXoAR9fz8J/+HAUx/Pp/exR29WSSDipLQGd95CYXS4J/FiZhQg8paqWAAQOnp/Ru
0fuvTlmfnk+AWoJV6JAGo2nXh3ZqdZrRRZ61QcpQtum9H+ZVmUD2Lts/gL5T/yT5
8wSPJ+ciztWYFMUKEQM56FPUJZ5qkofb6YHm16WrZAeqLS+cE1fwdLMnm8ppZqxt
He96HnkDoEd6dsU/teNb3zJQflj4xM8XTtkM4+LLXN5lnmi1HO8n
-----END RSA PRIVATE KEY-----

Contents of id_rsa_homework.pub

ssh-rsa

AAAAB3NzaC1yc2EAAAADAQABAAABgQDJGCEwap1oPn2OuCVdwdjg85khNRAxSH

S39SRMOSTonCf3yFyYcd7SM8+6dmLJsEFhmMbVetXNttz/1HCcJBuBKCrmd2LC73tj6

cUipuWmRd1tt/4+lX+O4vyhrzAEzJ28Y3wCsr9rAv3UOpzMfnyBJJMEmmuGD4/5wigJ5

GMA/S9JvUyodxi2bzj3uv8kW2b1O1Kijj4AvOt6EEHuh6msSmZyfZXN/AEdLqrHlJq9sCS

7/9MGY8n8EDGMbyftboe8vD2+ZmE7J9Lx5E2haaLJfw8syhVMbydOokVW3kbZ2kGNd

xmDPFQCDVAqg9pbD6eppHNQne+lAtGrWid1qmNEKNejAJD0Yj51KBY49nuMk3l6a0e

6fJXyHKUVYuXEDTFuvja/KcJcHHfwQS4MJymvGDKO4K/ya91V0mYtCS4xHt/w26Oc4

ghIQyot3zWQCpJfkDMwxYu0gs7tXG3quCDILeTnn8oypyk6+XVHgz7A3Xa1DC4zF77f3

lGKWaVg4jU= pegui@Nicholas_PC

PRIVATE KEY

I expect to see what the A.1.2 appendix says I will see:

- The version (not important)
- The number n Bob does mod n of to find the shared secret
- The unique number e in my private key
- The number d I share with Bob in my public key
- The two prime numbers p and q whose multiplication created n
- Numbers representing d mod (p-1) and d mod (q-1) (is this for figuring out least common multiples?)

- Nevermind upon looking it up, these two numbers and the next item, the "CRT coefficient" are numbers you need to use the "Chinese Remainder Theorem" to more efficiently and optimally decrypt the key. So I don't recognize them because we didn't use them in our last assignment.

To decode, I don't know man, I copied and pasted the entire private key into the Lapo Luchini ASN.1 decoder and pressed "decode" and it gave me a list of what all the important integers were.

Here's all the info about the integers in the file. They are in the same order as I said they would be in in the previous question with bullet points:

- The first integer, which is 0 (alternatively 0x 0 :), is the **version**.
- The second integer is the "**modulus**," also known as the integer $\mathbf{n_A}$ from our in class exercise. It is the second integer in my private key tuple. I expect to see this in the public key file too.
    - The value is 0x 00 C9 18 21 30 6A 9D 68 3E 7D 8E B8 25 5D C1 D8 E0 F3 99 21 35 10 31 48 74 B7 F5 24 4C 39 24 E8 9C 27 F7 C8 5C 98 71 DE D2 33 CF BA 76 62 C9 B0 41 61 98 C6 D5 7A D5 CD B6 DC FF D4 70 9C 24 1B 81 28 2A E6 77 62 C2 EF 7B 63 E9 C5 22 A6 E5 A6 45 DD 6D B7 FE 3E 21 7F 8E E2 FC A1 AF 30 04 CC 9D BC 63 7C 02 B2 BF 6B 02 FD D4 3A 9C CC 7E 7C 81 24 93 04 9A 6B 86 0F 8F F9 C2 28 09 E4 63 00 FD 2F 49 BD 4C A8 77 18 B6 6F 38 F7 BA FF 24 5B 66 F5 3B 52 A2 8E 3E 00 BC EB 7A 10 41 EE 87 A9 AC 4A 66 72 7D 95 CD FC 01 1D 2E AA C7 94 9A BD B0 24 BB FF D3 06 63 C9 FC 10 31 8C 6F 27 ED 6E 87 BC BC 3D BE 66 61 3B 27 D2 F1 E4 4D A1 69 A2 C9 7F 0F 2C CA 15 4C 6F 27 4E A2 45 56 DE 46 D9 DA 41 8D 77 19 83 3C 54 02 0D 50 2A 83 DA 5B 0F A7 A9 A4 73 50 9D EF A5 02 D1 AB 5A 27 75 AA 63 44 28 D7 A3 00 90 F4 62 3E 75 28 16 38 F6 7B 8C 93 79 7A 6B 47 BA 7C 95 F2 1C A5 15 62 E5 C4 0D 31 6E BE 36 BF 29 C2 5C 1C 77 F0 41 2E 0C 27 29 AF 18 32 8E E0 AF F2 6B DD 55 D2 66 2D 09 2E 31 1E DF F0 DB A3 9C E2 08 48 43 2A 2D DF 35 90 0A 92 5F 90 33 30 C5 8B B4 82 CE ED 5C 6D EA B8 20 C8 2D E4 E7 9F CA 32 A7 29 3A F9 75 47 83 3E C0 DD 76 B5 0C 2E 33 17 BE DF DE 51 8A 59 A5 60 E2 35
- The third integer is the **publicExponent,** the first item of the public key tuple I give to Bob, also known as the integer $\mathbf{e_A}$from our in class exercise. I expect to see this in the public key file too.
    - The value is 0x 01 00 01
- The fourth integer is the **privateExponent**, the first item of the private key tuple I keep for myself, also known as the integer $\mathbf{d_A}$ which I don't expect to see in the public key file.

    - The value is 0x 78 B2 EA F8 CC 85 3E 14 83 F8 2D BE FF CF 62 05 63 30 FE 25 83 5A 91 4A BD 3F 7E 51 E1 DA AD 63 4C FA 04 60 CF 7F 49 66 91 82 95 40 89 7A D5 37 E9 04 72 13 A0 6E C3 D2 86 43 67 97 4F A0 BC 81 81 42 7E A2 19 BD 37 50 B0 5F 97 DB 7C A1 4A 73 94 BC 4E 3B 63 BF 67 78 B3 F5 3E 76 90 33 D1 3C BB DE 5A 88 A4 11 10 81 F3 28 77 25 B5 F8 15 72 CE 29 10 12 3F 87 6F BF D3 01 1E 57 11 A5 FA E9 FC 80 1E 06 EC F6 CE F6 71 3B 98 35 BC FE 6D 7E 16 A3 4C AE 15 02 52 1F 12 97 69 3A A9 98 F3 D6 5D B1 AF F6 80 FB E9 CE 21 5D E9 53 71 76 8F 74 32 18 45 2C BC 21 19 13 1A 05 23 6D 74 61 A6 7C F0 E4 44 F1 1E 47 B2 78 06 8F 3B 59 26 CB A3 F4 92 14 46 F1 AC C5 7E B7 38 DD D9 33 1A 5E 58 D1 74 19 3C 9C CF A4 6A 10 66 5F 61 57 50 A3 3C BE 34 6D 3C 5D 58 17 E8 ED F8 C6 48 23 7E F2 CC 54 49 D6 FC E0 BD 68 9B BF 76 78 2D 93 21 02 8C 32 74 9A A8 56 D0 D4 7B 1A AD 53 BB 19 E3 AA FA 1D 9F 35 C0 B2 AC 4E 44 83 B4 82 61 C1 1D 41 D3 85 10 48 21 74 E6 49 19 89 E2 78 5F 57 27 CA D2 24 DD 7A 38 D9 C6 4E BF 3D DD 59 31 9E FE 3B E9 72 E9 AC F0 81 3A D6 8E 94 04 49 A7 04 A1 9C 40 B2 D5 5C 2A 63 3A 83 ED 3A E8 BA A7 E2 24 ED AE 83 31 76 86 FC E6 C2 B1 CE BF 66 DF 97 EA 75 96 01
- So I don't have to keep writing it, I don't expect to see any of the future integers in the public key file
- The next integer is **prime1,** which is the first prime I used to compute the **modus.** In our in-class exercise this integer was called
    - The value is 0x 00 F8 7C 0D EB 61 79 97 F3 6D 14 64 98 66 0D A6 2F 86 DF F3 6C 18 1F 41 7A 32 05 10 30 12 61 73 C9 0E 5A 24 D8 60 DC 62 62 A1 12 91 B0 EE 24 3D EE E0 ED CB 43 3E 0F 88 54 9D 2C 8F 64 4F CB 41 FF 78 17 5A DC D7 1A B8 1F A9 E0 32 54 DB 9A 8E 29 E4 9C 51 64 79 51 E8 D2 56 3C 02 B8 1F 44 18 18 8C 31 F5 C6 17 21 4E 13 2C 88 6A 93 5A 56 69 48 EA DF B6 97 7F D8 26 6B F4 99 68 E7 58 FB 21 89 1F FD F8 1F 55 03 3D 68 71 79 86 DC 72 EE 2F 71 31 3F 23 36 E7 C7 8C 31 C9 92 47 48 9C 10 63 2A 76 FF 2B F7 BD AD 0E 9D 31 98 38 0A 53 D9 B3 4E D1 6D 9D 7C EC BB A8 D2 08 3E 49 C7 E5 D2 69 F1
- The next integer is **prime2,** which is the second prime I used to compute the **modus.** In our in-class exercise this integer was called $\mathbf{q_A}$
    - The value is 0x 00 CF 2D 25 38 5E 08 96 DE 04 3F E3 4D EC 23 6D 51 B7 91 33 CF 41 B2 CE 00 E2 CA 2D 7D ED C7 CE 32 6F E3 D0 85 E2 2B EB A7 F5 61 1B 4E 32 B1 4A 9B 0C 37 ED 47 EA F4 FB AF 7E DE 10 4E FF 9D 8F 49 3B 60 7D EC 78 8D 89 FF 05 3B 0E 5A 9C CB 5A 6D F3 1F 68 DC 0F E4 77 D0 D7 C2 19 9B 6C A8 45 03 0F B7 BA 62 50 6C 5C 46 AB F1 F9 E1 8F 7C 67 03 12 F1 FE 77 E4 27 64 81 7E 5D 29 87 A8 CB 20 C4 A9 83 D8 B6 A1 E2 D1 3F 89 11 07 B1 62 80 B1 6F 0C 79 D0 85 88 6F 2D EC 34 B2 87 70 1E 98 1B 3B CB 50 27 6E F4 5E 99 F5 D8 31 E8 A7 C8 0C 5D 78 5C BE 08 27 D6 FF 33 26 09 CA 5E 68 F7 F5 58 85
- The next integer is **exponent1** used for the Chinese Remainder Theorem
    - The value is 0x 00 A7 78 A4 C1 01 9F 3C 04 3A 7E 46 16 58 A1 D6 61 AD DB 7E 6A 5A 0F A6 C7 BE B2 9E E8 67 5A FA F8 85 0D C8 4B 75 6C D3 7A 79 E8

D3 FC 95 64 23 C1 BE 23 46 9D 79 54 6A A3 EC 92 BD C9 F3 E5 A0 03 98 62 D5 40 86 51 61 CB 25 69 0F 43 16 07 22 CD A5 60 9D 92 C4 19 AA 7B 28 D7 A5 55 86 ED 13 F8 67 74 BC 9A D7 B8 00 B0 F7 90 B1 93 C2 8C FE C4 C0 58 2A EE 0C A2 17 B3 26 D5 C6 58 36 26 7B C1 B9 A8 27 F4 A6 0D 5A F3 EB 96 E6 87 E9 86 12 BF 8D 5B 6A 56 74 FC 8E C6 77 7E 79 40 BC BE 66 B7 96 20 00 F8 26 04 5F 17 1E F7 E2 BD 43 2A 34 A9 68 89 22 5D B2 08 00 8E 1E 34 2C F1 E7 39 5B 61

- The next integer is **exponent2** used for the Chinese Remainder Theorem
    - The value is 0x 7B D0 B0 40 DA B9 E9 68 30 43 FA 55 B4 7D 8B B4 DD AF 32 B6 77 BE 23 AD 95 DC 5C 89 B9 14 C3 5F 93 4D 24 6F 0F 37 4E 74 27 8A EC 65 B4 98 0E 4D 51 65 0F 32 FE F1 80 85 66 9D D2 43 54 32 D6 D2 5E 61 11 80 C4 9F 5A 4C 85 D4 C0 84 58 D6 93 69 C8 76 0A 16 12 8D 69 29 EF F8 39 93 90 C1 93 D4 FB 8C C3 20 44 8E 2E 2B B2 51 40 14 94 CB 35 49 3A 06 54 1D 54 C6 52 23 AD E0 81 94 68 C5 55 05 13 86 0F 7B 3E 8E AE 9D 79 0E 2B 79 D0 B3 B6 D6 F6 2E CA F7 33 0D B6 F5 46 9D 37 A2 43 36 86 78 1A CF 01 72 27 2B 6E 79 E5 C6 16 14 47 D3 53 FD 3D F7 F8 29 CF 2C DF 65 A9 4E 3F 54 1E 74 8B 7D
- The final integer is the **coefficient** also used for the Chinese Remainder Theorem
    - The value is 0x 1C BB 20 C6 55 2F F3 9F AA 8D 7A 00 47 D7 F3 F0 9F FE 1C 05 31 FC FA 7F 7B 14 76 F5 64 92 0E 2A 4B 40 67 7D E4 26 17 4B 82 7F 16 26 61 42 0F 29 6A A5 80 01 03 A7 A7 F4 6E D1 FB AF 4C 89 9F 9E 4F 80 5A 82 55 E8 90 06 A3 69 D7 87 76 6A 75 9A D1 45 9E B5 41 CA 50 B6 E9 BD 1F E6 55 99 40 F6 2E DB 3F 80 BE 53 FF 24 F9 F3 04 8F 27 E7 22 CE D5 98 14 C5 0A 11 03 39 E8 53 D4 25 9E 6A 92 87 DB E9 81 E6 D7 A5 AB 64 07 AA 2D 2F 9C 13 57 F0 74 B3 27 9B CA 69 66 AC 6D 1D EF 7A 1E 79 03 A0 47 7A 76 C5 3F B5 E3 5B DF 32 50 7C 88 F8 C4 CF 17 4E D9 0C E3 E2 CB 5C DE 65 9E 68 B5 1C EF 27

PUBLIC KEY

I mean, I won't lie, I just used the same RFC 8017 appendix manual to figure out what to expect in the public key file. There's a section about the public key file's syntax right above the section about the accurate private key file's syntax. In this appendix, it says there's only two important fields in the file, which are the **publicExponent** and the **modulus**, which, as you expect are respectively the items in the ($\mathbf{e_A}$, $\mathbf{n_A}$) pair from the in-class exercise. This info is also in the private key, so as you'd expect if you have the private key file the public key file is redundant.

To decode it, I did your Choice #1 and made the file into a PKCS#1 PEM file, and gave it to the Michael Holtstrom decoder, which gave me this:

Output

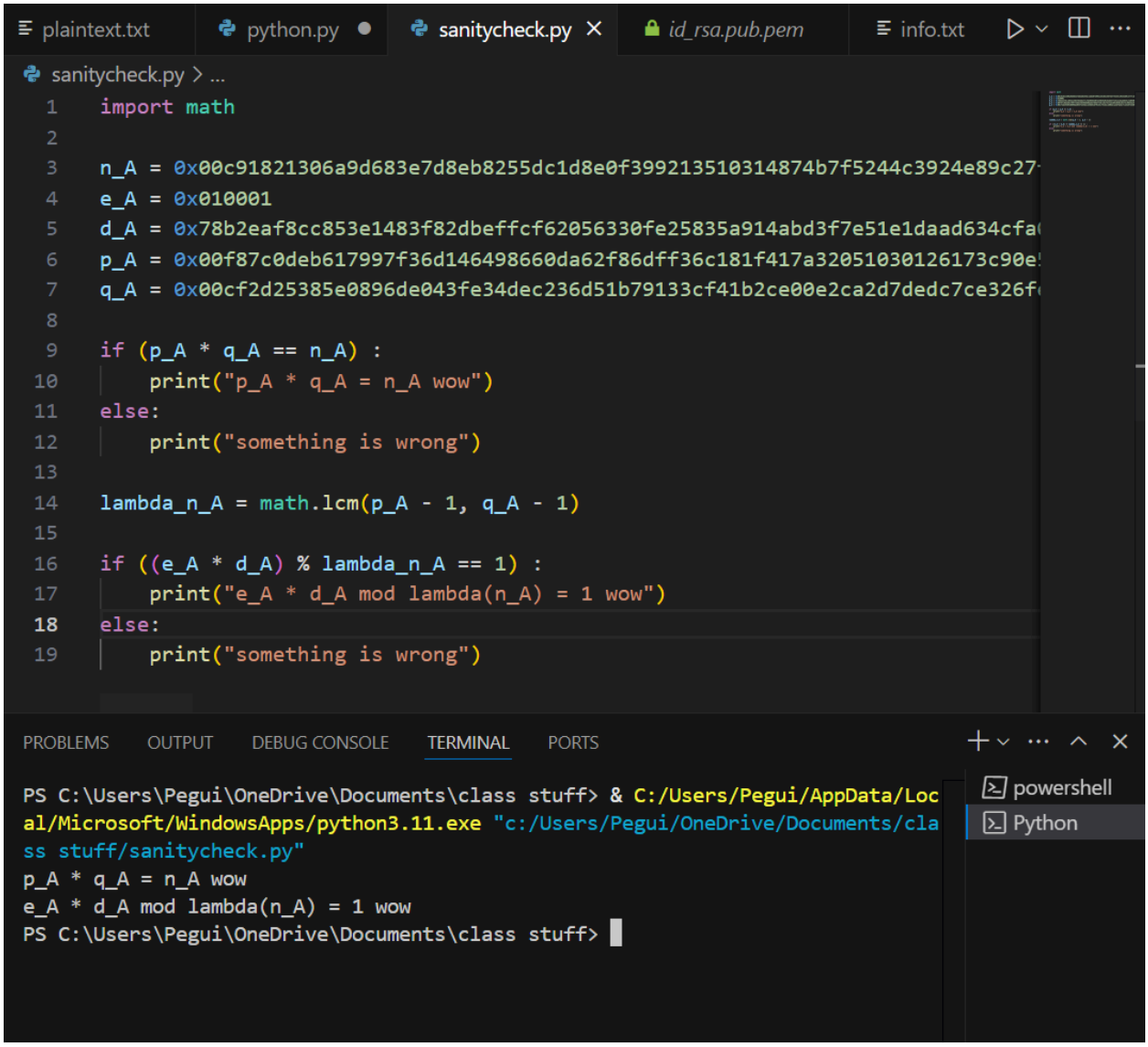
```
U.P.SEQUENCE {
   U.P.INTEGER
0x00c91821306a9d683e7d8eb8255dc1d8e0f399213510314874b7f5244c3924e89c27f7c85c9871ded233cfba7662c9b0416198c6d57ad5cdb6dcffd4709c241b
81282ae67762c2ef7b63e9c522a6e5a645dd6db7fe3e217f8ee2fca1af3004cc9dbc637c02b2bf6b02fdd43a9ccc7e7c812493049a6b860f8ff9c22809e46300fd
2f49bd4ca87718b66f38f7baff245b66f53b52a28e3e00bceb7a1041ee87a9ac4a66727d95cdfc011d2eaac7949abdb024bbffd30663c9fc10318c6f27ed6e87bc
bc3dbe66613b27d2f1e44da169a2c97f0f2cca154c6f274ea24556de46d9da418d7719833c54020d502a83da5b0fa7a9a473509defa502d1ab5a2775aa634428d7
a30090f4623e75281638f67b8c93797a6b47ba7c95f21ca51562e5c40d316ebe36bf29c25c1c77f0412e0c2729af18328ee0aff26bdd55d2662d092e311edff0db
a39ce20848432a2ddf35900a925f903330c58bb482ceed5c6deab820c82de4e79fca32a7293af97547833ec0dd76b50c2e3317bedfde518a59a560e235
   U.P.INTEGER 0x010001 (65537 decimal)
}
```

As you can see, the file consists of two integers, which perfectly fits with the syntax specifications:

- The first integer is the **modulus** the same one we saw in the private key
    - The value is
- The second integer is the **publicExponent**, the same one we saw in the private key
    - The value is 0x010001(also thank God for this, I may not have realized with the first decoder the first several bytes in each hexadecimal number were just specifying length and weren't part of the integer)

SANITY CHECK

Here I did the sanity check, see the next page for a picture of my python code and results. I tried two equations that should be true, and I think that is sufficient because every variable is used at some point within those two equations.


plaintext.txt
python.py
sanitycheck.py
id_rsa.pub.pem
info.txt
sanitycheck.py > ...
1 import math
2
3 n_A = 0x00c91821306a9d683e7d8eb8255dc1d8e0f399213510314874b7f5244c3924e89c27
4 e_A = 0x010001
5 d_A = 0x78b2eaf8cc853e1483f82dbeffcf62056330fe25835a914abd3f7e51e1daad634cfa
6 p_A = 0x00f87c0deb617997f36d146498660da62f86dff36c181f417a32051030126173c90e
7 q_A = 0x00cf2d25385e0896de043fe34dec236d51b79133cf41b2ce00e2ca2d7dedc7ce326f
8
9 if (p_A * q_A == n_A) :
10 print("p_A * q_A = n_A wow")
11 else:
12 print("something is wrong")
13
14 lambda_n_A = math.lcm(p_A - 1, q_A - 1)
15
16 if ((e_A * d_A) % lambda_n_A == 1) :
17 print("e_A * d_A mod lambda(n_A) = 1 wow")
18 else:
19 print("something is wrong")
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
powershell
Python
PS C:\Users\Pegui\OneDrive\Documents\class stuff> & C:/Users/Pegui/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/Pegui/OneDrive/Documents/class stuff/sanitycheck.py"
p_A * q_A = n_A wow
e_A * d_A mod lambda(n_A) = 1 wow
PS C:\Users\Pegui\OneDrive\Documents\class stuff>