SSH KEY FILE FORMATS

Zitian Yue

April 19, 2022

1. The content of id-rsa-homework is displayed in text as below:

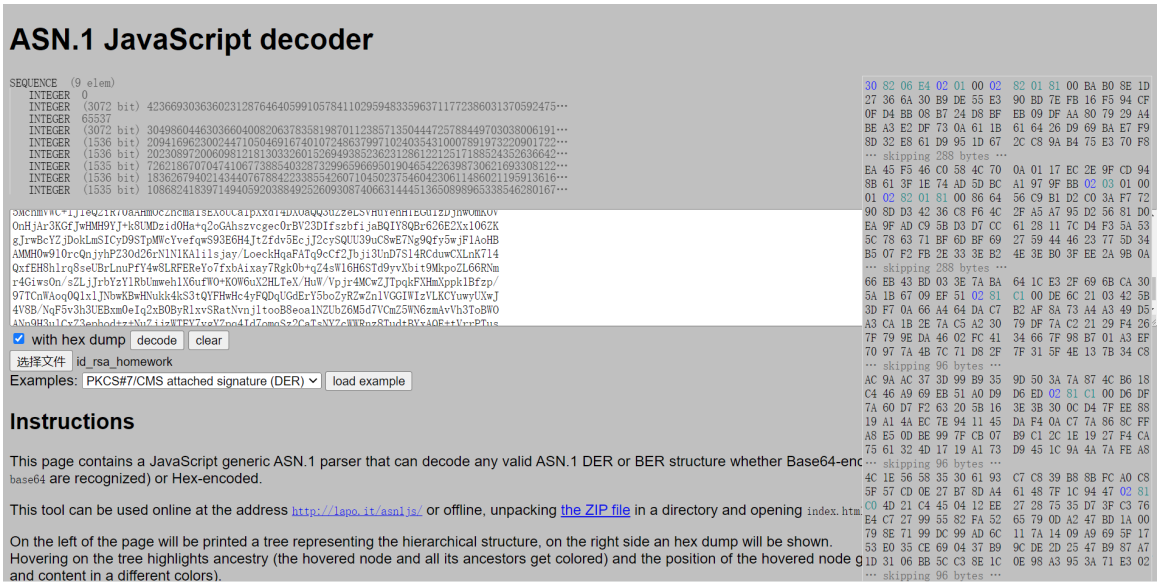
```
-----BEGIN RSA PRIVATE KEY-----
MIIG5AIBAAKCAYEAurCOHSc2ajC53lXjkL1++xb1lM8P1LsItyTYv+sJ36qAeSmk
vqPi33MKYRthZCbZabrn+Y0y6GHZlR1nLMiatHXjcPgfro3bLFG5ZapTrfEDH8CM
OWOTABit1aQndH8yCMJPdN1xtD1u7T/yKV1Kh/ymLMiaY5YGT5CkXQfDtiim5IgC
umgw/GEaGdCJgj+LsPZyz7iKsD6IdqJs17TcO48WvGhm6NI5cbTvcNuoIR6lPRtl
Mtlwd8rGRu78YRZBP5ocYprb31ItXqdtAcuX19VioqUlXFRDGSsLV6vIpvx0xvN9
wix7HAifnIOFpACb7Uk+kZgHOiQ9jyWC8/MF3yofIXwWGLShe1kw8MaOxULQBYPH
hBvcEnaKy7iIK6NRJqw9+f8bXS4jN4jVg7e+Cq9FB9Bh0liWOzjsXfzUDFXb489M
LhWYatDkfc9wXDmIBDisSDwaxU9/411Z774eEihtBfbqRfVGwFhMcAoBF+wun82U
i2E/HnStXbyhl5+7AgMBAAECggGBAIZkVsmx0sA693KQjdNCNsj2TC+lp5XSVoHQ
6p+tyVvT18xhKBF81PNaU1x4Y3G/bb9pJ1lERiN3XTS1B/L7LjM+sk4+sD/uKpsK
8dW3+ak72K3g5JDjl4z4NxuWojiA4Dnh92T/jLzdhJ8R63fPirm+LLoMZufNthlM
3KWD02WseggFyUD7hFgKZsmyj2rN5/eqUhgqltwtB/uvDX9y2R+e8BUNaxerTmMG
h1/NIIlHEFifAgjK8uWpeXbSz6fsZi1gOgf772no0r859vrJ/JnWetNomHAH/u5Q
0Qk9yGrnUu/mVa3hNUoK0CCoubGc29Pue6RhKVOxAeQhiI7U1lsTETH4IxkR56TP
xhsuZSUx4PndCSUukZNd4thZD7nczdtzj0gDdd7FR+OI5Catu0Pj5QPX+sDDluBB
M+nTaSefJHLJP3L2pQY12bFdNijhwW0osdJE4vBoGEQipprrL0EDzZR4kB7JGcD9
ZutDvQM+erpkHOMvaWvKMFobZwnvUQKBwQDebCEDQls99wpmpGTax7KvinOko0nV
o8obLnrFojB533rCISn0Jn95ntpGAvxBNGZ/mLcBo+9wl3pLfHHYL38xX04TezTI
RZ7MYOSBKslME2JBC5GlwL384UzjT0zeTPWpPb6rgwCT7+UieApMfxxTp5ktIMsP
Cp/E15KMKem3x4CUhWeY9P8ki7o3JoyrkmHNHd27DXce5JEsSQz9SYYsYaVwdeSI
rJqsNz2ZuTWdUDp6h0y2GMRGqWnrUaDZ1u0CgcEA1t96YNfyYyBbFj47MAzUf+6I
GaFK7H6UEUXa9ArHeoaM/6jlDb6Zf8sHucEsHhkn9Mp1YTJNFxmhc9lFHJpKev6o
WiX33rn0Puj1gIvqAVMUD9yn5RB4ROGTyFTfNqcXPLMUypyhWF7nrlTaBVf8I+Kg
cx1siubz3pi4MAF022hnsV1OiNS3zw+3ju69IWwAW1GjqDkKTs+CjAAtDctIRYXC
TB5WWDUwYZPHyDm4i/ygyF9XzQ4nt42kYUh/HJRHAoHATSHERQQS7icodTXXP8N2
5McnmVWC+lJleQ2iR70aAHmOcZncma1sEXoUCalpXxdT4DXOaQQ3uZzeLSVHuYen
HTEGu1zDjhwOmKOVOnHjAr3KGfJwHMH9YJ+k8UMDzid0Ha+q2oGAhszvcgec0rBV
23DIfszbfijaBQIY8QBr626E2Xx106ZKgJrwBcYZjDokLmSICyD9STpMWcYvefqw
S93E6H4JtZfdv5EcjJ2cySQUU39uC8wE7Ng9Qfy5wjF1AoHBAMMH0w9l0rcQnjyh
PZ3Od26rNlN1KAlilsjay/LoeckHqaFATq9cCf2Jbji3UnD7Sl4RCduwCXLnK7l4
QxfEH8hlrq8seUBrLnuPfY4w8LRFEReYo7fxbAixay7Rgk0b+qZ4sW16H6STd9yv
Xbit9MkpoZL66RNmr4GiwsOn/sZLjJrbYzYlRbUmweh1X6ufWO+KOW6uX2HLTeX/
HuW/Vpjr4MCwZJTpqkFXHmXppk1Bfzp/97TCnWAoq0Q1xlJNbwKBwHNukk4kS3tQ
YFHwHc4yFQDqUGdErY5boZyRZwZnlVGGIWIzVLKCYuwyUXwJ4V8B/NqF5v3h3UEB
xm0eIq2xB0ByRlxvSRatNvnjltooB8eoa1NZUbZ6M5d7VCmZ5WN6zmAvVh3ToBWO
ANn9H3ulCxZ3ephod+z+NuZjjzWTFY7vgYZpq4Id7omqSz2CaTsNYZcWWRnz8Tud
tBYxAQE+tVrrPTus/u0QS+FfB8ywkV2EgMuafoisDc+KCUlJ+EvJ7w==
-----END RSA PRIVATE KEY-----
```

The content of id-rsa-homework.pub is:
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQC6sI4dJzZqMLneVeOQvX77FvWUzw/Uuwi
3JNi/6wnfqoB5KaS+o+LfcwphG2FkJtlpuuf5jTLoYdmVHWcsyJq0deNw+B+ujdssUbllqlOt8QMf
wIw5Y5MAGK3VpCd0fzIIwk903XG0PW7tP/IpXUqH/KYsyJpjlgZPkKRdB8O2KKbkiAK6aDD8
YRoZ0ImCP4uw9nLPuIqwPoh2omzXtNw7jxa8aGbo0jlxtO9w26ghHqU9G2Uy2XB3ysZG7vxhFkE
/mhximtvfUi1ep20By5fX1WKipSVcVEMZKwtXq8im/HTG833CLHscCJ+cg4WkAJvtST6RmAc6
JD2PJYLz8wXfKh8hfBYYtKF7WTDwxo7FQtAFg8eEG9wSdorLuIgro1EmrD35/xtdLiM3iNWDt7
4Kr0UH0GHSWJY7OOxd/NQMVdvjz0wuFZhq0OR9z3BcOYgEOKxIPBrFT3/jXVnvvh4SKG0F9
upF9UbAWExwCgEX7C6fzZSLYT8edK1dvKGXn7s= james@LAPTOP-NID7KFEL

Private Key

In my private key file, I expect to see this list of terms: version; modulus; publicExponent; privateExponent; prime1; prime2; exponent1; exponent2; coefficient; otherPrimeInfos.

I checked all three decoders and decided to use the Lapo Luchini's ASN.1 decoder to decode my private key file. This decoder is able to read files directly and gives corresponding information. I submit my id-rsa-homework file to the website and contents of the file are displayed automatically. It should be noted that a big chunk of hexadecimal digits is provided by the decoder to represent the original text. The result is displayed as below:



The first integer corresponds to the version number. Its decimal value is 0. According to the decoded base64 data displayed, this integer starts at the offset of 4 and has length of 2+1. The first tag 02 indicates the type of Integer. The second tag 01 indicates the number of bytes of data needed for reading. The third tag 00 represents integer 0 as the value.

The second integer corresponds to the modulus. Its decimal value is:42366930363602312876464059910578411029594833596371177238603137059247584411029594833596371177238603137059247584... According to the decoded base64 data displayed, this integer starts at the offset of 7 and has length of 4+385. The first tag 02 indicates the type of Integer. The second tag 82 01 81 indicates the number of bytes of data needed for reading. The third tag 00 BA B0 8E 1D 27 36 6A 30 B9 DE 55 E3 90 BD 7E FB 16 F5 94 CF 0F D4 BB 08 B7 24 D8 BF EB 09 DF AA 80 79 29 A4 BE A3 E2 DF 73 0A 61 1B 61 64 26 D9 69 BA E7 F9 8D 32 E8 61 D9 95 1D 67 2C C8 9A B4 75 E3 70 F8 ... skipping 288 bytes ... EA 45 F5 46 C0 58 4C 70 0A 01 17 EC 2E 9F CD 94 8B 61 3F 1E 74 AD 5D BC A1 97 9F BB represents the value of this integer.

The third integer corresponds to the publicExponent. Its decimal value is 65537. According to the decoded base64 data displayed, this integer starts at the offset of 396 and has length of 2+3. The first tag 02 indicates the type of Integer. The second tag 03 indicates the number of bytes of data needed for reading. The third tag 01 00 01 represents integer 65537 as the value.

The fourth integer corresponds to the privateExponent. Its decimal value is: 3049860446303660400820637835819870112385713504447257884497030380061910... According to the decoded base64 data displayed, this integer starts at the offset of 401 and has length of 4+385. The first tag 02 indicates the type of Integer. The second tag 82 01 81 indicates the number of bytes of data needed for reading. The third tag 00 86 64 56 C9 B1 D2 C0 3A F7 72 90 8D D3 42 36 C8 F6 4C 2F A5 A7 95 D2 56 81 D0 EA 9F AD C9 5B D3 D7 CC 61 28 11 7C D4 F3 5A 53 5C 78 63 71 BF 6D BF 69 27 59 44 46 23 77 5D 34 B5 07 F2 FB 2E 33 3E B2 4E 3E B0 3F EE 2A 9B 0A ... skipping 288 bytes ... 66 EB 43 BD 03 3E 7A BA 64 1C E3 2F 69 6B CA 30 5A 1B 67 09 EF 51 represents the value of this integer.

The fifth integer corresponds to the prime1. Its decimal value is: 209416962300244710504691674010724863799710240354310007891973220901722... According to the decoded base64 data displayed, this integer starts at the offset of 790 and has length of 3+193. The first tag 02 indicates the type of Integer. The second tag 81 C1 indicates the number of bytes of data needed for reading. The third tag 00 DE 6C 21 03 42 5B 3D F7 0A 66 A4 64 DA C7 B2 AF 8A 73 A4 A3 49 D5 A3 CA 1B 2E 7A C5 A2 30 79 DF 7A C2 21 29 F4 26 7F 79 9E DA 46 02 FC 41 34 66 7F 98 B7 01 A3 EF 70 97 7A 4B 7C 71 D8 2F 7F 31 5F 4E 13 7B 34 C8 ... skipping 96 bytes ... AC 9A AC 37 3D 99 B9 35 9D 50 3A 7A 87 4C B6 18 C4 46 A9 69 EB 51 A0 D9 D6 ED represents the value of this integer.

The sixth integer corresponds to the prime2. Its decimal value is: 202308972006098121813033260152694938523623128612212517188524352636642... According to the decoded base64 data displayed, this integer starts at the offset of 986 and has length of 3+193. The first tag 02 indicates the type of Integer. The second tag 81 C1 indicates the number of bytes of data needed for reading. The third tag 00 D6 DF 7A 60 D7 F2 63 20 5B 16 3E 3B 30 0C D4 7F EE 88 19 A1 4A EC 7E 94 11 45 DA F4 0A C7 7A 86 8C FF A8 E5 0D BE 99 7F CB 07 B9 C1 2C 1E 19 27 F4 CA 75 61 32 4D 17 19 A1 73 D9 45 1C 9A 4A 7A FE A8 ... skipping 96 bytes ... 4C 1E 56 58 35 30 61 93 C7 C8 39 B8 8B FC A0 C8 5F 57 CD 0E 27 B7 8D A4 61 48 7F 1C 94 47 represents the value of this integer.

The seventh integer corresponds to the exponent1. Its decimal value is: 726218670704741067738854032873299659669501904654226398730621693308122... According to the decoded base64 data displayed, this integer starts at the offset of 1182 and has length of 3+192. The first tag 02 indicates the type of Integer. The second tag 81 C0 indicates the number of bytes of data needed for reading. The third tag 4D 21 C4 45 04 12 EE 27 28 75 35 D7 3F C3 76 E4 C7 27 99 55 82 FA 52 65 79 0D A2 47 BD 1A 00 79 8E 71 99 DC 99 AD 6C 11 7A 14 09 A9 69 5F 17 53 E0 35 CE 69 04 37 B9 9C DE 2D 25 47 B9 87 A7 1D 31 06 BB 5C C3 8E 1C 0E 98 A3 95 3A 71 E3 02 ... skipping 96 bytes ... 24 14 53 7F 6E 0B CC 04 EC D8 3D 41 FC B9 C2 31 75 represents the value of this integer.

The eighth integer corresponds to the exponent2. Its decimal value is: 183626794021434407678842233855426071045023754604230611486021195913616... According to the decoded base64 data displayed, this integer starts at the offset of 1377 and has length of 3+193. The first tag 02 indicates the type of Integer. The second tag 81 C1 indicates the number of bytes of data needed for reading. The third tag 00 C3 07 D3 0F 65 D2 B7 10 9E 3C A1 3D 9D CE 77 6E AB 36 53 75 28 09 62 96 C8 DA CB F2 E8 79 C9 07 A9 A1 40 4E AF 5C 09 FD 89 6E 38 B7 52 70 FB 4A 5E 11 09 DB B0 09 72 E7 2B B9 78 43 17 C4 1F C8 65 AE AF 2C 79 40 6B 2E 7B 8F 7D ... skipping 96 bytes ... 65 E9 A6 4D 41 7F 3A 7F F7 B4 C2 9D 60 28 AB 44 35 C6 52 4D 6F represents the value of this integer.

The ninth integer corresponds to the coefficient. Its decimal value is: 108682418397149405920388492526093087406631444513650898965338546280167... According to the decoded base64 data displayed, this integer starts at the offset of 1573 and has length of 3+192. The first tag 02 indicates the type of Integer. The second tag 81 C1 indicates the number of bytes of data needed for reading. The third tag 73 6E 92 4E 24 4B 7B 50 60 51 F0 1D CE 32 15 00 EA 50 67 44 AD 8E 5B A1 9C 91 67 06 67 95 51 86 21 62 33 54 B2 82 62 EC 32 51 7C 09 E1 5F 01 FC DA 85 E6 FD E1 DD 41 01 C6 6D 1E 22 AD B1 07 40 72 46 5C 6F 49 16 AD 36 ... skipping 96 bytes ... E1 5F 07 CC B0 91 5D 84 80 CB 9A 7E 88 AC 0D CF 8A 09 49 49 F8 4B C9 EF represents the value of this integer.

Since the version number is 0, it means that the term of otherPrimeInfos does not appear in my RSA private key file.

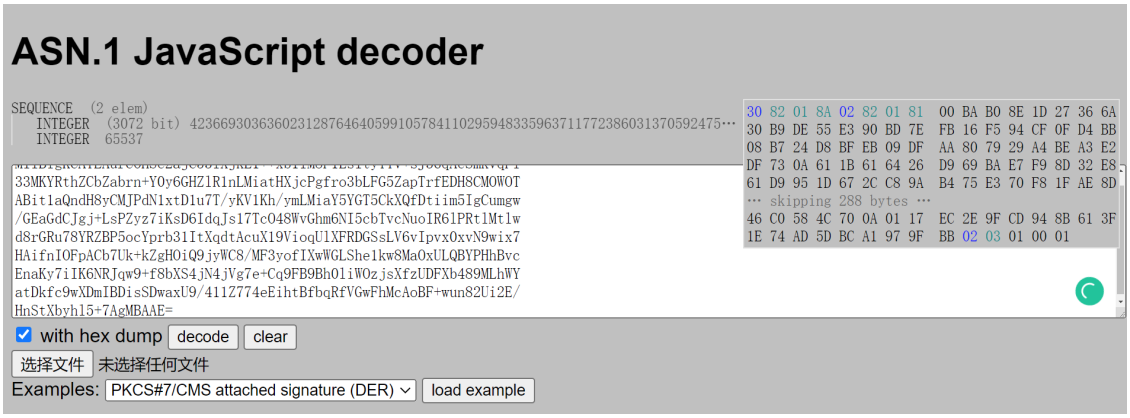
Public Key

In my public key file, I expect to see this list of terms: modulus; publicExponent.

I find that the three decoders do not work if I take my .pub file as the input directly. After looking it up through almighty internet, I learn about a command line which converts my OpenSSH public key format into a PEM format according to the blog at: https://blog.oddbit.com/post/2011-05-08-converting-openssh-public-keys/. My result in the terminal is displayed as below:

```
PS C:\Users\james\Desktop> ssh-keygen -f id_rsa_homework.pub -e -m pem
-----BEGIN RSA PUBLIC KEY-----
MIIBigKCAYEAurCOHSc2ajC53lXjkL1++xb1lM8P1LsItyTYv+sJ36qAeSmkvqPi
33MKYRthZCbZabrn+Y0y6GHZlR1nLMiatHXjcPgfro3bLFG5ZapTrfEDH8CMOWOT
ABit1aQndH8yCMJPdN1xtD1u7T/yKV1Kh/ymLMiaY5YGT5CkXQfDtiim5IgCumgw
/GEaGdCJgj+LsPZyz7iKsD6IdqJs17TcO48WvGhm6NI5cbTvcNuoIR6lPRtlMtlw
d8rGRu78YRZBP5ocYprb31ItXqdtAcuX19VioqUlXFRDGSsLV6vIpvx0xvN9wix7
HAifnIOFpACb7Uk+kZgHOiQ9jyWC8/MF3yofIXwWGLShe1kw8MaOxULQBYPHhBvc
EnaKy7iIK6NRJqw9+f8bXS4jN4jVg7e+Cq9FB9Bh0liWOzjsXfzUDFXb489MLhWY
atDkfc9wXDmIBDisSDwaxU9/411Z774eEihtBfbqRfVGwFhMcAoBF+wun82Ui2E/
HnStXbyhl5+7AgMBAAE=
-----END RSA PUBLIC KEY-----
PS C:\Users\james\Desktop>
```

I then copy the translated text to the Lapo Luchini's ASN.1 decoder and derive the result similar to private key file's as following:



The first integer corresponds to the modulus. Its decimal value is: 423669303636023128764640599105784110295948335963711772386031370592475... According to the decoded base64 data displayed, this integer starts at the offset of 4 and has length of 4+385. The first tag 02 indicates the type of Integer. The second tag 82 01 81 indicates the number of bytes of data needed for reading. The third tag 00 BA B0 8E 1D 27 36 6A 30 B9 DE 55 E3 90 BD 7E FB 16 F5 94 CF 0F D4 BB 08 B7 24 D8 BF EB 09 DF AA 80 79 29 A4 BE A3 E2 DF 73 0A 61 1B 61 64 26 D9 69 BA E7 F9 8D 32 E8 61 D9 95 1D 67 2C C8 9A B4 75 E3 70 F8 1F AE 8D … skipping 288 bytes … 46 C0 58 4C 70 0A 01 17 EC 2E 9F CD 94 8B 61 3F 1E 74 AD 5D BC A1 97 9F BB represents the value of this integer.

The second integer corresponds to the publicExponent. Its decimal value is 65537. According to the decoded base64 data displayed, this integer starts at the offset of 393 and has length of 2+3. The first tag 02 indicates the type of Integer. The second tag 03 indicates the number of bytes of data needed for reading. The third tag 01 00 01 represents the value of this integer.

I note that these two values contained in the public key file are the same as corresponding values contained in the private key file. This makes sense to me since they belong to this one pair of keys I created. These two values have to be the same so that I can use my private and public keys for encryption and decryption appropriately.

Sanity check

To examine the integers I found in my files, I extract the values and write a python program to execute complicated mathematical calculations for me. According to the result generated by my program, I conclude that: $n = p \times q$, $e \times d \ mod \ \lambda(n) = 1$, e and $(p-1)(q-1)$ have greatest common divisor 1. These relationships match my expectations of these integers I found in my RSA key pair. Results and codes of my program is displayed below: