# First of all, we need to find out the details of the structure in this certificate :

```
openssl x509 -in cert -text -noout
```

# We will see something like :

```
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number: 12345 (0x3039)
        Signature Algorithm: md2WithRSAEncryption
        Issuer: CN=PicoCTF
        Validity
            Not Before: Jul  8 07:21:18 2019 GMT
            Not After : Jun 26 17:34:38 2019 GMT
        Subject: OU=PicoCTF, O=PicoCTF, L=PicoCTF, ST=PicoCTF, C=US, CN=PicoCTF
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (53 bit)
                Modulus: 4966306421059967 (0x11a4d45212b17f)
                Exponent: 65537 (0x10001)
    Signature Algorithm: md2WithRSAEncryption
    Signature Value:
        07:6a:5d:61:32:c1:9e:05:bd:eb:77:f3:aa:fb:bb:83:82:eb:
        9e:a2:93:af:0c:2f:3a:e2:1a:e9:74:6b:9b:82:d8:ef:fe:1a:
        c8:b2:98:7b:16:dc:4c:d8:1e:2b:92:4c:80:78:85:7b:d3:cc:
        b7:d4:72:29:94:22:eb:bb:11:5d:b2:9a:af:7c:6b:cb:b0:2c:
        a7:91:87:ec:63:bd:22:e8:8f:dd:38:0e:a5:e1:0a:bf:35:d9:
        a4:3c:3c:7b:79:da:8e:4f:fc:ca:e2:38:67:45:a7:de:6e:a2:
        6e:71:71:47:f0:09:3e:1b:a0:12:35:15:a1:29:f1:59:25:35:
        a3:e4:2a:32:4c:c2:2e:b4:b5:3d:94:38:93:5e:78:37:ac:35:
        35:06:15:e0:d3:87:a2:d6:3b:c0:7f:45:2b:b6:97:8e:03:a8:
        d4:c9:e0:8b:68:a0:c5:45:ba:ce:9b:7e:71:23:bf:6b:db:cc:
        8e:f2:78:35:50:0c:d3:45:c9:6f:90:e4:6d:6f:c2:cc:c7:0e:
        de:fa:f7:48:9e:d0:46:a9:fe:d3:db:93:cb:9f:f3:32:70:63:
        cf:bc:d5:f2:22:c4:f3:be:f6:3f:31:75:c9:1e:70:2a:a4:8e:
        43:96:ac:33:6d:11:f3:ab:5e:bf:4b:55:8b:bf:38:38:3e:c1:
        25:9a:fd:5f
```

# We can see the “n” (4966306421059967) is quite small, so we have to use “https://factordb.com” to break down the "n" in prime numbers

# The result should be something like : 67867967 · 73176001 (p, q)
# After a few tries, I found out the flag format is picoCTF{q,p}

THE FLAG : picoCTF{73176001,67867967}
~Z4que