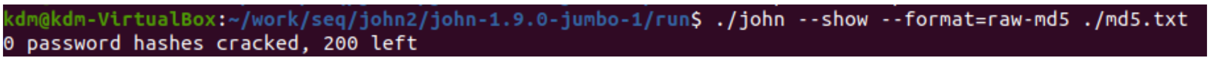
Password Cracking

2017313135 소프트웨어학과 권동민

Q1. Password cracking md5.txt using John the Ripper

John the ripper version : john-1.9.0-jumbo-1

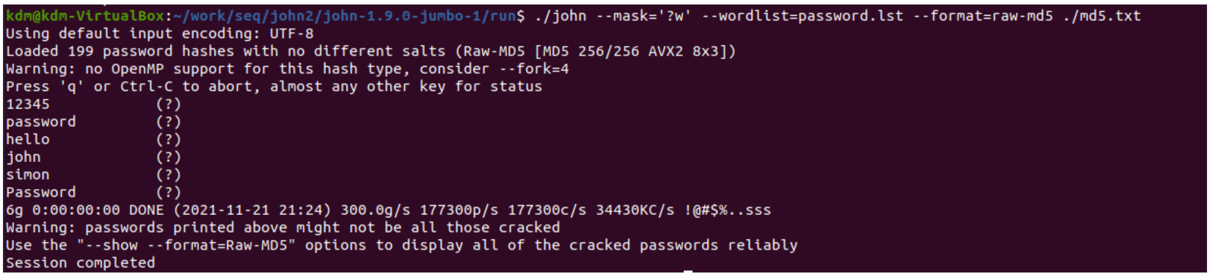
Check the first state



```
kdm@kdm-VirtualBox:~/work/seq/john2/john-1.9.0-jumbo-1/run$ ./john --show --format=raw-md5 ./md5.txt
0 password hashes cracked, 200 left
```

I first using just default password.lst file to crack the file md5.txt.

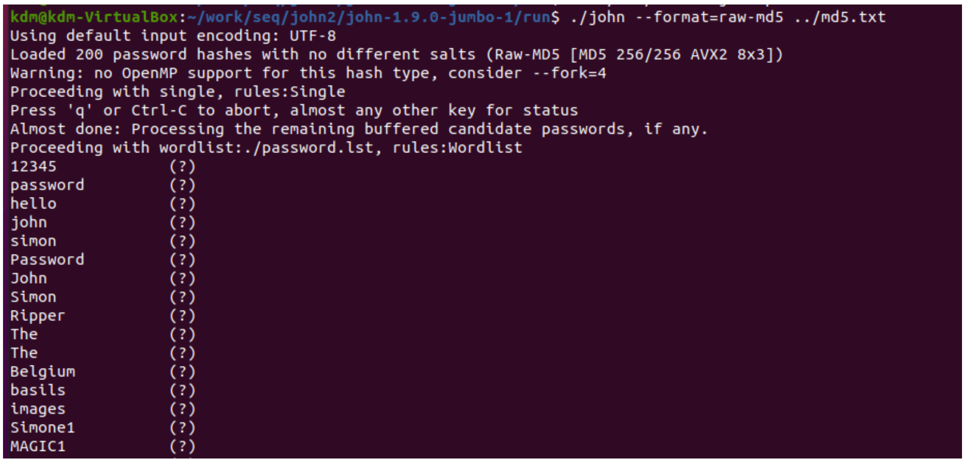
(I just set the mask '₩w' It means that I used password.lst only – not repeat)



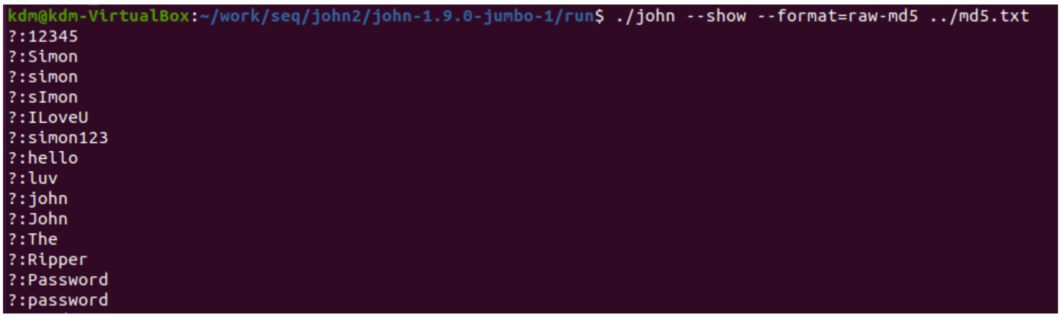
```
kdm@kdm-VirtualBox:~/work/seq/john2/john-1.9.0-jumbo-1/run$ ./john --mask='?w' --wordlist=password.lst --format=raw-md5 ./md5.txt
Using default input encoding: UTF-8
Loaded 199 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
12345            (?)
password         (?)
hello            (?)
john             (?)
simon            (?)
Password         (?)
6g 0:00:00:00 DONE (2021-11-21 21:24) 300.0g/s 177300p/s 177300c/s 34430KC/s !@#$%..sss
Warning: passwords printed above might not be all those cracked
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed
```

There are just 6 password that is cracked.

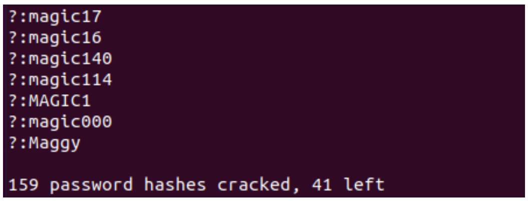
In general, if it is hybrid mode, it means to crack by mask and password list to specify the string's form. But I don't have any idea about strings of sample md5.txt, so I just used standard mode to crack(not using mask and wordlist).



```
kdm@kdm-VirtualBox:~/work/seq/john2/john-1.9.0-jumbo-1/run$ ./john --format=raw-md5 ../md5.txt
Using default input encoding: UTF-8
Loaded 200 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=4
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:./password.lst, rules:Wordlist
12345            (?)
password         (?)
hello            (?)
john             (?)
simon            (?)
Password         (?)
John             (?)
Simon            (?)
Ripper           (?)
The              (?)
The              (?)
Belgium          (?)
basils           (?)
images           (?)
Simone1          (?)
MAGIC1           (?)
```



```
kdm@kdm-VirtualBox:~/work/seq/john2/john-1.9.0-jumbo-1/run$ ./john --show --format=raw-md5 ../md5.txt
?:12345
?:Simon
?:simon
?:sImon
?:ILoveU
?:simon123
?:hello
?:luv
?:john
?:John
?:The
?:Ripper
?:Password
?:password
```



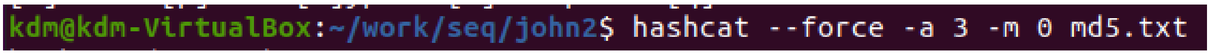
```
?:magic17
?:magic16
?:magic140
?:magic114
?:MAGIC1
?:magic000
?:Maggy

159 password hashes cracked, 41 left
```

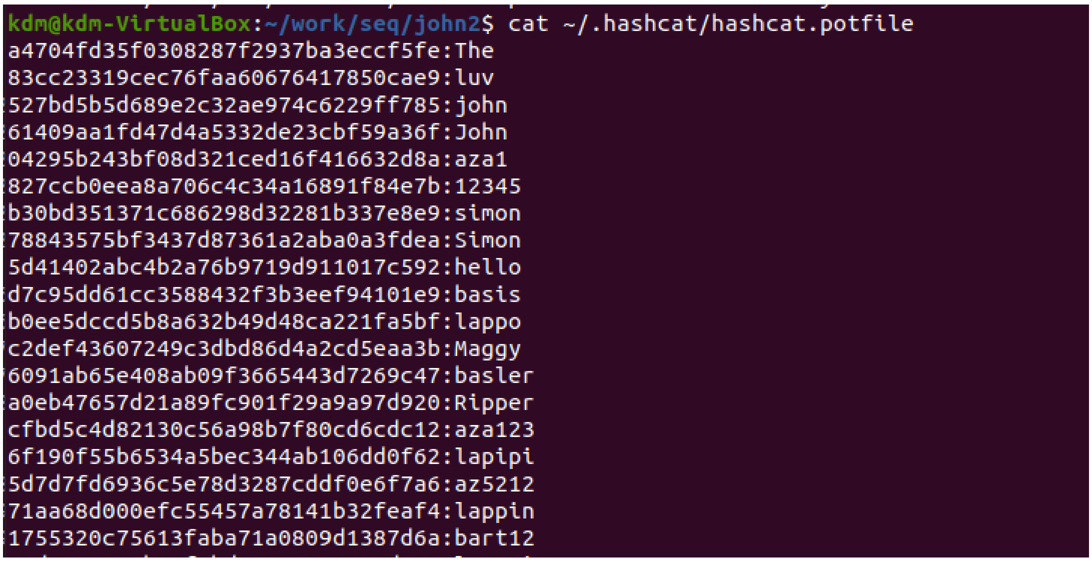
But I can't crack 41 words. (I think that last 41 has special char)

Q2. Password cracking md5.txt using Hashcat

Straight mode:

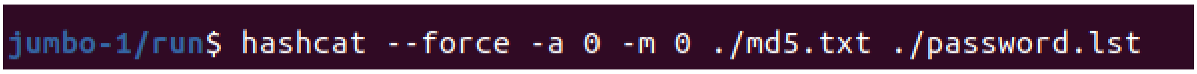


```
kdm@kdm-VirtualBox:~/work/seq/john2$ hashcat --force -a 3 -m 0 md5.txt
```



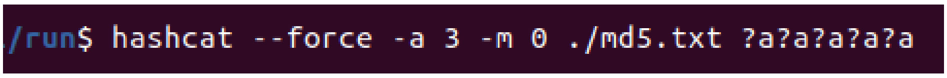
```
kdm@kdm-VirtualBox:~/work/seq/john2$ cat ~/.hashcat/hashcat.potfile
a4704fd35f0308287f2937ba3eccf5fe:The
83cc23319cec76faa60676417850cae9:luv
527bd5b5d689e2c32ae974c6229ff785:john
61409aa1fd47d4a5332de23cbf59a36f:John
04295b243bf08d321ced16f416632d8a:aza1
827ccb0eea8a706c4c34a16891f84e7b:12345
b30bd351371c686298d32281b337e8e9:simon
78843575bf3437d87361a2aba0a3fdea:Simon
5d41402abc4b2a76b9719d911017c592:hello
d7c95dd61cc3588432f3b3eef94101e9:basis
b0ee5dccd5b8a632b49d48ca221fa5bf:lappo
c2def43607249c3dbd86d4a2cd5eaa3b:Maggy
6091ab65e408ab09f3665443d7269c47:basler
a0eb47657d21a89fc901f29a9a97d920:Ripper
cfbd5c4d82130c56a98b7f80cd6cdc12:aza123
6f190f55b6534a5bec344ab106dd0f62:lapipi
5d7d7fd6936c5e78d3287cddf0e6f7a6:az5212
71aa68d000efc55457a78141b32feaf4:lappin
1755320c75613faba71a0809d1387d6a:bart12
```

Wordlist mode:



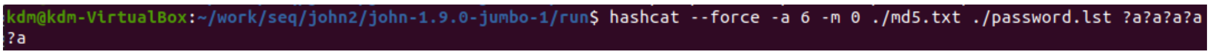
```
jumbo-1/run$ hashcat --force -a 0 -m 0 ./md5.txt ./password.lst
```

Using mask:

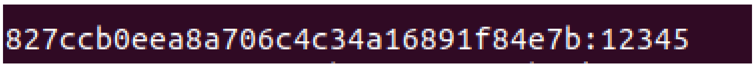


```
/run$ hashcat --force -a 3 -m 0 ./md5.txt ?a?a?a?a?a
```

Hybrid mode:



```
kdm@kdm-VirtualBox:~/work/seq/john2/john-1.9.0-jumbo-1/run$ hashcat --force -a 6 -m 0 ./md5.txt ./password.lst ?a?a?a?a
?a
```

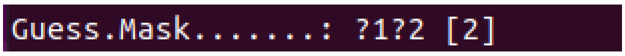


```
827ccb0eea8a706c4c34a16891f84e7b:12345
```

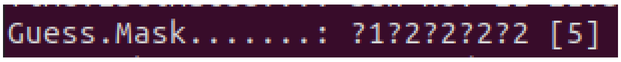
Q3. Which is faster(JDR, Hashcat)

After I used both crack tools(john the ripper and Hashcat). I can see that there is huge difference between Hashcat and John the Ripper. In Hashcat, sometimes it showed some messages during execution because mask is change.

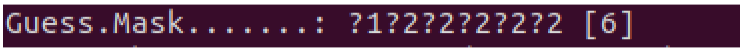
Hashcat:



```
Guess.Mask.......: ?1?2 [2]
```

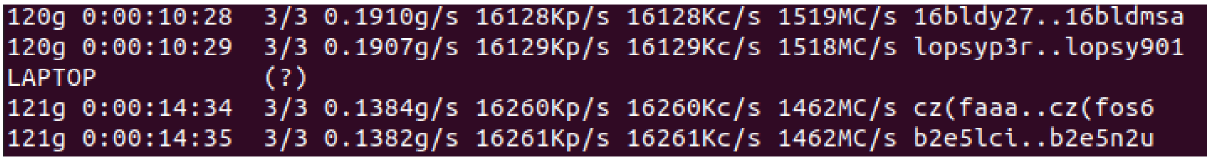


```
Guess.Mask.......: ?1?2?2?2?2 [5]
```



```
Guess.Mask.......: ?1?2?2?2?2?2 [6]
```

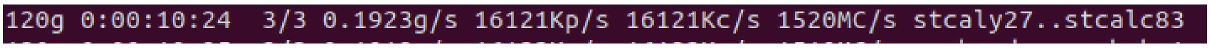
John the Ripper:



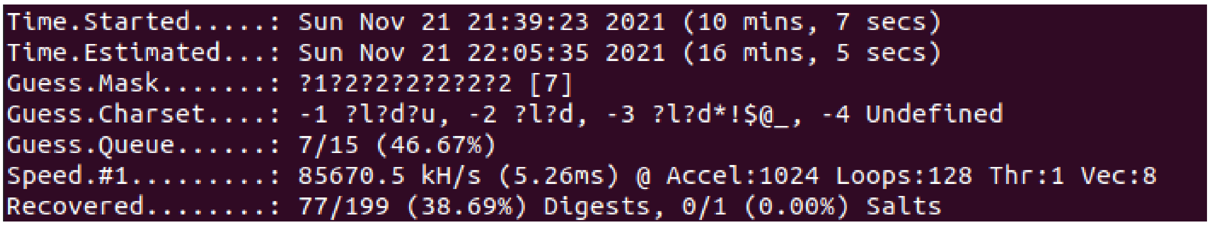
```
120g 0:00:10:28  3/3 0.1910g/s 16128Kp/s 16128Kc/s 1519MC/s 16bldy27..16bldmsa
120g 0:00:10:29  3/3 0.1907g/s 16129Kp/s 16129Kc/s 1518MC/s lopsyp3r..lopsy901
LAPTOP           (?)
121g 0:00:14:34  3/3 0.1384g/s 16260Kp/s 16260Kc/s 1462MC/s cz(faaa..cz(fos6
121g 0:00:14:35  3/3 0.1382g/s 16261Kp/s 16261Kc/s 1462MC/s b2e5lci..b2e5n2u
```

We can know that Hashcat cracks ordered by string's length, but john the ripper's cracking order is not huge related to length of the string.

These are the result of cracking by using each crack tool after 10 minutes from starting.



```
120g 0:00:10:24  3/3 0.1923g/s 16121Kp/s 16121Kc/s 1520MC/s stcaly27..stcalc83
```



```
Time.Started.....: Sun Nov 21 21:39:23 2021 (10 mins, 7 secs)
Time.Estimated...: Sun Nov 21 22:05:35 2021 (16 mins, 5 secs)
Guess.Mask.......: ?1?2?2?2?2?2?2 [7]
Guess.Charset....: -1 ?l?d?u, -2 ?l?d, -3 ?l?d*!$@_, -4 Undefined
Guess.Queue......: 7/15 (46.67%)
Speed.#1.........: 85670.5 kH/s (5.26ms) @ Accel:1024 Loops:128 Thr:1 Vec:8
Recovered........: 77/199 (38.69%) Digests, 0/1 (0.00%) Salts
```

John the ripper cracked 120 words but Hashcat crack just 77 words. So it seems that John the

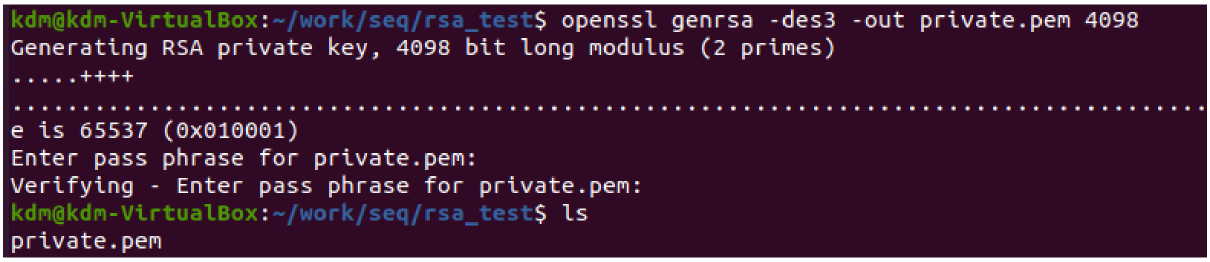
Ripper is more faster than Hashcat.

Q4. Create own password cracking dictionary(wordlist)

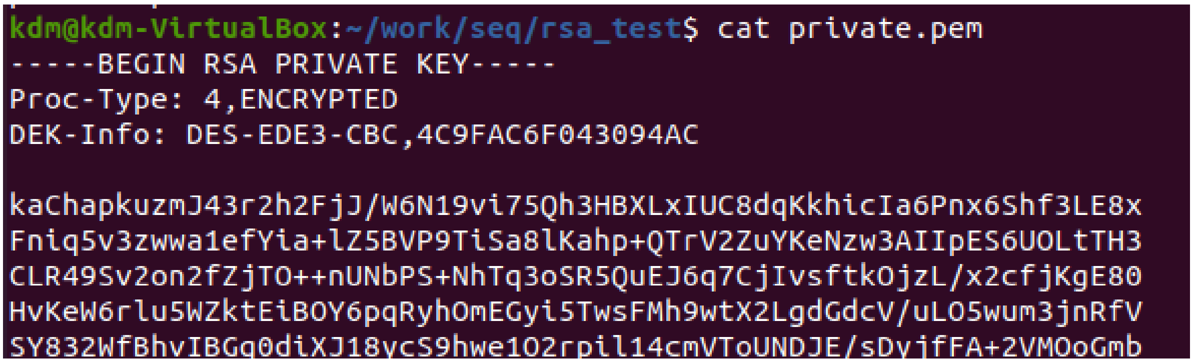
-attach separate file(.txt)

Q5. Create own 4098 bit RSA private, public key using openssl

Order of commands:

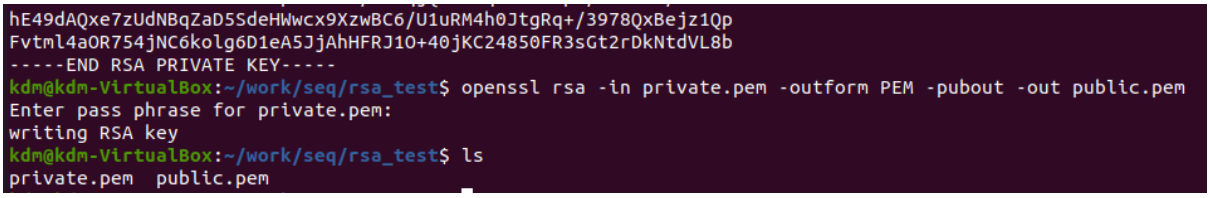


```
kdm@kdm-VirtualBox:~/work/seq/rsa_test$ openssl genrsa -des3 -out private.pem 4098
Generating RSA private key, 4098 bit long modulus (2 primes)
.....++++
.......................................................................................
e is 65537 (0x010001)
Enter pass phrase for private.pem:
Verifying - Enter pass phrase for private.pem:
kdm@kdm-VirtualBox:~/work/seq/rsa_test$ ls
private.pem
```

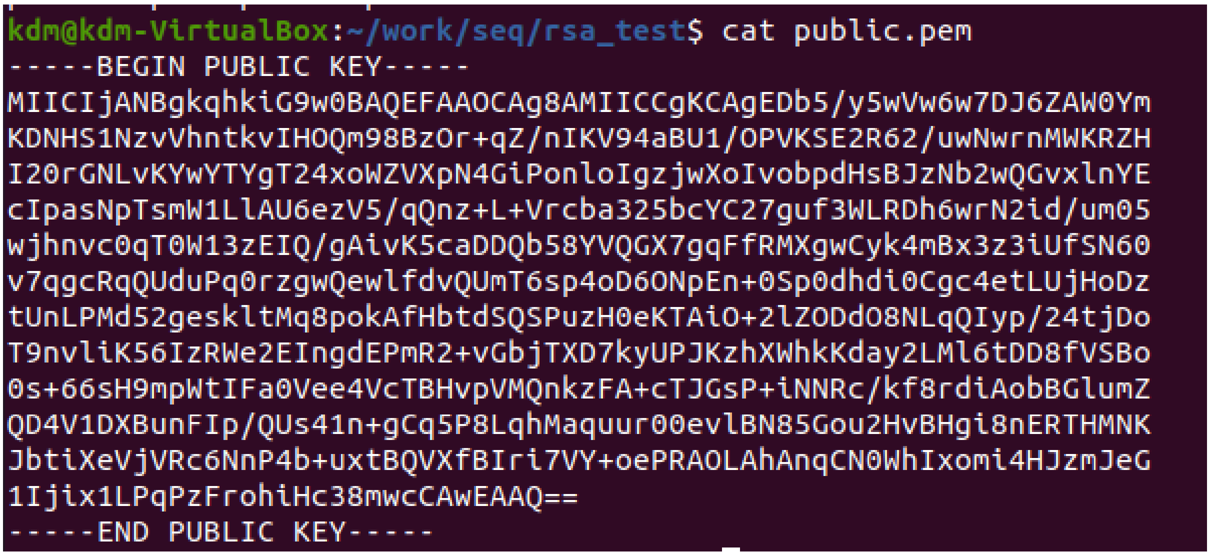


```
kdm@kdm-VirtualBox:~/work/seq/rsa_test$ cat private.pem
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,4C9FAC6F043094AC

kaChapkuzmJ43r2h2FjJ/W6N19vi75Qh3HBXLxIUC8dqKkhicIa6Pnx6Shf3LE8x
Fniq5v3zwwa1efYia+lZ5BVP9TiSa8lKahp+QTrV2ZuYKeNzw3AIIpES6UOLtTH3
CLR49Sv2on2fZjTO++nUNbPS+NhTq3oSR5QuEJ6q7CjIvsftkOjzL/x2cfjKgE80
HvKeW6rlu5WZktEiBOY6pqRyhOmEGyi5TwsFMh9wtX2LgdGdcV/uLO5wum3jnRfV
SY832WfBhvIBGq0diXJ18ycS9hwe1O2rpil14cmVToUNDJE/sDyjfFA+2VMOoGmb
```

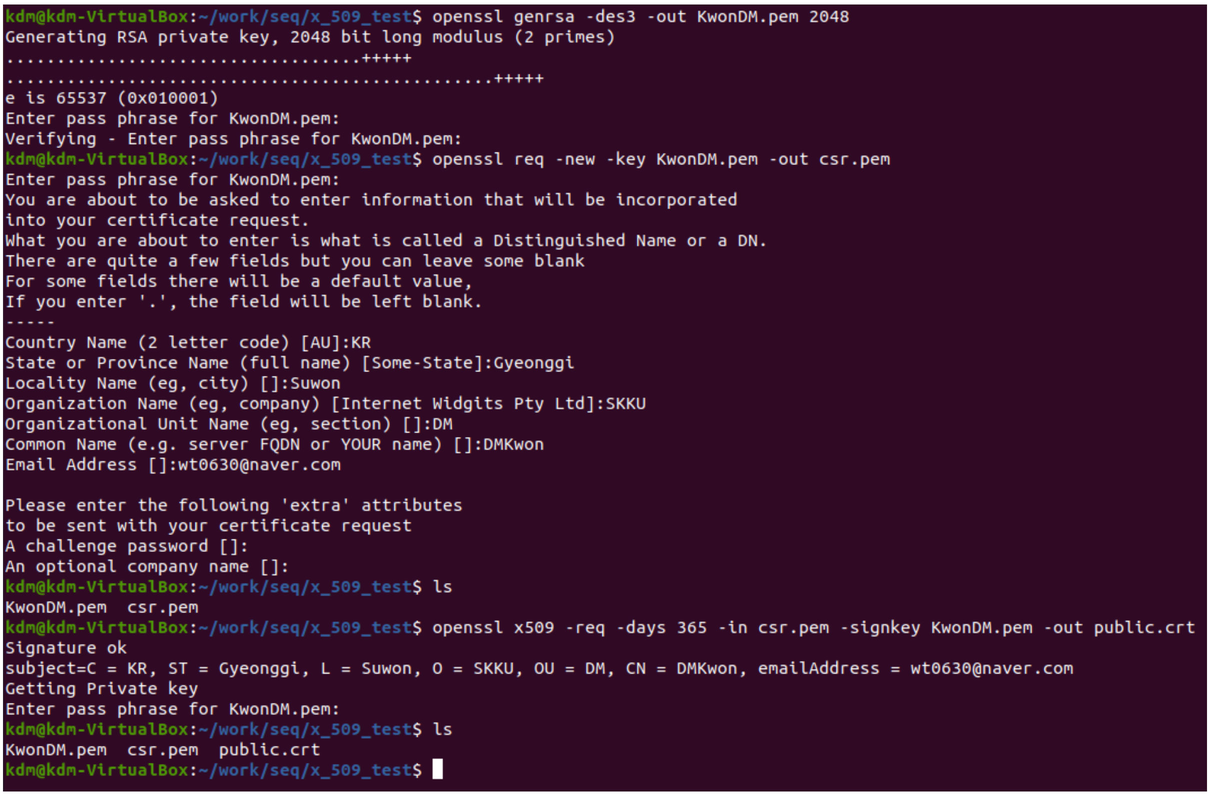


```
hE49dAQxe7zUdNBqZaD5SdeHWwcx9XzwBC6/U1uRM4h0JtgRq+/3978QxBejz1Qp
Fvtml4aOR754jNC6kolg6D1eA5JjAhHFRJ1O+40jKC24850FR3sGt2rDkNtdVL8b
-----END RSA PRIVATE KEY-----
kdm@kdm-VirtualBox:~/work/seq/rsa_test$ openssl rsa -in private.pem -outform PEM -pubout -out public.pem
Enter pass phrase for private.pem:
writing RSA key
kdm@kdm-VirtualBox:~/work/seq/rsa_test$ ls
private.pem  public.pem
```



```
kdm@kdm-VirtualBox:~/work/seq/rsa_test$ cat public.pem
-----BEGIN PUBLIC KEY-----
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEDb5/y5wVw6w7DJ6ZAW0Ym
KDNHS1NzvVhntkvIHOQm98BzOr+qZ/nIKV94aBU1/OPVKSE2R62/uwNwrnMWKRZH
I20rGNLvKYwYTYgT24xoWZVXpN4GiPonloIgzjwXoIvobpdHsBJzNb2wQGvxlnYE
cIpasNpTsmW1LlAU6ezV5/qQnz+L+Vrcba325bcYC27guf3WLRDh6wrN2id/um05
wjhnvc0qT0W13zEIQ/gAivK5caDDQb58YVQGX7gqFfRMXgwCyk4mBx3z3iUfSN60
v7qgcRqQUduPq0rzgwQewlfdvQUmT6sp4oD6ONpEn+0Sp0dhdi0Cgc4etLUjHoDz
tUnLPMd52geskltMq8pokAfHbtdSQSPuzH0eKTAiO+2lZODdO8NLqQIyp/24tjDo
T9nvliK56IzRWe2EIngdEPmR2+vGbjTXD7kyUPJKzhXWhkKday2LMl6tDD8fVSBo
0s+66sH9mpWtIFa0Vee4VcTBHvpVMQnkzFA+cTJGsP+iNNRc/kf8rdiAobBGlumZ
QD4V1DXBunFIp/QUs41n+gCq5P8LqhMaquur00evlBN85Gou2HvBHgi8nERTHMNK
JbtiXeVjVRc6NnP4b+uxtBQVXfBIri7VY+oePRAOLAhAnqCN0WhIxomi4HJzmJeG
1Ijix1LPqPzFrohiHc38mwcCAwEAAQ==
-----END PUBLIC KEY-----
```

Q6. Create own X.509 digital certificate using openssl



```
kdm@kdm-VirtualBox:~/work/seq/x_509_test$ openssl genrsa -des3 -out KwonDM.pem 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
...................................+++++
................................................+++++
e is 65537 (0x010001)
Enter pass phrase for KwonDM.pem:
Verifying - Enter pass phrase for KwonDM.pem:
kdm@kdm-VirtualBox:~/work/seq/x_509_test$ openssl req -new -key KwonDM.pem -out csr.pem
Enter pass phrase for KwonDM.pem:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:KR
State or Province Name (full name) [Some-State]:Gyeonggi
Locality Name (eg, city) []:Suwon
Organization Name (eg, company) [Internet Widgits Pty Ltd]:SKKU
Organizational Unit Name (eg, section) []:DM
Common Name (e.g. server FQDN or YOUR name) []:DMKwon
Email Address []:wt0630@naver.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
kdm@kdm-VirtualBox:~/work/seq/x_509_test$ ls
KwonDM.pem  csr.pem
kdm@kdm-VirtualBox:~/work/seq/x_509_test$ openssl x509 -req -days 365 -in csr.pem -signkey KwonDM.pem -out public.crt
Signature ok
subject=C = KR, ST = Gyeonggi, L = Suwon, O = SKKU, OU = DM, CN = DMKwon, emailAddress = wt0630@naver.com
Getting Private key
Enter pass phrase for KwonDM.pem:
kdm@kdm-VirtualBox:~/work/seq/x_509_test$ ls
KwonDM.pem  csr.pem  public.crt
kdm@kdm-VirtualBox:~/work/seq/x_509_test$ 
```