**Lab 2 Report**

**Task 1**:

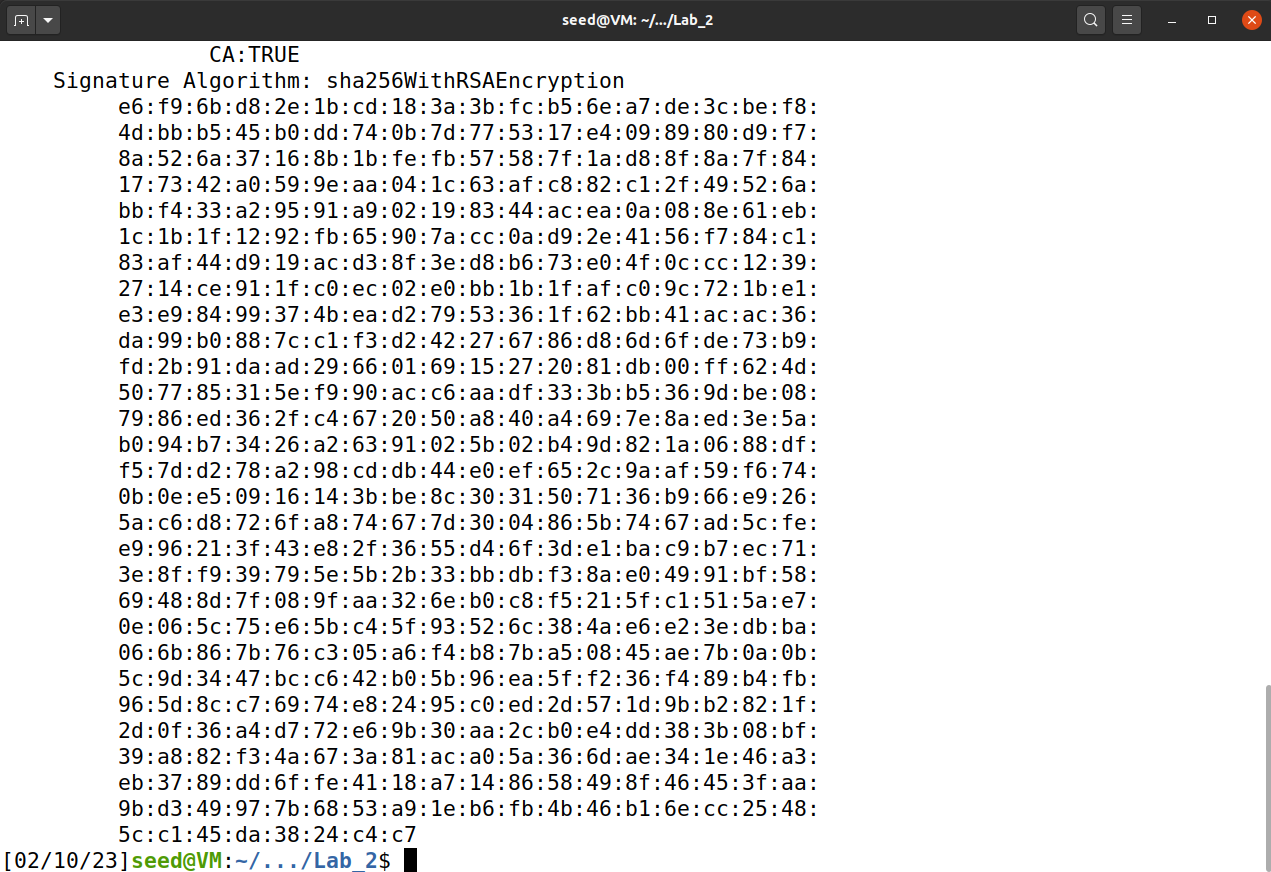
Generate CA

ca.crt

Text

Description automatically generated

Text

Description automatically generated

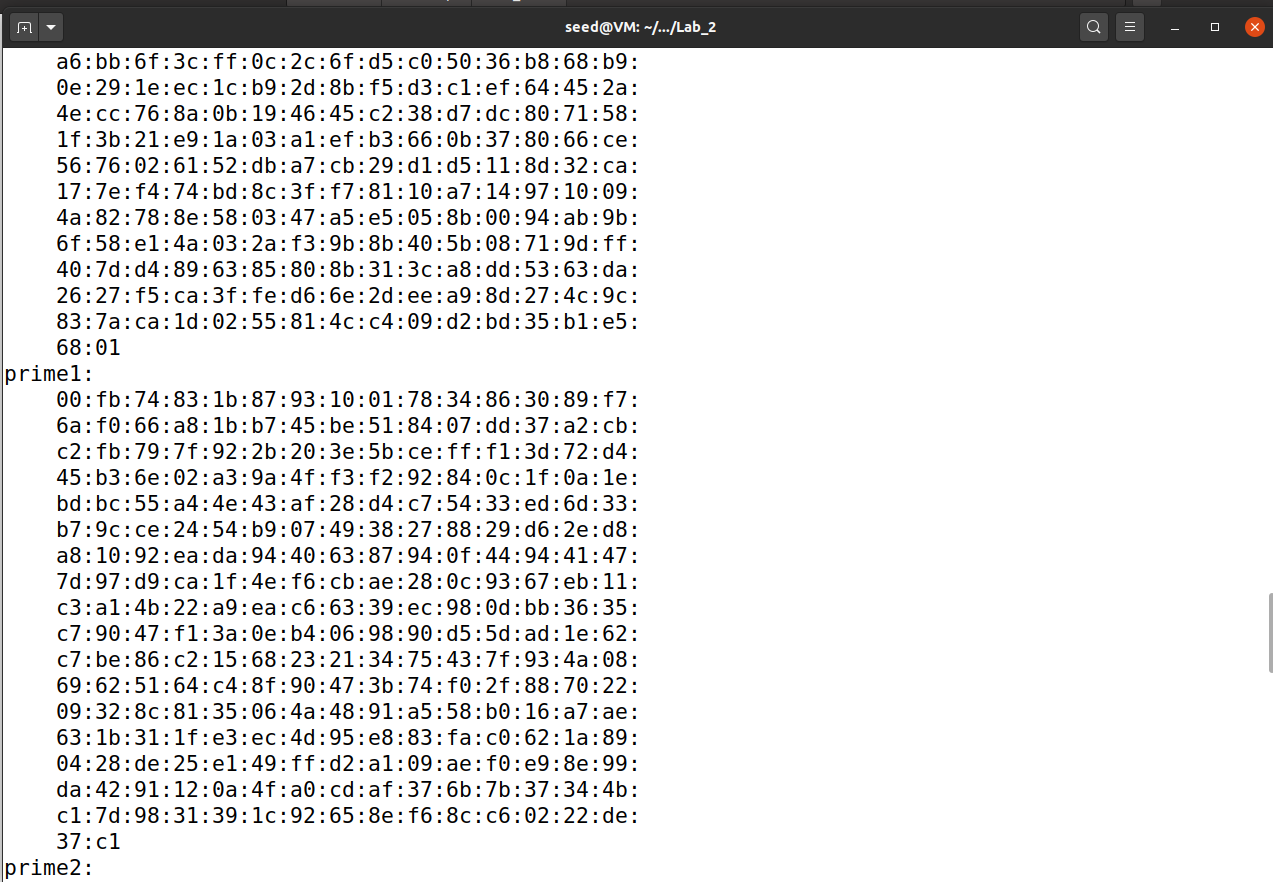
Ca key

Text

Description automatically generated

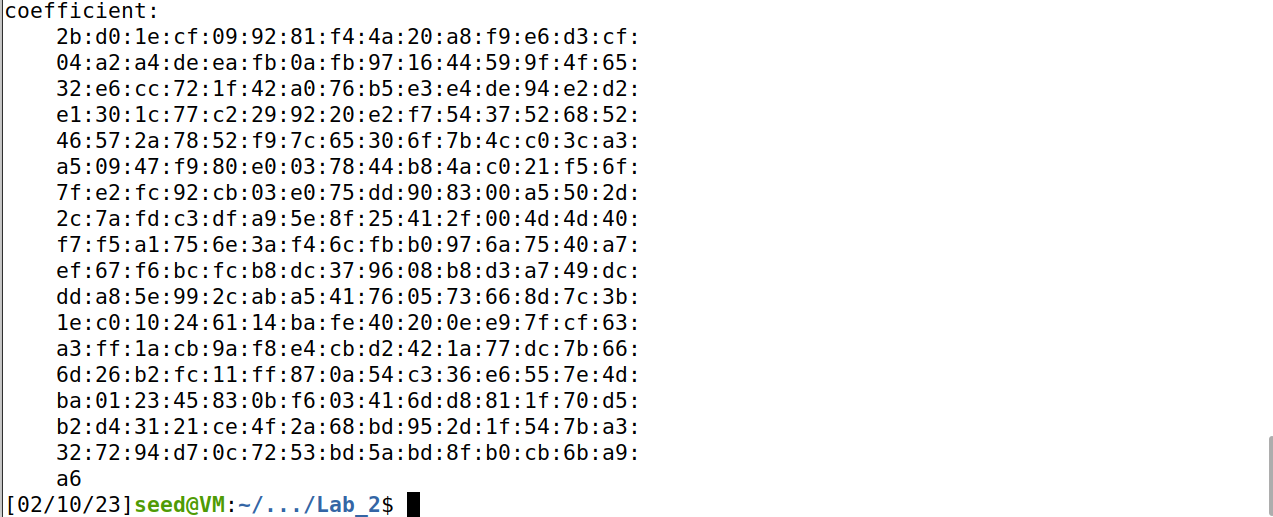
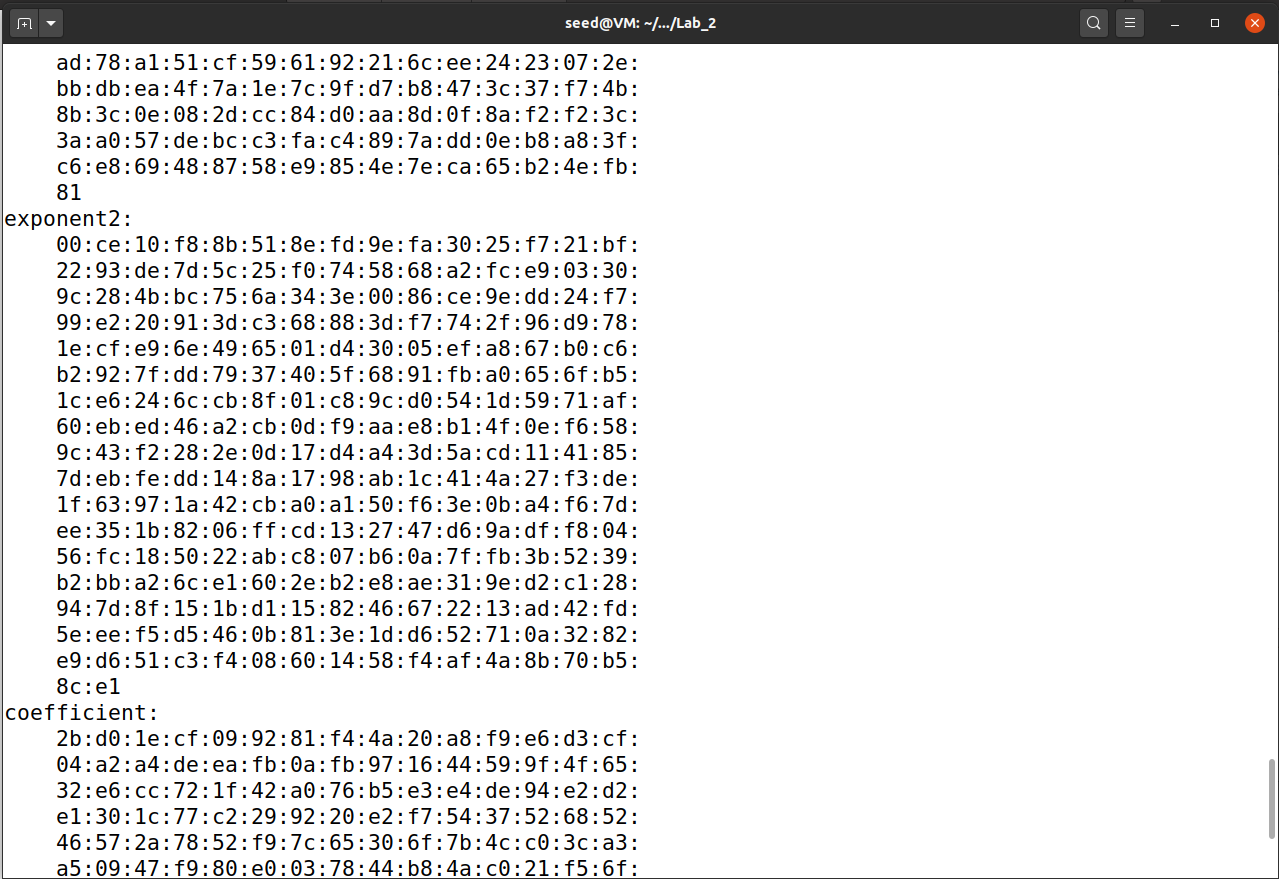
Table

Description automatically generated with medium confidence



Table

Description automatically generated



The CA is generated by rsa 4096 sha256 using my own config.

The Issuer and Subject are identical, indicating that is a self-signed certificate.

When the certificate contains digital signatures, then it is CA certificate. In this case, my own certificate is self-signed.

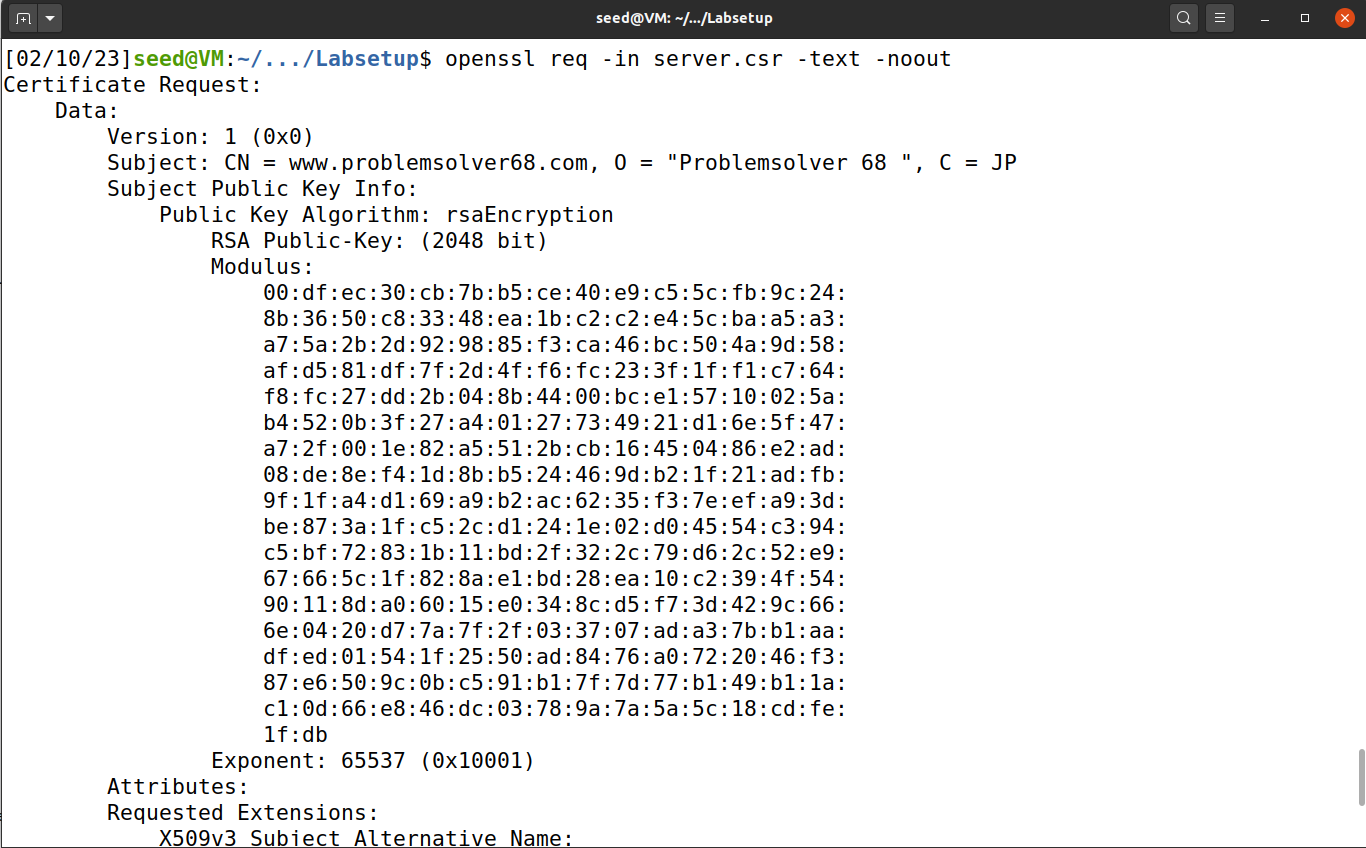
**Task 2:**

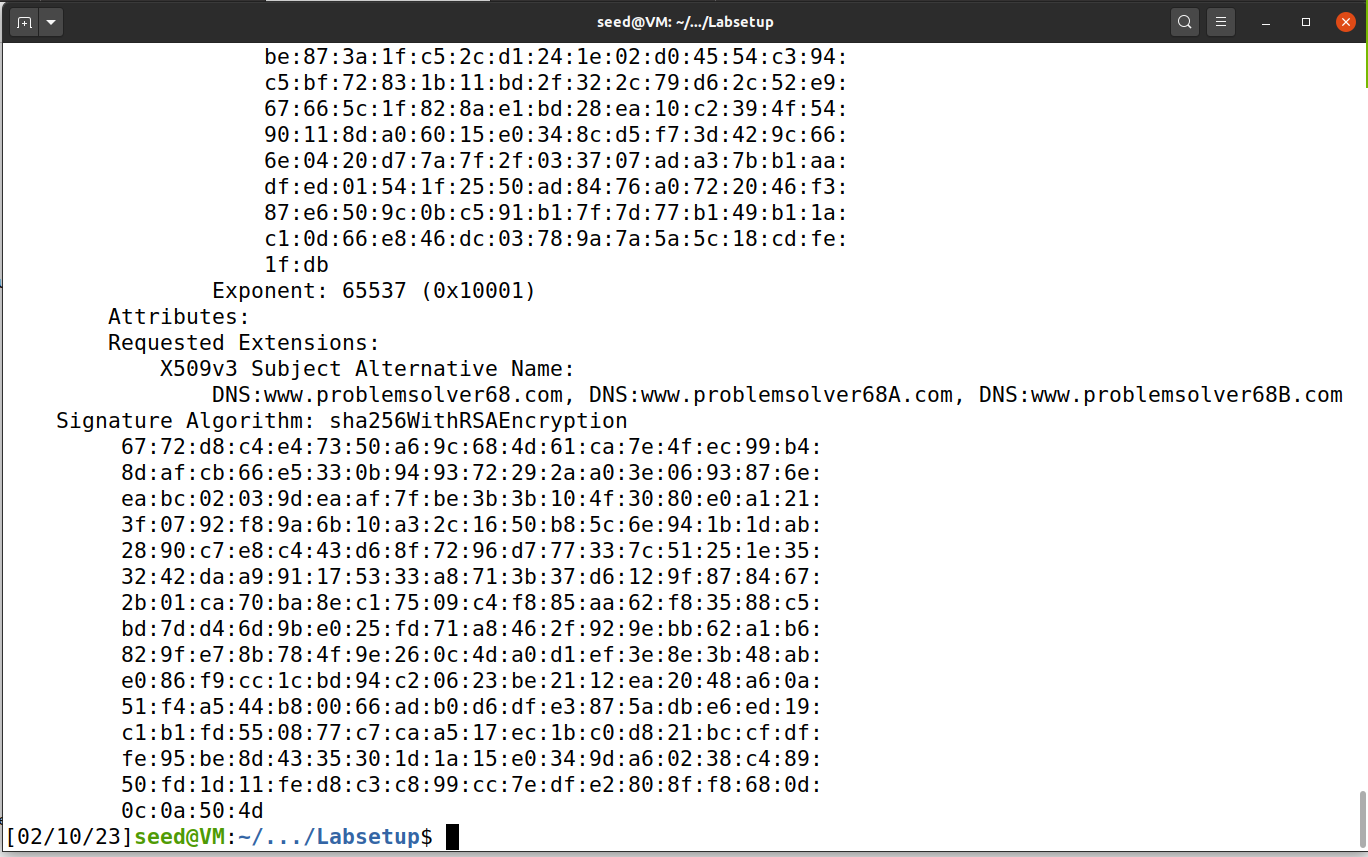
Creating own server key and csr. My own DNS: [www.problemsolver68.com](http://www.problemsolver68.com)

Alternative names command are in the bottom part of the $openssl req command

****

Server csr

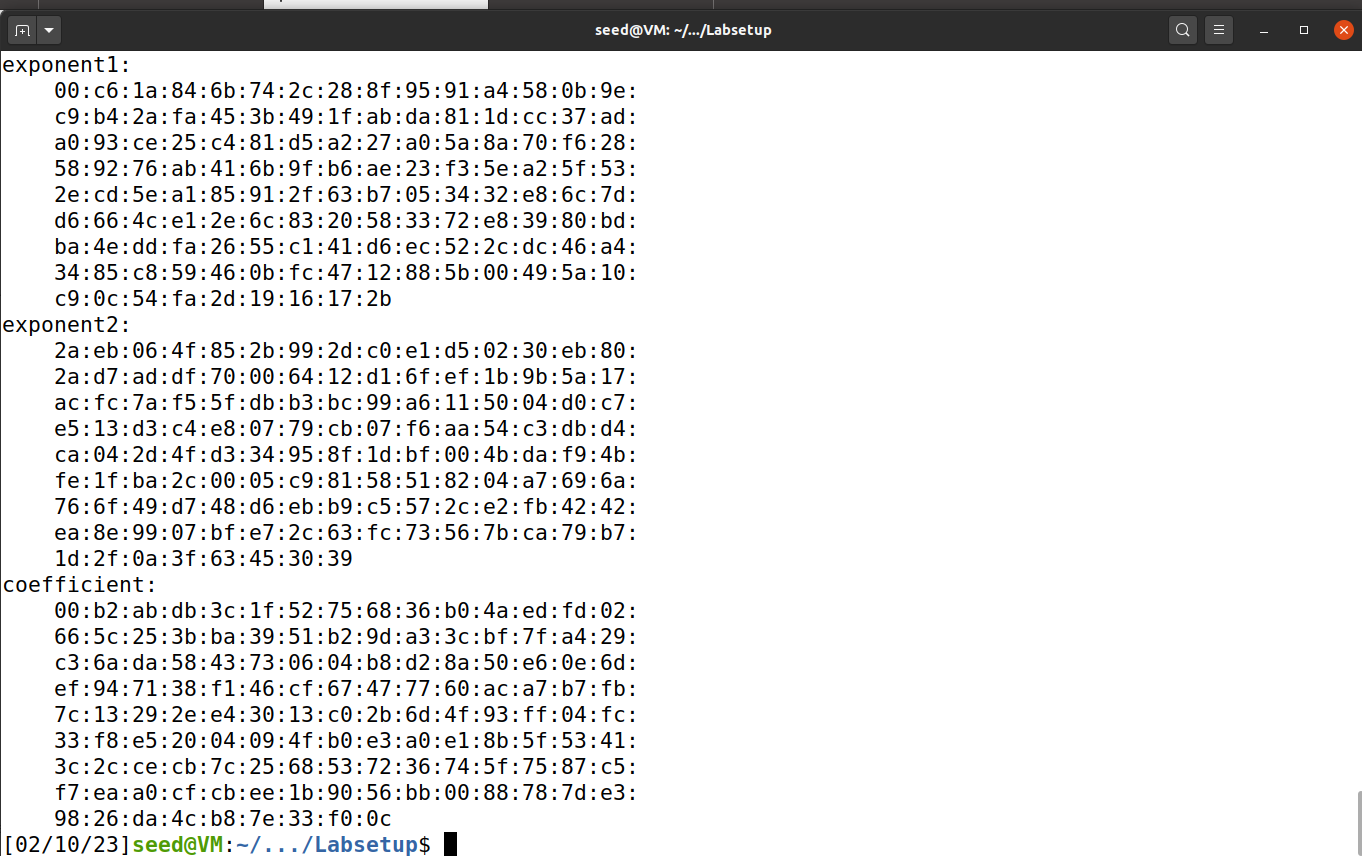
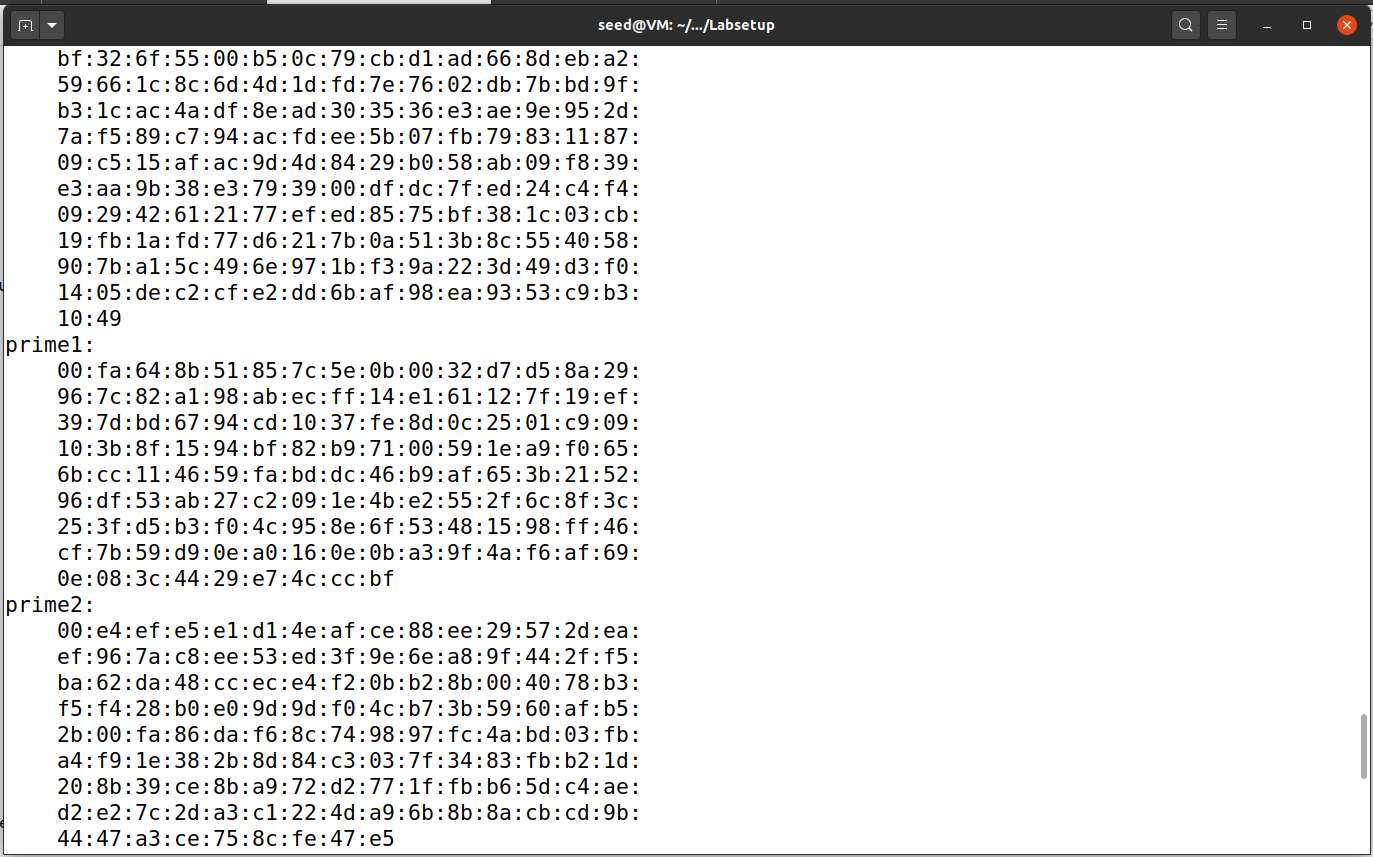




Server key

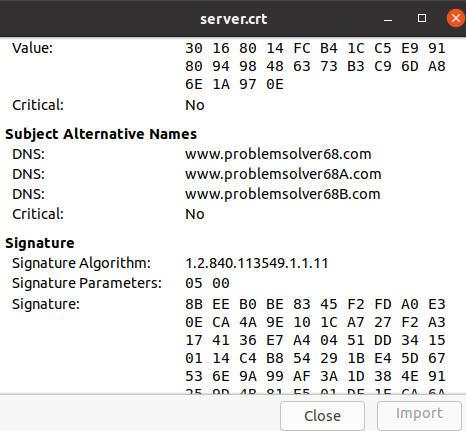
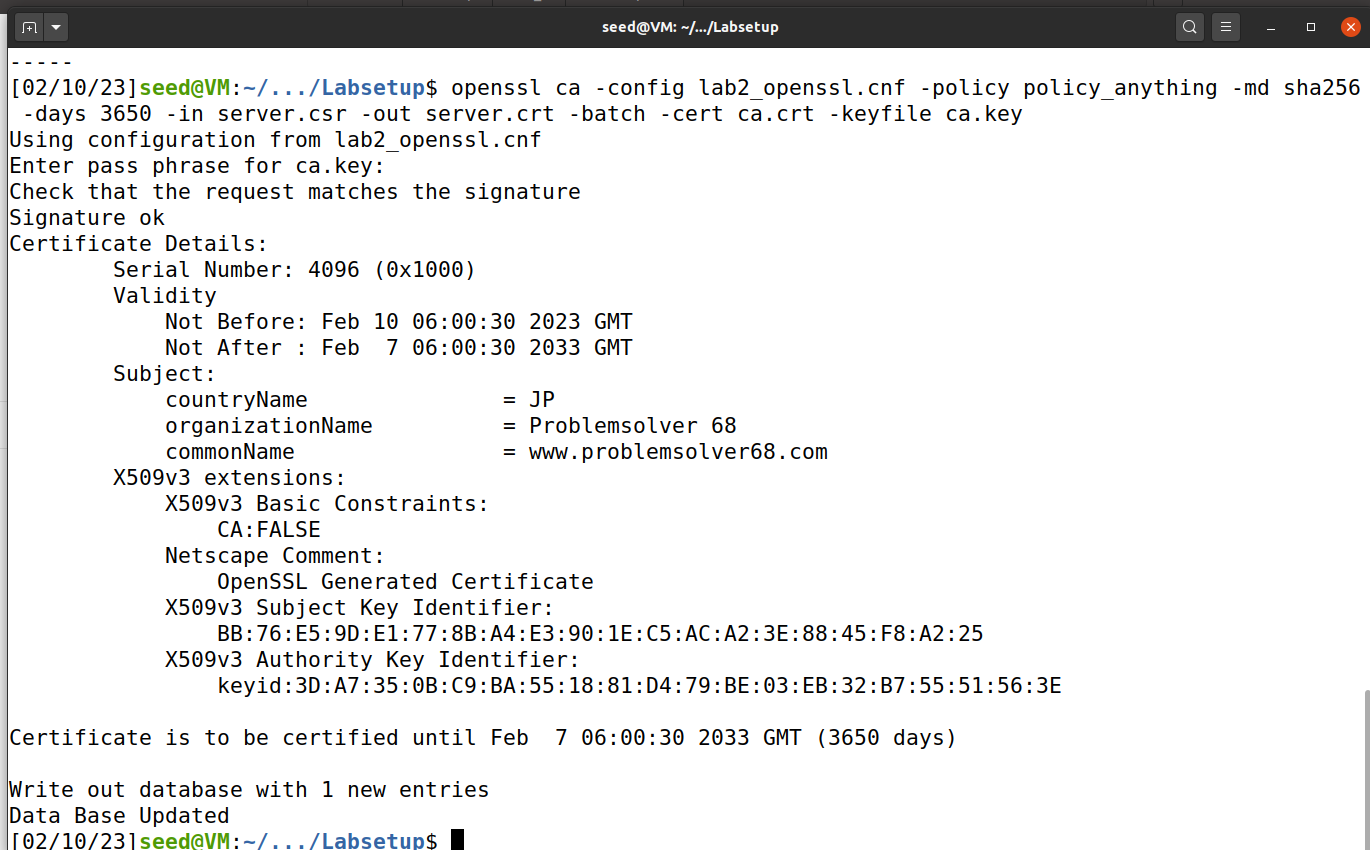
Text

Description automatically generated with medium confidence



**Task 3:**

Sign ca to own server, then server.crt will appear along side server.key and server.csr

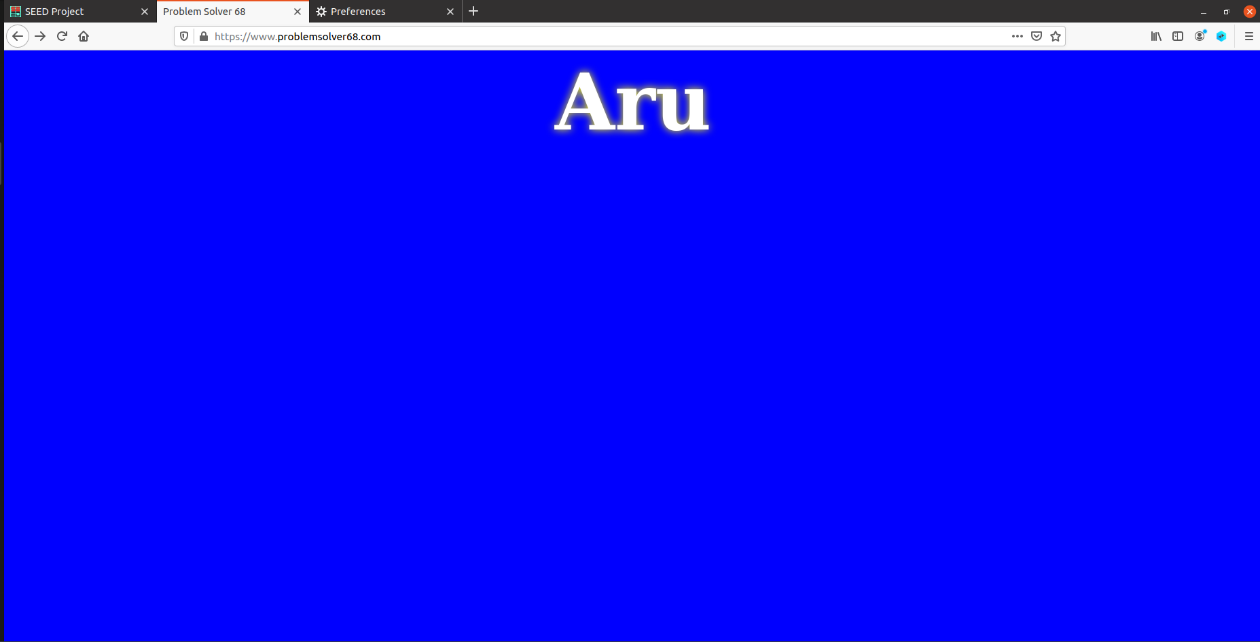


**Task 4**

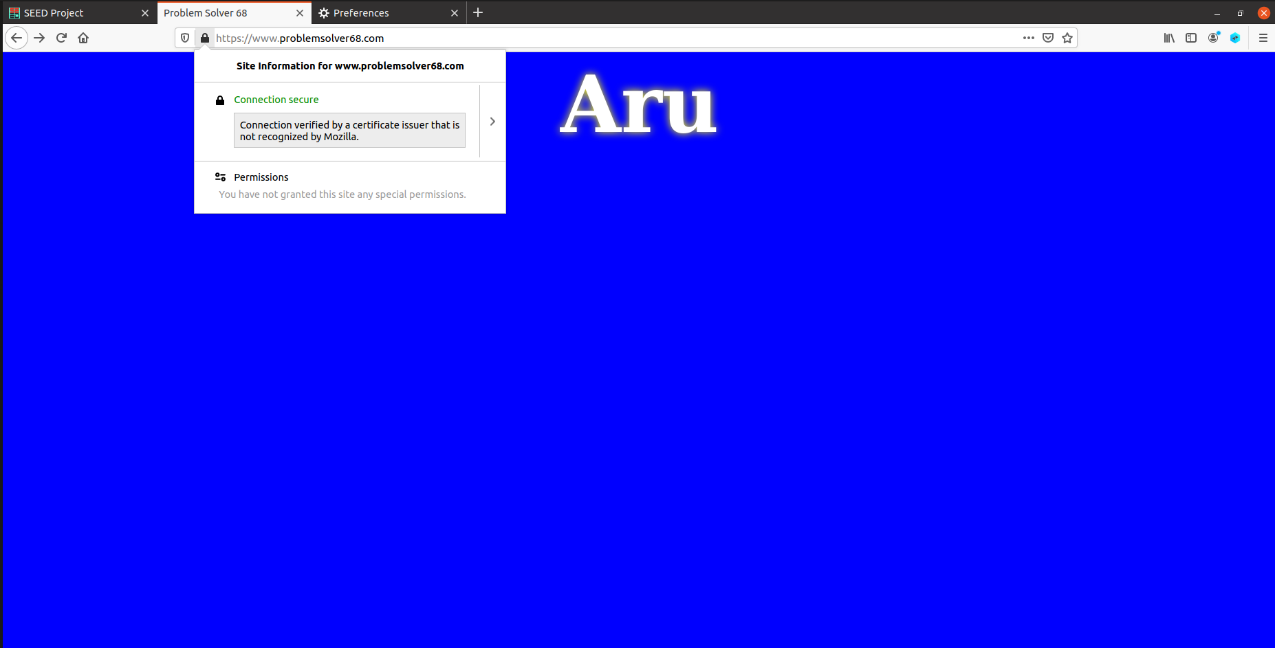
Custom bank32 config file with new certificate directory and YouTube DNS

Graphical user interface, text, application, email

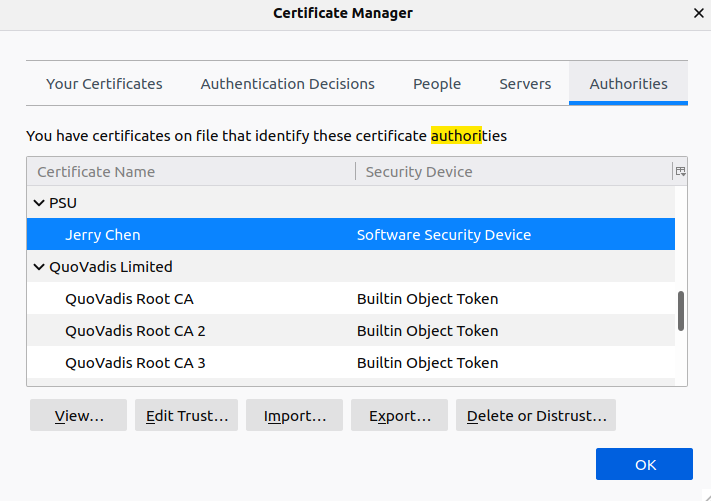
Description automatically generated

Accessed own certificated website

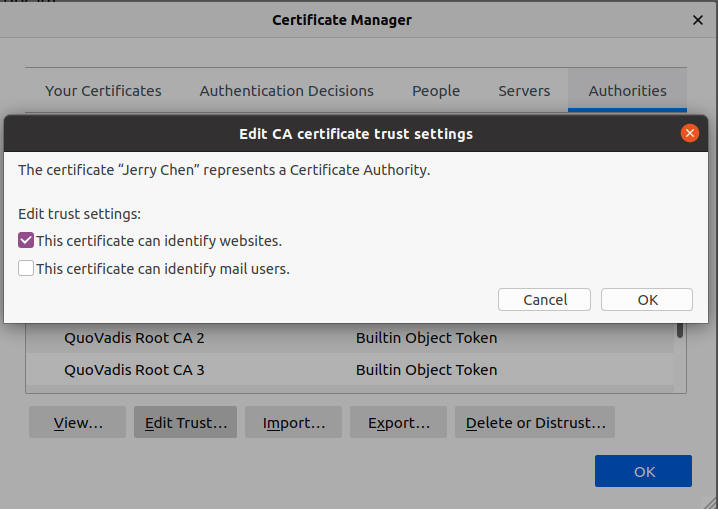
Browser shows certificate on, therefore secure connection



Imported CA



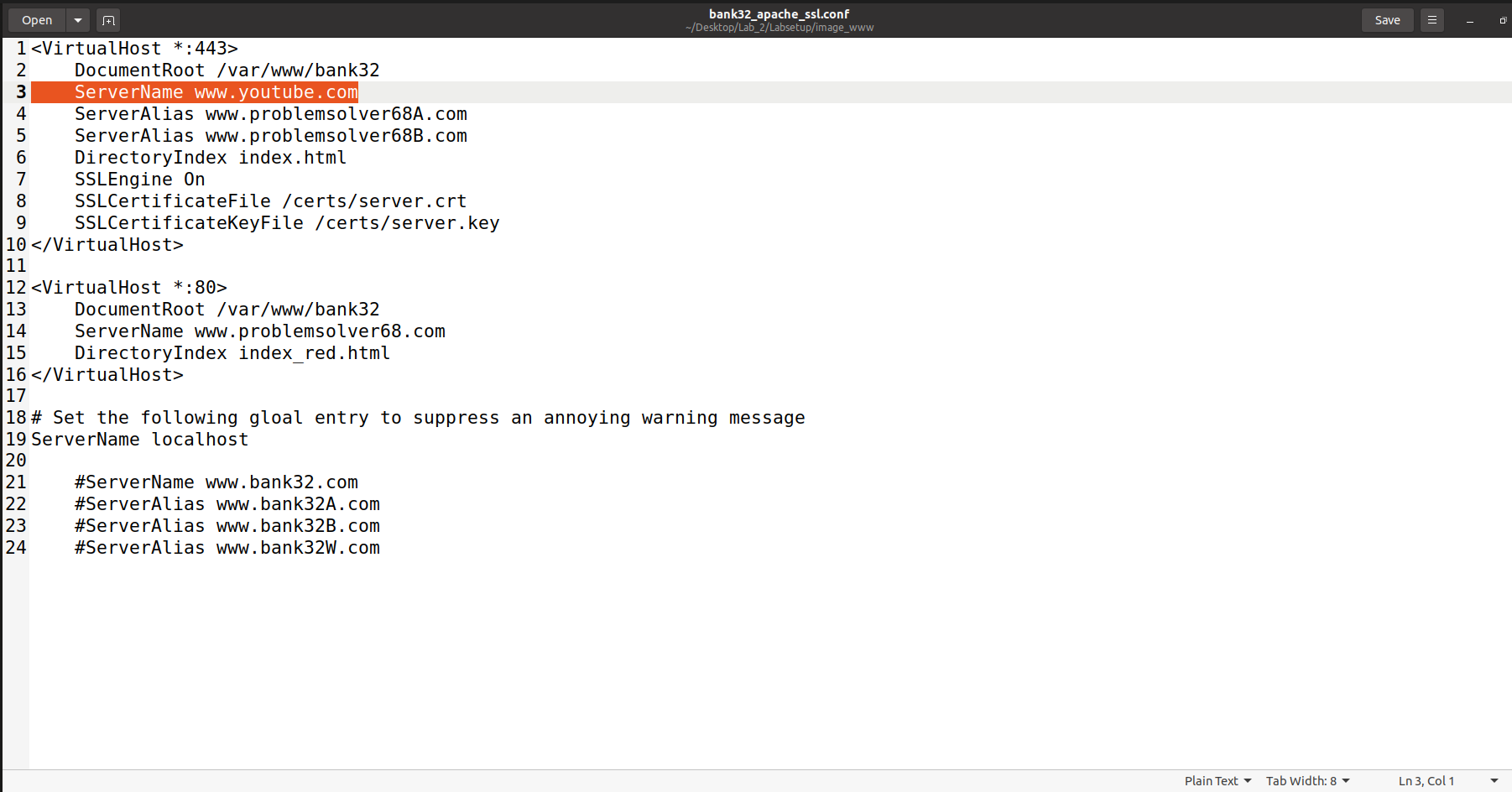
Check identify websites to let browser trust it.



**Task 5**

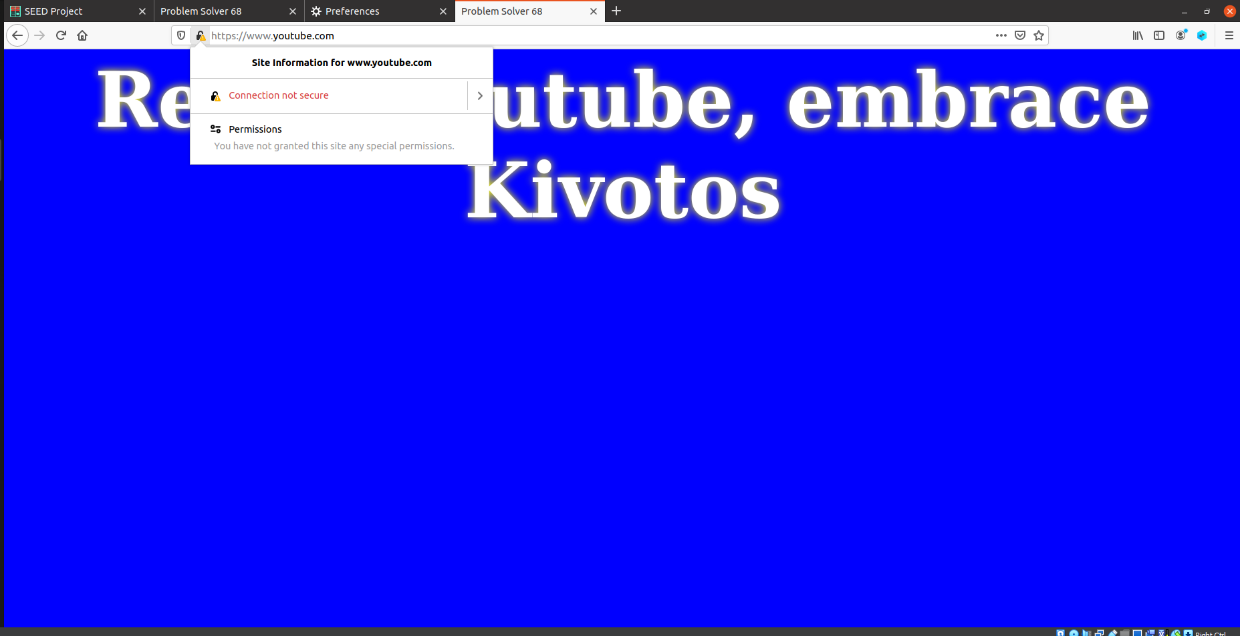
The site I chose to fake is [www.youtube.com](http://www.youtube.com) .

Fake DNS ssl configuration

****

Also add youtube.com to host for DNS polluting

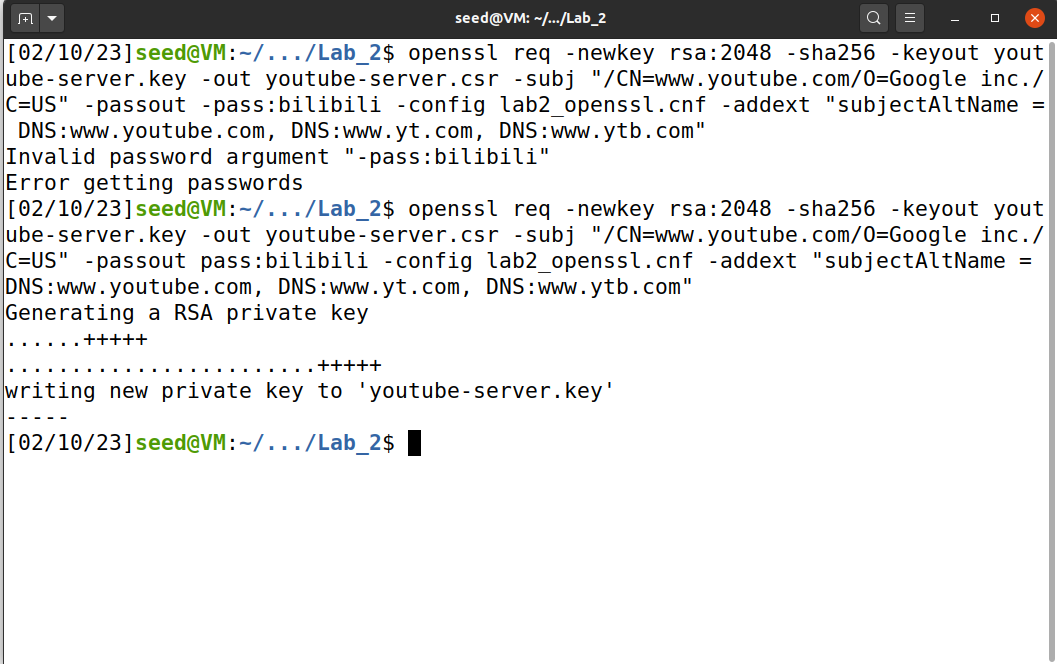


Youtube.com now redirects to my own website. However, the browser always shows unsecured connection as the certificate is absent or mismatching.

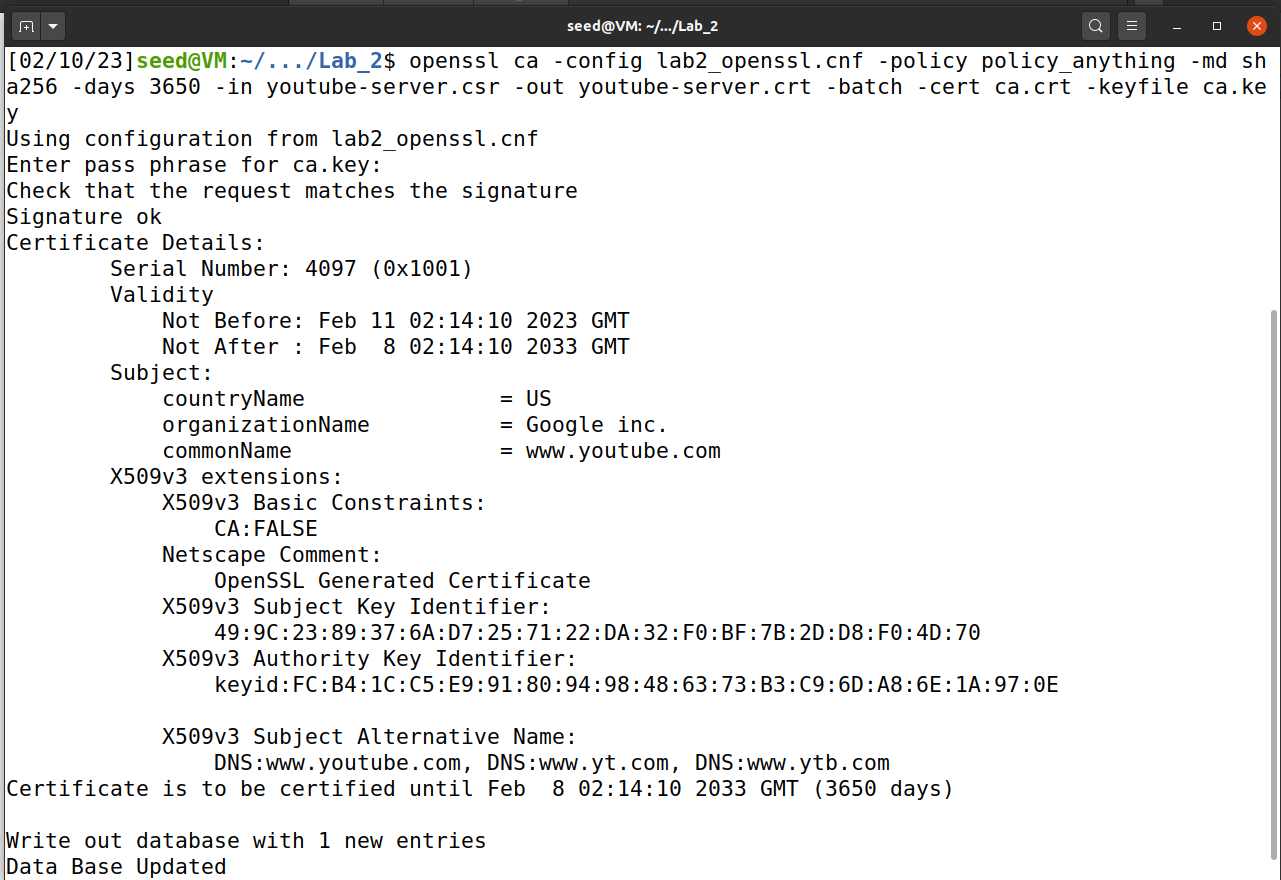
**Task 6:**

To launch man-in-middle attack:

First, create a dummy server.csr as youtube-server.csr, youtube-server.key to impersonate Youtube’s certificate.

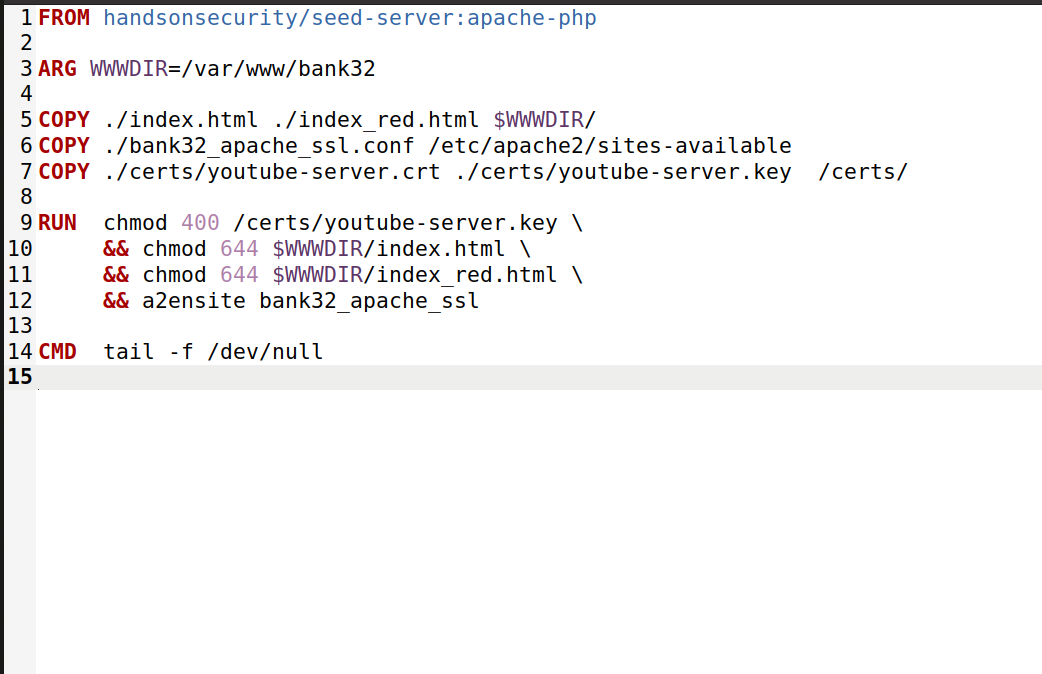


Then, sign the dummy csr using my own CA.



Next, modify Dockerfile and ssl configuration with its CA & key path to the dummy CA & key.

\*Maybe alter the /etc/apache2/sites-available/000-default.conf. In 000-default.conf, add entry virtual host 443 just like the given configuration from the lab shown in task4. But this time, the server name will be [www.youtube.com](http://www.youtube.com). I did it yet not sure about if this is the crucial part.

.

Finally, do $docker-compose build and ‘up’ the docker so that new configuration will be utilized. Enable apache2 service. This time the website will be your own but impersonating [www.youtube.com](http://www.youtube.com), and the browser will not raise awareness.

Definitely Youtube

