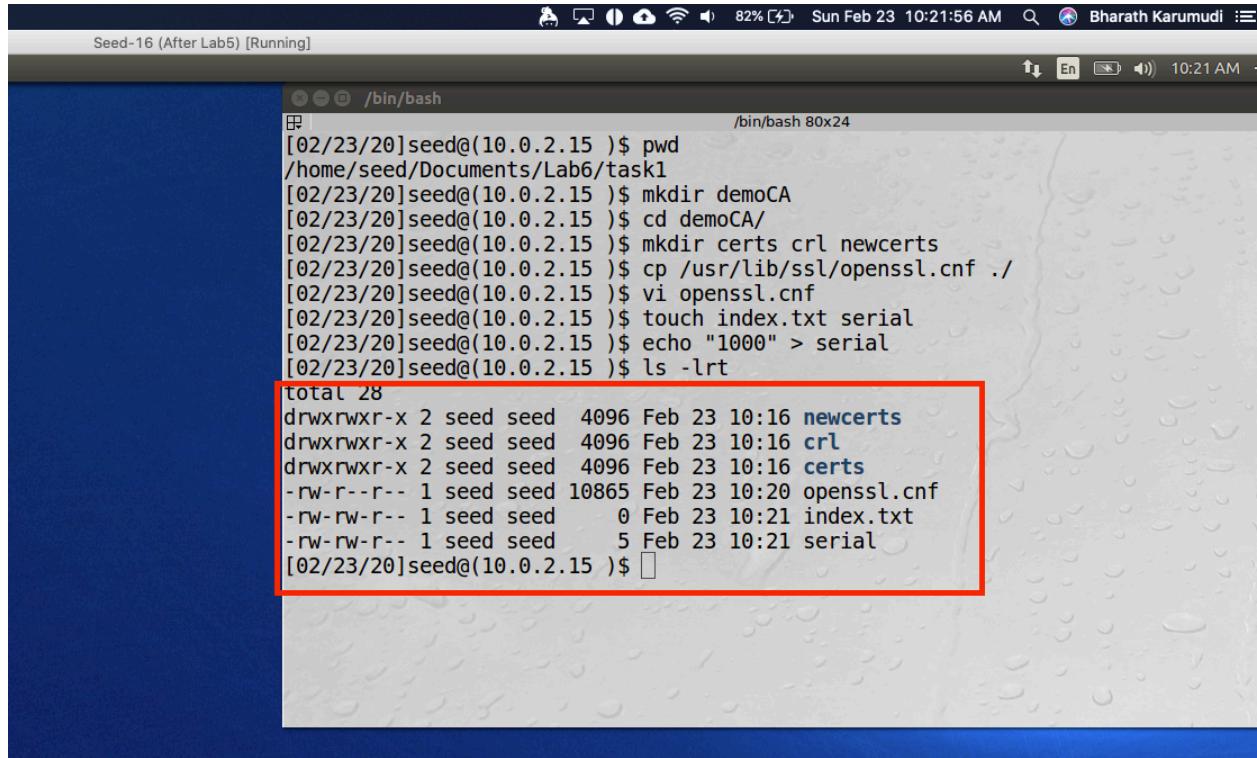


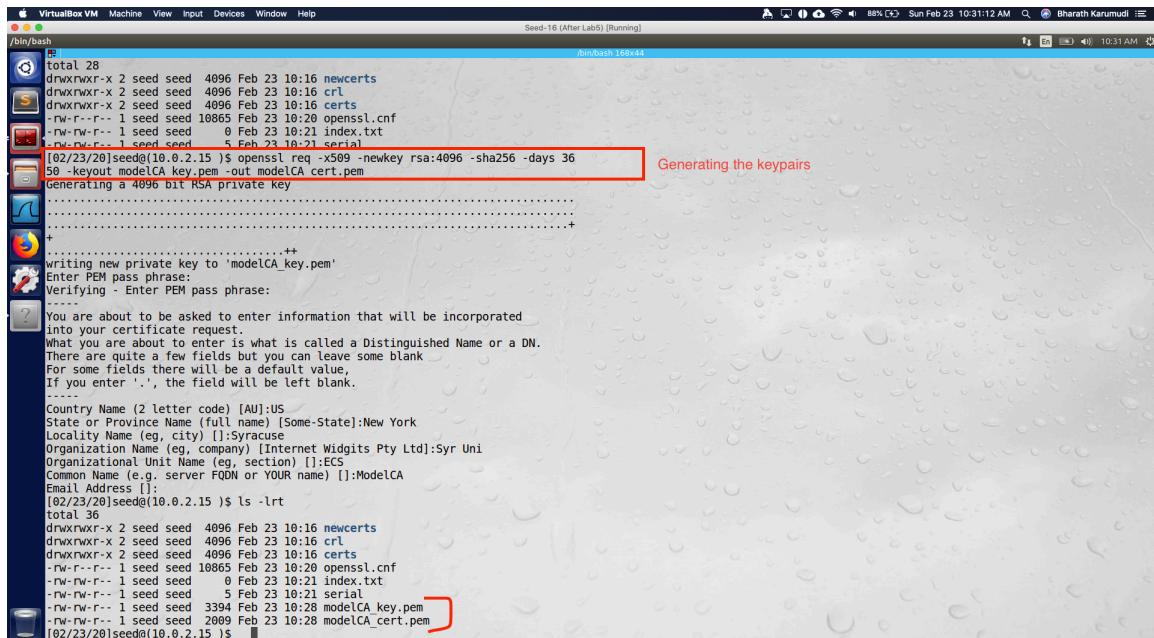
**Name:** Bharath Karumudi  
**Lab:** Public Key Infrastructure

## Task 1: Becoming a Certificate Authority (CA)



```
[02/23/20]seed@(10.0.2.15)$ pwd
/home/seed/Documents/Lab6/task1
[02/23/20]seed@(10.0.2.15)$ mkdir demoCA
[02/23/20]seed@(10.0.2.15)$ cd demoCA/
[02/23/20]seed@(10.0.2.15)$ mkdir certs crt newcerts
[02/23/20]seed@(10.0.2.15)$ cp /usr/lib/ssl/openssl.cnf ./
[02/23/20]seed@(10.0.2.15)$ vi openssl.cnf
[02/23/20]seed@(10.0.2.15)$ touch index.txt serial
[02/23/20]seed@(10.0.2.15)$ echo "1000" > serial
[02/23/20]seed@(10.0.2.15)$ ls -lrt
total 28
drwxrwxr-x 2 seed seed 4096 Feb 23 10:16 newcerts
drwxrwxr-x 2 seed seed 4096 Feb 23 10:16 crt
drwxrwxr-x 2 seed seed 4096 Feb 23 10:16 certs
-rw-r--r-- 1 seed seed 10865 Feb 23 10:20 openssl.cnf
-rw-rw-r-- 1 seed seed 0 Feb 23 10:21 index.txt
-rw-rw-r-- 1 seed seed 5 Feb 23 10:21 serial
[02/23/20]seed@(10.0.2.15)$
```

Fig1: Created the default directories and modified openssl.cnf file for demoCA



```
[02/23/20]seed@(10.0.2.15)$ openssl req -x509 -newkey rsa:4096 -sha256 -days 36
50 -keyout modelCA.key.pem -out modelCA.cert.pem
Generating the keypairs
Generating a 4096 bit RSA private key
.
.
.
writing new private key to 'modelCA.key.pem'
Enter PEM pass phrase:
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
.
.
.
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]:US
State or Province Name (full name) [Some-State]:New York
Locality Name (eg, city) []:Syracuse
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Syr Uni
Organizational Unit Name (eg, section) []:ECS
Common Name (e.g. server FQDN or YOUR name) []:ModelCA
Email Address []:
[02/23/20]seed@(10.0.2.15)$ ls -lrt
total 36
drwxrwxr-x 2 seed seed 4096 Feb 23 10:16 newcerts
drwxrwxr-x 2 seed seed 4096 Feb 23 10:16 crt
drwxrwxr-x 2 seed seed 4096 Feb 23 10:16 certs
-rw-r--r-- 1 seed seed 10865 Feb 23 10:20 openssl.cnf
-rw-rw-r-- 1 seed seed 0 Feb 23 10:21 index.txt
-rw-rw-r-- 1 seed seed 5 Feb 23 10:21 serial
-rw-rw-r-- 1 seed seed 3394 Feb 23 10:28 modelCA.key.pem
-rw-rw-r-- 1 seed seed 2089 Feb 23 10:28 modelCA.cert.pem
[02/23/20]seed@(10.0.2.15)$
```

Fig2: Generated the keypairs and self-signed cert for the ModelCA

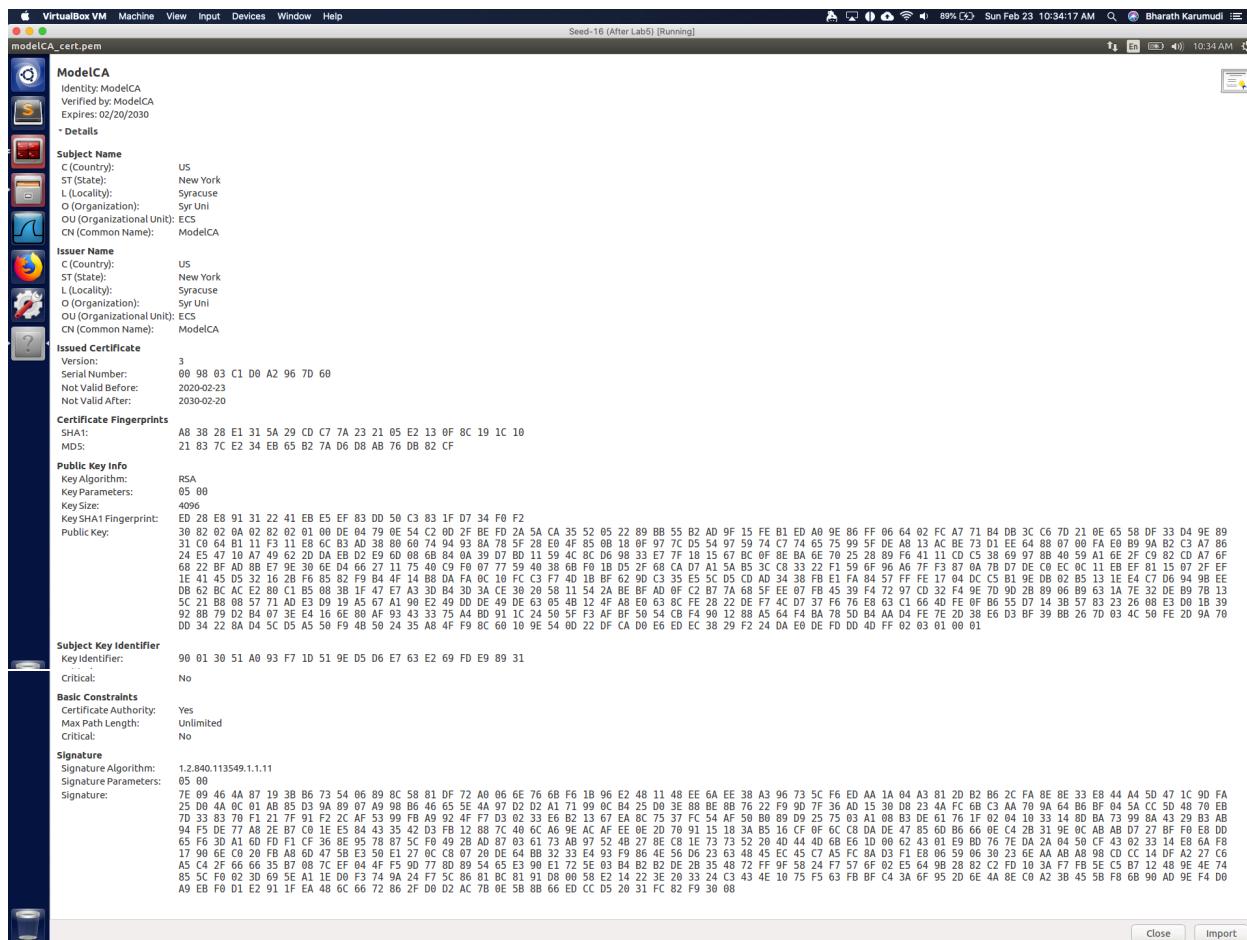


Fig3: Self Signed certificate of ModelCA

**Observation:** A Certificate Authority was established of name ModelCA and generated the public and private keys for the ModelCA and the self-signed public key certificate and became the root CA to vouch the other entities.

**Explanation:** Created the required default directories for the CA as shown in Fig1 and then generated the public and private key as shown in Fig2 for the ModelCA along with the self-signed public key certificate (shown in Fig3). With these will now be able to issue the certificates to the other entities.

## Task 2: Creating a Certificate for SEEDPKILab2018.com

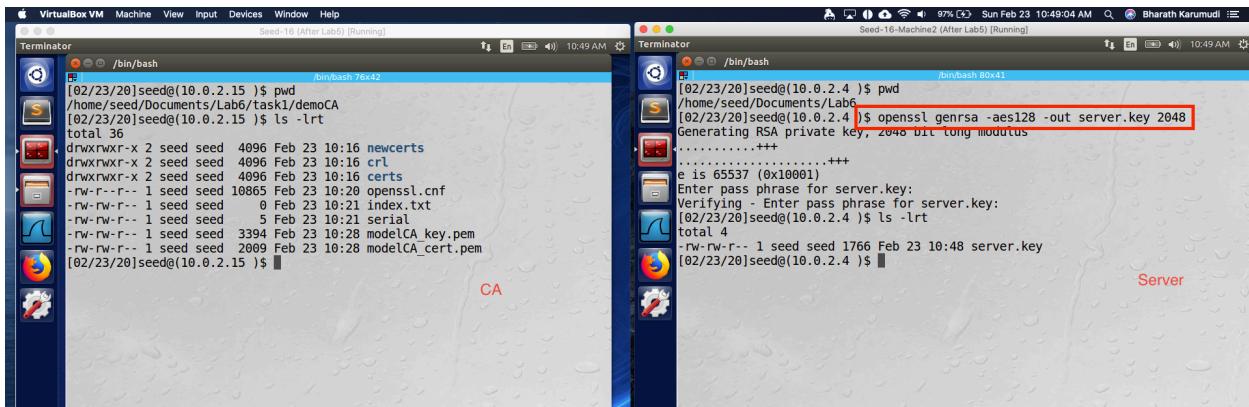


Fig1: Generated the public and private key pair for server

The image shows a single terminal window displaying the contents of the 'server.key' RSA private key file. The output includes the modulus (a large hex string), the private exponent (65537 or 0x10001), and the public exponent (3). The terminal window is titled 'Server' and is running on a Linux desktop environment.

```
Enter pass phrase for server.key:  
Verifying - Enter pass phrase for server.key:  
[02/23/20]seed@(...)$ ls -lrt  
total 4  
-rw-rw-r-- 1 seed seed 1766 Feb 23 10:48 server.key  
[02/23/20]seed@(...)$ openssl rsa -in server.key -text  
Enter pass phrase for server.key:  
Private-Key: (2048 bit)  
modulus:  
    00:b5:5d:80:a0:fe:9f:52:3c:54:49:f6:8a:fb:7d:  
    d5:0b:87:a4:13:63:cd:df:8e:89:26:1c:62:67:fb:  
    38:df:aa:85:40:b9:96:f7:27:c1:a1:5d:78:98:c3:  
    e5:20:94:df:53:9e:6e:77:1c:68:5f:e1:63:31:0d:  
    9c:f4:a5:c6:36:84:99:45:aa:88:74:22:02:70:55:  
    18:98:4f:de:69:10:87:c6:25:18:a3:60:db:b8:4b:  
    a7:0c:9d:5b:8a:a4:7a:f1:f8:3c:66:26:41:cc:73:  
    1e:4d:48:b7:35:c9:d5:9e:8e:6a:97:0b:ce:f0:98:  
    f0:12:63:7b:4c:81:95:64:53:55:d1:3e:cd:f9:46:  
    02:80:be:da:16:67:d5:81:5b:12:fb:e6:f9:5f:1b:  
    7d:18:05:38:55:71:7b:e1:5d:e5:46:65:c7:13:86:  
    8f:97:d9:ac:1d:84:ad:60:2d:21:fc:4e:e1:8c:29:  
    3d:6f:2d:74:e9:81:85:2c:29:f0:7f:0f:00:a9:f5:  
    23:f8:4c:1a:bc:ef:bc:86:52:08:1a:f2:61:6f:79:  
    01:15:af:78:a0:0e:81:cc:2b:b4:5b:ff:a0:ea:c1:  
    40:39:f9:a6:4a:84:c1:45:e9:a0:86:9a:83:74:1c:  
    4b:53:b0:b1:03:a0:25:eb:ba:c5:01:6b:55:95:87:  
    7d:e3  
publicExponent: 65537 (0x10001)  
privateExponent:
```

```

dc:a9:c4:24:29:dd:ff:4f:cd:8a:02:0a:a3:ee:19
b3:7d:1a:d8:9d:8f:73:2b:4f:3b:bf:74:96:f2:12:
bc:3a:b0:d1:64:6c:df:41:1e:48:e6:9c:09:43:36:
7d:73:9e:31:04:3c:30:8c:c4:40:d0:26:d7:d8:6f:
40:24:de:69:28:0c:3d:44:15:c5:0d:84:04:ed:bf:
51
prime1:
00:d9:e1:47:4b:54:dc:93:67:9c:04:ac:7f:a0:9d:
01:35:a1:fe:a4:54:c4:28:8a:51:87:9d:28:02:ae:
2d:a4:4e:22:aa:26:a9:a3:bc:c1:c9:79:87:eb:
24:91:25:6b:f5:df:71:fb:0d:d8:c4:e1:96:44:9e:
4c:8c:96:0b:2d:07:72:fe:f7:d3:91:7b:f4:19:01:
ae:ce:a3:4c:c5:c0:2c:1b:9e:3c:75:01:ba:81:a4:
1c:92:e8:a1:94:55:84:d9:9b:60:21:8f:54:41:0d:
e6:15:28:aa:bd:b9:4e:44:fb:67:f6:c6:1a:ee:9d:
08:9d:e5:1b:2c:73:f0:dc:29
prime2:
00:d5:18:bf:bd:d1:a9:4a:16:ca:0b:3a:7f:6d:9d:
8c:57:0b:df:63:bd:bf:dc:46:92:b1:c7:88:45:f4:
07:71:24:67:a5:fb:56:76:8b:f1:3c:a4:c0:02:21:
b6:1e:47:16:7d:a3:24:d7:3e:83:82:bc:bb:6c:e1:
9f:67:68:6d:50:7c:4c:f5:5c:67:ea:96:e2:84:6e:
32:91:24:4c:cd:1f:19:a5:4f:35:99:5d:63:a2:6a:
4f:ed:d1:41:f1:d2:94:f8:c8:e8:02:6b:c8:dd:f4:
35:9d:6c:25:a3:d8:c8:d8:8c:0f:19:dd:07:bc:06:
0f:a7:d5:27:13:e1:4e:cb:2b
exponent1:
00:d8:a2:1f:8b:0d:43:8b:e4:b6:c2:06:e9:97:3e:
c4:97:19:0c:e1:43:cf:0e:28:09:4c:77:bb:f2:7e:
50:af:4f:69:34:18:e1:14:e4:d2:a7:33:b5:fc:a8:
96:61:22:fe:27:28:16:62:72:b0:17:81:dd:31:38:
dc:0f:e4:98:14:1b:95:92:05:63:7d:01:ec:00:79:
25:e0:1b:8a:18:0d:0d:4e:00:95:48:19:81:e1:a8:
d5:28:54:09:63:6d:6c:7c:18:fb:0a:23:42:b8:b4:
23:12:e2:ae:71:99:af:f2:f2:ad:ba:35:db:4d:69:
21:ca:2c:a4:4a:1c:b5:5d:91
exponent2:
3d:8a:c3:3d:5f:c2:a3:6a:07:79:cc:b0:26:89:1c:
91:bb:15:0d:ea:2b:f1:c2:d3:f4:d1:ea:52:41:92:
b0:a3:51:e7:c6:b9:86:cd:0d:a3:74:f5:1a:e8:7f:
7f:44:c4:23:0e:ad:c2:18:04:c5:0a:1a:9a:ec:62:
2b:ec:25:bd:db:d8:21:f8:48:ee:a0:1a:d5:44:e2:
0f:b6:cb:13:66:4a:f2:c9:f8:71:cd:5d:cb:f7:26:
14:49:9d:0e:7:ef:2c:16:61:a2:e9:5b:fe:75:db:
d2:15:fc:9c:0e:02:2b:c9:16:51:fa:27:32:67:a3:dc:
7b:9c:fc:87:2a:47:a6:37
coefficient:
00:ce:c0:7b:24:e3:be:ba:3c:78:49:9b:78:a9:1e:
95:49:0f:c4:1f:01:18:8a:65:ba:af:9f:3b:d3:c9:
7c:73:e4:8f:2e:24:be:98:a8:13:21:69:9e:fa:ez:
41:76:da:51:c5:8d:6b:ec:3d:61:3b:d1:14:7d:a4:
db:b2:7d:45:59:56:2b:bc:98:23:c6:0f:13:4e:cb:
e1:50:65:76:ed:4d:4e:2b:2a:b5:49:fb:8f:71:69:
5c:25:bc:00:10:1a:74:ea:90:5d:57:f2:ea:bl:d0:
c8:b2:e2:a1:15:ec:34:44:77:47:94:17:d9:24:bf:
9c:7a:e9:65:5c:53:fd:5e:38
writing RSA key
-----BEGIN RSA PRIVATE KEY-----
MIIEpaIBAAKCAQEAteV2A0P6fUjxUsfaK+33VC4eKE2PWN346JJhxiz/s436qFQLmW
9yfBoV14mMPLIJTFu55SudxxoX+FjM02c9kXGNoSzRaqIdCicFUyMe/earChxiUY
o2DbuEunDj1biqR68fg8ZiBzHMeTu3Ncnno5qlw08JjwEmN7TIGvZFnv0t7N
+UYCgL7afMfvigVs+-+b5XXt9GAU4VXF74V3lRmxHE4aPl9msHyStCoh/E7hjk9
by106YGFCLnwfw8AqfuJ+Evaw0+BhLLYgVJhb3kBFa94a6bZCu0W/+g6sFA0fmm
SoTBRemghpqd8xL7CxA6A167rFWtV1Yd94wIDAQABoIBADZSHa2cVcG+ugwx
IK10odpD05xnHyLf0qrEg60Db0My+cSSkp/G842ZgZxeqTL02ebfE2ZcpVgWpxjp
F+KaQer58KJSEm6ejRbaF/FTarZ/0qKvrPnbozz/0jhoBrwAaaaBkC2UNC7zh9bj
+aPnQlNaxV0fVxHCNi.phSvrr/+9CP1bb0oM80jBqUUL3+vP8PL8LfcdCLRk1j6Z
byzyn153/exc0dT8iBfhrgHvZq30T9zfkTypxQ0pf9P2yoCgPuGb9NgTidj3Mr
Tzu/dJbyErw6sNFkbN9BHkjmaNLDNn1znjEEPDCMxEDQjtfyb0Ak3mk0DD3UfcuN
hAtv1EcgYEAE2eFHS17ck2ecBkx/oJ0BNah+pTEKIprh50a04tpE4iqaapqa08
wc15h+skKSv9d9x+w3yx0GWRj5MjYLLqdjyvFTkx0GuzoNMxcAsG548dQ66
gaQckuihIWE22tg1Y9UQ3mFSiqvb10Rptn9sYa7p0IneUbLPw3CkCgYEAIri/
vdGpshKCzp/b22Mvwfy72/3EaSscceIRFQhcSrnpftWdovxPKTAAlG2hqcWfaMk
1z60gry7b0GfZ2htUhM9Vxn6pbh0YksRMzR8ZpU81mV1jompP7dB8dKu+Mjo
Amv13f01nWl09j12IwPGd0HvAYPp9Un+E+F04ysCgYEAK1Fiv1D1+S2wgbplzE
1xkhM4UPDpigjTHe78n5Qr09nPbjhF0tSpz01/KiWYSL+JygWYnkWf4HdMTjCD+S
FBuVkgVjfqHsAhkL4BuKGA0tTgCV5BmBa4jVKFTZY21sfBj7C1nCuLojEuKucZmv
8VktujxbTwkhyykShy1ZECgY91sM9sXkjagd5LAmIRyRuxUN61vxxwP00epS
Q2Kwo1hnxrMz02jdpUaGh9/RM0j0q3CgATFChqa7Gr7CW929gh+EjuoBrVR01P
tsstZkruryfhxz3L9yUSZ3g5+8fM0i6V+ddvSFyc4cvFLH6jzJn09x7pH
KkenMwkBg0Q0whsK4766PHjh3iphvJ080fArkZkbqvnzvTyXx5zI8uJL6YqBmh
az764Kf22lHFjWvsPWE70RR9pNuyfUVZiu8mCPGDxN0y+FQZkbTU4rKrVJ+49x
avWlVAAGnTqkf1X8uoxM0iy4qoV7DREd0+uF9kv5x66VcU/1e0A==
-----END RSA PRIVATE KEY-----
[02/23/20]seed@[10.0.2.4 )$ 

```

Fig2: Showing the private key contents

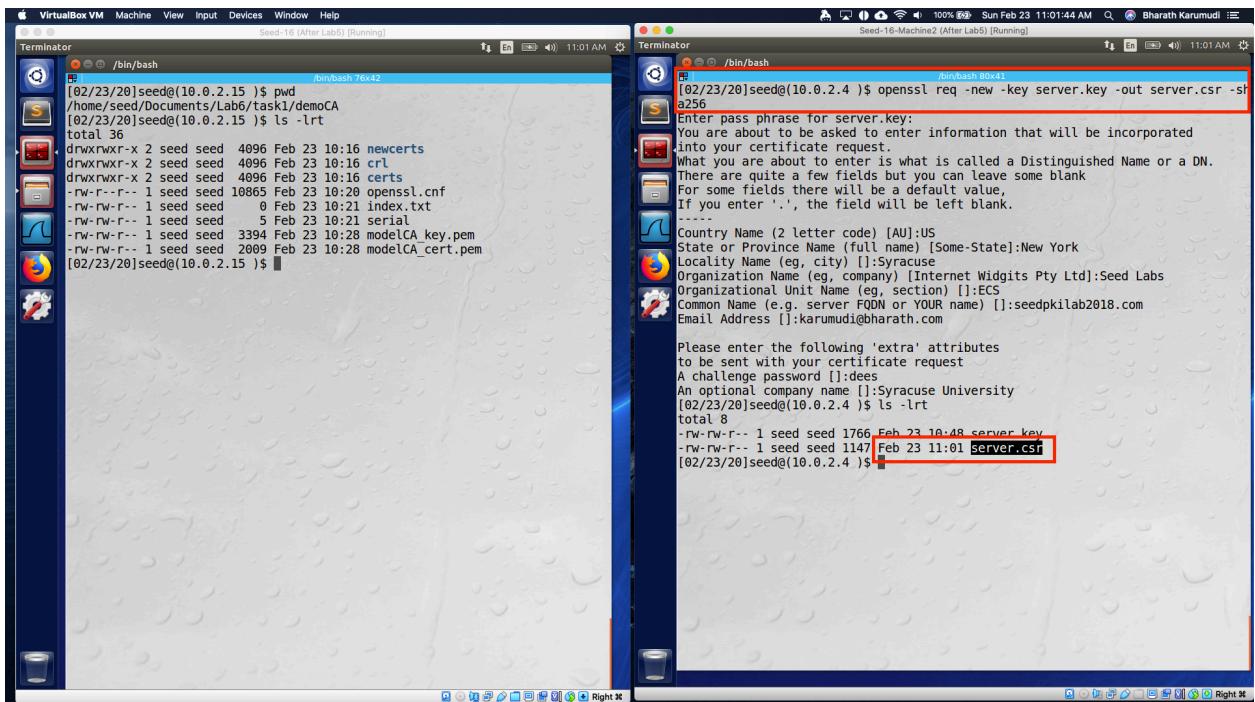


Fig3: Generated the Certificate Signing Request (csr)

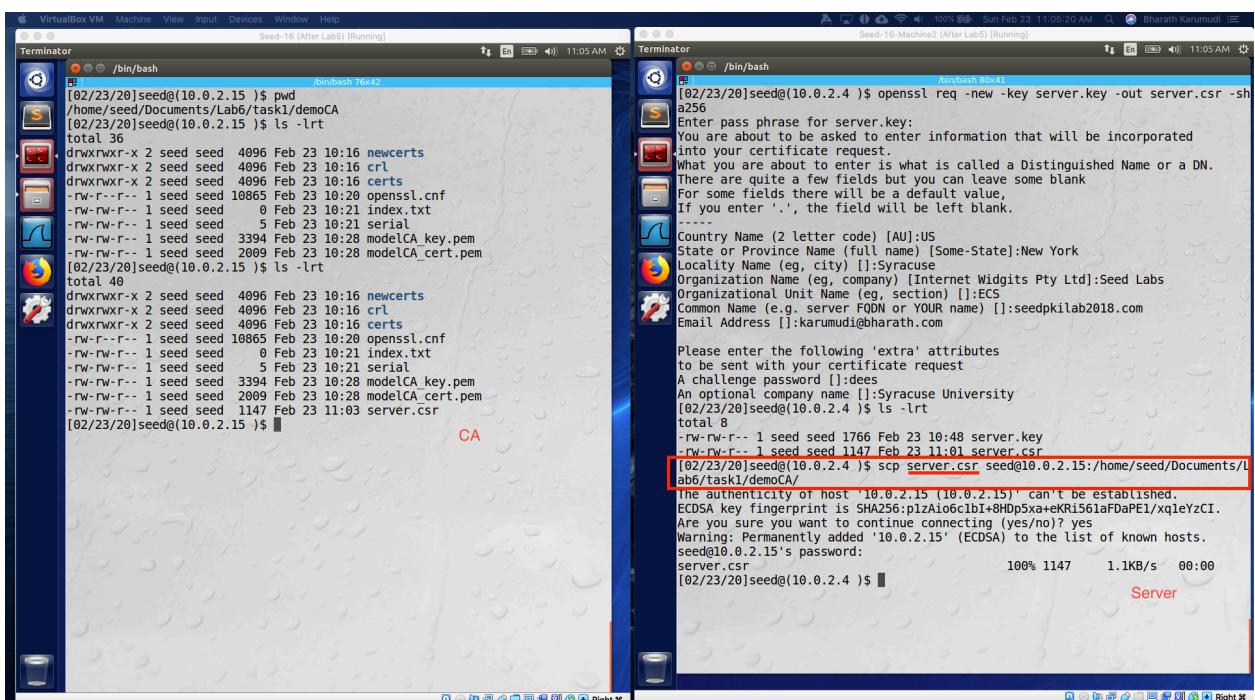


Fig4: Giving the csr to CA to sign over scp. In real-world it will be shared in their website.

The screenshot shows two terminal windows side-by-side, both titled "Terminator".

**Left Terminal (Seed-16 (After Lab5) [Running]):**

```

Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for modelCA.key.pem:
I am unable to access the ./demoCA/newcerts directory
./demoCA/newcerts: No such file or directory
[02/23/20]seed@(10.0.2.15 )$ openssl ca -in server.csr -out server.pem -md sha256 -cert modelCA.cert.pem -keyfile modelCA.key.pem -config openssl.cnf
Using configuration from openssl.cnf
Enter pass phrase for modelCA.key.pem:
Check that the request matches the signature
Signature ok
The organizationName field needed to be the same in the CA certificate (Syr Uni) and the request (Seed Labs)
[02/23/20]seed@(10.0.2.15 )$ vi openssl.cnf
[02/23/20]seed@(10.0.2.15 )$ openssl ca -in server.csr -out server.pem -md sha256 -cert modelCA.cert.pem -keyfile modelCA.key.pem -config openssl.cnf
using configuration from openssl.cnf
Enter pass phrase for modelCA.key.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
    Serial Number: 4096 (0x1000)                               CA
    Validity
        Not Before: Feb 23 16:11:43 2020 GMT
        Not After : Feb 22 16:11:43 2021 GMT
    Subject:
        countryName          = US
        stateOrProvinceName   = New York
        localityName         = Syracuse
        organizationName     = Seed Labs
        organizationalUnitName = ECS
        commonName            = seedpkilab2018.com
        emailAddress         = karumudi@bharath.com
X509v3 extensions:
    X509v3 Basic Constraints:
        CA:FALSE
        Netscape Comment:
        OpenSSL Generated Certificate
    X509v3 Subject Key Identifier:
        7D:74:46:4E:AB:9B:6C:1B:8C:CB:86:47:75:12:CD:D1:24:BE:AE:32
    X509v3 Authority Key Identifier:
        keyid:90:01:30:51:A0:93:F7:D1:51:9E:D5:D6:E7:63:E2:69:FD:E9
[02/23/20]seed@(10.0.2.15 )$ ls -l
total 60
drwxrwxr-x 2 seed seed 4096 Feb 23 10:16 crl
drwxrwxr-x 2 seed seed 4096 Feb 23 10:16 certs
-rw-rw-r-- 1 seed seed  0 Feb 23 10:21 index.txt.old
-rw-rw-r-- 1 seed seed  5 Feb 23 10:21 serial.old
-rw-rw-r-- 1 seed seed 3394 Feb 23 10:28 modelCA.key.pem
-rw-rw-r-- 1 seed seed 2089 Feb 23 10:28 modelCA.cert.pem
-rw-rw-r-- 1 seed seed 1147 Feb 23 11:03 server.csr
-rw-r--r-- 1 seed seed 10868 Feb 23 11:11 openssl.cnf
-rw-rw-r-- 1 seed seed 5854 Feb 23 11:11 server.pem
-rw-rw-r-- 1 seed seed  5 Feb 23 11:11 serial
drwxrwxr-x 2 seed seed 4096 Feb 23 11:11 newcerts
-rw-rw-r-- 1 seed seed  21 Feb 23 11:11 index.txt.attr
-rw-rw-r-- 1 seed seed 134 Feb 23 11:11 index.txt
[02/23/20]seed@(10.0.2.15 )$ ls -l newcerts/
total 8
-rw-rw-r-- 1 seed seed 5854 Feb 23 11:11 1000.pem
[02/23/20]seed@(10.0.2.15 )$ 
```

**Right Terminal (Seed-16-Machine2 (After Lab5) [Running]):**

```

[02/23/20]seed@(10.0.2.4) $ 
```

A red box highlights the command "server.pem" in the left terminal's output.

Fig5: Signed the csr and generated the certificate for the server

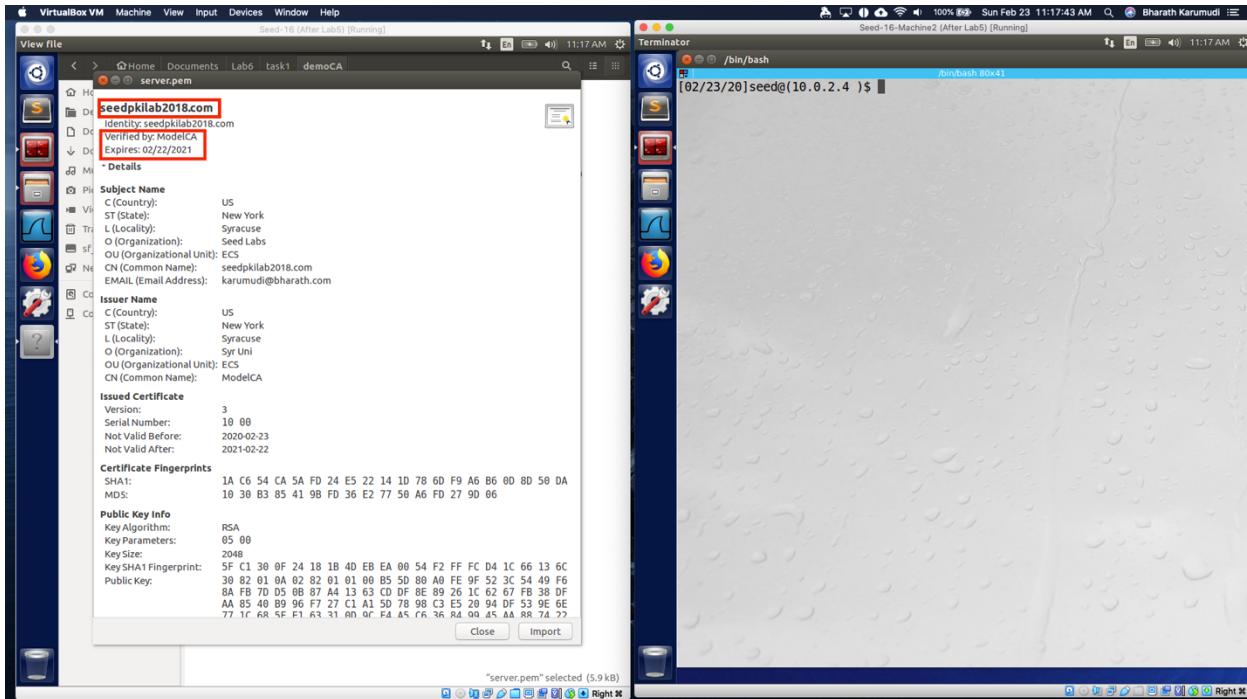


Fig6: View of the Signed and generated certificate by our ModelCA

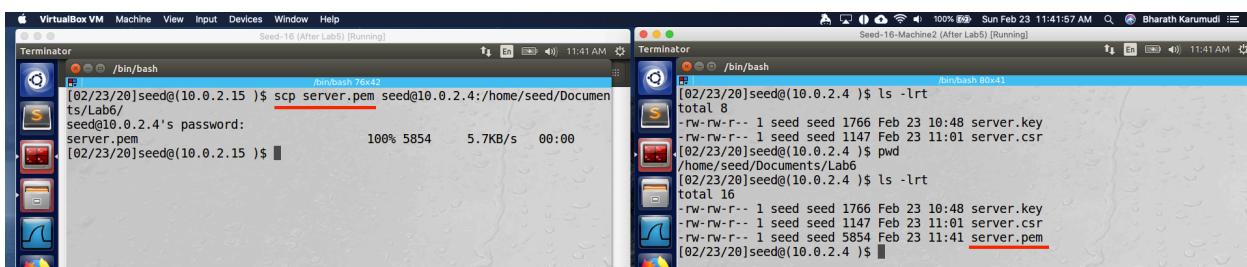


Fig7: Delivered the signed certificate to requestor (here over scp)

**Observation:** Generated the key pairs for the server and using them generated the Certificate Signing request for website (seedpkilab2018.com) and requested the ModelCA to sign and provide the Certificate.

**Explanation:** As a requestor, we first need the public and private key pair to establish a TLS/SSL connection on our website, so generated a keypair using openssl with RSA 2048 as shown in Fig1 and the generated private key is as shown in Fig2, it will have modulus, exponents and prime numbers that are used to generate the private key. Once the keypair is generated, created a Certificate Signing Request with all the details provided and sent to CA over scp as shown in Fig3 and Fig4.

As a CA, verified the requestor and signed the csr with ModelCA private key as shown in Fig 5 and issued a certificate to seedpkilab2018.com as shown in Fig6 and delivered to the requestor.

### Task 3: Deploying Certificate in an HTTPS Web Server

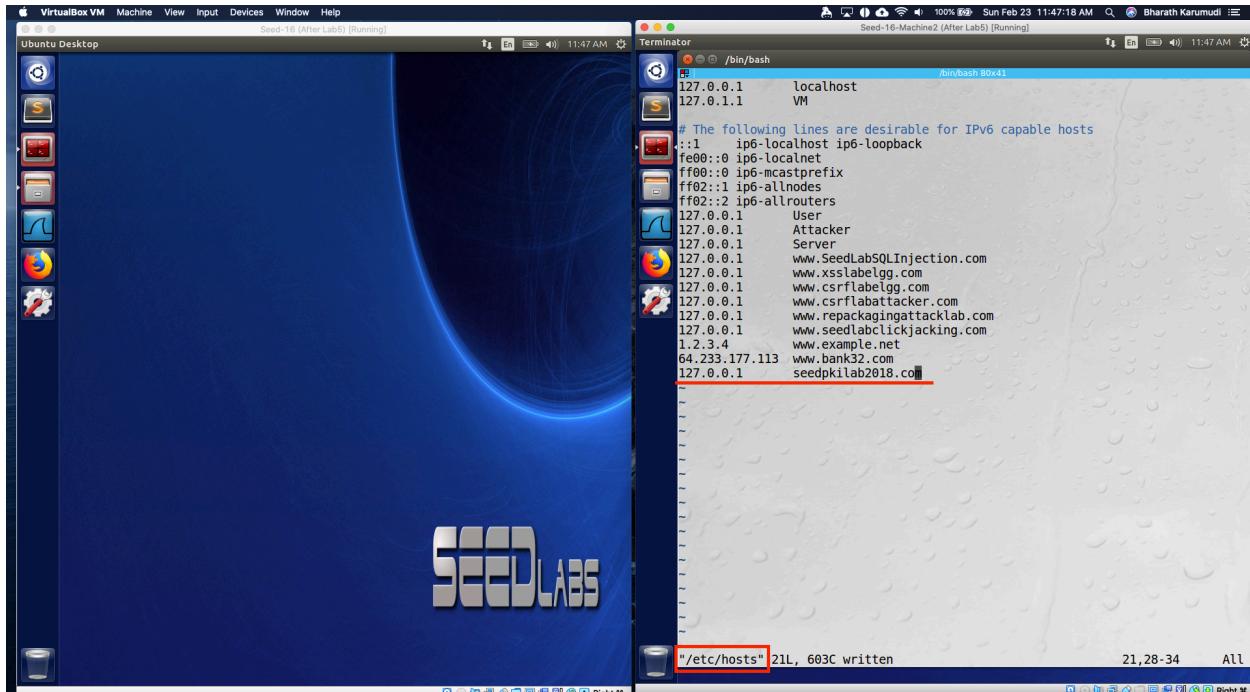


Fig1: Modified the /etc/hosts files by adding an entry for seedpkilab2018.com

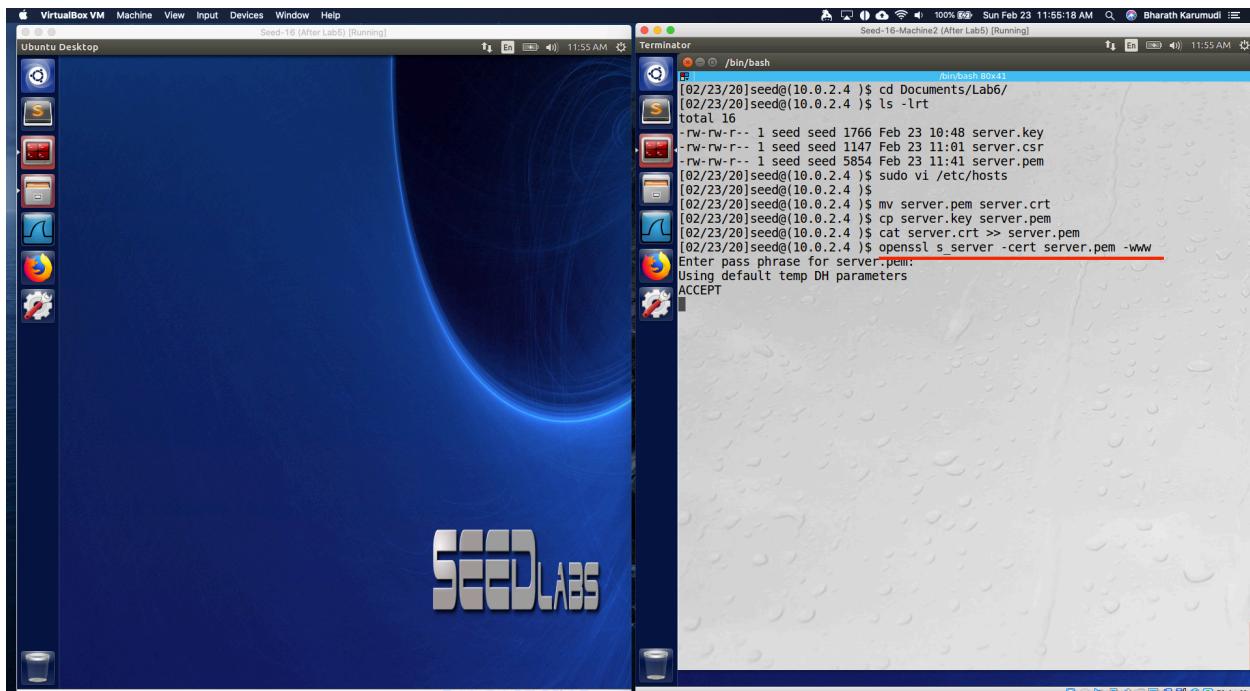


Fig2: Launching the server using openssl utility

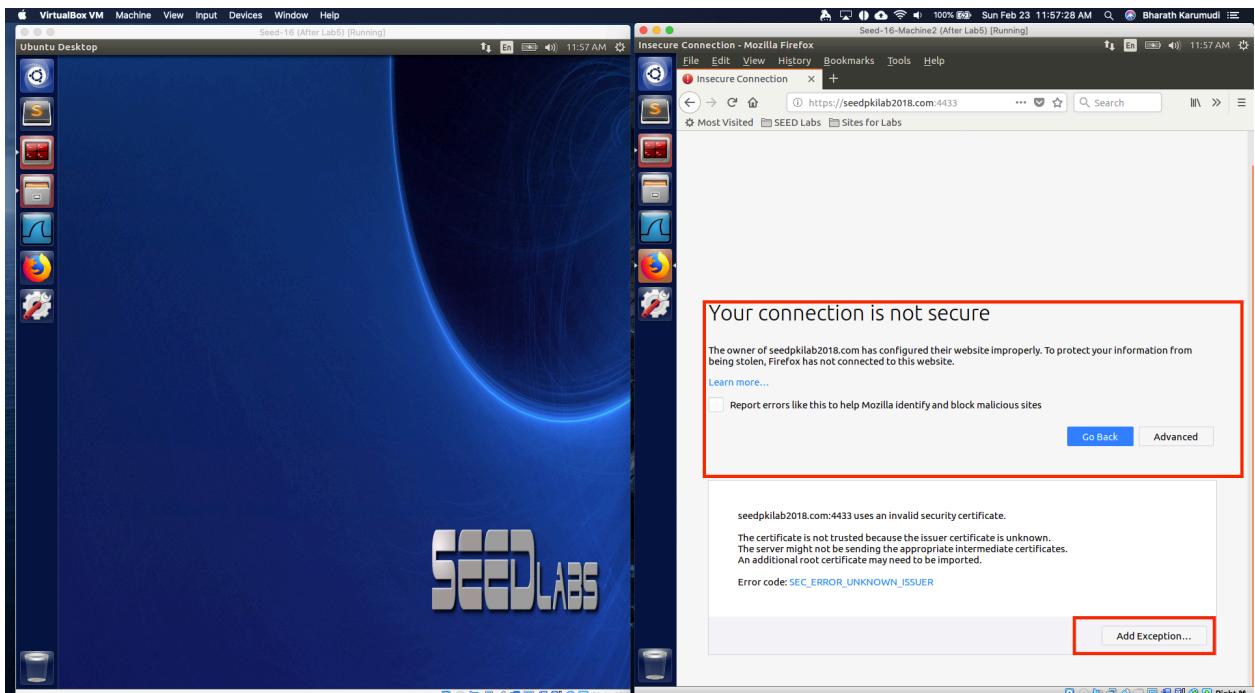


Fig3: Accessing the website using Firefox and can see the exceptions

Authority Name	Security Device
-ACCV	Builtin Object Token
ACCVRAIZ1	Builtin Object Token
-Adatis S.p.A./03358520967	Builtin Object Token
Adatis Authentication Root CA	Builtin Object Token
-AddTrust AB	Builtin Object Token
AddTrust Low-Value Services Root	Builtin Object Token
AddTrust External Root	Builtin Object Token
-AffirmTrust	Builtin Object Token
AffirmTrust Commercial	Builtin Object Token

Fig4: CA shared the cert (left) and Firefox Certificate manager (right)

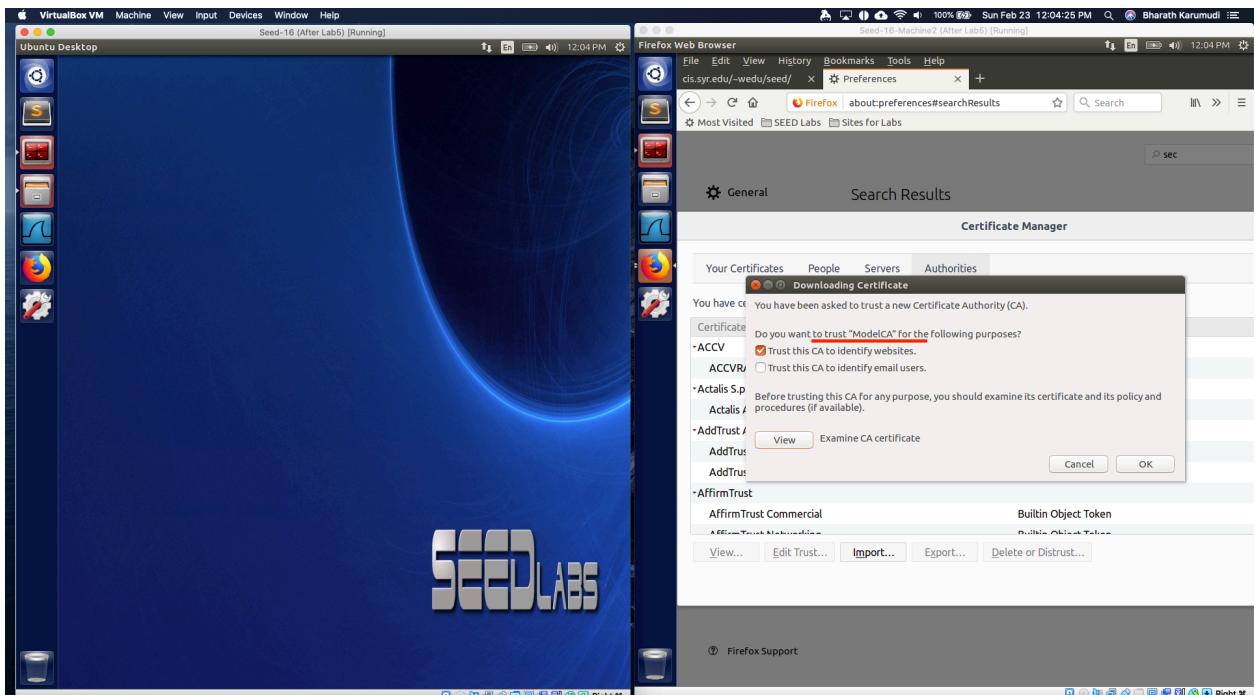


Fig5: Importing the ModelCA cert to Firefox

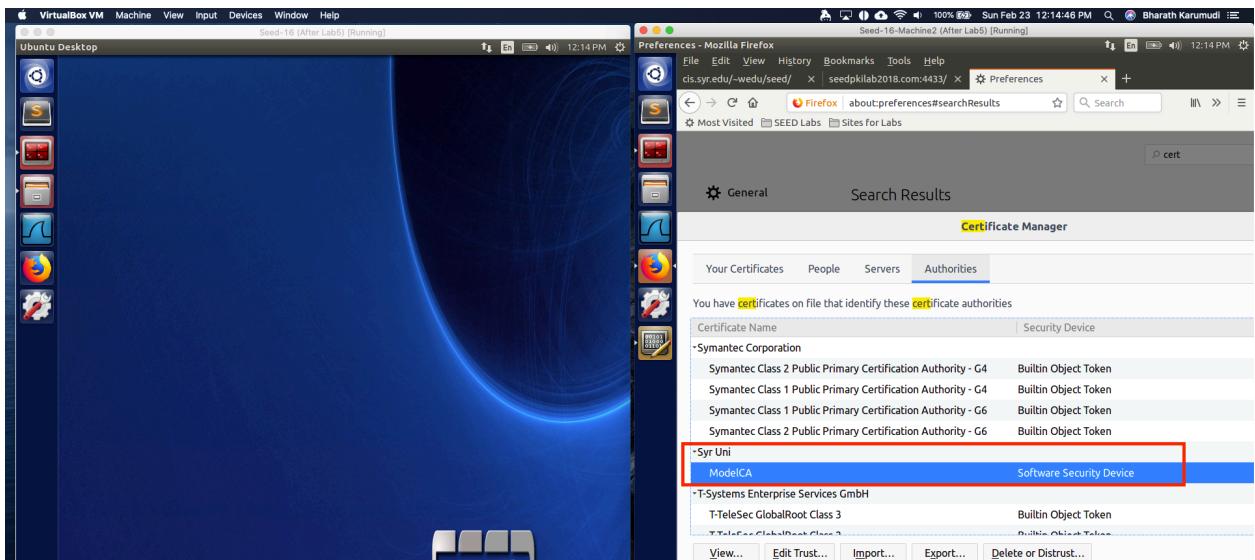


Fig6: Syr Uni ModelCA is now added to trusted CA

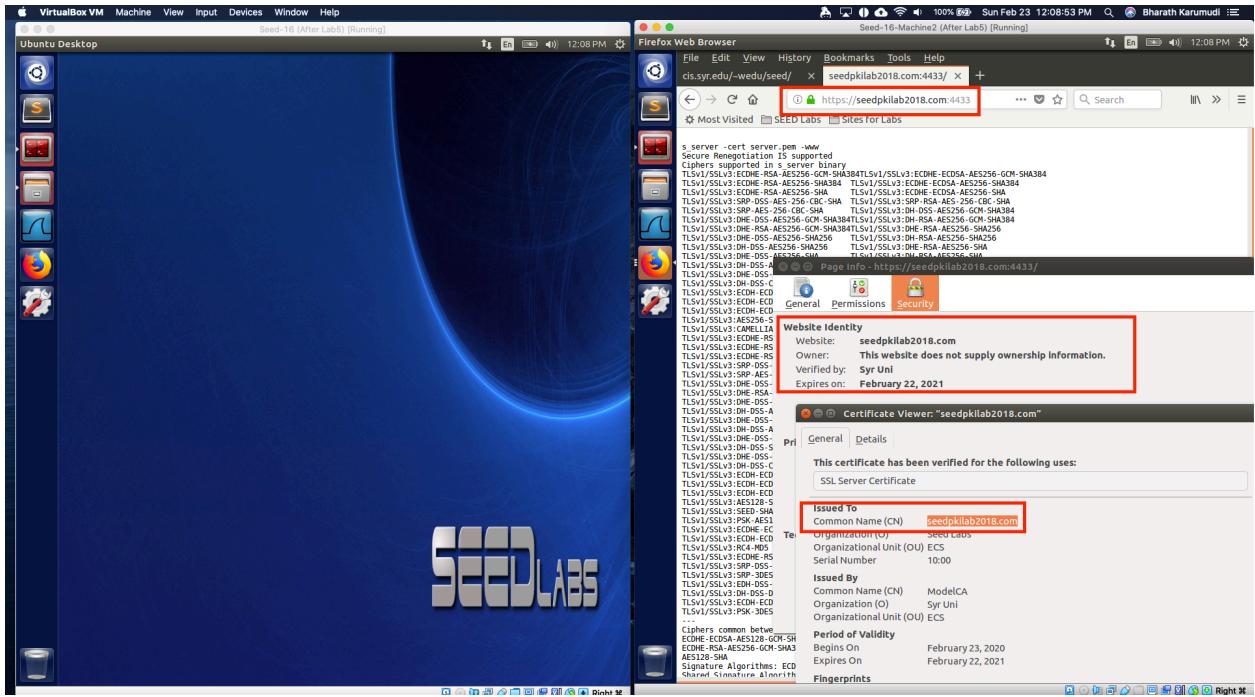


Fig7: The browser now trusted the website certificate

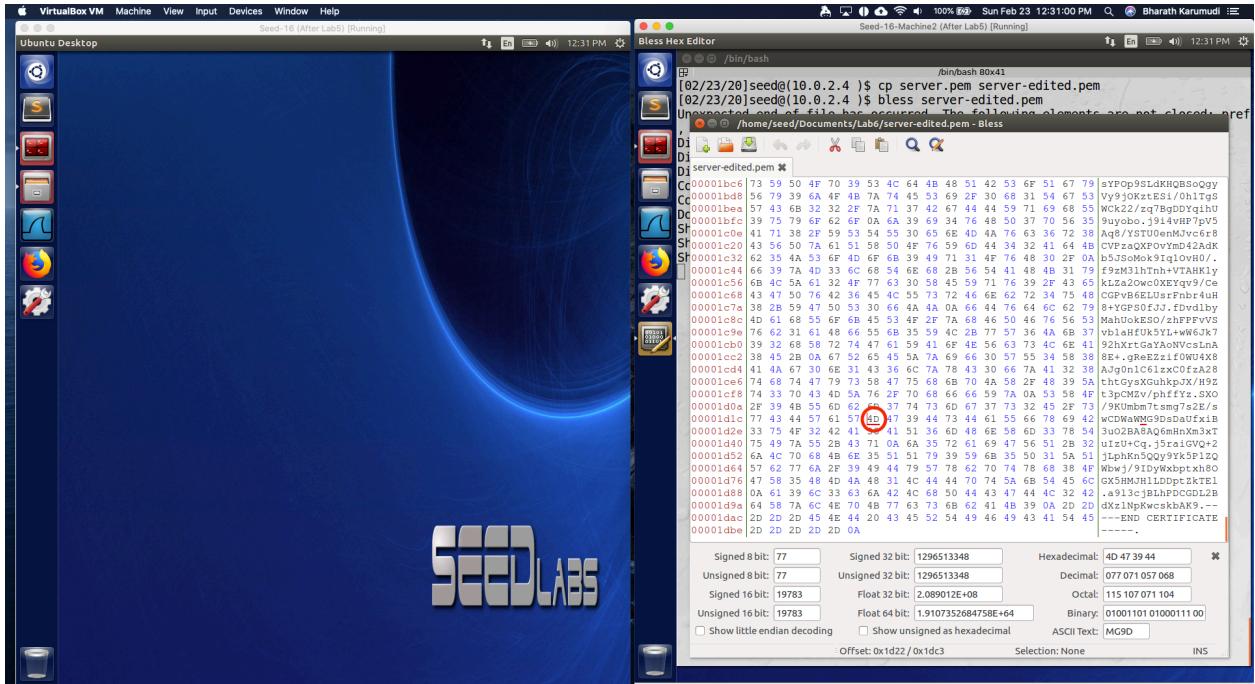


Fig8: Modifying the highlighted byte from 4D to 4F

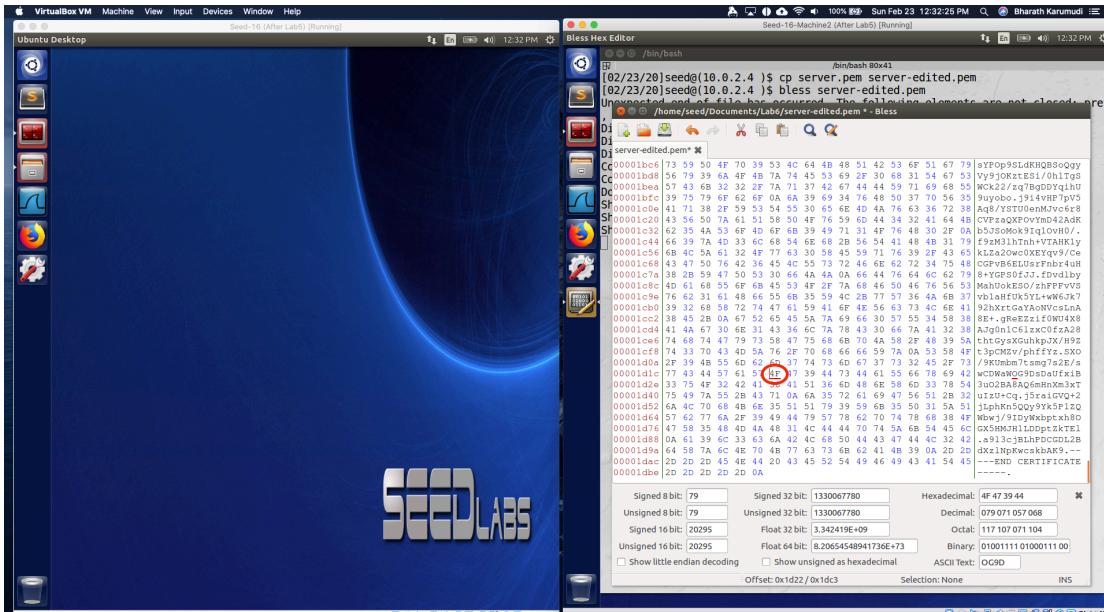


Fig9: Modified and Saved the cert as server-edited.pem

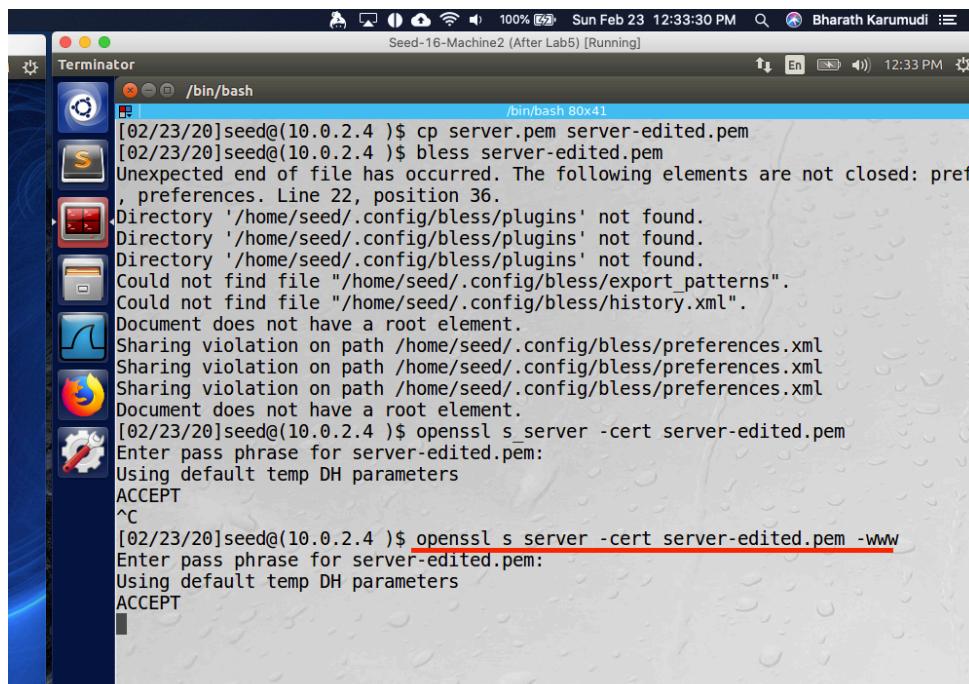


Fig10: Starting the server with modified cert



Fig11: Connection to the server failed with Invalid Signature

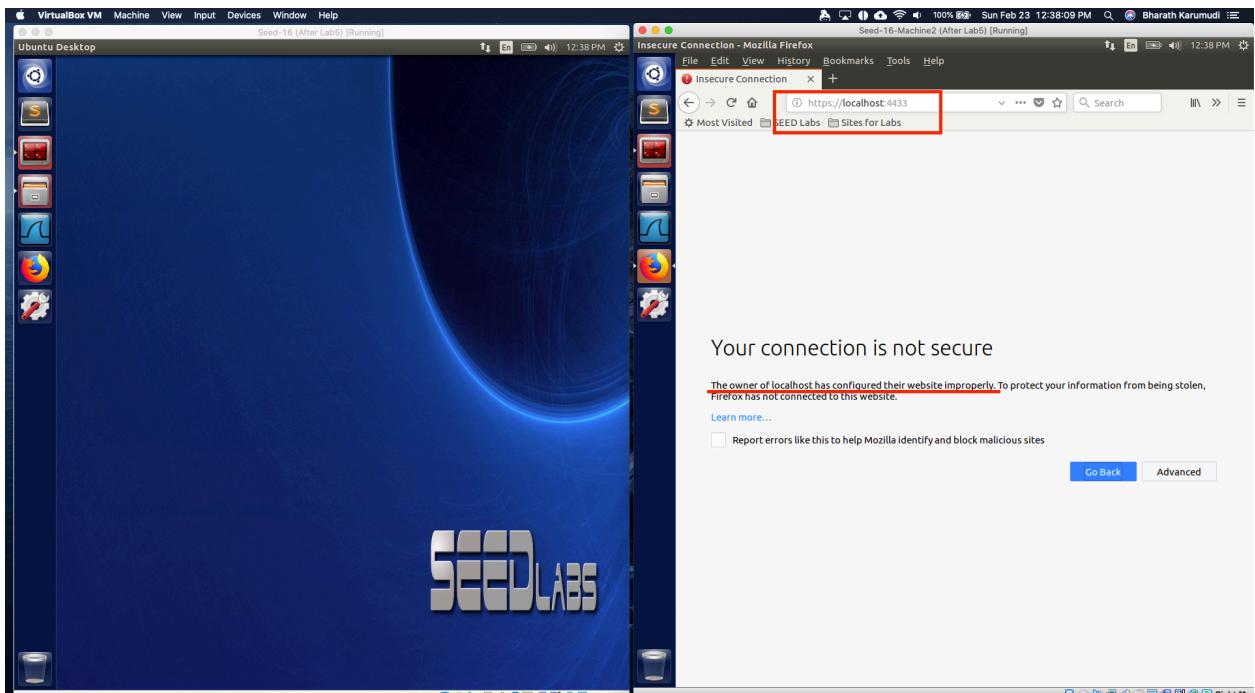


Fig12: Error when used on localhost instead of cert issued CN

**Observation:** The CA issued a signed certificate and then the certificate was used for the seedpkilab2018.com, but as the CA is not a default trusted one by Firefox, we have added the CA's certificate to the trust store. Once the CA certificate is added, the browser was able to trust the website certificate. Also, when modified the certificate by single byte the cert is no

longer valid and failed signature validation and also when used on localhost domain, it failed because the certificate was issued to a different Common Name (CN).

**Explanation:**

- Modified the /etc/hosts file and added an entry for the seedpkilab2018.com with localhost IP address.
- Received the signed certificate from CA and using openssl s\_server launched a server to debug and from the browser when access the site https:// seedpkilab2018.com:4433 the Firefox issued a warning as the signed CA is not known.
- Received the CA Self-signed certificate and then imported the cert to the Firefox trust store and once added the Firefox was able to able to accept the website certificate without any errors or warnings.
- Modified the server.pem by a single byte from 4D to 4F as shown in Fig8 and Fig9 and saved the cert as server-edited.pem and launched the openssl s\_server utility with the modified cert. When accessed the site from the browser, the Firefox was unable to validate the signature and failed to establish the connection (Fig 11).
- Also, when accessed the website with https://localhost:4433 domain, the Firefox failed to establish as the certificate is issued to different CN (Fig 12). This is a security measure to protect against Common Name attacks.

#### Task 4: Deploying Certificate in an Apache-Based HTTPS Website

```
drwxr-xr-x 2 root root 4096 Apr 27 2018 sites-available
[02/23/20]seed@(10.0.2.4 )$ cd sites-available/
[02/23/20]seed@(10.0.2.4 )$ ls -lrt
total 12
-rw-r--r-- 1 root root 6338 Apr  5 2016 default-ssl.conf
-rw-r--r-- 1 root root 2104 Apr 27 2018 000-default.conf
[02/23/20]seed@(10.0.2.4 )$ vu default-ssl.conf
vu: command not found
[02/23/20]seed@(10.0.2.4 )$ sudo vi default-ssl.conf Modifying the conf file
[02/23/20]seed@(10.0.2.4 )$ sudo vi default-ssl.conf
[02/23/20]seed@(10.0.2.4 )$ mkdir -p /var/www/seedpkilab
mkdir: cannot create directory '/var/www/seedpkilab': Permission denied
[02/23/20]seed@(10.0.2.4 )$ sudo !!
sudo mkdir -p /var/www/seedpkilab
[02/23/20]seed@(10.0.2.4 )$ cd /var/www/seedpkilab Site html files
[02/23/20]seed@(10.0.2.4 )$ sudo vi index.html
[02/23/20]seed@(10.0.2.4 )$ sudo apachectl configtest
AH00112: Warning: DocumentRoot [/var/www/seedlabclickjacking] does not exist
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
Syntax OK
[02/23/20]seed@(10.0.2.4 )$ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  service apache2 restart
[02/23/20]seed@(10.0.2.4 )$ sudo a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  service apache2 reload
[02/23/20]seed@(10.0.2.4 )$ sudo service apache2 restart Restarting the apache
Enter passphrase for SSL/TLS keys for seedpkilab2018.com:443 (RSA): ****
[02/23/20]seed@(10.0.2.4 )$
```

Fig1: Modifications to apache

# Added for seedpkilab2018.com by BharathKarumudi

```
Terminator /bin/bash /bin/bash 80x41
#      SSL close notify alert is send and mod_ssl waits for th
#      alert of the client. This is 100% SSL/TLS standard comp
#      practice often causes hanging connections with brain-de
#      this only for browsers where you know that their SSL im
#      works correctly.
#      Notice: Most problems of broken clients are also related to
#      keep-alive facility, so you usually additionally want to dis
#      keep-alive for those clients, too. Use variable "nokeepalive"
#      Similarly, one has to force some clients to use HTTP/1.0 to
#      their broken HTTP/1.1 implementation. Use variables "downgra
#      "force-response-1.0" for this.
#      BrowserMatch "MSIE [2-6]" \
#                  nokeepalive ssl-unclean-shutdown \
#                  downgrade-1.0 force-response-1.0

</VirtualHost>

# Added for seedpkilab2018.com by BharathKarumudi
<VirtualHost *:443>
    ServerName seedpkilab2018.com
    DocumentRoot /var/www/seedpkilab
    DirectoryIndex index.html

    SSLEngine On
    SSLCertificateFile /etc/apache2/ssl/server.crt
    SSLCertificateKeyFile /etc/apache2/ssl/server.key
</VirtualHost>

</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
146,1 Bot
```

Fig2: default-ssl.conf file modifications

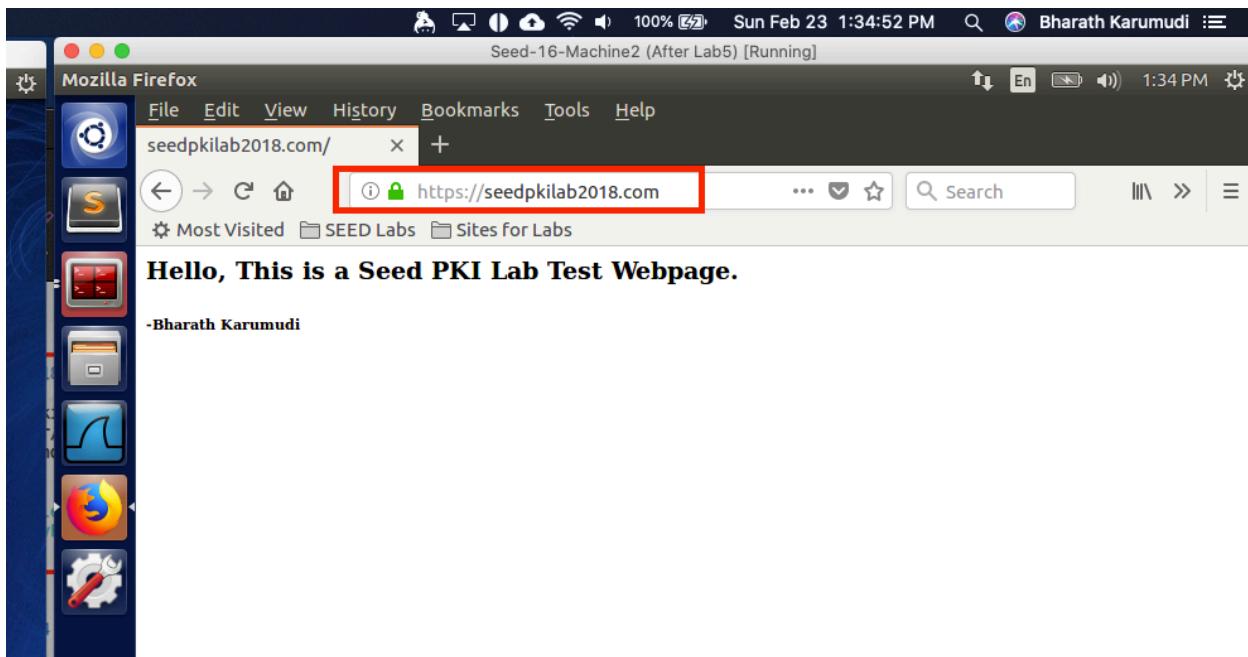


Fig3: Website running on SSL and apache

**Observation:** Modified the apache server to host the new website seedpkilab2018.com and configured the SSL details in default-ssl.conf file. Once restarted the apache, the site is now running on HTTPS without content.

**Explanation:**

1. Modified the default-ssl.conf file under /etc/apache2/sites-enabled/ to add the new virtualhost to listen on port 443. Also configured the site private key and CA signed certificate as shown in Fig1 and Fig2.
2. Created the site home as /var/www/seedpkilab/ and created a index.html
3. Ran the following commands to test the configuration and also to enable the SSL modules:
  - a. sudo apachectl configtest
  - b. sudo a2enmod ssl
  - c. sudo a2ensite default-ssl
4. Finally restarted the apache service to make the configurations effective and on restart apache prompted the site private key (server.key) password and entered and started the service.
5. From browser accessed the <https://seedpkilab2018.com> and the site came up on SSL with our content (Fig3). As the CA self-signed cert is already on the Firefox trust store, the Firefox didn't issue any warnings or errors. The connection now uses encrypted communications between Client and Server.

Thus, configured the CA issued Certificate on Apache Webserver.