X.509 certificates for user authentication in MongoDB

specific to windows 10

Reference: A similar tutorial is available in the MongoDB documentation:
https://docs.mongodb.com/manual/tutorial/configure-x509-client-authentication/

Prep-work

We'd need to install OpenSSL for windows first.

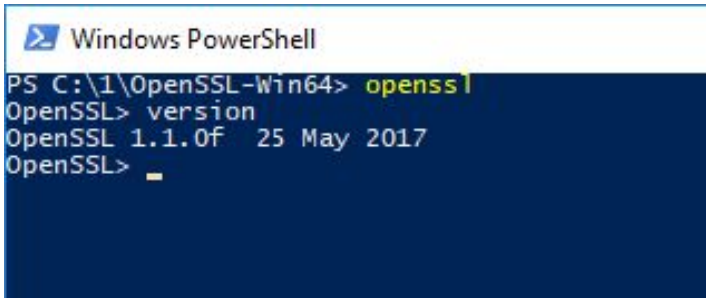
OpenSSL does not provide a binary for windows, however one is available from:

https://slproweb.com/products/Win32OpenSSL.html

We'll use the 64-bit version listed towards the bottom of this page. Look for the words "Win64 OpenSSL" and download the full package (not the 'lite' or 'light' package).

1. Install the downloaded package (you may need administrative privileges).
2. Once installed, open a command window and type:

```
openssl
```



3. You should see the openssl prompt - this means the install worked successfully.

Overview

This tutorial uses MongoDB version 3.4

The full exercise will be done in 3 steps:

1. Creation of a self-signed x.509 certificate
2. Addition of a user (with the same credentials as the x.509 cert)
3. Logging in into mongo using the certificate

Creating a self-signed x.509 certificate

Typically a certificate would be provided by a certifying authority, however for the purposes of development, we'll create our own i.e. self-signed certificate.

Creating the x.509 certificate requires the following steps:

1. Configure OpenSSL to create mongodb friendly certificates
2. Create the root private key and certificate
3. Create the public certificate signing request
4. Create the certificate
5. Merge the certificate and the private key into a single pem file
6. Validate the certificate to make sure everything is fine

Let's carry these out step-by-step:

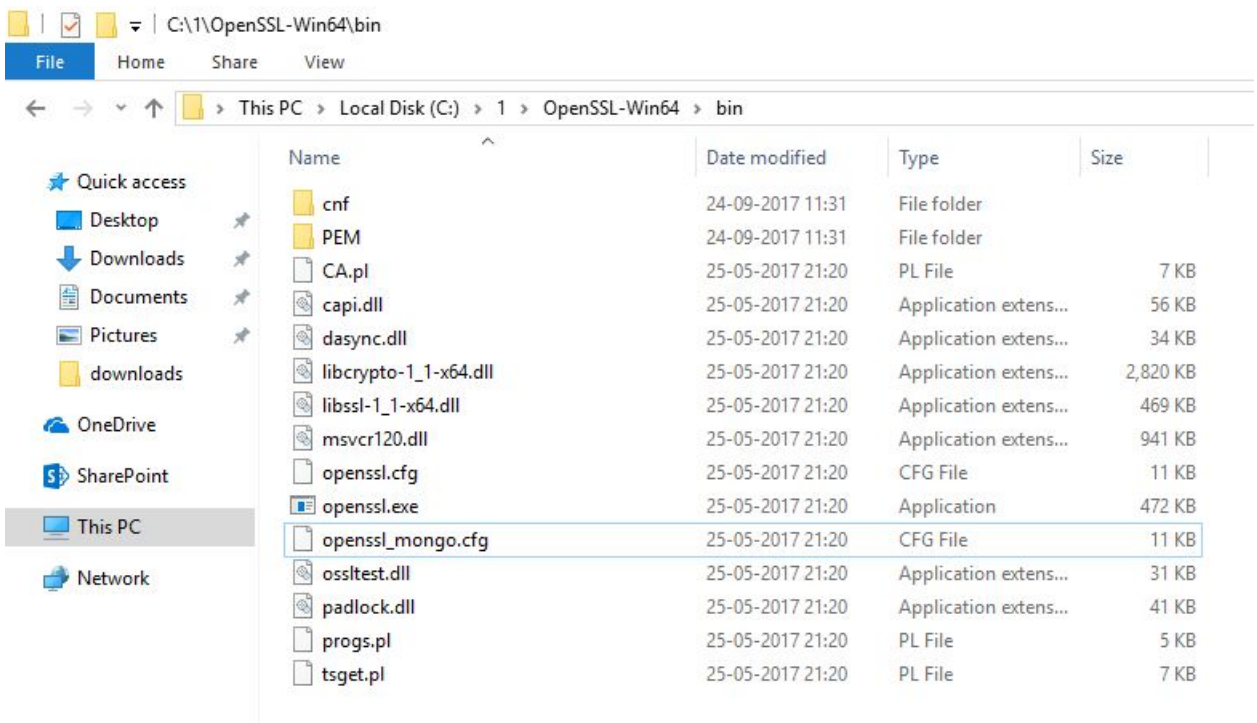
Configure OpenSSL

For MongoDB to work with x.509 certificates, the following conditions need to be satisfied:

- A single Certificate Authority (CA) must issue the certificates for both the client and the server.
- Client certificates must contain the following fields:
  - `keyUsage = digitalSignature`
  - `extendedKeyUsage = clientAuth`
- Each unique MongoDB user must have a unique certificate.

There's other conditions too but this is our focus for now. Let's set keyUsage and extendedKeyUsage fields.

0. Optional: You may want to add `[OpenSSL install location]\bin` to your Environment Variables' Path variable. For e.g. in my case OpenSSL is installed in `C:\1\OpenSSL-Win64` so I have added `C:\1\OpenSSL-Win64\bin` to my Path.
1. Navigate to the directory where OpenSSL is installed, go into the folder called 'bin' and make a copy of `openssl.cfg`. Call this `openssl_mongo.cfg`.



2. Edit openssl_mongo.cfg and search for the configuration section titled:

   `[ usr_cert ]`

   Add the following lines in this section:

   ```
   # added for mongoDB
   keyUsage = keyCertSign, digitalSignature
   extendedKeyUsage = clientAuth, serverAuth
   ```

```
[ usr_cert ]

# These extensions are added when 'ca' signs a request.

# This goes against PKIX guidelines but some CAs do it and some software
# requires this to avoid interpreting an end user certificate as a CA.

basicConstraints=CA:FALSE

# added for mongoDB
keyUsage = keyCertSign, digitalSignature
extendedKeyUsage = clientAuth, serverAuth
```

3. In the sections titled:

   `[ v3_ca ]` and

   `[ v3_req ]`

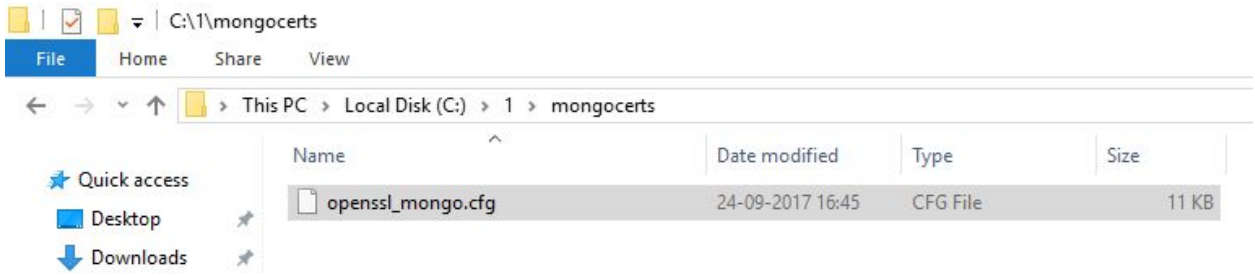
   Add the following lines:

   `# added for mongoDB`

   `extendedKeyUsage = clientAuth, serverAuth`

```
[ v3_req ]

# Extensions to add to a certificate request
# added for mongoDB
extendedKeyUsage = clientAuth, serverAuth

basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment

[ v3_ca ]


# Extensions for a typical CA
# added for mongoDB
extendedKeyUsage = clientAuth, serverAuth
```

4. Save & close `openssl_mongo.cfg` and move it to a directory where we'll create our certificates. In my case I moved it to a folder called mongocerts



You are now ready to start making the certificates.

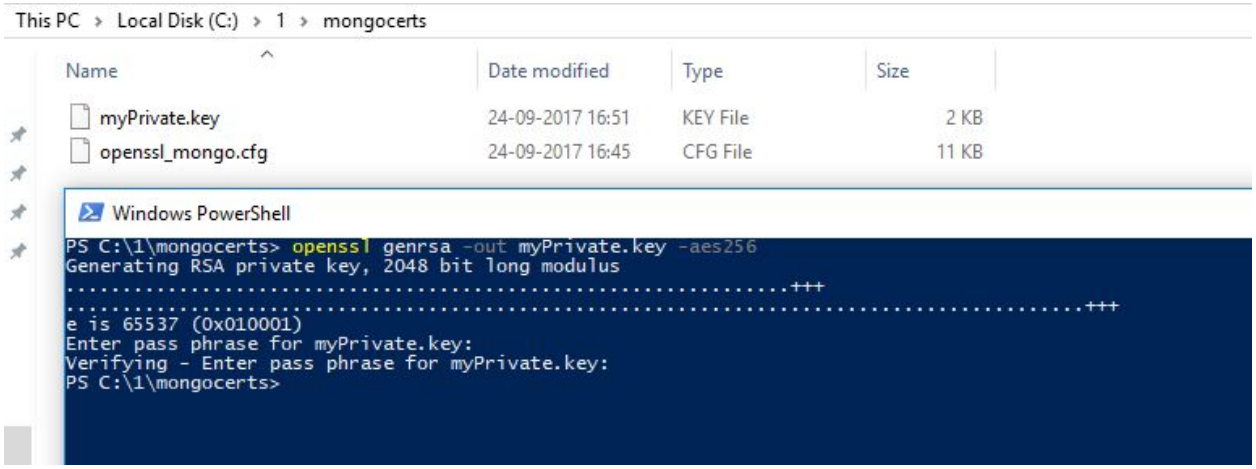
Creating the private key and certificate

Key

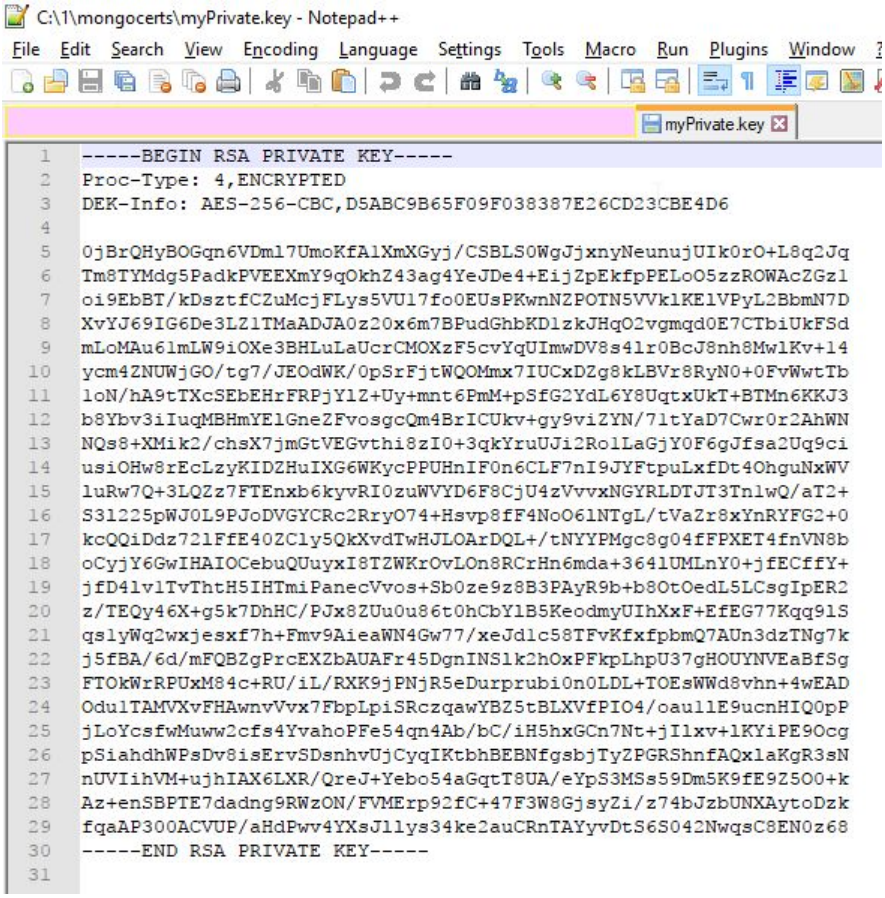
To create the private key, use the following command:

```
openssl genrsa -out myPrivate.key -aes256
```

The system will ask for a `pass phrase`, I used: `1234`



Notice that the system creates a .key file. If you were to open this in a text editor, you'll realize that it's a plain text file.




```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-256-CBC,D5ABC9B65F09F038387E26CD23CBE4D6

0jBrQHyBOGqn6VDml7UmoKfAlXmXGyj/CSBLS0WgJjxnyNeunujUIk0rO+L8q2Jq
Tm8TYMdg5PadkPVEEXmY9qOkhZ43ag4YeJDe4+EijZpEkfpPELoO5zzROWAcZGzl
oi9EbBT/kDsztfCZuMcjFLys5VUl7fo0EUsPKwnNZPOTN5VVklKElVPyL2BbmN7D
XvYJ69IG6De3LZlTMaADJA0z20x6m7BPudGhbKDlzkJHqO2vgmqd0E7CTbiUkFSd
mLoMAu6lmLW9iOXe3BHLuLaUcrCMOXzF5cvYqUImwDV8s4lr0BcJ8nh8MwlKv+14
ycm4ZNUWjGO/tg7/JEOdWK/0pSrFjtWQOMmx7IUCxDZg8kLBVr8RyN0+0FvWwtTb
loN/hA9tTXcSEbEHrFRPjYlZ+Uy+mnt6PmM+pSfG2YdL6Y8UqtxUkT+BTMn6KKJ3
b8Ybv3iIuqMBHmYElGneZFvosgcQm4BrICUkv+gy9viZYN/7ltYaD7Cwr0r2AhWN
NQs8+XMik2/chsX7jmGtVEGvthi8zI0+3qkYruUJi2RolLaGjY0F6gJfsa2Uq9ci
usiOHw8rEcLzyKIDZHuIXG6WKycPPUHnIF0n6CLF7nI9JYFtpuLxfDt4OhguNxWV
luRw7Q+3LQZz7FTEnxb6kyvRI0zuWVYD6F8CjU4zVvvxNGYRLDTJT3TnlwQ/aT2+
S3l225pWJ0L9PJoDVGYCRc2RryO74+Hsvp8fF4NoO6lNTgL/tVaZr8xYnRYFG2+0
kcQQiDdz72lFfE40ZCly5QkXvdTwHJLOArDQL+/tNYYPMgc8g04fFPXET4fnVN8b
oCyjY6GwIHAIOCebuQUuyxI8TZWKrOvLOn8RCrHn6mda+3641UMLnY0+jfECffY+
jfD4lvlTvThtH5IHTmiPanecVvos+Sb0ze9z8B3PAyR9b+b8OtOedL5LCsgIpER2
z/TEQy46X+g5k7DhHC/PJx8ZUu0u86t0hCbYlB5KeodmyUIhXxF+EfEG77Kqq9lS
qslyWq2wxjesxf7h+Fmv9AieaWN4Gw77/xeJdlc58TFvKfxfpbmQ7AUn3dzTNg7k
j5fBA/6d/mFQBZgPrcEXZbAUAFr45DgnINSlk2hOxPFkpLhpU37gHOUYNVEaBfSg
FTOkWrRPUxM84c+RU/iL/RXK9jPNjR5eDurprubi0n0LDL+TOEsWWd8vhn+4wEAD
OdulTAMVXvFHAwnvVvx7FbpLpiSRczqaWYBZ5tBLXVfPIO4/oaullE9ucnHIQ0pP
jLoYcsfwMuww2cfs4YvahoPFe54qn4Ab/bC/iH5hxGCn7Nt+jIlxv+lKYiPE9Ocg
pSiahdhWPsDv8isErvSDsnhvUjCyqIKtbhBEBNfgsbjTyZPGRShnfAQxlaKgR3sN
nUVIihVM+ujhIAX6LXR/QreJ+Yebo54aGqtT8UA/eYpS3MSs59Dm5K9fE9Z5O0+k
Az+enSBPTE7dadng9RWzON/FVMErp92fC+47F3W8GjsyZi/z74bJzbUNXAytoDzk
fqaAP300ACVUP/aHdPwv4YXsJllys34ke2auCRnTAYyvDtS6S042NwqsC8EN0z68
-----END RSA PRIVATE KEY-----
```


Cool!

On to creating the root certificate.

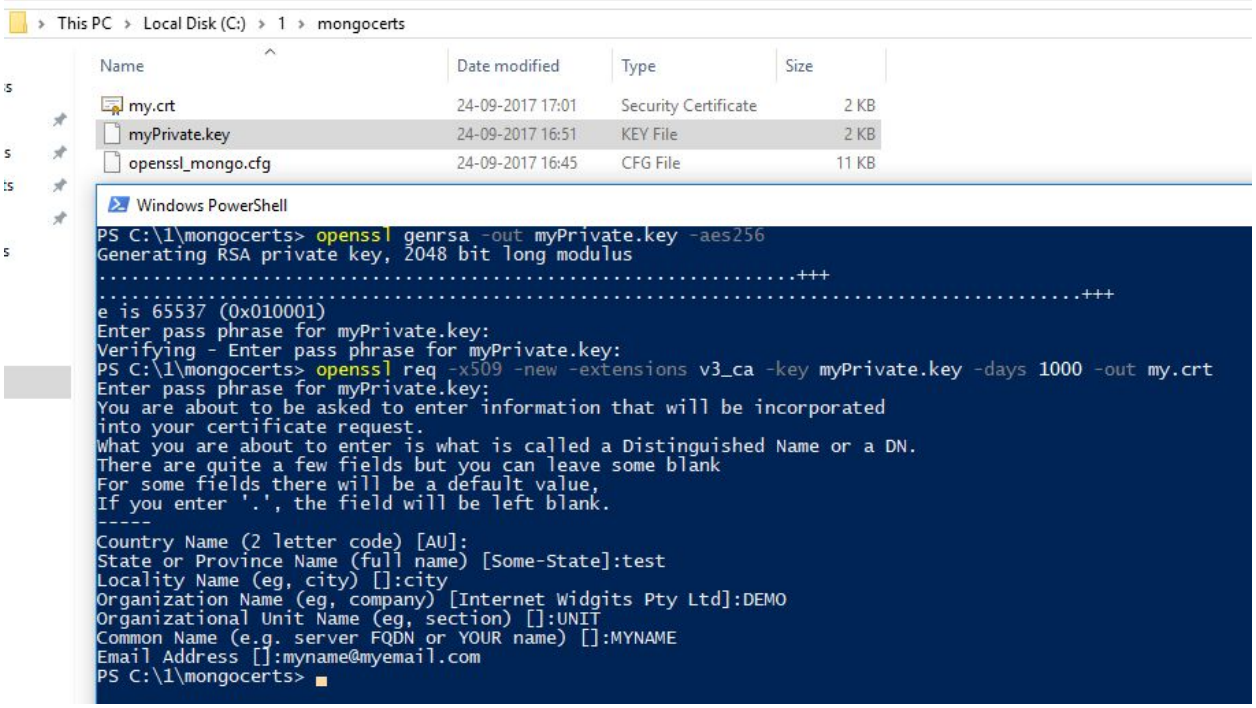
Certificate

Use the following command.

```
openssl req -x509 -new -extensions v3_ca -key myPrivate.key
-days 1000 -out my.crt
```

The system will ask you to enter the passphrase for the key (which is `1234` if you used the same one as me) and then ask a bunch of questions to make the certificate.

The values that you enter here are combined to create the "Distinguished Name" of your certificate. *Remember these values.*



Here are the values that I used:

- Country Name - `AU`
- State or Province name - `test`
- Locality name - `city`
- Organization name - `DEMO`
- Organizational unit name - `UNIT`
- Common Name - `MYNAME`
- Email Address - `myname@myemail.com`

Feel free to use the ones you like. Remember to note these values down - they're going to be useful shortly.