SSL exercise 1: Create certificates

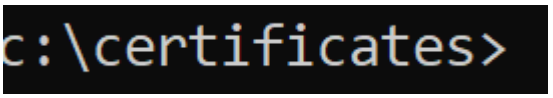
The purpose of this exercise is to make sure your computer has the tools to create certificates (self-signed) that you can use locally. And then create a certificate you can use.

Part 1: Install OpenSSL

In order to create certificates, you must have something that can create them
OpenSSL is a fine tool for this. It is free, and easy to use.
So let's install OpenSSL. I would recommend using the installer found here.
Alternatively you can install it from the OpenSSL webpage, but the process is a little more complicated.
OpenSSL is installed properly, if you can use the write "OpenSSL" in the command prompt and it shows a lot of information about OpenSSL.
If this is not the case, you might have to restart your computer (sometimes some environment variables aren't updating until a restart)
And if it still doesn't work, check if you have a Path reference to the directory where you installed OpenSSL (Help part in the bottom of this document)

Part 2: Create a private key:

In a command prompt, navigate to the directory where you want your certificate.
I would recommend creating a new directory in the root of your c: drive called certificates.



It should look like the above picture in your command prompt.
Then enter the following command:

openssl genrsa -aes256 -out key.pem 2048

It will ask you for a password. Remember this password, as you have to use it several times later.
If nothing went wrong, you now have a private key file in your certificate directory 🙂

Part 3: Create a Certificate Signing Request and a Certificate

Now a private key is not enough. You also need a certificate file for your server to send data encrypted.
This is a 2 step procedure. First you need to create a Certificate Signing Request (CSR) file, before you can generate the actual certificate file.
You do this by entering the following command in your command prompt:

openssl req -new -key key.pem -out signreq.csr

It will now ask for the password you used for the key.
Then it will ask for a lot of information. Normally this information is important for a certificate to work properly, but for our "self-signed" simple case, it is not important.

When you have the CSR file, you can create the certificate. Enter the following:

openssl x509 -req -days 365 -in signreq.csr -signkey key.pem -out certificate.pem

It will again ask for the key password, but nothing else.

In your certificates folder, you should now have 3 files.

- certificate.pem
- key.pem
- signreq.csr

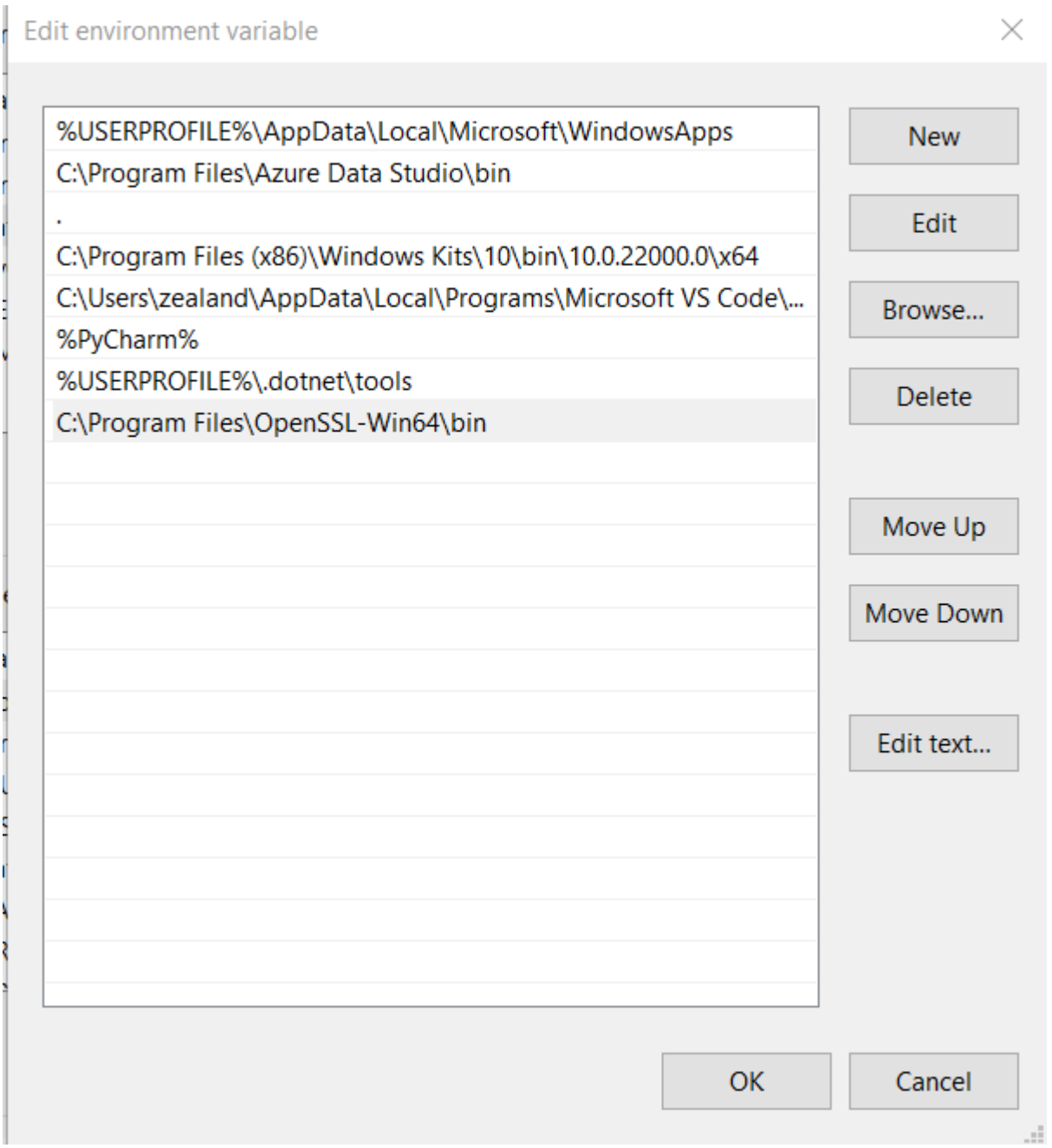
The certificate and key files are the ones you will use in the next exercise, where you should encrypt the data being sent between server and client.

Useful links (also for exercise 2):

https://snyk.io/blog/implementing-tls-ssl-python/
https://www.openssl.org/
https://docs.python.org/3/library/ssl.html
https://gist.github.com/newmind/dffd160cfd94a9206c5c

Help: Add value to environment variable PATH:

- Open your control Panel
- Under System and Security -> System
- Then click Advanced system settings -> Advanced tab
- Click on Environment Variables and under System Variables
- Find Path and click edit
- Then add the directory as shown (if this was also your installation directory)



- Click ok and save for all open windows
- Reopen the command prompt and try again. You might have to restart your computer