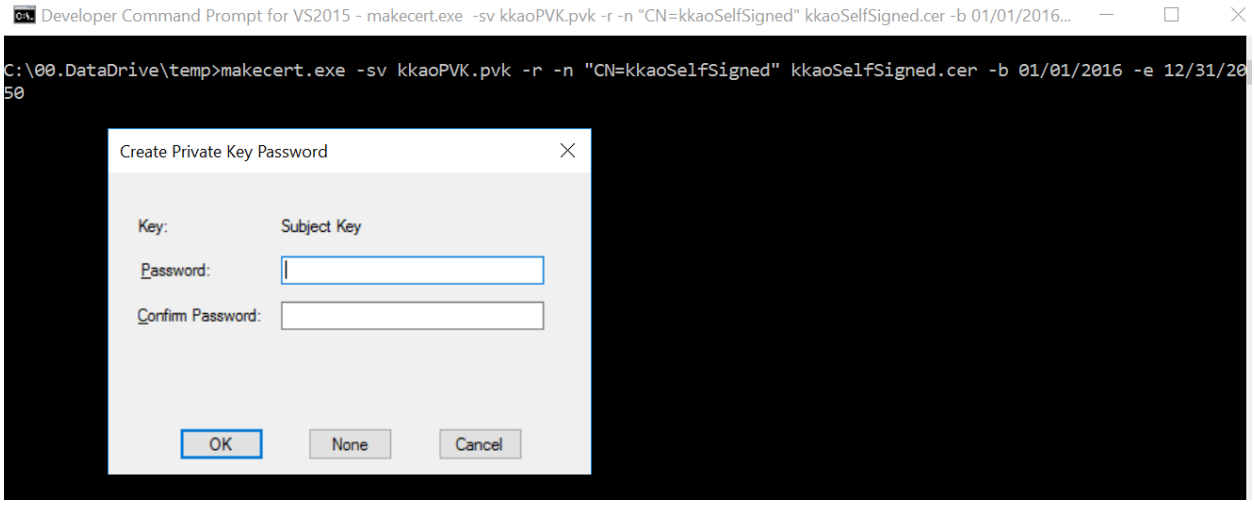
Azure IoT Device SDK + X.509 Certification

1. Open Visual Studio Developer Command, and run below command to create self-signed X.509 cert with private Key.

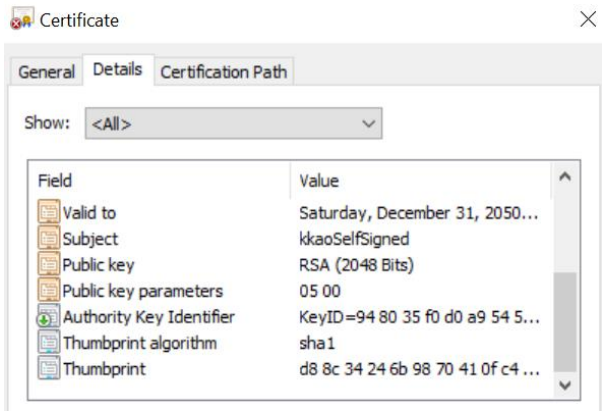
   makecert.exe -sv kkaoPVK.pvk -r -n "CN=kkaoSelfSigned" kkaoSelfSigned.cer -b 01/01/2016 -e 12/31/2050

   - -sv kkaoPVK.pvk is the name of the file containing the private key.
   - -n "CN=kkaoSelfSigned" is the name that will appear on the certificate (and in the certificate store).
   - kkaoSelfSigned.cer is the name of the certificate file.
   - -b mm/dd/yyyy is the date when the certificate becomes valid.
   - -e mm/dd/yyyy is the date when the certificate expires.
   - -r indicates that this will be a self-signed certificate.



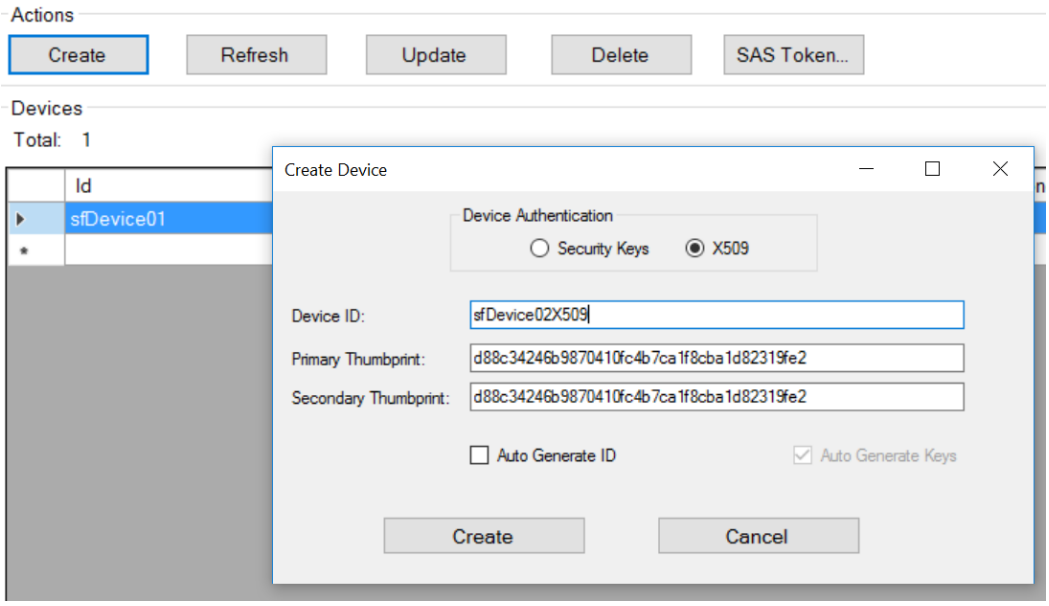
   You will be asked to input password of Private Key.

2. Double click the certification file (kkaoSelfSigned.cer), and find out thumbprint.



Copy the thumbprint value, remove all space from value, and create a IoT device on Device Explorer by X.509 certificate.

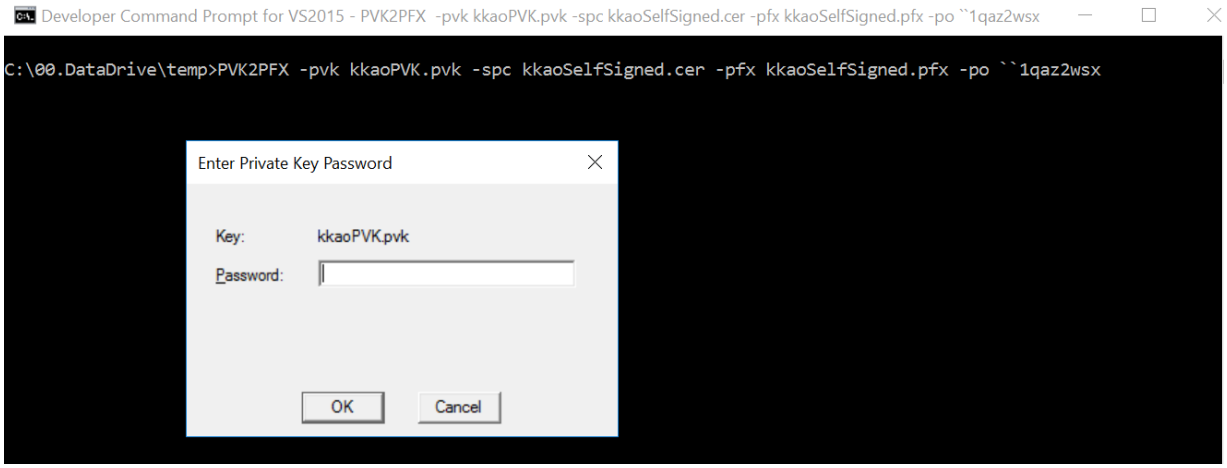
The thumbprint value captured from windows is Unicode, you shall save it to be ANSI, before past it into Device Explorer.



3. Open Visual Studio Developer Command, and run below command to generate Personal Information Exchange (.pfx) file

   PVK2PFX –pvk kkaoPVK.pvk –spc kkaoSelfSigned.cer –pfx kkaoSelfSigned.pfx –po ``1qaz2wsx

   - -pvk kkaoPVK.pvk is the private key file that you created in previous step.
   - -spc kkaoSelfSigned.cer is the certificate file you created in previous step.
   - -pfx kkaoSelfSigned.pfx is the name of the .pfx file that will be created.
   - -po yourpfxpassword is the password that you want to assign to the .pfx file.



* kkaoSelfSigned.pfx and password: ``1qaz2wsx are required on code.

4. Code on Visual Studio by using X.509 certificate on IoT Device SDK

```
string pfxCertificate = "C:\\00.DataDrive\\temp\\kkaoSelfSigned.pfx";
string pfxCertificatePassword = "``1qaz2wsx";
var x509Certificate = new X509Certificate2(pfxCertificate, pfxCertificatePassword);
var authMethod = new DeviceAuthenticationWithX509Certificate("sfDevice02X509", x509Certificate);
var deviceClient = DeviceClient.Create("sfIoTHub-Asia-1.azure-devices.net", authMethod,
                   Microsoft.Azure.Devices.Client.TransportType.Mqtt);
var message = new Message(Encoding.UTF8.GetBytes("This is a Message from X.509 Device"));
try
{
        deviceClient.SendEventAsync(message).Wait();
}
catch (UnauthorizedException)
{
       await deviceClient.CloseAsync();
       throw;
}
catch (Exception ex)
{
        Console.WriteLine(ex.Message);
}
```

Device Explorer screenshot: