

**Worksheet 7 – Digital Certificates****Certificate Manipulation with the .NET Framework****Topics:**

- Certificate Manipulation with .Net
- Digital Signatures using Digital Certificates
- Confidentiality using Digital Certificates

©2022: { rui.ferreira,marisa.maximiano,ricardo.p.gomes}@ipleiria.pt

## 1. Digital Certificates

The following exercises show how to manipulate Digital Certificates in .Net

The following classes and enums are important:

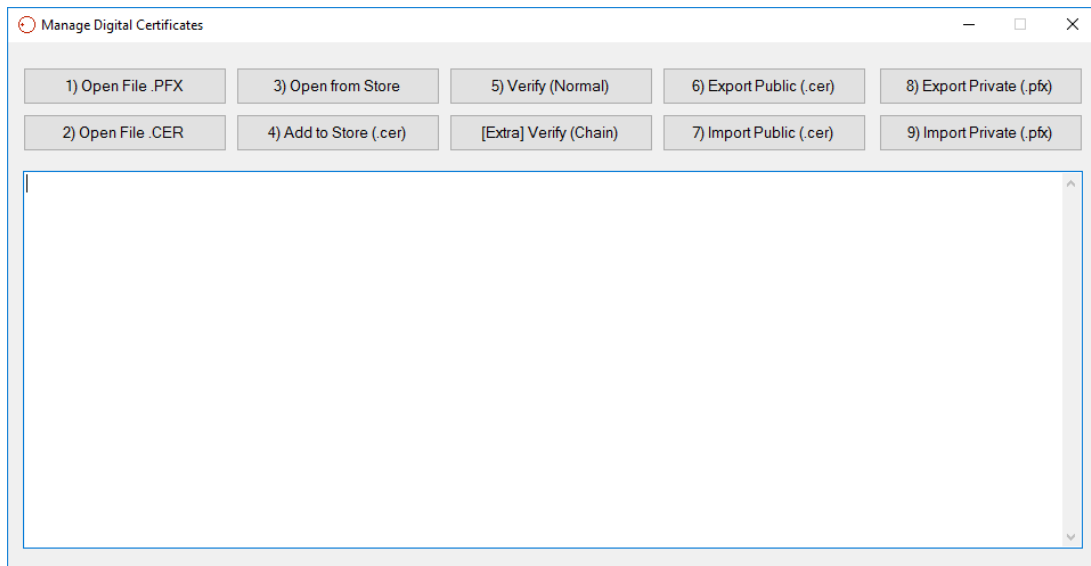
- X509Certificate2
- X509Certificate2Collection
- X509Certificate2UI // requires the assembly *System.Security* (References)
- X509Store e StoreName (enum)
- X509Chain
- ContentInfo
- CmsSigner
- SignedCms
- CmsRecipient
- EnvelopedCms

These are their *namespaces*:

- System.Security.Cryptography;
- System.Security.Cryptography.Pkcs;  
Note: requires the assembly *System.Security* (References)
- System.Security.Cryptography.X509Certificates;

## Exercises

1. Download the project “ei.si-worksheet7-ex1.1”, from moodle, review the code and:



- a) Place the certificates “...a.cer” and “...a.pfx” in the folder “bin/debug” and change the constant values accordingly.

**Note:** The “...” in the certificate filename must be completed with the certificate to be used (e.g.: [certa.cer](#) or [estg.ei.si.a.cer](#), etc.)

- b) Implement a method that writes in the textbox (*textBoxInfo.Text*) the main information of a digital certificate:

```
private void ShowCertificate(X509Certificate2 cert)
```

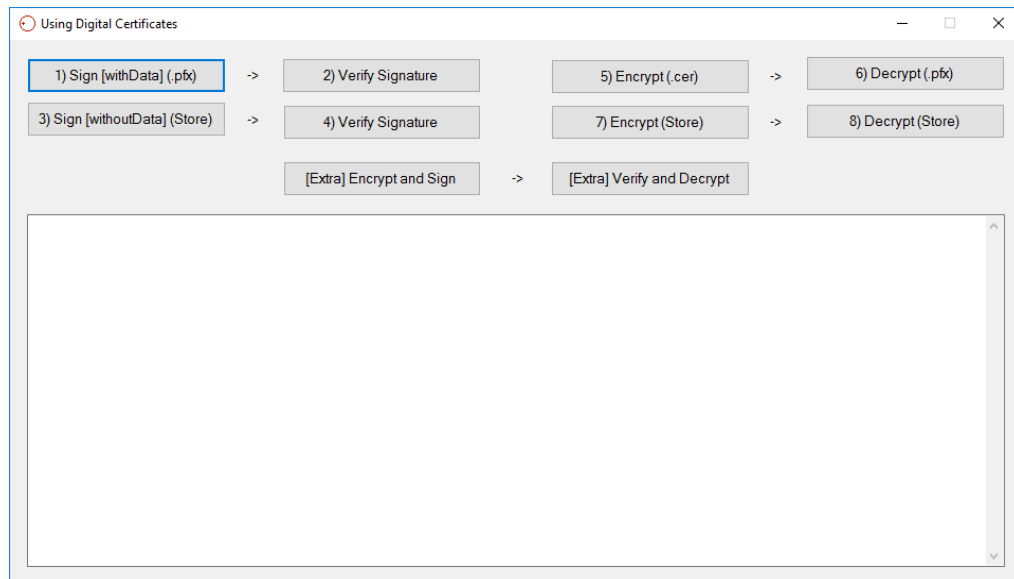
- c) Implement the following features for each of the buttons:
  - 1) Open the file “...a” (that has the private key) and show its main information in the textbox.
  - 2) Open the file “...a” (that has only the public key) and show its main information in the textbox.
  - 3) Open a certificate from the **Personal** folder in the **Certificate Store** of your operating system and show its main information in the textbox.
  - 4) Add the “...a” digital certificate to the “**Other People**” folder of the **Certificate Store**.
  - 5) Verify if the “...a” certificate is valid.
  - 6) Export, to a “temp.cer” file, the certificate on the file “...a.cer”.
  - 7) Import, the certificate on the “temp.cer” file, to a X509Certificate2 object, using the **.Import()** method.

8) Export, to a “temp.pfx” file, the certificate and the corresponding private key that are on the “...a.pfx” file.

Note: you should assure the security of this file with a password of your choice.

9) Import the digital certificate and corresponding private key, in the “temp.pfx” file, to a X509Certificate2 object, using the **.Import()** method.

2. Download the “ei.si-worksheet7-ex1.2” project, from *moodle*, review the code and:



a) Implement the following features for each button:

- 1) Create a digital signature in the PKCS#7 format, using a certificate on your hard drive, that uses as data the content of the textbox. The result should include both the data and the signature.
- 2) Verify the signature created.
- 3) Create a signature in the PKCS#7 format, using a certificate on **certificates store**, that uses as data the content of the textbox. The result should include both the data and the signature.
- 4) Verify the signature created.
- 5) Create a confidential message, with a certificate on your hard drive, in the PKCS#7 format, that uses the content of the textbox as it's data.
- 6) Decrypt the message.
- 7) Create a confidential message, with a certificate on the certificate store, in the PKCS#7 format, that uses the content of the textbox as it's data.
- 8) Decrypt the message.

***Extra Class:***

- 1) Using the ei.si-worksheet7-ex1.1 project:
  - a. Verify that the "...a" certificate is valid, using the certificate chain.
- 2) Using the ei.si-worksheet7-ex1.2 project:
  - a. Create a confidential message, with its integrity and authentication assured, using a certificate of your choice. The data should be the content of the textbox and the format of the message PKCS#7.
  - b. Verify the signature and decrypt the message.