Public Key Infrastructure

Use of digital signatures in information security

Mihai-Lica Pura
2020.03.18

Topics

- **Information security**
  - Cryptography and its use for accomplishing the goals of information security:
    - Authentication
    - Confidentiality
    - Integrity
    - Non-repudiation
- **Electronic signatures vs. digital signatures**

Topics

- **Symmetric cryptography**
  - Problem with sharing the key between the parties
- **Asymmetric cryptography**
  - Problem with trusting that a given public key belongs to a certain entity (person, server)
  - Speed problem
- Solving the speed problem
  - **Asymmetric cryptography for confidentiality**
    - Hybrid cryptography

Topics

- **Solving the trust problem**
  - Public Key Infrastructure
    - Certificate issuance flow
    - Trust flow
  - Web of trust (PGP)
    - Same as above
  - Identity based cryptography
    - Same as above

Topics

- **Public Key Infrastructure**
- **Certificate issuance in PKI**
  - PKCS#12, PEM, DER, X509 v3, ASN.1, OID
- **Management of certificates in Microsoft Windows**
  - Windows certificate stores, certmgr.msc, mc.exe
- **Validation of a certificate**
  - Time validity
  - Revocation status
  - Trust
    - Certificate trust path
    - Self-signed CA certificates
    - Microsoft Trusted Root Certificate Program
    - Mozilla CA Certificate Policy

Topics

- **Certificate revocation**
  - CRL, OCSP
- **Certificate search**
  - CA web page manual search, LDAP
- **Performing a digital signature**
  - Using Microsoft CNG API through certificate stores
    - flow
  - Using PKCS#11
    - flow
  - Remote/cloud/mobile through Cloud Signature Consortium signing protocol
    - flow

Topics

- **E-signature vs. e-seal**
- **Use of digital signed documents**
  - ANAF
  - CNAS
  - ONRC
  - UNNPR
  - ITM
  - MySMIS 2014
  - MO
  - etc.

Topics

- **Digital signing documents**
  - **Qualified Electronic Signatures**
    - PDF (Adobe Acrobat Reader)
      - Signing PDF files
      - Signing PDF forms with a digital signature field
    - Microsoft Office documents (Microsoft Office software)
    - PKCS#7 (DeskSeal, Prosigner, etc.)
      - Attached, detached, cosignatures
    - XMLDSig
  - **Advanced Electronic Signatures**
    - CAdES, PAdES, XAdES
  - **Time stamping of digital signatures**

Topics

- **Digital signing code**
  - Java applets (obsolete)
  - Java Web Start (signed jars for JNLP)
  - Macros (Microsoft Office)
  - Microsoft software installation packages
- **Digital signing for authentication**
  - Certificate login
- **Cryptography APIs**
  - High level
  - low level

The trust problem

- Asymmetric key cryptography based on two keys:
  - a public key known to anybody,
  - a private key known only by the owner of the pair
  - the two keys are in a mathematical relation

- **How to be sure that a public key really belongs to a certain person** or entity?

- Solutions:
  - Based on certificates
    - 1. **Public Key Infrastructure**
    - 2. **Web of Trust**

  - Certificateless
    - **Identity based Cryptography**

The trust problem

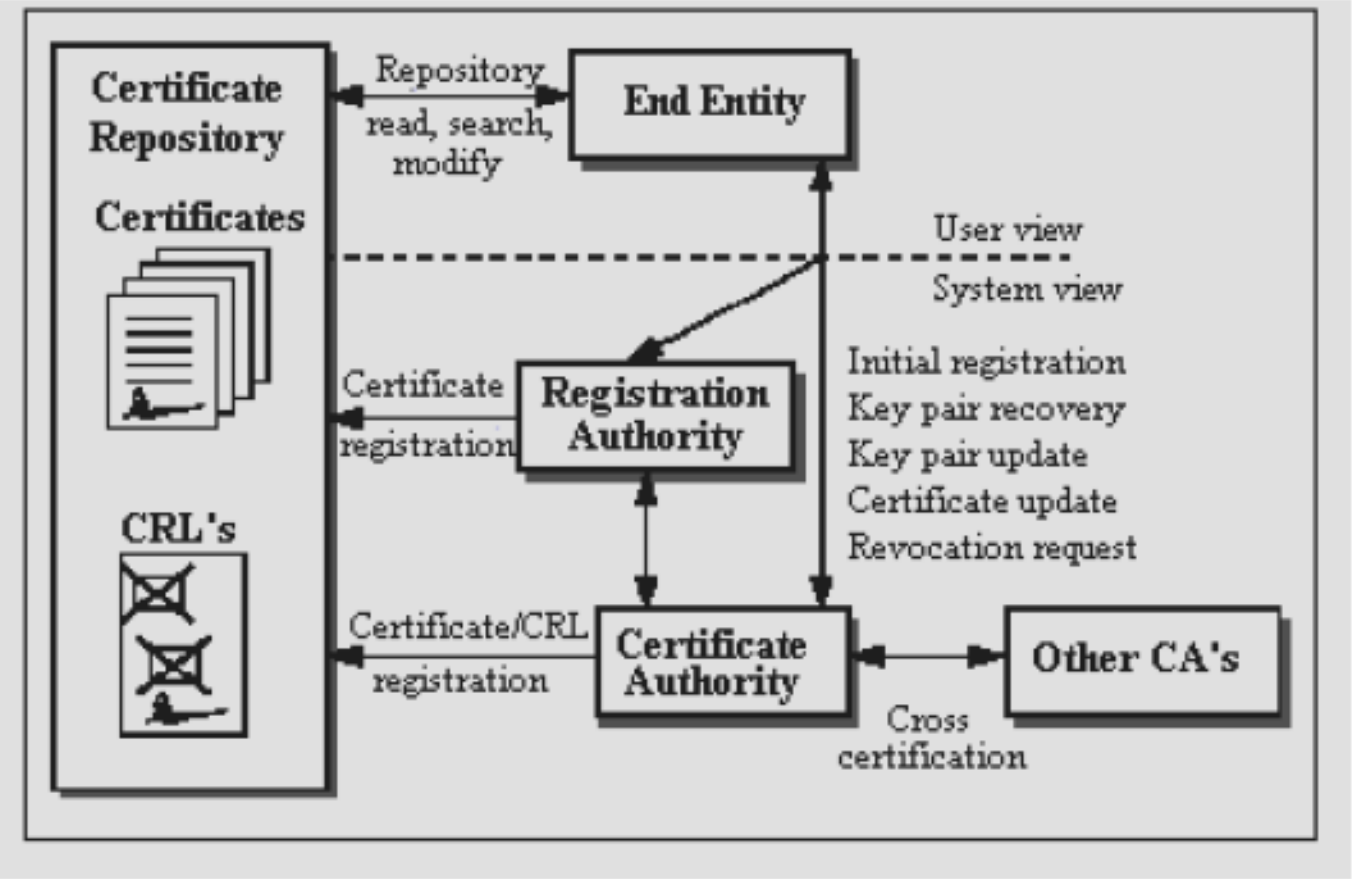
- **PKI solution**
- The public key is bounded to the identity of the owner through a Certificate Authority (CA)
- The CA guarantees the identity of the owner of the public key
- The element that securely proves this bound is the digital certificate

Public Key Infrastructure

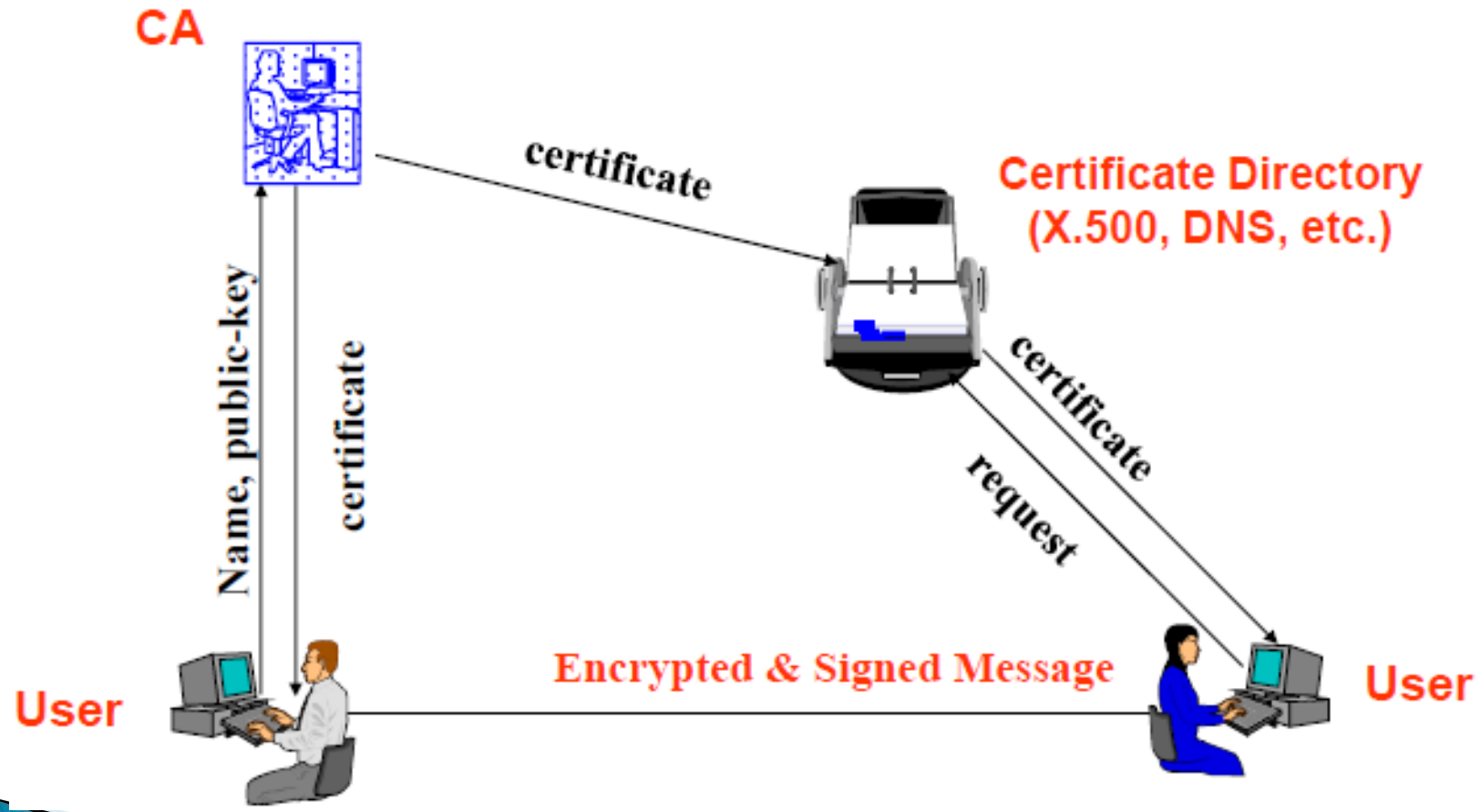
- PKI
  - set of components (hardware and software) that work together for using public-key technology in a secure manner

- Components of a PKI
  - The end-entities (EE)
  - The certificate authority (CA)
  - The certificate repository (CR)
  - The registration authority (RA)
  - Digital certificates

Public Key Infrastructure



Public Key Infrastructure



End-entity

- is defined as a user of PKI certificate and/or end-user system that is the subject of a certificate

- is a generic term for a subject that uses some services or functions of the PKI system, which may be
  - a certificate owner (human being or organization or some other entities)
  - a requestor (it might be application program) for certificate or CRL

Certificate Authority

- CA
  - a trusted authority
  - central administration
  - provides a statement (the digital certificate) that the enclosed public key belongs to the specified entity
    - organization/company to its employees
    - university to its students
    - public CAs to their clients

- Primary operations
  - certificate issuance
  - certificate renewal
  - certificate revocation

Certificate Authority

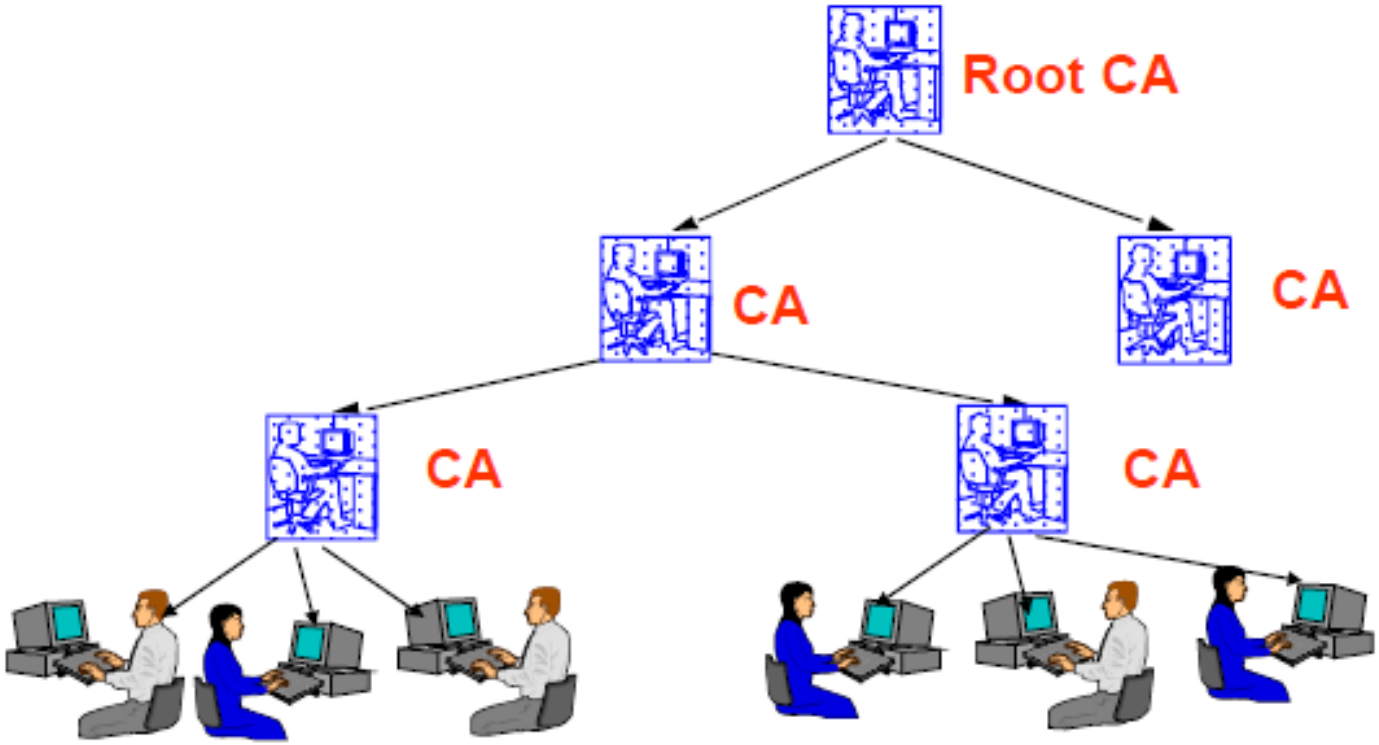
- Original intent was to certify that a public key really did belong to a given entity

- Role was later expanded to certify all sorts of other things
  - Are they a bona fide business?
  - Can you trust their web server?
  - Can you trust the code they write?
  - Is their account in good standing?
  - Are they over 18?

Certificate Authority Hierarchy

- Each CA issues certificates to its users community
  - *Root-CA* is used to establish TRUST between communities
  - Root certificates (self-signed) are installed/configured as **trusted** points

Certificate Authority Hierarchy



Registration Authority

- An optional component in a PKI
  - The CA can incorporate the role of an RA
- When used, the RA is
  - a trusted end-entity certified by the CA
  - acting as a subordinate server of the CA
- The CA can delegate some of its management functions to the RA
  - personal authentication tasks
  - generate keys
  - archive key pairs
- **The RA, however, does not issue certificates or CRLs**

Certificate Repository

- Store for certificates and CRLs
- X.509 certificate format fit to an X.500 directory
- Best implemented as a directory accessed by Lightweight Directory Access Protocol (LDAP)
- CR can be accessed with LDAP commands or procedures (bind, search, modify, unbind)

How to obtain a digital certificate (qualified)

- http://www.transsped.ro/
- http://ca.transsped.ro/Index.aspx?Document=enrol
- http://ca.transsped.ro/Index.aspx?Document=UA_Form

Free digital certificates

- Getting a free **S/MIME digital certificate (personal)**
- http://kb.mozillazine.org/Getting_an_SMIME_certificate
- Getting a free **SSL/TLS digital certificate (server)**
- https://geekflare.com/free-ssl-tls-certificate/

Free digital certificates (S/MIME)

- SECTIGO (formerly COMODO CA)
- https://www.instantssl.com/ssl-certificate-products/free-email-certificate.html

- Actalis
- https://www.actalis.it/products/certificates-for-secure-electronic-mail.aspx

Free digital certificates (S/MIME)

- WoSign
- https://www.wosign.com/english/Free_Email_Certificate.htm

- GlobalSign
- https://www.globalsign.com/en/personalsign/

- secorio
- https://www.secorio.com/en/?S_MIME_Email_Certificates

Free digital certificates (S/MIME)

- **Tutorials** – COMODO
- https://fastersolutions.com/wp-lib/wp-content/uploads/2015/07/comodoinstructions.pdf
- https://support.postbox-inc.com/hc/en-us/articles/202200540-Obtaining-an-S-MIME-Certificate-to-Sign-Emails
- https://www.netwoven.com/2015/07/07/secure-encrypt-emails-in-outlook-with-a-digital-id-certificate/

Free digital certificates (SSL/TLS)

- COMODO
- https://ssl.comodo.com/free-ssl-certificate.php

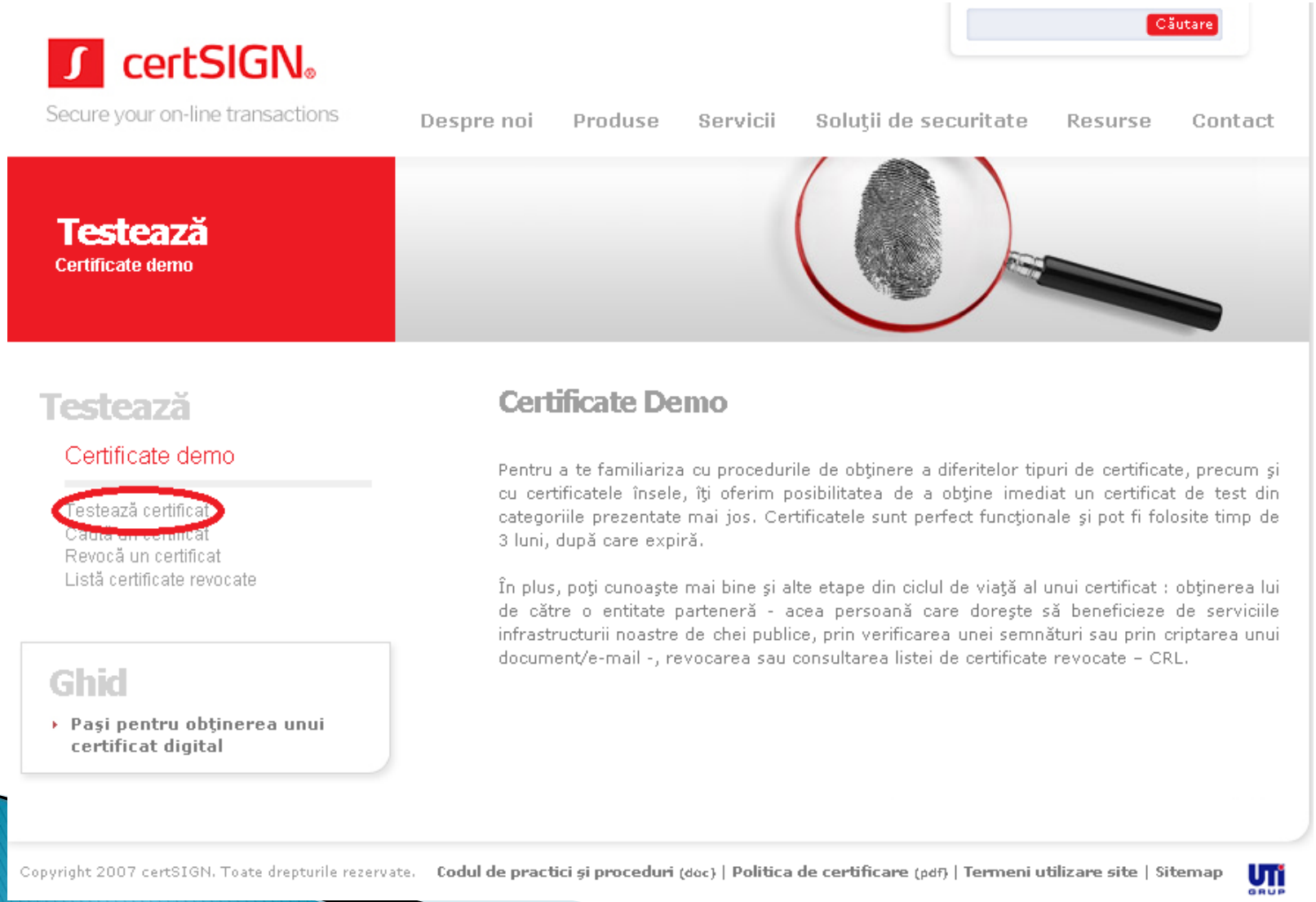
- Let's encrypt
- https://letsencrypt.org/
- https://www.sslforfree.com/

- ZeroSSL
- https://zerossl.com/

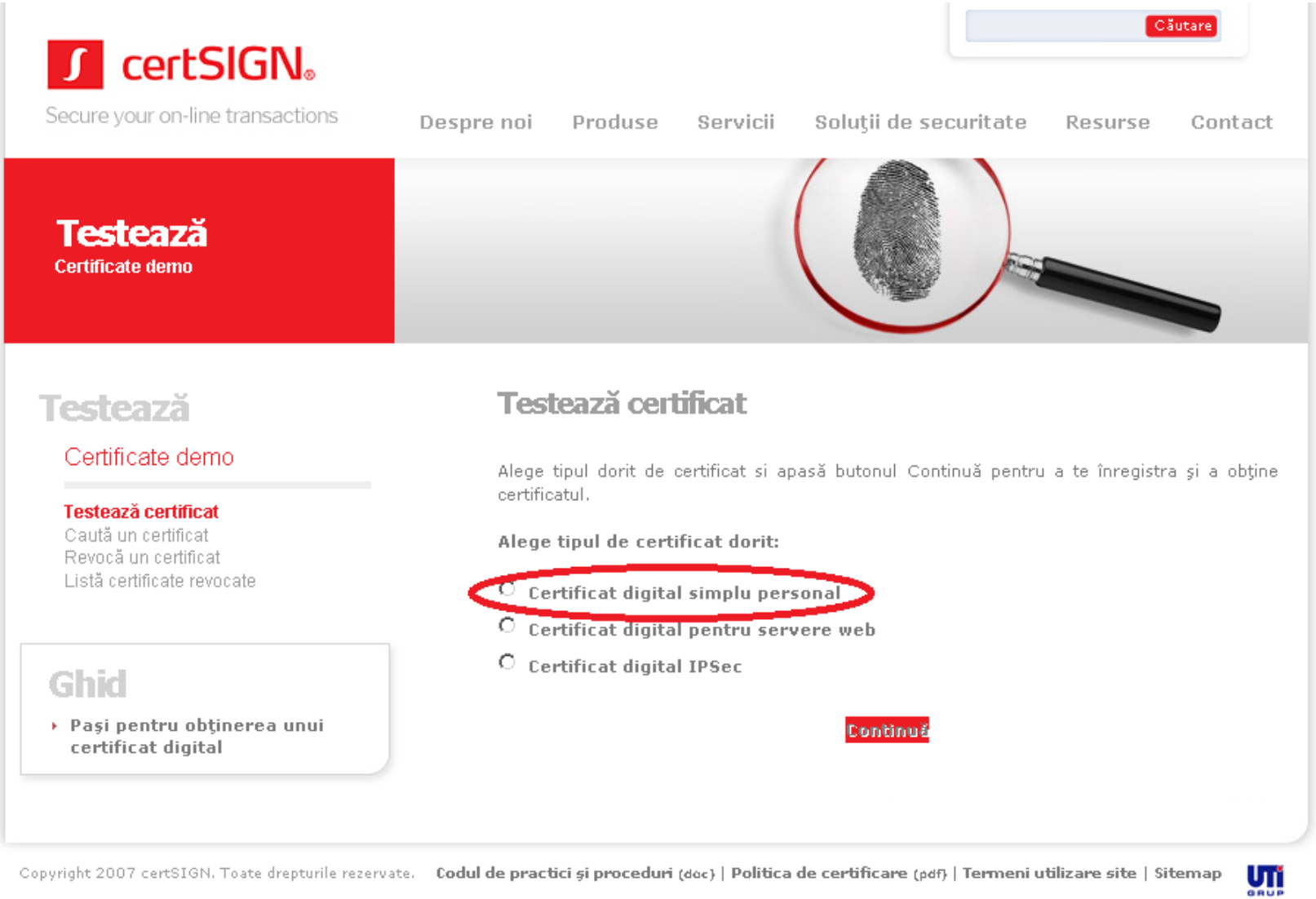
How to obtain a digital certificate (for testing)

- https://www.certsign.ro

How to obtain a digital certificate

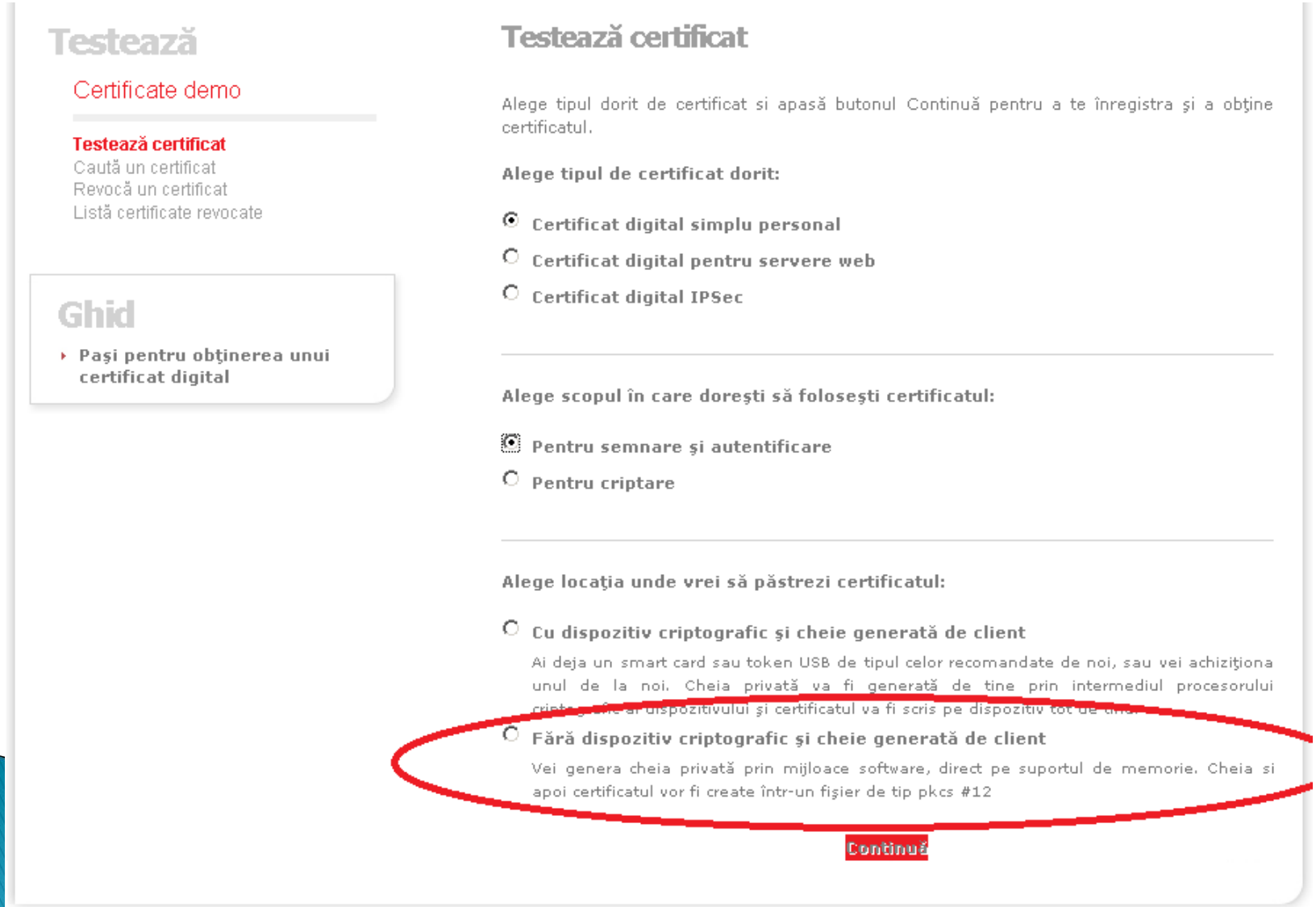


How to obtain a digital certificate

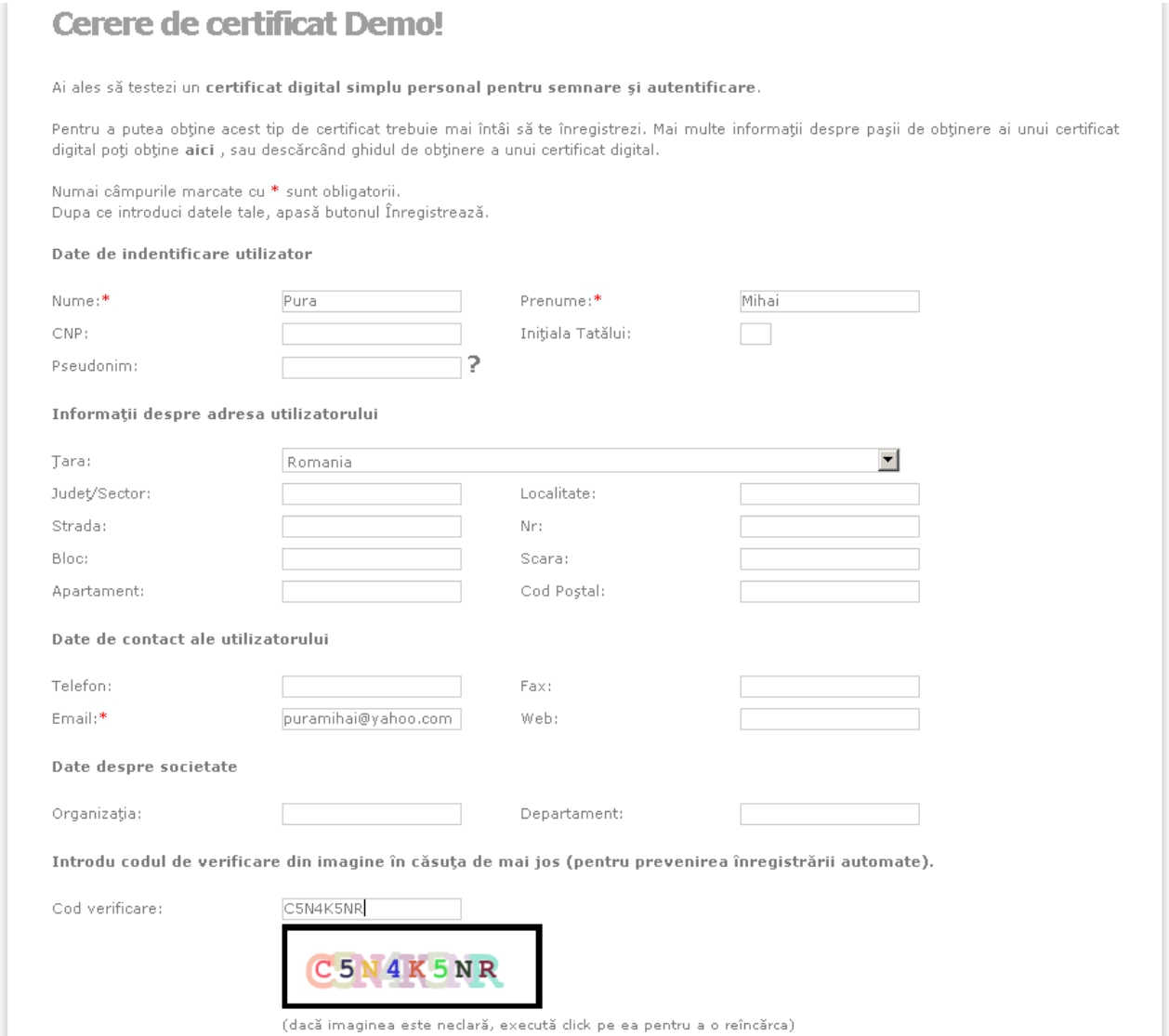


How to obtain a digital certificate

How to obtain a digital certificate



How to obtain a digital certificate



Cerere de certificat Demo!

Ai ales să testezi un **certificat digital simplu personal pentru semnare şi autentificare**.

Pentru a putea obţine acest tip de certificat trebuie mai întâi să te înregistrezi. Mai multe informaţii despre paşii de obţinere ai unui certificat digital poţi obţine **aici** , sau descărcând ghidul de obţinere a unui certificat digital.

Numai câmpurile marcate cu * sunt obligatorii.
Dupa ce introduci datele tale, apasă butonul Înregistrează.

Date de indentificare utilizator

Nume:* Pura
Prenume:* Mihai
CNP:
Iniţiala Tatălui:
Pseudonim: ?

Informaţii despre adresa utilizatorului

Ţara: Romania
Judeţ/Sector:
Localitate:
Strada:
Nr:
Bloc:
Scara:
Apartament:
Cod Poştal:

Date de contact ale utilizatorului

Telefon:
Fax:
Email:* puramihai@yahoo.com
Web:

Date despre societate

Organizaţia:
Departament:

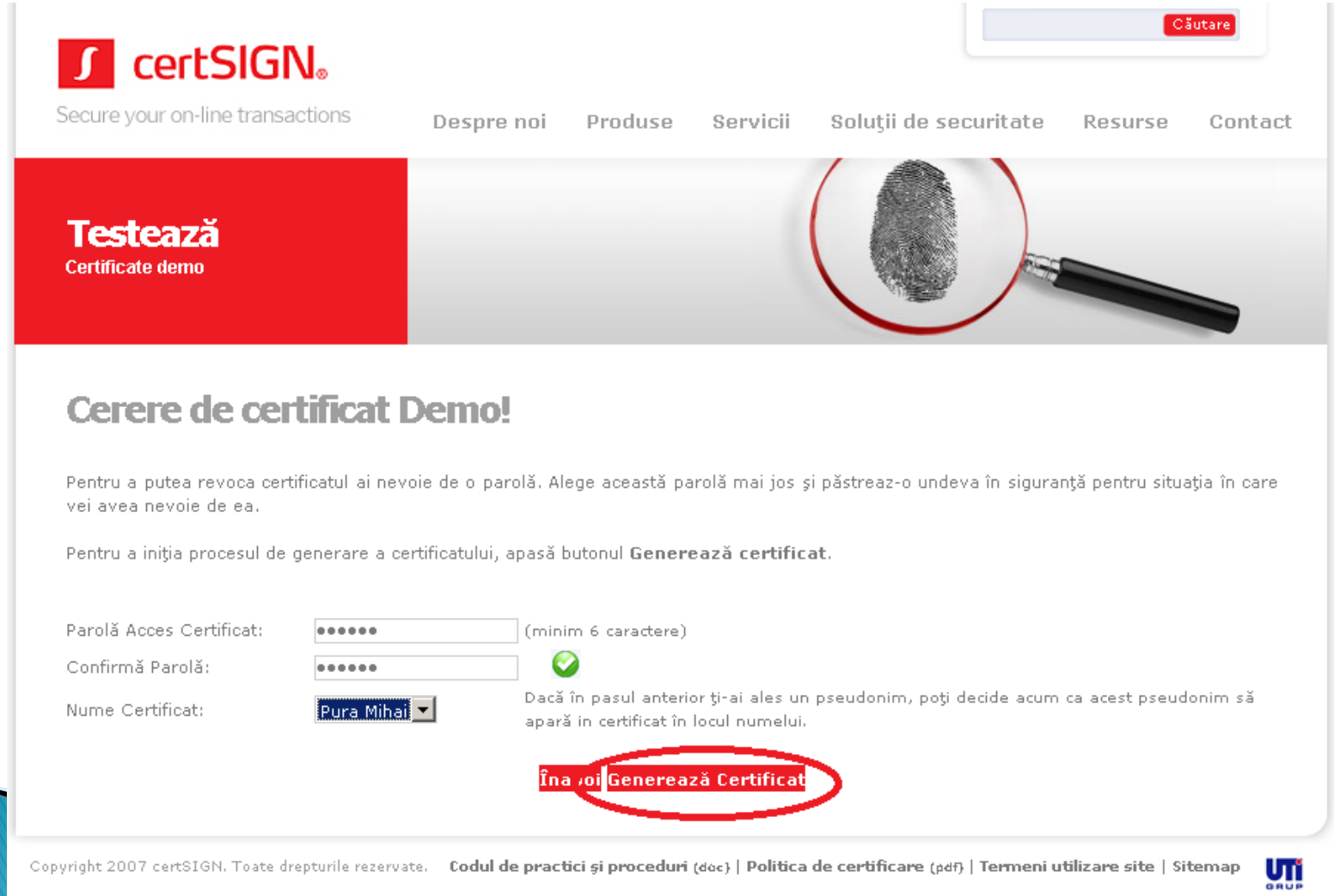
Introdu codul de verificare din imagine în căsuţa de mai jos (pentru prevenirea înregistrării automate).

Cod verificare: C5N4K5NR

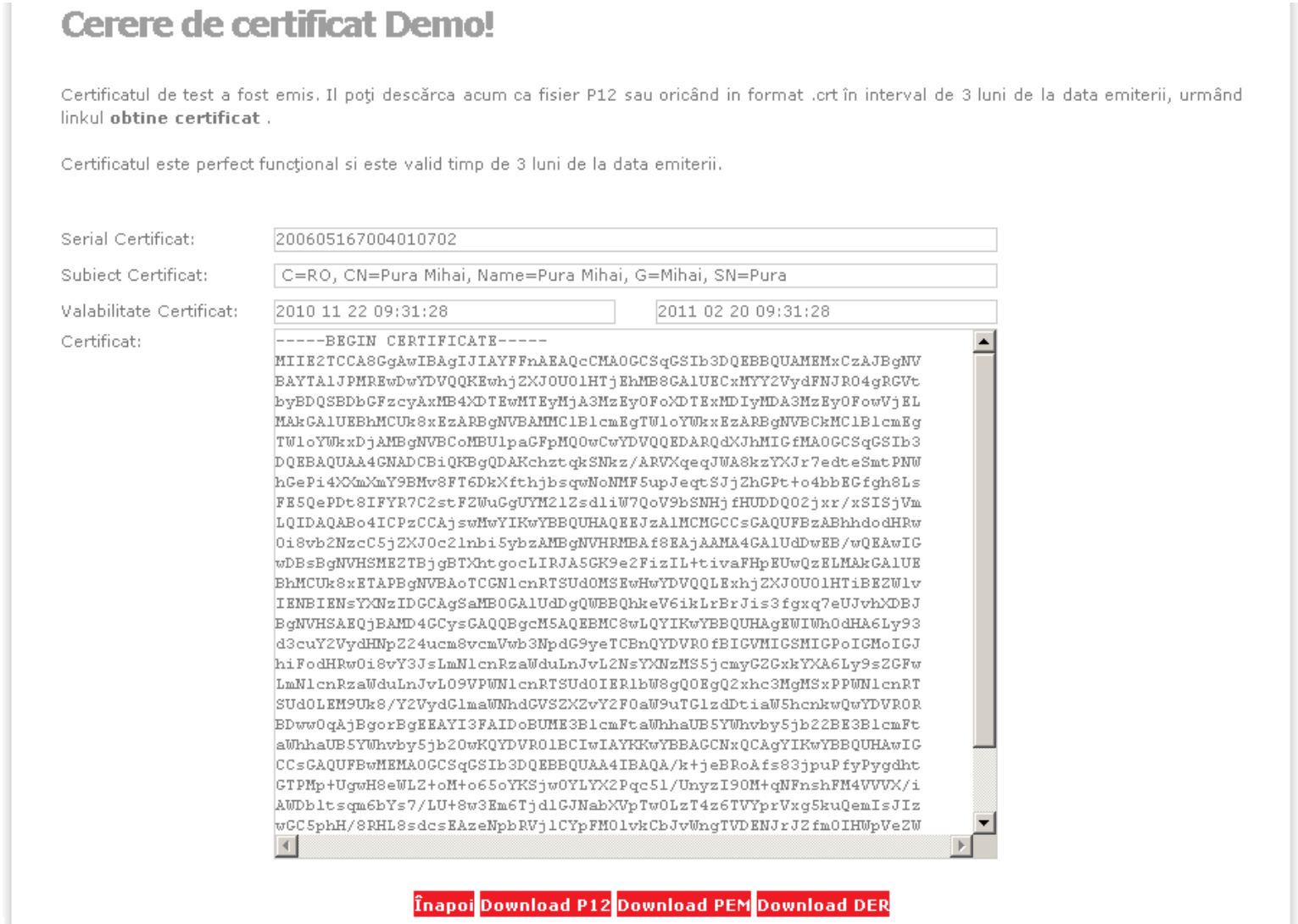
C5N4K5NR

(dacă imaginea este neclară, execută click pe ea pentru a o reîncărca)

How to obtain a digital certificate



How to obtain a digital certificate



Cerere de certificat Demo!

Certificatul de test a fost emis. Il poţi descărca acum ca fisier P12 sau oricând in format .crt în interval de 3 luni de la data emiterii, urmând linkul **obtine certificat** .

Certificatul este perfect funcţional si este valid timp de 3 luni de la data emiterii.

Serial Certificat: 200605167004010702

Subiect Certificat: C=RO, CN=Pura Mihai, Name=Pura Mihai, G=Mihai, SN=Pura

Valabilitate Certificat: 2010 11 22 09:31:28 — 2011 02 20 09:31:28

Certificat:

```
-----BEGIN CERTIFICATE-----
MIIE2TCCA8GgAwIBAgIJIAYFFnAEAQcCMA0GCSqGSIb3DQEBBQUAMEMxCzAJBgNV
BAYTAlJPMREwDwYDVQQKEwhjZXJ0UOlHTjEhMB8GAlUECxMYY2VydFNJRO4gRGVt
byBDQSBDbGFzcyAxMB4XDTEwMTEyMjA3MzEyOFoXDTExMDIyMDA3MzEyOFowVjEL
MAkGAlUEBhMCUk8xEzARBgNVBAMMClBlcmEgTWloYWkxEzARBgNVBCkMClBlcmEg
TWloYWkxDjAMBgNVBCoMBUlpaGFpMQ0wCwYDVQQEDARQdXJhMIGfMA0GCSqGSIb3
DQEBAQUAA4GNADCBiQKBgQDAKchztqkSNkz/ARVXqeqJWA8kzYXJr7edteSmtPNW
hGePi4XXmXmY9BMv8FT6DkXfthjbsqwNoNMF5upJeqtSJjZhGPt+o4bbEGfgh8Ls
FE5QePDt8IFYR7C2stFZWuGgUYM2lZsdliW7QoV9bSNHjfHUDDQO2jxr/xSISjVm
LQIDAQABo4ICPzCCAjswMwYIKwYBBQUHAQEEJzAlMCMGCCsGAQUFBzABhhdodHRw
Oi8vb2NzcC5jZXJ0c2lnbi5ybzAMBgNVHRMBAf8EAjAAMA4GAlUdDwEB/wQEAwIG
wDBsBgNVHSMEZTBjgBTXhtgocLIRJA5GK9e2FizIL+tivaFHpEUwQzELMAkGAlUE
BhMCUk8xETAPBgNVBAoTCGNlcnRTSUdOMSEwHwYDVQQLExhjZXJ0UOlHTiBEZWlv
IENBIENsYXNzIDGCAgSaMB0GAlUdDgQWBBQhkeV6ikLrBrJis3fgxq7eUJvhXDBJ
BgNVHSAEQjBAMD4GCysGAQQBgcM5AQEBMC8wLQYIKwYBBQUHAgEWIWh0dHA6Ly93
d3cuY2VydHNpZ24ucm8vcmVwb3NpdG9yeTCBnQYDVR0fBIGVMIGSMIGPoIGMoIGJ
hiFodHRwOi8vY3JsLmNlcnRzaWduLnJvL2NsYXNzMS5jcmyGZGxkYXA6Ly9sZGFw
LmNlcnRzaWduLnJvLO9VPWNlcnRTSUdOIERlbW8gQOEgQ2xhc3MgMSxPPWNlcnRT
SUdOLEM9Uk8/Y2VydGlmaWNhdGVSZXZvY2FOaW9uTGlzdDtiaW5hcnkwQwYDVR0R
BDwwOqAjBgorBgEEAYI3FAIDoBUME3BlcmFtaWhhaUB5YWhvby5jb22BE3BlcmFt
aWhhaUB5YWhvby5jb2OwKQYDVR0lBCIwIAYKKwYBBAGCNxQCAgYIKwYBBQUHAwIG
CCsGAQUFBwMEMA0GCSqGSIb3DQEBBQUAA4IBAQA/k+jeBRoAfs83jpuPfyPygdht
GTPMp+UgwH8eWLZ+oM+o65oYKSjwOYLYX2Pqc5l/UnyzI9OM+qNFnshFM4VVVX/i
AWDbltsqm6bYs7/LU+8w3Em6TjdlGJNabXVpTwOLzT4z6TVYprVxg5kuQemIsJIz
wGC5phH/8RHL8sdcsEAzeNpbRVjlCYpFMOlvkCbJvWngTVDENJrJZfmOIHWpVeZW
```

Înapoi | Download P12 | Download PEM | Download DER

P12

- P12 – PKCS#12 – Public Key Cryptography Standards 12 –
  - RSA Laboratories standard
  - defines the format of the files used to store a private key together with its corresponding digital certificate
  - encrypted with a symmetric key or a password
  - binary content

DER

- DER – Distinguished Encoding Rules
  - message transfer syntax published by ITU in X.690
  - are binary files
  - also used to store digital certificates

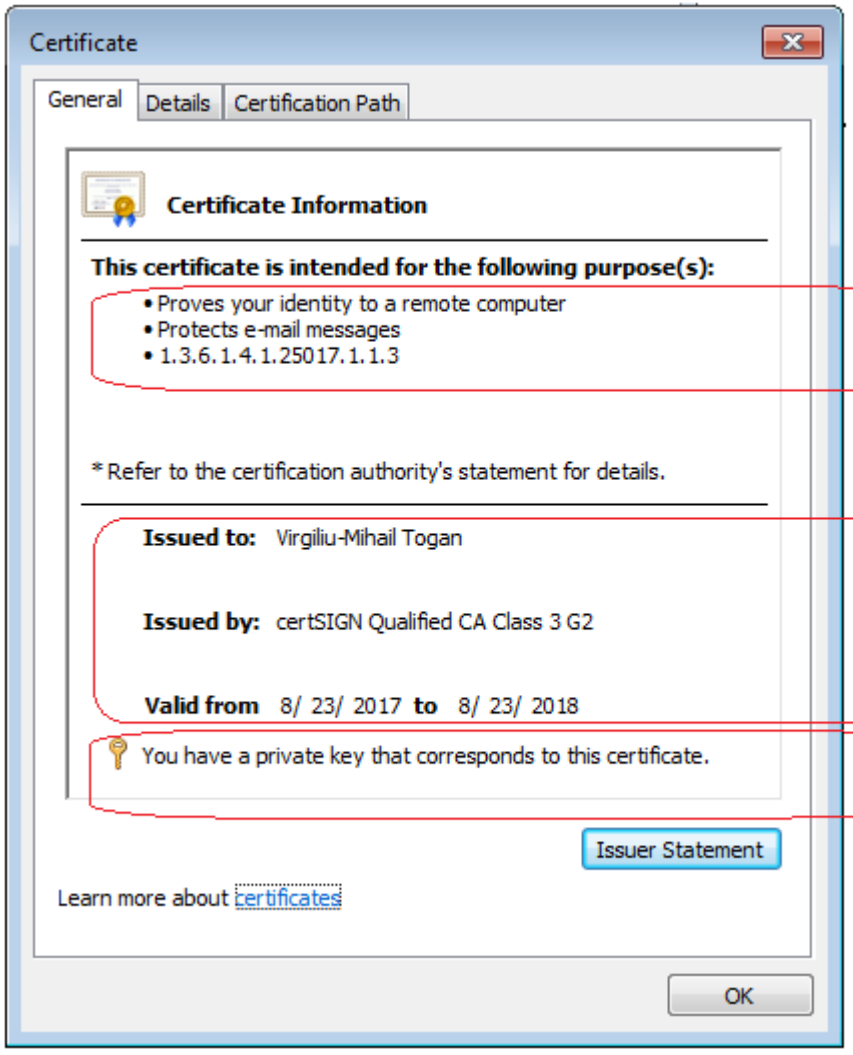
PEM

- PEM – Privacy Enhanced Mail
  - early standard for securing electronic mail, but never adopted as a widely used Internet Mail Standard
  - text files that contain a US-ASCII representation of a DER digital certificate
    - BASE64 encoding
    - enclosed between "-----BEGIN CERTIFICATE-----" and "-----END CERTIFICATE-----“
  - BASE64
    - encoding scheme used to represent binary data in an ASCII string by translating it into radix-64 representation
    - ASCII characters used are "A-Z", "a-z", "0-9", plus two more characters, often "+" and "/"

Digital certificates

- A digital certificate
  - a collection of information referring to a public key, digitally signed by the CA
  - conforms to the ITU (IETF) standard X.509 v3
  - the CA signs the information from the digital certificate using its private key

Digital certificates



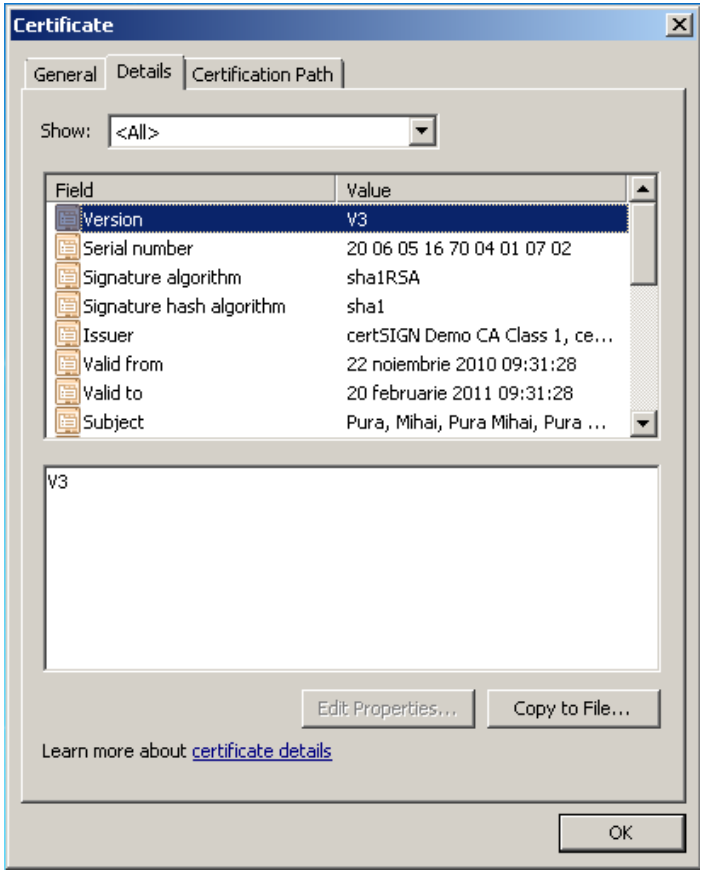
There are several categories of information:

The main scope of certificate

The SUBJECT of certificate
The ISSUER of certificate
The validity of certificate

If there is a registered private-key, pairing with this public key

Digital certificates



Digital certificates

- Extensions
  - The scope of the certificate, usage policies, constraints, etc.
  - X.509 v3 standard extensions
    - 1. key information
      - authority key identifier
      - subject key identifier
      - key usage
      - private key usage period
    - 2. policy information
      - certificate policies
      - policy mappings

Digital certificates

- Extensions
  - X.509 v3 standard extensions
    - 3. user and CA attributes
      - subject alternative name
      - issuer alternative name
      - subject directory attributes
    - 4. certification path constraints
      - basic constraints
      - name constraints
      - policy constraints

Digital certificates

- Extensions
  - Authority key identifier extension contains a key identifier, if not necessary the issuer name and the serial number
  - CRL distribution points extension contains the location where CRL may be found
  - Authority information access extension contains the repository in which the own CA certificate may be found

Digital certificates

- End-entity certificates
  - Are issued to subjects that are not CAs
  - Contain public keys used for verifying digital signatures or for performing key management
  - Subject:
    - human user
    - system (Web server or router)
  - two types:
    - User certificates
    - System certificates

Digital certificates

- **CA certificates**
  - Are issued to subjects that are CAs
  - Are part of certificate paths
  - Contain public keys used for
    - verifying digital signatures on certificates and CRLs
  - Must contain sufficient information for certificate users to construct certification paths and locate CRLs

Digital certificates

- **CA certificates**
  - Subject:
    - other CA in the same enterprise
    - a CA in other enterprise
    - a bridge
  - three types
    - CA certificates within an enterprise PKI
    - CA certificates between enterprise PKIs
    - CA certificates in a Bridge CA Environment

Digital certificates

- Self-signed CA certificates (self-issued)
  - Issuer and Subject are the same
  - Used to
    - establish trust points
    - distribute a new signing public key
    - modify the certificate policies supported in a PKI

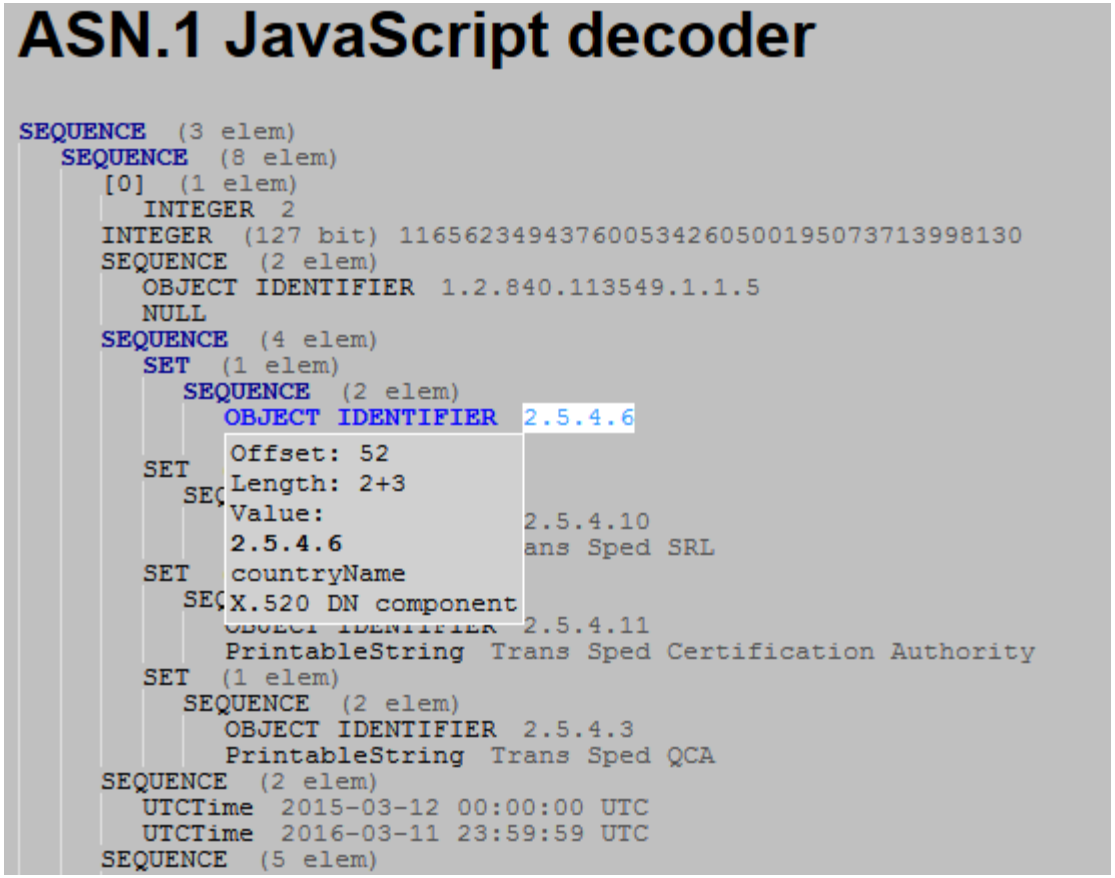
ASN.1

- ASN.1 – Abstract Syntax Notation One
  - a standard and notation that describes
    - rules
    - and structures

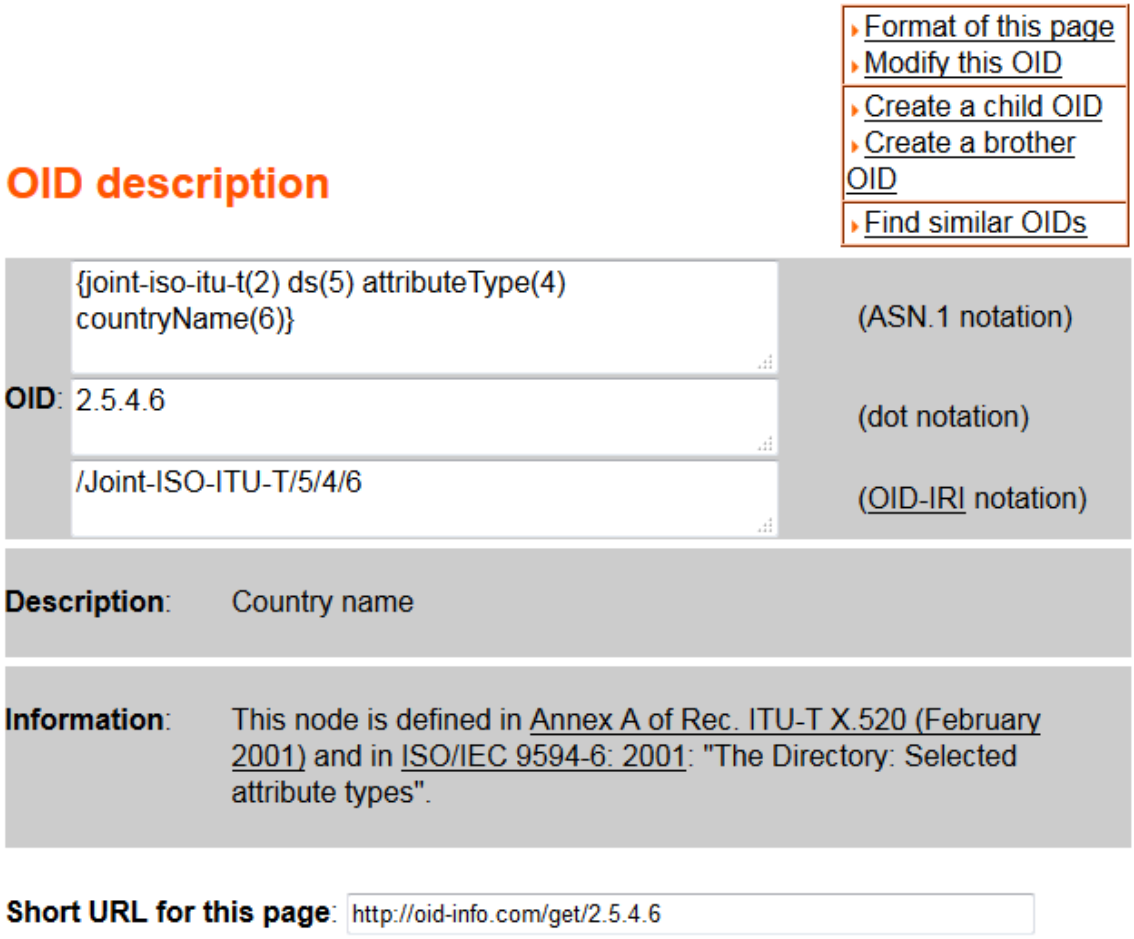
    for representing, encoding, transmitting, and decoding data in telecommunications and computer networking

- ASN.1 decoder:
  - https://holtstrom.com/michael/tools/asn1decoder.php
  - http://phpseclib.sourceforge.net/x509/asn1parse.php
  - https://lapo.it/asn1js/
- Object Identifier (OID) Repository
  - http://www.oid-info.com/

Digital certificates – ASN.1



Digital certificates – ASN.1



- Format of this page
- Modify this OID
- Create a child OID
- Create a brother OID
- Find similar OIDs

OID description

OID:	{joint-iso-itu-t(2) ds(5) attributeType(4) countryName(6)}	(ASN.1 notation)
	2.5.4.6	(dot notation)
	/Joint-ISO-ITU-T/5/4/6	(OID-IRI notation)

Description: Country name

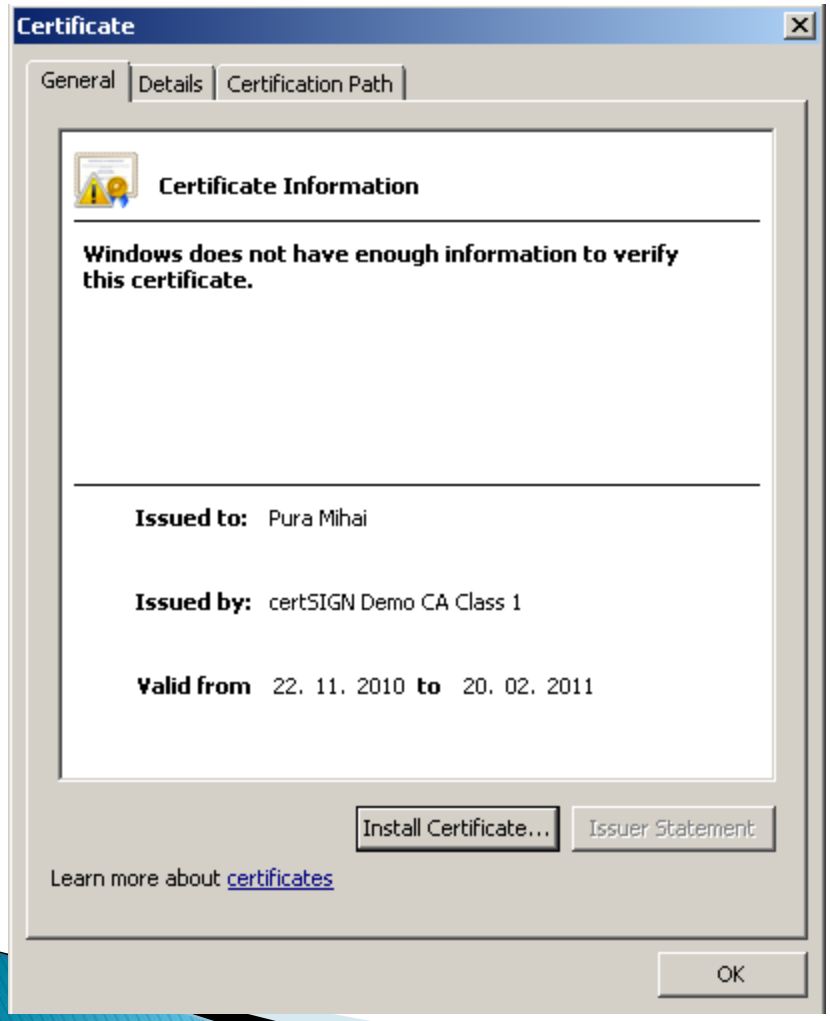
Information: This node is defined in Annex A of Rec. ITU-T X.520 (February 2001) and in ISO/IEC 9594-6: 2001: "The Directory: Selected attribute types".

Short URL for this page: http://oid-info.com/get/2.5.4.6

Certification Path

- CA's digital signature on the certificate
  - is the guaranty that the digital certificate is owned by the person whose identification information are contained in it
- To verify the bond between the public key and the identity of the owner
  - verify the signature of the CA over the certificate
- The digital certificate of the issuer (a CA) is needed

Certification Path



Certification Path

- Certification path
  - all the digital certificates needed to verify someone's digital certificate
  - each certificate has a *certification path starting from a ROOT-CA*
  - usually, Root-CA certificate is a trusted point in the *system*

- The root certificate for this path is a self-signed certificate
  - The self-signed certificate cannot be further validated
  - So another verification element is added: the trust in a CA

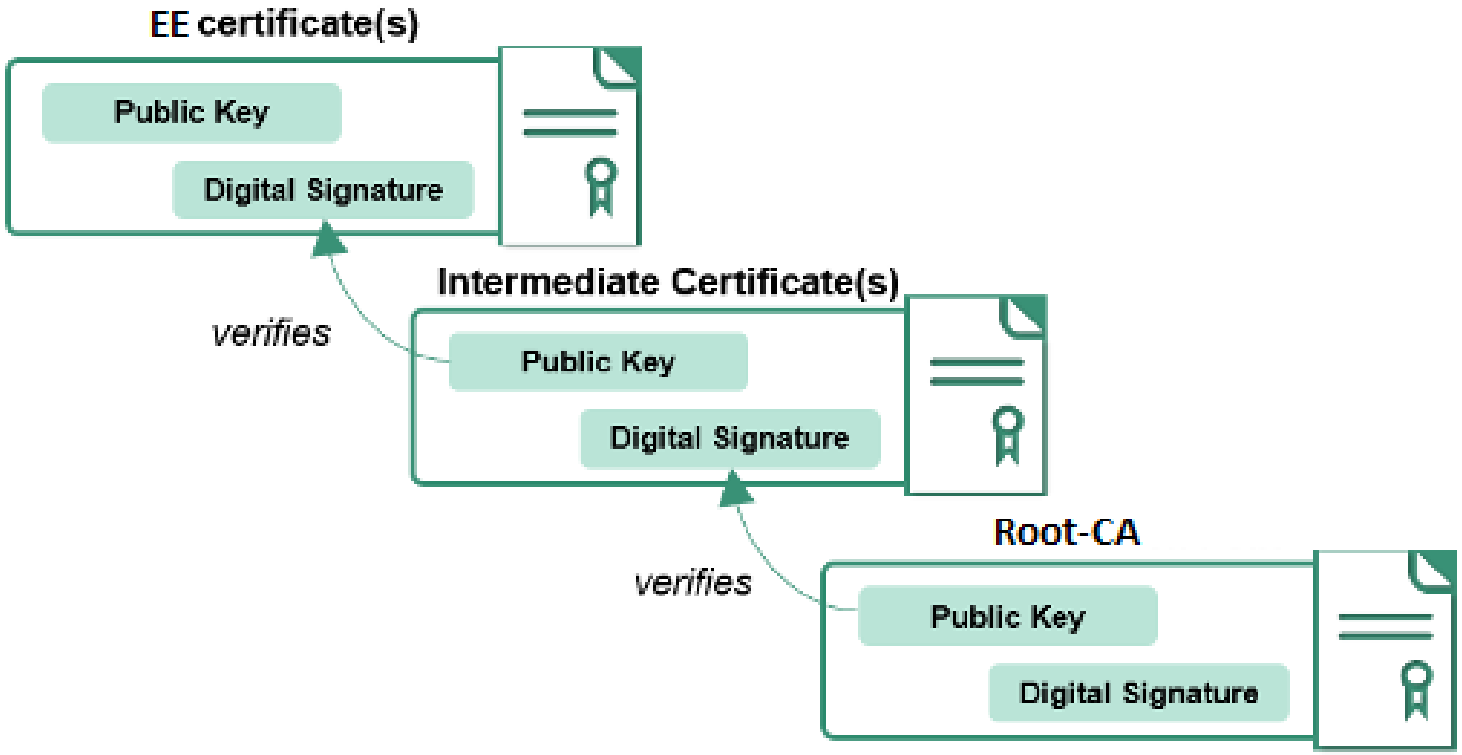
Certification Path

- If a CA is trusted it does not matter that it had sign its own certificate
- If a CA is not trusted, the user will be informed about it
- The digital certificates of a CA can be downloaded from their web site

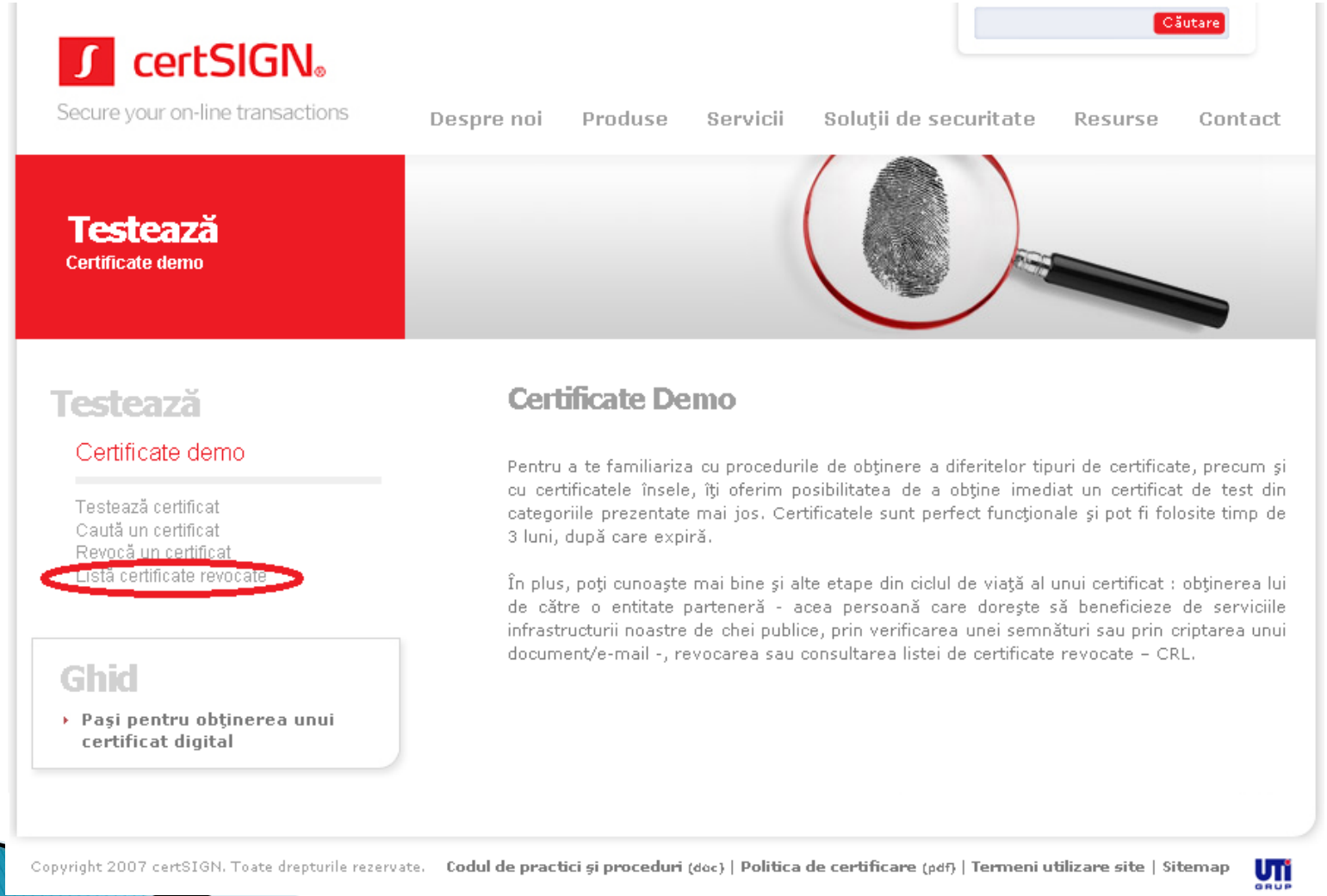
Certification Path

- An ordered sequence of certificates between
  - the end entity
  - and the trusted point in the hierarchy (Root-CA)
- The result is a certificate chain that
  - begins at the end entity
  - and ends at the Root-CA

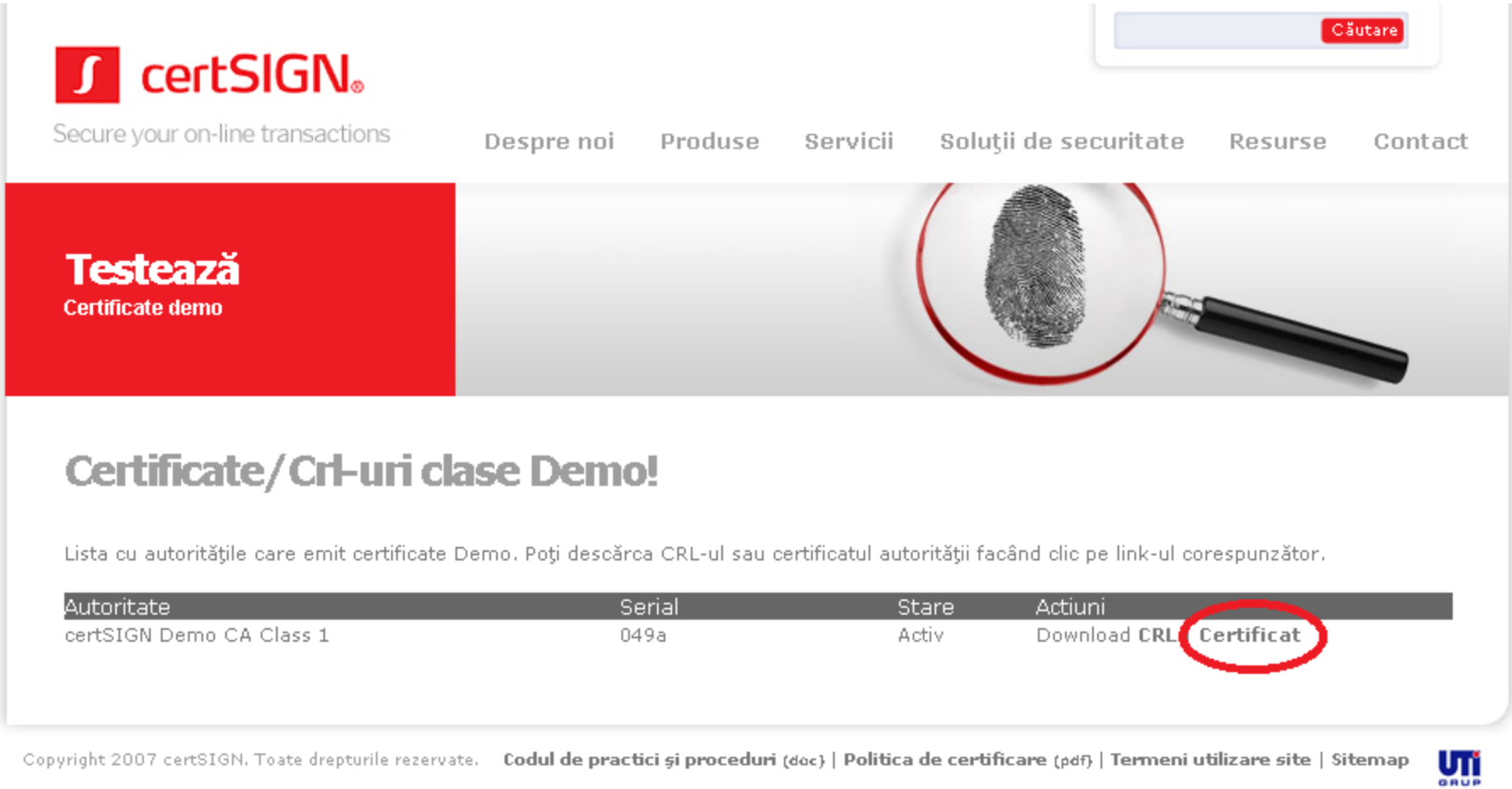
Certification Path



Certification Path



Certification Path



Certificate stores

- Microsoft Windows operating systems manages digital certificates through special store places
  - certificate stores
- To execute operations over these stores, the Certificates snap-in should be used
- Start, Run: **certmgr.msc**

MS Windows certificate stores

- A system store
  - a collection that consists of one or more physical sibling stores
  - for each system store, there are predefined physical sibling stores
- All certificates in those physical stores are available through the logical system store collection
- For each system store location, the predefined systems stores are: MY, Root, Trust, CA
- In CERT_SYSTEM_STORE_CURRENT_USER, there is also a predefined UserDS store. A smart card store is planned for this location

MS Windows certificate stores

- The system stores:
- CERT_SYSTEM_STORE_CURRENT_USER
- CERT_SYSTEM_STORE_LOCAL_MACHINE
- CERT_SYSTEM_STORE_CURRENT_SERVICE
- CERT_SYSTEM_STORE_SERVICES
- CERT_SYSTEM_STORE_USERS
- CERT_SYSTEM_CURRENT_USER_GROUP_POLICY
- CERT_SYSTEM_LOCAL_MACHINE_GROUP_POLICY
- CERT_SYSTEM_STORE_LOCAL_MACHINE_ENTERPRISE

MS Windows certificate stores

- CERT_SYSTEM_STORE_CURRENT_USER system stores are at the following registry location:

- **HKEY_CURRENT_USER**
- **Software**
- **Microsoft**
- **SystemCertificates**

- The predefined physical stores associated with those system stores are as follows

http://msdn.microsoft.com/en-us/library/windows/desktop/aa388136%28v=vs.85%29.aspx

MS Windows certificate stores

System store	Physical store
MY	.Default
Root	.Default .LocalMachine .SmartCard
Trust	.Default .GroupPolicy .LocalMachine
CA	.Default .GroupPolicy .LocalMachine
UserDS	.UserCertificate

Create MMC snap-in to manage certificate stores

- 1. Run "**mmc.exe**" from a command line window. A blank Console window shows up.
- 2. Click "Add/Remove Snap-in" from the "File" menu. The "Add/Remove Snap-in" dialog box shows up.
- 3. Click the "Add" button. A list of "Available standalone snap-ins" shows up.

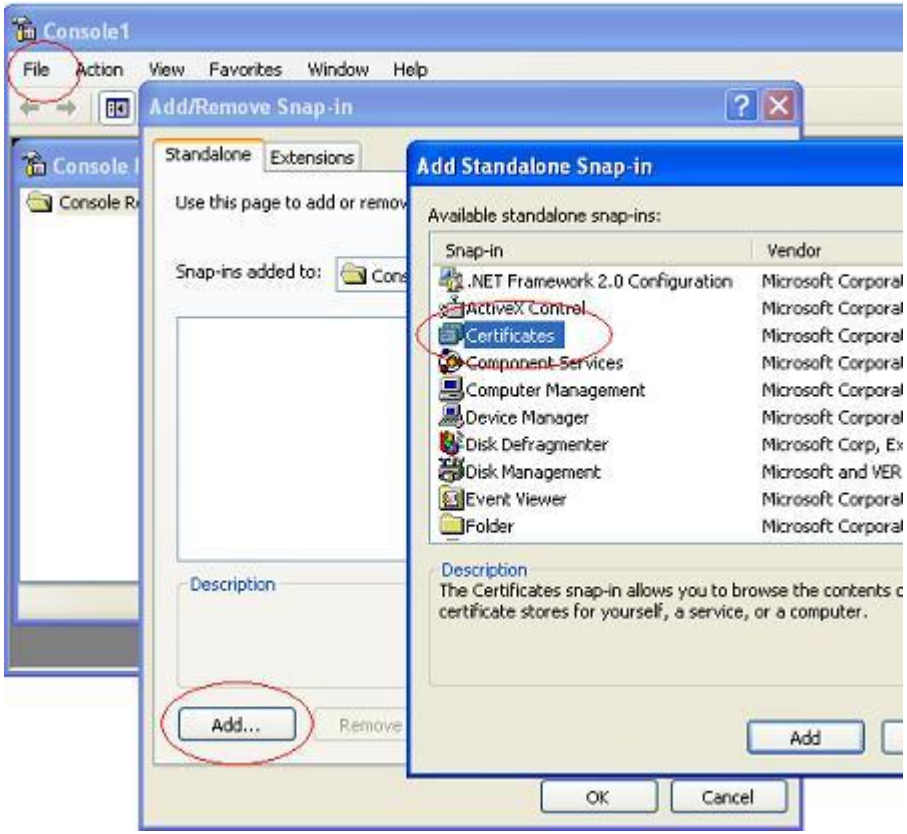
Create MMC snap-in to manage certificate stores

- 4. Select "Certificates" and click "Add". The "Certificates snap-in" options dialog box shows up.
- 5. Select "Computer account" and click "Next". The "Select Computer" dialog box shows up.
- 6. Select "Local computer" and click "Finish". The "Certificates" snap-in is added.

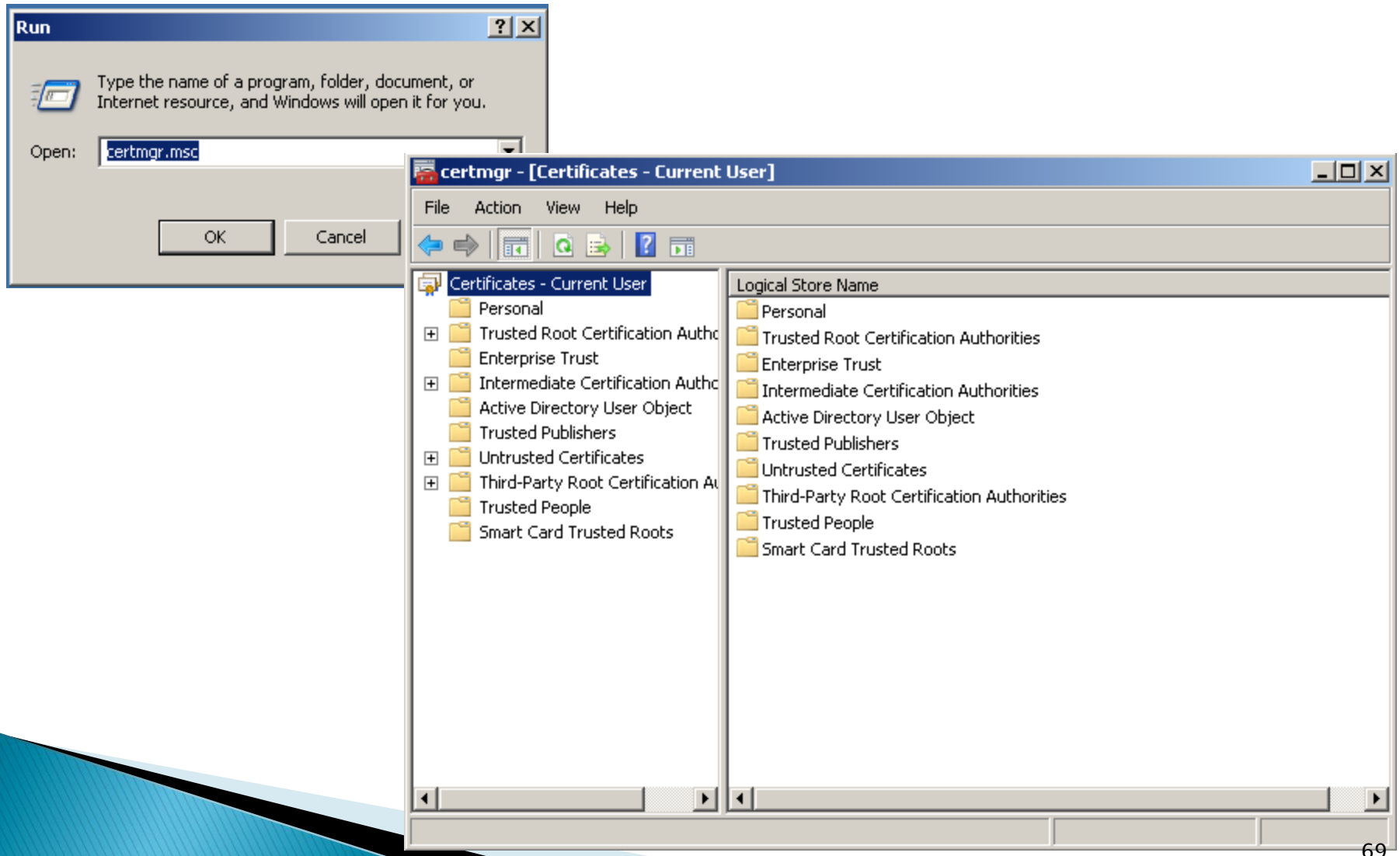
Create MMC snap-in to manage certificate stores

- 7. Close the "Available standalone snap-ins" list dialog box.
- 8. Click "OK" on the "Add/Remove Snap-in" dialog box to accept the new snap-in.
- 9. Click "Save" from the "File" menu. The "Save As" dialog box shows up.
- 10. Enter "MyCertificatesConsole.msc" as the file name and click "Save".

Create MMC snap-in to manage certificate stores



MS Windows certificate stores



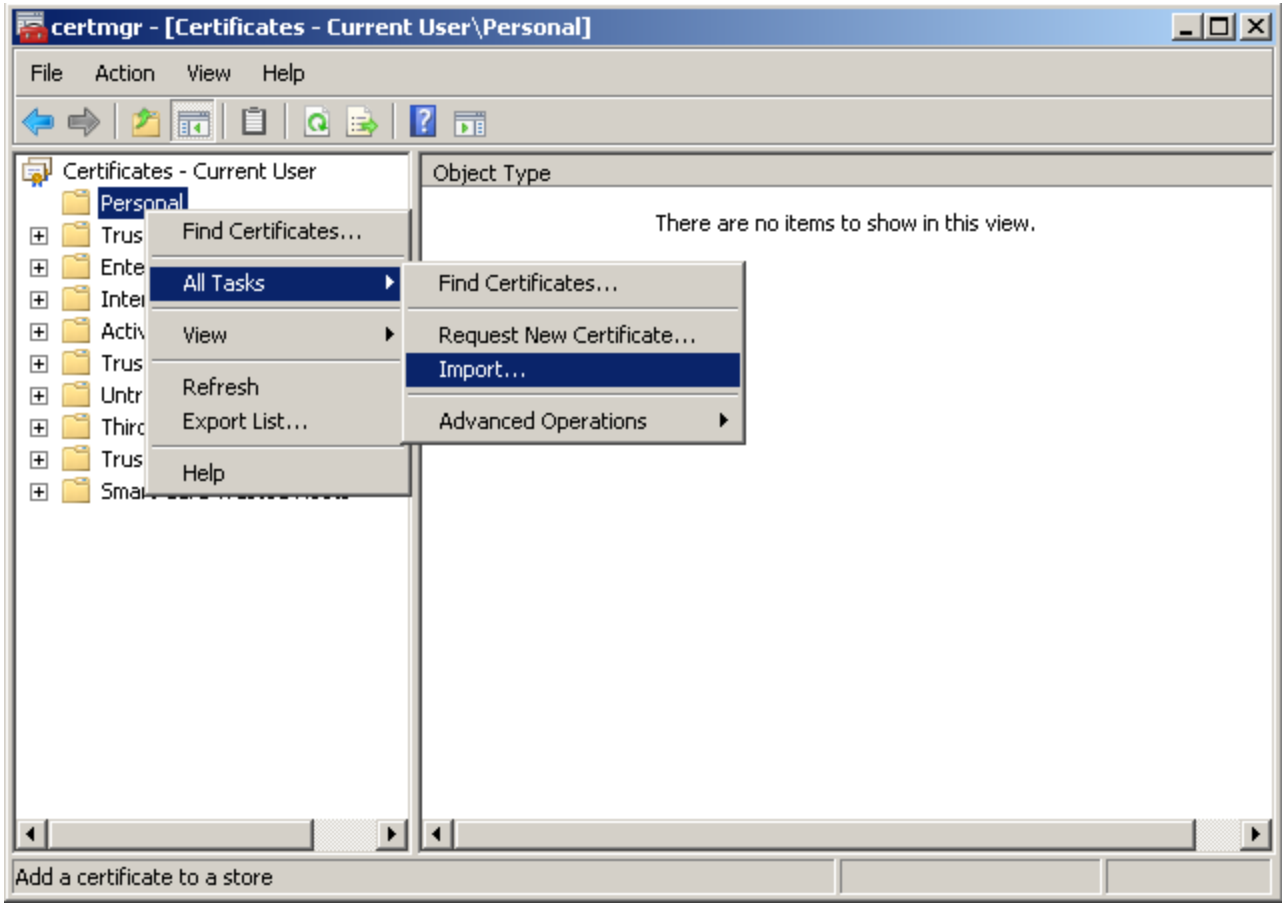
MS Windows certificate stores

- **Personal**
  - stores all the certificates for which the current user has the corresponding private keys:
    - certificates issued for the current user
    - for that computer
    - or for a service that runs on it

MS Windows certificate stores

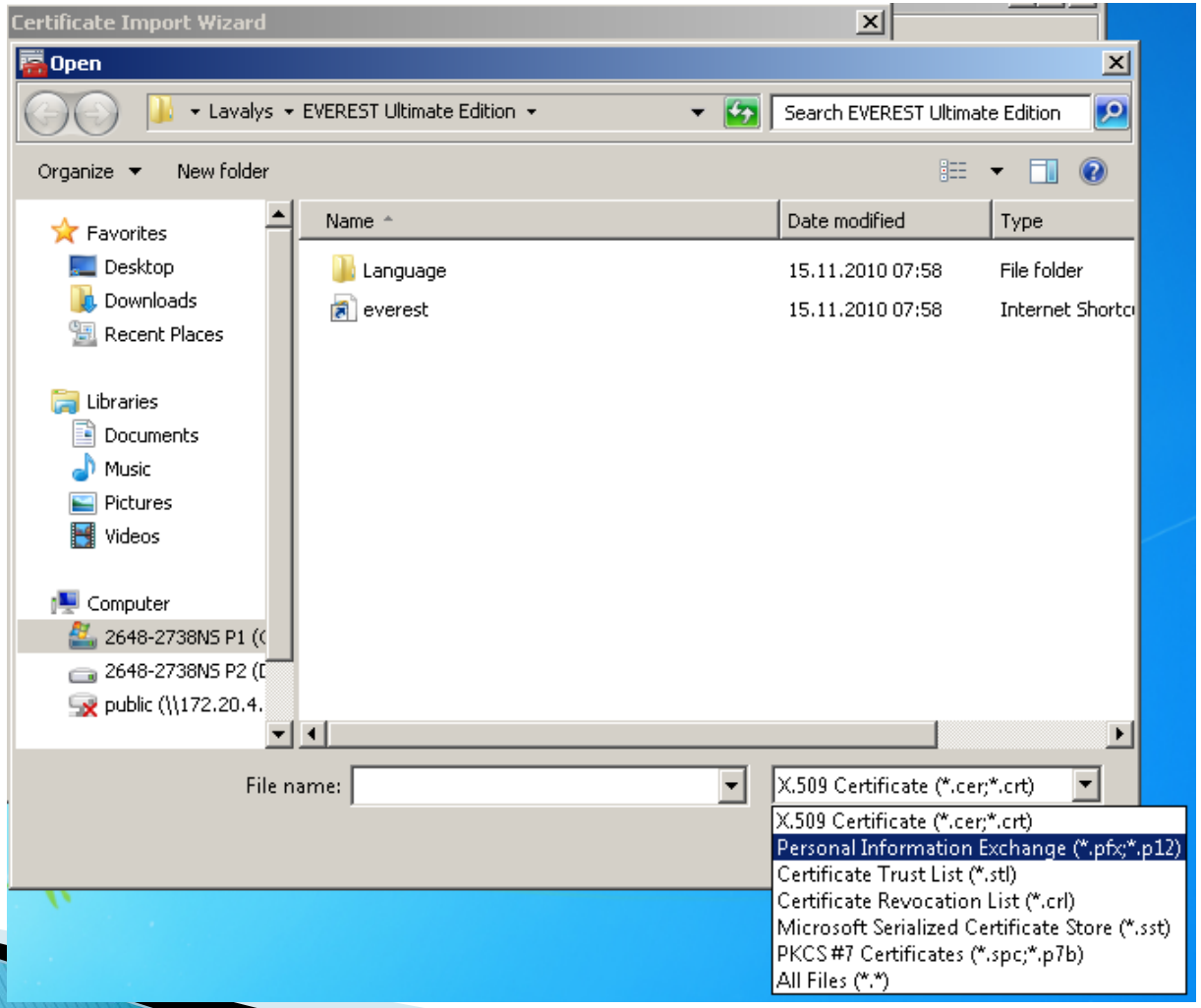
- **Trusted Root Certification Authorities**
  - stores the digital certificates of the CAs that are implicit trusted
- **Trusted People**
  - stores certificates issued for persons or entities that are trusted

Certificate import

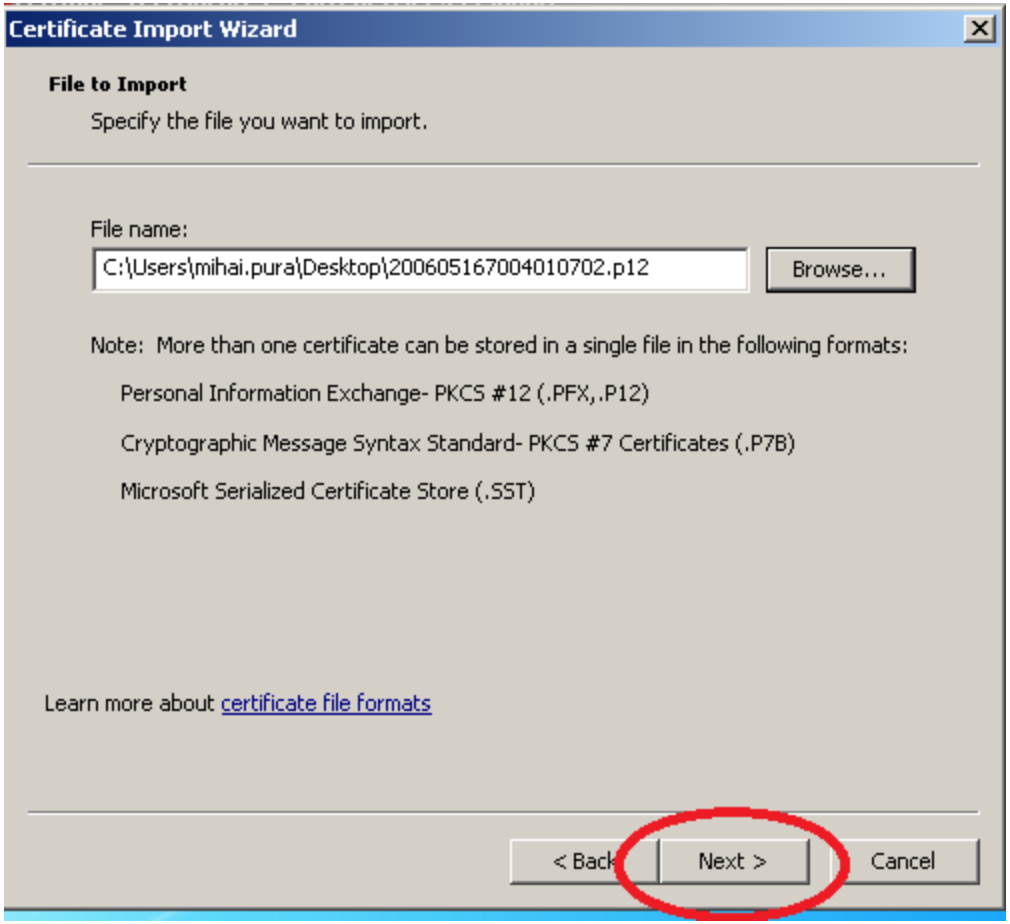


Certificate import

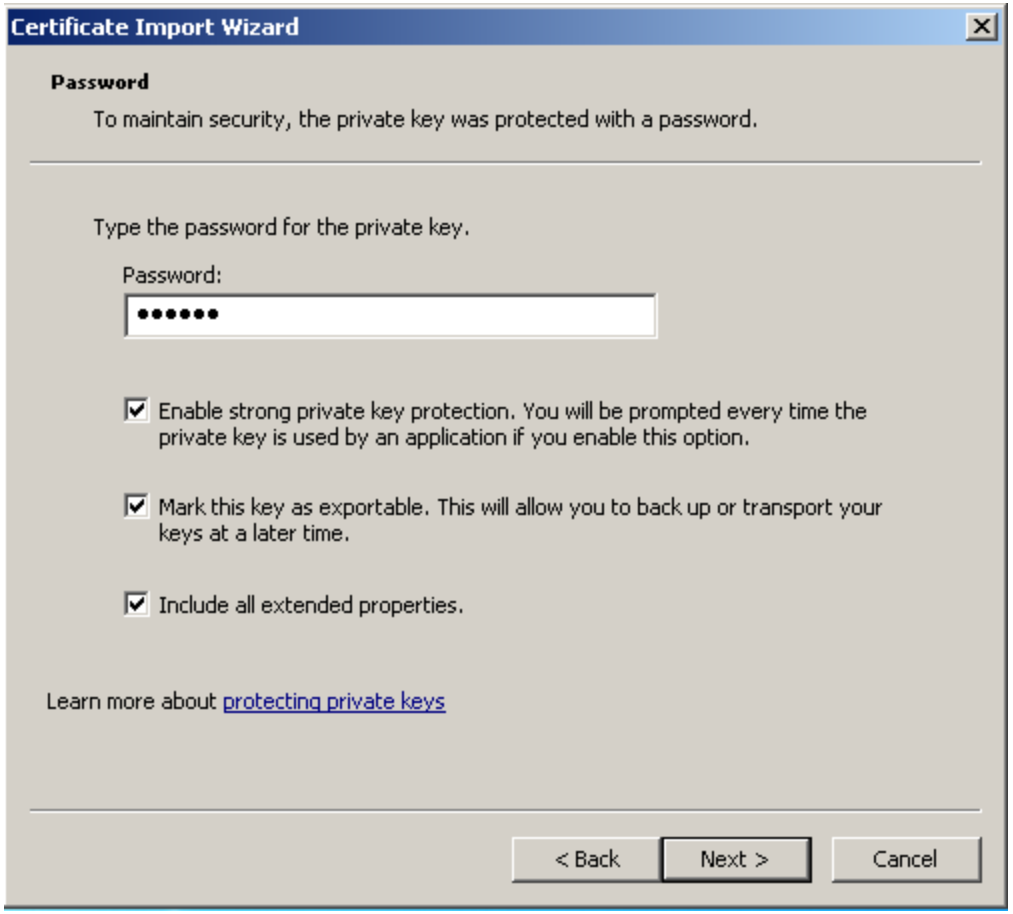
Certificate import



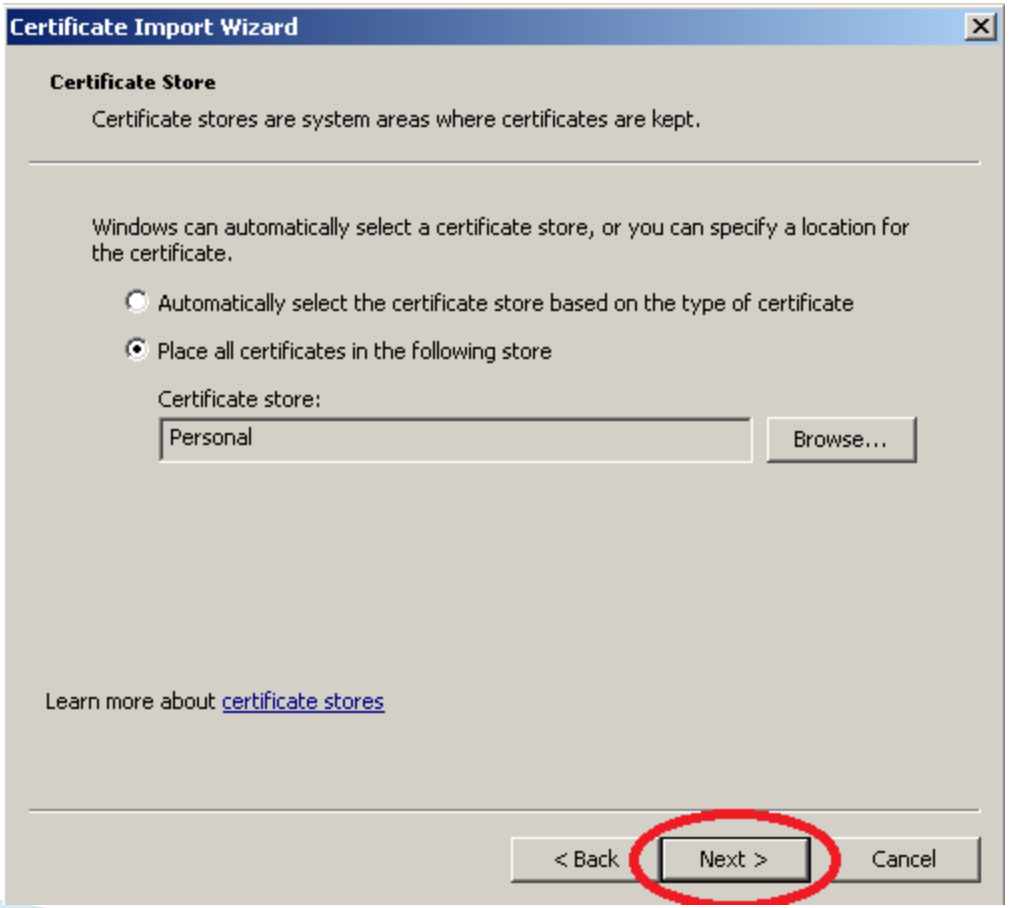
Certificate import



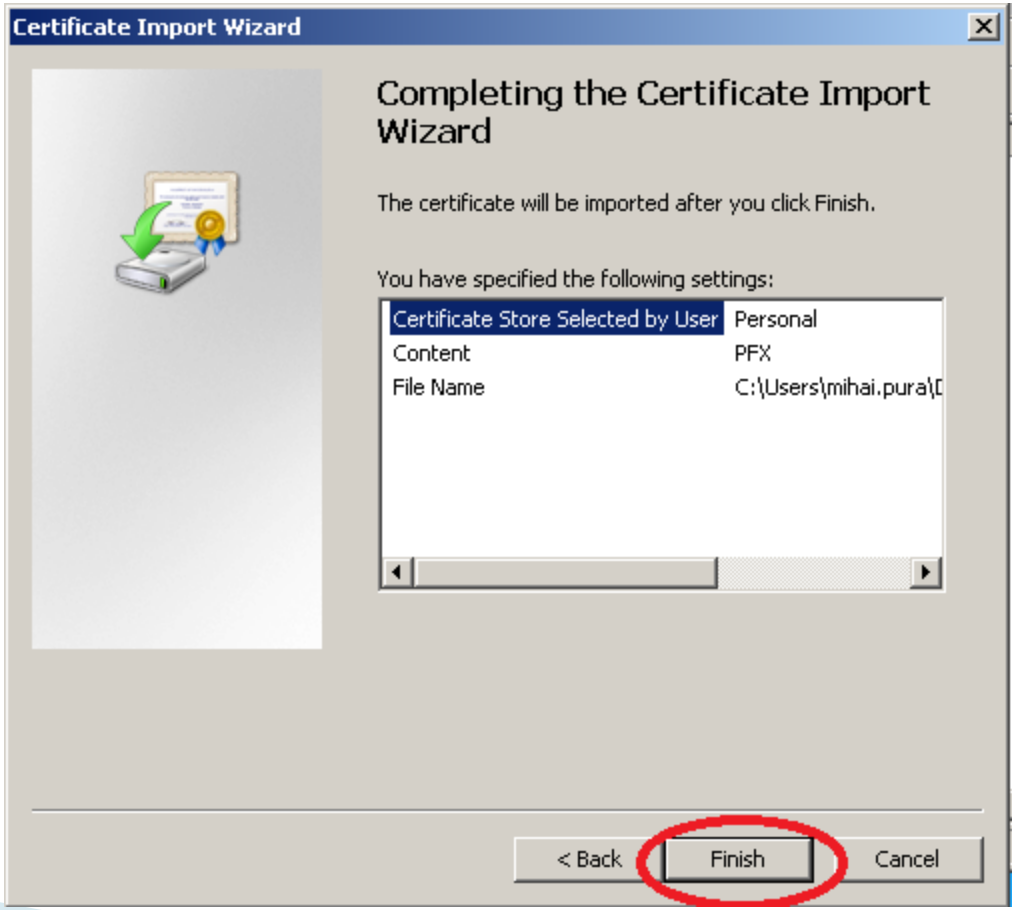
Certificate import



Certificate import



Certificate import



Certificate import

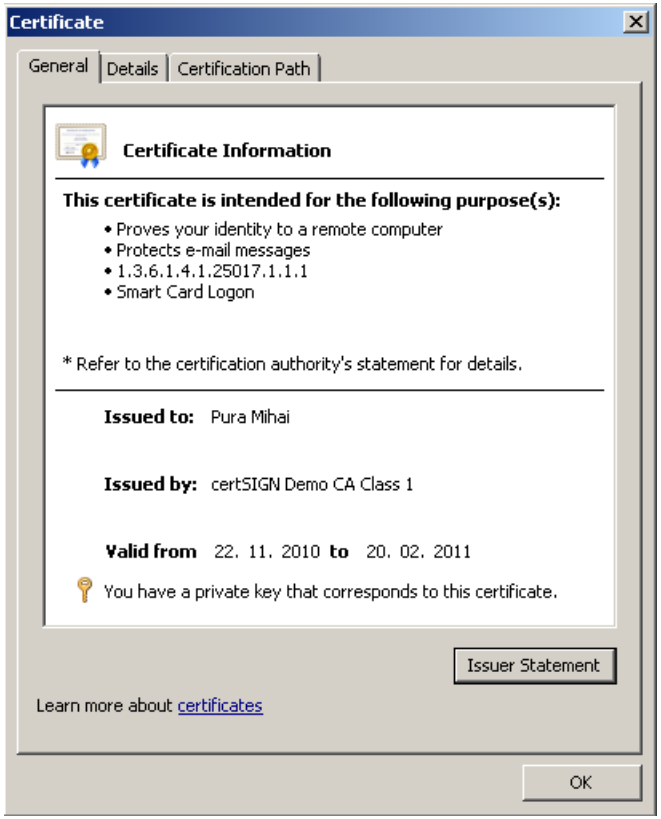
Certificate import for the CA

Certification Path

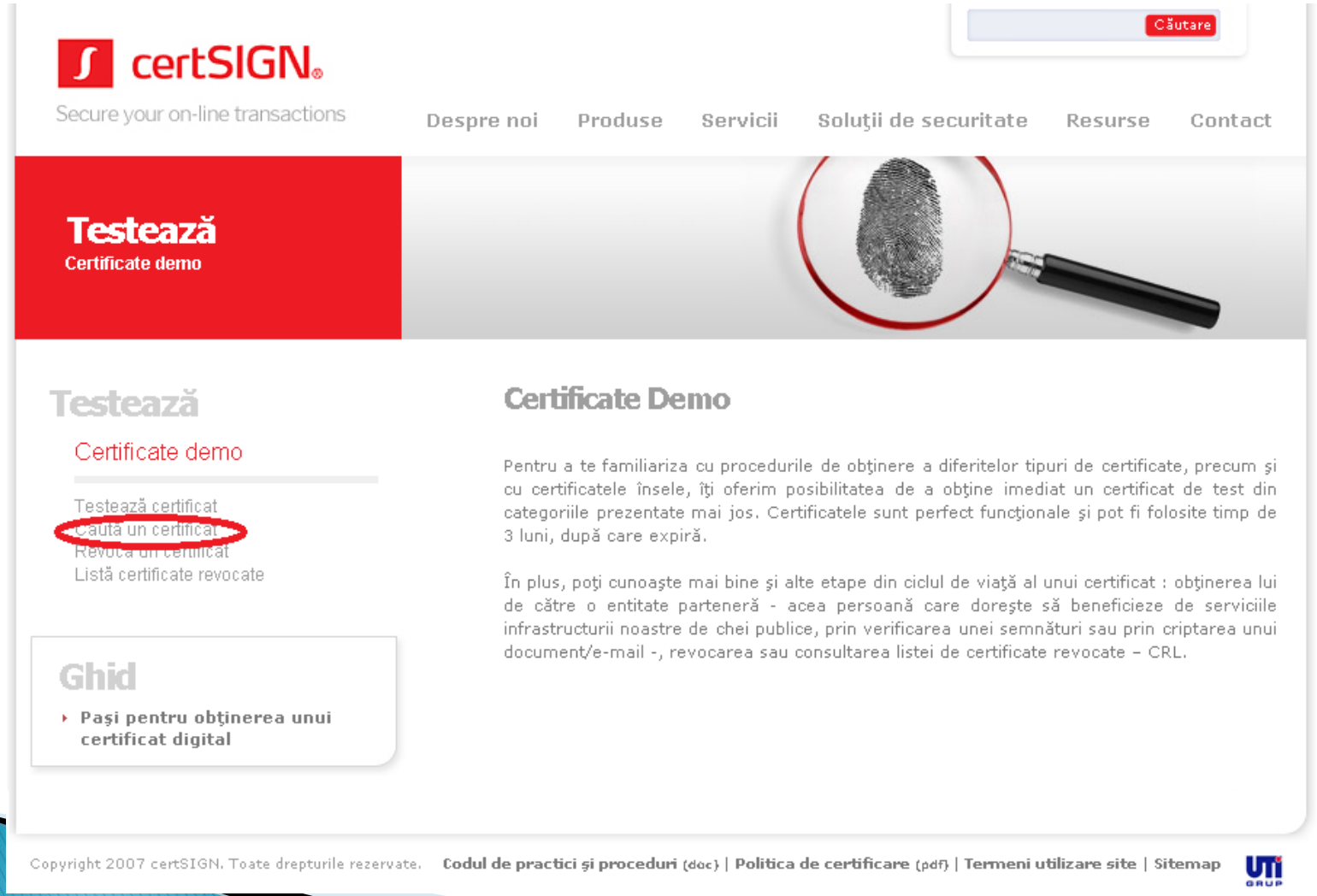
Certification Path

- To mark the CA as trusted
  - its digital certificates should be imported to the **Trusted Root Certification Authorities** store

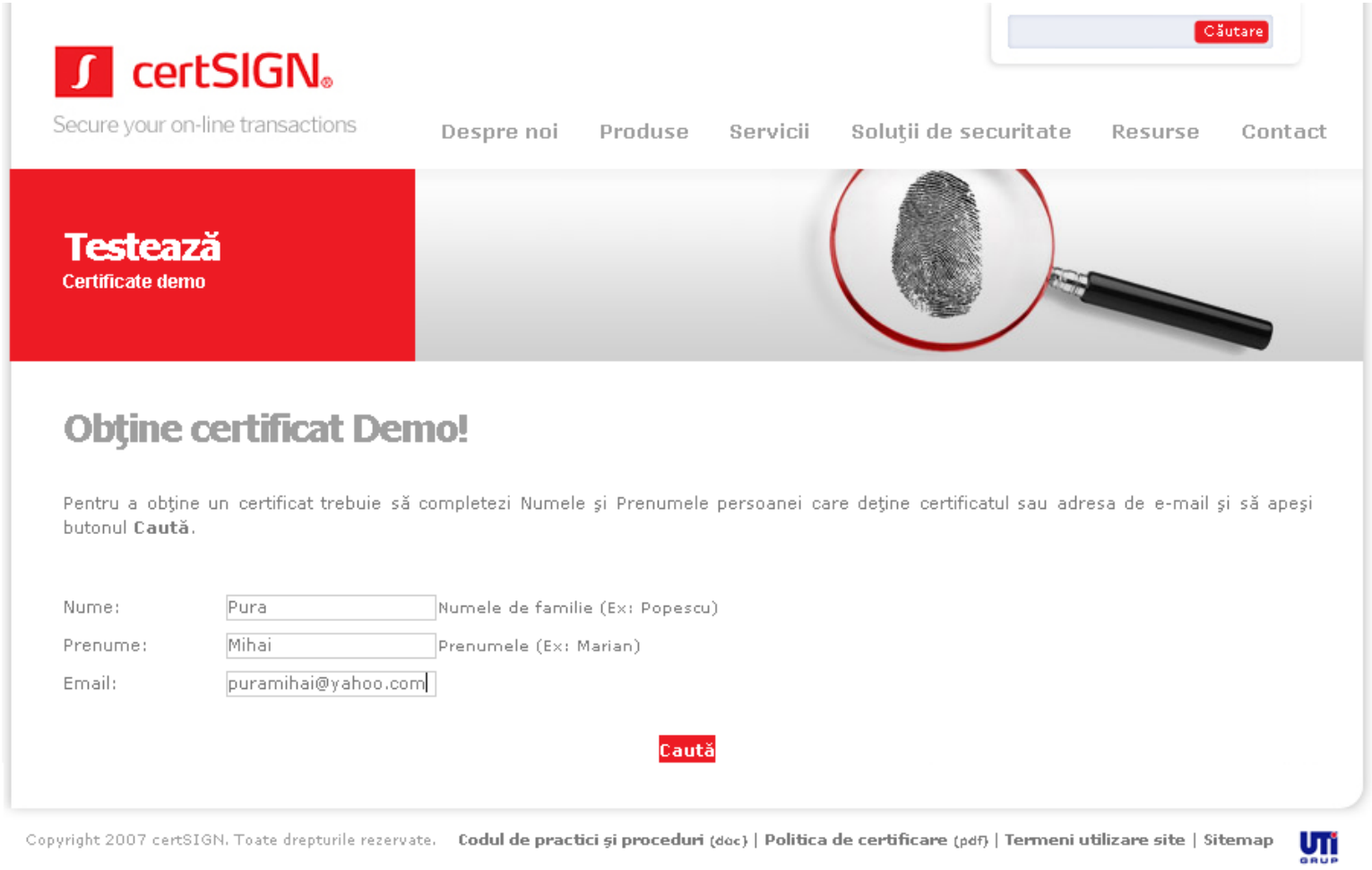
Certification Path



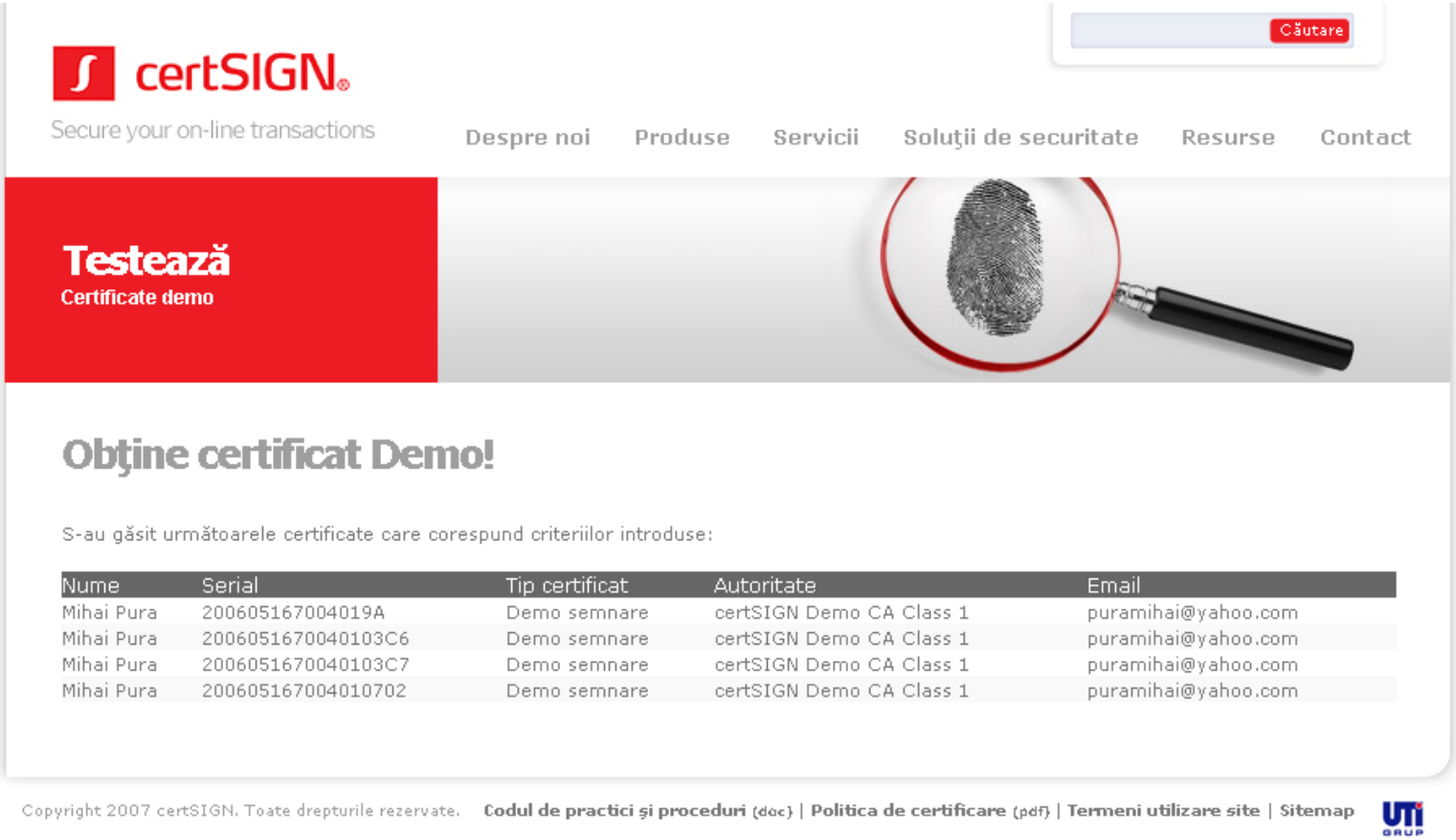
Certificate search



Certificate search



Certificate search



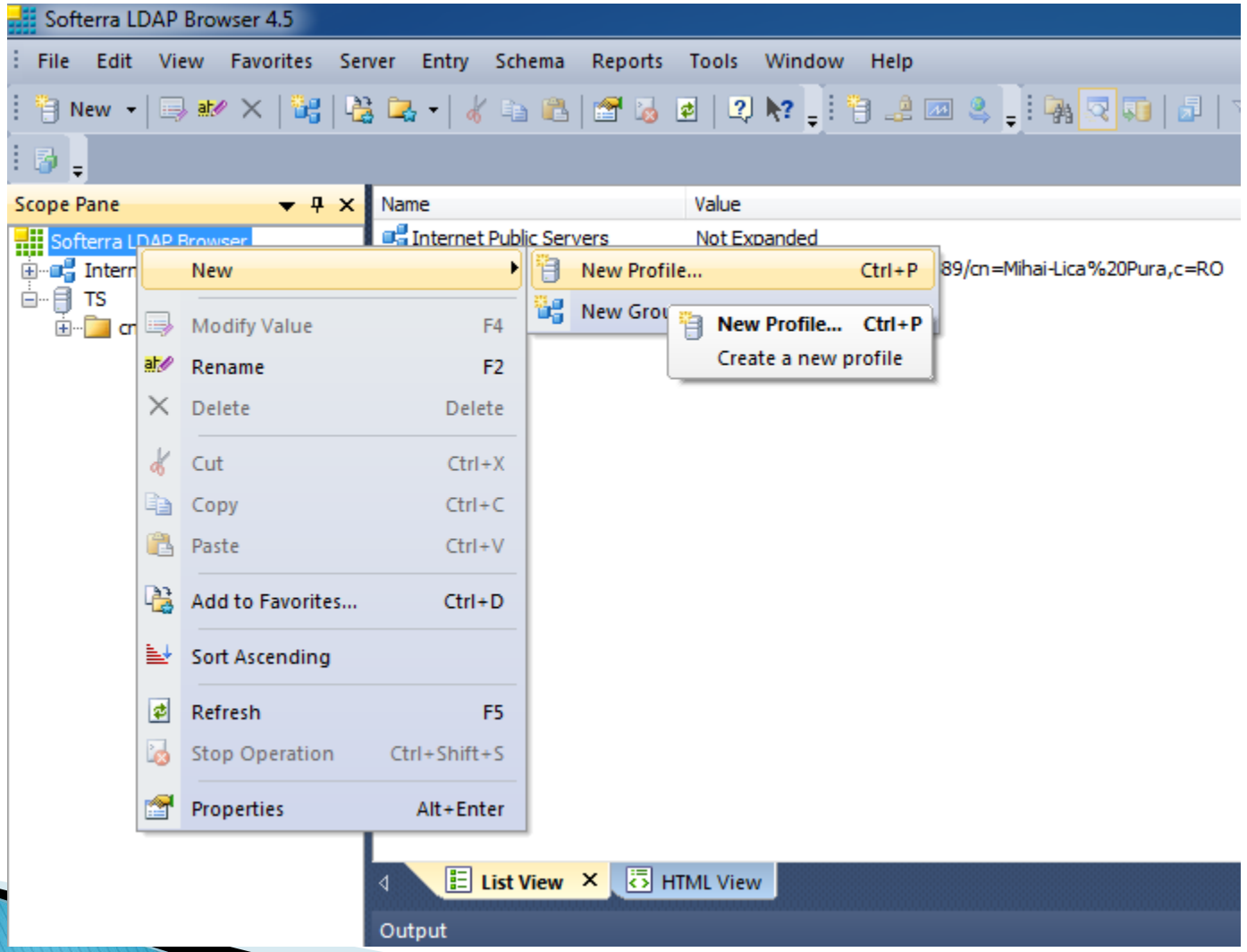
Certificate search – LDAP

- LDAP – Lightweight Directory Access Protocol
  - application protocol for querying and modifying data of directory services implemented in Internet Protocol networks
- A directory is a set of objects with attributes organized logically in a hierarchical manner
- Exemplificare
  - **Softerra LDAP Browser**
  - http://www.ldapbrowser.com/download.htm

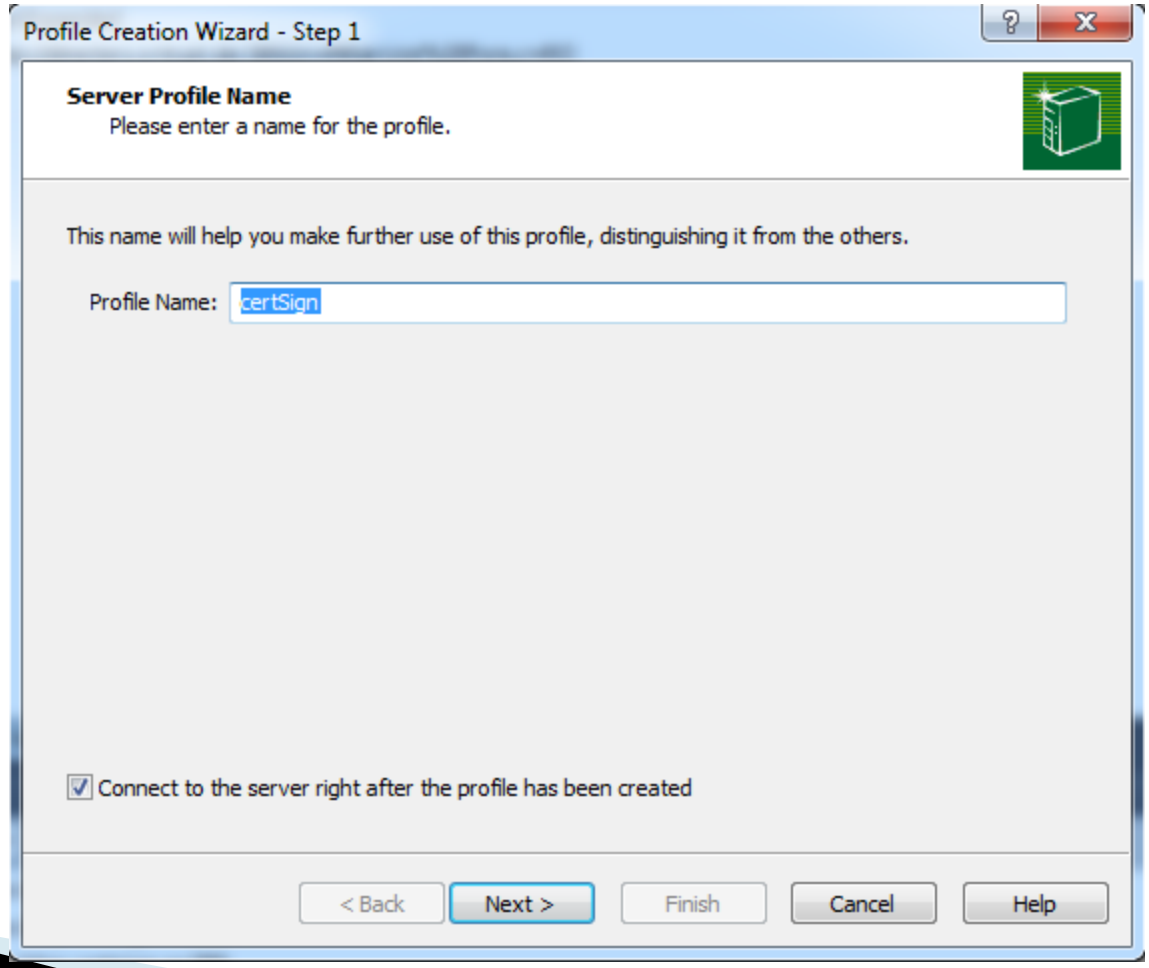
LDAP

- RFC 2587 – Internet X.509 Public Key Infrastructure LDAPv2 Schema
  - https://www.ietf.org/rfc/rfc2587.txt
  - defines the attributes and object classes to be supported by an LDAP CR server
- RFC 2559 – Internet X.509 Public Key Infrastructure Operational Protocols – LDAPv2
  - https://tools.ietf.org/html/rfc2559
  - defines the access method to a repository with which an End-Entity or a CA can retrieve or modify the certificate and CRL information stored in a CR

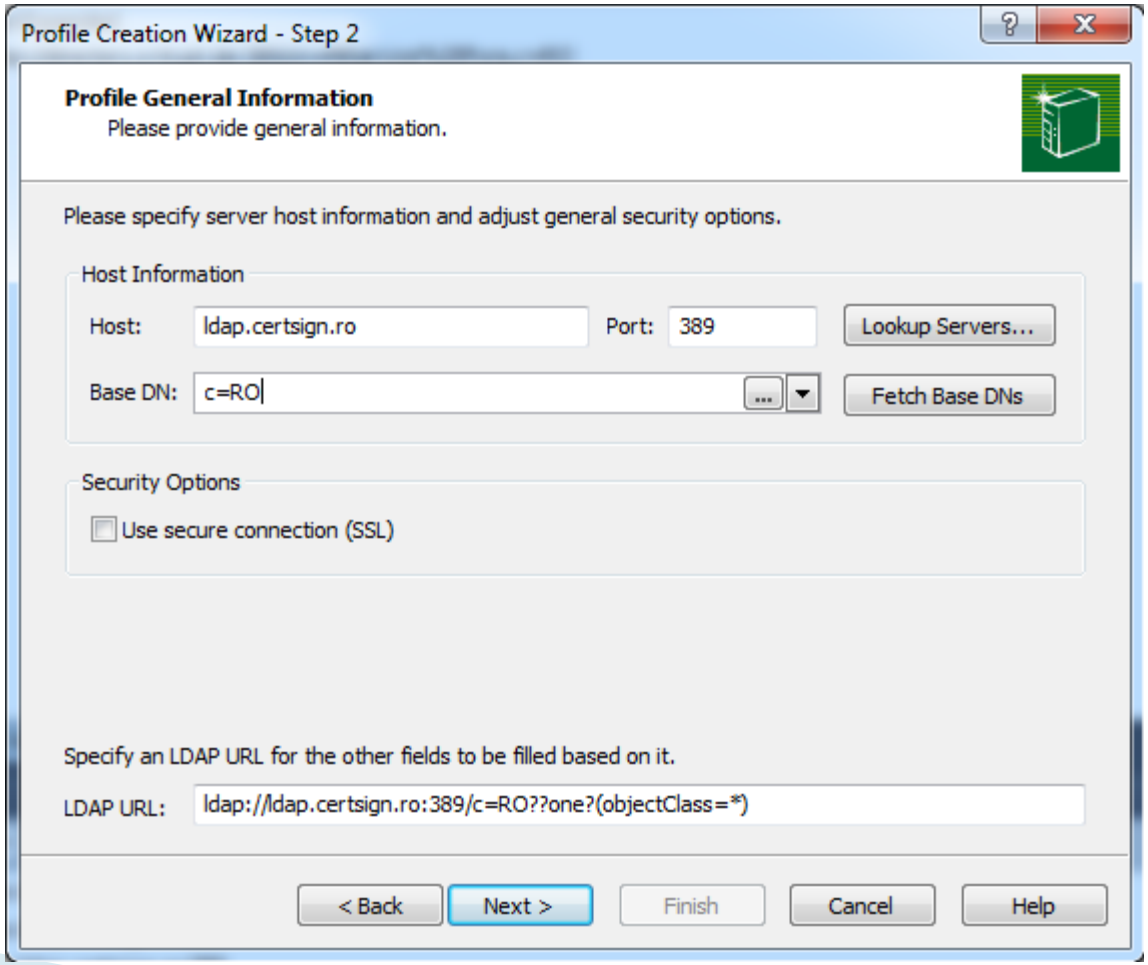
Softerra LDAP Browser



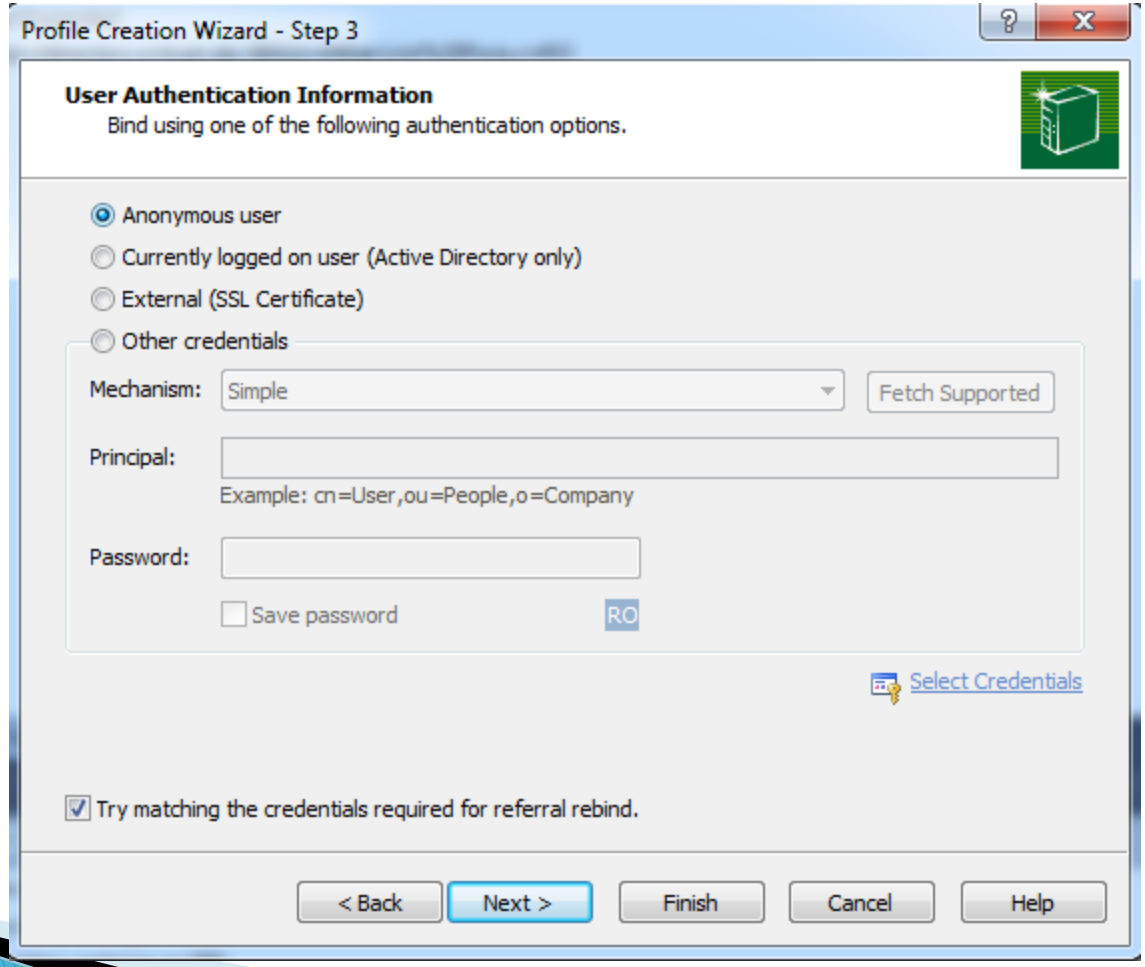
Softerra LDAP Browser



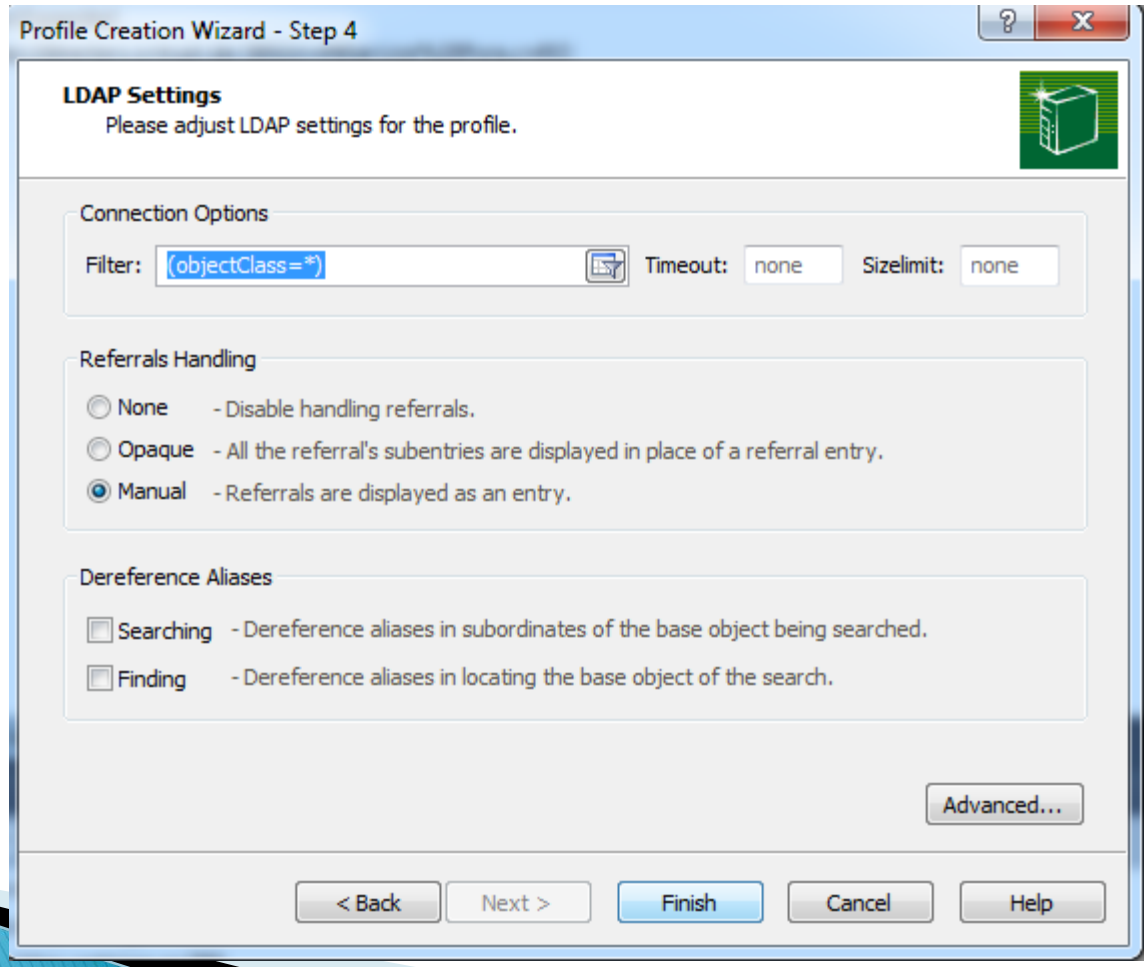
Softerra LDAP Browser



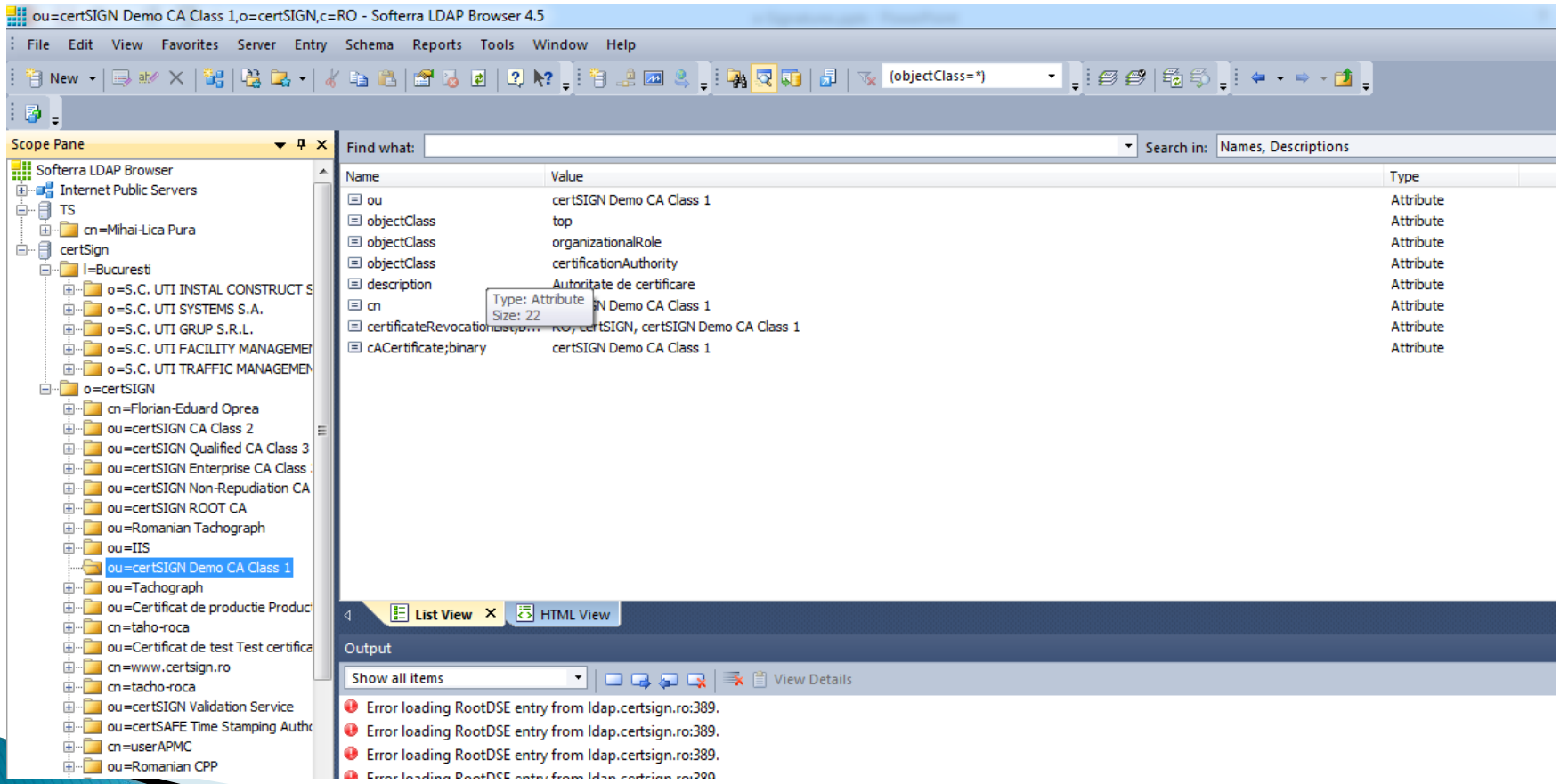
LDAP – search for and save a digital certificate



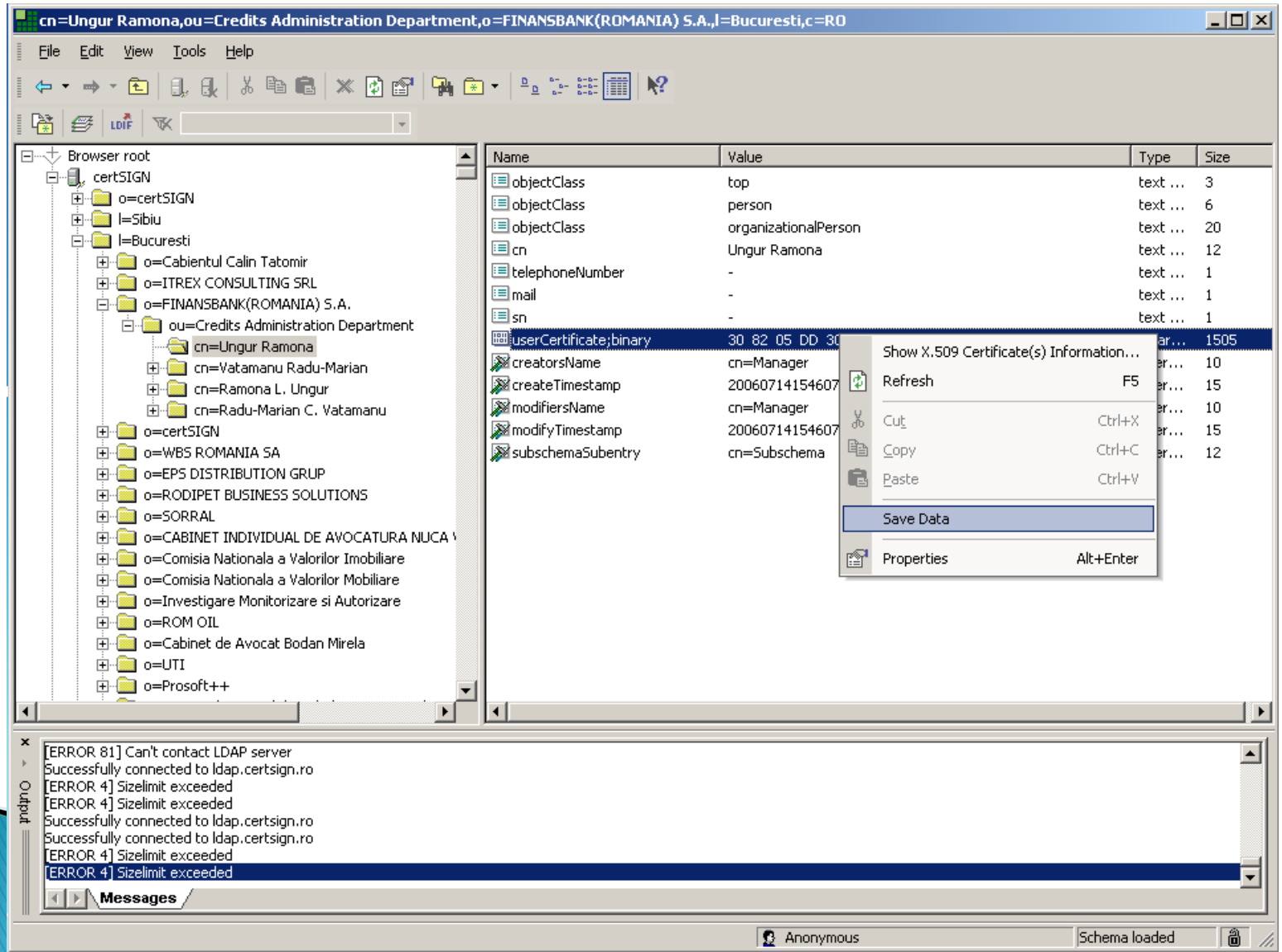
LDAP – search for and save a digital certificate



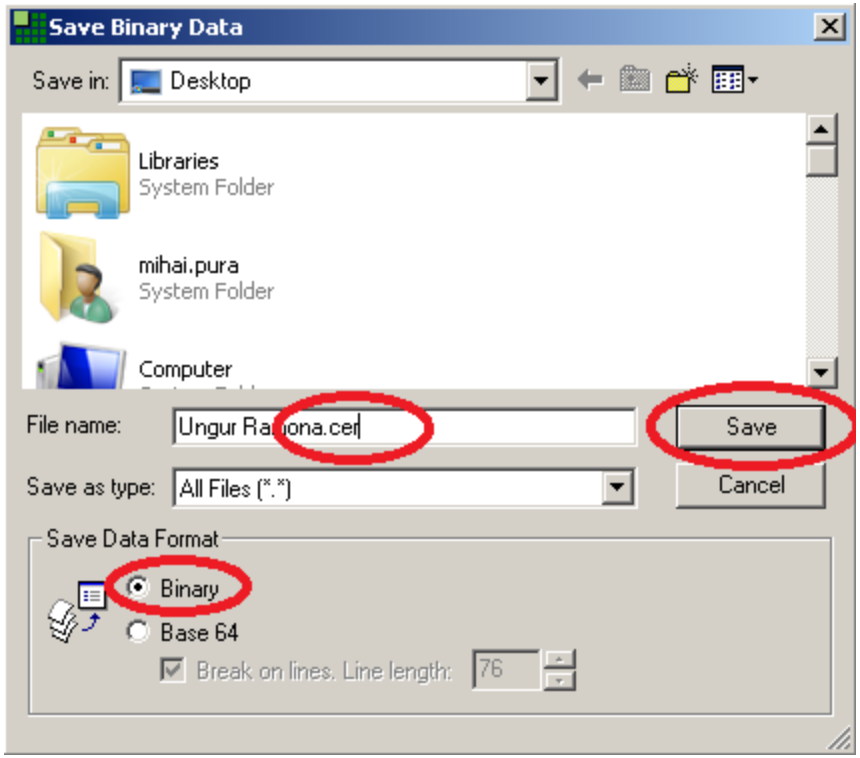
LDAP – search for and save a digital certificate



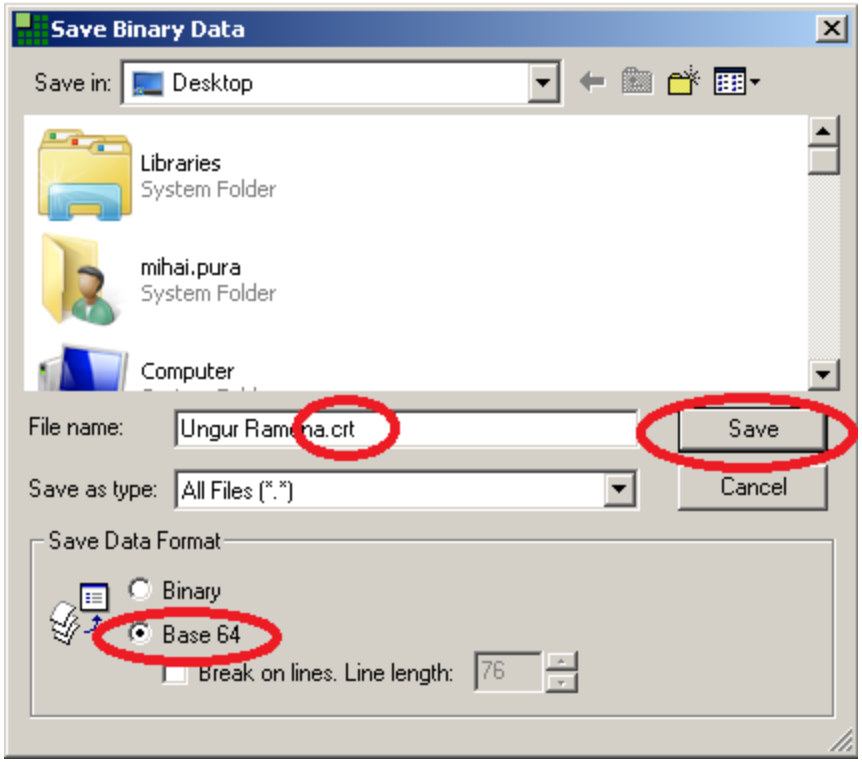
LDAP – search for and save a digital certificate



LDAP – search for and save a digital certificate

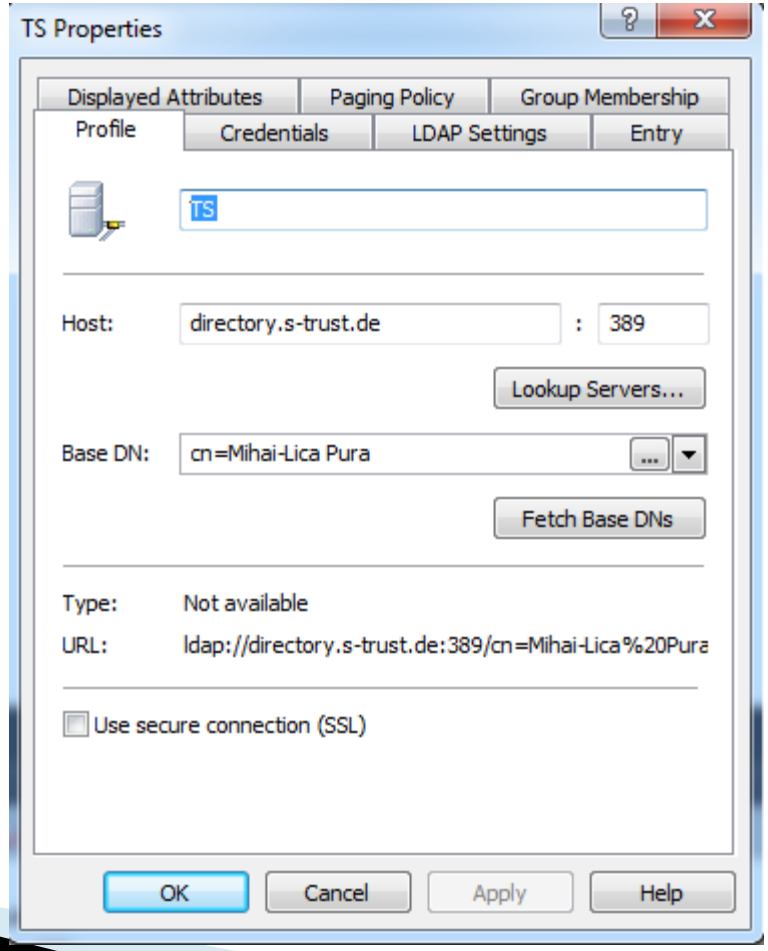


LDAP – search for and save a digital certificate

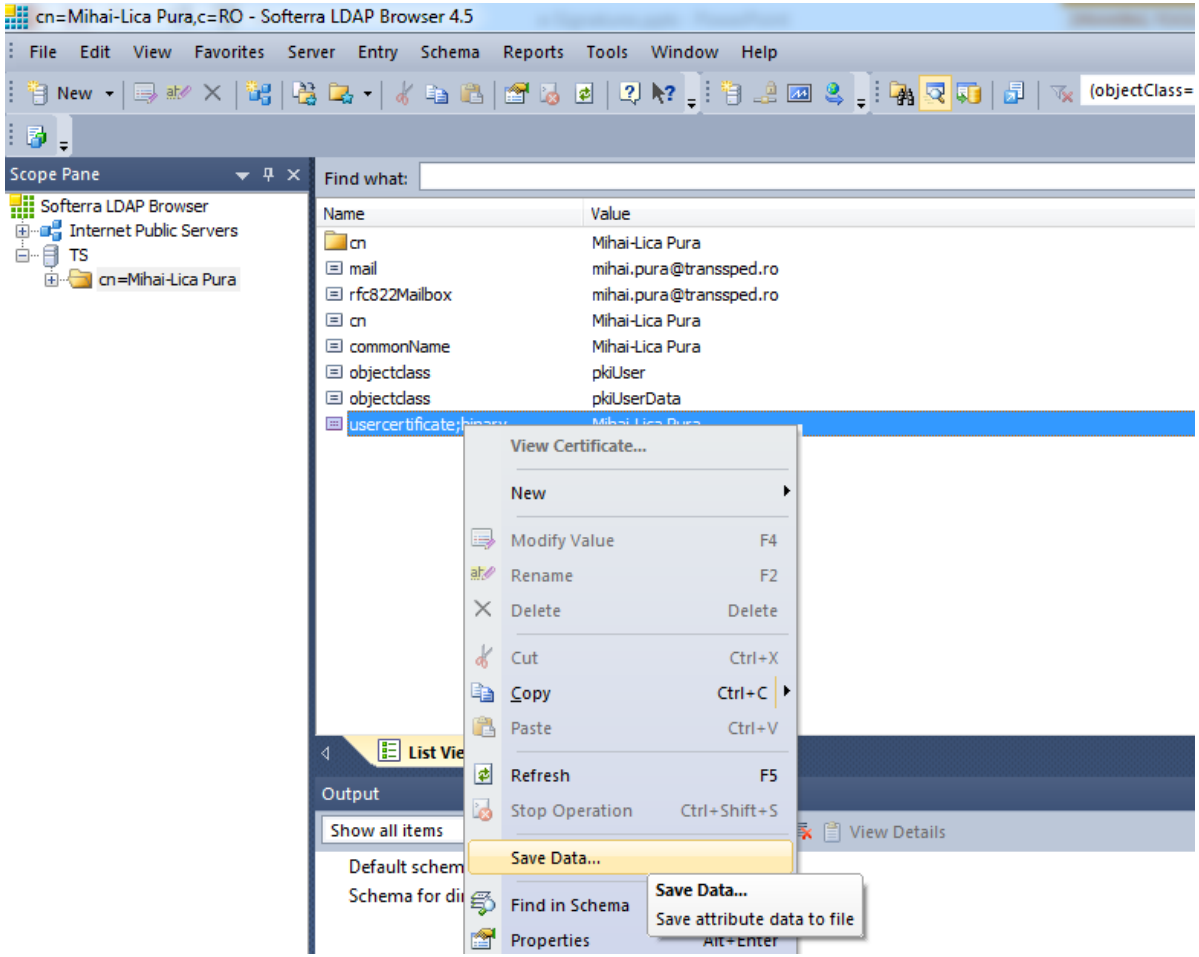


Trans Sped LDAP Server

- ldap://directory.s-trust.de



Trans Sped LDAP Server



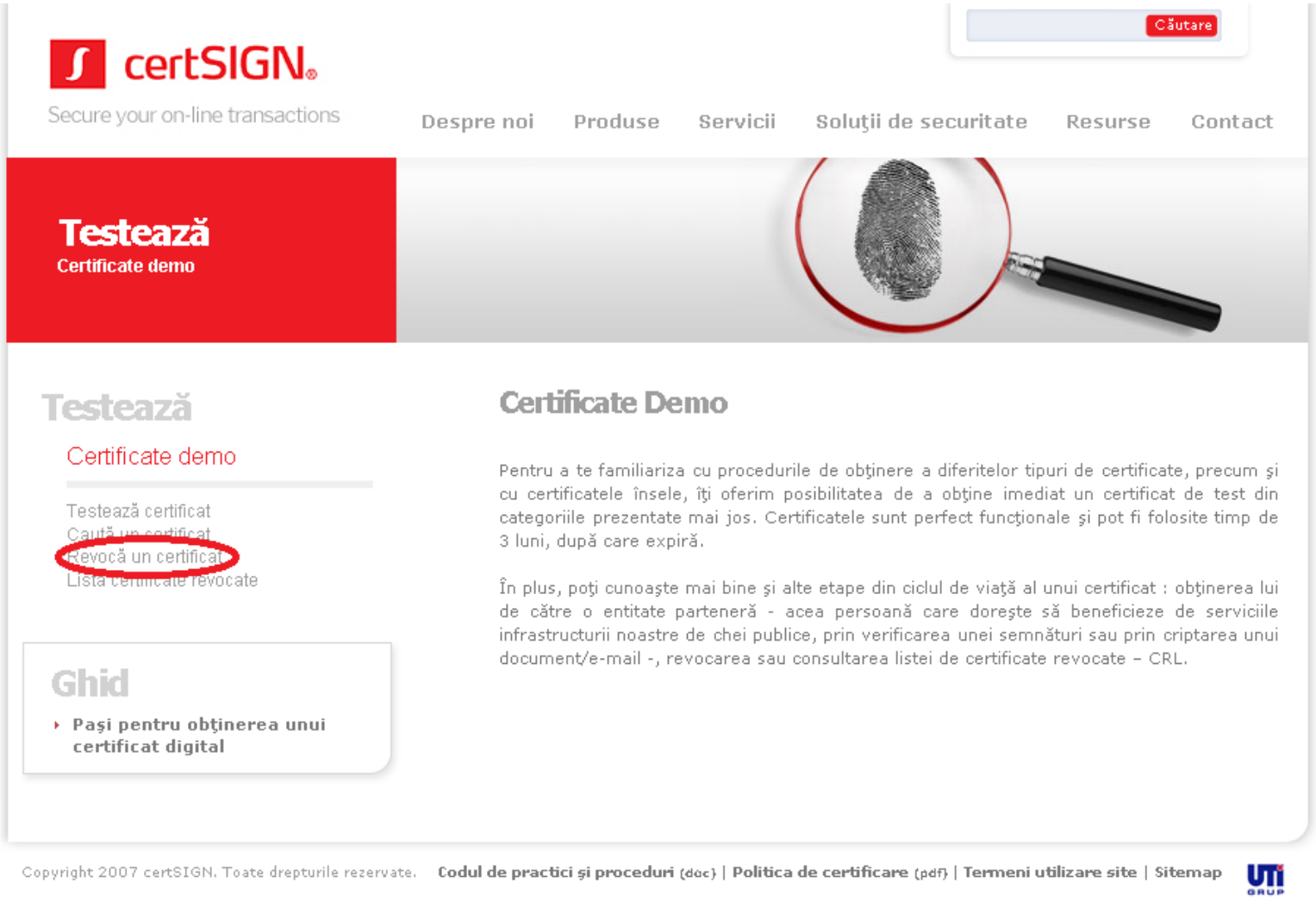
Certificate revocation

- A certificate must be revoked when
  - the private key pair is compromised
  - the private key pair is lost
  - the person leaves the company

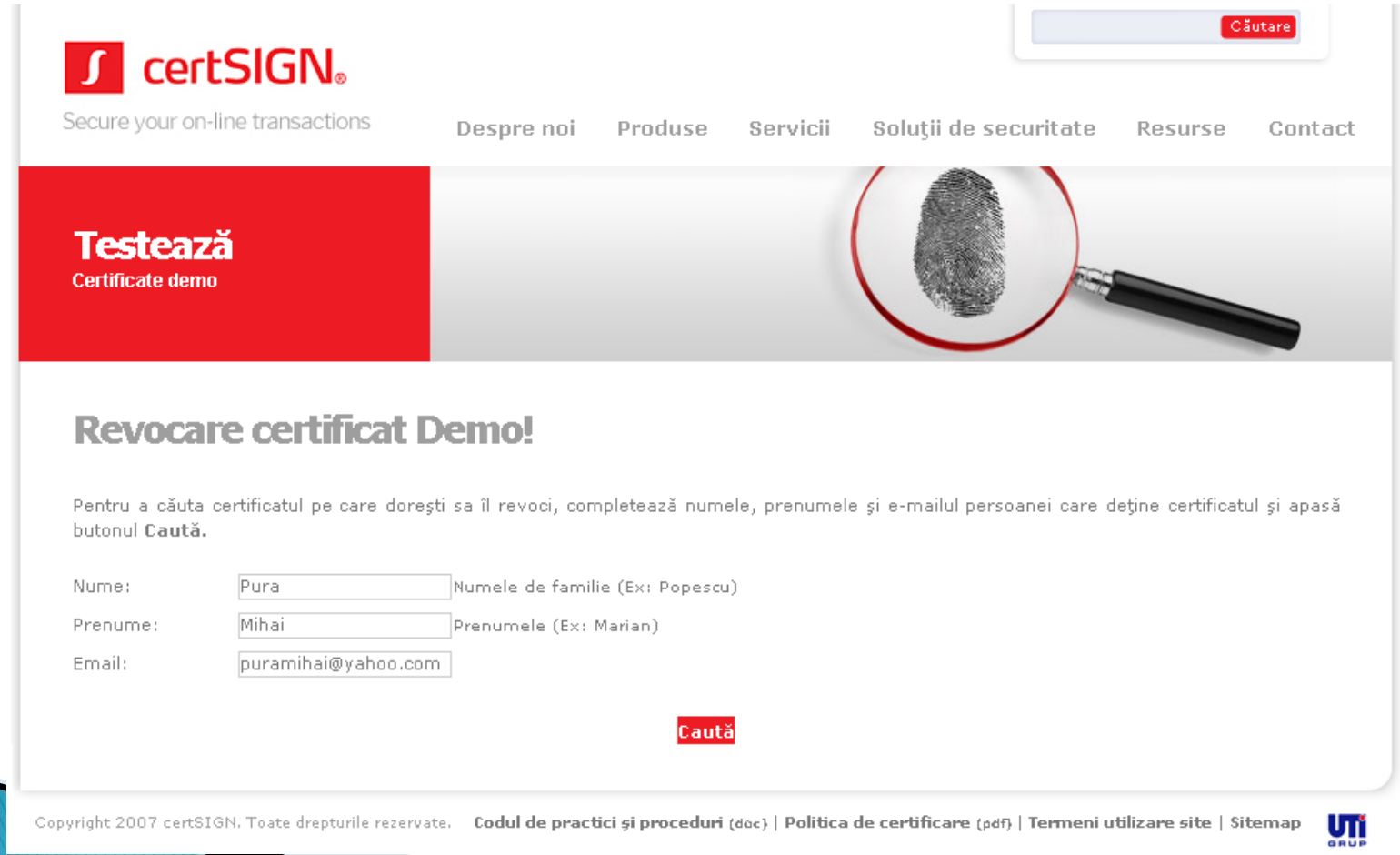
- All users know to **no longer trust** in the revoked certificates

- The CA makes available the revocation information for the certificates it has issued
  - CRL – Certificate Revocation List
  - OCSP – Online Certificate Status Protocol

Certificate revocation



Certificate revocation



Certificate revocation

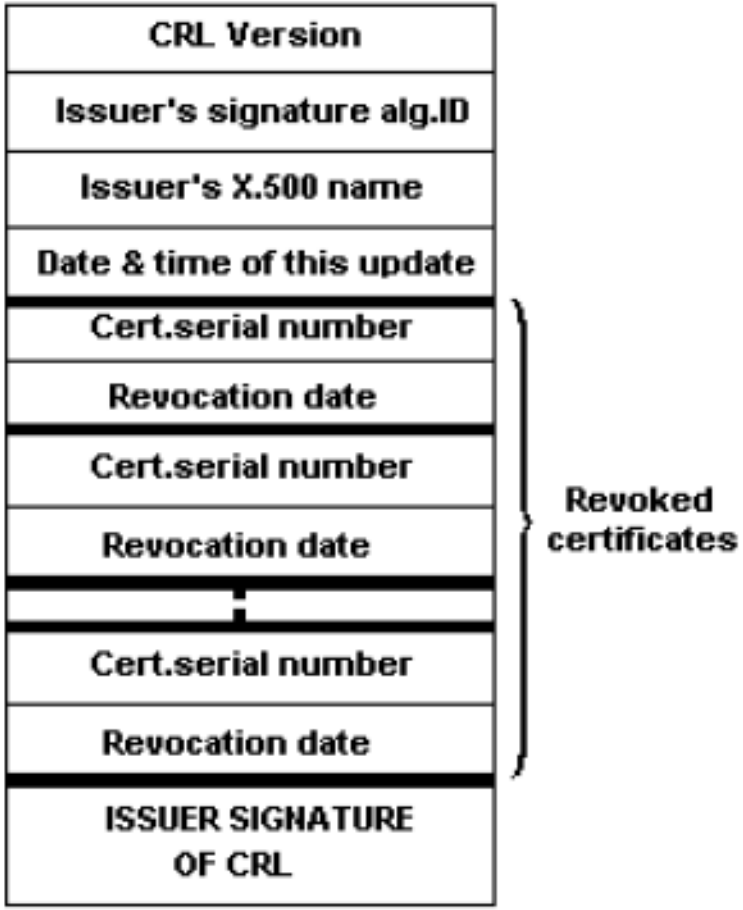
Numărul serial al certificatului:

- **trebuie să fie exact în forma în care a fost scris în fereastra de la emiterea certificatului din care am descărcat fişierul P12**
- **trebuie scris cu majuscule** (pentru cifrele hexazecimale A, B, C, D, E, F) şi **fără spaţii**

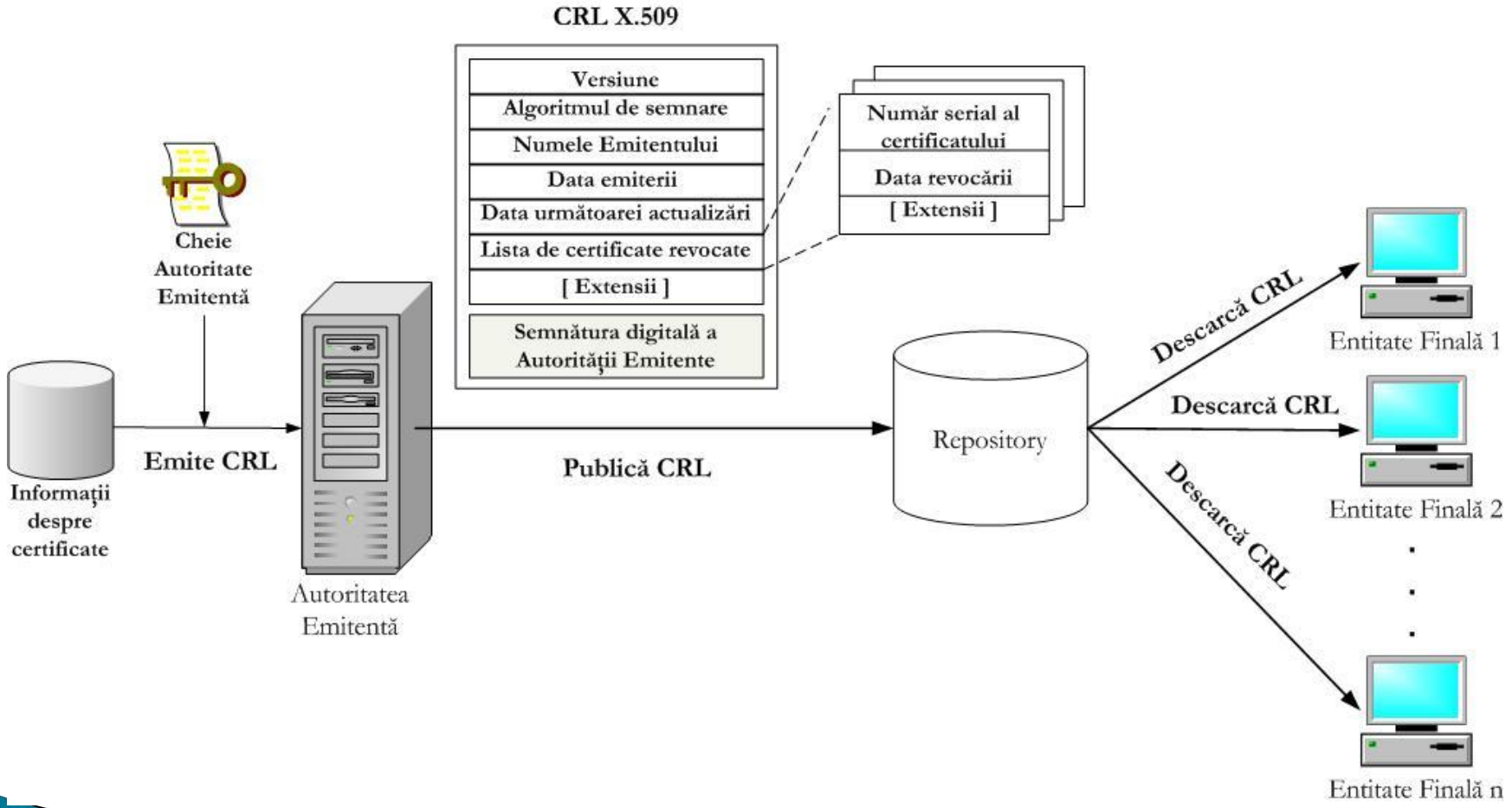
Certificate Revocation List

- contains a list of serial numbers for the certificates that were revoked by the CA
- the information contained in the CRL is digitally signed by the issuing CA
- the CRLs are made public by the CA and can be downloaded usually from a web site

Certificate Revocation List



Certificate Revocation List



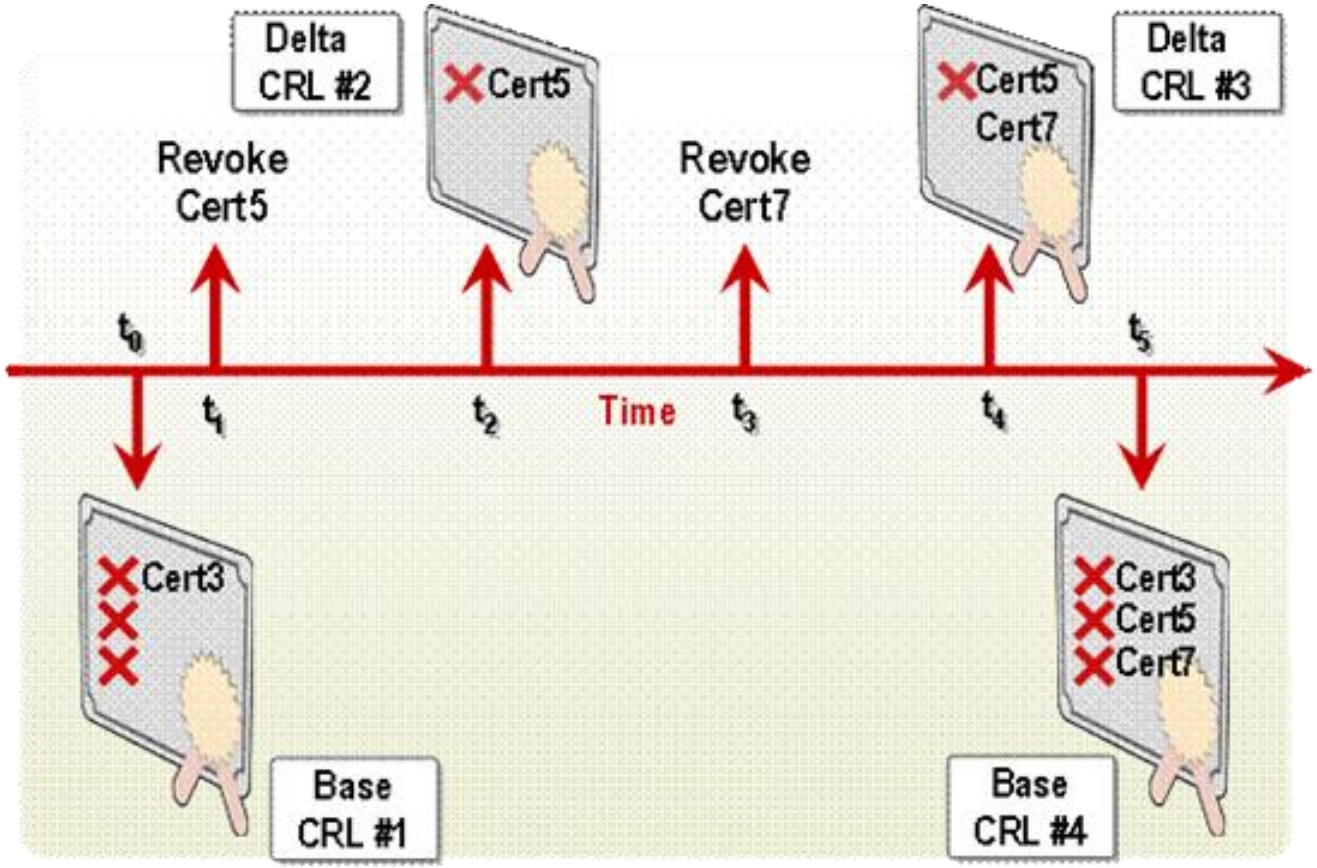
Certificate Revocation List

- To check if a certificate has been revoked
  - look for its serial number in the list from the CRL
  - if the serial number is found in the CRL, than the certificate has been revoked
  - otherwise, the certificate has not been revoked

Certificate Revocation List

- Relaying parties check CRL before using a certificate
- Caching a CRL in a local cache
- Rather than one long CRL, keep multiple shorter CRLs

Certificate Revocation List

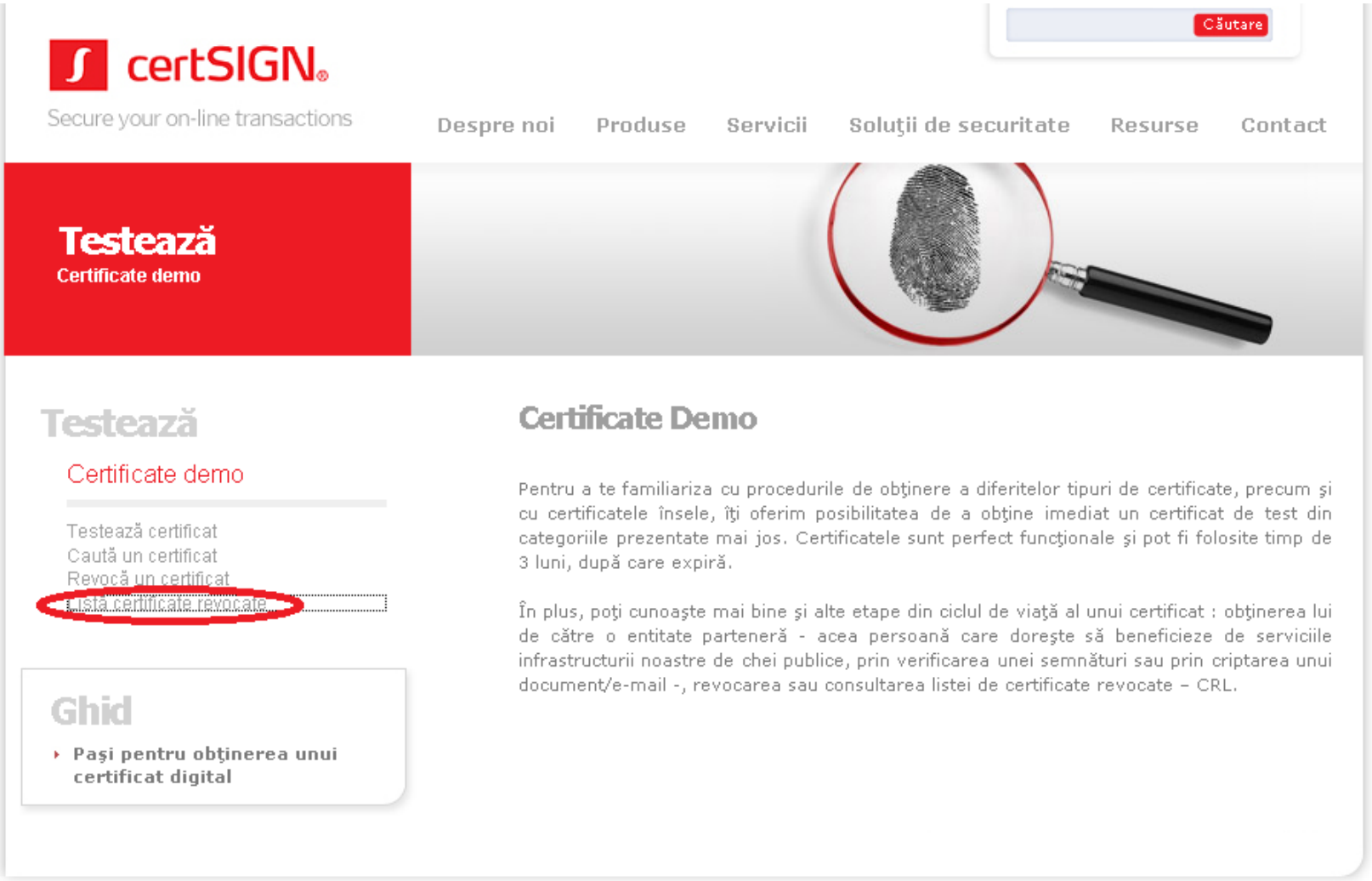


Certificate Revocation List

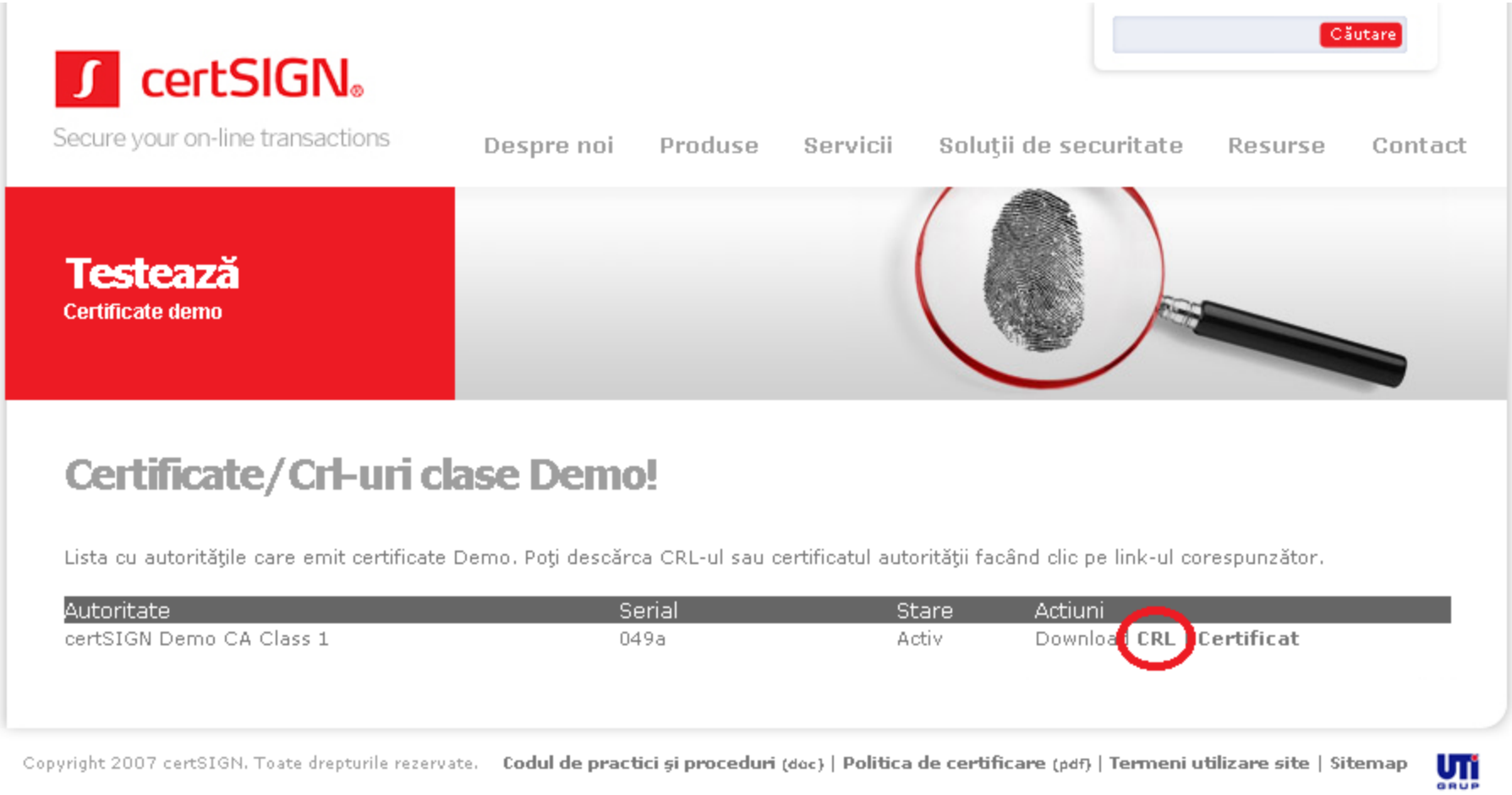
- Distribute the CRL to multiple places and spread the load using the certificate extension field cRLDistributionPoints

- Use a sufficiently scalable and powerful CR server

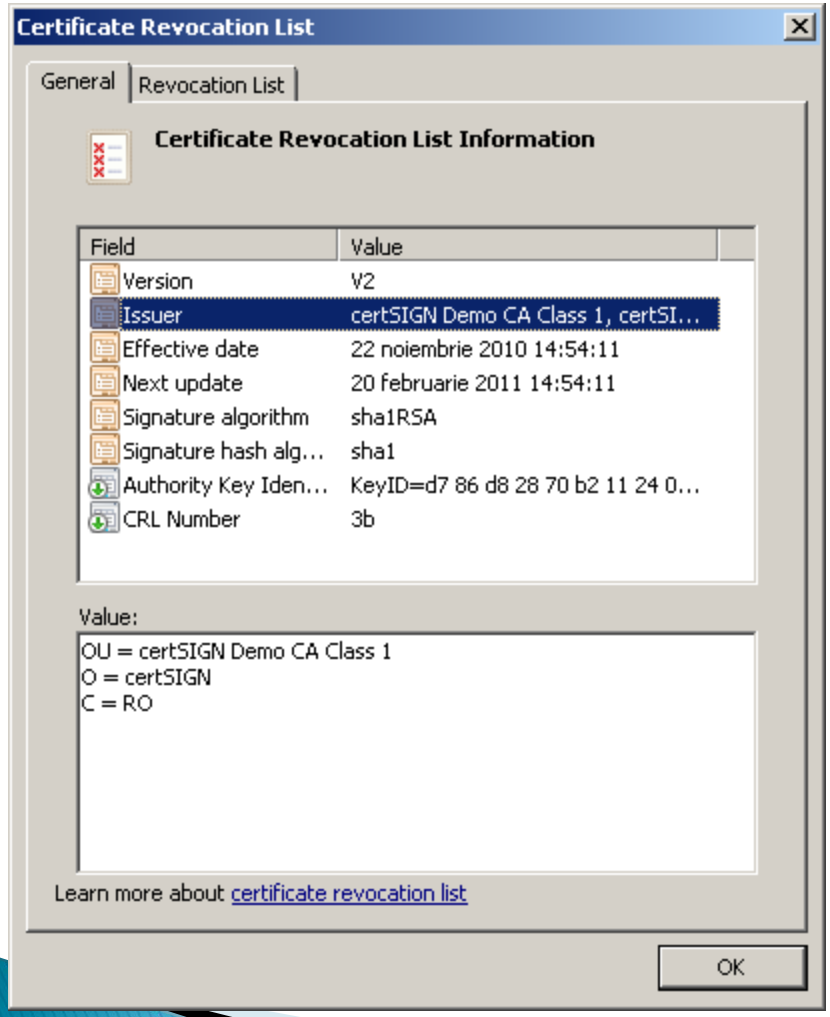
Certificate Revocation List

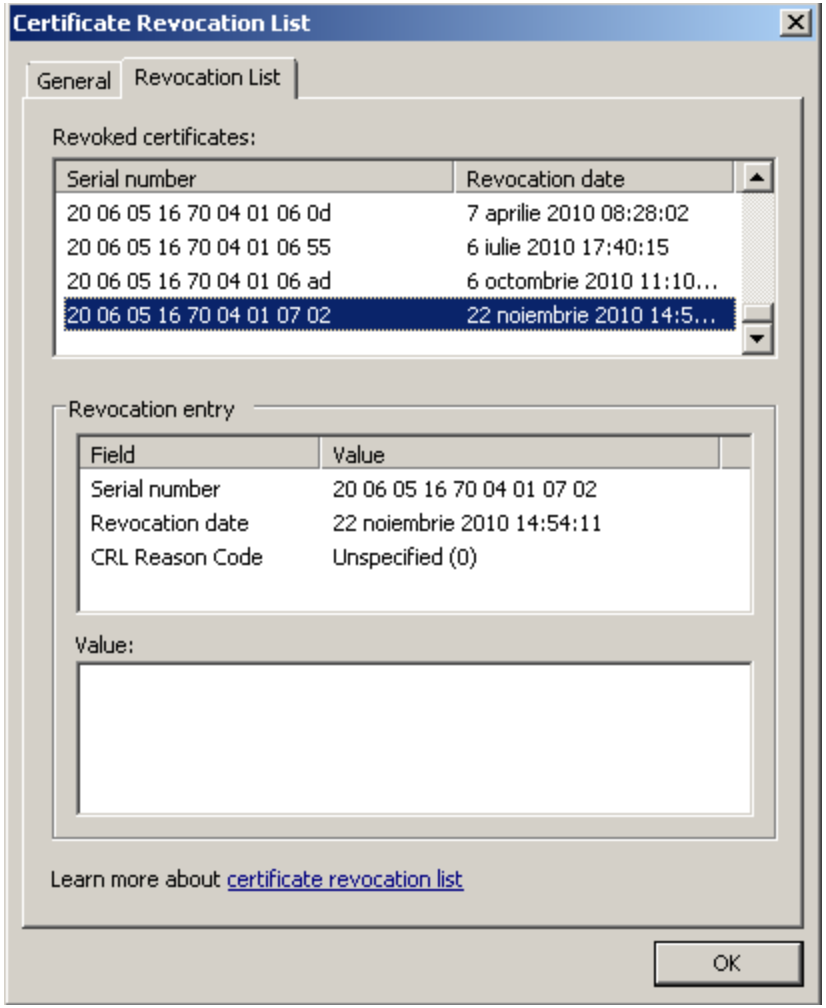


Certificate Revocation List



Certificate Revocation List





Import CRL into certificate store

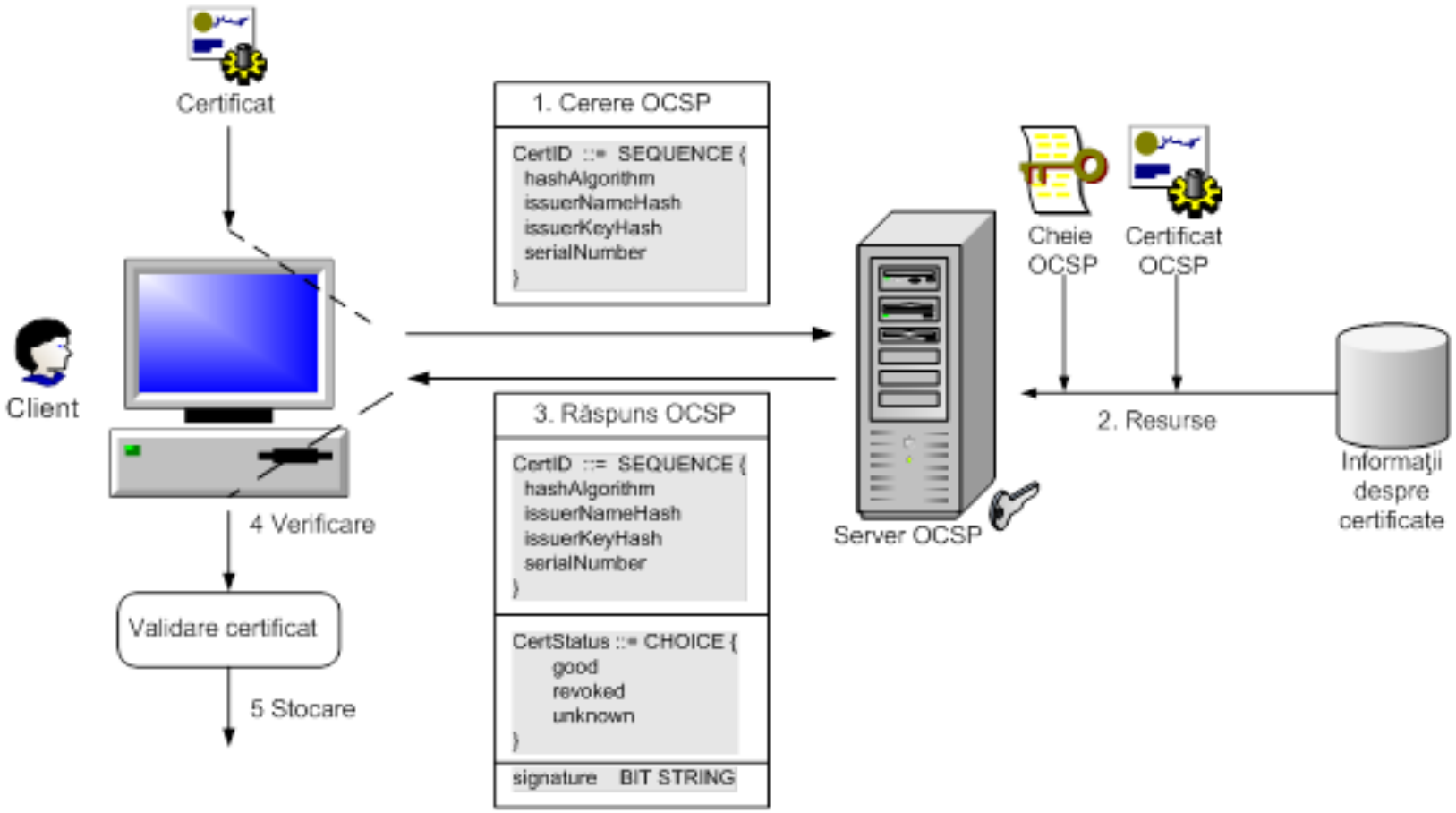
- The same way a digital certificate is imported

OCSP

- RFC 6960 – **X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP**
  - https://tools.ietf.org/html/rfc6960
  - Inquires issuing CA whether a certificate is still valid
  - Response: YES/NO

- http://www.openssl.org/docs/apps/ocsp.html

OCSP


Certificat
1. Cerere OCSP
CertID ::= SEQUENCE {
hashAlgorithm
issuerNameHash
issuerKeyHash
serialNumber
}
Cheie OCSP
Certificat OCSP
Client
3. Răspuns OCSP
CertID ::= SEQUENCE {
hashAlgorithm
issuerNameHash
issuerKeyHash
serialNumber
}
CertStatus ::= CHOICE {
good
revoked
unknown
}
signature BIT STRING
2. Resurse
Informaţii despre certificate
Server OCSP
4 Verificare
Validare certificat
5 Stocare

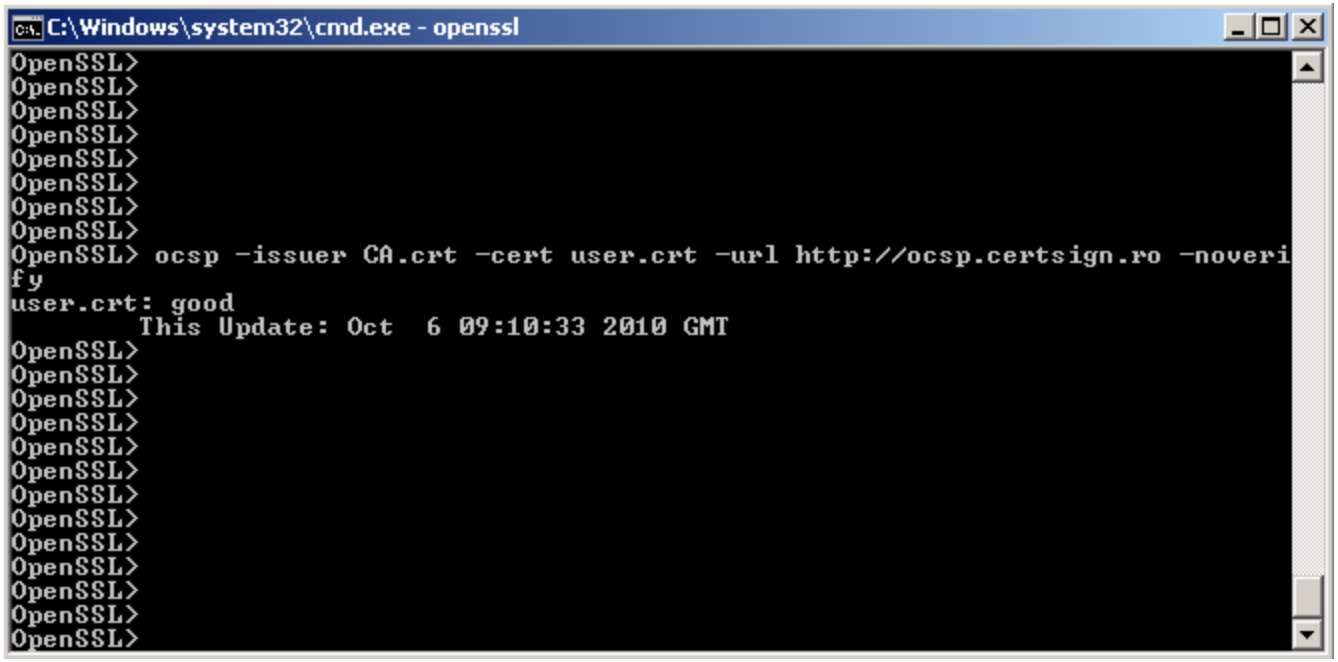
Verify revocation status with openssl

- openssl can be downloaded from
- http://gnuwin32.sourceforge.net/packages/openssl.htm
- The command for OCSP validation of a certificate stored in user.crt is:

- **openssl ocsp -issuer CA.crt -cert user.crt -url http://ocsp.certsign.ro -noverify -text**

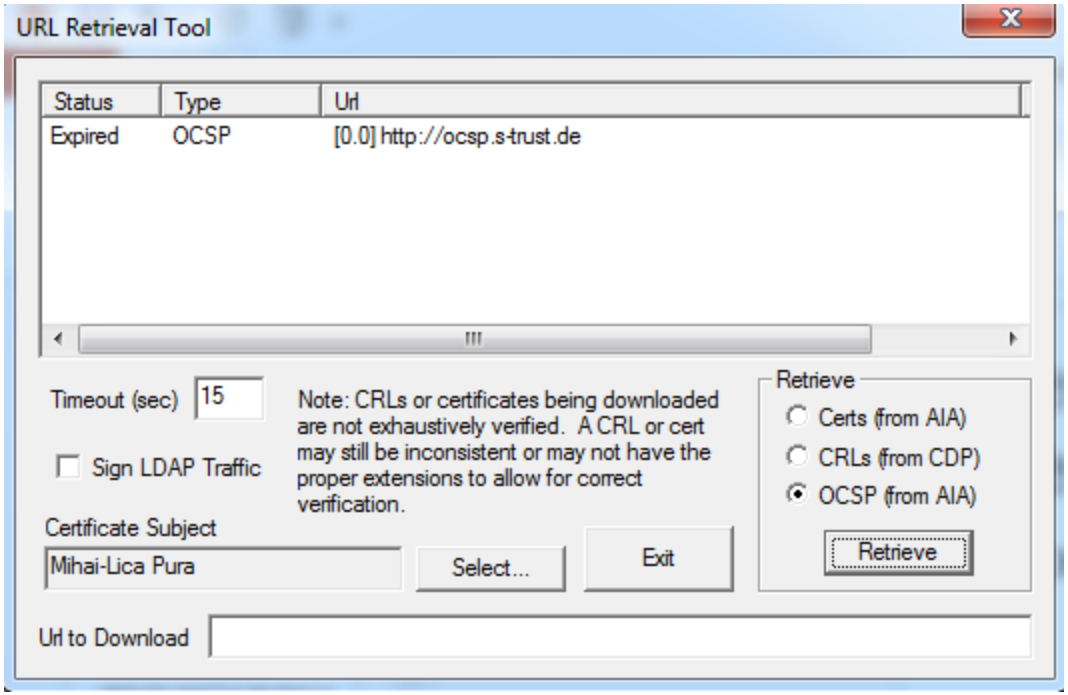
- CA certificate and the certificate that needs to be validated must be in **PEM (BASE64)** format.

Verify revocation status with openssl



Verify revocation status with CertUtil

- **certutil -url path\user.cer**

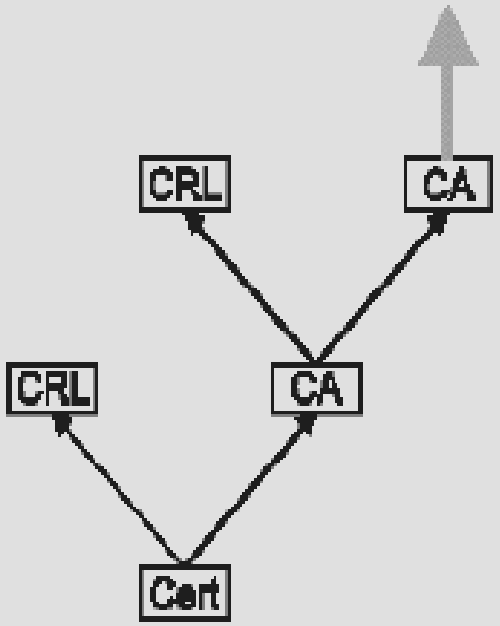


Certificate validation

- Certificate validation
  - verify certificate information (**expiration**, scope, policies, etc.)
- Path validation
  - verify certificate's signature using issuer CA public-key (from CA's certificate)
- Status validation
  - verify **revocation** status using issuer CA's CRL/OCSP

Certificate validation

Checking works in reverse order to normal lookup



Check certificate

Check certificates CRL

repeat

Check CA's certificate

Check CA's CRL

until root reached

Free/Open source CA

- https://www.ejbca.org/
- https://www.dogtagpki.org/
- http://simpleauthority.com/
- http://www.cacert.org/
  - the URL for source code download can be found at About CACert.org
- https://letsencrypt.org/

PKI in the real world

- Document signing
  - Signatures with legal value
- Email encryption & signing (S/MIME)
- Transport-layer VPNs (using SSL)
- Network-layer VPNs (using IPSec)
- Web servers authentication (HTTPS)

PKI in the real world

- Web servers & Client mutual authentication (HTTPS)
- TCP servers & Clients authentication (SSL)
- User authentication to services (databases, etc.)
- Network/IT Infrastructure services access (smartcard logon)
- …

Electronic signature vs. digital signature

- **Electronic signatures**:
  - legal concept that captures
    - intent
    - and consent
  - can be anything, from a scripted text, click-to-sign, image or voice image that illustrates an action of consent

- **Digital signatures**: It's the mechanism that secures documents with cryptography

https://security.stackexchange.com/questions/17554/what-is-the-difference-between-an-electronic-signature-and-a-digital-signature

Electronic signature vs. digital signature

- Are electronic signatures (according to the above definition) **legally binding**?
- There is no absolute answer, because it depends of the national regulations
- e.g. U.S.A.
  https://www.globalsign.com/en/blog/electronic-signatures-vs-digital-signatures/

E-signature vs. e-seal

- According to eIDAS (EU regulation 2014/910)

https://ec.europa.eu/digital-single-market/en/trust-services-and-eid

  - **e-seal**
    - Created by a legal person (e.g. an organization)
    - Without any human interaction (automatically)
  - **e-signature**
    - Created by a natural person (e.g. a human person)
    - The person adding the signature performs some action for this operation
    - The person is authenticated

The use of digital signatures

- **1. Signing digital documents**
  - PDF
  - XML
  - Any other kind of files (PKCS#7)
- **2. Certificate login**
- **3. Code signing**
  - Java Web Start applications
  - Microsoft Office macros
  - Microsoft software installation packages

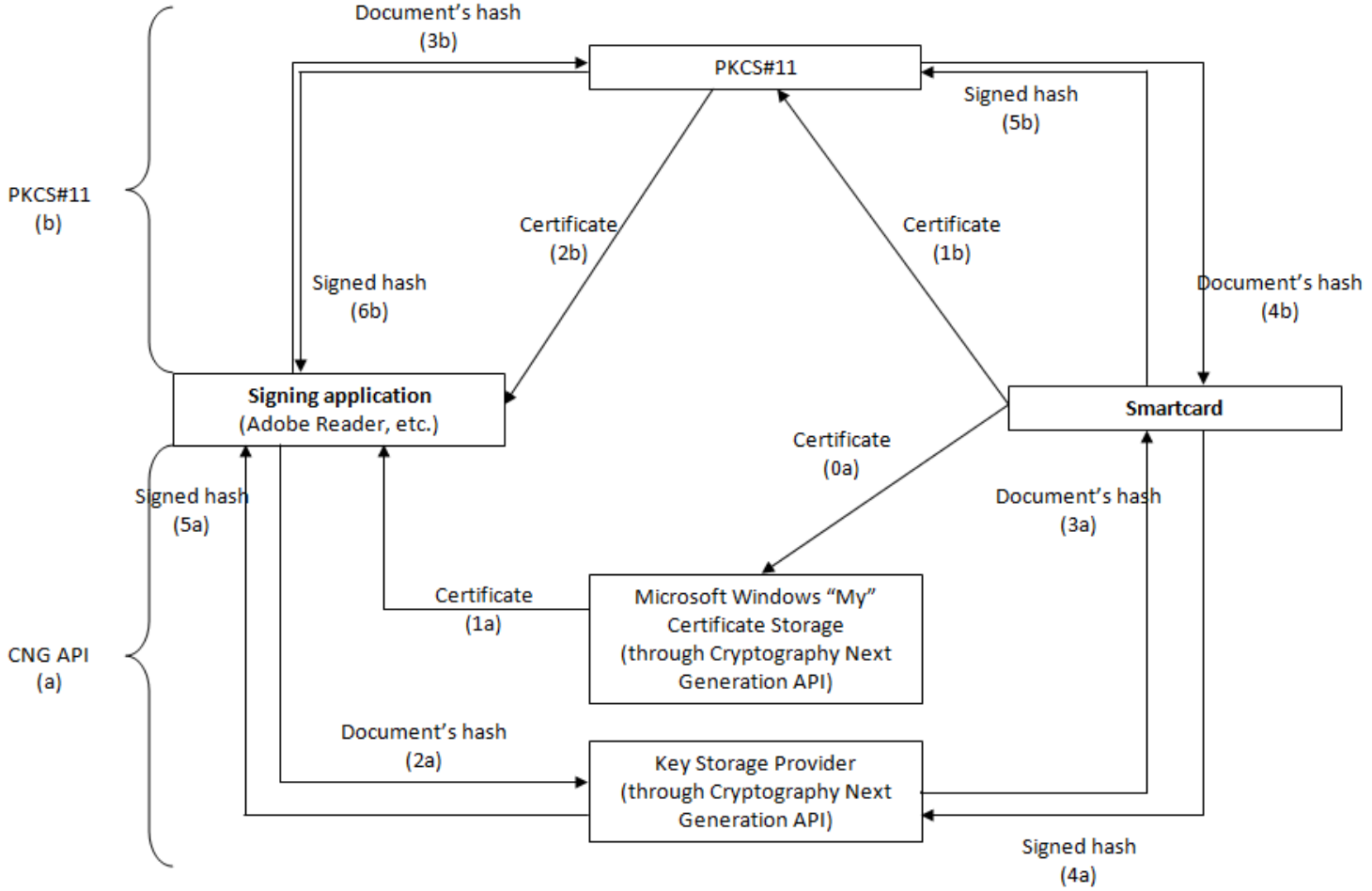
The use of digital signatures

- **ANAF**
  - Annual financial reports
  - https://static.anaf.ro/static/10/Anaf/Declaratii_R/instructiuni/etape_depunere.htm
  - https://static.anaf.ro/static/10/Anaf/Declaratii_R/instructiuni/Instructiuni_D200.htm
- **CNAS**
  - Reports of medics and pharmacists
  - http://www.cnas.ro/casmb/page/procedura-de-inregistrare-a-certificatelor-digitale.html
- **UNNPR**

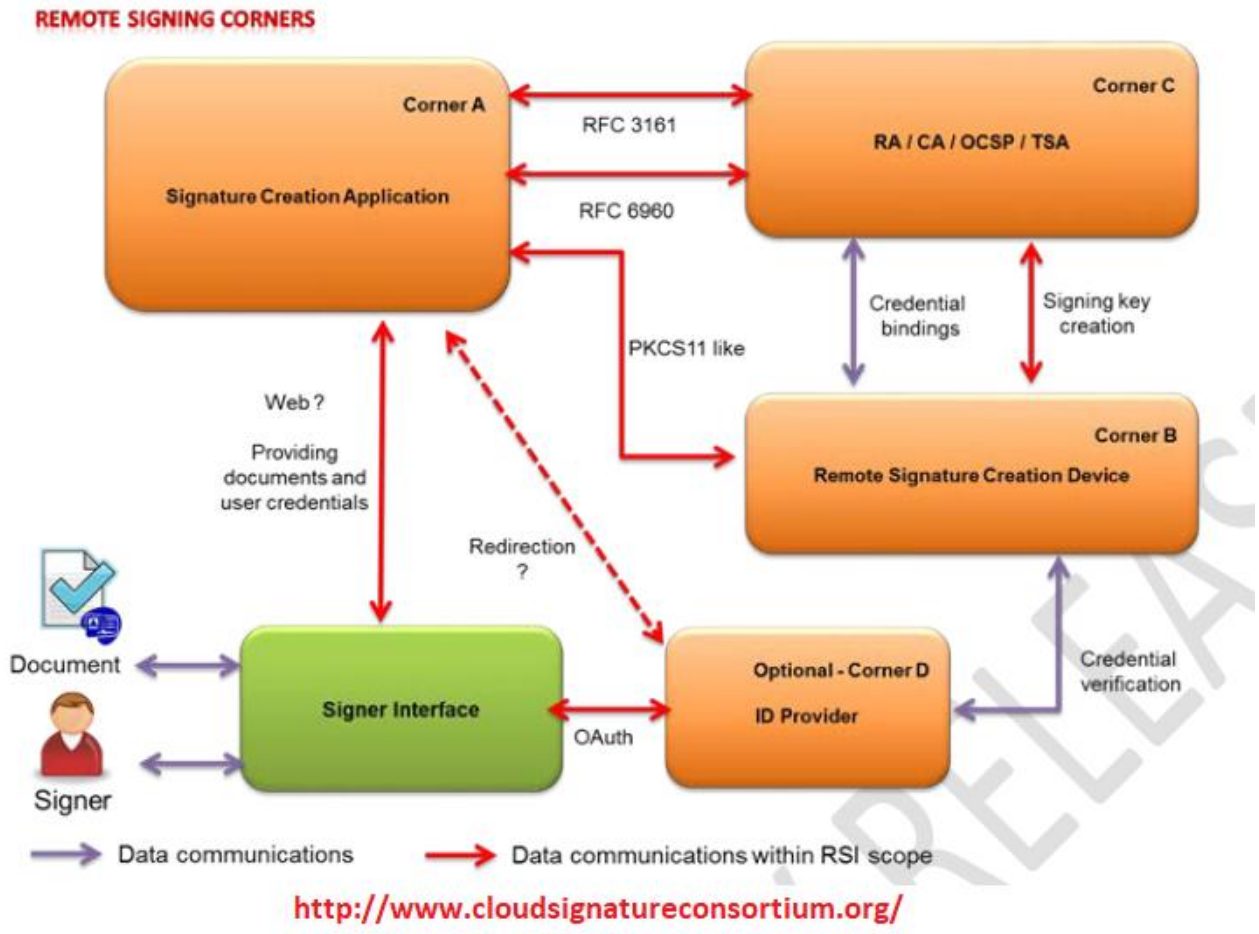
The use of digital signatures

- **MySMIS 2014**
  - Access EU funds – Ministry of European Funds
  - https://2014.mysmis.ro/
  - http://www.fonduri-ue.ro/mysmis
- **SEAP (Sistemul Electronic de Achiziții Publice)**
  - http://licitatiiseap.ro/seap/
  - https://www.e-licitatie.ro/
- **ITM (Inspectoratul Teritorial al Muncii)**
- **Monitorul Oficial**
- **ONRC (Oficiul Național al Registrului Comerțului)**
- **etc.**

Performing a digital signature (local)



Performing a digital signature (cloud/remote/mobile/server)



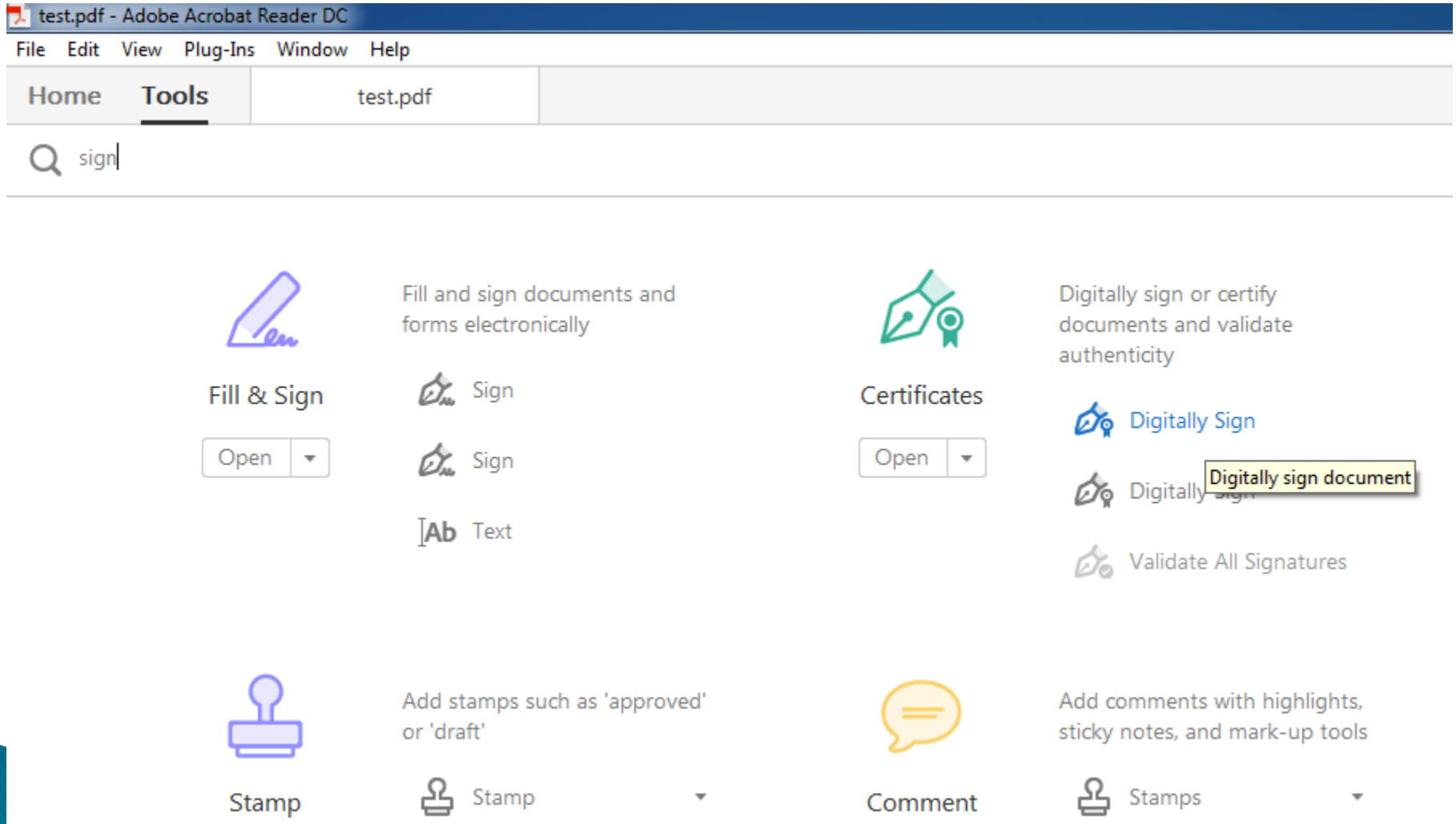
Digital signatures and PDF files

- **Adobe Acrobat Reader DC**
- http://get.adobe.com/reader/?promoid=KLXME

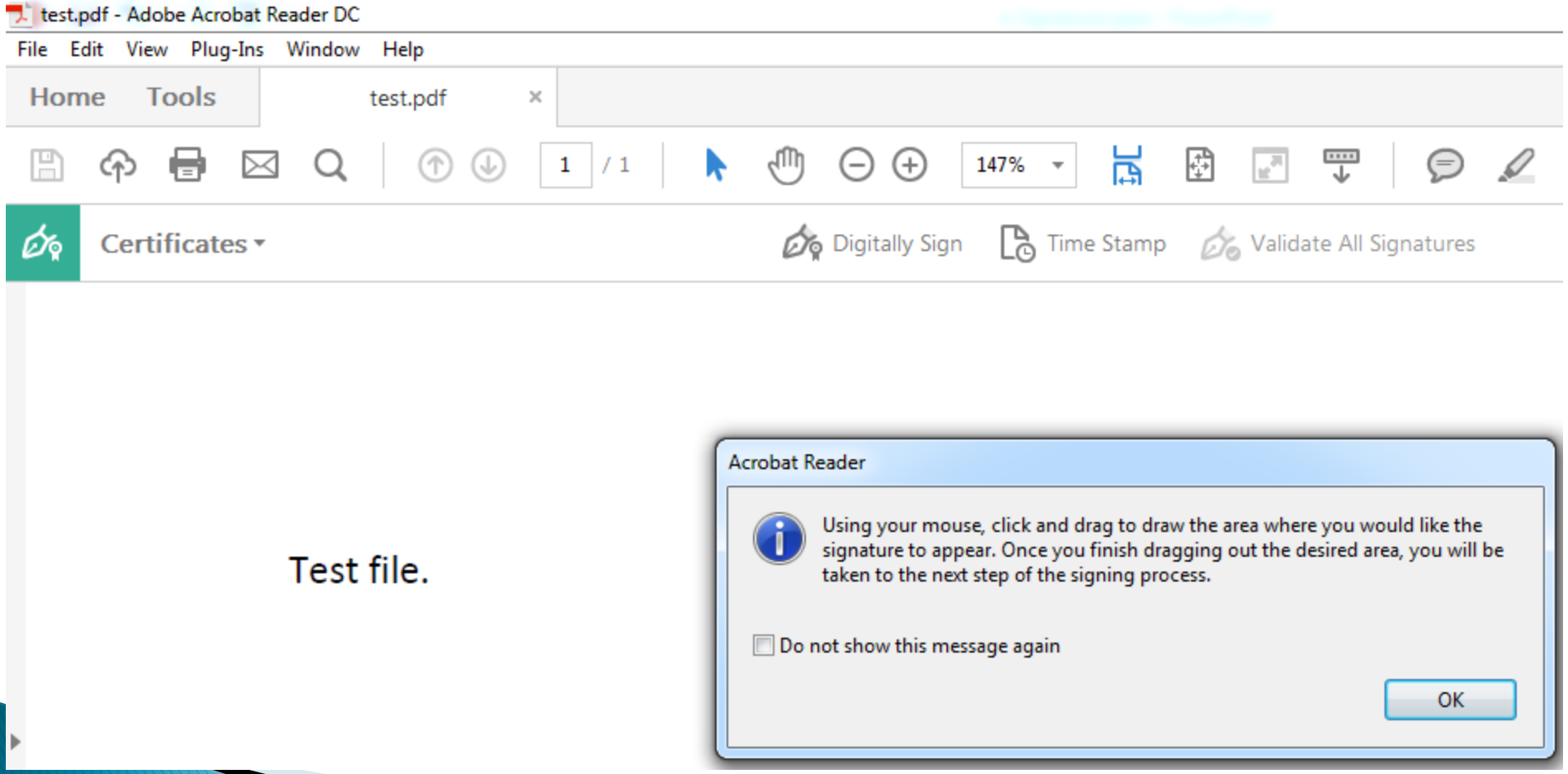
- Previously, a PDF document could be signed only with Adobe Acrobat Professional

- Starting with Adobe Acrobat Reader X, signing can now be done from the reader

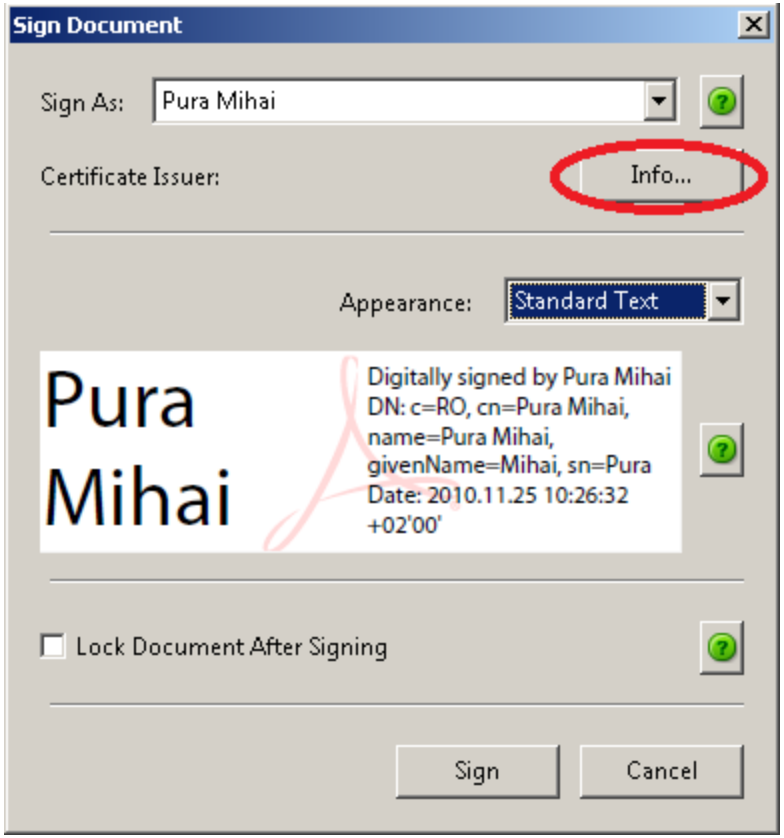
Sign a PDF file



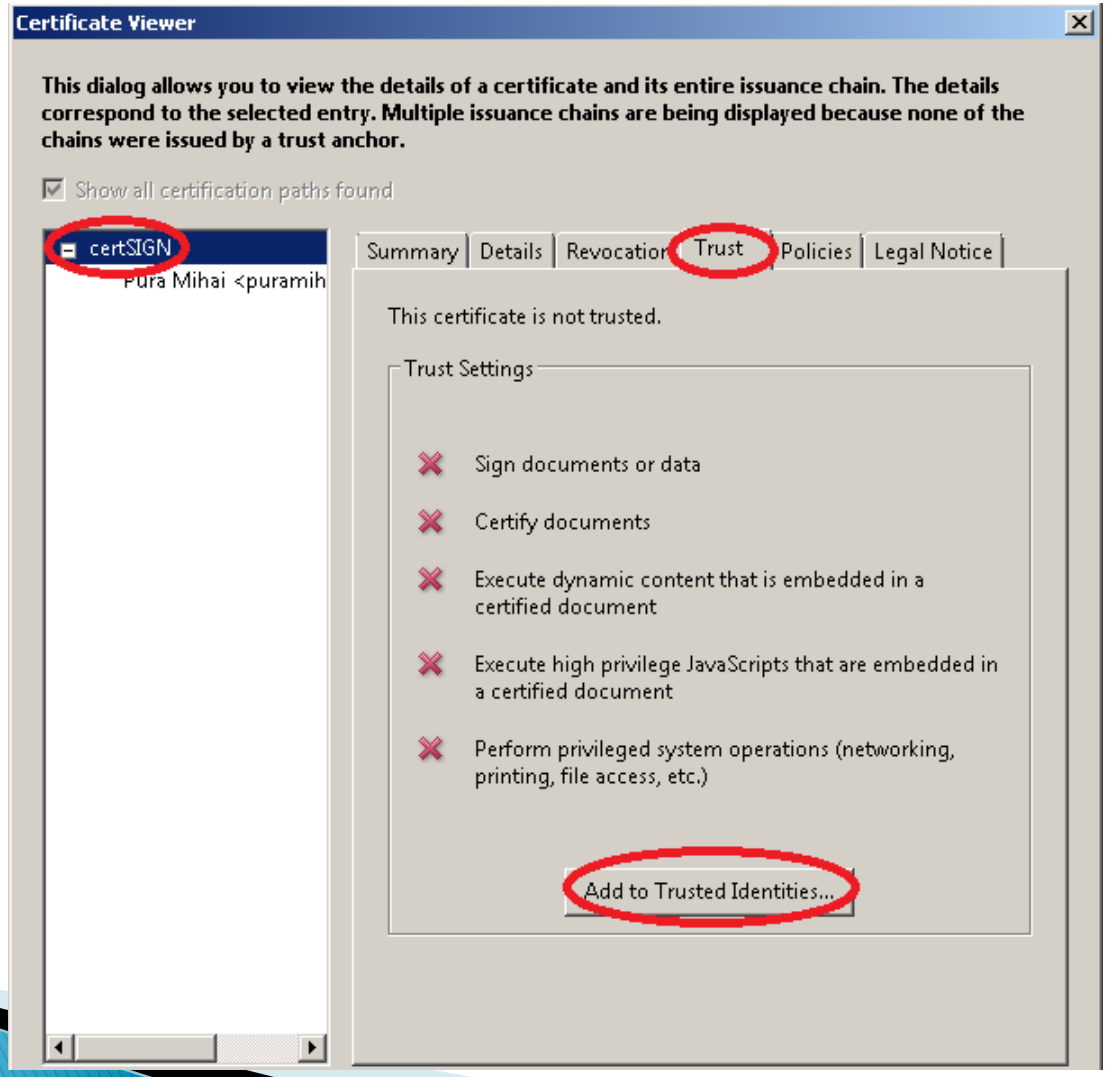
Sign a PDF file



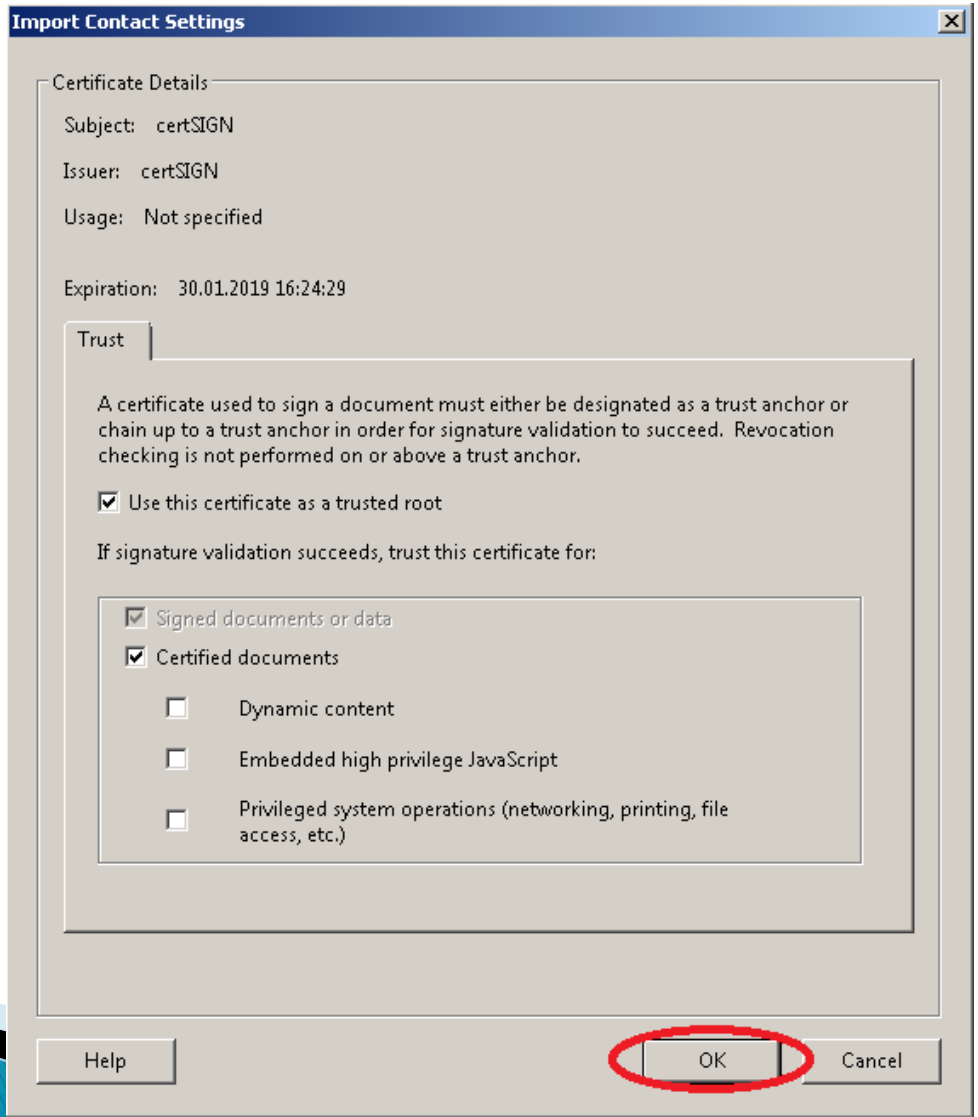
Sign a PDF file



Sign a PDF file

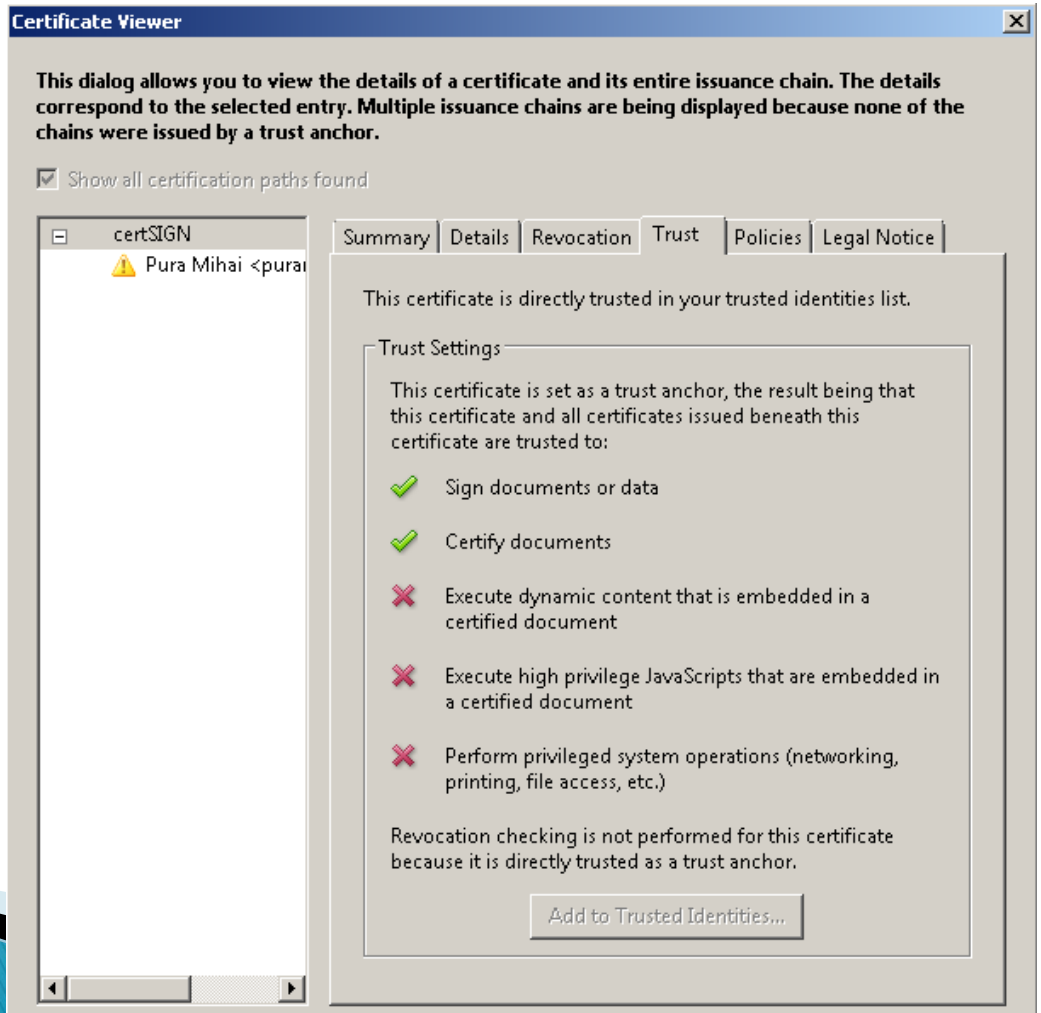


Sign a PDF file

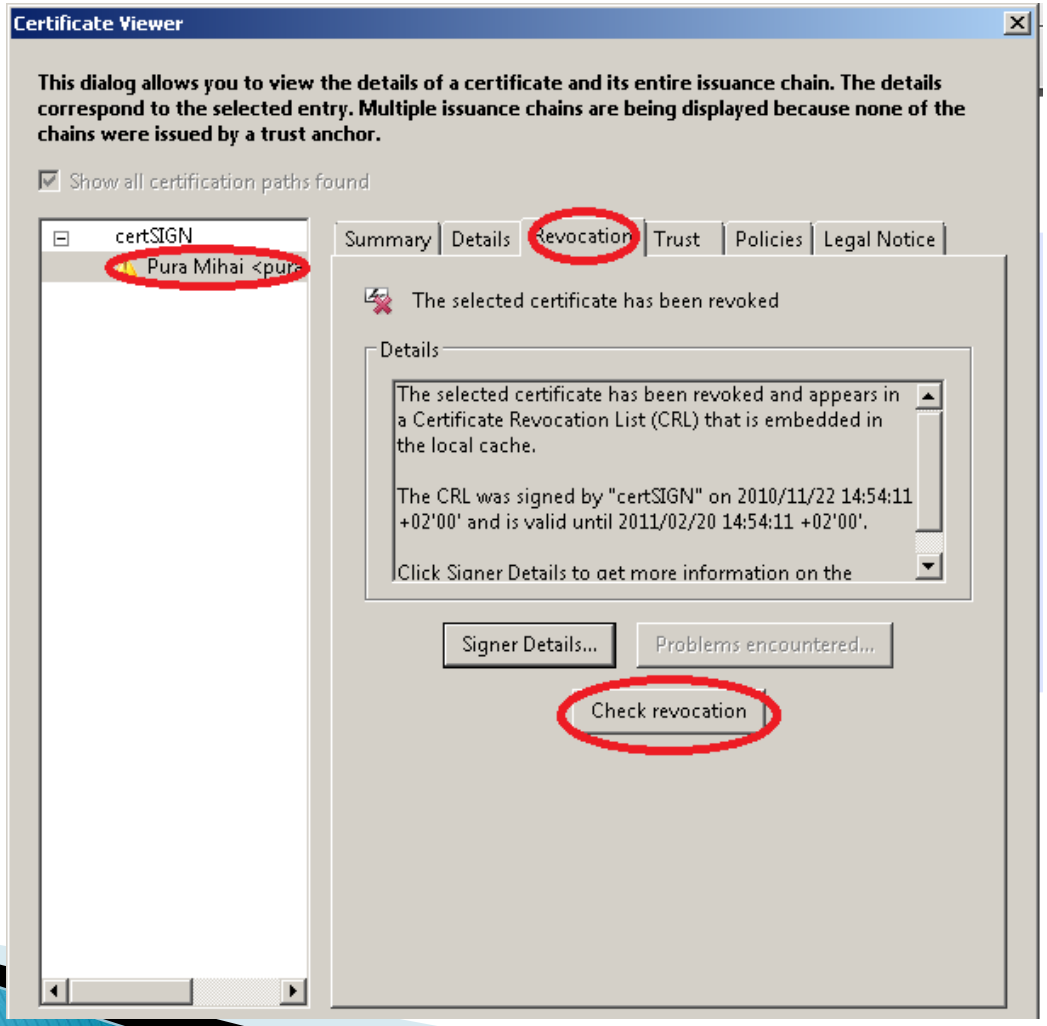


Sign a PDF file

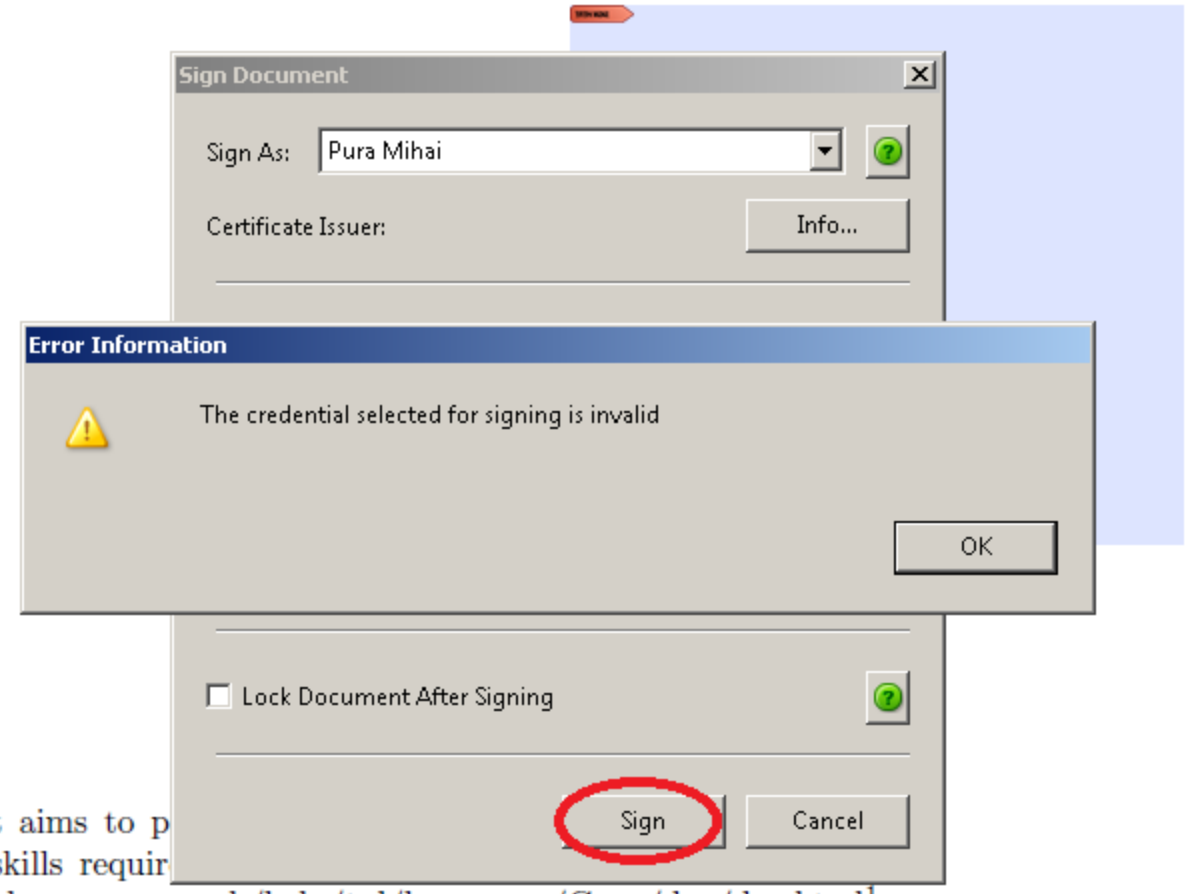
- Close and then reopen Certificate Viewer



Sign a PDF file

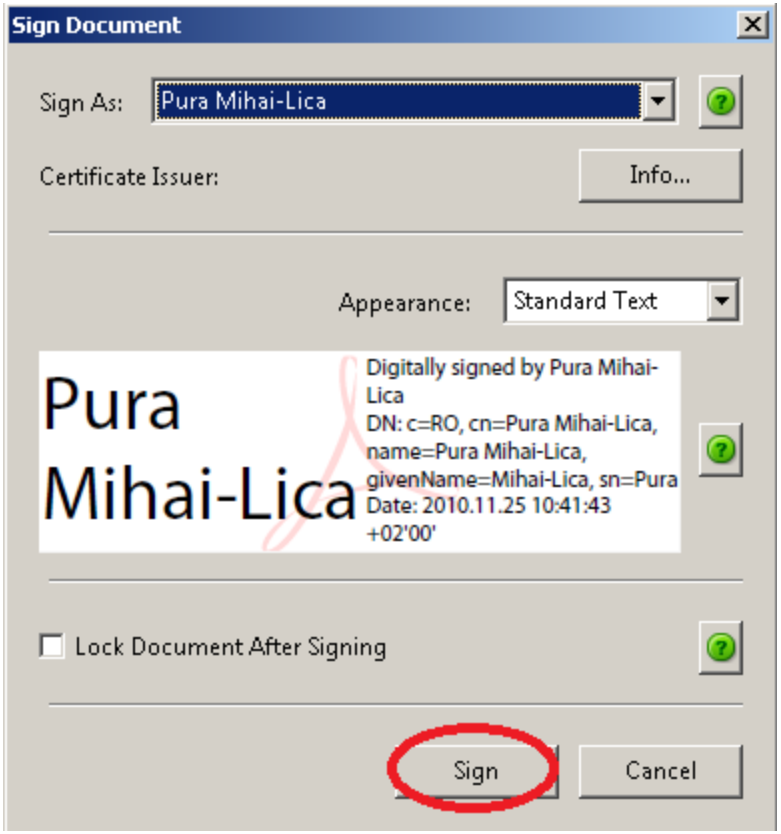


Sign a PDF file

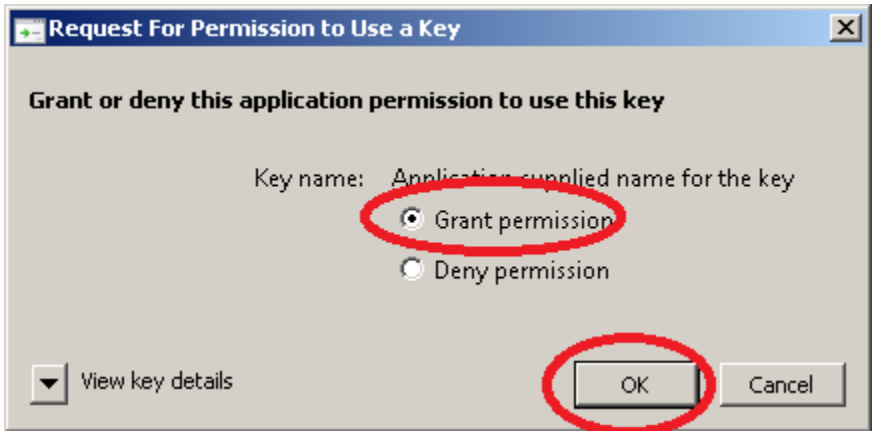


ocument aims to p
se the skills requir
//www-h.eng.cam.ac.uk/help/tpl/languages/C++/doc/doc.html[1]
ttp://www-h.eng.cam.ac.uk/help/tpl/languages/C++/doc/doc.pdf[2]
http://www-h.eng.cam.ac.uk/help/documentation/docsource/index.html[3]

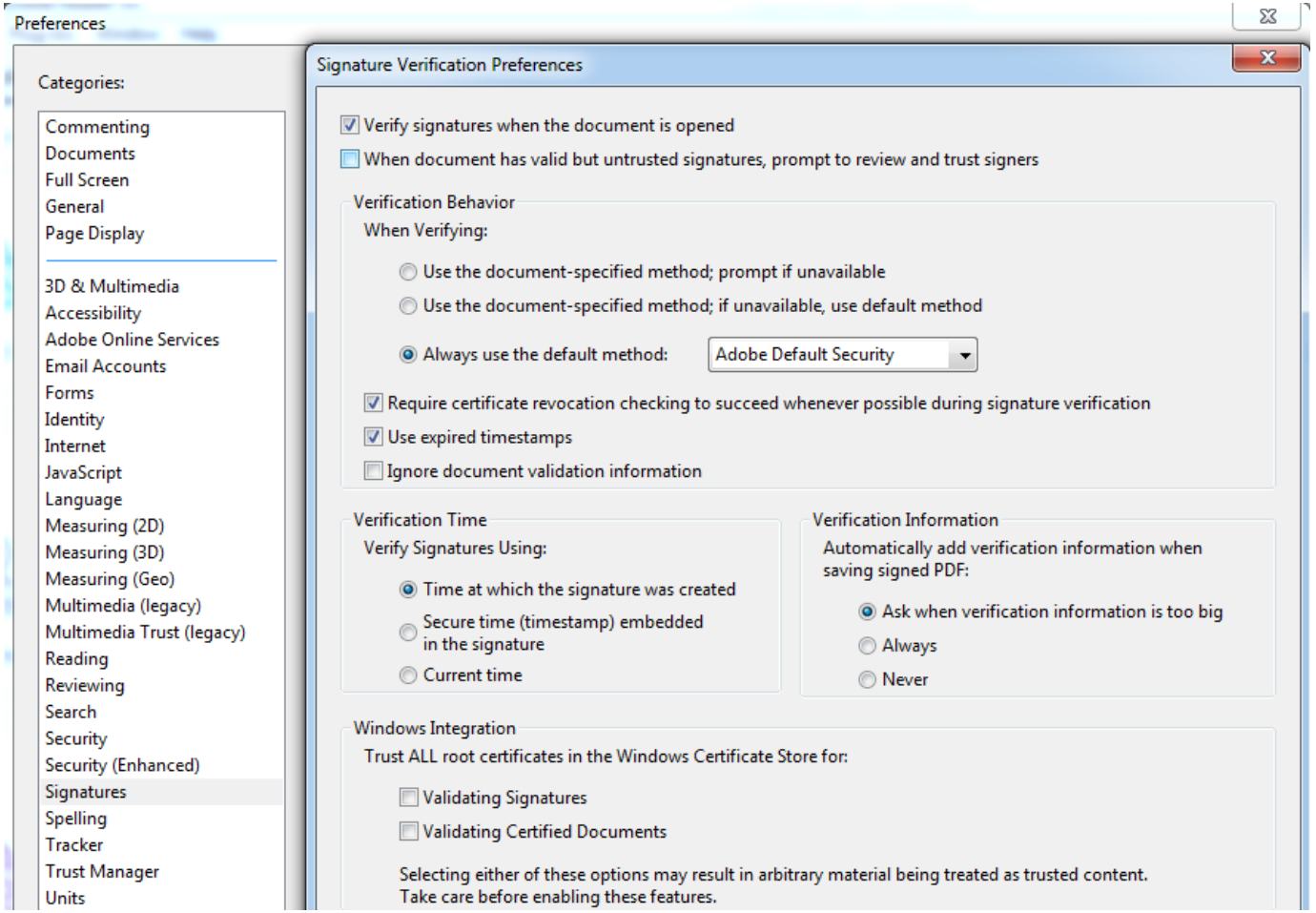
Sign a PDF file



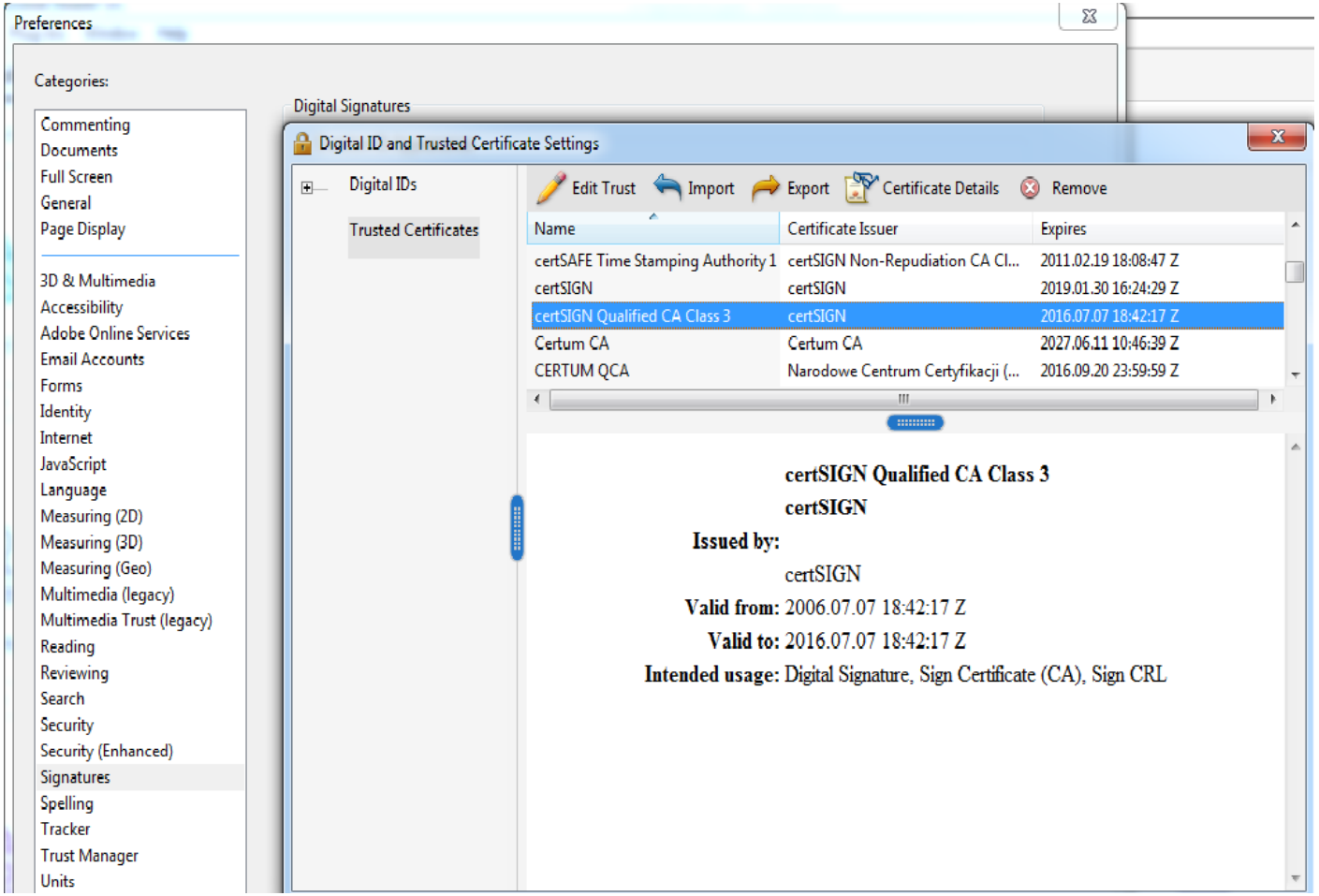
Sign a PDF file



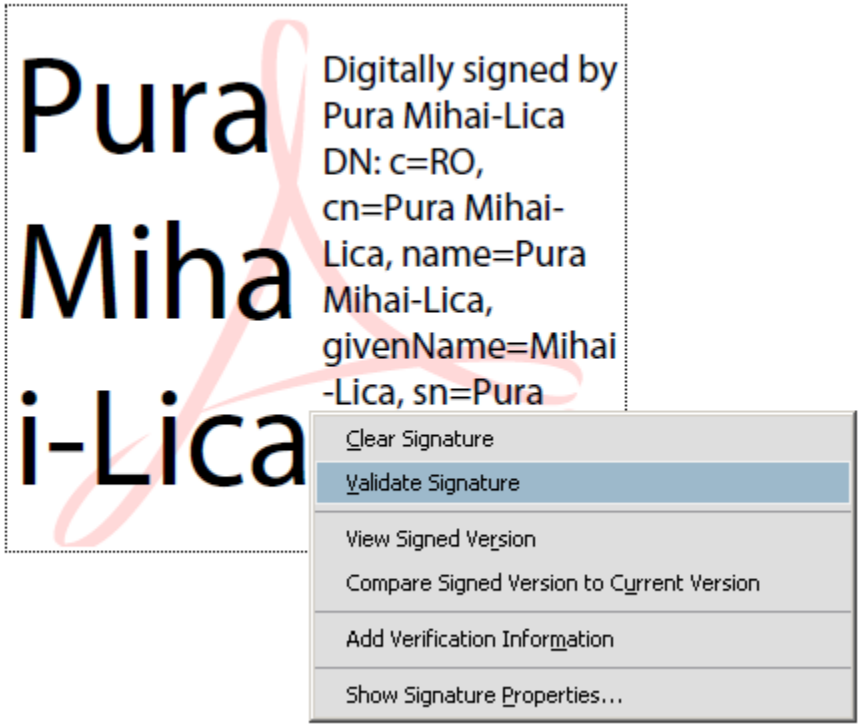
Verify signatures from a PDF file



Verify signatures from a PDF file

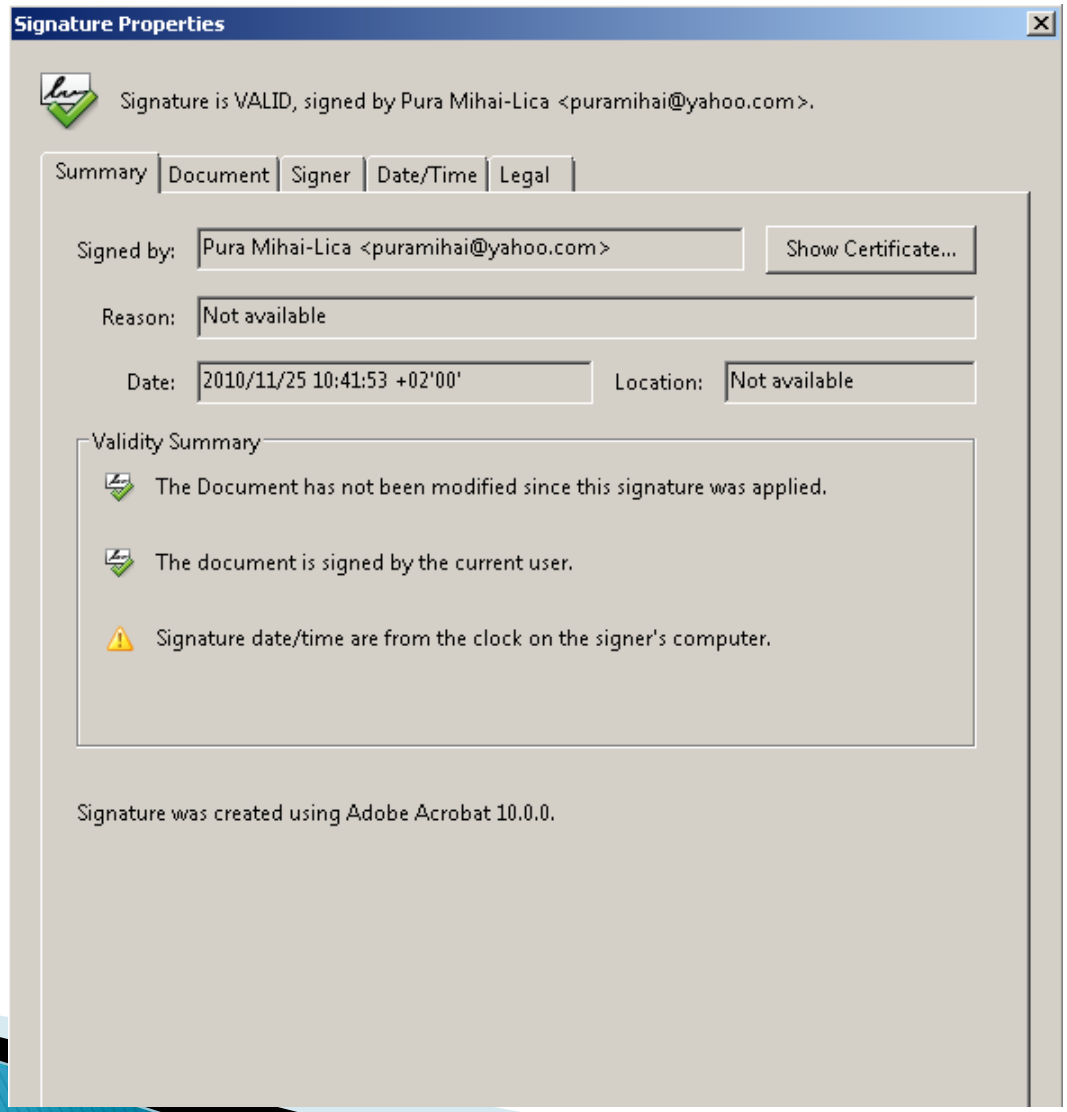


Verify signatures from a PDF file



Verify signatures from a PDF file

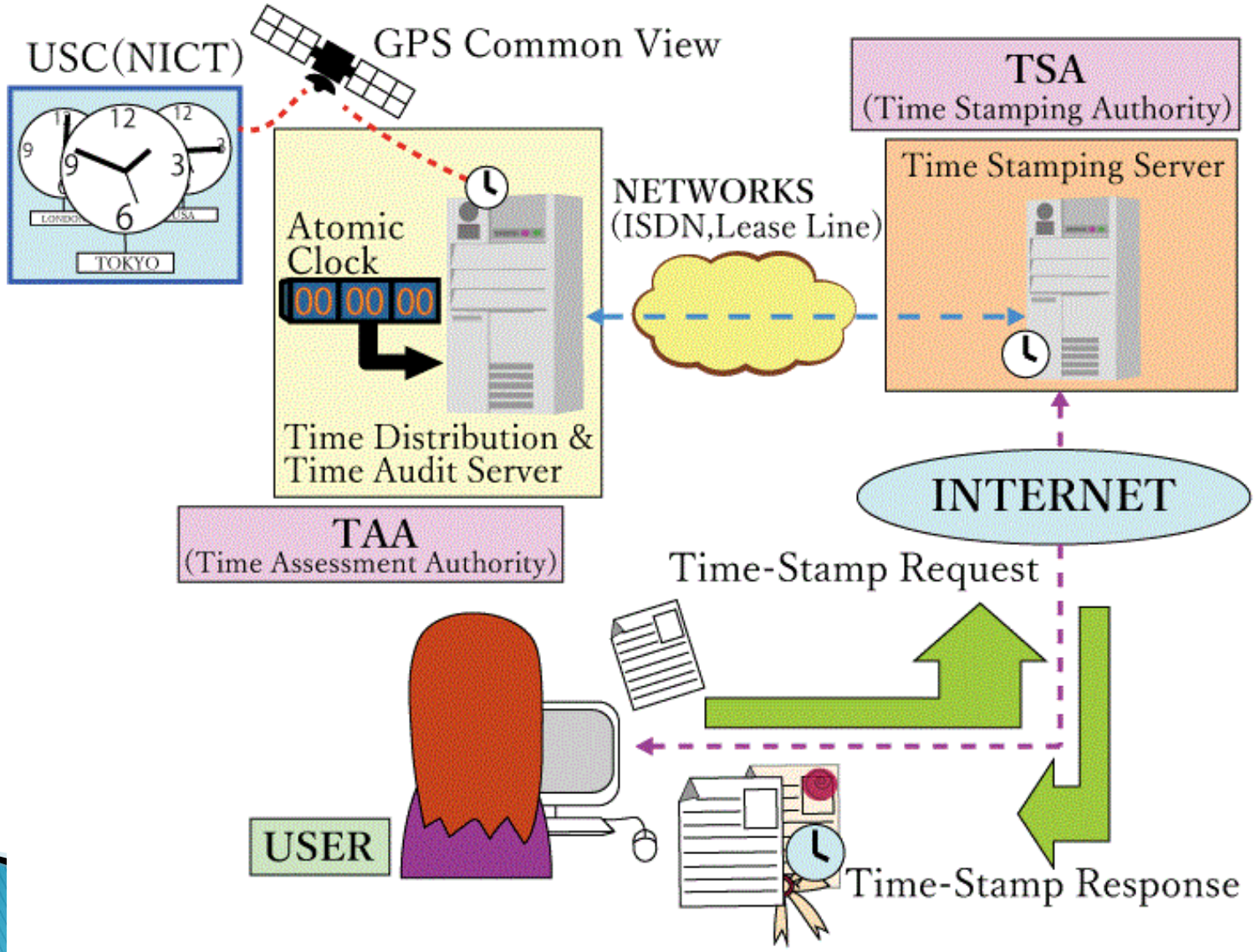
Verify signatures from a PDF file



Timestamps

- TSA – Timestamp Authority (trusted party)
  - applies timestamps to documents
  - Allows a client to prove at a later date that some datum existed before a particular time
    - e.g. an electronic signature on a document existed before a particular time
  - Usually used to add a timestamp-attribute to a digital signature

Timestamps


USC(NICT)
GPS Common View
TOKYO
Atomic Clock
Time Distribution & Time Audit Server
NETWORKS (ISDN,Lease Line)
TSA (Time Stamping Authority)
Time Stamping Server
INTERNET
TAA (Time Assessment Authority)
Time-Stamp Request
USER
Time-Stamp Response

Timestamps

- Free timestamp test servers:

- http://clepsydre.edelweb.fr/dvcs/service-tsp
- http://time.certum.pl/
- http://timestamping.edelweb.fr/service/tsp
- http://www.edelweb.fr/cgi-bin/service-tsp

- http://tsa.safecreative.org/
- https://freetsa.org/tsr

- https://opentimestamps.org/

Timestamps

- Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)
- RFC 3161
- https://www.ietf.org/rfc/rfc3161.txt

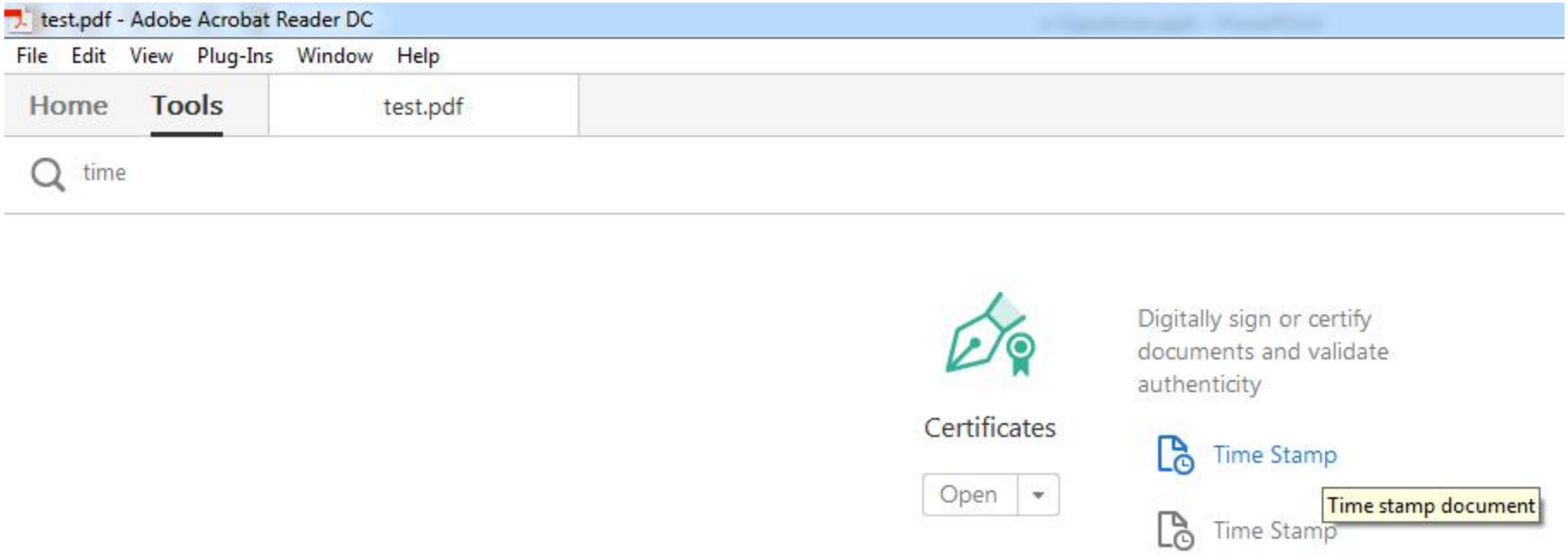
- Policy Requirements for Time-Stamping Authorities (TSAs)
- RFC 3628
- https://tools.ietf.org/html/rfc3628

Timestamps

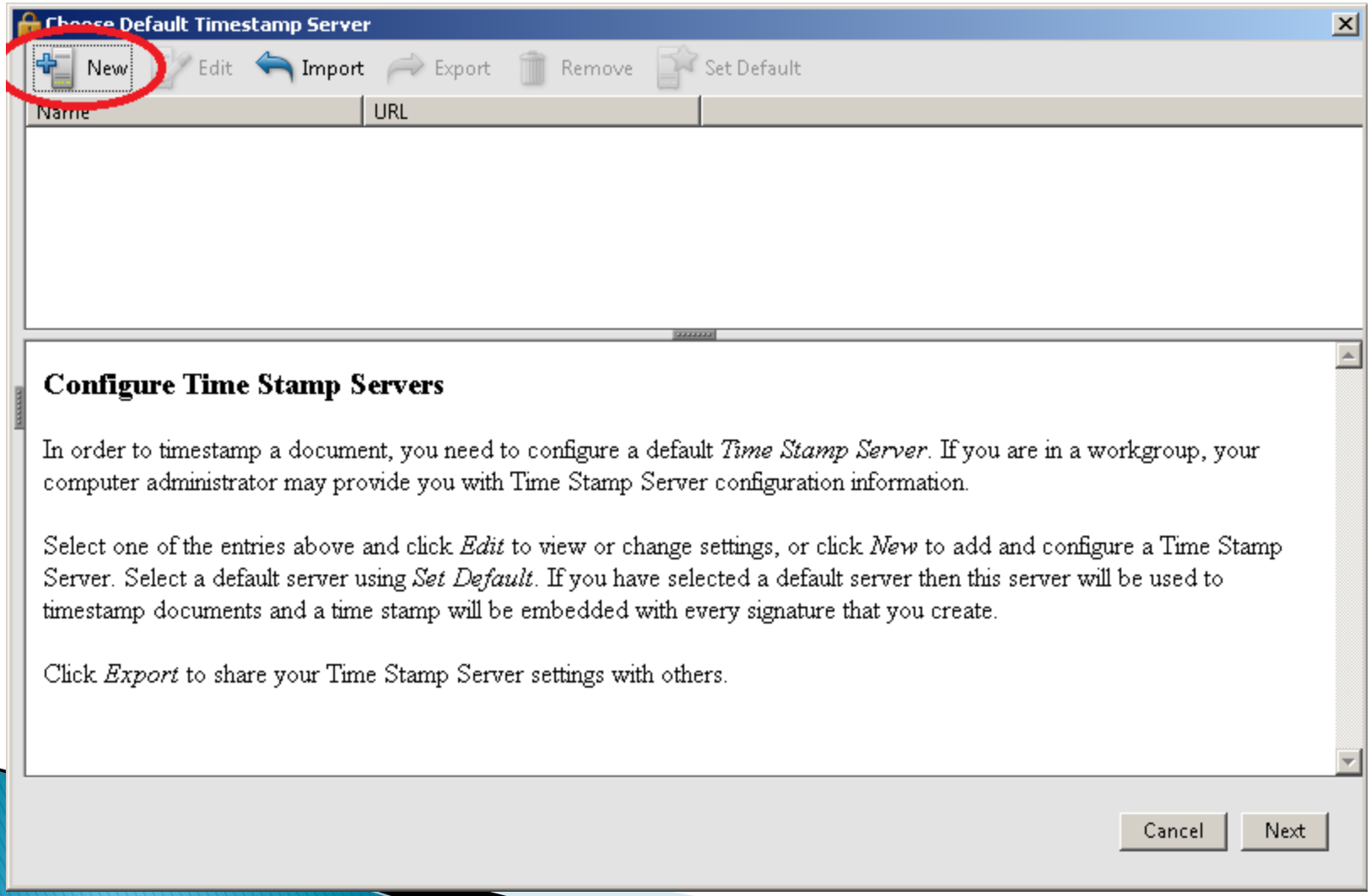
- RFC 3628
- https://tools.ietf.org/html/rfc3628#section-5.1

- **NOTE 1:**
- Without additional measures the relying party may not be able to ensure the validity of a time-stamp token beyond the end of the validity period of the supporting certificate.
- See Annex C on verification of the validity of a time-stamp token beyond the validity period of the TSU's certificate.

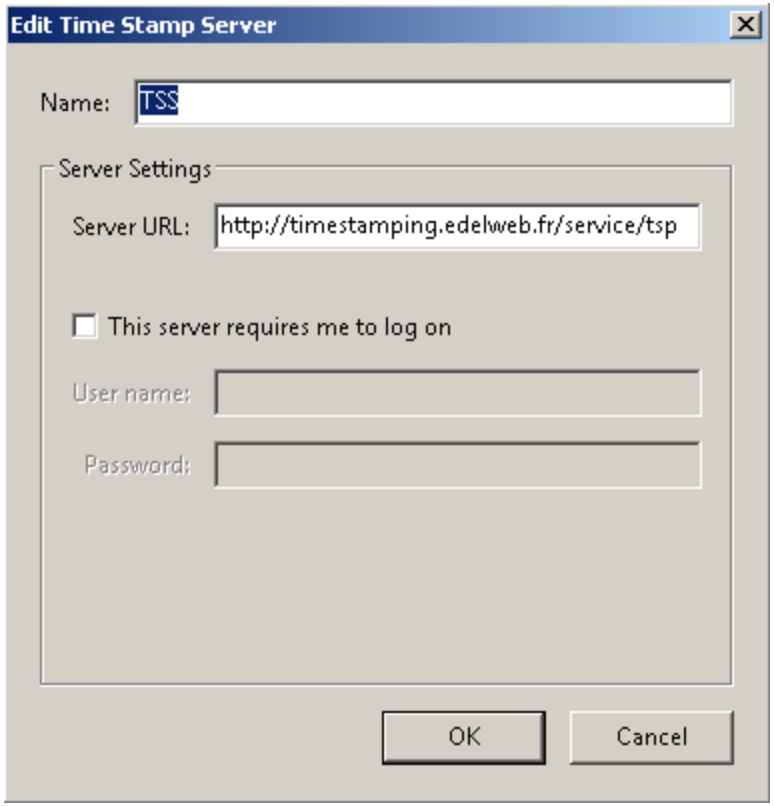
Time stamp a PDF file



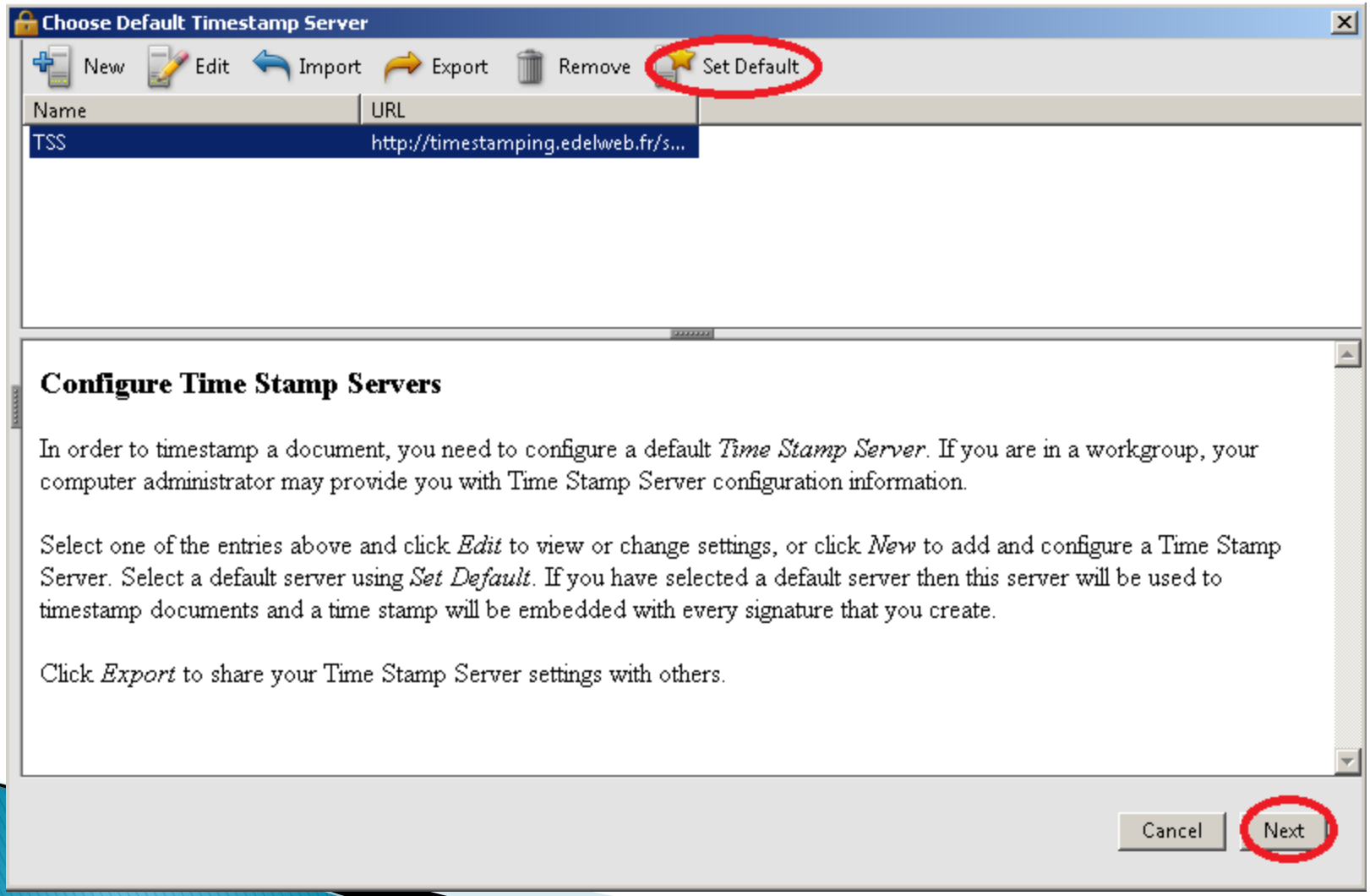
Time stamp a PDF file



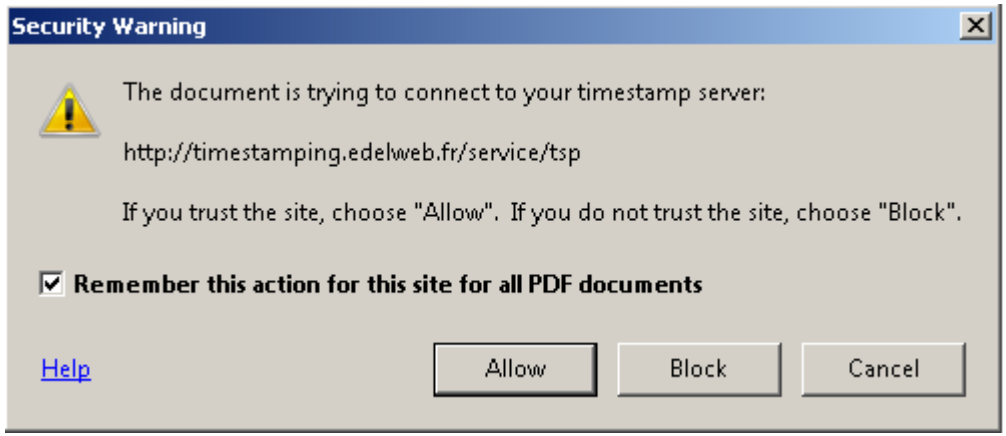
Time stamp a PDF file



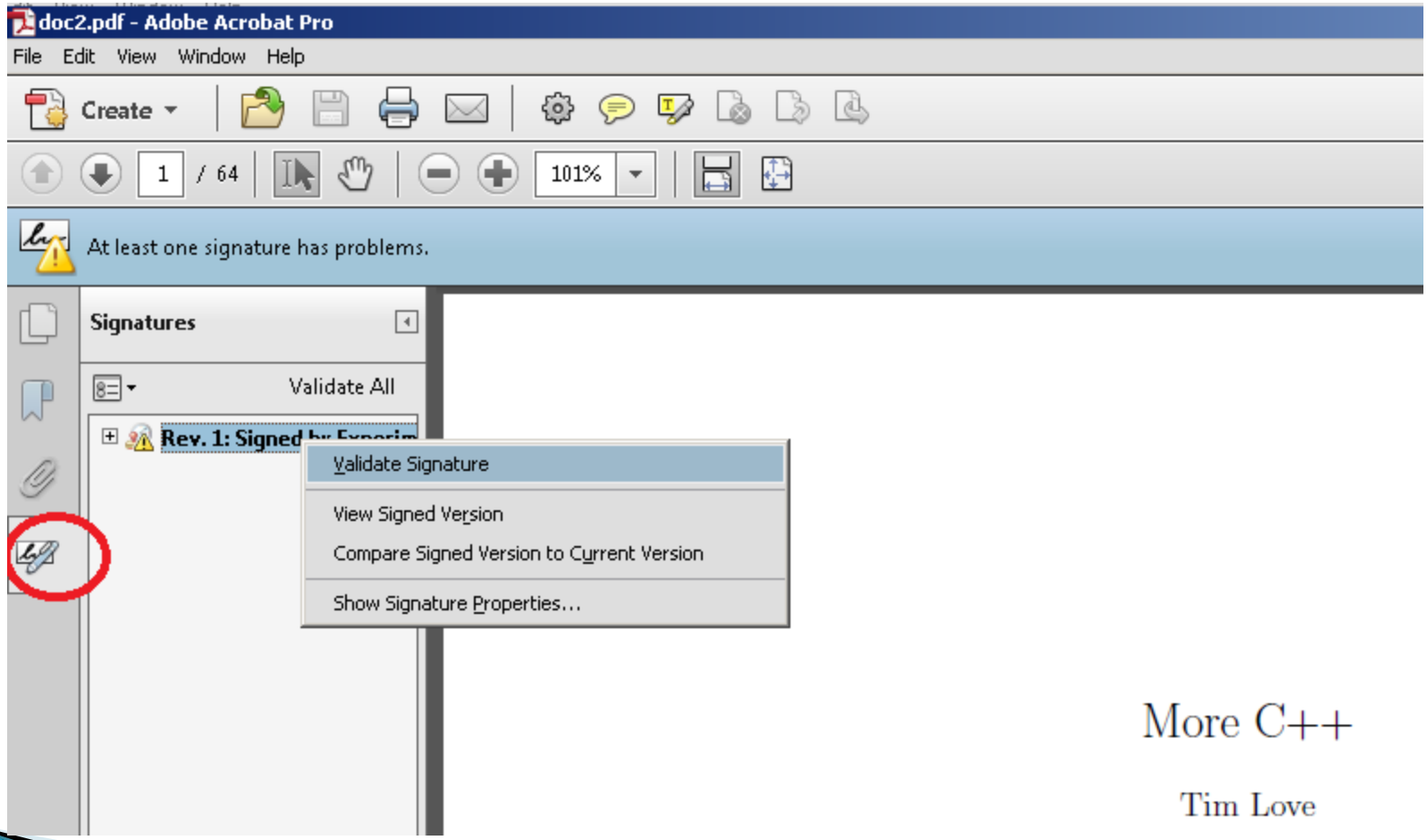
Time stamp a PDF file



Time stamp a PDF file



Verify a timestamp from a PDF file

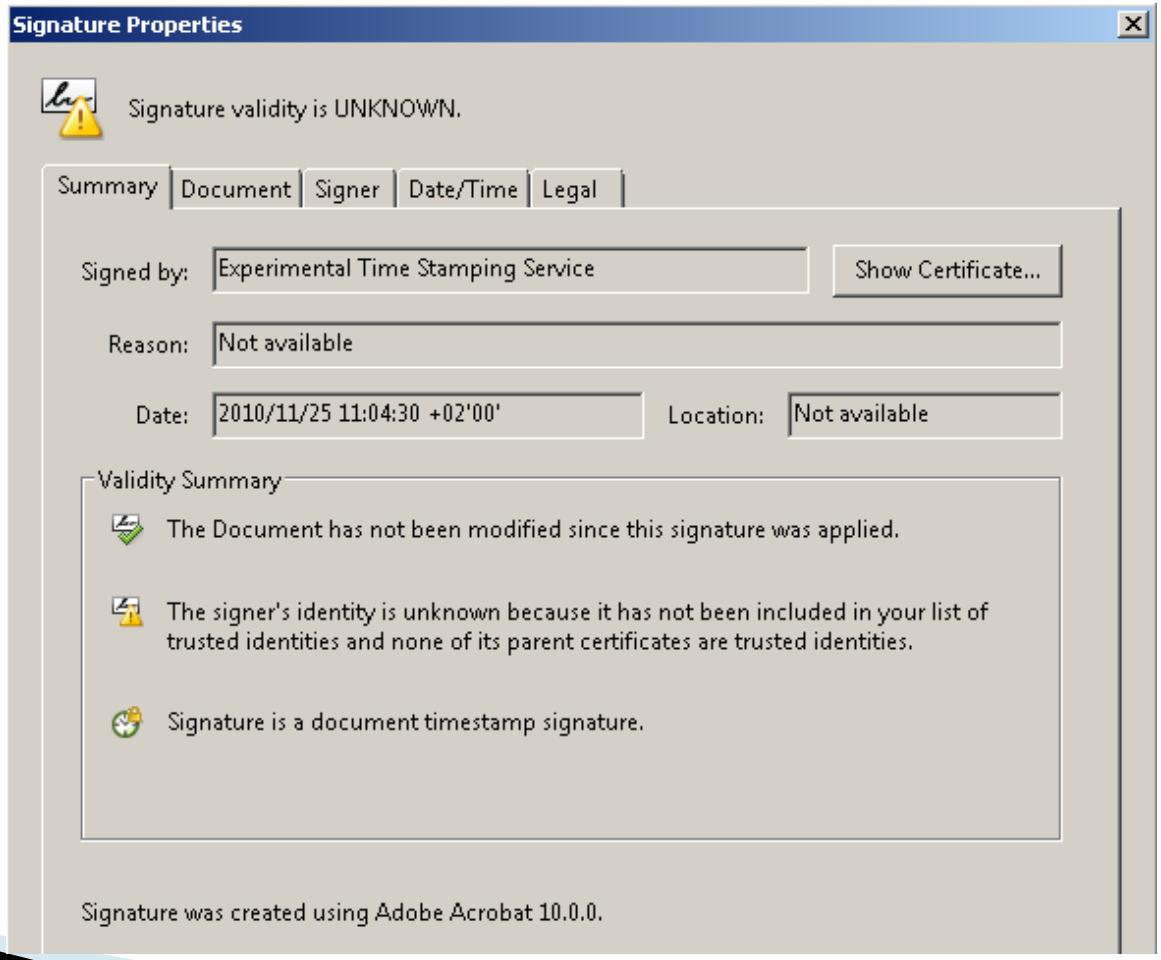


More C++

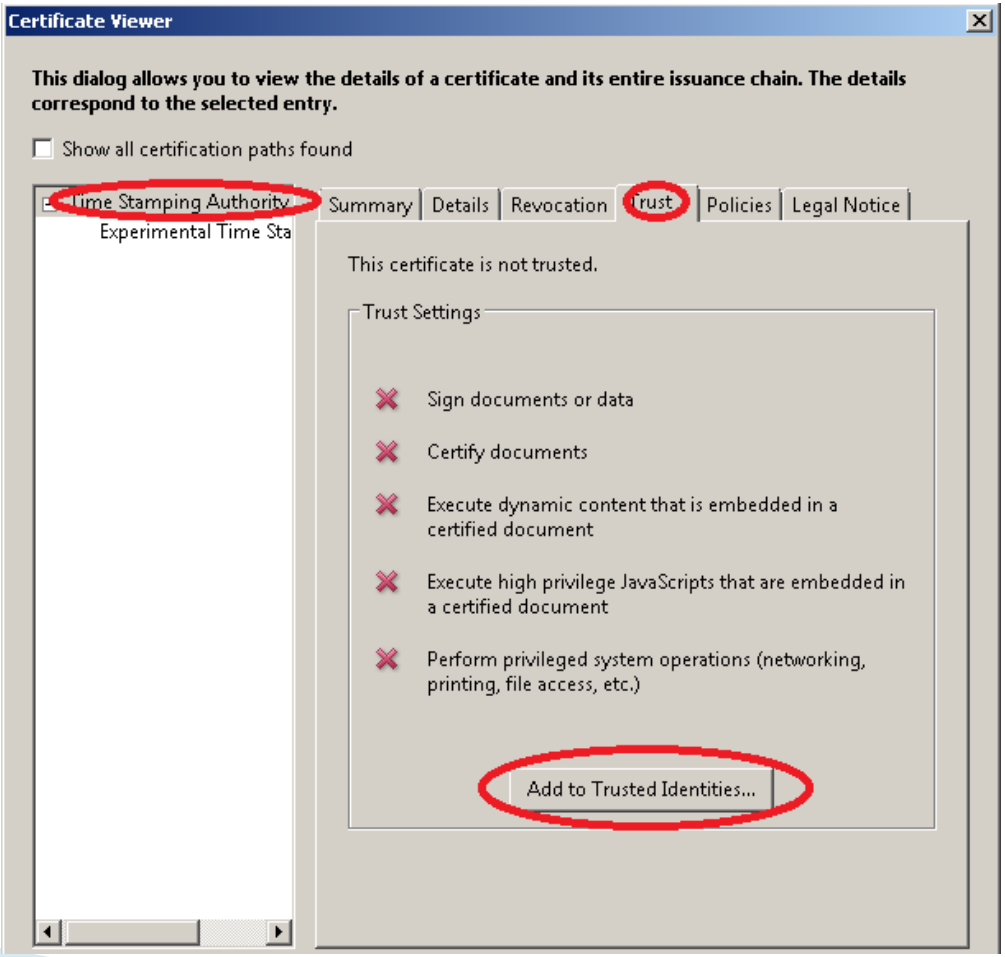
Tim Love

Verify a timestamp from a PDF file

Verify a timestamp from a PDF file

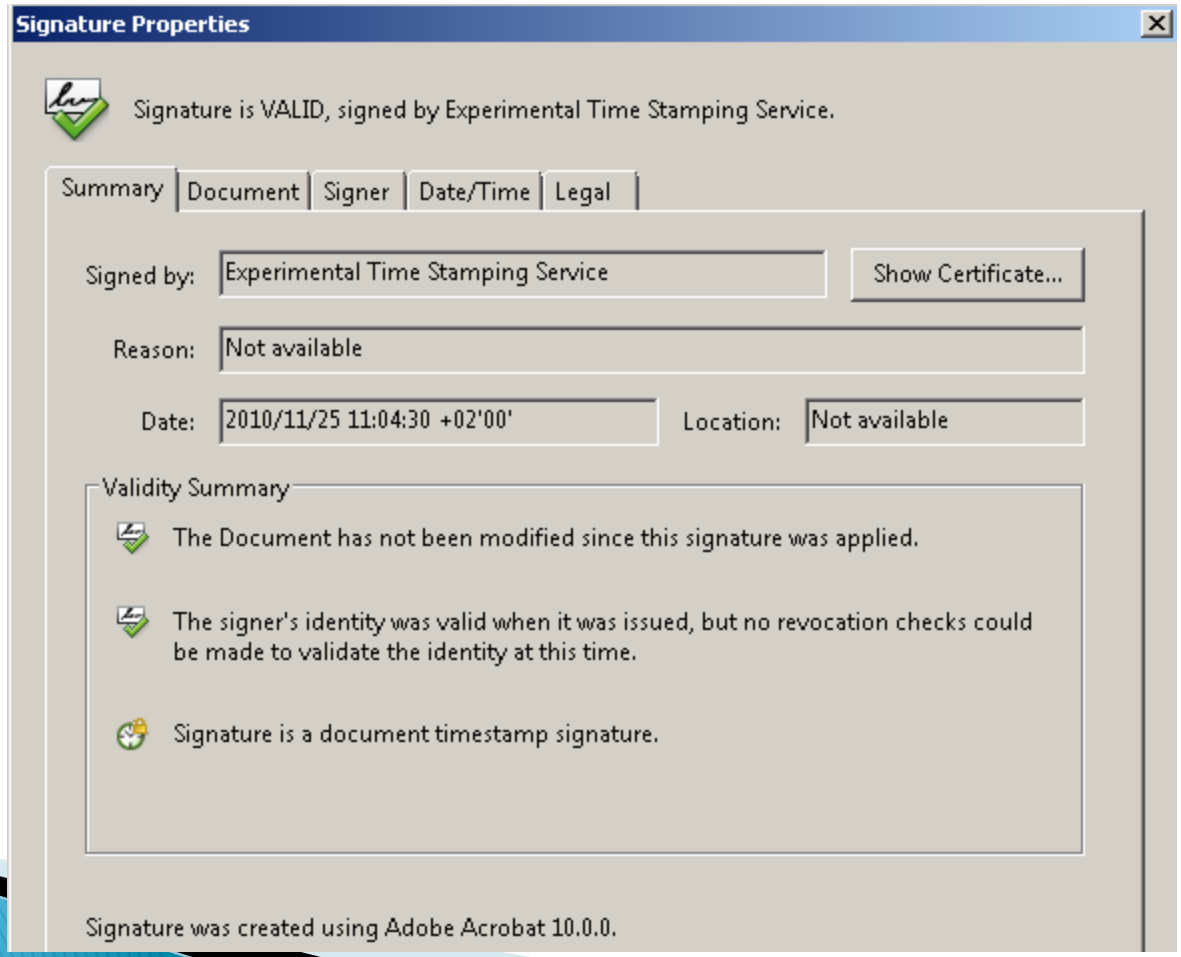


Verify a timestamp from a PDF file



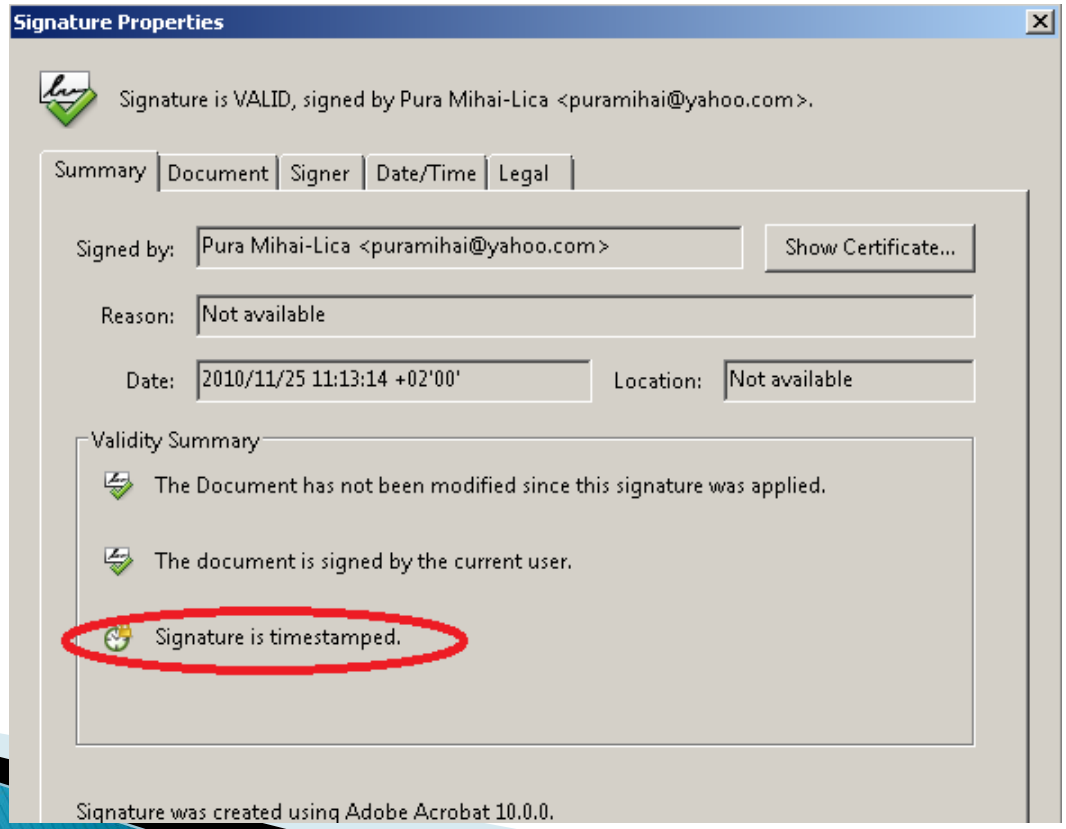
Verify a timestamp from a PDF file

- Close dialog box and revalidate signature



Sign and timestamp a PDF file

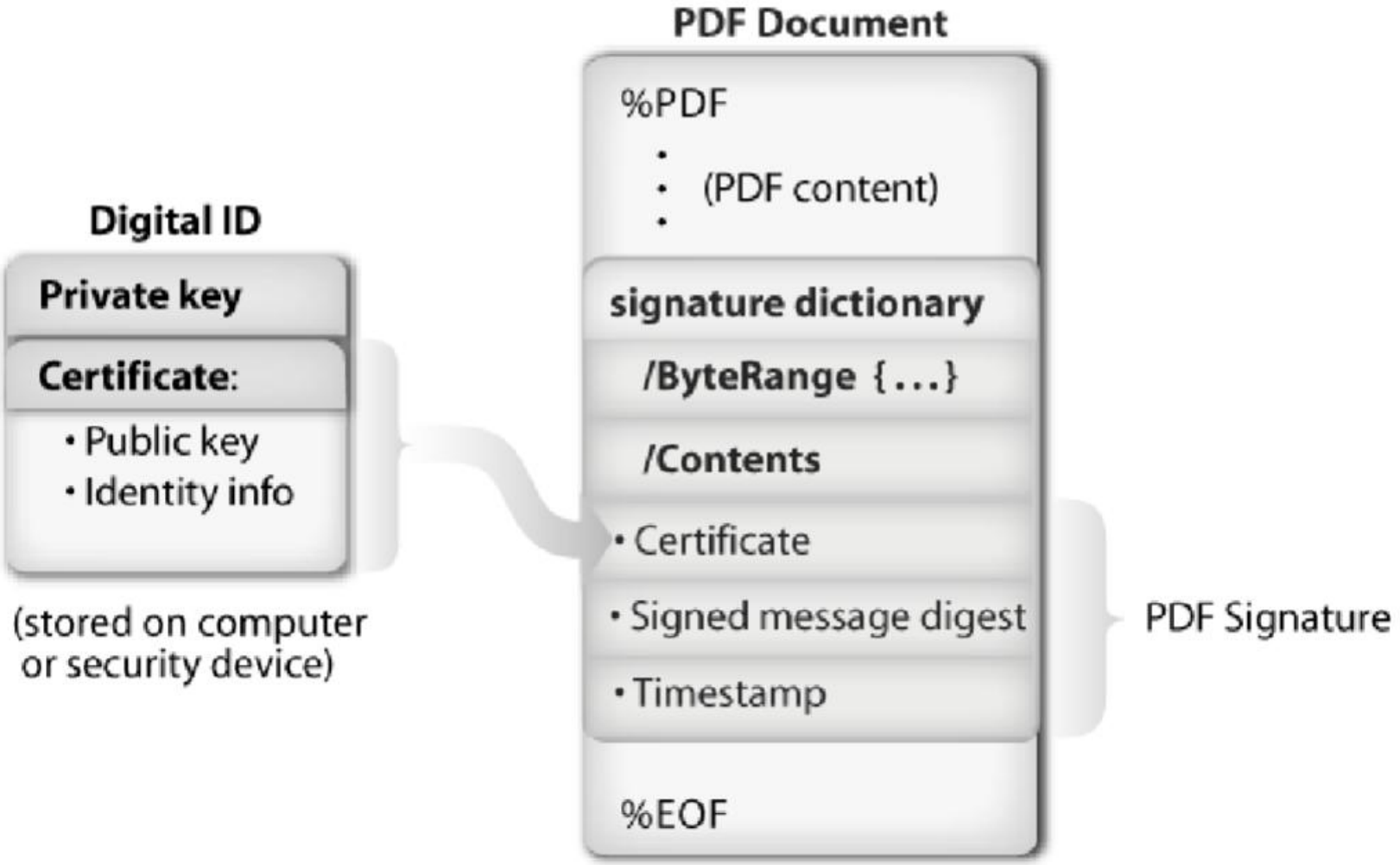
- After configuring a time stamp server, every digital signature that will be created will be time stamped.



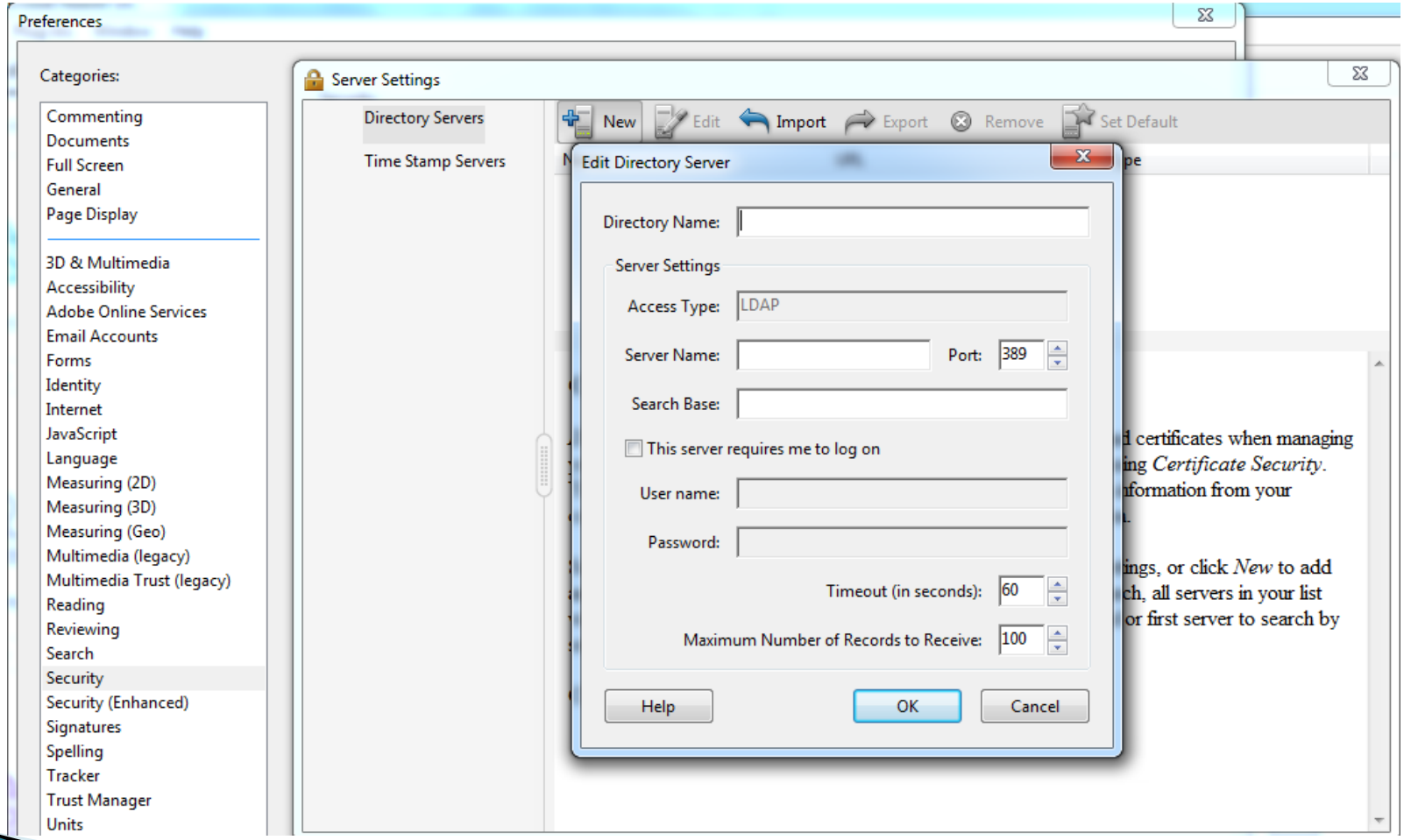
Timestamped URL screenshot

- https://freetsa.org/index_en.php

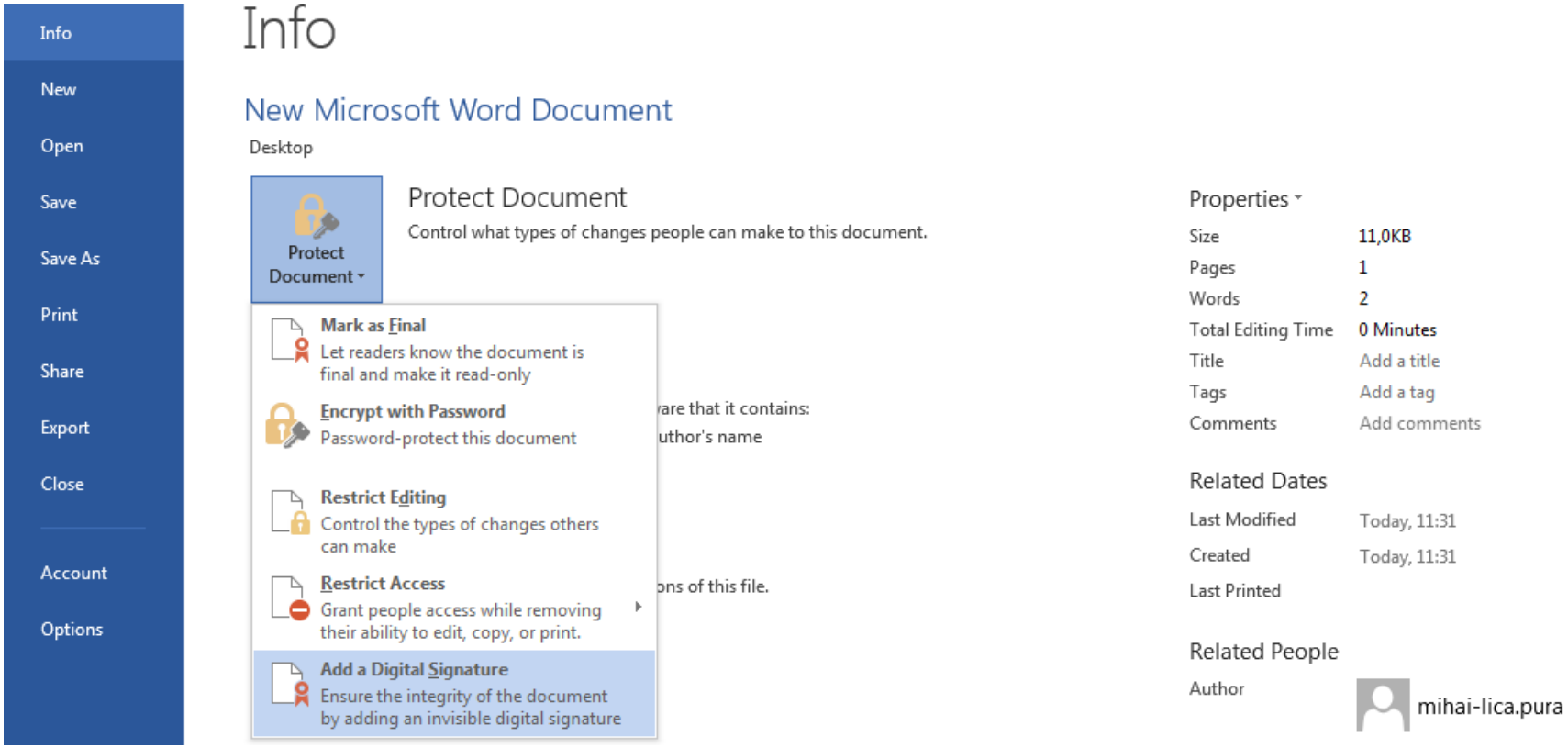
PDF signatures



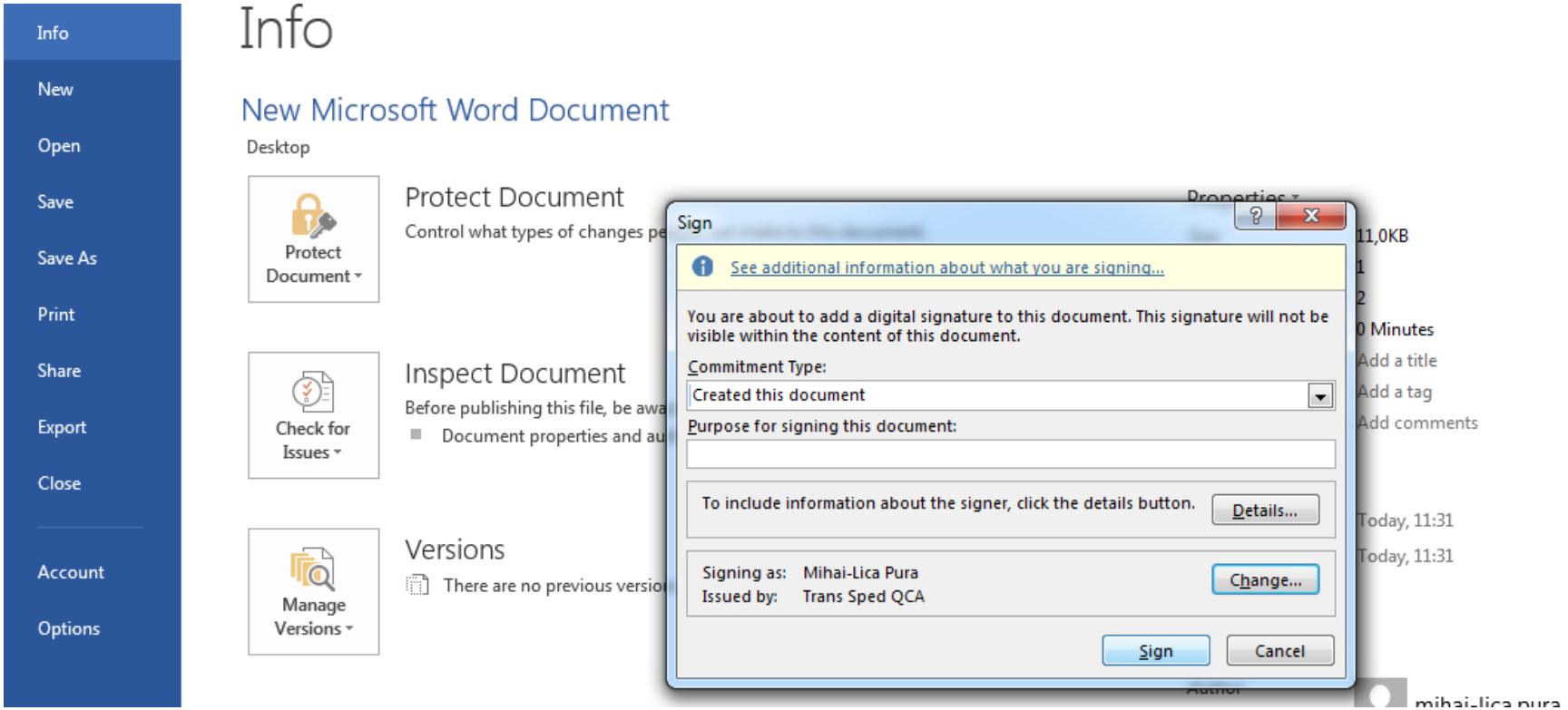
Using LDAP for searching certificates



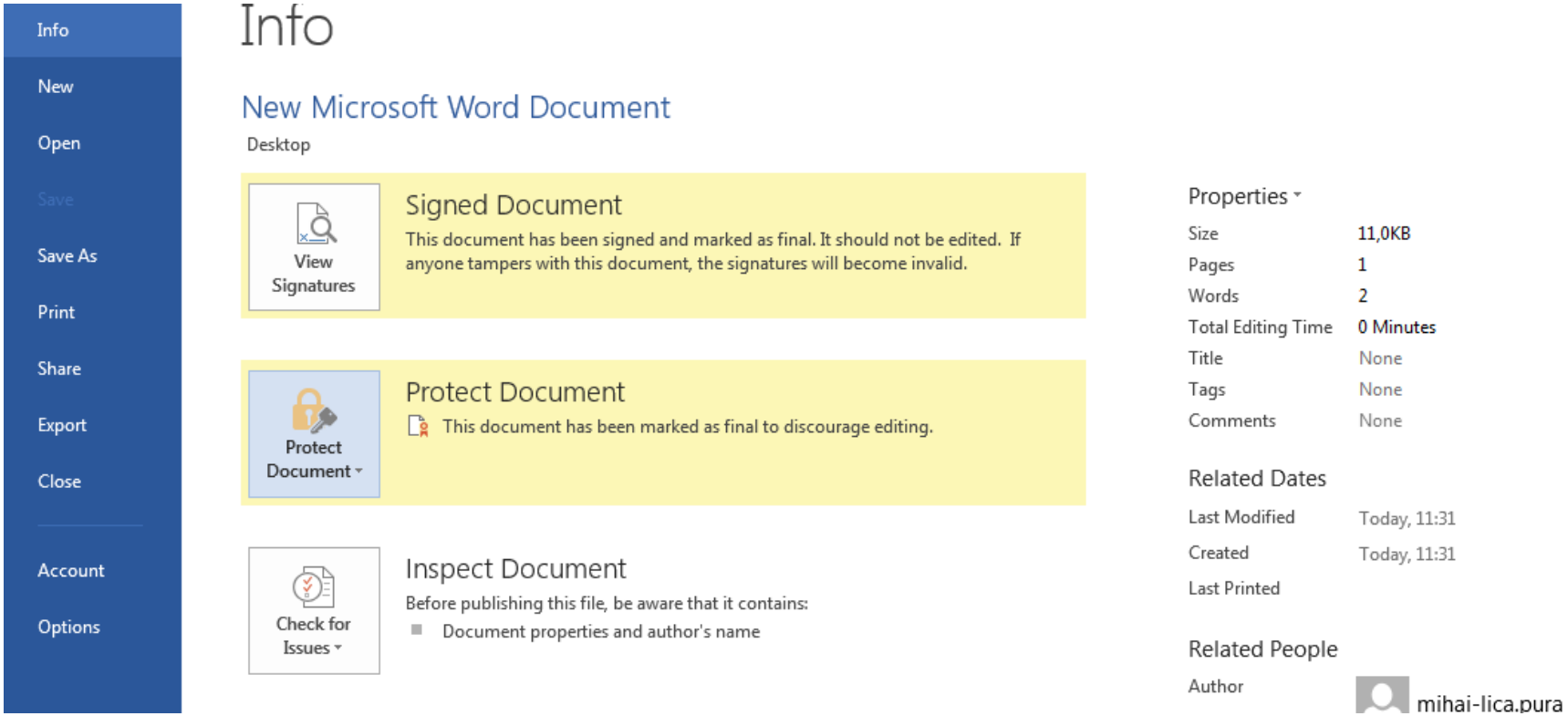
Sign Microsoft Office Documents



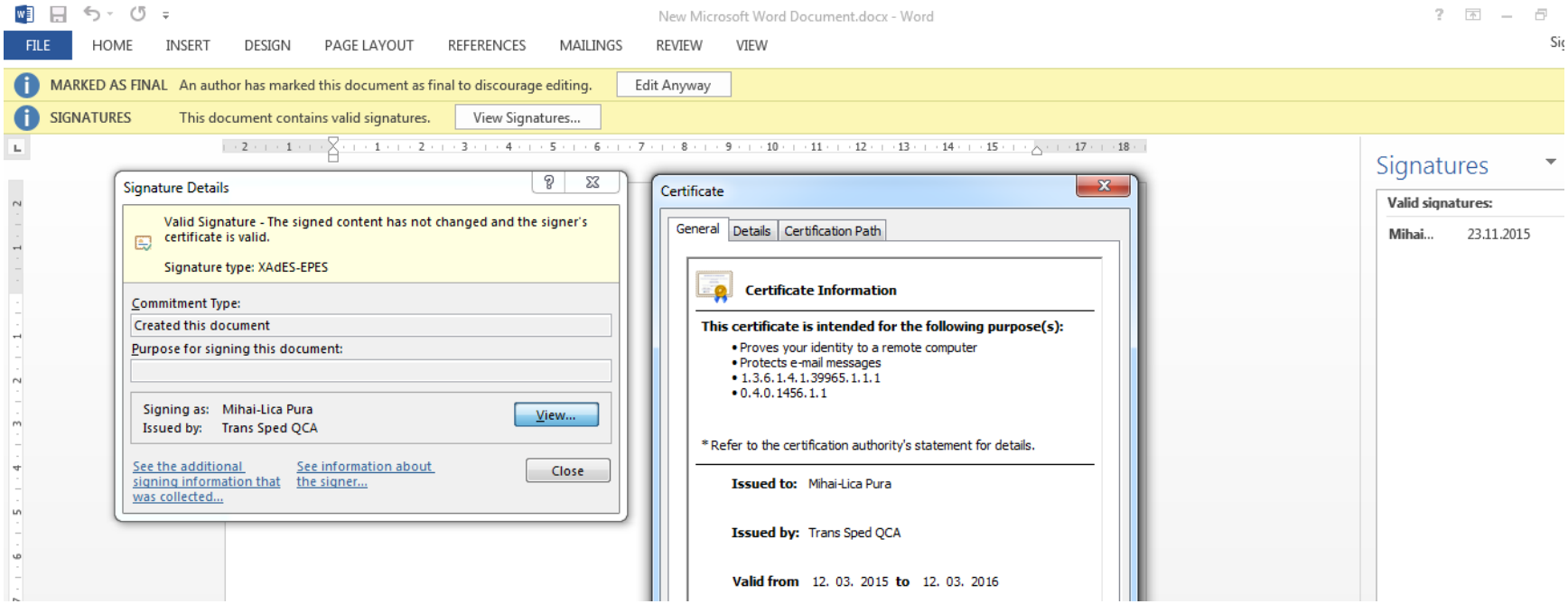
Sign Microsoft Office Documents



Sign Microsoft Office Documents



Sign Microsoft Office Documents



Sign Microsoft Office Documents

- The same steps apply for:
  - Microsoft PowerPoint
  - Microsoft Excel

Sign/Verify any kind of file

- **Exercise scenarios:**
- 1. Signing with an expired certificate
- 2. Signing with a revoked certificate
- 3. Signing with a valid certificate
- 4. Verifying signatures obtained from 1, 2 and 3

Sign/Verify any kind of file

- **Exercise scenarios:**

- 5. Verifying a signature after the certificate has expired
  - certificate was valid at the moment of signing

- 6. Verifying a signature after the certificate was revoked
  - certificate was valid at the moment of signing

Sign/Verify any kind of file

- 7. Signing with a timestamp for the document
- 8. Signing with a timestamp for the signature
- 9. Cosigning a signed file
- 10. Signing with a detached signature
- 11. Signing a PDF file (ProSigner)
- 12. Signing Microsoft Office files: docx, xlsx, pptx (ProSigner)

Sign/Verify any kind of file

- 13. Encrypting a file for a recipient
- 14. Encrypting a file for multiple recipients
- 15. Decrypting a file
- 16. Sign a file, then encrypt it for one or more recipients
- 17. Decrypt an encrypted file, then verify signature

Sign/Verify any kind of file

- **PKCS #7**: Cryptographic Message Syntax Standard
- http://www.rsa.com/rsalabs/node.asp?id=2129

- ASN.1 compiler
  - https://holtstrom.com/michael/tools/asn1decoder.php
  - http://phpseclib.sourceforge.net/x509/asn1parse.php
  - http://lionet.info/asn1c/asn1c.cgi

Sign/Verify any kind of file

- **DeskSeal**

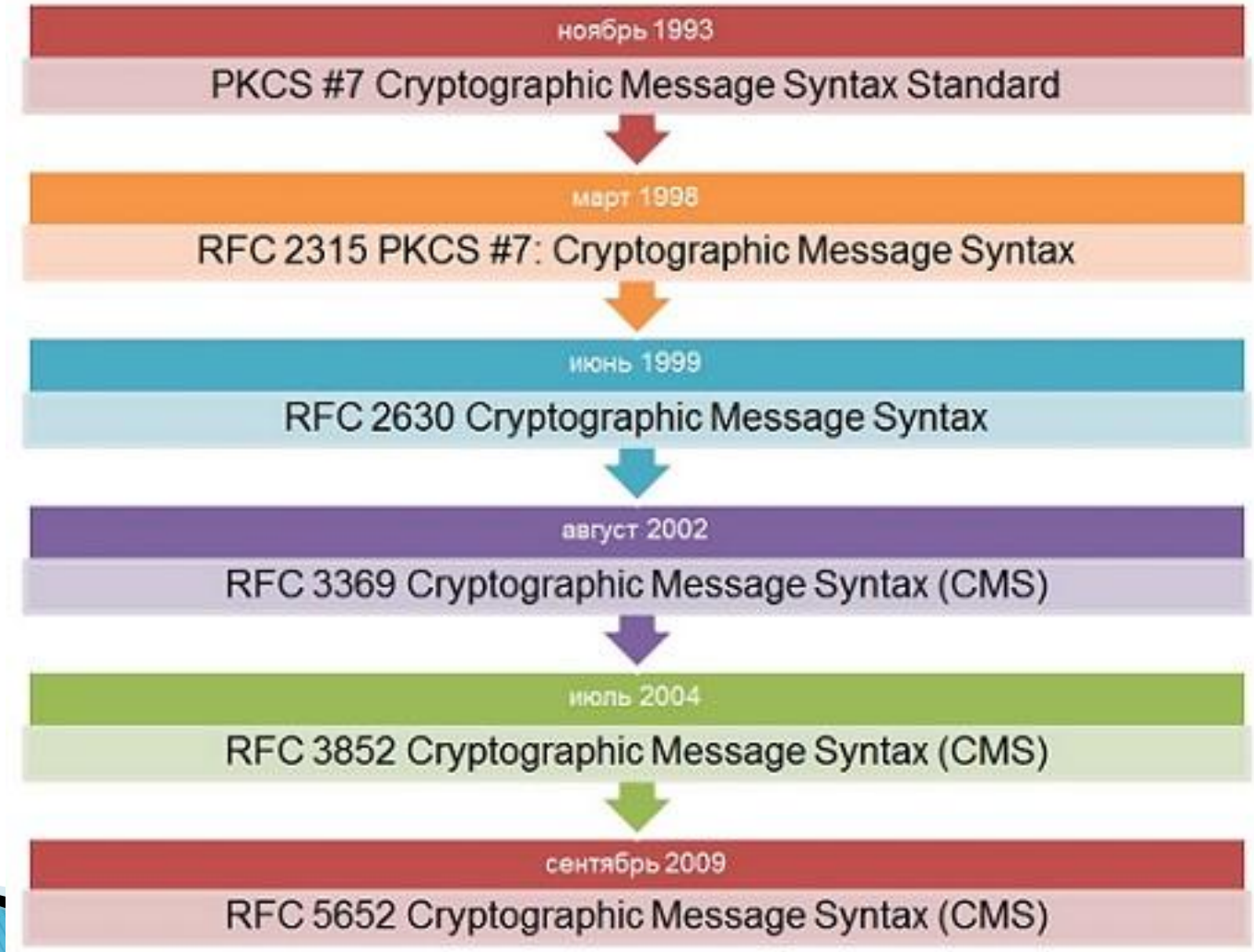
- Attached/Detached Signatures
- Time stamp signature
- Co signatures

Sign/Verify any kind of file

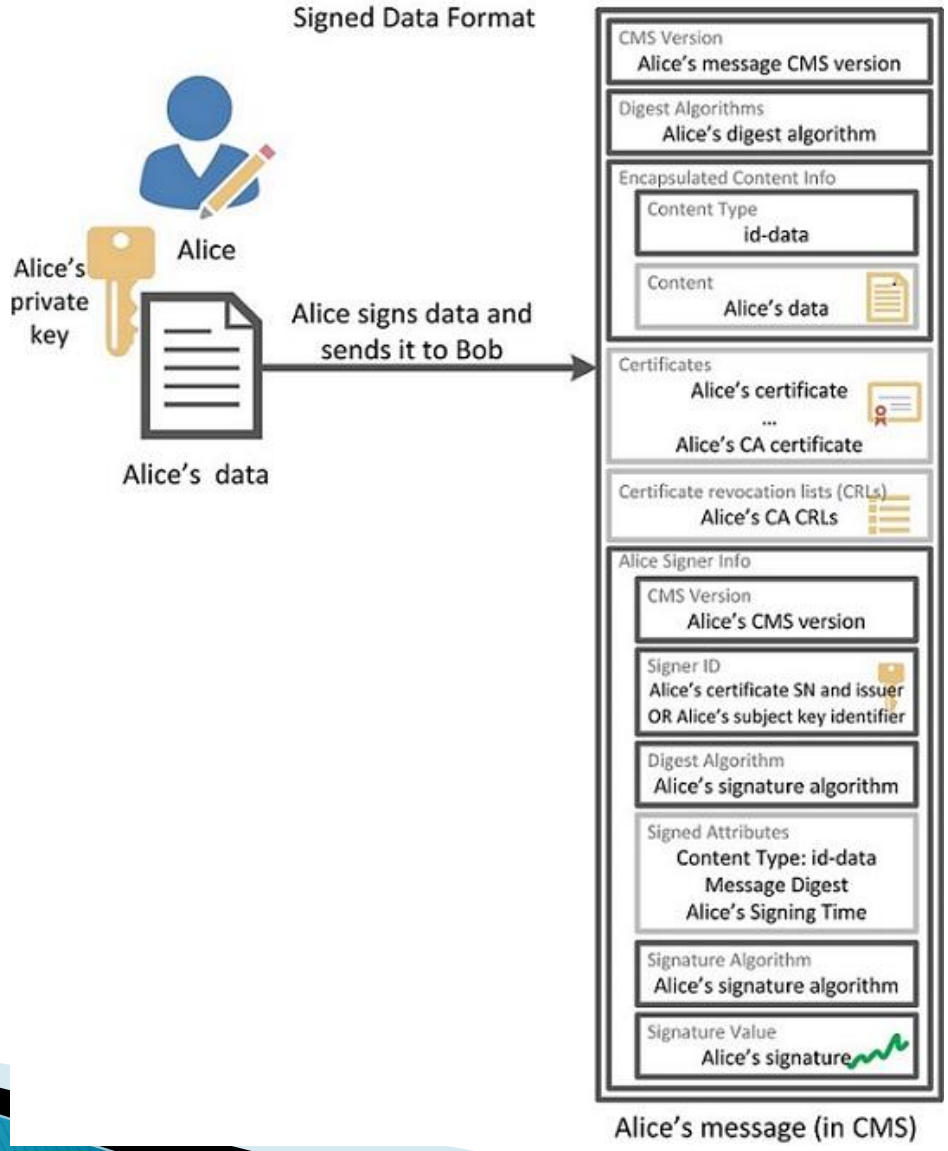
- **ProSigner**

- **LDAP configuration**
  - LDAP over SSL
- **OCSP configuration**
  - Signing validation request
  - Verify OCSP server response
- **PDF signing configuration**
  - Add a background image to the signature

PKCS#7 signatures



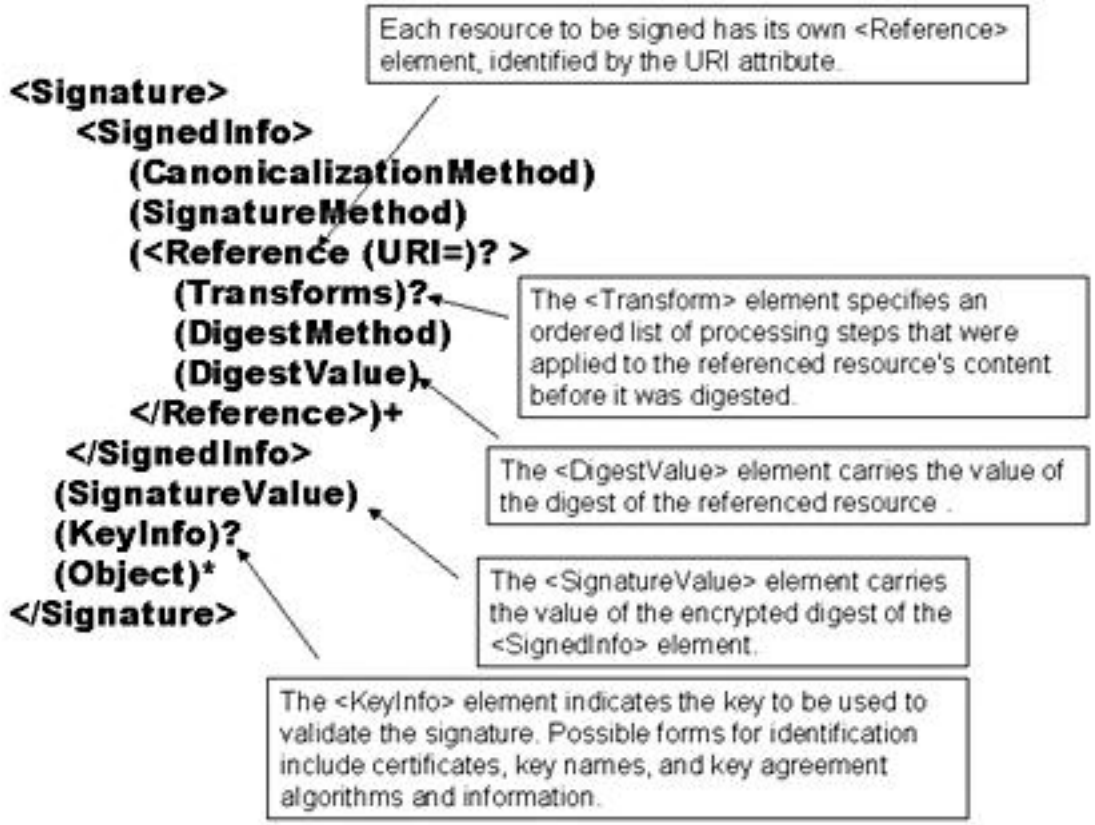
PKCS#7 signatures



Sign/verify XML files (XMLDSig)

- **XMLDSig specification**
  - http://www.w3.org/TR/xmldsig-core/
- **XML Signer**
  - https://www.signfiles.com/xml-signer/
  - Signs XML files (XML Signer)
  - And verifies XML signatures (XML Signer Viewer)
- Sample XML file to sign
  - https://www.w3schools.com/xml/note.xml

Sign/verify XML files (XMLDSig)



https://www.xml.com/pub/a/2001/08/08/xmldsig.html

Sign/verify XML files (XMLDSig)

- **Determine which resources are to be signed**
  - will take the form of identifying the resources through a Uniform Resource Identifier (URI)

Sign/verify XML files (XMLDSig)

- **Calculate the digest of each resource**
  - each referenced resource is specified through
    - a <**Reference**> element
    - and its digest
      - (calculated on the identified resource and not the <Reference> element itself)
      - placed in a <**DigestValue**> child element like
  - If reference URI is “” it refers the content of the current XML document
  - The <**DigestMethod**> element identifies the algorithm used to calculate the digest

Sign/verify XML files (XMLDSig)

- **Collect the Reference elements**
  - Collect the <**Reference**> elements (with their associated digests) within a <**SignedInfo**> element like
  - The <**CanonicalizationMethod**> element indicates the algorithm was used to canonize the <SignedInfo> element
  - Different data streams with the same XML information set may have different textual representations, e.g. differing as to whitespace

Sign/verify XML files (XMLDSig)

- **Collect the Reference elements**
  - To help prevent inaccurate verification results, XML information sets **must first be canonized** before extracting their bit representation for signature processing
  - The <**SignatureMethod**> element identifies the algorithm used to produce the signature value

Sign/verify XML files (XMLDSig)

- **Signing**
  - Calculate the digest of the <**SignedInfo**> element, sign that digest and put the signature value in a <**SignatureValue**> element
- **Add key information**
  - If keying information is to be included, it will be placed in a <**KeyInfo**> element
  - Here the keying information contains the X.509 certificate for the sender, which would include the public key needed for signature verification

Sign/verify XML files (XMLDSig)

- **Enclose in a Signature element**
  - Place the
    - **<SignedInfo>**
    - **<SignatureValue>**,
    - **<KeyInfo>**
  - into a **<Signature>** element

  - The <Signature> element comprises the XML signature

Signature expiration

- What is the problem?

- Exercises using Adobe Acrobat Reader DC

Signature expiration

- *User A*
  - signs his document using a signature scheme
  - applies to a TSA to get a timestamp on the signature

- *User B* can do the *User A*'s signature validation
  - this involves validation of the *User A*'s certificate
- many years later when the certificate of *User A* can be already expired or revoked

Signature expiration

- The validation decision are based on the timestamp applied on the signature
  - The signature was made before the timestamp
  - The applied timestamp is considered the signing-time

- Without the timestamp, the *User B* can not be sure about the status of the User A's certificate at the *signing-time*

Advanced Electronic Signatures

- CAdES
- http://www.etsi.org/deliver/etsi_ts/101700_101799/101733/

- PAdES
- http://www.etsi.org/deliver/etsi_ts/102700_102799/10277801/01.01.01_60/ts_10277801v010101p.pdf

- XAdES
- https://portal.etsi.org/webapp/WorkProgram/Report_WorkItem.asp?WKI_ID=28064

Signatures with Long-Term Validation Material

- Validation material
  - is needed to validate the signature
    - certificates
    - and revocation information
      - OCSP responses
      - and/or CRLs
  - is included in the signature

- This signature allows the validation as long as there are no technology changes
  (e.g. algorithms becomes weak)

Signatures with Long-Term Validation Material

- Examples:
  - CAdES-B-LT [i.11],
  - XAdES-B-LT [i.14],
  - PAdES-B-LT [i.16]

Signatures providing Long Term Availability and Integrity of Validation Material

- Re-compute
  - the hash of the originally signed document
  - and the signed attributes
- with a specific hash algorithm

- Protect
  - re-computed values
  - and previously added validation material and timestamps
- with a new timestamp

Signatures providing Long Term Availability and Integrity of Validation Material

- The original signature can be validated as long as:
  - the latest added timestamp can be validated
  - and the latest used hash-algorithm is still trustworthy

- Examples:
  - CAdES-B-LTA [i.11], CAdES-E-A [i.12],
  - XAdES-B-LTA [i.14], XAdES-E-A [i.15],
  - PAdES-B-LTA [i.16], PAdES-E-LTV [i.17]

Signatures providing Long Term Availability and Integrity of Validation Material

- A Signature providing
  - Long Term Availability
  - and Integrity of Validation Material
- is suitable for long-term preservation

Signatures providing Long Term Availability and Integrity of Validation Material

- Mechanism (similar for the different formats):
  - All the missing validation material is added
  - a hash is computed over the existing signature including the originally signed document
  - and a time-stamp is computed over the final hash computation
  - The time-stamp is added into the signature

Signatures providing Long Term Availability and Integrity of Validation Material

- Mechanism (similar for the different formats):
  - Once a signature is already in this form, the preservation service
    - only considers the validity of the last archival time-stamp
    - e.g.
      - the expiration of the signing certificate of the TSU
      - and the trustworthiness of the cryptographic algorithms used within the time-stamp

Signatures providing Long Term Availability and Integrity of Validation Material

- Each of the AdES formats provides levels suitable for long-term preservation

- In part one of CAdES [i.11], PAdES [i.16] and XAdES [i.14] baseline levels are defined:
  - CAdES-B-LTA,
  - PAdES-B-LTA and
  - XAdES-B-LTA

Signatures providing Long Term Availability and Integrity of Validation Material

- The baseline levels were designed for high interoperability.

- In addition, part two of CAdES [i.12], PAdES [i.17] and XAdES [i.15] provide more general extended levels:
  - CAdES-E-A,
  - PAdES-E-LTV and
  - XAdES-E-A

Timestamp expiration

- Timestamps (RFC 3161) are also signatures
  - they rely on certificates (the TSA certificate)
  - and thus they expire
  - also, timestamp becomes useless in the same moment as the TSA ceases operation or is compromised

Timestamp expiration

- Solutions (PKI)

- I. Use just one timestamp, but whenever the latest time stamp is about to expire
  - but *before* expiration
  - one needs to obtain a new timestamp computed over the previous timestamp

- II. get several timestamps on one file
  - regularly timestamp it before any of the TSA certificates is about to expire

Timestamp expiration

- If the TSA's private key is compromised (and thus revoked)
  - the time stamp signature cannot be trusted
  - as whoever compromised it could sign documents with old time stamps
  - users would want to be warned before accepting a time stamp signature from a compromised certificate

Timestamp expiration

- Solutions
  - Get a signed time stamp from **multiple TSAs**
    - if any one is not revoked, that signature can still be trusted
  - Use **threshold signing**
    - an attacker would have to compromise a number of parties in order to forge signatures
    - which is much less likely than compromise of a single party

Timestamp expiration

- https://crypto.stackexchange.com/questions/3291/signature-and-timestamp-for-long-term-document-archival-question

- https://en.wikipedia.org/wiki/Trusted_timestamping

Advanced Electronic Signatures

- Signing tools

- A. web:
- https://ec.europa.eu/cefdigital/DSS/webapp-demo/home

- B. desktop:
- http://xolidosign.en.softonic.com/

- C. others:
- https://www.agid.gov.it/it/piattaforme/firma-elettronica-qualificata/software-verifica

Certificate login

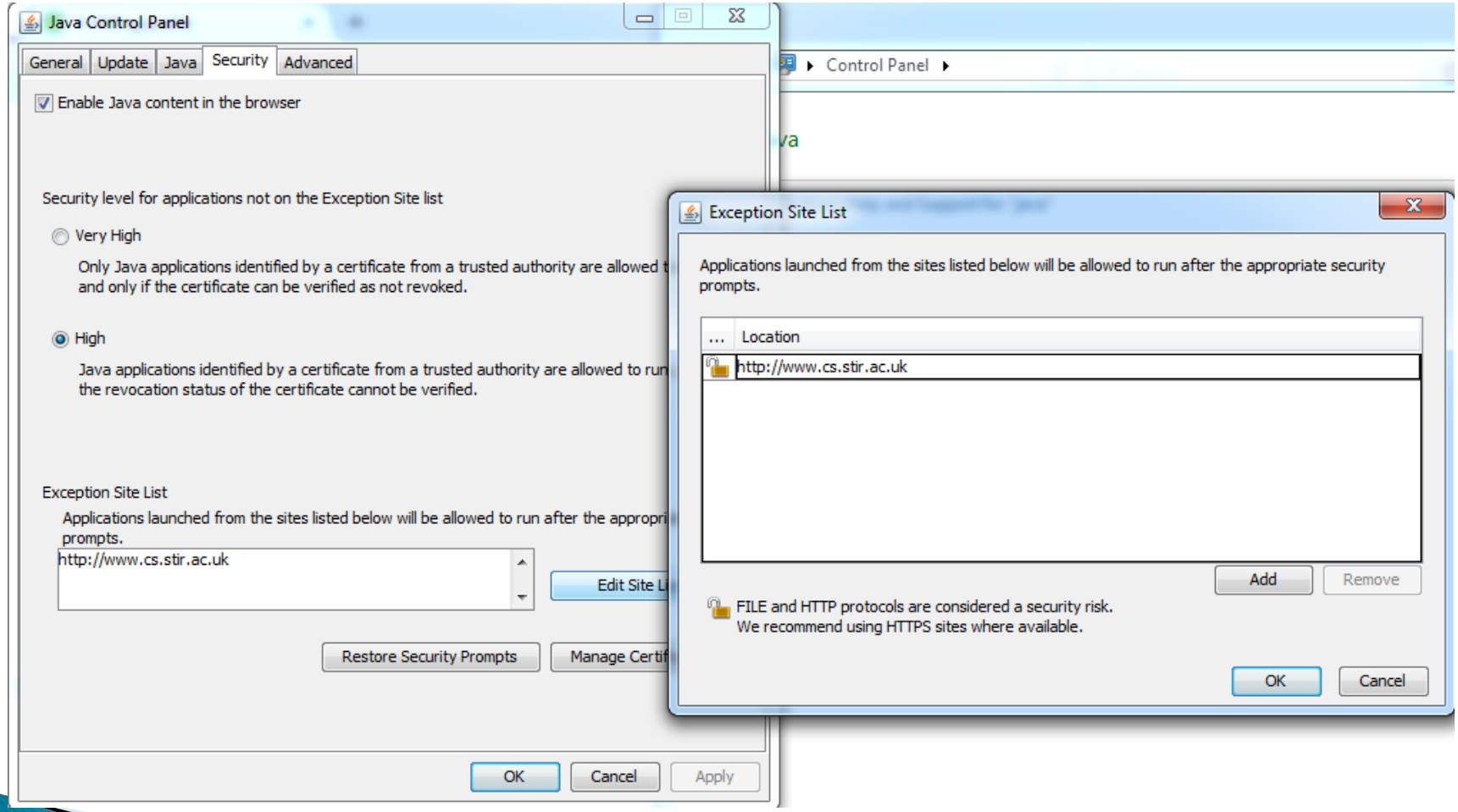
- **Centrul Național de Administrare a Registrelor Naționale Notariale**
  - https://eguvernare.stsisp.ro/
  - https://www.infonot.eu/

- https://www.cacert.org/
  - First **install cacert CA certificate**
    - **in Trusted Root Certification Authorities** (for Microsoft Edge, Microsoft Internet Explorer and Google Chrome)
    - or the corresponding certificate store in Mozilla Firefox
  - If using Google Chrome, **enable certificate request generation** in the browser settings

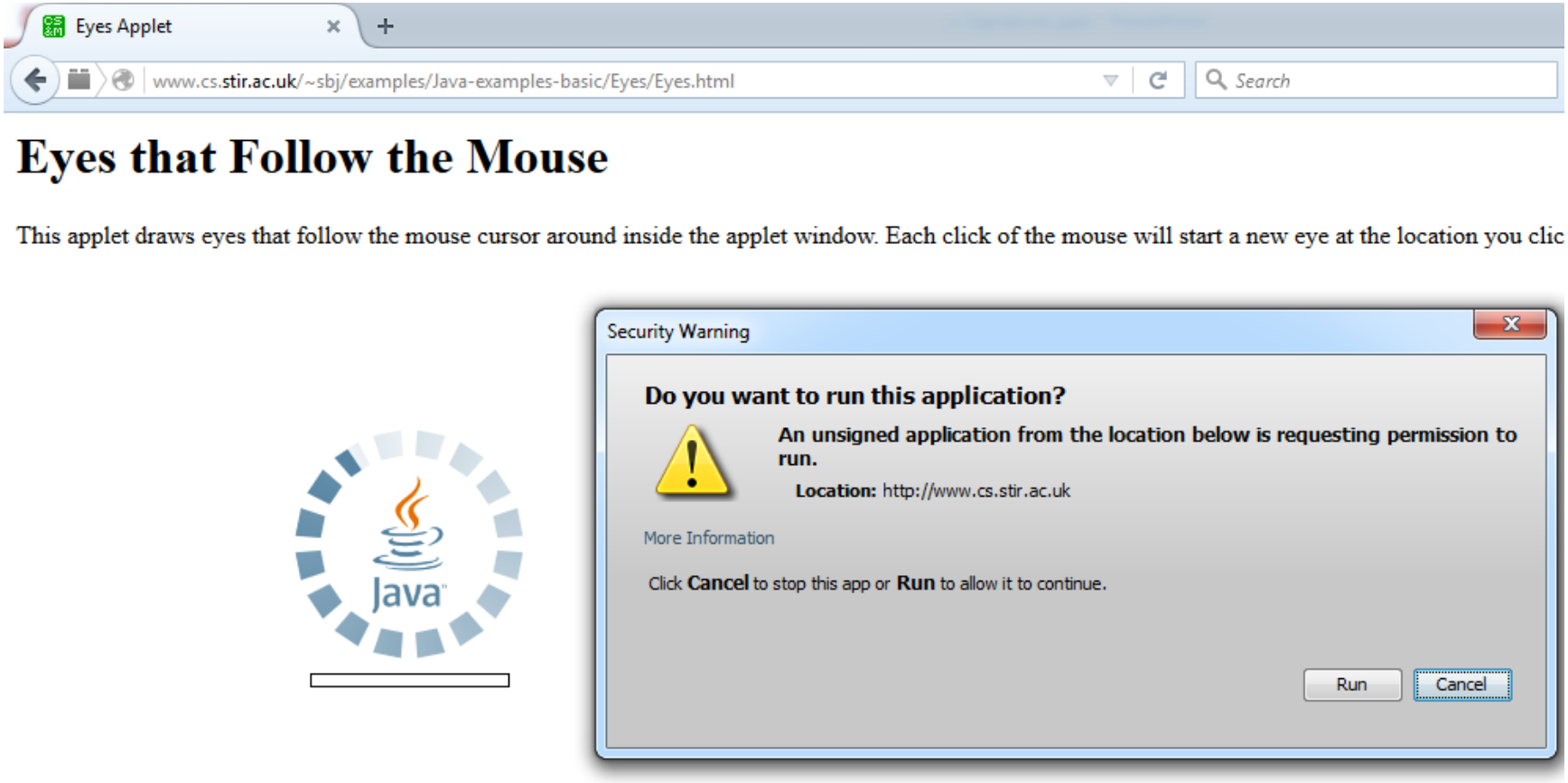
Code signing – Unsigned Java applets (obsolete)

- http://www.cs.stir.ac.uk/~sbj/examples/Java-examples-basic/
- http://www.sitepoint.com/java-applets-obsolete-still-part-web/

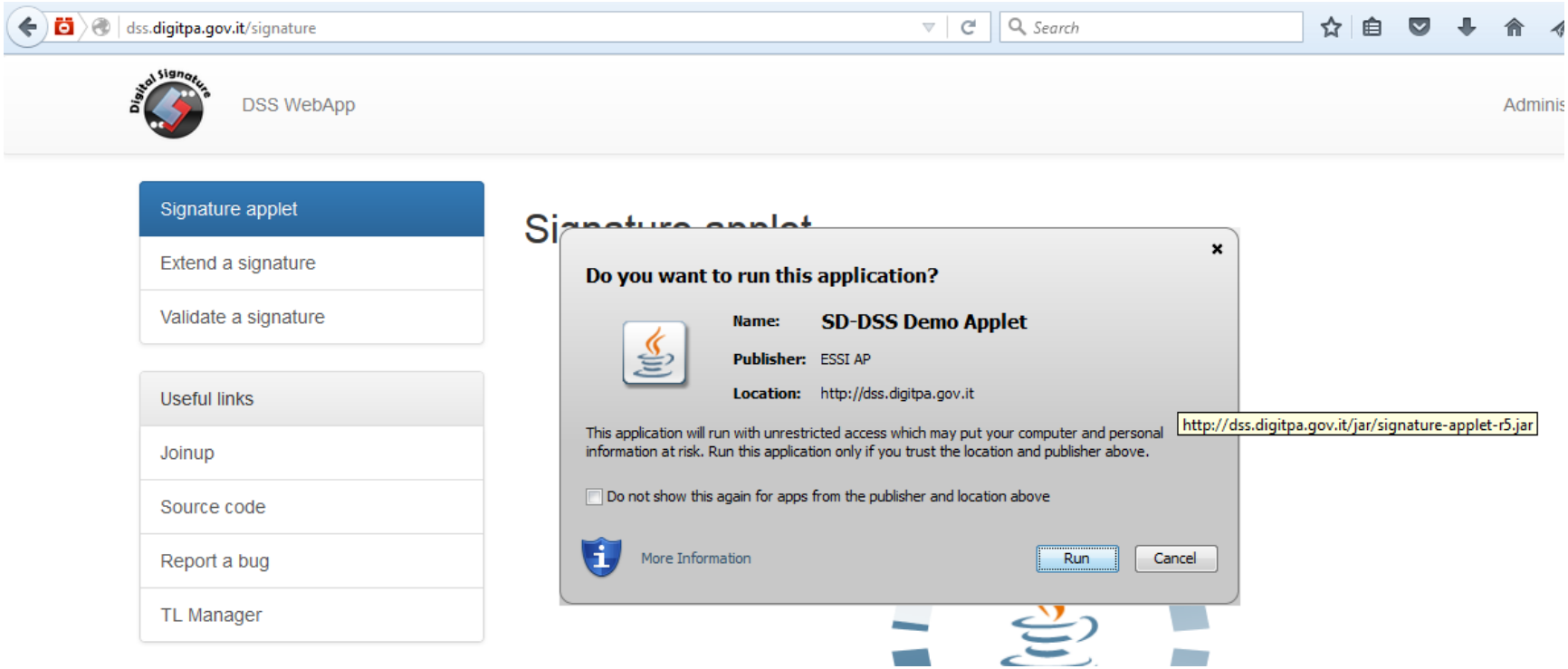
Code signing – Unsigned Java applets (obsolete)



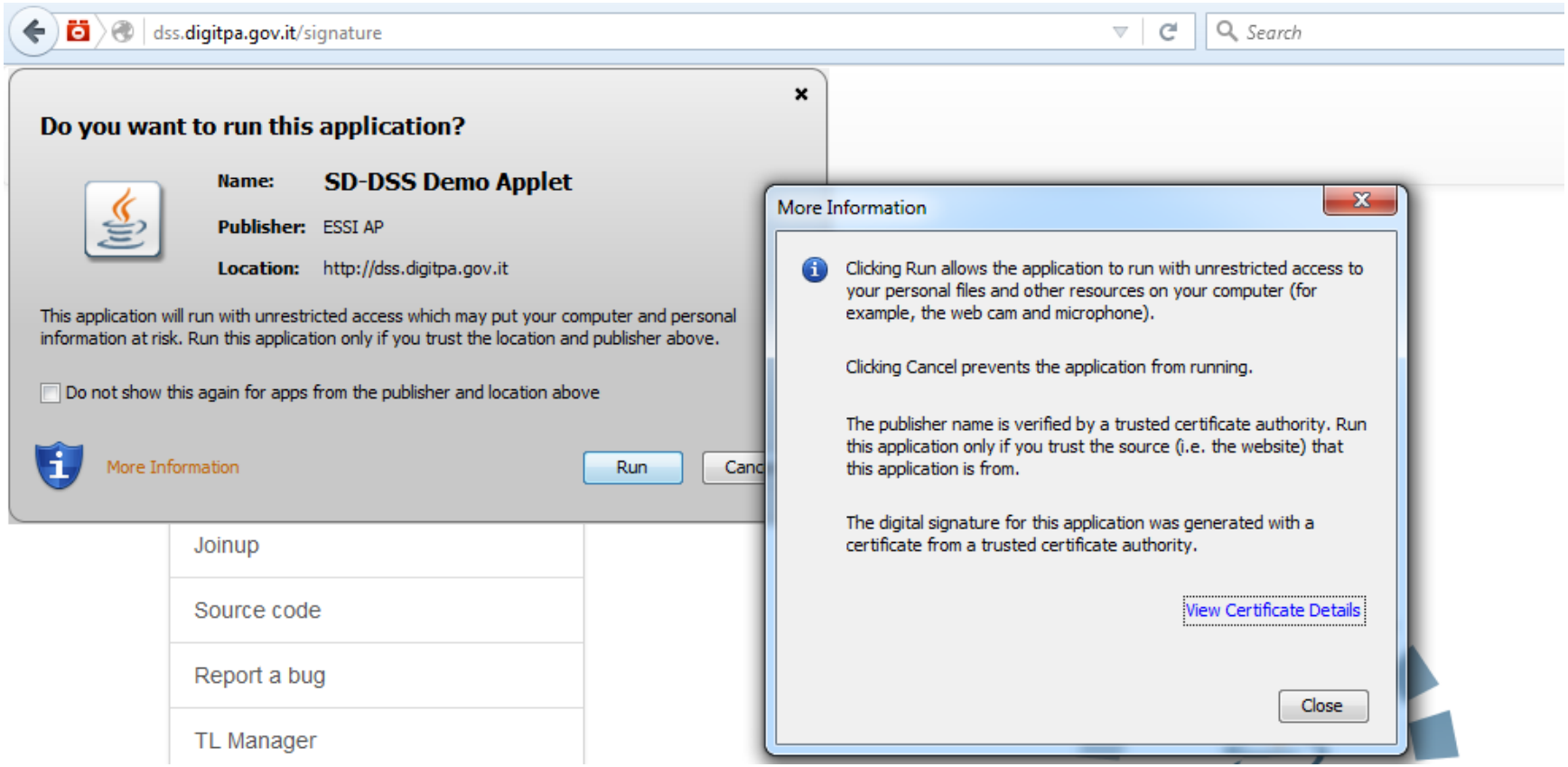
Code signing – Unsigned Java applets (obsolete)



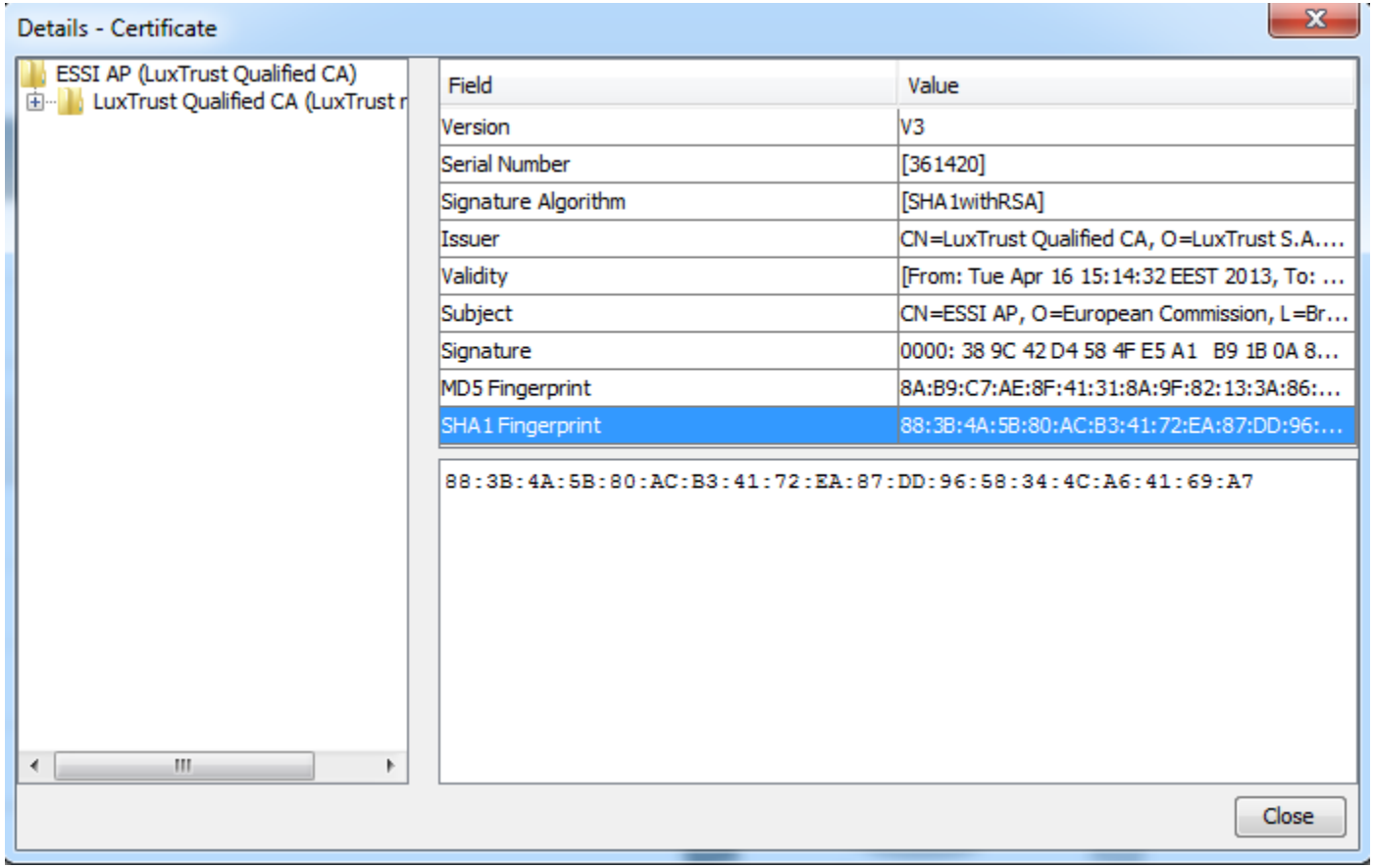
Code signing – Signed Java applets (obsolete)



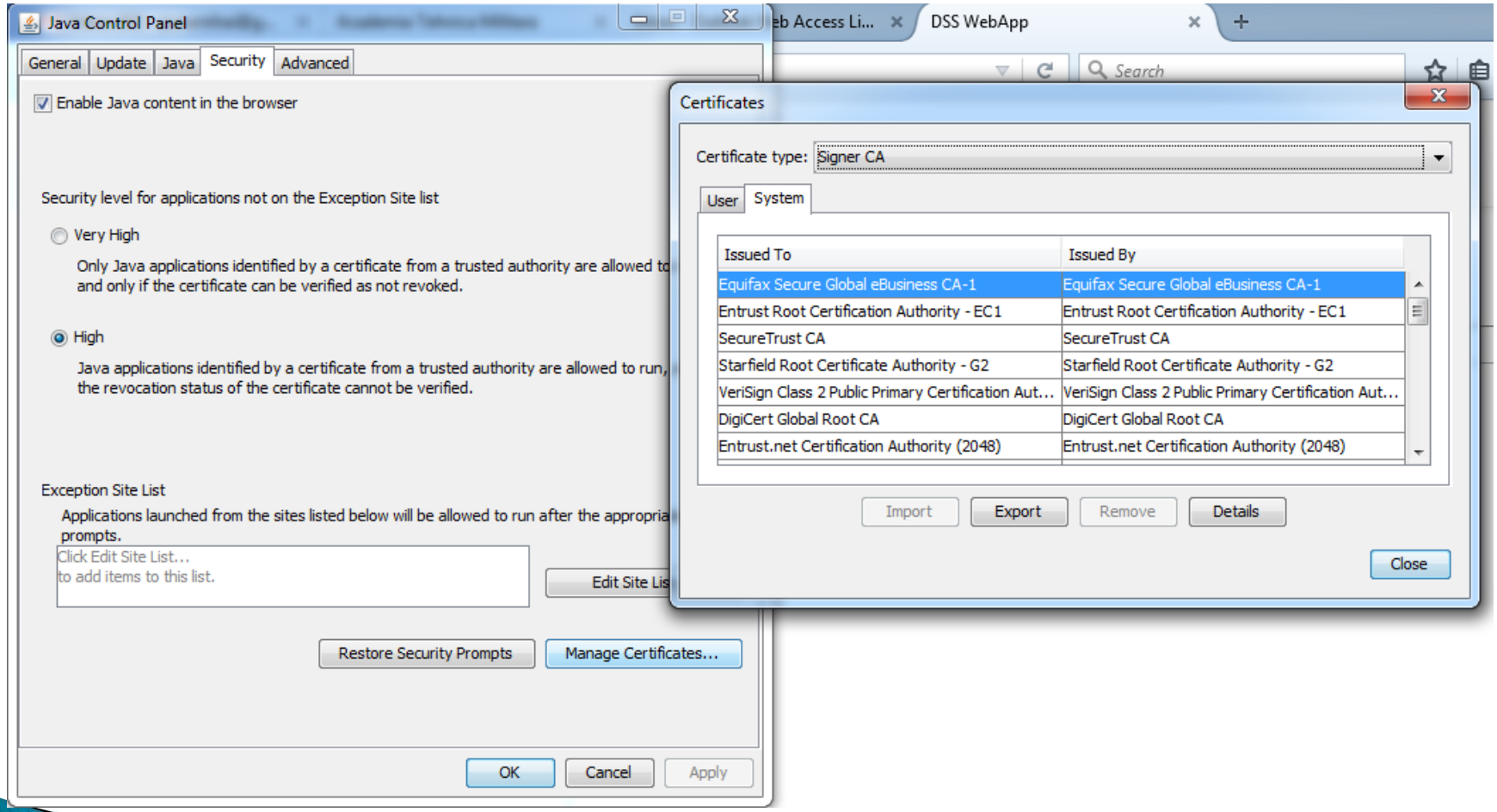
Code signing – Signed Java applets (obsolete)



Code signing – Signed Java applets (obsolete)



Code signing – Signed Java applets (obsolete)

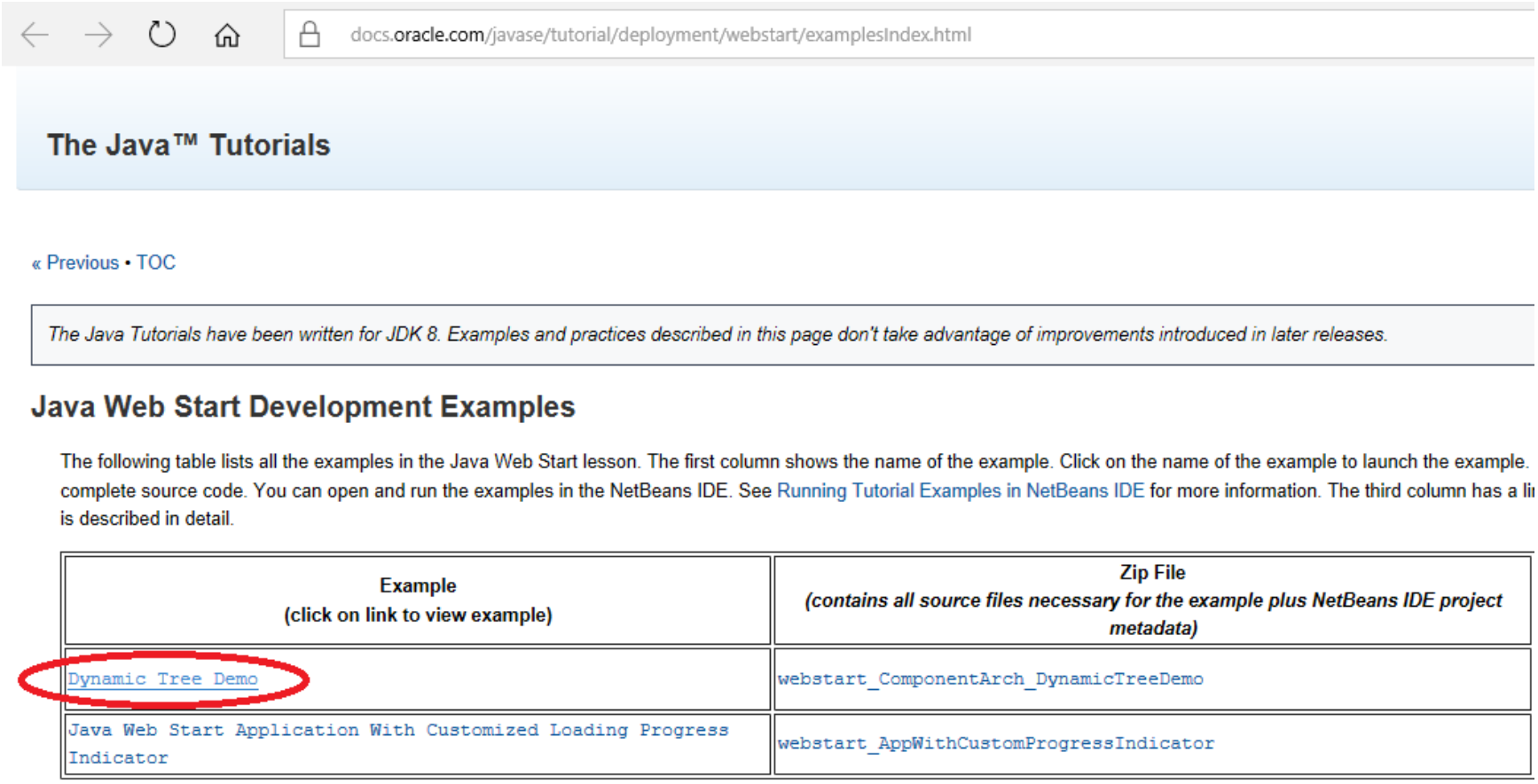


Code signing – Java Web Start

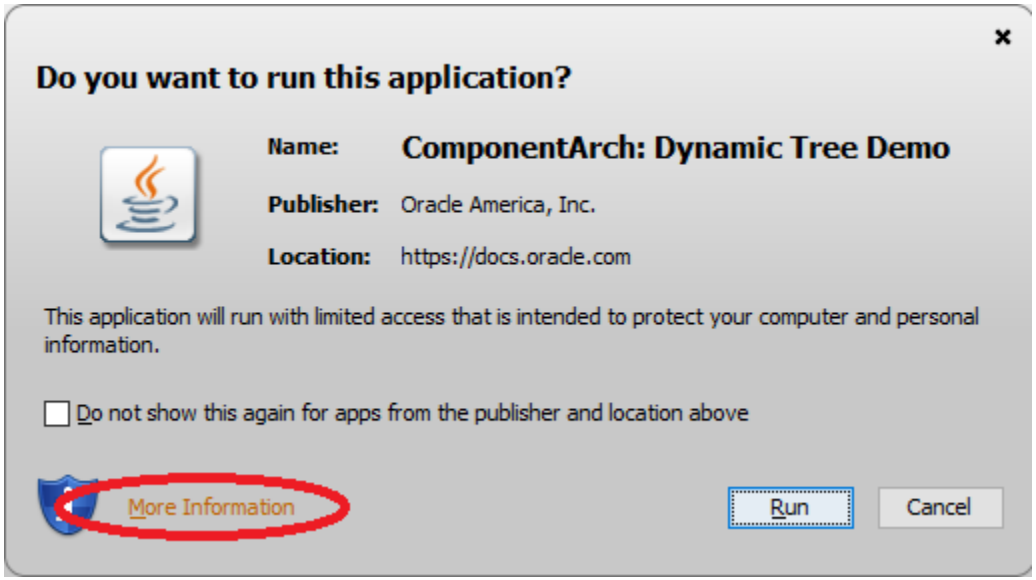
- Java Web Start sample applications
  - https://docs.oracle.com/javase/tutorial/deployment/webstart/examplesIndex.html

- **Java certificate stores**
  - Control Panel -> Java -> Security tab -> Manage certificates

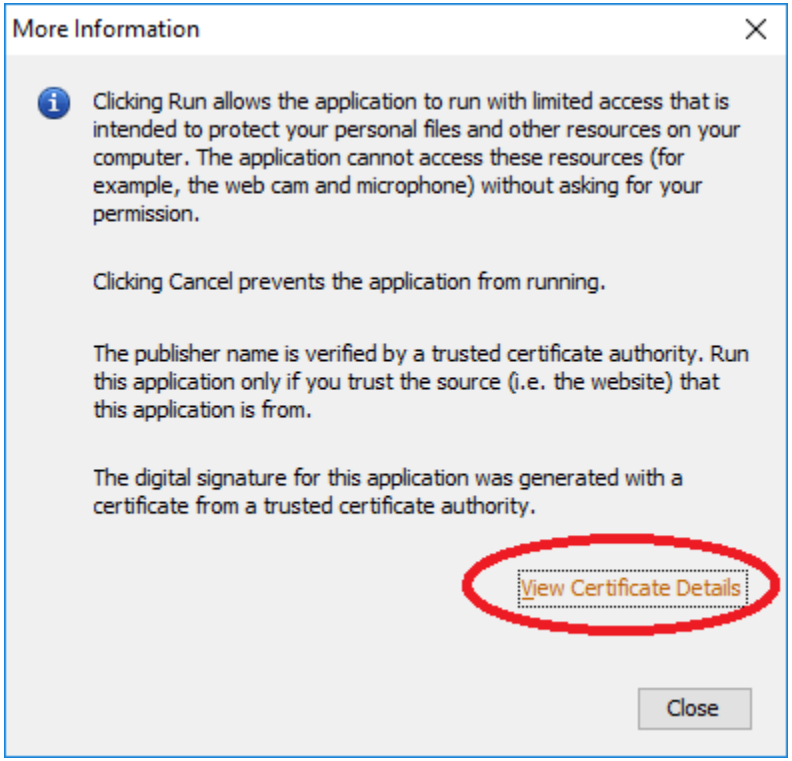
Code signing – Java Web Start



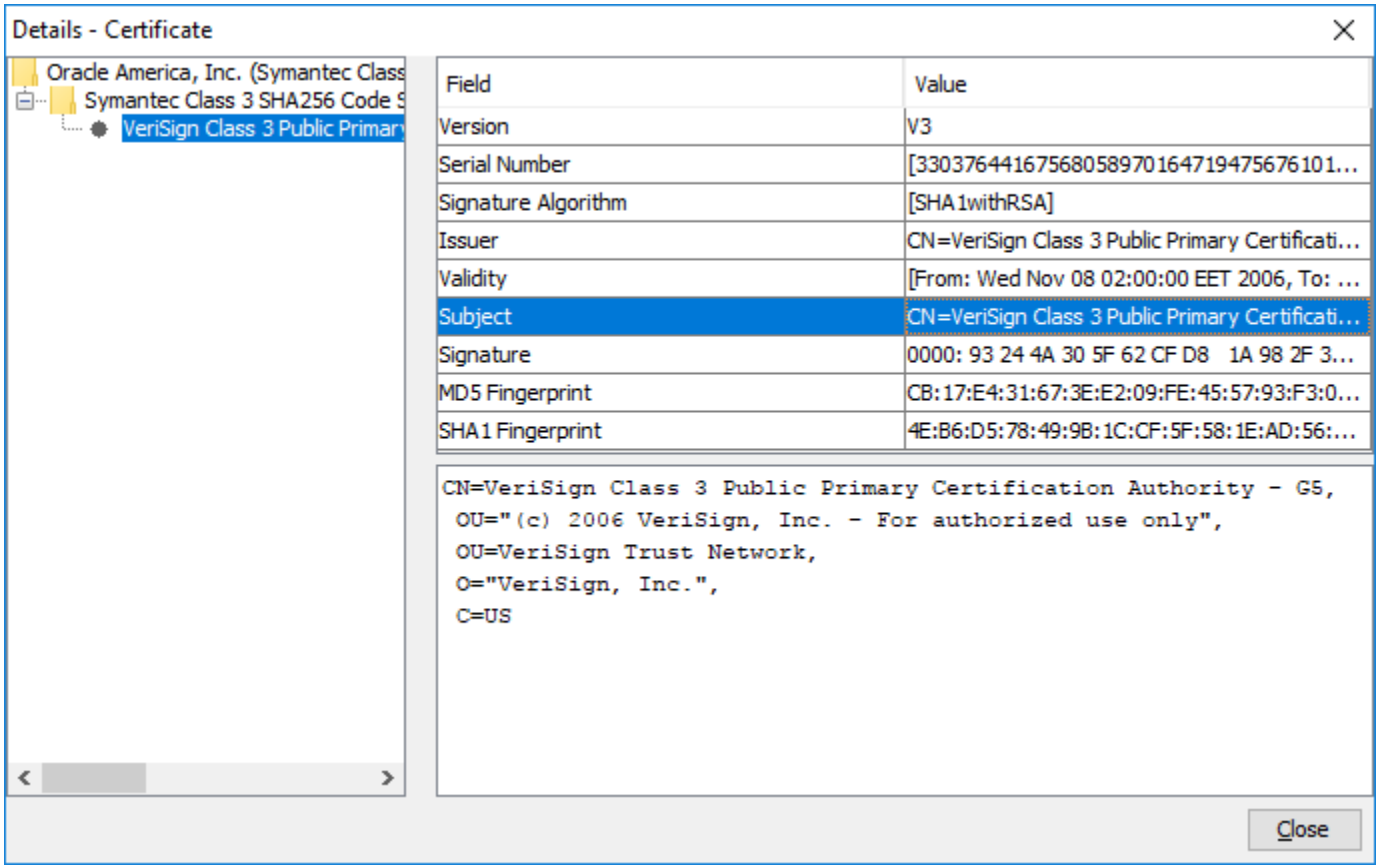
Code signing – Java Web Start



Code signing – Java Web Start

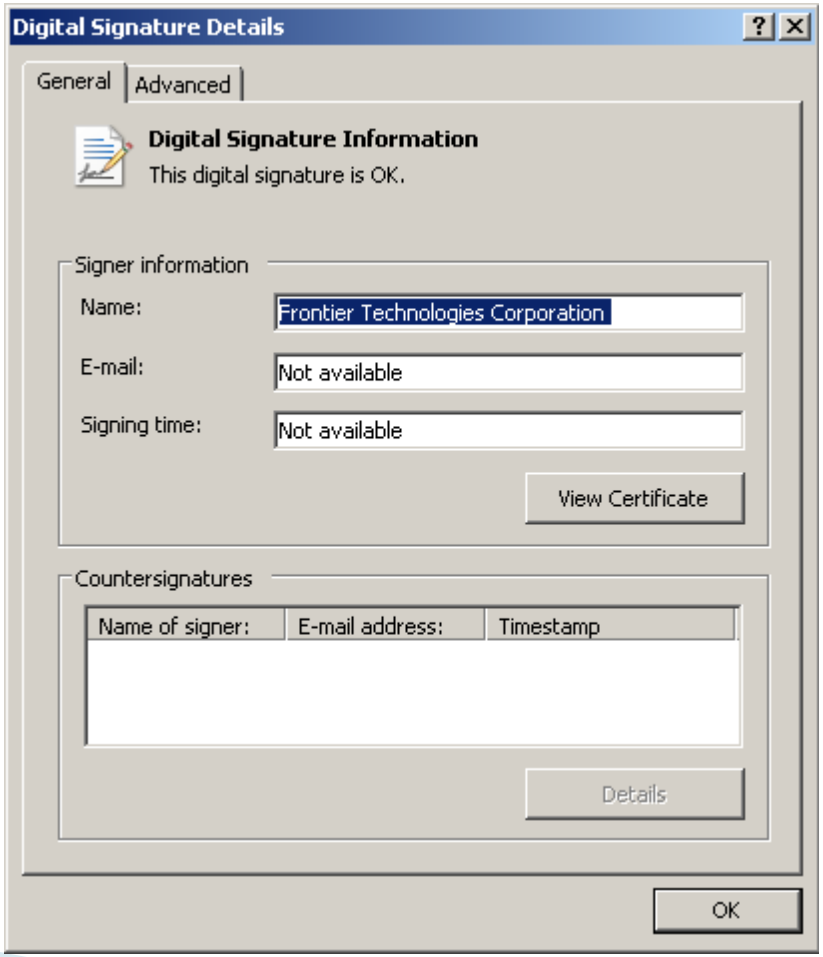


Code signing – Java Web Start

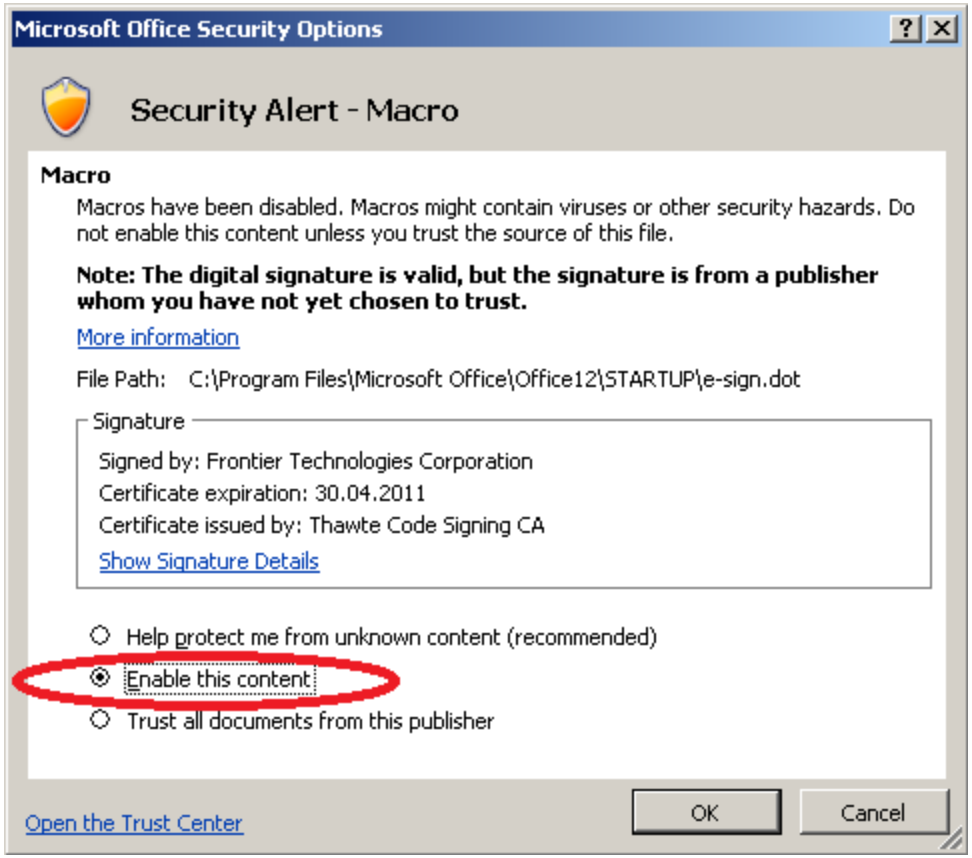


Code signing – Macros

Code signing – Macros



Code signing – Macros



Code signing – Executables

- User Account Control on Microsoft Windows 7 and greater

Cryptography libraries – High Level

- **Digital Signature Services**

https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/Start+using+Digital+Signature+Services+%28DSS%29+-+Releases+and+Bitbucket

- **Bouncy Castle**

https://www.bouncycastle.org/

- **SecureBlackbox**

http://www.secureblackbox.com/#

Cryptography libraries – Low Level

- **OpenSSL**

https://www.openssl.org/

- **MIRACL**

https://github.com/miracl/MIRACL
https://libraries.docs.miracl.com/miracl-explained/what-is-miracl

- **libecc**

https://github.com/ANSSI-FR/libecc

Microsoft Trusted Root Certificate Program

- https://technet.microsoft.com/en-us/library/cc751157.aspx

- Provides to clients a list of all the certificates of the certificate authorities that are trusted by Microsoft

Microsoft Trusted Root Certificate Program

- Trusted Root Certification Authorities logical certificate store
  - is populated with all these certificates from the moment that a Windows operating system is installed
- The list of the certificate authorities that Microsoft trusts is permanently updated

Microsoft Trusted Root Certificate Program

- The update of the contents of Trusted Root Certification Authorities (**automatic update**):
  - By Internet Explorer:
  - every time a user
    - access a HTTPS site,
    - or opens a S/MIME e-mail message,
    - or downloads a signed ActiveX control,
  - and the operating system, in its verifications, encounters a certificate that is not present in the Trusted store
    - it will verify online if this certificate is on the trusted CAs list of Microsoft
    - and if yes, it will be downloaded in the store

Microsoft Trusted Root Certificate Program

- For a CA to have its certificate included on the Microsoft trusted CAs list, it must:
  - Obtain a preliminary approval for the participation, which is granted based on some information requested by Microsoft
  - The CA must send a test certificate issued by its root certificate authority
  - The CA must then sign an agreement with Microsoft

Microsoft Trusted Root Certificate Program

- The root certificate of the CA must have
  - a minimum 2048 bit length RSA key
  - the validity period must be at least 8 years
  - but it should expire before 2030
- The certificates issued by the CA must
  - support the CRL Distribution Point extension
  - and the CRL distribution point must be publically accessible

Microsoft Trusted Root Certificate Program

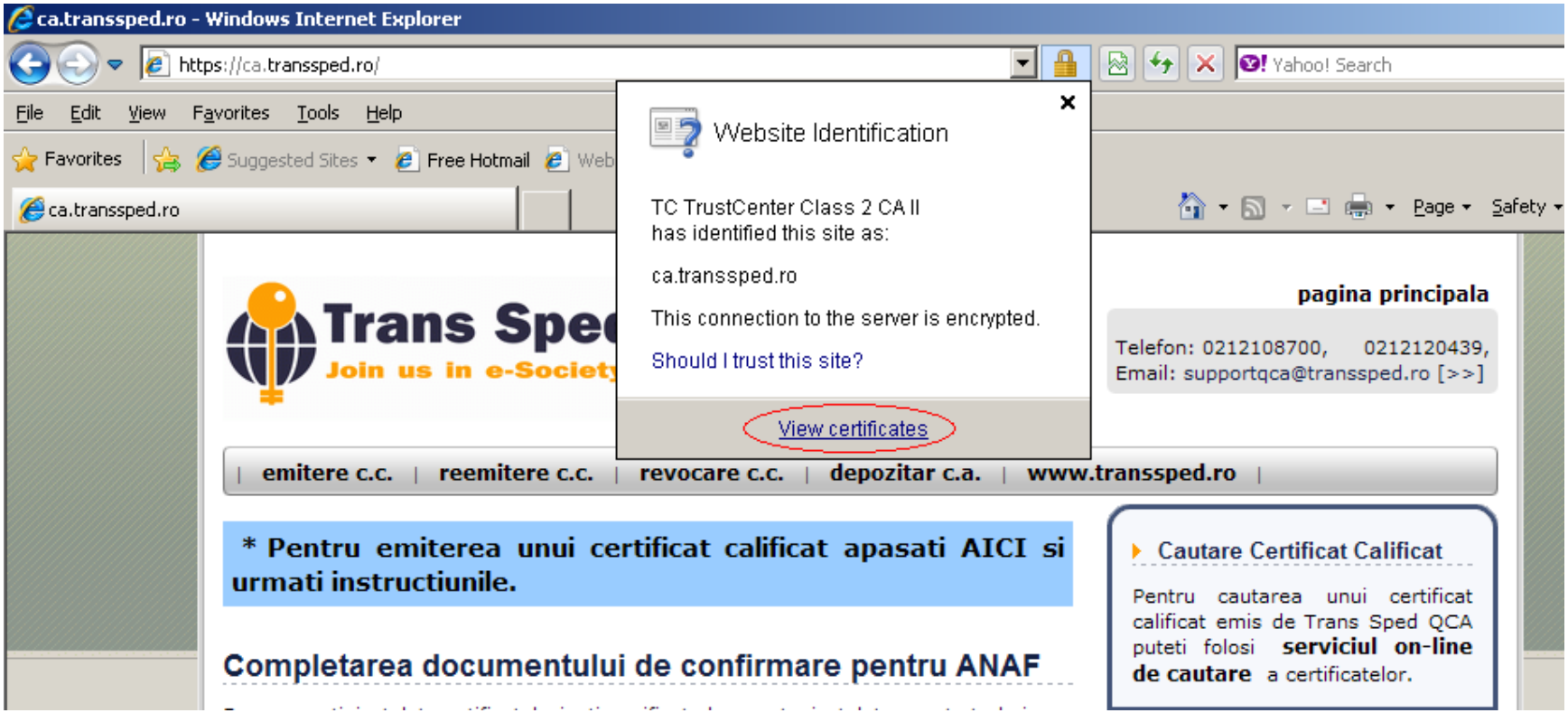
- The CA
  - must have a written certificate revocation policy
  - must be able to revoke any of the issued certificates
  - must be the subject of an annual security audit
  - and it must sent the results to Microsoft

Microsoft Trusted Root Certificate Program

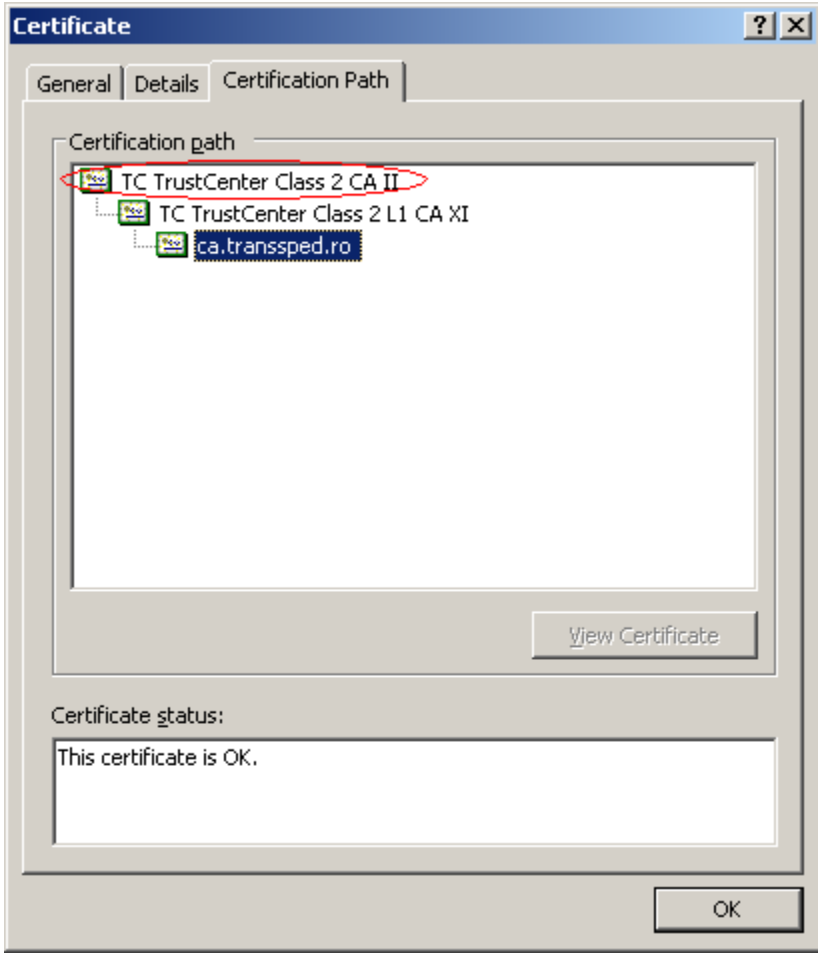
- Using Internet Explorer
  - access a HTTPS web site for which the certificate is issued by a certificate authority that has its root certificate on the Microsoft trusted CAs list
  - E.g.
  - https://ca.transsped.ro
- The page will be automatically opened, like a regular HTTP web page

Microsoft Trusted Root Certificate Program

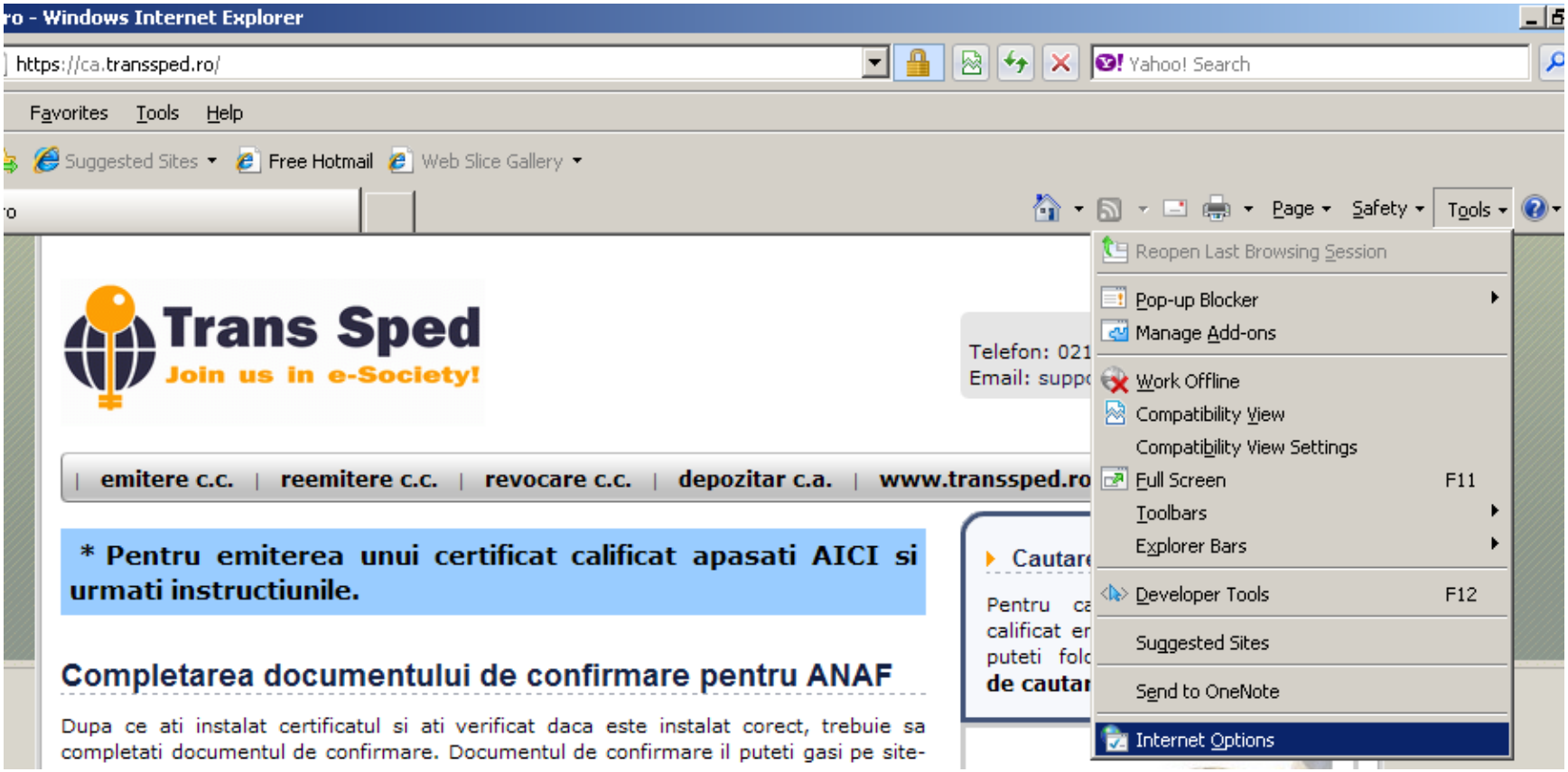
Microsoft Trusted Root Certificate Program



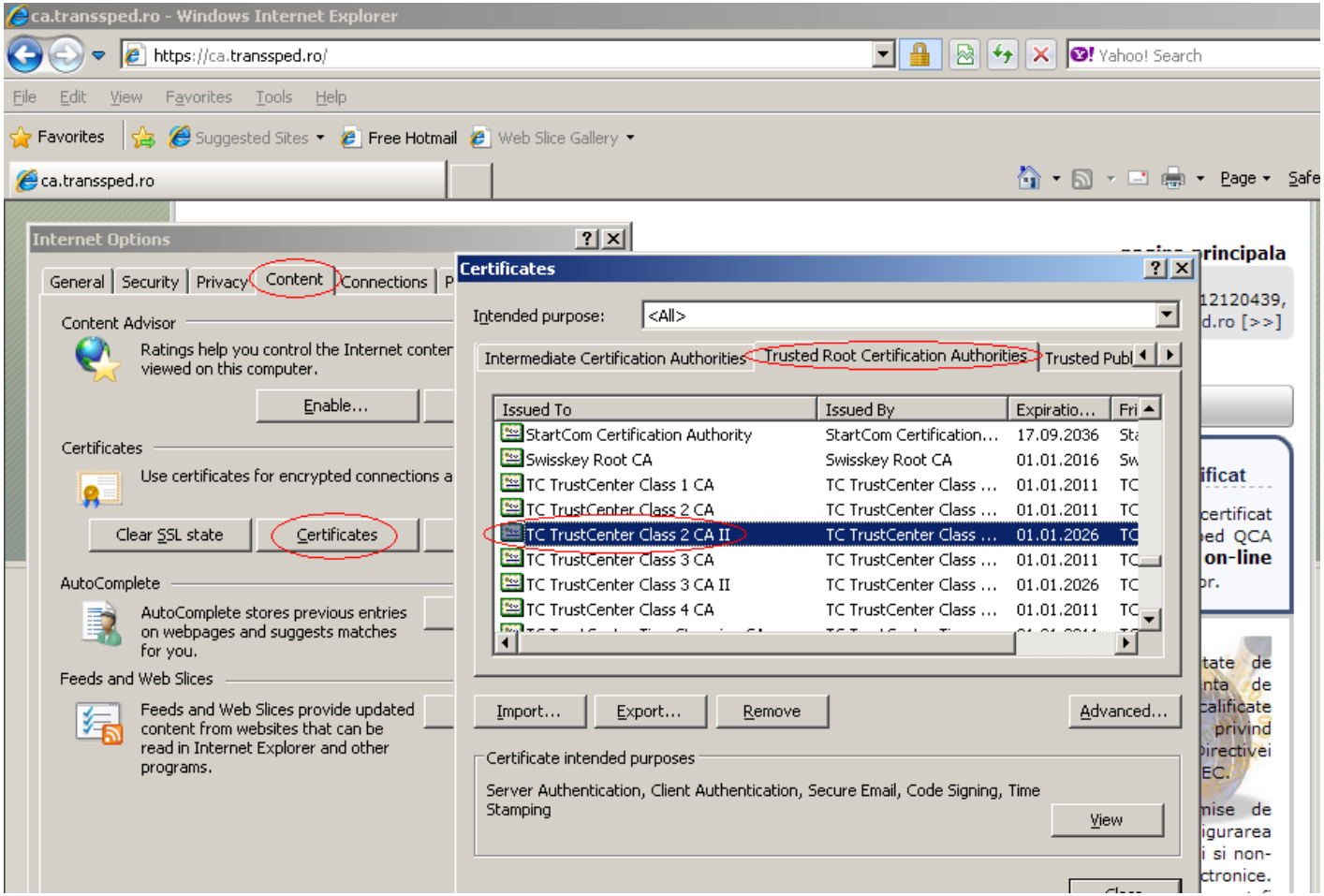
Microsoft Trusted Root Certificate Program



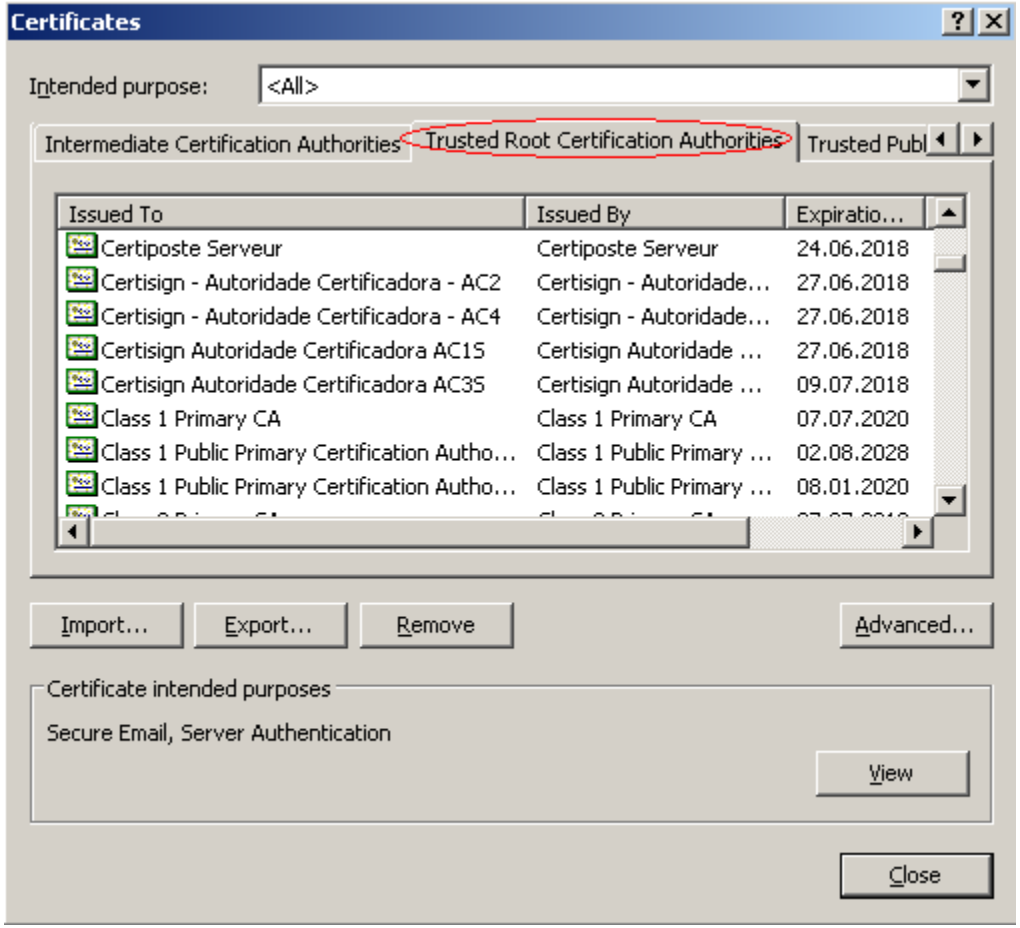
Microsoft Trusted Root Certificate Program



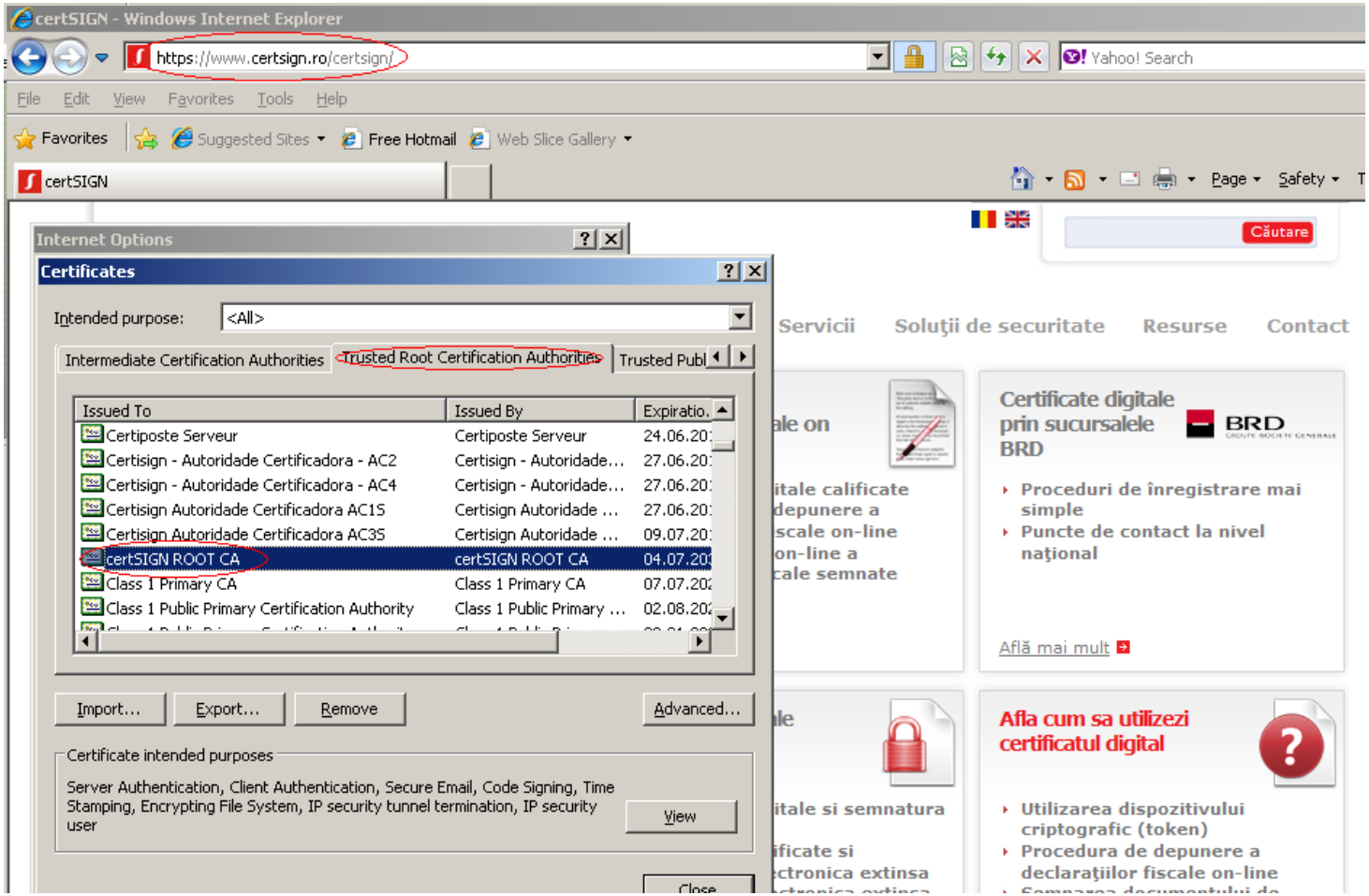
Microsoft Trusted Root Certificate Program



Microsoft Trusted Root Certificate Program



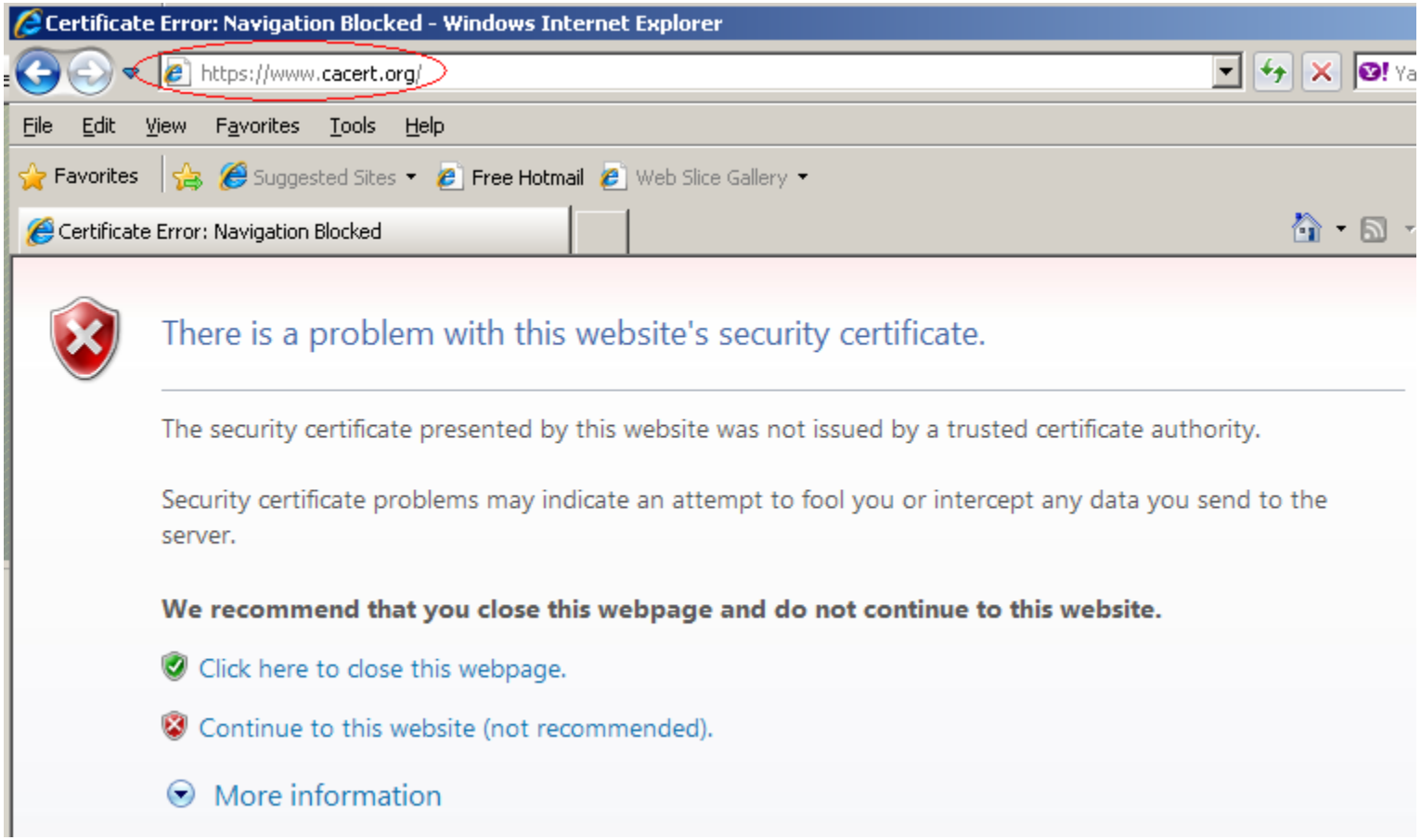
Microsoft Trusted Root Certificate Program



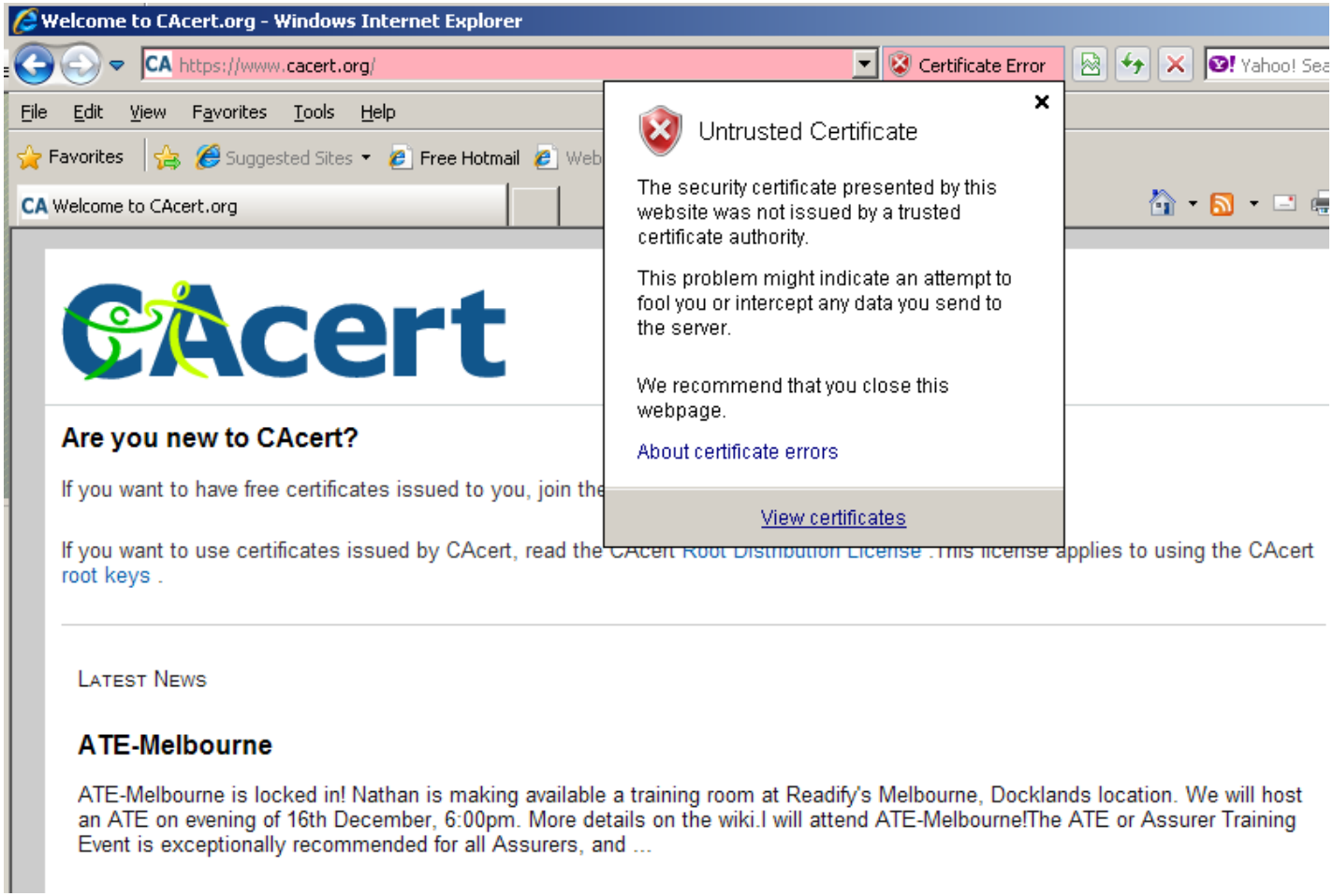
Microsoft Trusted Root Certificate Program

- Using Internet Explorer
  - access a HTTPS web site for which the certificate is NOT issued by a certificate authority that has its root certificate on the Microsoft trusted CAs list
  - E.g.
  - https://www.cacert.org
- The page will NOT be automatically opened

Microsoft Trusted Root Certificate Program

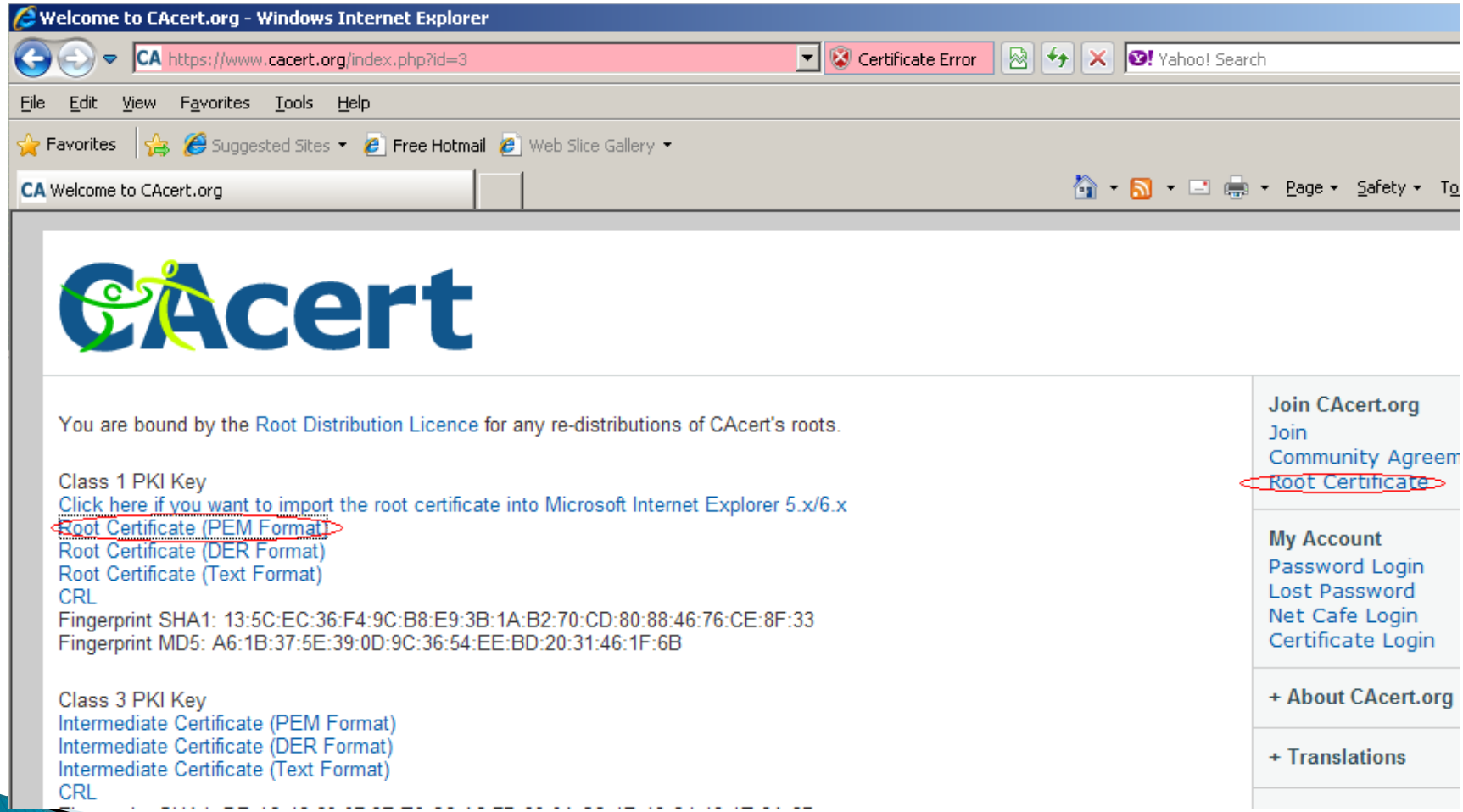


Microsoft Trusted Root Certificate Program

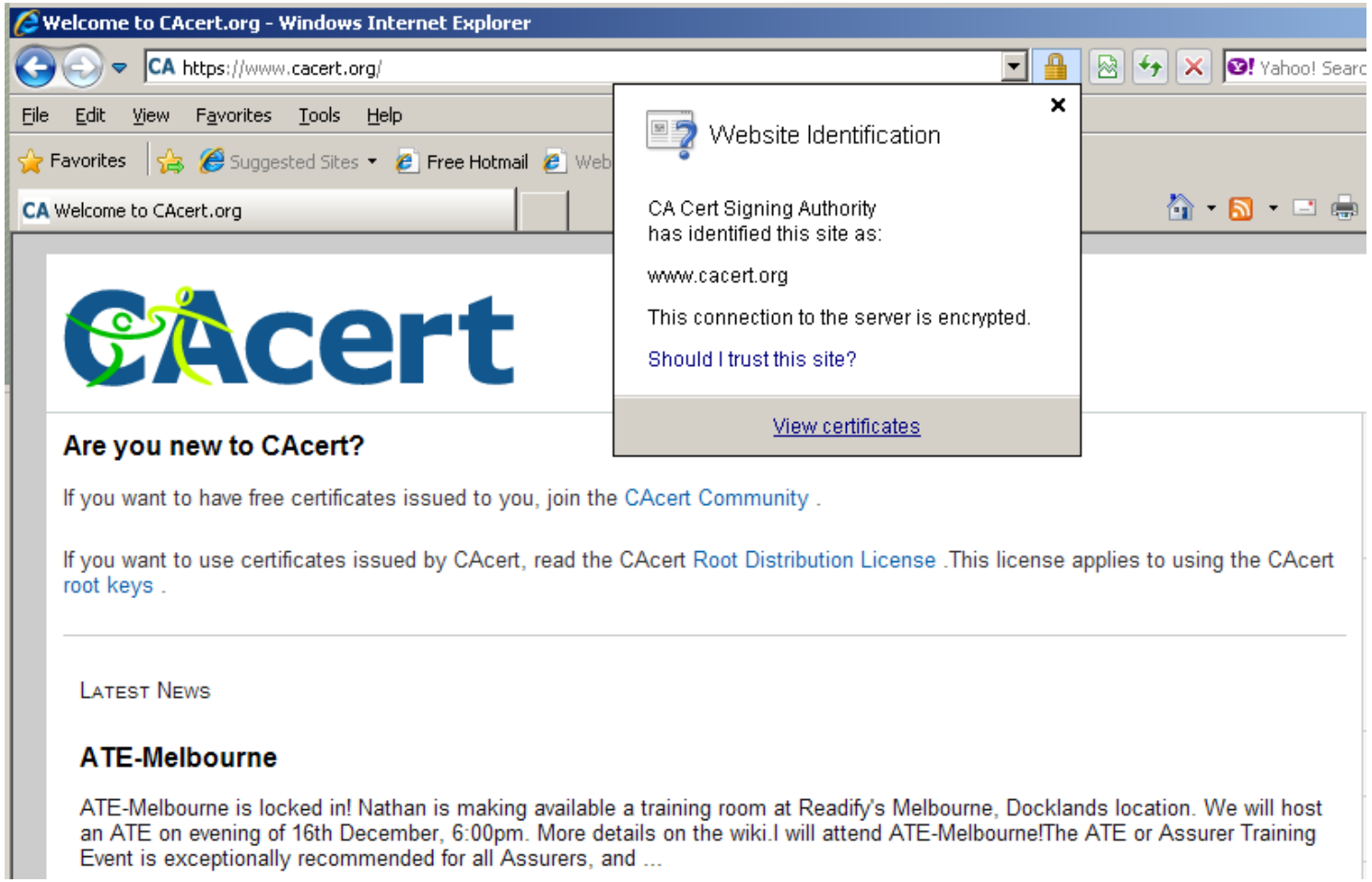


Microsoft Trusted Root Certificate Program

Microsoft Trusted Root Certificate Program



Microsoft Trusted Root Certificate Program



Mozilla CA Certificate Policy

- https://www.mozilla.org/en-US/about/governance/policies/security-group/certs/policy/

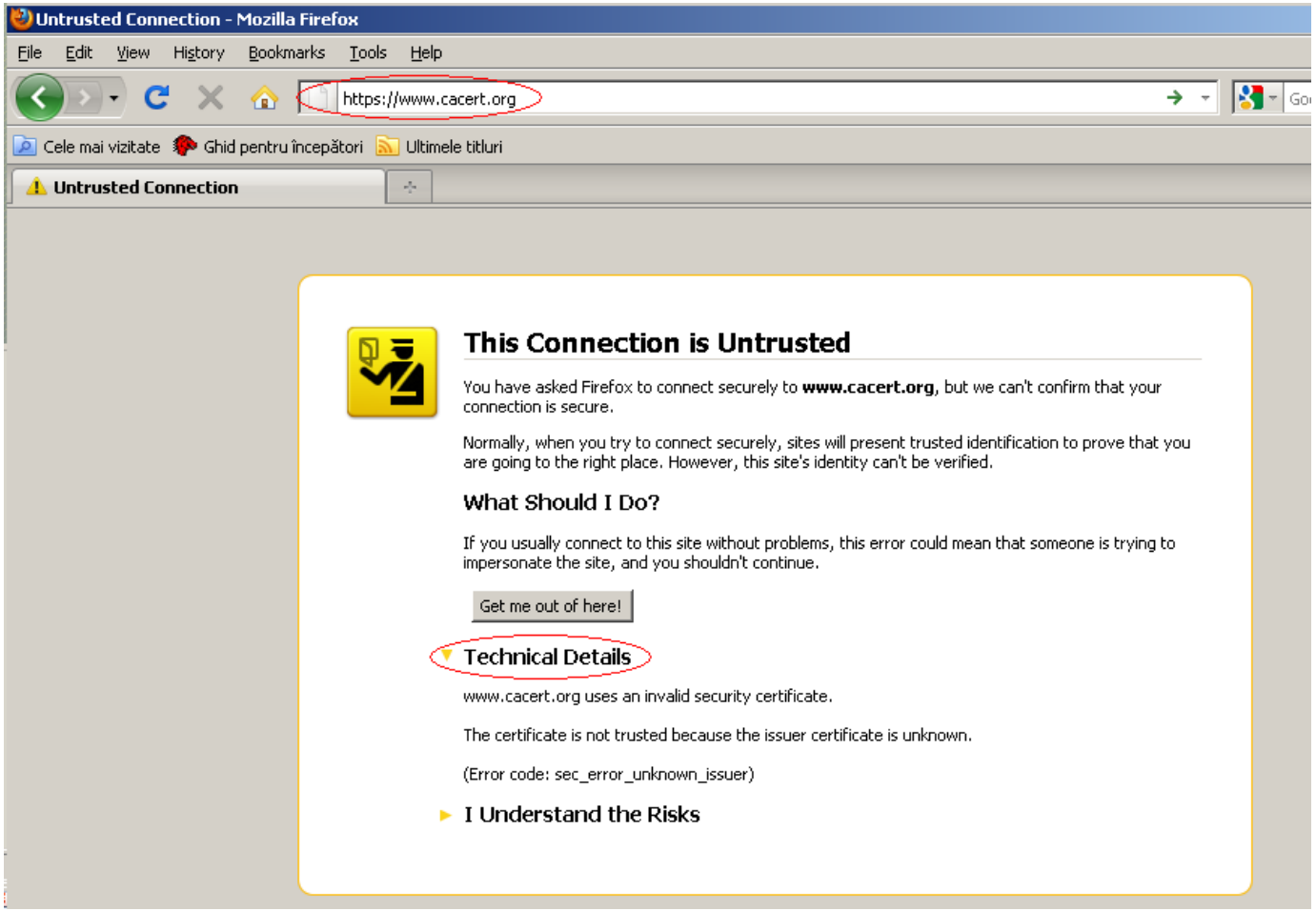
- When downloading and installing
  - Firefox,
  - Thunderbird
  - or other Mozilla applications
- their certificate store it is populated will a list of certificates

- All these certificates are marked as trusted so Firefox and Thunderbird, for example, will use them for digital signatures validation without displaying any message to the user

Mozilla CA Certificate Policy

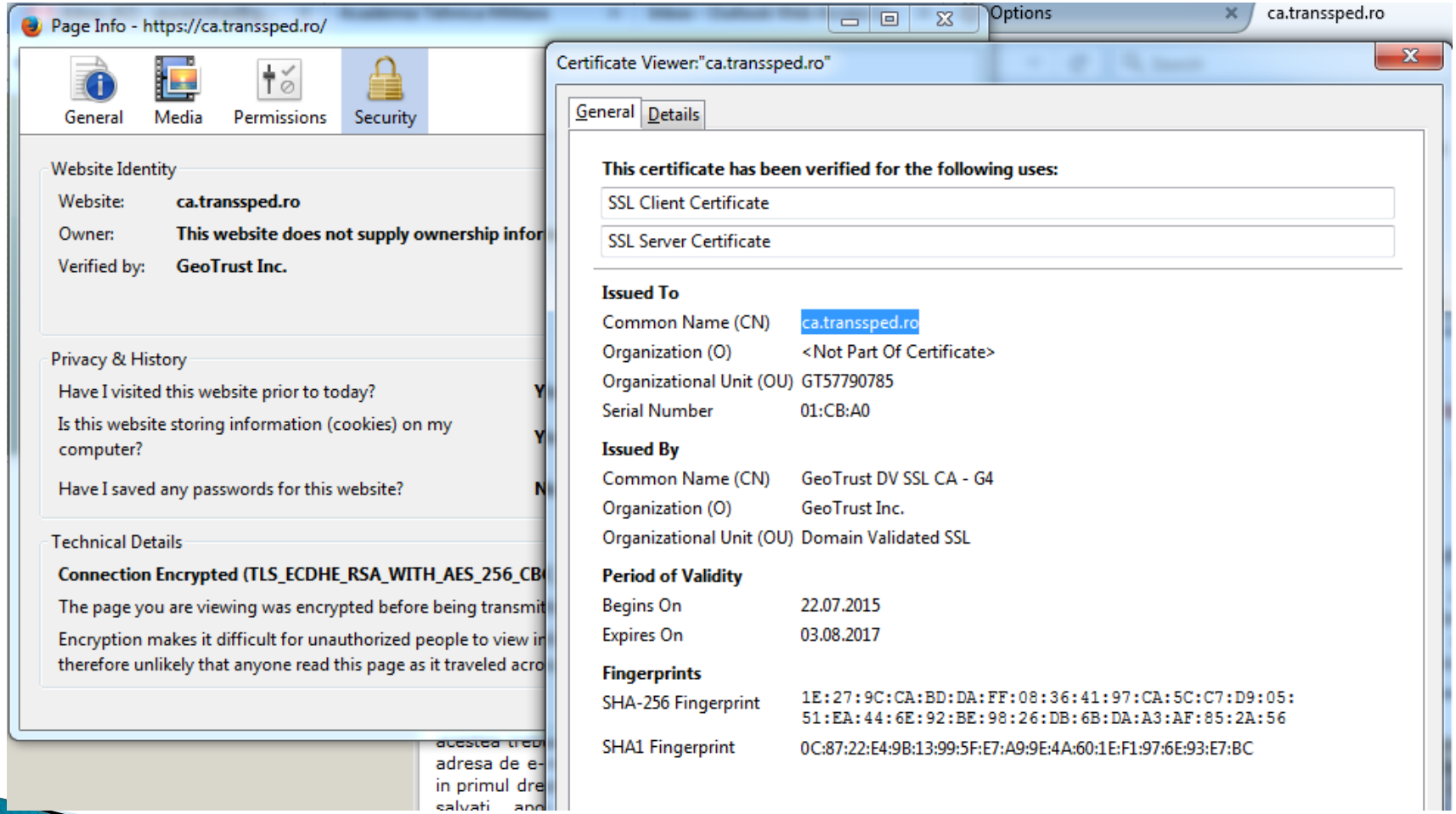
- As for the Microsoft Program, a CA will be included on this list after a request and a special verification process
- The major differences from the Microsoft verification process are that no fees are requested, and the verification is public

Mozilla CA Certificate Policy

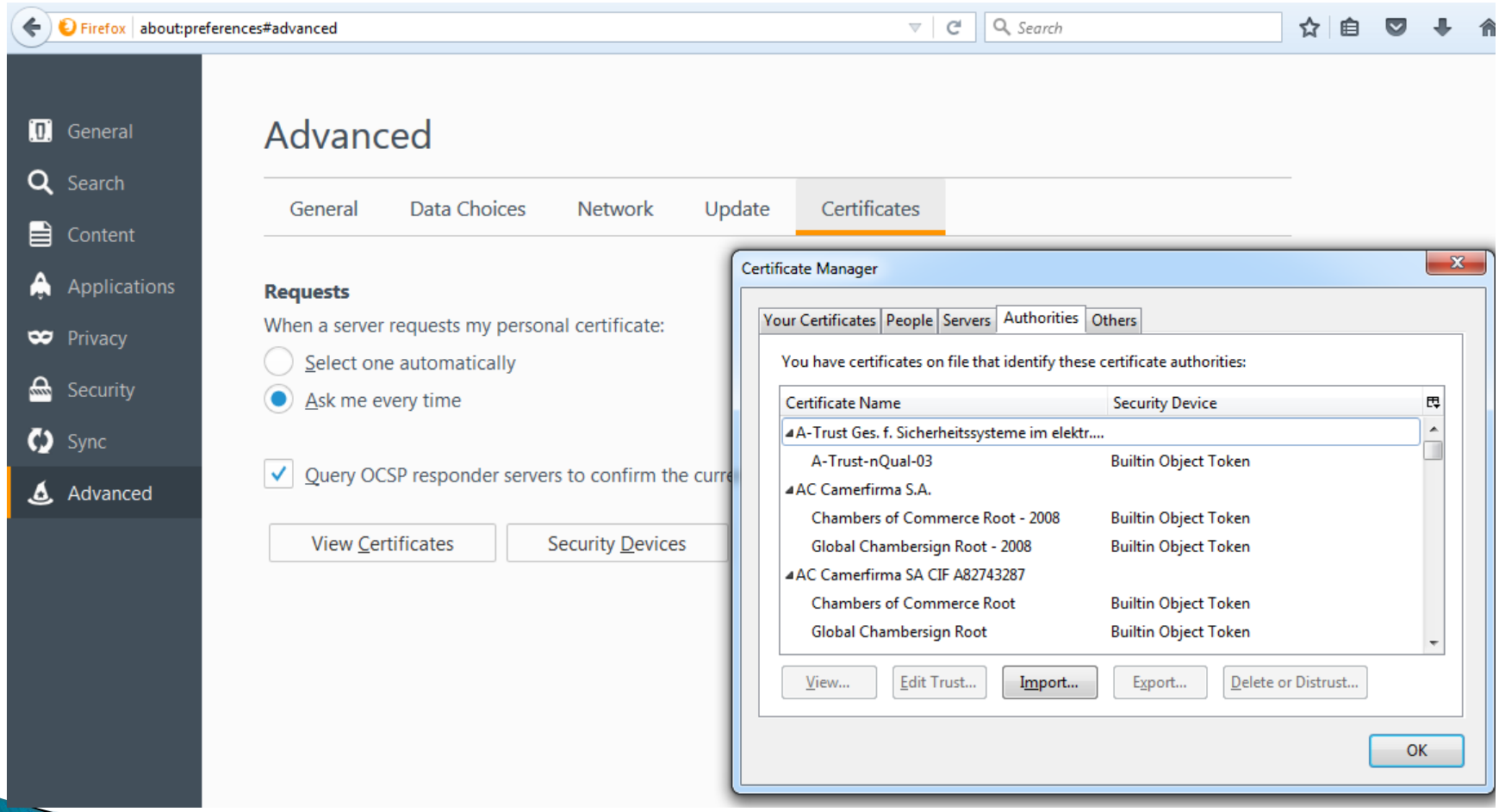


Mozilla CA Certificate Policy

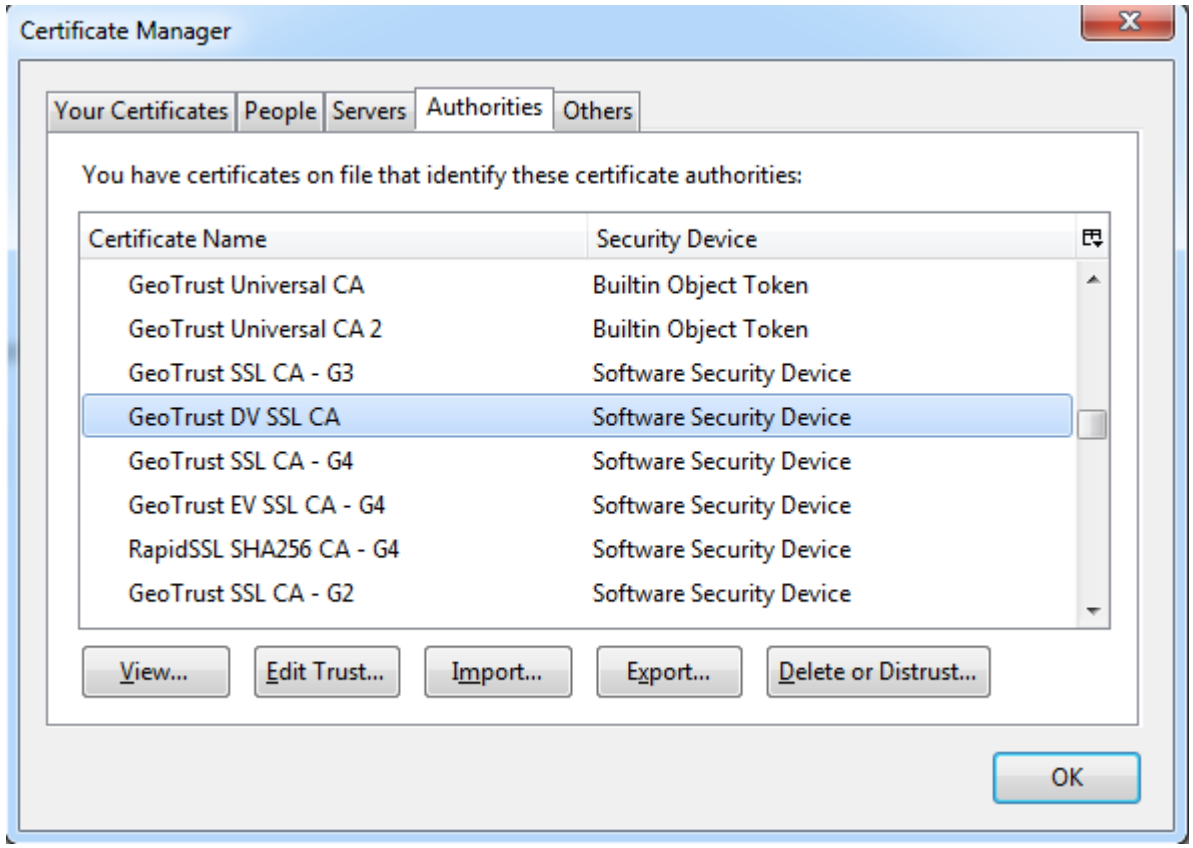
Mozilla CA Certificate Policy



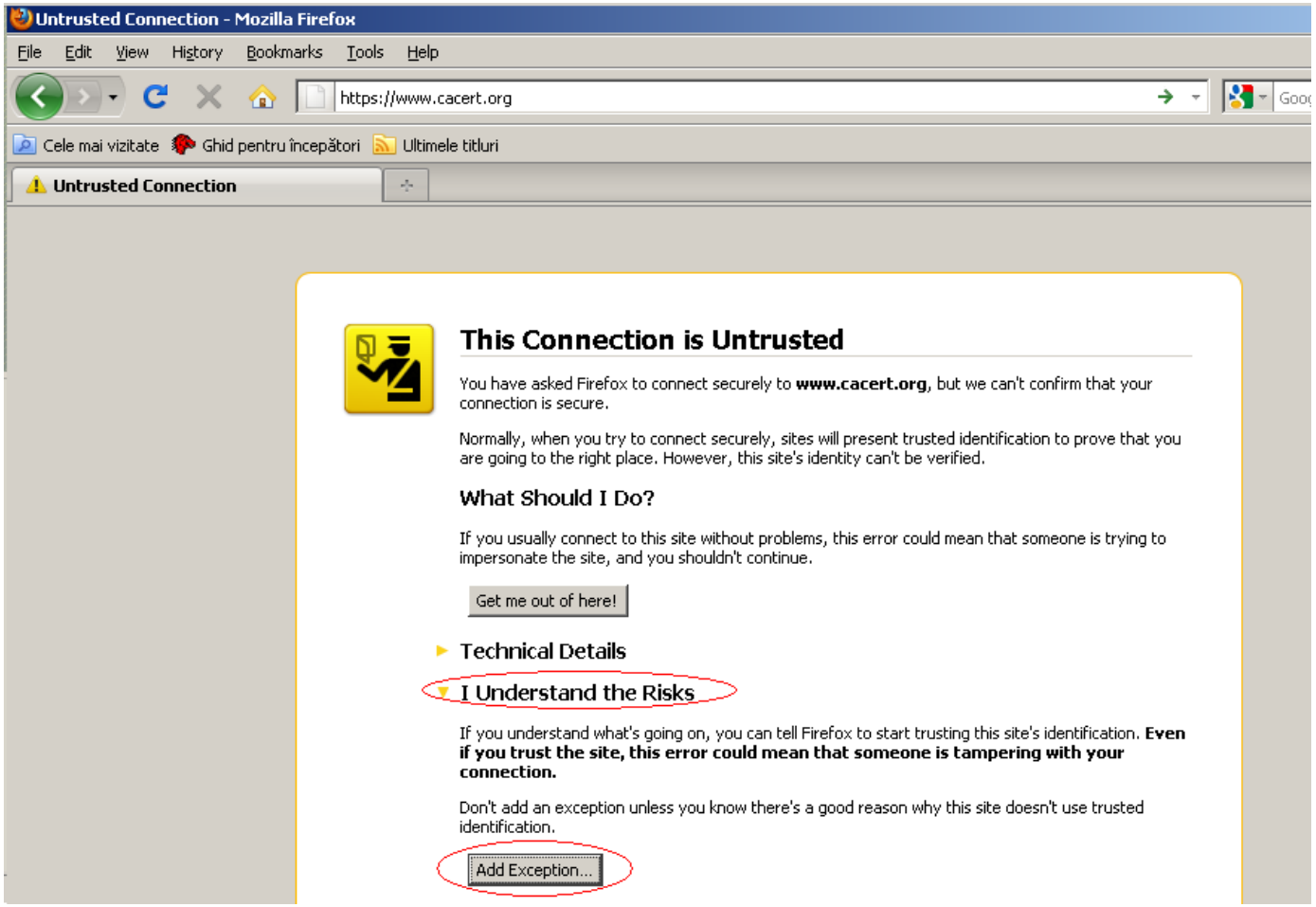
Mozilla CA Certificate Policy



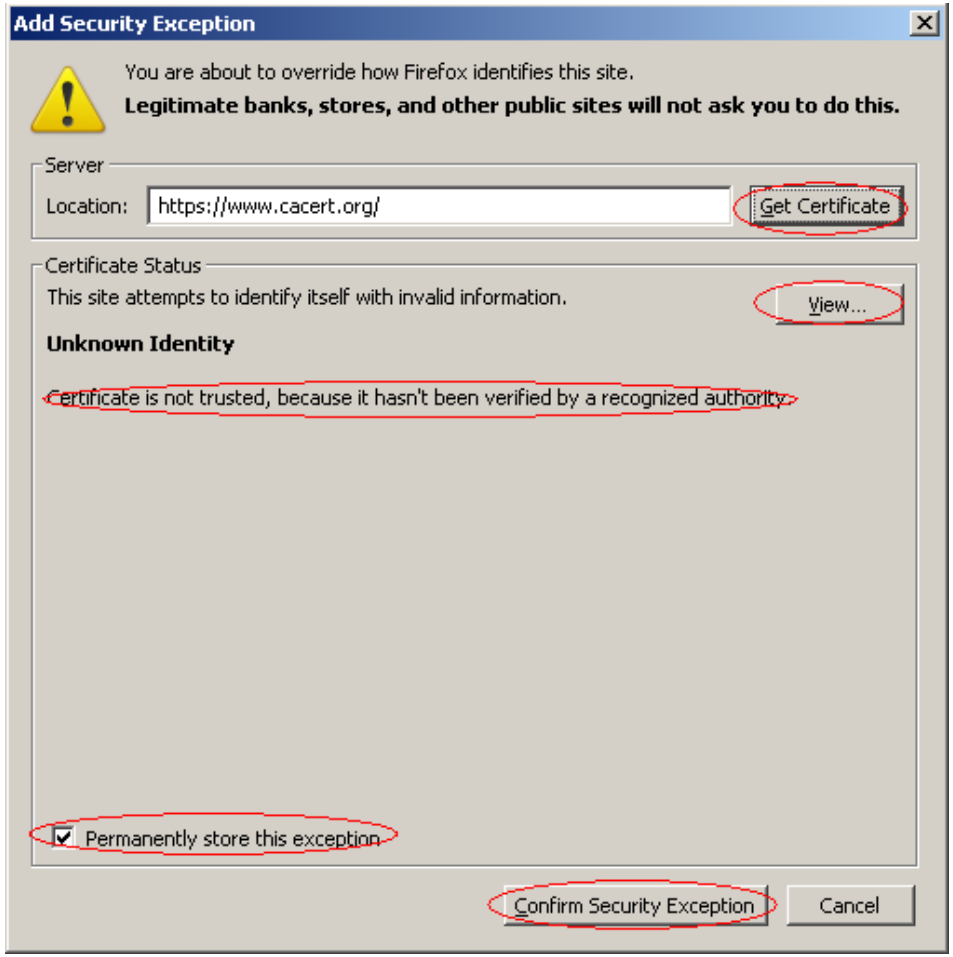
Mozilla CA Certificate Policy



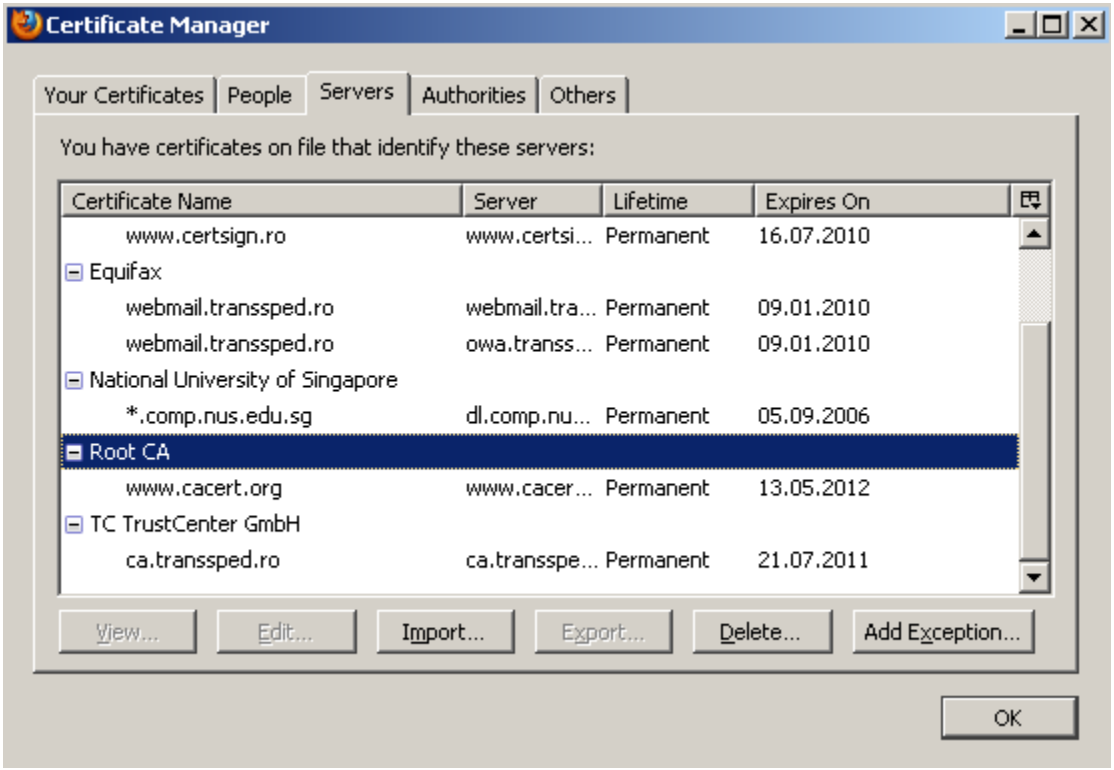
Mozilla CA Certificate Policy



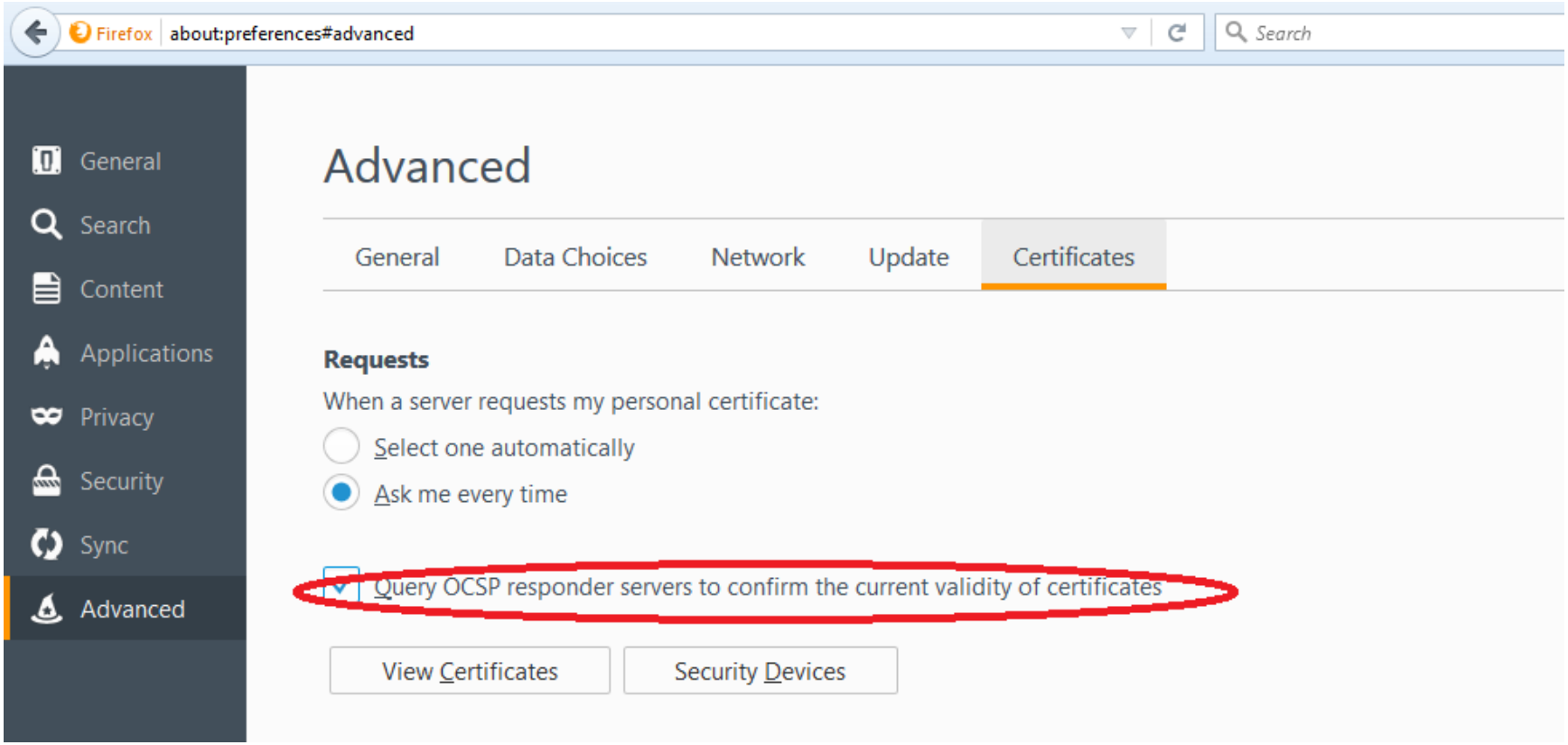
Mozilla CA Certificate Policy



Mozilla CA Certificate Policy



Mozilla CA Certificate Policy



Legislation

- Legea nr. 455/2001 privind **semnătura electronică**
  - http://www.transsped.ro/content/L_2001_07_18_455.pdf
- HG 1259/2001 privind Normele tehnice şi metodologice pentru aplicarea Legii nr. 455/2001 privind semnatura electronicã
  - http://www.transsped.ro/content/HG_2001_12_13_1259.pdf
- eIDAS Regulation no. 910/2014
  - https://eur-lex.europa.eu/legal-content/RO/TXT/PDF/?uri=CELEX:32014R0910&from=EN

Legislation

- Legea nr. 451 din 1 noiembrie 2004 privind **marca temporală**
  - https://www.certsign.ro/ro/documentatie/legislatie/legea-451-2004-privind-marca-temporala
- Legea nr. 589 din 15 decembrie 2004 privind regimul juridic al **activităţii electronice notariale**
  - https://www.certsign.ro/ro/documentatie/legislatie/legea-589-2004-activitate-electronica-notariala
- Normele tehnice şi metodologice pentru aplicarea legii 589/2004 privind regimul juridic al activităţii electronice notariale
  - https://www.certsign.ro/ro/documentatie/legislatie/norme-aplicare-lege-activitate-electronica-notariala

Legislation

- Legea 135/2007 a **arhivării electronice**
  - https://www.certsign.ro/ro/documentatie/legislatie/legea-135-2007-arhivare-electronica

- Ordin MEF 858/2008 privind depunerea **declaraţiilor fiscale prin mijloace electronice** de transmitere la distanţă
  - https://www.certsign.ro/ro/documentatie/legislatie/ordin-declaratii-fiscale-online

Romanian CAs

- AlfaSign
- https://www.alfasign.ro/

- certSIGN
- https://www.certsign.ro/

- CERT DIGITAL
- https://www.certdigital.ro/

- DigiSign
- https://www.digisign.ro/

- Trans Sped
- http://www.transsped.ro/

References

- The Value of Certificate Revocation Lists (CRLs) in a PKI
  - https://blogs.technet.microsoft.com/staysafe/2012/02/09/the-value-of-certificate-revocation-lists-crls-in-a-pki/
- Formats of a digital signature
  - http://developers-club.com/posts/191866/
- Introduction to XML Digital Signatures
  - https://flylib.com/books/en/2.329.1.75/1/

References

- Certificate Trust
  - https://piv.idmanagement.gov/pivcertchains/
- Windows XP: Certificate Status and Revocation Checking
  - https://social.technet.microsoft.com/wiki/contents/articles/4954.windows-xp-certificate-status-and-revocation-checking.aspx
- Digital Signatures in a PDF
  - https://www.adobe.com/devnet-docs/acrobatetk/tools/DigSig/Acrobat_DigitalSignatures_in_PDF.pdf