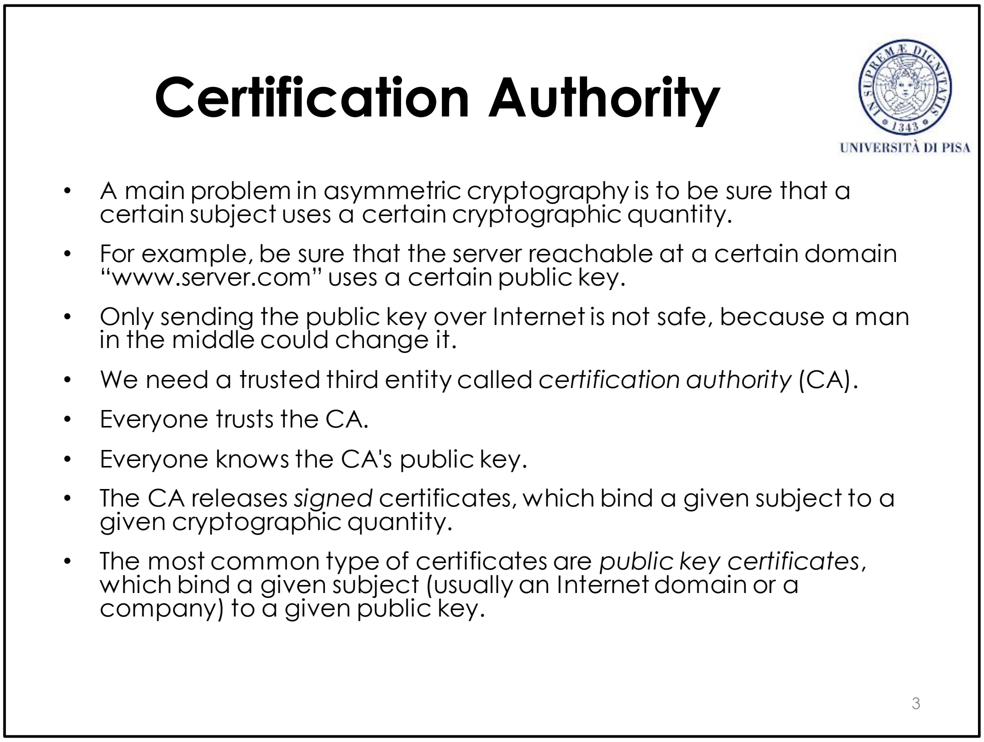
Setup Simple Authority

Michele La Manna
Dept. of Information Engineering
University of Pisa
michele.lamanna@phd.unipi.it

Version: 2022-05-11

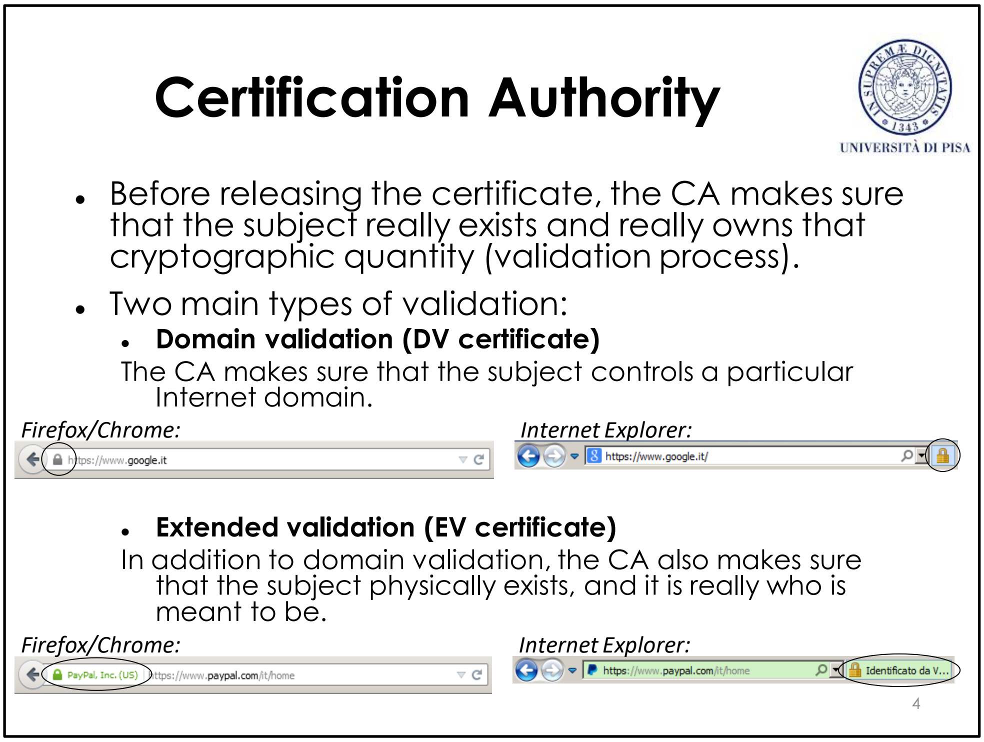
SIMPLE AUTHORITHY

CERTIFICATE GENERATION



Certification Authority

- A main problem in asymmetric cryptography is to be sure that a certain subject uses a certain cryptographic quantity.
- For example, be sure that the server reachable at a certain domain “www.server.com” uses a certain public key.
- Only sending the public key over Internet is not safe, because a man in the middle could change it.
- We need a trusted third entity called *certification authority* (CA).
- Everyone trusts the CA.
- Everyone knows the CA's public key.
- The CA releases *signed* certificates, which bind a given subject to a given cryptographic quantity.
- The most common type of certificates are *public key certificates*, which bind a given subject (usually an Internet domain or a company) to a given public key.



Certification Authority

UNIVERSITÀ DI PISA

- Before releasing the certificate, the CA makes sure that the subject really exists and really owns that cryptographic quantity (validation process).
- Two main types of validation:
  - **Domain validation (DV certificate)**

    The CA makes sure that the subject controls a particular Internet domain.

Firefox/Chrome:

Internet Explorer:

  - **Extended validation (EV certificate)**

    In addition to domain validation, the CA also makes sure that the subject physically exists, and it is really who is meant to be.

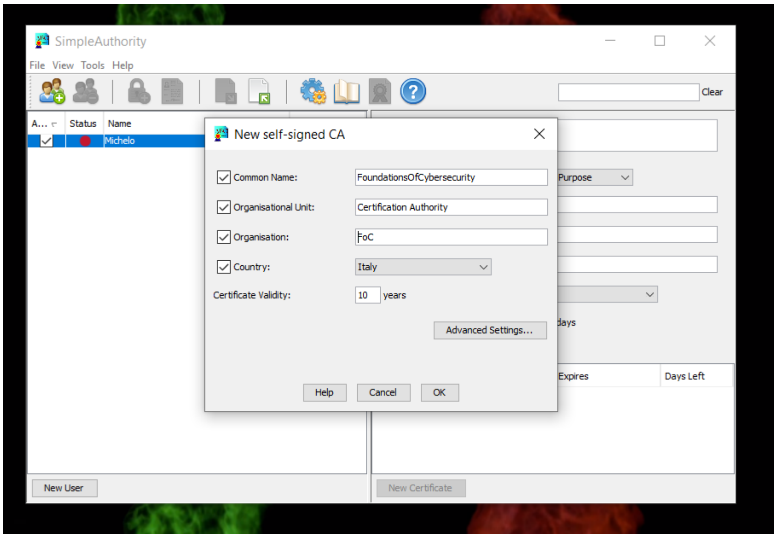
Firefox/Chrome:

Internet Explorer:

There are several types of certificates: the most common ones are certificates with *domain validation* (DV certificates) and certificates with *extended validation* (EV certificates). With domain validation, the CA only assures that the requesting subject owns a particular Internet domain. Usually it is done by sending a challenge email to an email server running on that domain. The subject proves to own the domain by responding to the email.
With extended validation, the CA assures the physical existence and the identity of the requesting subject. In particular, three checks must be performed: (i) legal existence; (ii) physical existence; (iii) operational existence (that is, the subject is a still-operating company). The extended validation requires a face-to-face identification with a CA's employee, a notary, or a lawyer. Not all CA's are permitted to issue EV certificates. Web browsers usually signals the presence of an EV certificate with green colors (a green lock in Firefox/Chrome, a green address bar in Internet Explorer).

SimpleAuthority CA

UNIVERSITÀ DI PISA



New self-signed CA

☑ Common Name: FoundationsOfCybersecurity

☑ Organisational Unit: Certification Authority

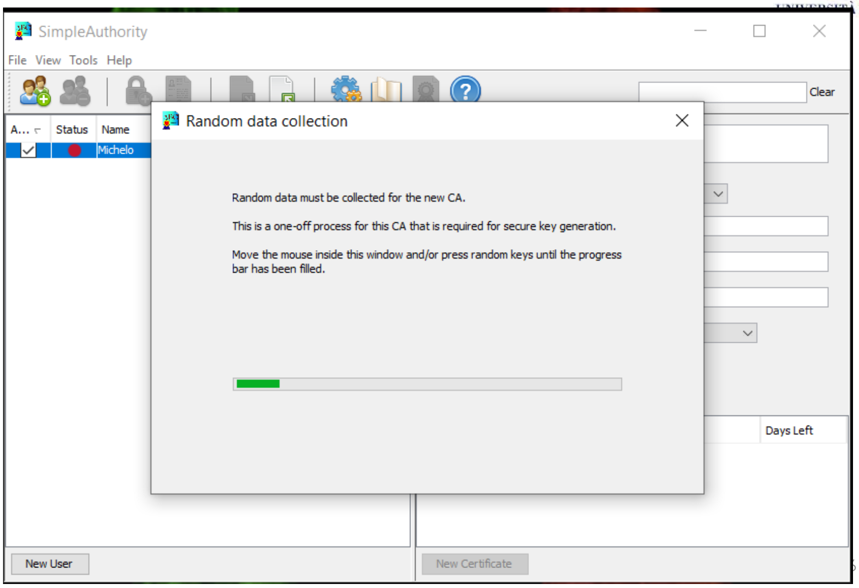
☑ Organisation: FoC

☑ Country: Italy

Certificate Validity: 10 years

SimpleAuthority is a GUI-based Certification Authority software. It is java-based and multi-platform, it can run under Windows, Linux and Mac OS. With the free license, a maximum of 4 users are possible, and only 2048-bit RSA signatures are supported. On Linux Ubuntu machines of this class, it can be run with the "sh /opt/simpleauthority/sauth" commandline. The instructions included in these slides refer to SimpleAuthority version 4.10 on Windows 10.
At the first execution of the program, Simple authority will ask you for some infos in order to create a Certification Authority (See the slides).
You can edit the fields as you wish.
Remember them, however, because they will be useful in order to verify the certificates issued by the generated CA.

SimpleAuthority CA



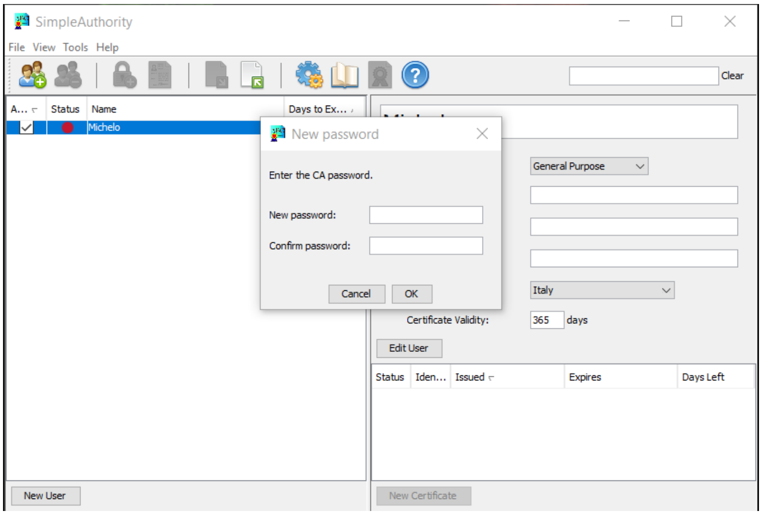
Random data must be collected for the new CA.

This is a one-off process for this CA that is required for secure key generation.

Move the mouse inside this window and/or press random keys until the progress bar has been filled.

Follow the instructions and have some fun time!
The randomness of your action will serve the purpose to generate an RSA keypair with 2048 bits.

SimpleAuthority CA



Michelo

Enter the CA password.

New password:

Confirm password:

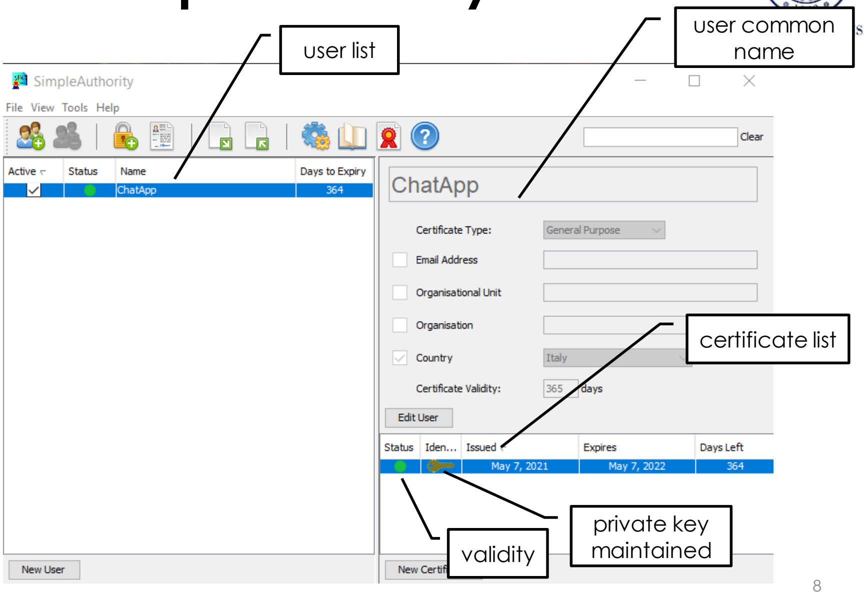
General Purpose

Italy

Certificate Validity: 365 days

Insert and REMEMBER the password to manage the CA!
Mine will be, OF COURSE, “password”

SimpleAuthority CA



Active	Status	Name	Days to Expiry
✓	●	ChatApp	364

ChatApp

Certificate Type: General Purpose

□ Email Address

□ Organisational Unit

□ Organisation

☑ Country: Italy

Certificate Validity: 365 days

Status	Iden...	Issued	Expires	Days Left
●		May 7, 2021	May 7, 2022	364

user list

user common name

certificate list

validity

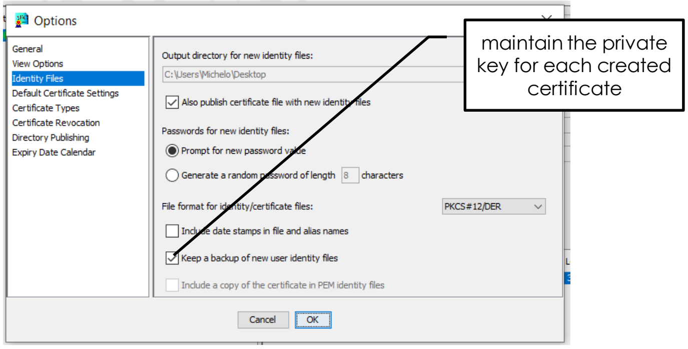
private key maintained

In the main window, the right panel shows the list of created users, which are the subjects for which the CA can create certificates (e.g., a web server). The left panel shows the details of the selected user (e.g., its common name) and the list of certificates created for it. For each created certificate, the “Status” icon is a green circle if the certificate is valid (not expired and not revoked), and the “Identity” icon contains a key if SimpleAuthority maintains also the private key associated to the certificate.

SimpleAuthority CA

- Tools -> Options… -> Identity Files

Tells SimpleAuthority to maintain a copy of private keys



Options

- General
- View Options
- Identity Files
- Default Certificate Settings
- Certificate Types
- Certificate Revocation
- Directory Publishing
- Expiry Date Calendar

Output directory for new identity files: C:\Users\Michelo\Desktop

☑ Also publish certificate file with new identity files

Passwords for new identity files:

● Prompt for new password value

○ Generate a random password of length 8 characters

File format for identity/certificate files: PKCS#12/DER

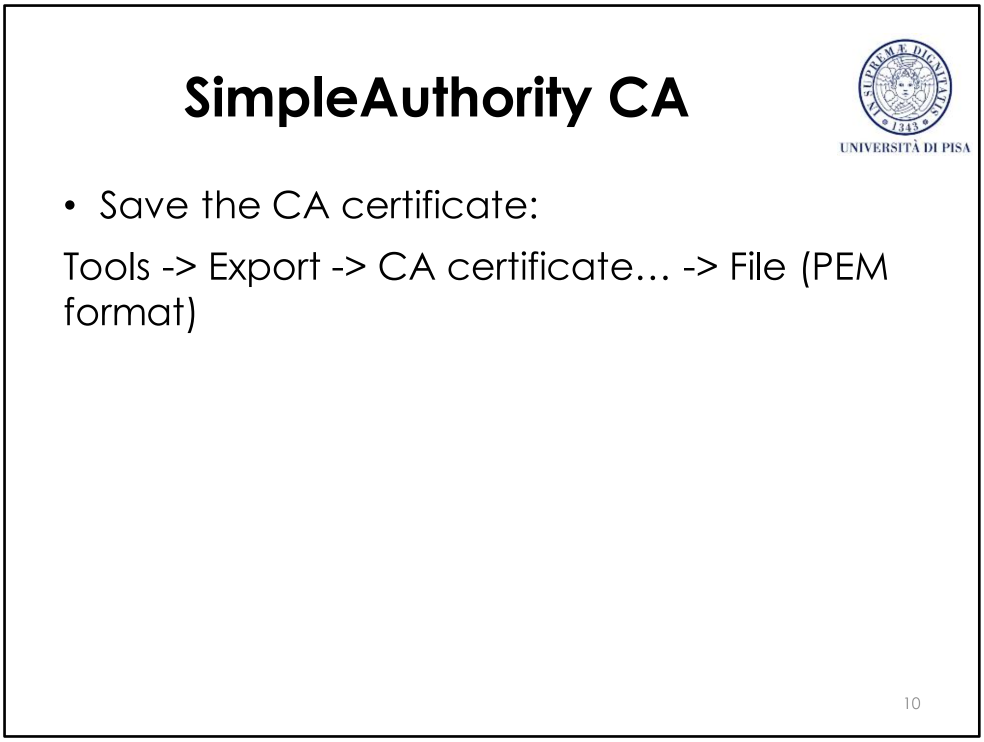
□ Include date stamps in file and alias names

☑ Keep a backup of new user identity files

□ Include a copy of the certificate in PEM identity files

maintain the private key for each created certificate

By checking the “Keep a backup of new user identity files” option in the Option->Identity Files tab, SimpleAuthority will internally maintain a copy of private key for each created certificate. This is necessary to export private keys in PEM format.

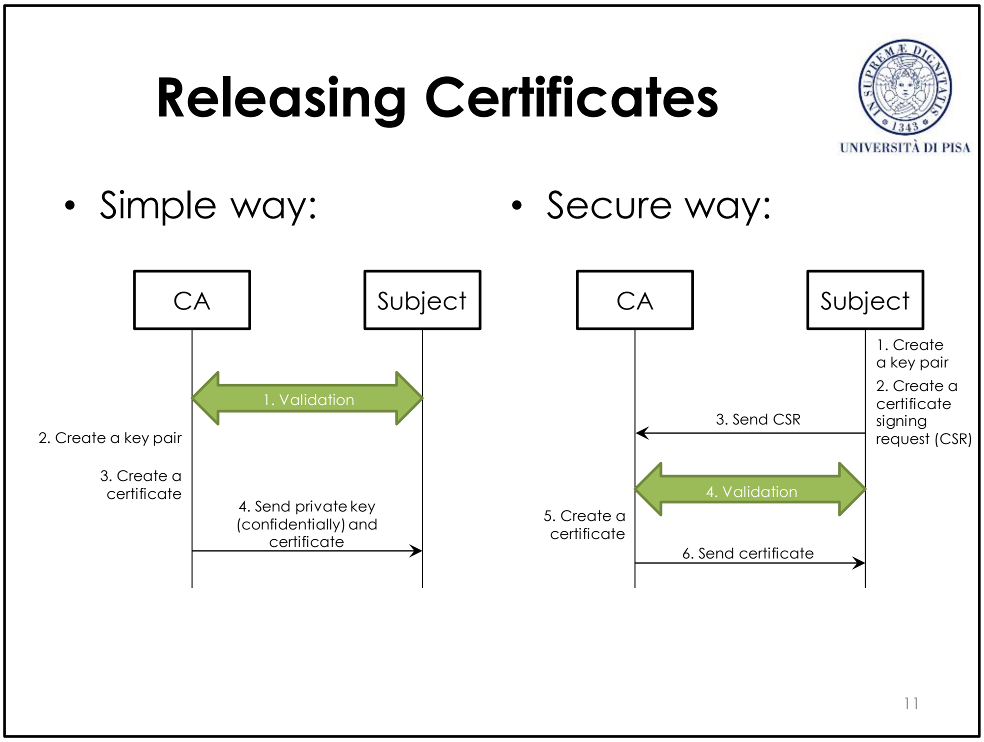


SimpleAuthority CA

- Save the CA certificate:

Tools -> Export -> CA certificate… -> File (PEM format)

This slide shows the instructions to save the X.509 certificate of the certification authority in PEM format. When a Public Key Infrastructure (PKI) is involved, public keys come always included in their certificate. So it is common to use the certificates as replacer of public keys.



Releasing Certificates

- Simple way:
- Secure way:

CA

Subject

1. Validation

2. Create a key pair

3. Create a certificate

4. Send private key (confidentially) and certificate

CA

Subject

1. Create a key pair

2. Create a certificate signing request (CSR)

3. Send CSR

4. Validation

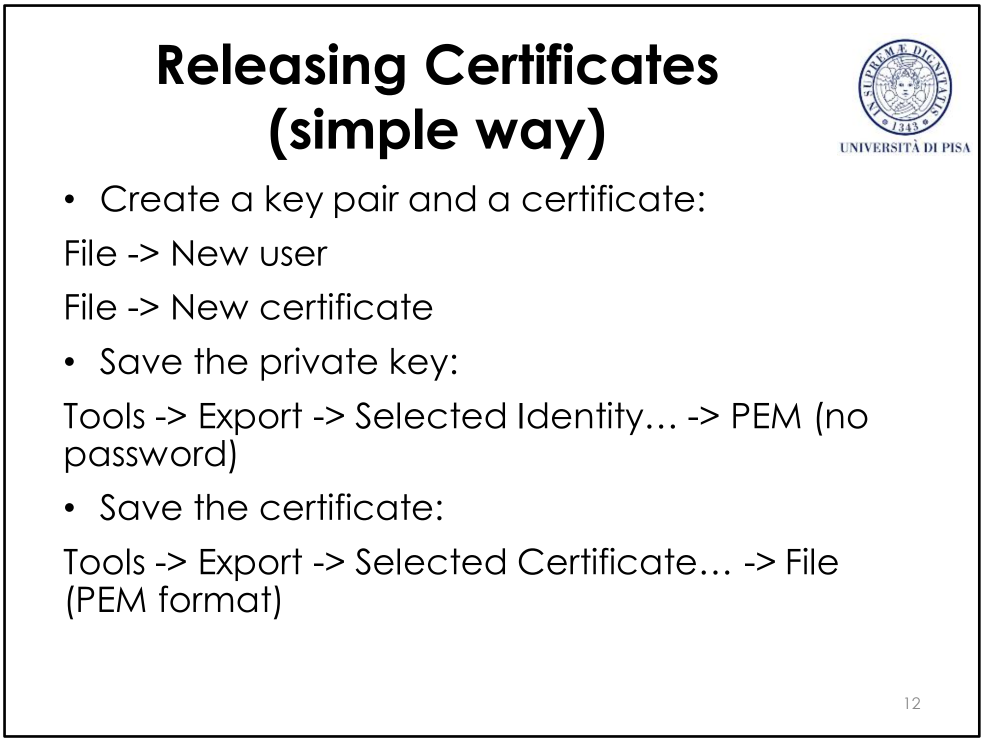
5. Create a certificate

6. Send certificate

There are two ways of creating and deploying a certificate. The simplest way is to create the key pair at the certification authority, create the certificate, and then deploy the private key and the certificate to the subject. Of course, the subject must trust the certification authority to create a good key pair, and to guarantee the confidentiality of the created private key (e.g., by securely destroying it). Moreover, a secure channel is needed to transmit the private key. This way is suitable for small organizations, when the CA and the subject are administrated by a single entity.

A more secure way is to create the key pair at the subject, which creates also a *certificate signing request* (CSR) signed with the newly created private key. The CA receives the CSR, performs the validation, creates the certificate, and send it to the subject. In this way, the subject does not have to trust the CA in creating good key pairs and in destroying the private key, and it does not need a secure channel. This way is suitable when the CA and the subject behold to different organizations.

SimpleAuthority supports both ways.



Releasing Certificates (simple way)

UNIVERSITÀ DI PISA

- Create a key pair and a certificate:

File -> New user

File -> New certificate

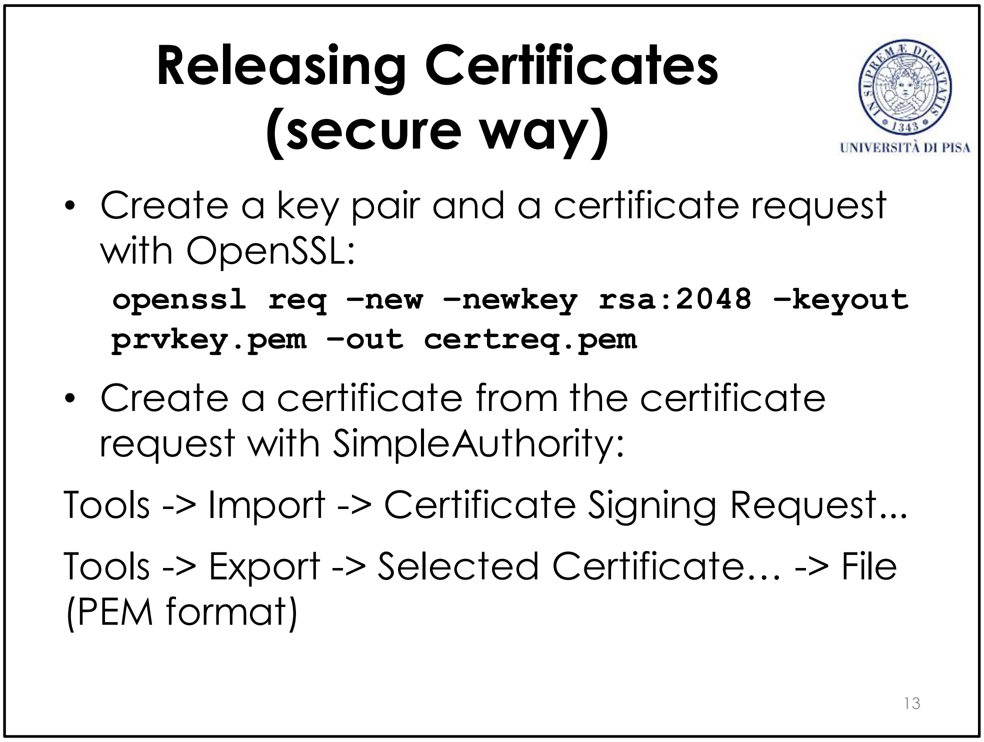
- Save the private key:

Tools -> Export -> Selected Identity… -> PEM (no password)

- Save the certificate:

Tools -> Export -> Selected Certificate… -> File (PEM format)

To create a certificate in the simple way with SimpleAuthority, we first have to create a new user, specifying at least his/her common name. Then, we create a new certificate for the newly created user. This slide shows the instructions to save the private key and the X.509 certificate of the subject (including his/her public key) in PEM format. IMPORTANT NOTE! If you are not able to select «Selected Identity» be sure to have higlighted a certificate inside the certificates window (bottom right of the SimpleAuthority main window).



Releasing Certificates (secure way)

- Create a key pair and a certificate request with OpenSSL:

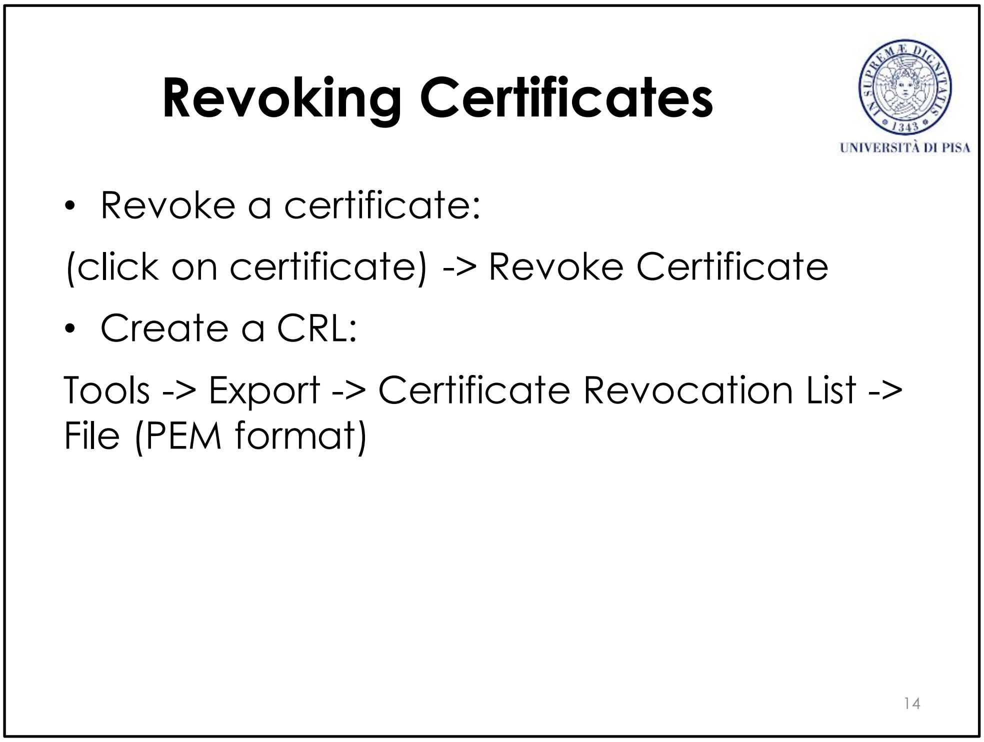
```
openssl req –new –newkey rsa:2048 –keyout
prvkey.pem –out certreq.pem
```

- Create a certificate from the certificate request with SimpleAuthority:

Tools -> Import -> Certificate Signing Request...

Tools -> Export -> Selected Certificate… -> File (PEM format)

To create a certificate in the secure way with SimpleAuthority, the subject have to generate a key pair and a signed CSR with an external tool, for example OpenSSL. The CSR includes also the generated public key, and it can be saved in a PEM format. This slide shows the OpenSSL command line to do it. Then, we have to create a certificate from the CSR with SimpleAuthority.



Revoking Certificates

UNIVERSITÀ DI PISA

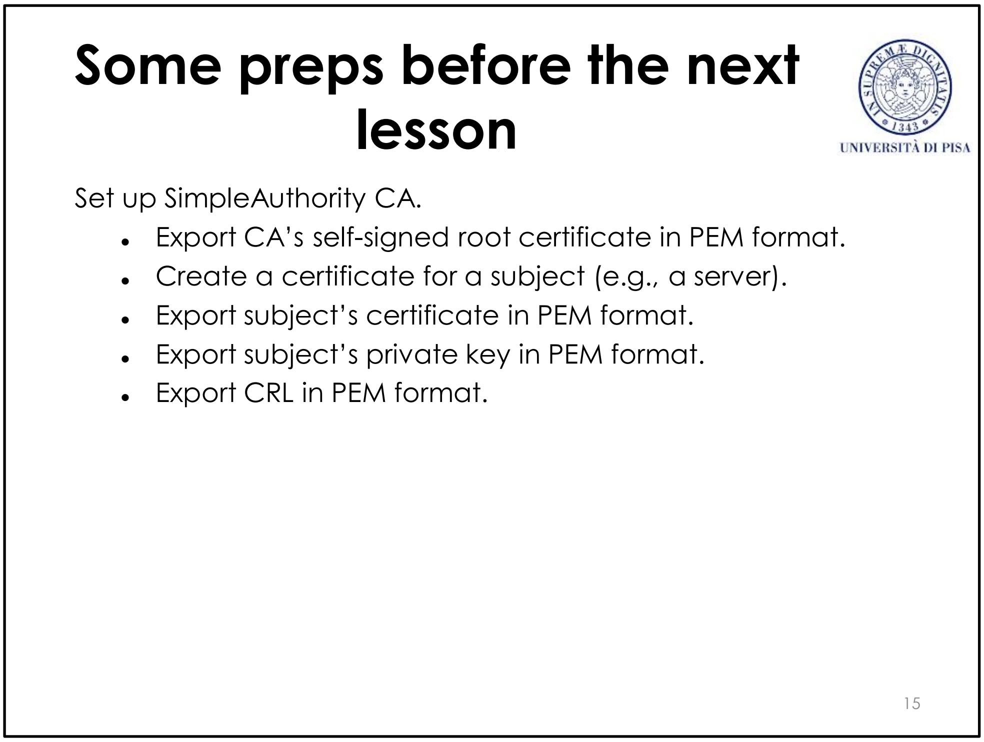
- Revoke a certificate:

(click on certificate) -> Revoke Certificate

- Create a CRL:

Tools -> Export -> Certificate Revocation List -> File (PEM format)

This slide shows the instructions to revoke a certificate and save a signed CRL in PEM format.



Some preps before the next lesson

Set up SimpleAuthority CA.

- Export CA’s self-signed root certificate in PEM format.
- Create a certificate for a subject (e.g., a server).
- Export subject’s certificate in PEM format.
- Export subject’s private key in PEM format.
- Export CRL in PEM format.