Authentication & Authorization

Nadav Grossaug

Nadav.Grossaug@isragrid.org.il

Material from:

Andrea Sciabà

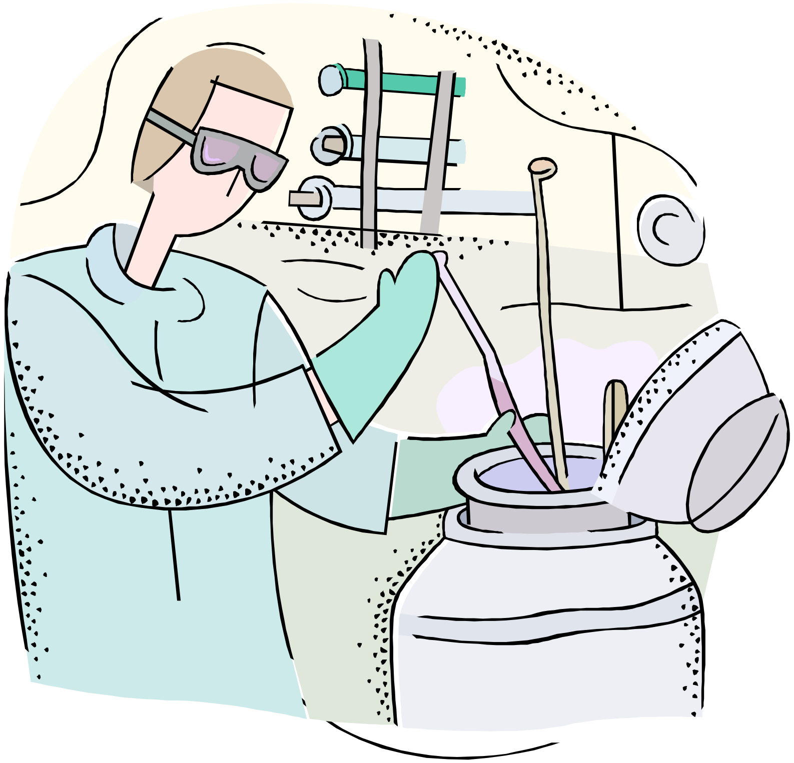
Åke Edlund, JRA3 Manager, KTH

David Groep, EUGridPMA chair, NIKHEF

Information Society

Overview

- Basic security concepts
- Certificates & Proxies – Authentication
- Virtual Organisations - Authorization



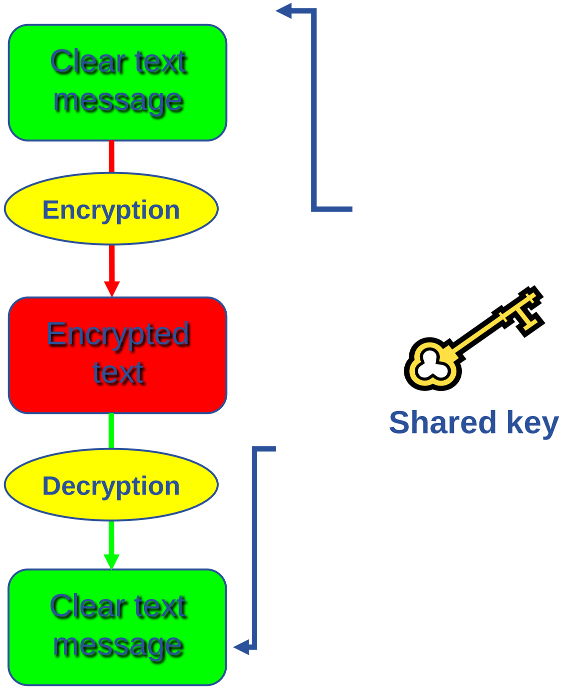
Basic security concepts

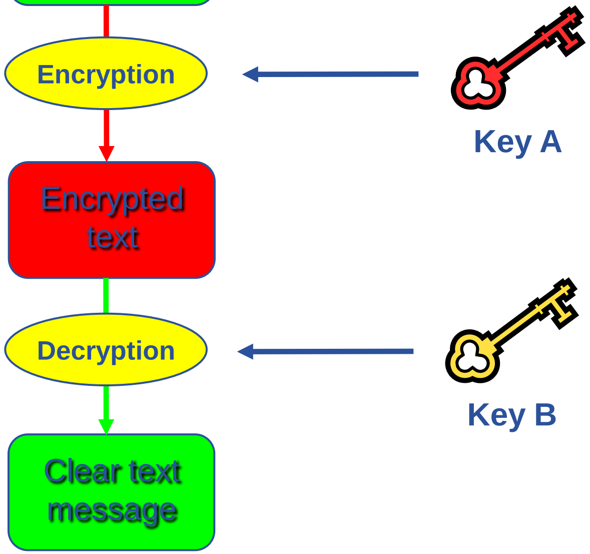
- Authentication
  - Verify the identity of the peer
- Authorization
  - Map an entity to some set of privileges
- Confidentiality
  - Encrypt the message so that only the recipient can understand it
- Integrity
  - Ensure that the message has not be altered in the transmission
- Non-repudiation
  - Impossibility of denying the authenticity of a digital signature
- Accounting
  - What did you do, when did you do it and where did you do it from?

Encryption

- **Symmetric encryption: same key ("secret") used for encryption and decryption**
  - Kerberos, DES / 3DES, IDEA

- Asymmetric encryption: different keys used for encryption and decryption
  - RSA, DSA





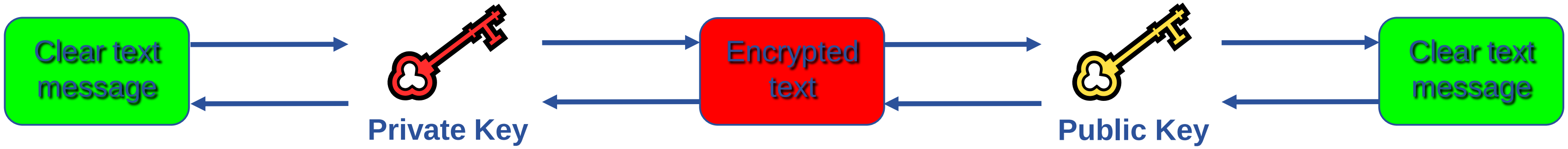
Using Asymmetric Keys

- **Sending a message**
  - Encrypt message using Receiver’s public key
  - Send encrypted message
  - Receiver decrypts message using own private key

  Only someone with Receiver’s private key can decrypt message

- **Authenticating**
  - Encrypt message with Sender’s private key
  - Send encrypted message
  - Message is readable by ANYONE with Sender’s public key
  - Receiver decrypts message with Sender’s public key

Receiver can be confident that only someone with Sender’s private key could have sent the message



Digital Certificates

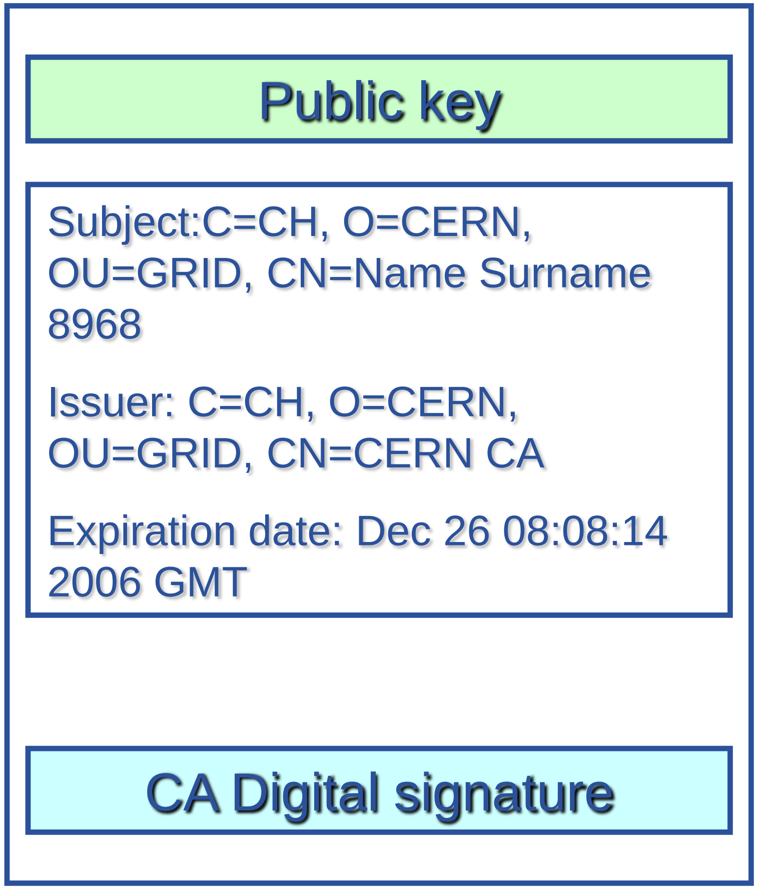
- Digital signatures
  - A hash derived from the message and encrypted with the signer’s private key
  - Signature checked decrypting with the signer’s public key
- A’s digital signature is safe if:
  1. A’s private key is not compromised
  2. B knows A’s public key
- How can B be sure that A’s public key is really A’s public key and not someone else’s?
  - A *third party* guarantees the correspondence between public key and owner’s identity, by signing a document which contains the owner’s identity and his public key (**Digital Certificate**)
  - Both A and B must trust this third party
- Two models:
  - X.509: hierarchical organization;
  - PGP: “web of trust”.

Certification Authorities

- **Issue certificates for users, programs and machines**
- **Check the identity and the personal data of the requestor**
  - Registration Authorities (RAs) do the actual validation
- **Manage Certificate Revocation Lists (CRLs)**
  - They contain all the revoked certificates yet to expire
- **CA certificates are self-signed**

X.509: content of the Certificate

- **An X.509 Certificate contains:**
  - owner’s public key;
  - identity of the owner;
  - info on the CA;
  - time of validity;
  - digital signature of the CA



Public key

Subject:C=CH, O=CERN, OU=GRID, CN=Name Surname 8968

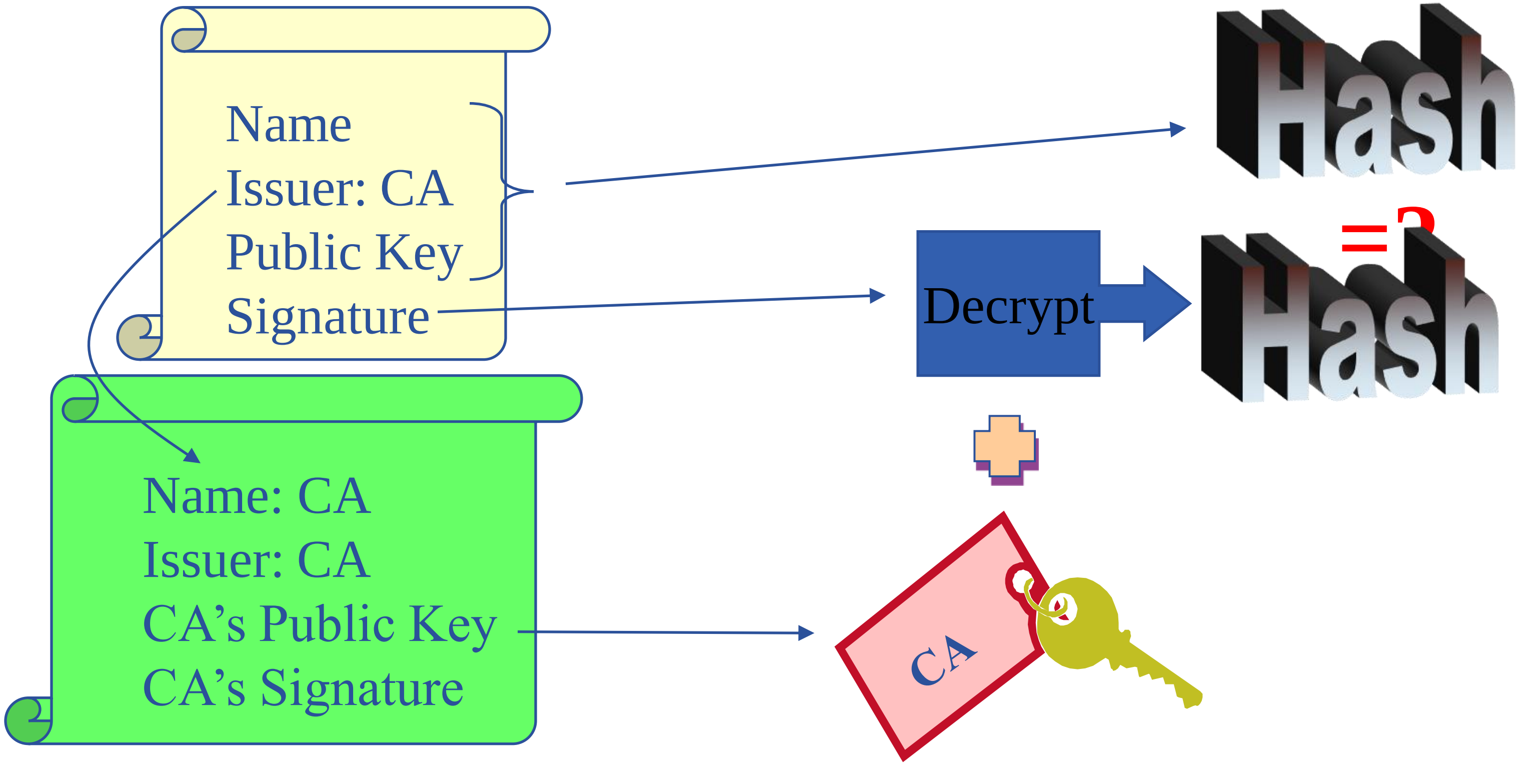
Issuer: C=CH, O=CERN, OU=GRID, CN=CERN CA

Expiration date: Dec 26 08:08:14 2006 GMT

CA Digital signature

Certificate Validity

- **The public key from the CA certificate can then be used to verify the certificate.**



slide based on presentation given by Carl Kesselman at GGF Summer School 2004

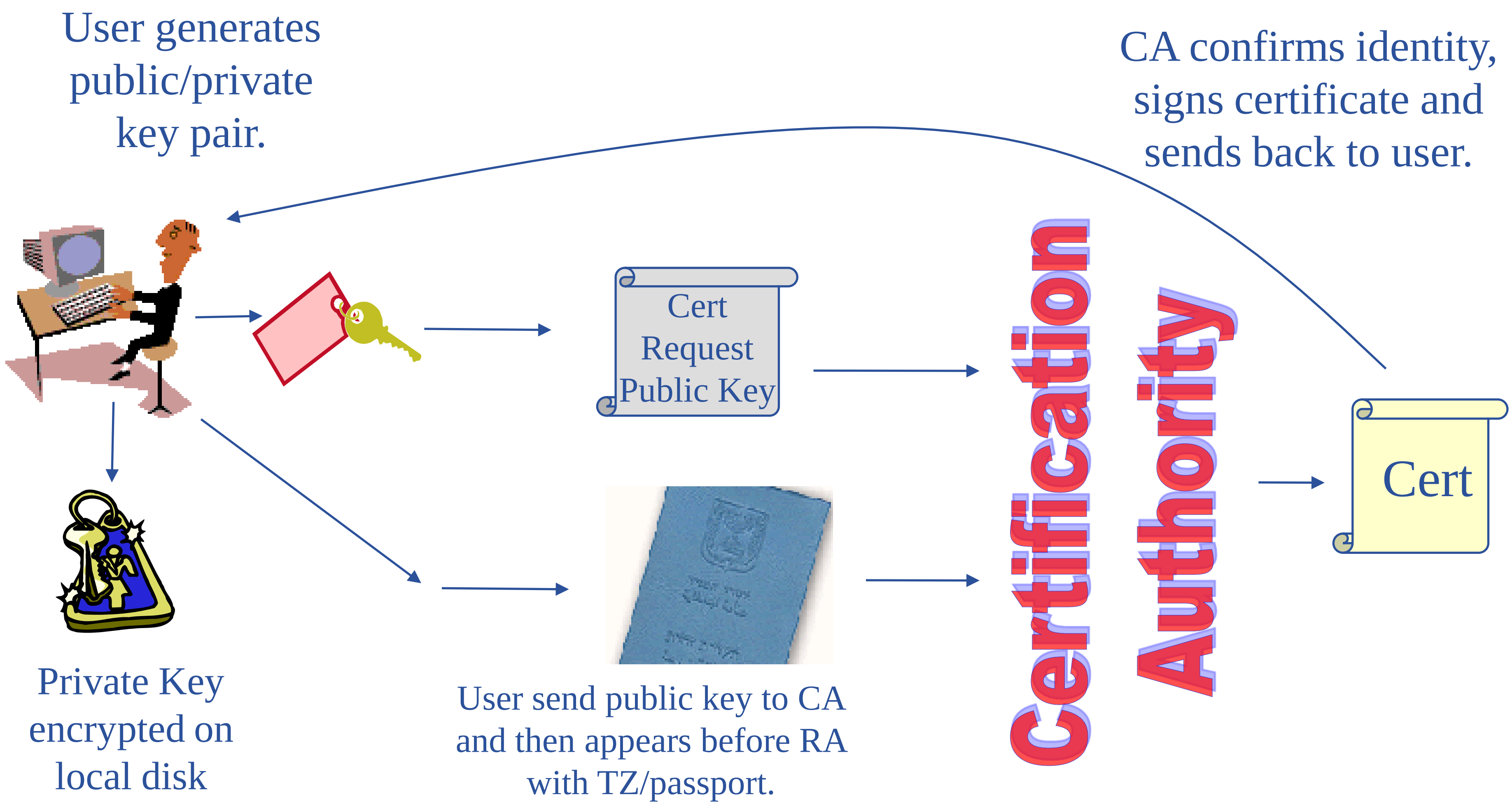
User Responsibilities

- Keep your private key secure.
- Do not loan your certificate to anyone.
- Report to your local/regional contact if your certificate has been compromised.
- Do not launch a delegation service for longer than your current task needs.

If your certificate or delegated service is used by someone other than you, it cannot be proven that it was not you.

IT IS YOUR PASSPORT AND CREDIT CARD

Certificate Request



User generates public/private key pair.

Cert Request Public Key

Certification Authority

CA confirms identity, signs certificate and sends back to user.

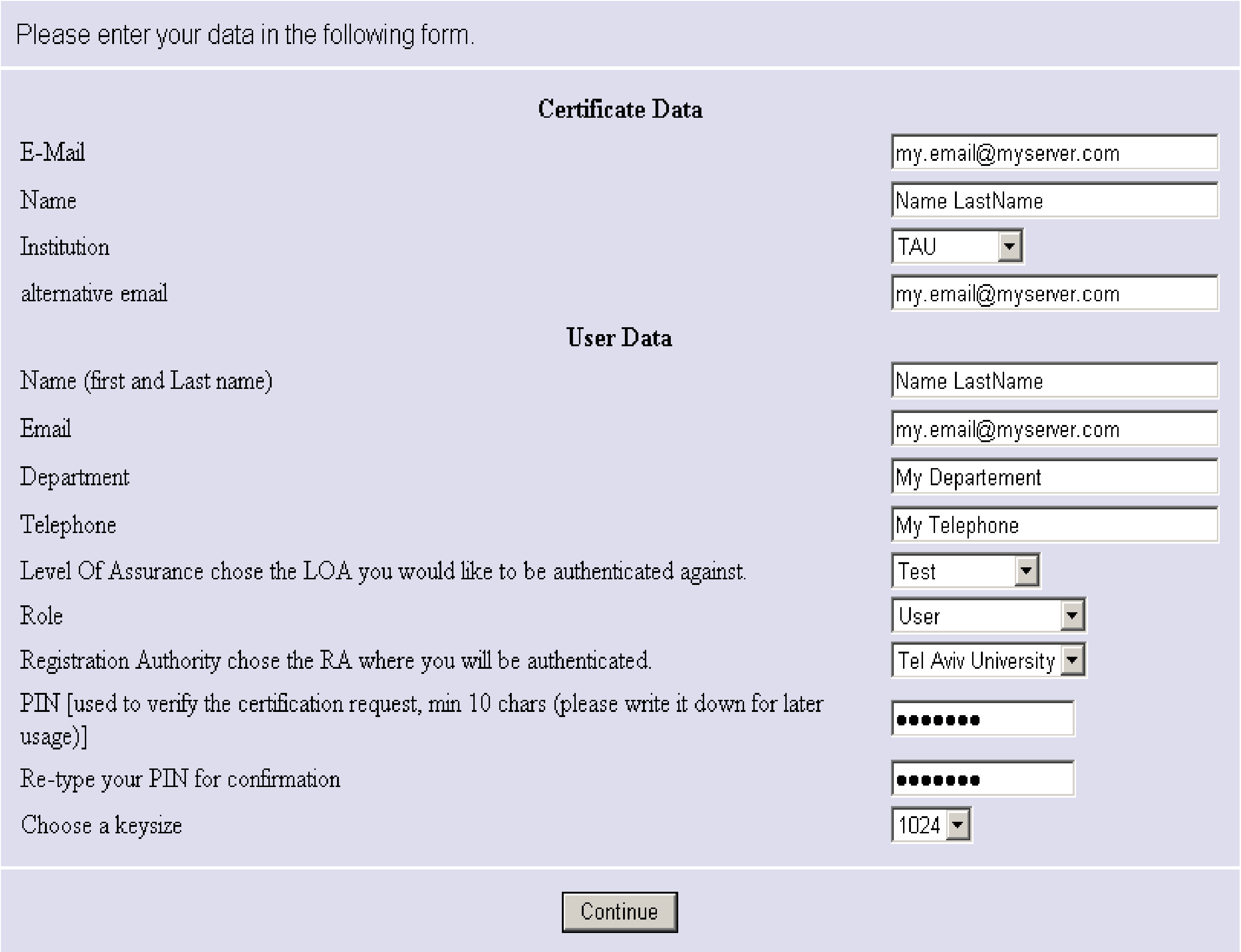
Cert

Private Key encrypted on local disk

User send public key to CA and then appears before RA with TZ/passport.

Request a certificate

- Requesting a certificate - https://certificate.iucc.ac.il/
- Receiving the certificate - https://certificate.iucc.ac.il/pub



Please enter your data in the following form.

Certificate Data

E-Mail	my.email@myserver.com
Name	Name LastName
Institution	TAU
alternative email	my.email@myserver.com

User Data

Name (first and Last name)	Name LastName
Email	my.email@myserver.com
Department	My Departement
Telephone	My Telephone
Level Of Assurance chose the LOA you would like to be authenticated against.	Test
Role	User
Registration Authority chose the RA where you will be authenticated.	Tel Aviv University
PIN [used to verify the certification request, min 10 chars (please write it down for later usage)]	●●●●●●●
Re-type your PIN for confirmation	●●●●●●●
Choose a keysize	1024

LIST of Israeli CA and RAs

- **Eddie Aronovich, Certificate Authority Manager eddiea@tau.ac.il, 03-6406915**
- **Currenlty also performing RA role.**

University	Name	e-mail	phone
Hebrew	Ayelet Hashachar Drori	ayeleth@savion.cc.huji.ac.il	02-6584475
Haifa	Herakel Endrawes	herakel@univ.haifa.ac.il	04-8249249
Technion	Anne Weill	anne@tx.technion.ac.il	04-8294997
Weizmann	Pierre Choukroun	pierre@weizmann.ac.il	08-9343038
BGU	Amir Zofnat	zofnat@bgu.ac.il	08-6479449
Open-U	Reuven Aviv	aviv@openu.ac.il	09-7781252
TAU	Avi Raber	avir@tauex.tau.ac.il	03-6409117

X.509: security inside the Grid

- **For the Grid to be an effective framework for largely distributed computation, users, user processes and grid services must work in a secure environment.**

- **The user has to possess a valid X.509 certificate on the submitting machine, consisting of two files:**
  the *certificate file* and the *private key file*.

  - ***"$HOME/.globus/usercert.pem“***
  - ***"$HOME/.globus/userkey.pem"***

X.509: extracting user{cert | key} files

Usually X.509 Certificates are downloaded using a browser and managed by the browser itself.

- Anyway it is possible to export your certificate in a file PKCS12 (which will probably have the extension .p12 or .pfx).

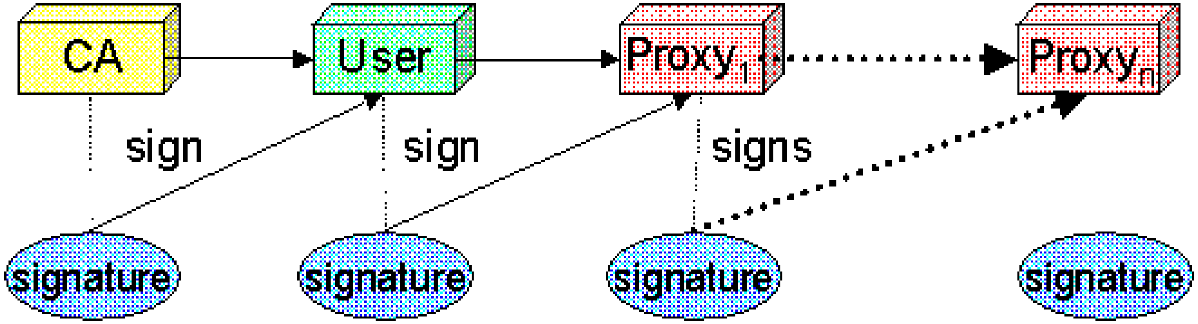
- Unfortunately PKCS12 format is not accepted by Globus security infrastructure, but you can easily convert it into the supported standard (PEM). This operation will split your *.p12 file in two files: the certificate (usercert.pm) and the private key (userkey.pm).

- *With openssl tool:*
- *$ openssl pkcs12 -nocerts -in mycert.p12 -out userkey.pem*
- *$ openssl pkcs12 -clcerts -nokeys -in mycert.p12 -out usercert.pem*
- *$ chmod 0400 userkey.pem*
- *$ chmod 0600 usercert.pem*

- Permission must be set as shown not only for security reasons: *voms-proxy-init* and *grid-proxy-init* commands will fail if your private key is not protected as listed above.

Globus Grid Security Infrastructure (GSI)

- *de facto* **standard for Grid middleware**
- **Based on PKI**
- **Implements some important features**
  - Single sign-on: no need to give one’s password every time
  - Delegation: a service can act on behalf of a person
  - Mutual authentication: both sides must authenticate to the other
- **Introduces proxy certificates**
  - Short-lived certificates including their private key and signed with the user’s certificate



Command line interface: certificate and proxy management

- Get information on a user certificate
  - `grid-cert-info[-help] [-file certfile] [OPTION]...`

`-all`	whole certificate
`-subject | -s`	subject string
`-issuer | -I`	Issuer
`-startdate | -sd`	Start of validity
`-enddate | -ed`	End of validity

- Create a proxy certificate
  - `grid-proxy-init/voms-proxy-init`
- Destroy a proxy certificate
  - `grid-proxy-destroy/voms-proxy-destroy`
- Get information on a proxy certificate
  - `grid-proxy-info/voms-proxy-info`

Long term proxy

- **Proxy has limited lifetime (default is 12 h)**
  - Bad idea to have longer proxy
- **However, a grid task might need to use a proxy for a much longer time**
- **myproxy server:**
  - **Consists of a server and a set of client tools that can be used to delegate and retrieve credentials to and from a server.**
  - `myproxy-init -s <host_name> -d -n`
    - `-s <host_name>` specifies the hostname of the myproxy server
  - `myproxy-info`
    - Get information about stored long living proxy
  - `myproxy-get-delegation`
    - Get a new proxy from the MyProxy server
  - `myproxy-destroy`
- **A service running continuously can renew automatically a proxy created from a long term use proxy and use it to interact with the Grid**

Virtual Organizations and authorization

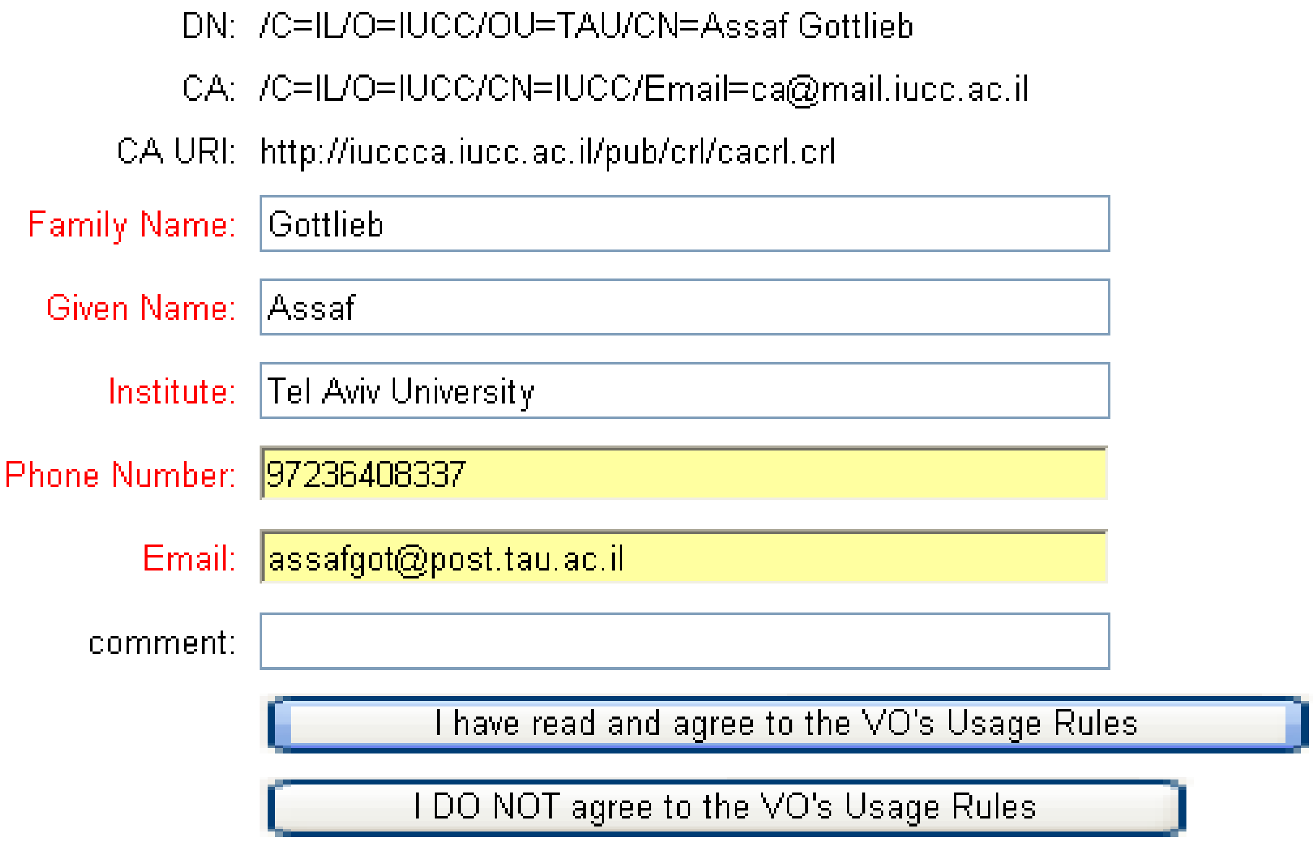
- **gLite users MUST belong to a Virtual Organization**
  - Sets of users belonging to a collaboration
  - Each VO user has the same access privileges to Grid resources
  - List of supported VOs:
    - https://lcg-registrar.cern.ch/virtual_organization.html
- **VOs maintain a list of their members**
  - The list is downloaded by Grid machines to map user certificate subjects to local “pool” accounts: only mapped users are authorized in gLite

```
...
"/C=CH/O=CERN/OU=GRID/CN=Simone Campana 7461" .dteam
"/C=CH/O=CERN/OU=GRID/CN=Andrea Sciaba 8968" .cms
"/C=CH/O=CERN/OU=GRID/CN=Patricia Mendez Lorenzo-ALICE" .alice
...
```

  - Sites decide which VOs to accept
  - A list of supported VOs can be found here:
    - *https://lcg-registrar.cern.ch/virtual_organization.html*

Singing for a VO

- **Major VOs can be joined through https://lcg-registrar.cern.ch/cgi-bin/register/account.pl**



DN: /C=IL/O=IUCC/OU=TAU/CN=Assaf Gottlieb

CA: /C=IL/O=IUCC/CN=IUCC/Email=ca@mail.iucc.ac.il

CA URI: http://iuccca.iucc.ac.il/pub/crl/cacrl.crl

Family Name: Gottlieb

Given Name: Assaf

Institute: Tel Aviv University

Phone Number: 97236408337

Email: assafgot@post.tau.ac.il

comment:

I have read and agree to the VO's Usage Rules
I DO NOT agree to the VO's Usage Rules

Summary

- **In order to use the grid a user must have**
  - A valid certificate, given by the CA
  - Join a VO.

- **Each action on the grid requires a valid Proxy, generated from your certificate.**

- **Long duration jobs can use MyProxy server for automatic generation of proxies.**

- **Instructions available at http://iag.iucc.ac.il/workshop-2006II/JoinGrid.htm**