the globus project™
www.globus.org

Grid Security Infrastructure

Globus Toolkit™ Developer Tutorial

The Globus Project™

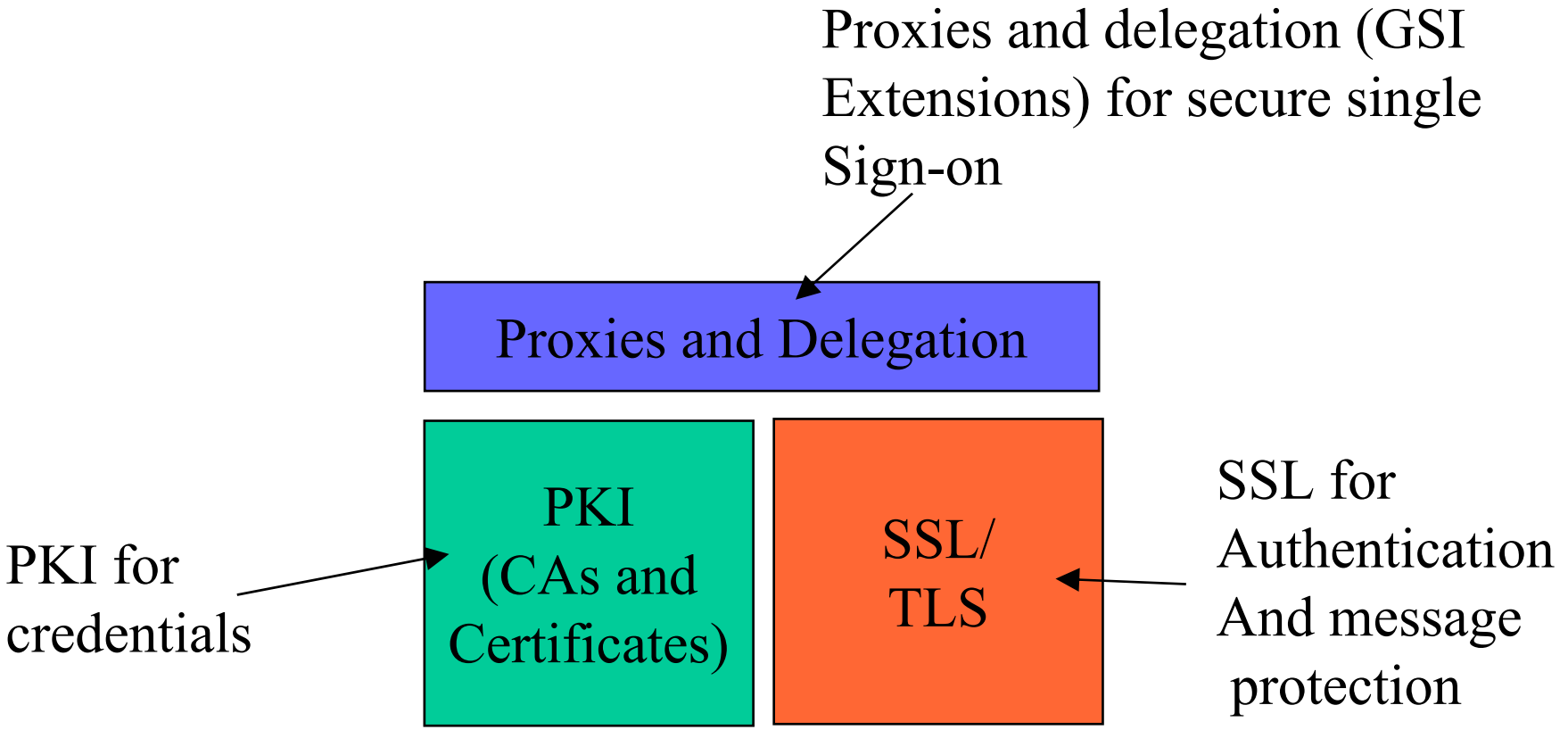
Argonne National Laboratory

USC Information Sciences Institute

http://www.globus.org/

Grid Security Infrastructure (GSI)

- GSI is:



Public Key Infrastructure (PKI)

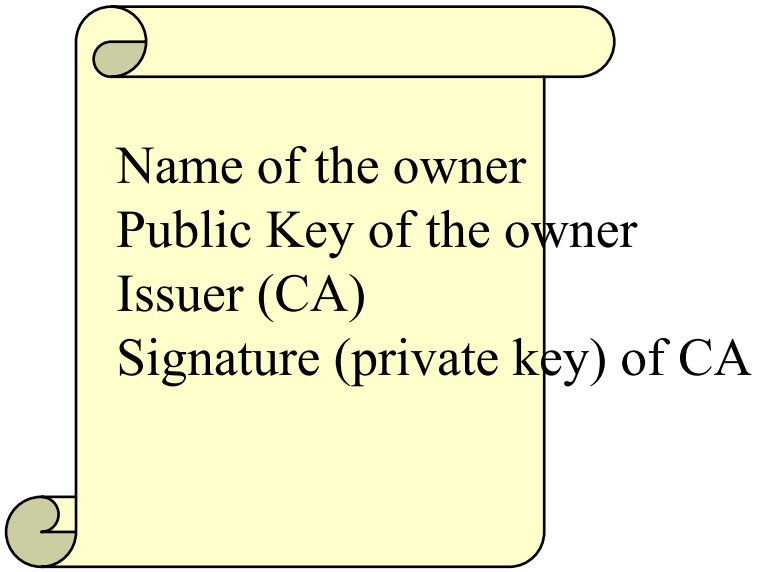
- PKI allows you to know that a given public key belongs to a given user
- PKI builds off of asymmetric encryption:
  - Each entity has two keys: public and private
  - Data encrypted with one key can only be decrypted with other.
  - The private key is known only to the entity
- The public key is given to the world encapsulated in a X.509 certificate

Public Key Infrastructure (PKI) Overview

- X.509 Certificates
- Certificate Authorities (CAs)
- Certificate Policies
  - Namespaces
- Requesting a certificate
  - Certificate Request
  - Registration Authority

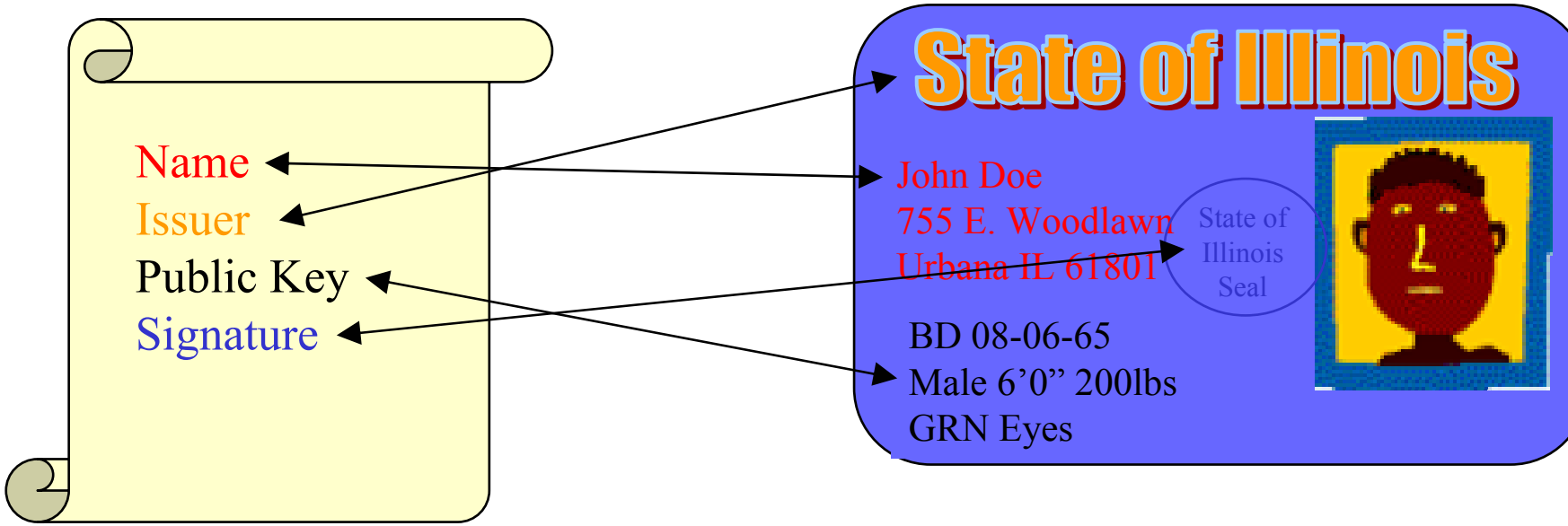
Certificates

- A X.509 certificate binds a public key to a name
- It includes a name and a public key (among other things) bundled together and signed by a trusted party (Issuer)



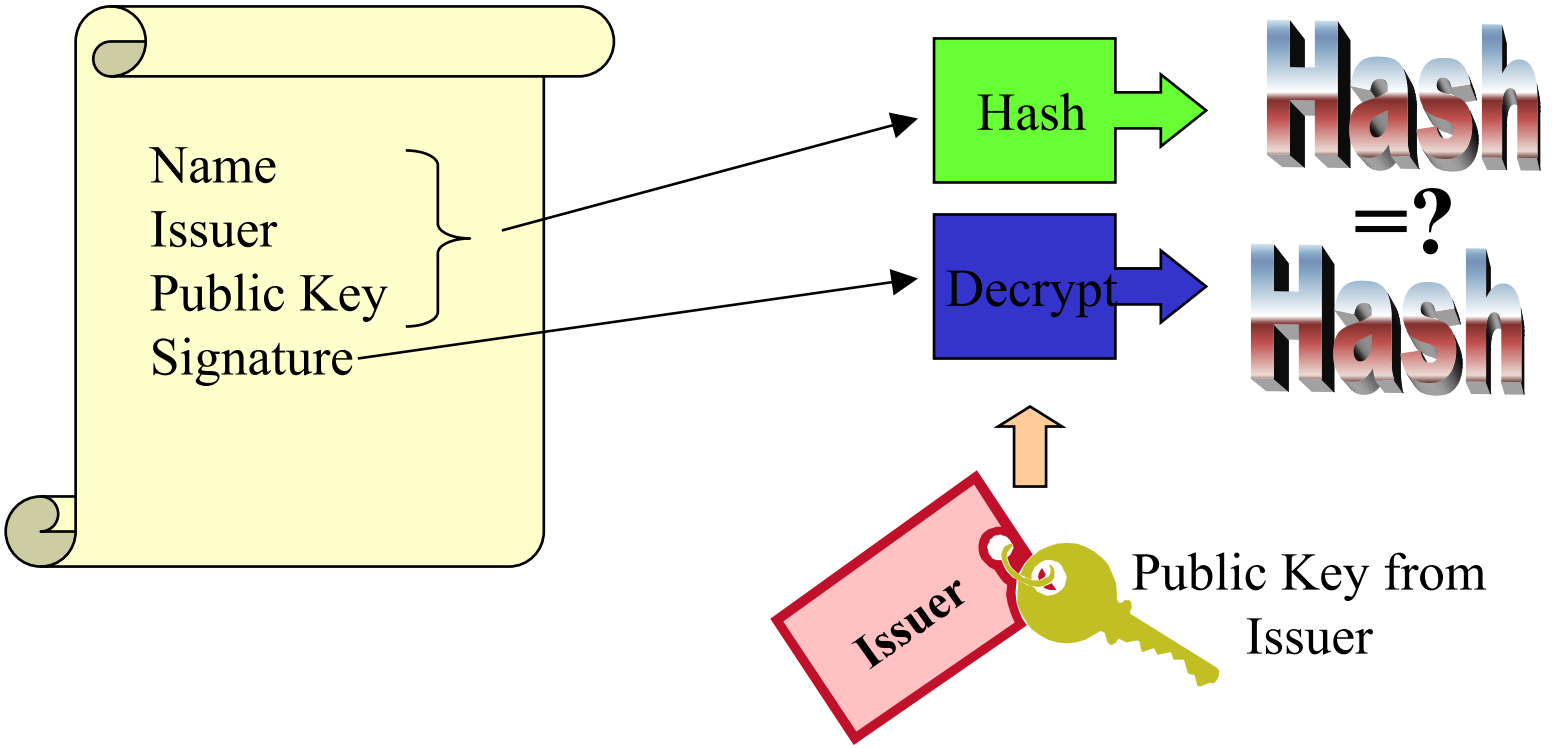
Certificates

- Similar to passport or driver's license



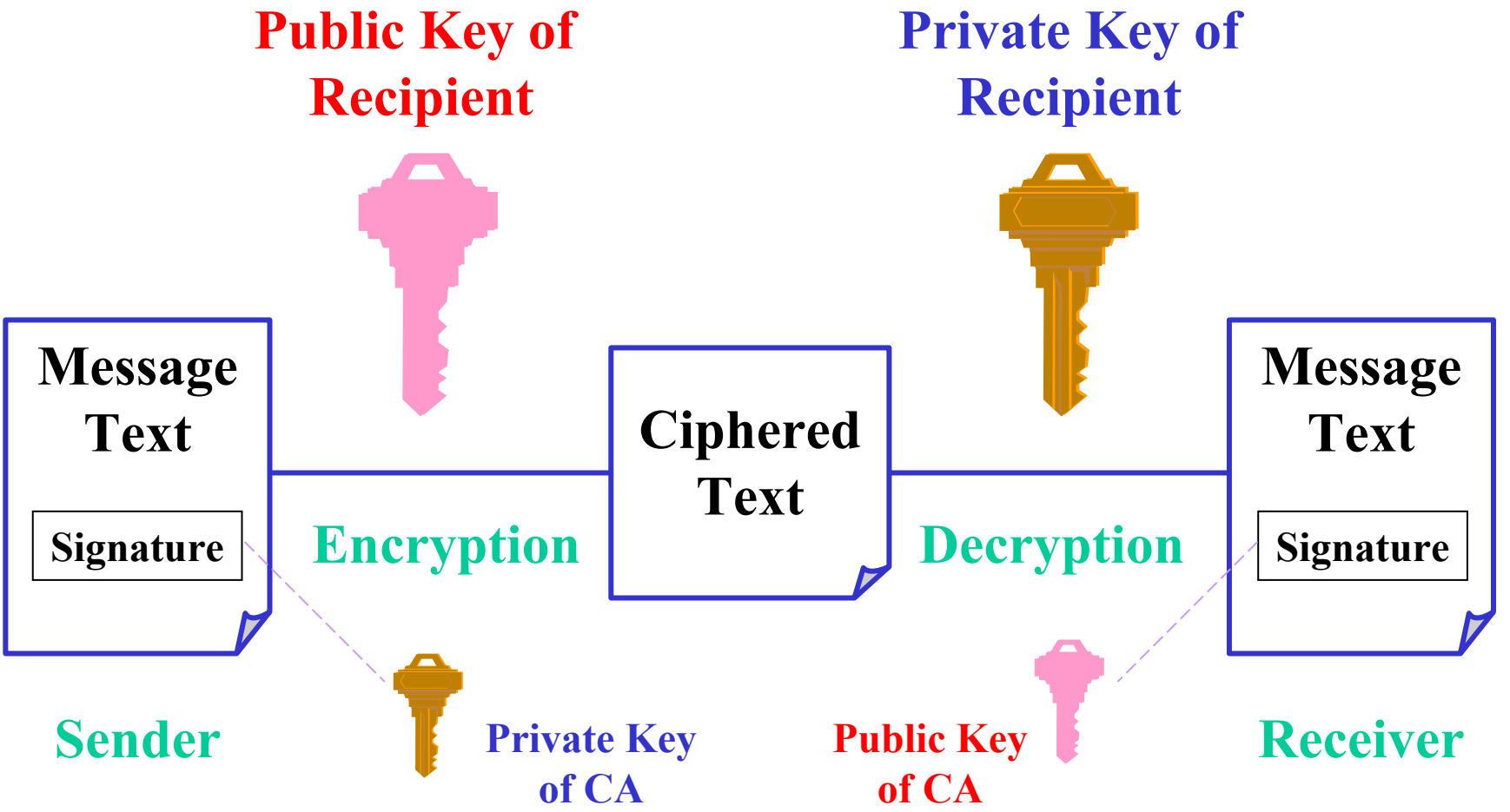
Certificates

- By checking the signature, one can determine that a public key belongs to a given user.



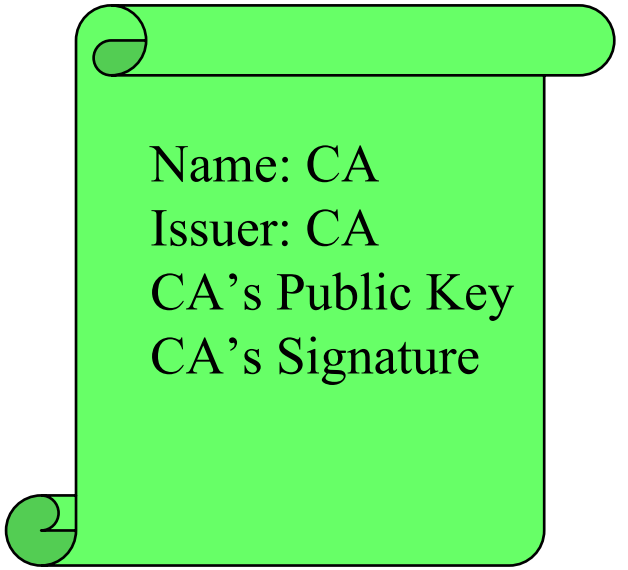
Digital Signatures

- Digital Signatures : Authenticity and Non-Denial



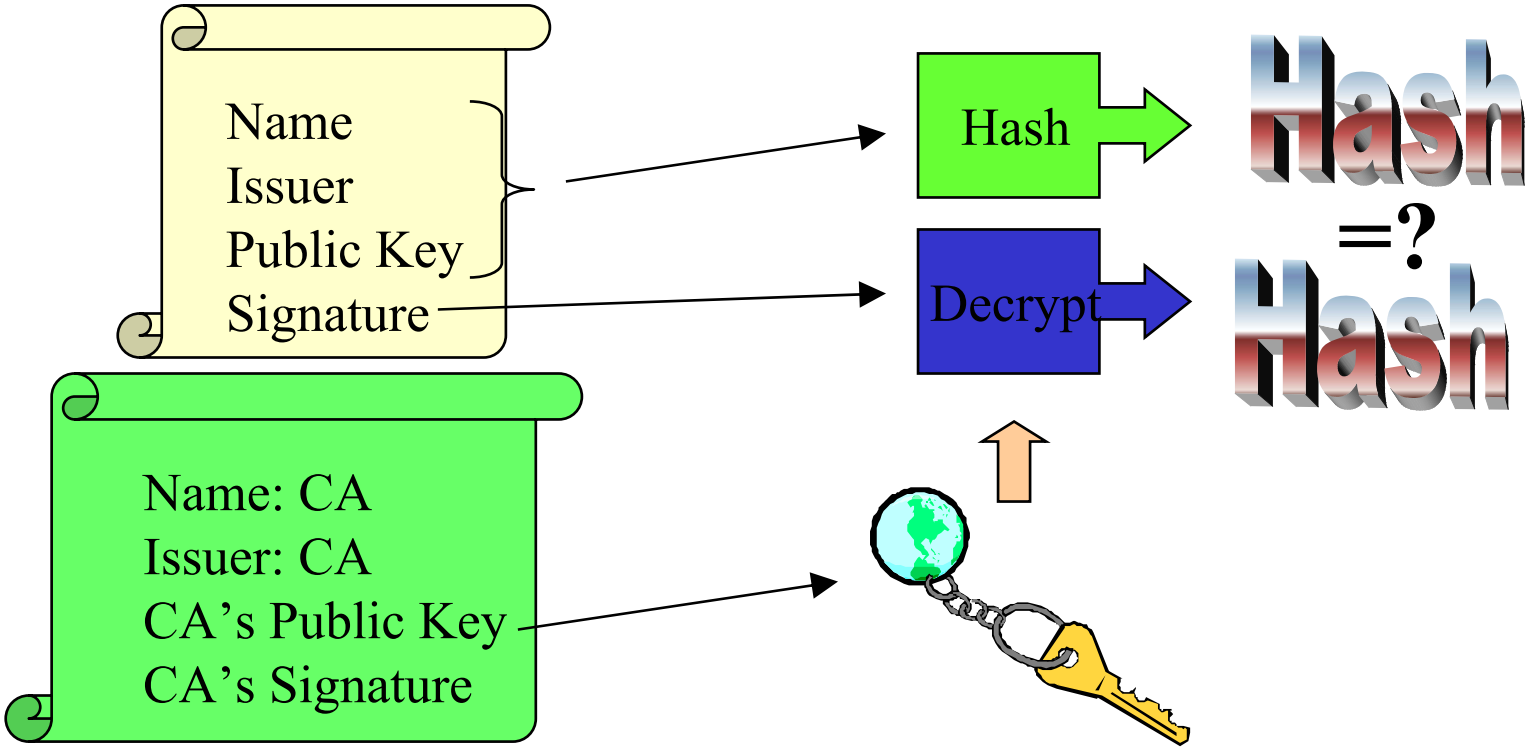
Certificate Authorities (CAs)

- A small set of trusted entities known as Certificate Authorities (CAs) are established to sign certificates
- A Certificate Authority is an entity that exists only to sign user certificates
- The CA signs it's own certificate which is distributed in a trusted manner



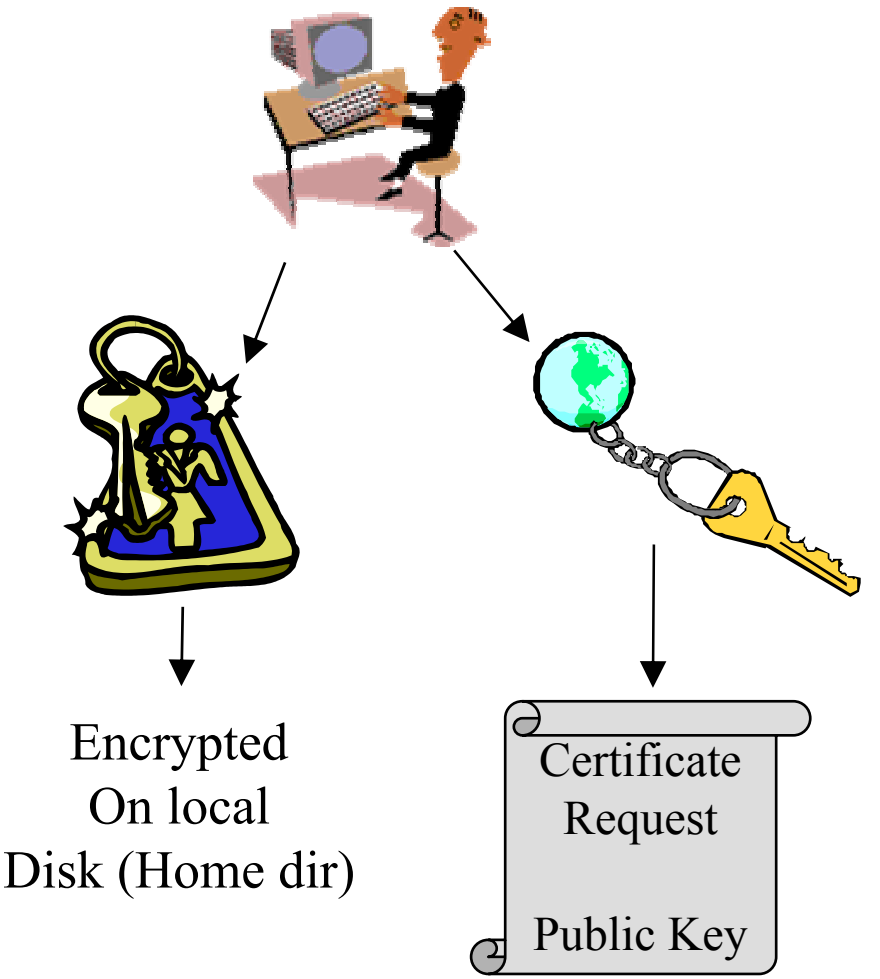
Certificate Authorities (CAs)

- The public key from the CA certificate can then be used to verify other certificates



Requesting a Certificate

- To request a certificate a user starts by generating a key pair
- The private key is stored encrypted with a pass phrase the user gives
- The public key is put into a certificate request

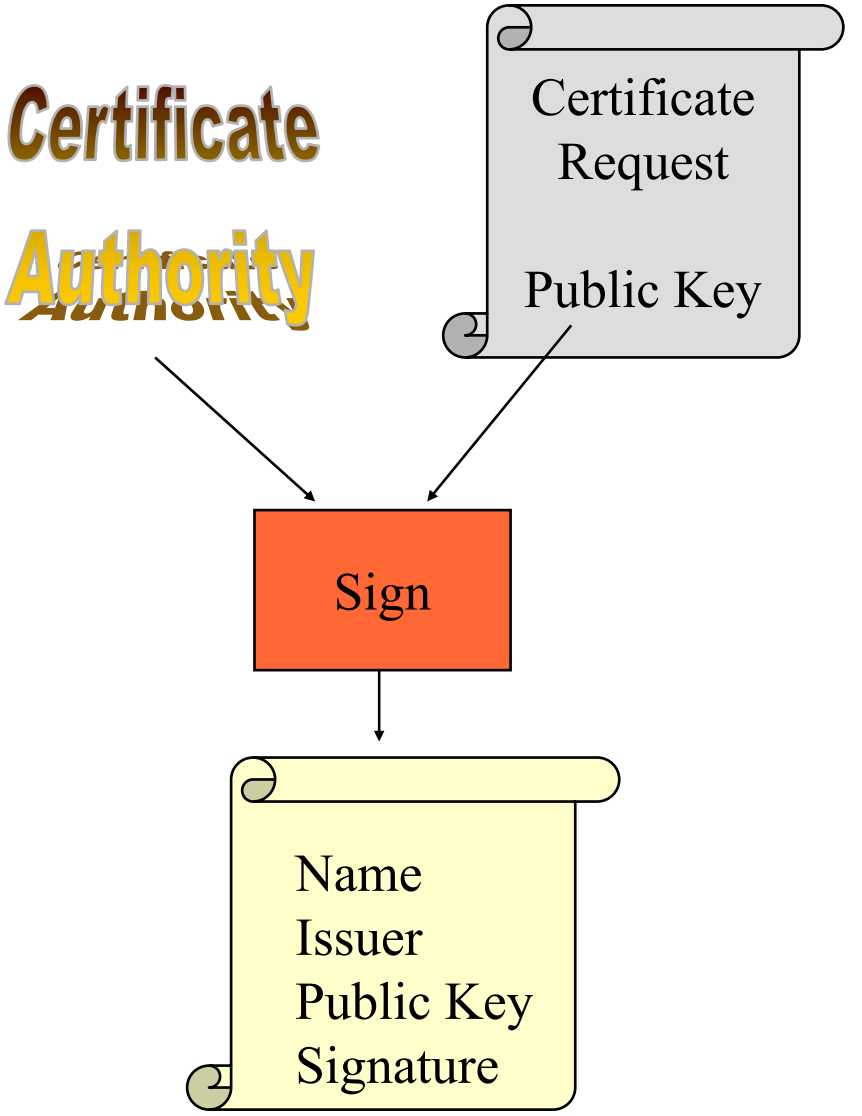


Certificate Issuance

- The user then takes the certificate to the CA
- The CA usually includes a Registration Authority (RA) which verifies the request:
  - The name is unique with respect to the CA
  - It is the real name of the user
  - Etc.

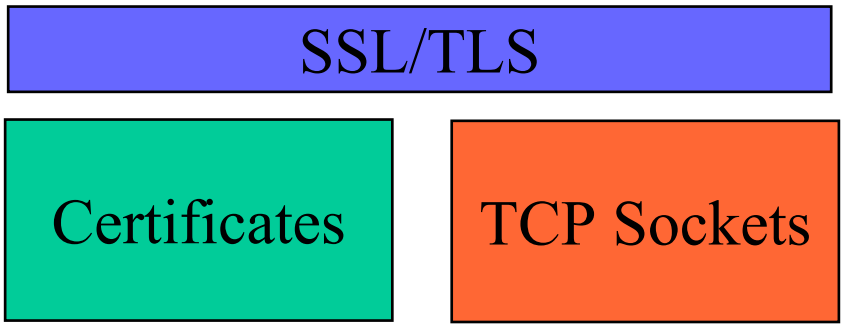
Certificate Issuance

- The CA then signs the certificate request and issues a certificate for the user

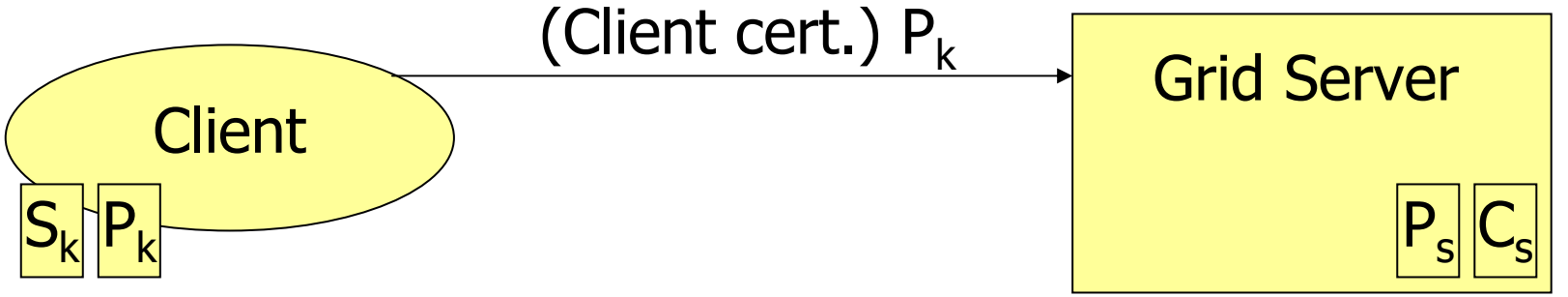


Secure Socket Layer (SSL)

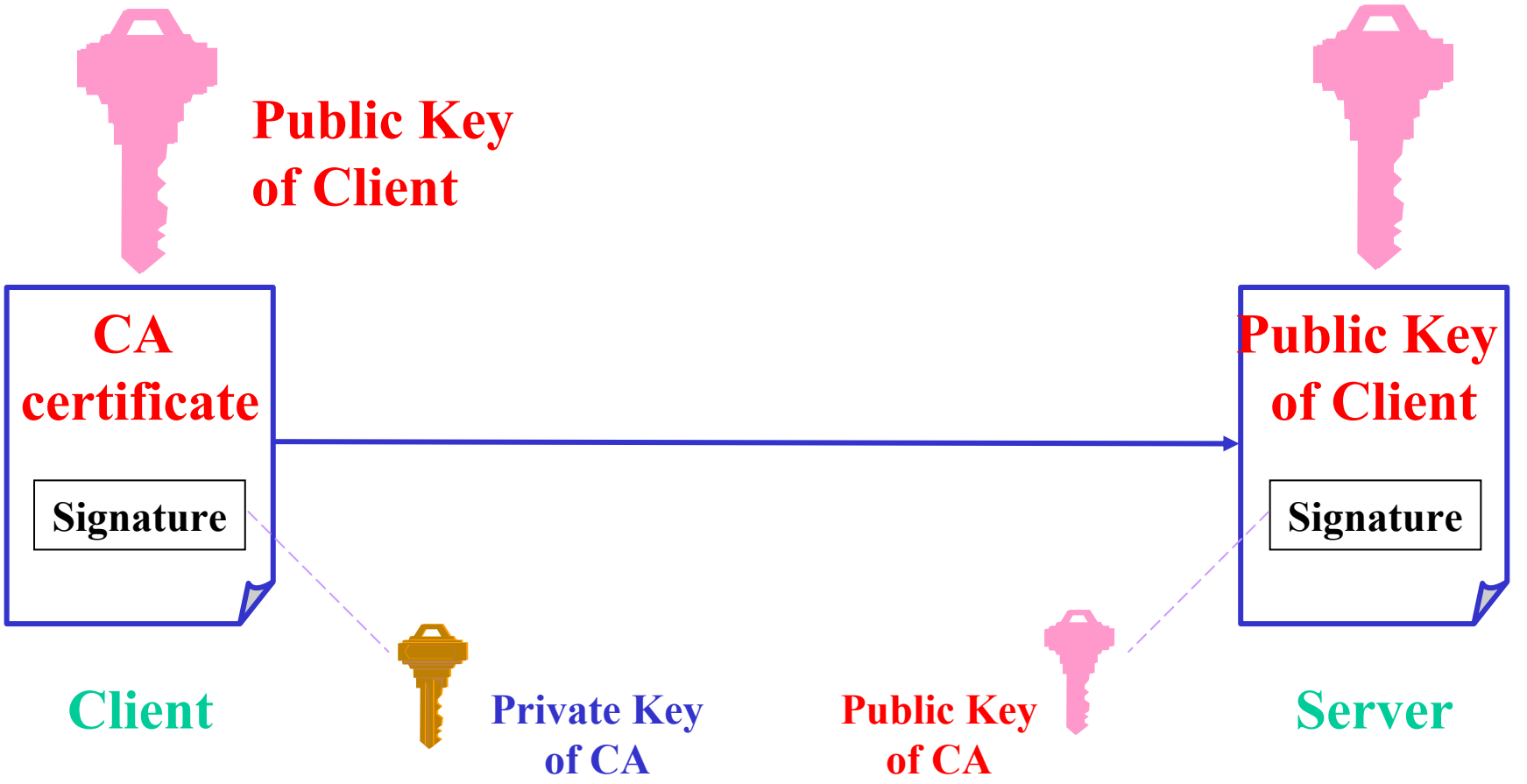
- Also known as TLS (Transport Layer Security)
- Uses certificates and TCP sockets to provide a secured connection
  - Authentication of one or both parties using the certificates
  - Message protection
    - > Confidentiality (encryption)
    - > Integrity



Mutual authentication

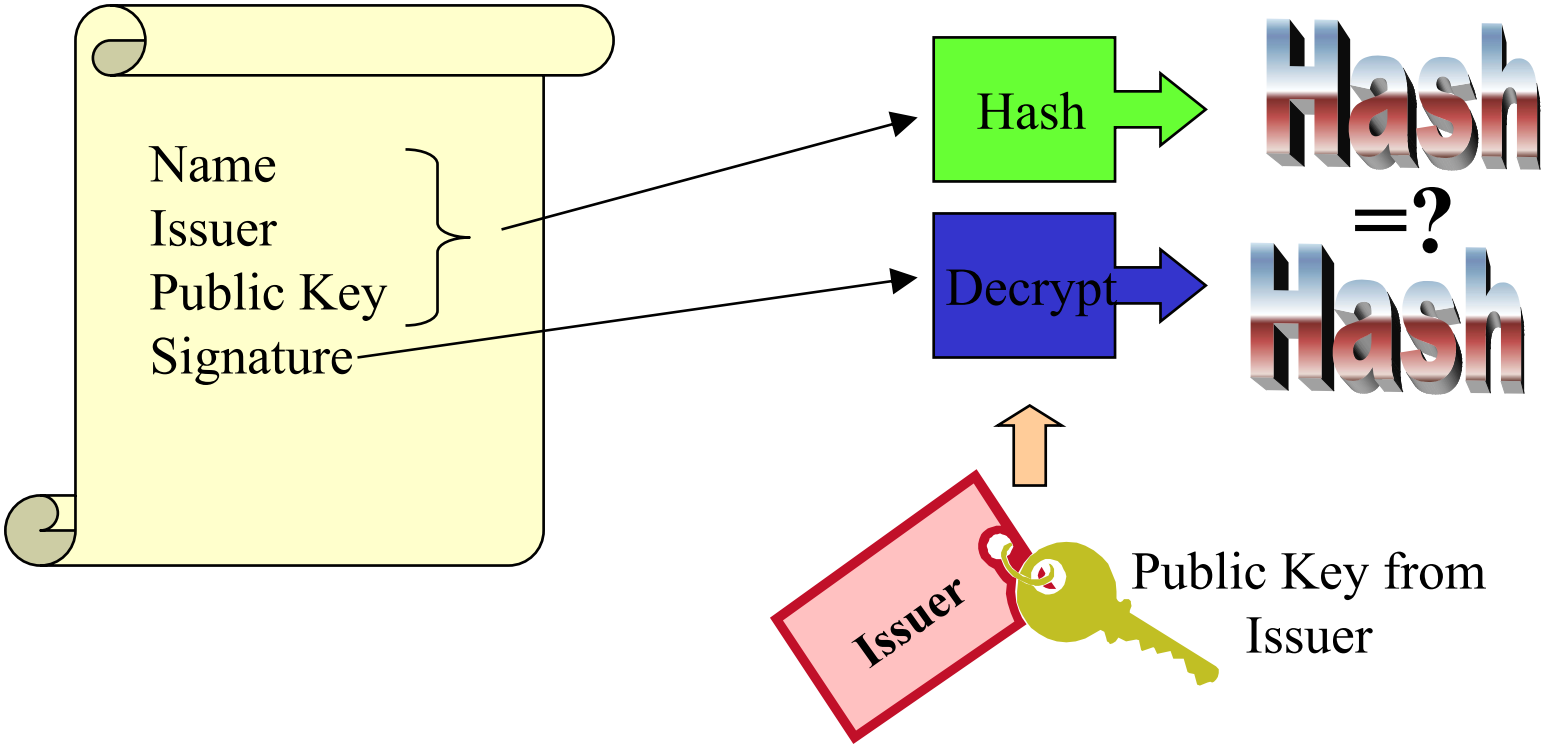


Mutual authentication

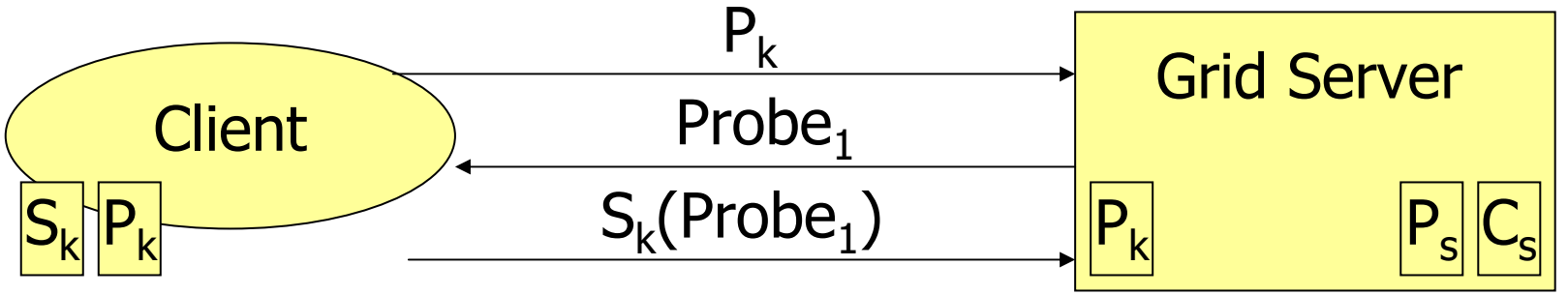


Certificates

- By checking the signature, The server can determine that the public key belongs to the client.

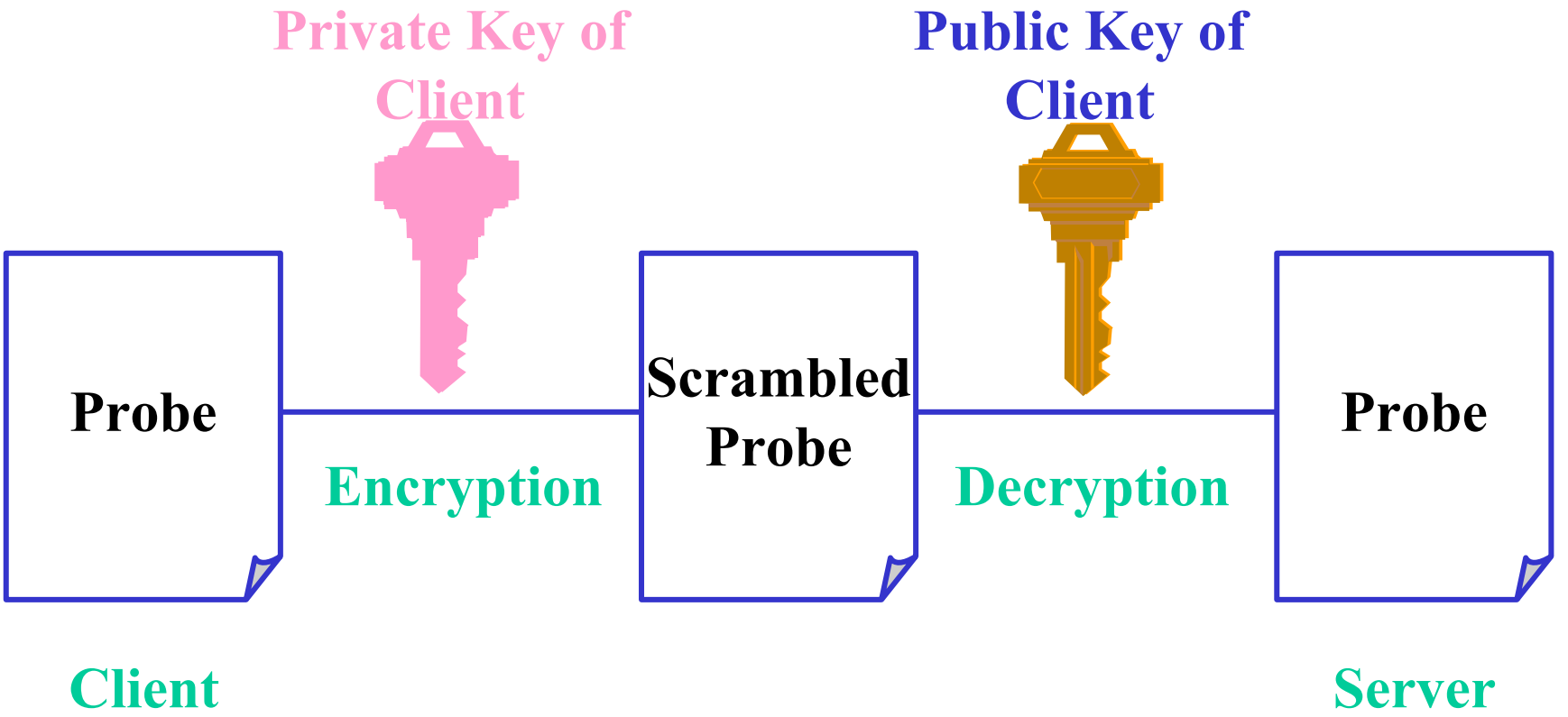


Mutual authentication

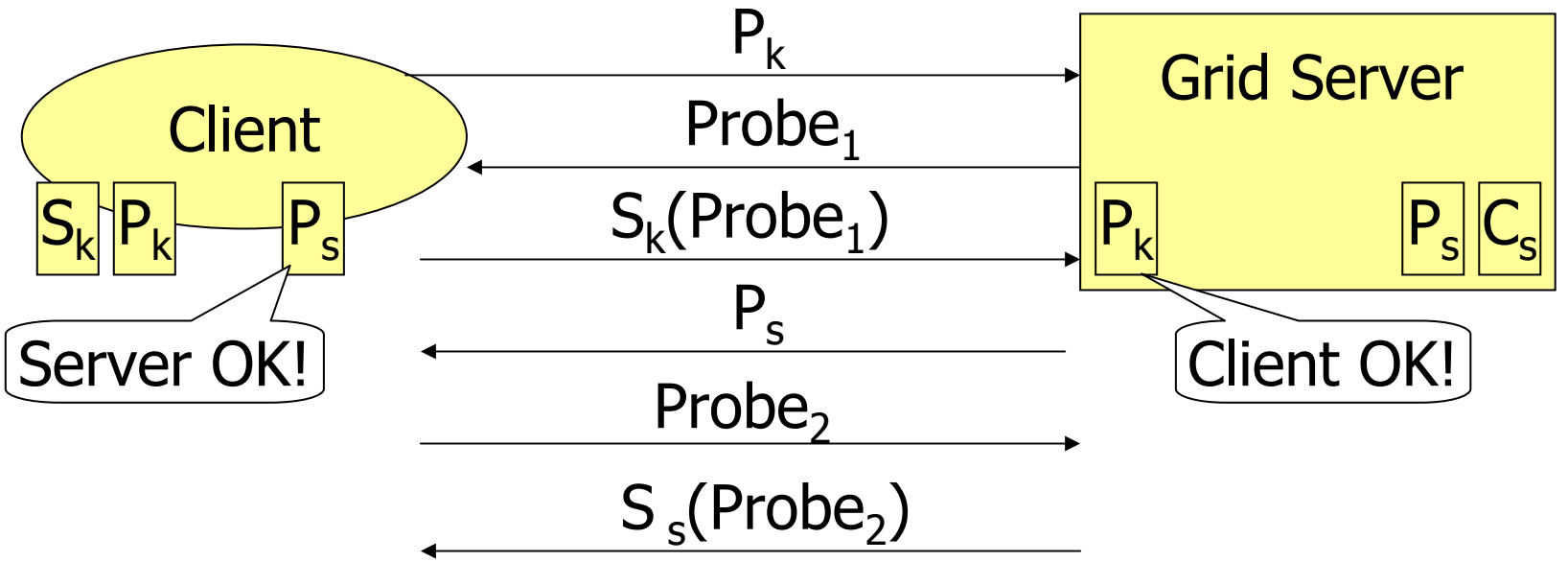


Identifying the sender

- Client can use his private key to sign the probe
- Server opens the sealed document using client's public key – this way, he is sure the sender is really the client



Mutual authentication



Globus Security Review

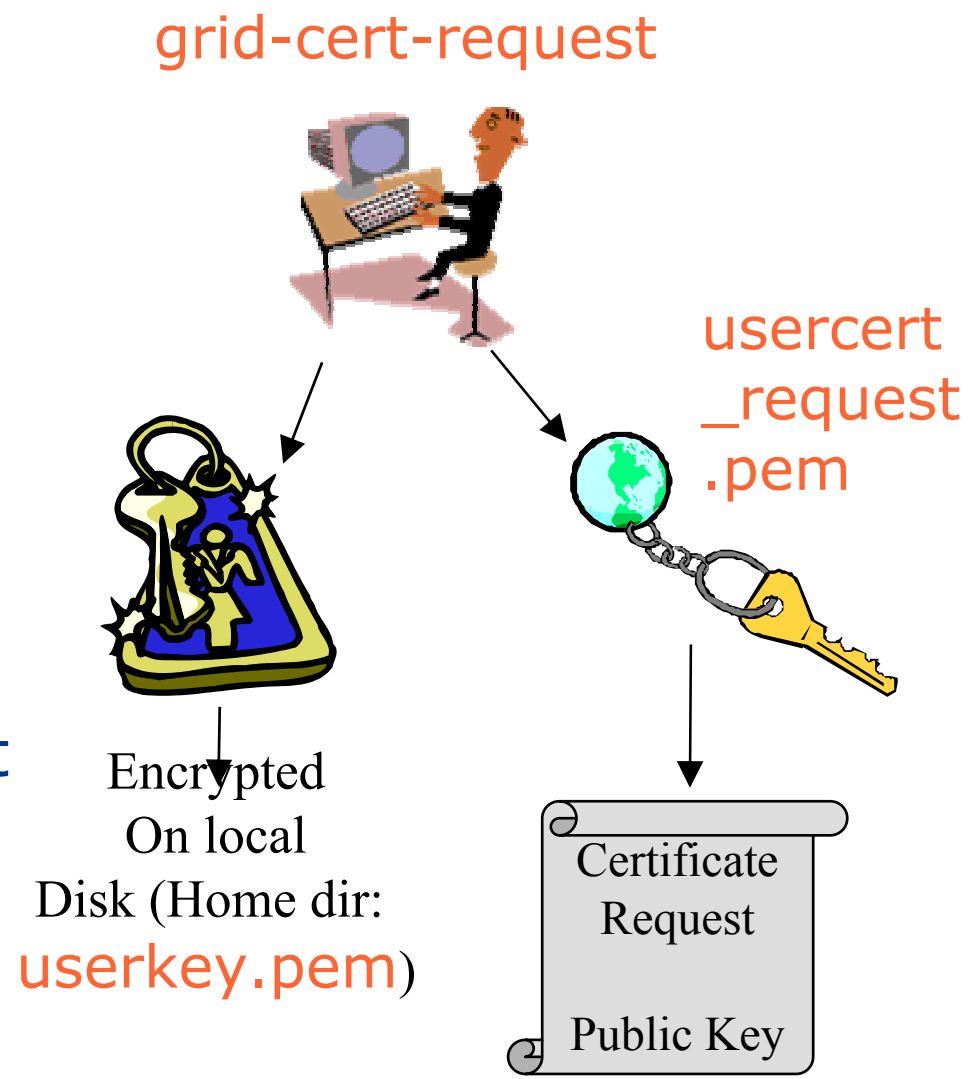
- GSI extends existing standard protocols & APIs
  - Based on standards: SSL/TLS, X.509, GSS-API
  - Extensions for single sign-on and delegation
- The Globus Toolkit provides:
  - Generic Security Services API (GSS-API) on GSI protocols
    - > **The GSS-API is the IETF standard for adding authentication, delegation, message integrity, and message confidentiality to applications.**
  - Various tools for credential management, login/logout, etc.

Obtaining a Certificate I.

- The program grid-cert-request is used to create a public/private key pair and unsigned certificate in ~/.globus/:
  - usercert_request.pem: Unsigned certificate file
  - userkey.pem: Encrypted private key file
    - > Must be readable **only** by the owner

Requesting a Certificate

- To request a certificate a user starts by generating a key pair
- The private key is stored encrypted with a pass phrase the user gives
- The public key is put into a certificate request



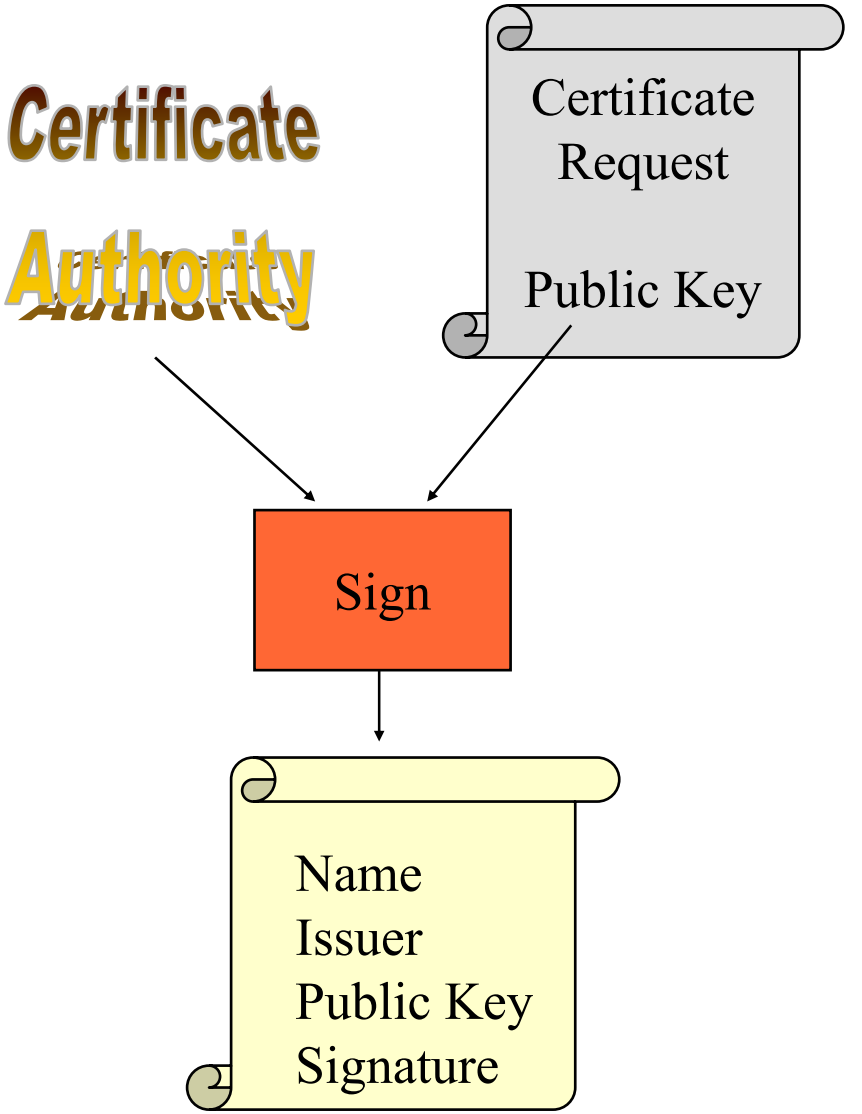
Obtaining a Certificate II.

- Mail usercert_request.pem to ca@globus.org
- Receive a Globus-signed certificate

  Place in ~/.globus/usercert.pem
- Other organizations use different approaches
  - NCSA, NPACI, NASA, etc. have their own CA

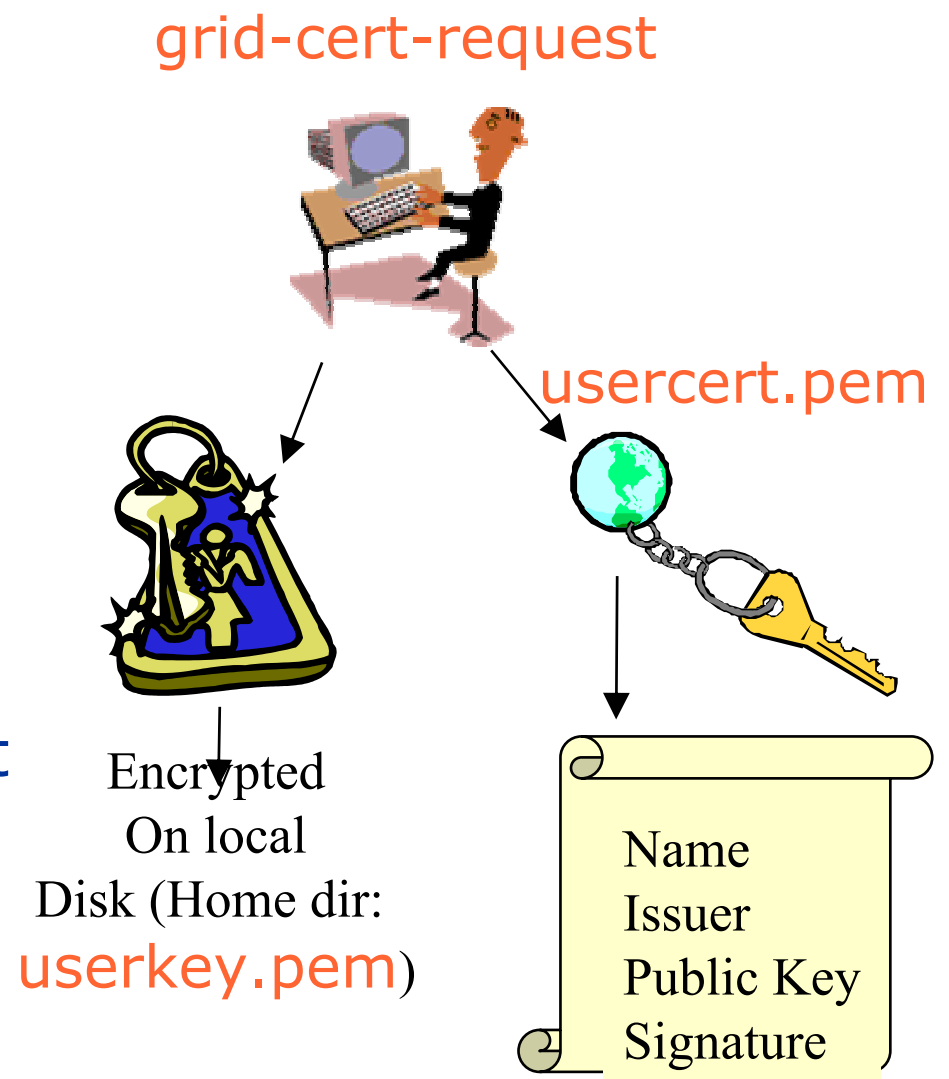
Certificate Issuance

- The CA then signs the certificate request and issues a certificate for the user



Requesting a Certificate

- To request a certificate a user starts by generating a key pair
- The private key is stored encrypted with a pass phrase the user gives
- The public key is put into a certificate request



Your New Certificate

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 28 (0x1c)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=US, O=Globus, CN=Globus Certification Authority
        Validity
            Not Before: Apr 22 19:21:50 1998 GMT
            Not After : Apr 22 19:21:50 1999 GMT
        Subject: C=US, O=Globus, O=NACI, OU=SDSC, CN=Richard Frost
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:bf:4c:9b:ae:51:e5:ad:ac:54:4f:12:52:3a:69:
                    <snip>
                    b4:e1:54:e7:87:57:b7:d0:61
                Exponent: 65537 (0x10001)
    Signature Algorithm: md5WithRSAEncryption
        59:86:6e:df:dd:94:5d:26:f5:23:c1:89:83:8e:3c:97:fc:d8:
        <snip>
        8d:cd:7c:7e:49:68:15:7e:5f:24:23:54:ca:a2:27:f1:35:17:
```

NTP is highly recommended

Certificate and Key Data

Sample usercert.pem:

```
-----BEGIN CERTIFICATE-----
MIICAzCCAWygAwIBAgIBCDANBgkqhkiG9w0BAQQFADBHMQswCQY
        <snip>
u5tX5R1m7LrBeI3dFMviJudlihloXfJ2BduIg7XOKk5g3JmgauK4
-----END CERTIFICATE-----
```

Sample userkey.pem:

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,1E924694DBA7D9D1
+W4FEPdn/oYntAJPw2tfmrGZ82FH611o1gtvjSKH79wdFxzKhnz474Ijo5Bl
        <snip>
et5QnJ6hAO4Bhya1XkWyKHTPs/2tIflKn0BNIIIYM+s=
-----END RSA PRIVATE KEY-----
```

Certificate Information

- To get cert information run grid-cert-info

  % grid-cert-info -subject

  /C=US/O=Globus/O=ANL/OU=MCS/CN=Ian Foster

- Options for printing cert information

  -all
  -subject
  -issuer
  -startdate
  -enddate
  -help

Autentikáció a Gridben III.

- Egyszeri beléptetés (Single Sign-On)
  - A felhasználó csak egyszer azonosítja magát, azután minden Grid szolgáltatást használhat
- Jogosultság-delegáció (credential delegation)
  - A felhasználó által indított munkafolyamatok a felhasználó nevében tevékenykedhetnek
- A titkos kulcsot nem akarjuk a munkafolyamatokra bízni
- Megoldás: Proxy tanúsítványok
  - A felhasználó által létrehozott kulcspárok
  - A nyilvános kulcsot a felhasználó a saját titkos kulcsával hitelesíti
  - Korlátozott ideig érvényesek és/vagy korlátozott jogosultságokkal bírnak

"Logging on" to the Grid

- To run programs, authenticate to Globus:

  % grid-proxy-init

  Enter PEM pass phrase: ******
- Creates a temporary, local, short-lived proxy credential for use by our computations
- Options for grid-proxy-init:

  -hours <lifetime of credential>

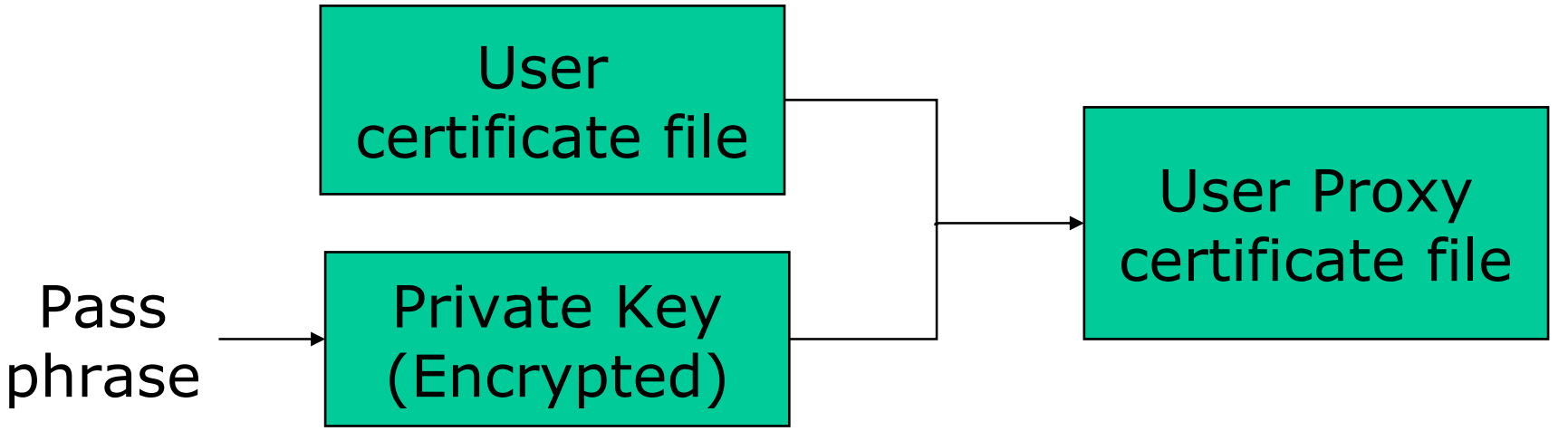
  -bits <length of key>

  -help

grid-proxy-init Details

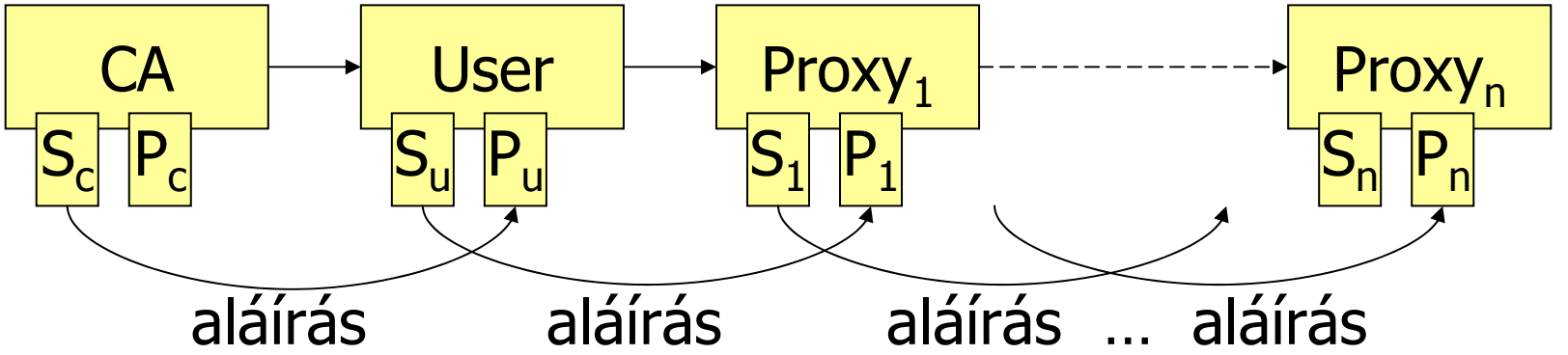
- grid-proxy-init creates the local proxy file.
- User enters pass phrase, which is used to decrypt private key.
- Private key is used to sign a proxy certificate with its own, new public/private key pair.
  - User's private key not exposed after proxy has been signed
- Proxy placed in /tmp, read-only by user
- NOTE: *No* network traffic!
- grid-proxy-info displays proxy details

Grid Sign-On With grid-proxy-init



Autentikáció a Gridben IV.

- A proxy tanúsítványok további proxyk létrehozására is alkalmasak (delegáció), így tanúsítvány-lánc hozható létre



- A proxy hitelességének ellenőrzéséhez az egész láncolatot be kell járni

Destroying Your Proxy (logout)

- To destroy your local proxy that was created by grid-proxy-init:

  % grid-proxy-destroy

- This does *NOT* destroy any proxies that were delegated from this proxy.
  - You cannot revoke a remote proxy
  - Usually create proxies with short lifetimes

Proxy Information

- To get proxy information run grid-proxy-info

  % grid-proxy-info -subject

  /C=US/O=Globus/O=ANL/OU=MCS/CN=Ian Foster

- Options for printing proxy information

  -subject -issuer
  -type -timeleft
  -strength -help

- Options for scripting proxy queries

  -exists -hours <lifetime of credential>
  -exists -bits <length of key>

  – Returns 0 status for true, 1 for false:

Important Files

- /etc/grid-security
  - hostcert.pem: certificate used by the server in mutual authentication
  - hostkey.pem: private key corresponding to the server's certificate (read-only by root)
  - grid-mapfile: maps grid subject names to local user accounts (really part of gatekeeper)
- /etc/grid-security/certificates
  - CA certificates: certs that are trusted when validating certs, and thus needn't be verified
  - ca-signing-policy.conf: defines the subject names that can be signed by each CA

Important Files

- $HOME/.globus
  - usercert.pem: User's certificate (subject name, public key, CA signature)
  - userkey.pem: User's private key (encrypted using the user's pass phrase)
- /tmp
  - Proxy file(s): Temporary file(s) containing unencrypted proxy private key and certificate (readable only by user's account)
    - > Same approach Kerberos uses for protecting tickets

Secure Services

- On most unix machines, inetd listens for incoming service connections and passes connections to daemons for processing.
- On Grid servers, the gatekeeper securely performs the same function for many services
  - It handles mutual authentication using files in /etc/grid-security
  - It maps to local users via the gridmap file

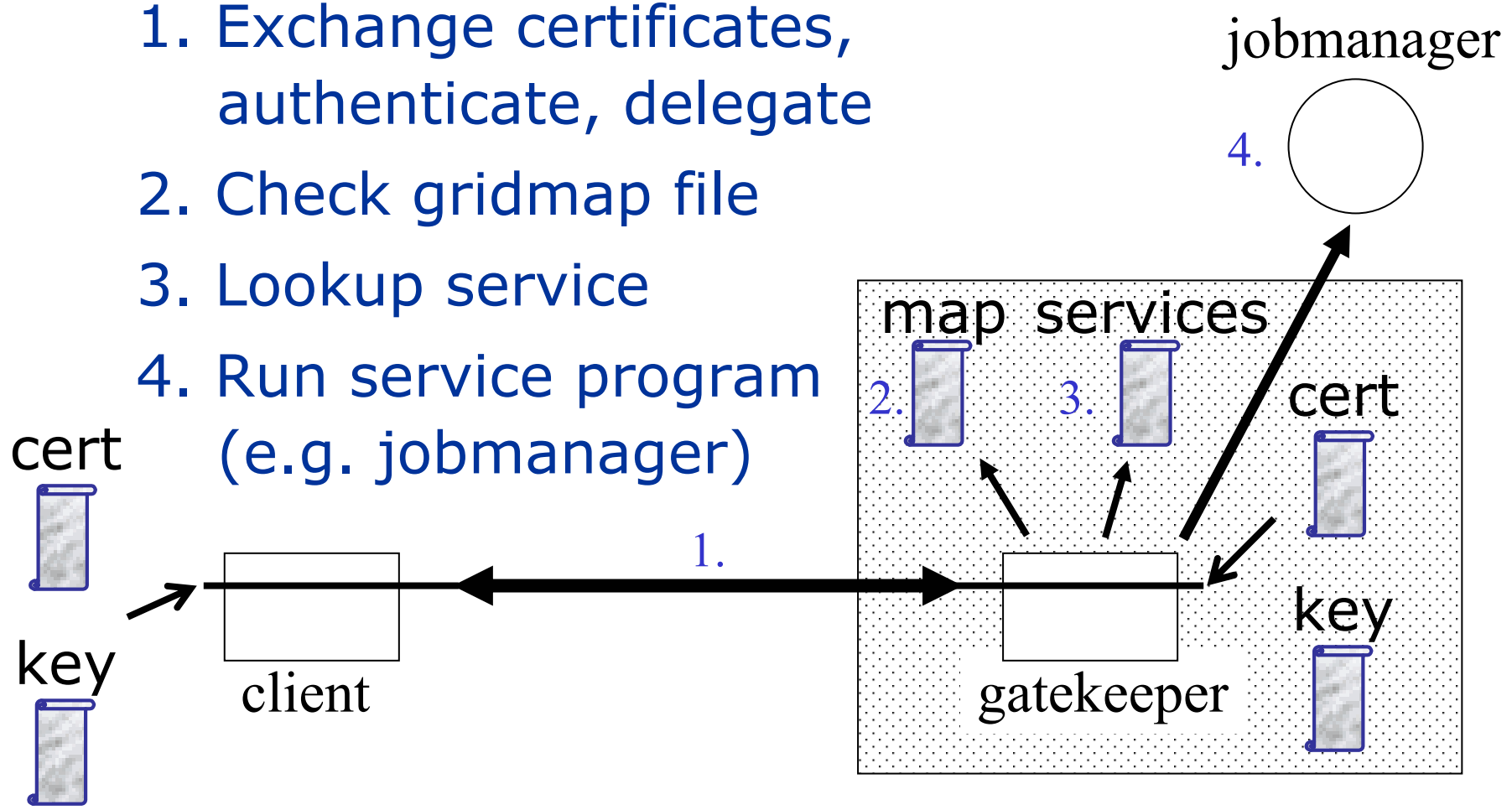
Sample Gridmap File

- Gridmap file maintained by Globus administrator
- Entry maps Grid-id into local user name(s)

```
# Distinguished name                                    Local
#                                                      username
"/C=US/O=Globus/O=NPACI/OU=SDSC/CN=Rich Gallup”          rpg
"/C=US/O=Globus/O=NPACI/OU=SDSC/CN=Richard Frost”        frost
"/C=US/O=Globus/O=USC/OU=ISI/CN=Carl Kesselman”          u14543
"/C=US/O=Globus/O=ANL/OU=MCS/CN=Ian Foster”              itf
```

Example
Secure Remote Startup

1. Exchange certificates, authenticate, delegate
2. Check gridmap file
3. Lookup service
4. Run service program (e.g. jobmanager)



Simple job submission

- globus-job-run provides a simple RSH compatible interface

```
% grid-proxy-init
  Enter PEM pass phrase: *****
% globus-job-run host program [args]
```

- Job submission will be covered in more detail later

Delegation

- Delegation = remote creation of a (second level) proxy credential
  - New key pair generated remotely on server
  - Proxy cert and public key sent to client
  - Clients signs proxy cert and returns it
  - Server (usually) puts proxy in /tmp
- Allows remote process to authenticate on behalf of the user
  - Remote process "impersonates" the user

Limited Proxy

- During delegation, the client can elect to delegate only a "limited proxy", rather than a "full" proxy
  - GRAM (job submission) client does this
- Each service decides whether it will allow authentication with a limited proxy
  - Job manager service requires a full proxy
  - GridFTP server allows either full or limited proxy to be used

Restricted Proxies

- A generalization of the simple limited proxies
  - Desirable to have fine-grained restrictions
  - Reduces exposure from compromised proxies
- Embed restriction policy in proxy cert
  - Policy is evaluated by resource upon proxy use
  - Reduces rights available to the proxy to a subset of those held by the user
    - A proxy no longer grants full impersonation rights
  - Extensible to support any policy language
- Will be in future version > GT 2.0

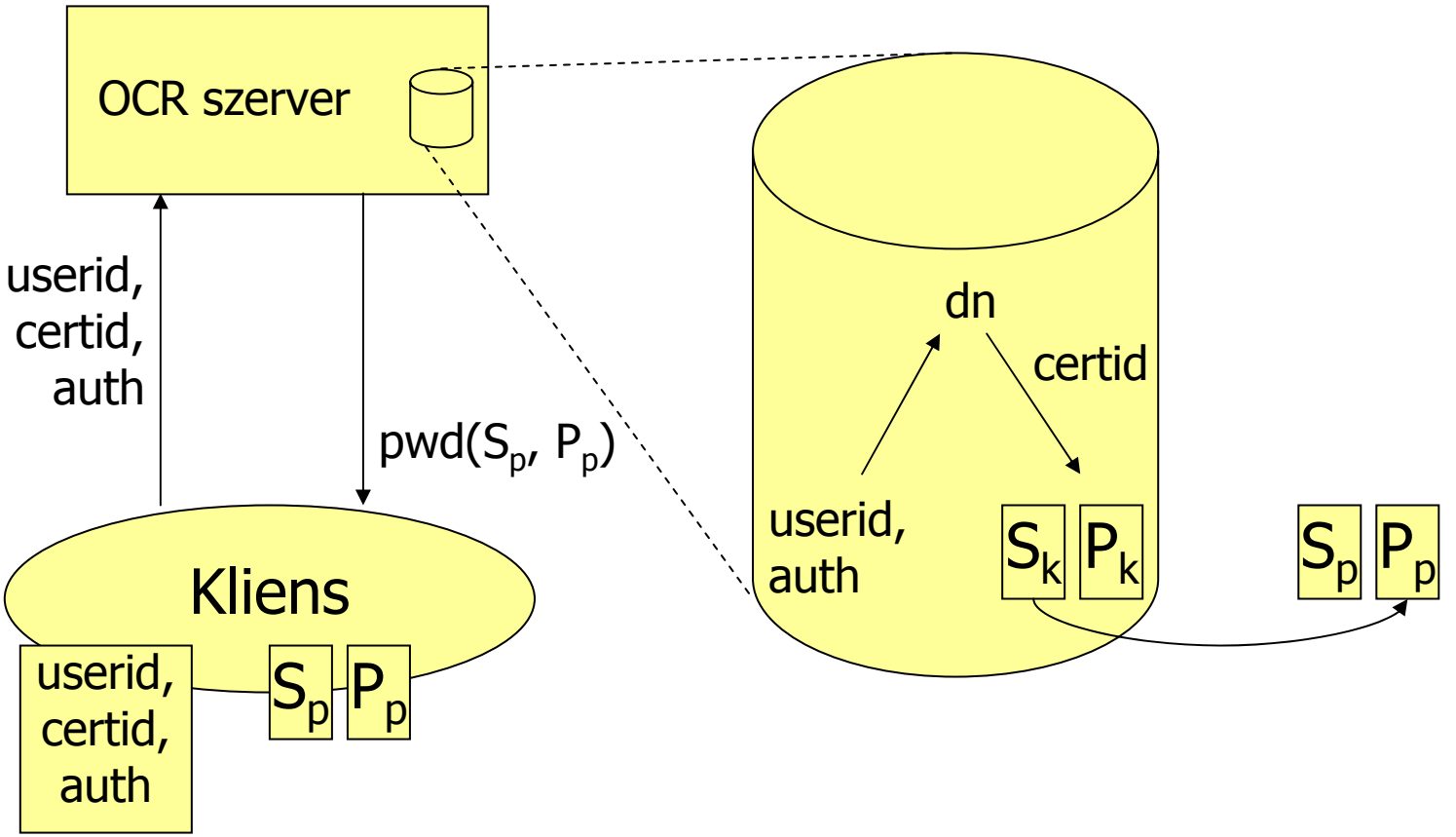
Elektronikus kulcskiosztó rendszer I.

- A proxy tanúsítvány létrehozásához a felhasználónak szüksége van a titkos kulcsára
- Általában a titkos kulcsot a felhasználó felügyeli
- Egy jelszóval védett fájl a home könyvtárban
- A kulcskezelés terhét a felhasználó viseli, aki esetleg nem tudja, vagy nem akarja megfelelő gondossággal kezelni kulcsait
  - Véletlen törlések, akaratlan kulcskiszivárogtatás
  - A különböző erőforrásokhoz más-más kulcspárok tartozhatnak

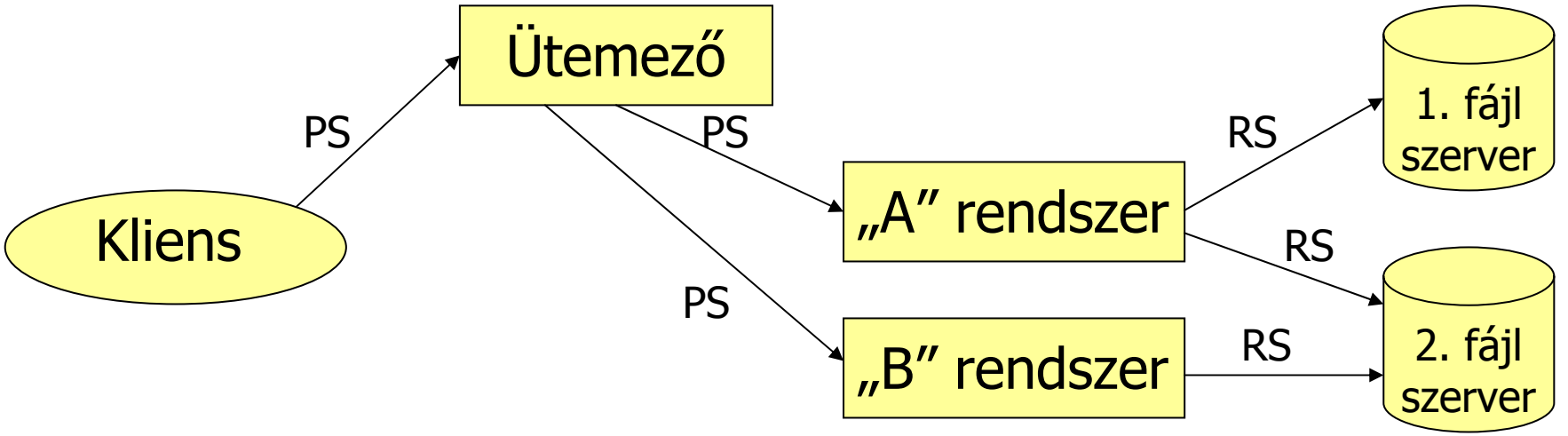
Elektronikus kulcskiosztó rendszer II.

- Az elektronikus kulcskiosztó rendszer (Online Credential Retrieval System, OCRS)
  - Központi adatbázisban tárolja a felhasználók kulcspárait
  - A felhasználó kérésére proxy tanúsítványt állít elő és küld vissza
- Az OCRS előnyei
  - Leegyszerűsíti a rendszeradminisztrátorok munkáját (több ezer felhasználós rendszerek is előfordulhatnak)
  - A felhasználót mentesíti a kulcskezelés feladata alól
  - Az egész rendszer biztonságát növeli
  - A váratlanul hosszú ideig futó munkafolyamatok a felhasználó közbeavatkozása nélkül is igényelhetnek kiterjesztett érvényességű proxy tanúsítványokat

Alapvető működés

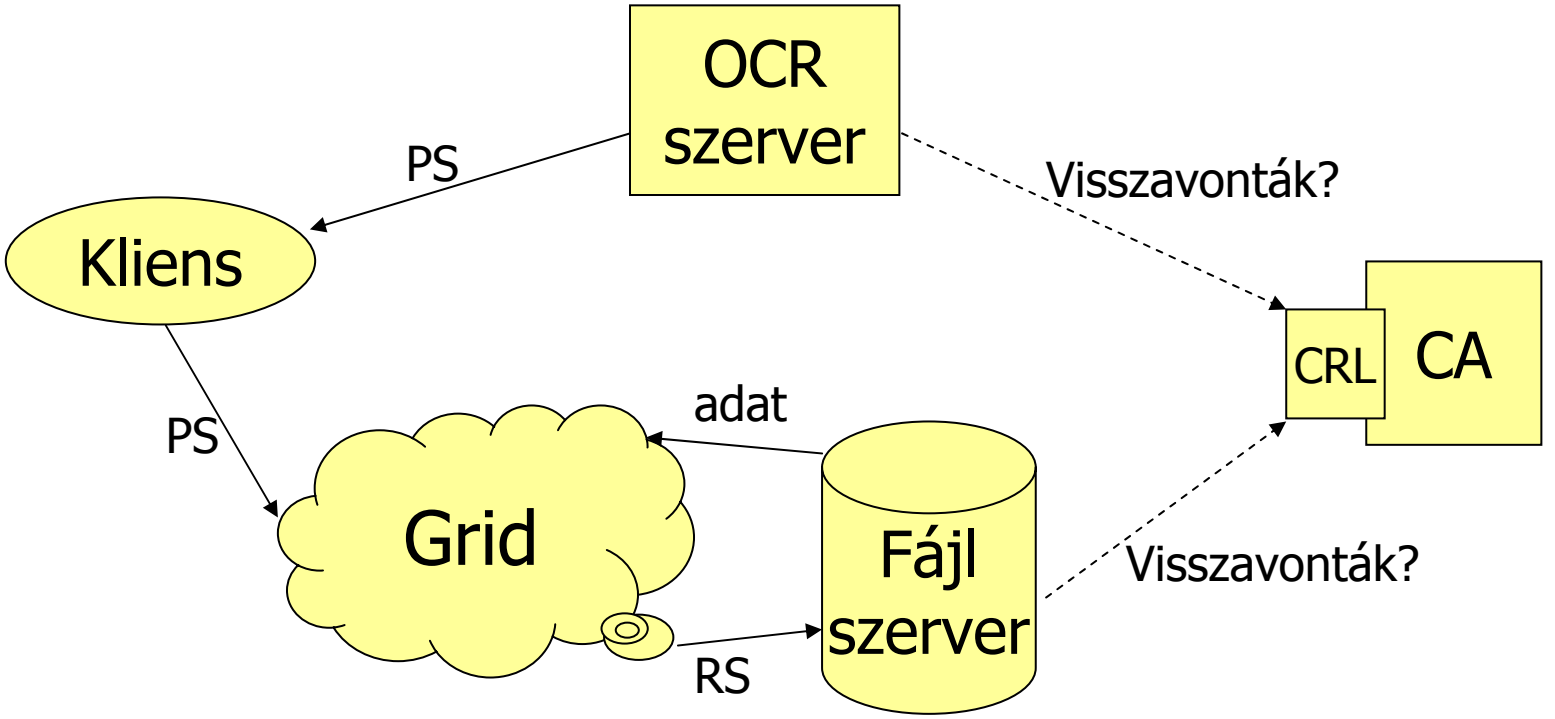


A proxy tanúsítványok felhasználása



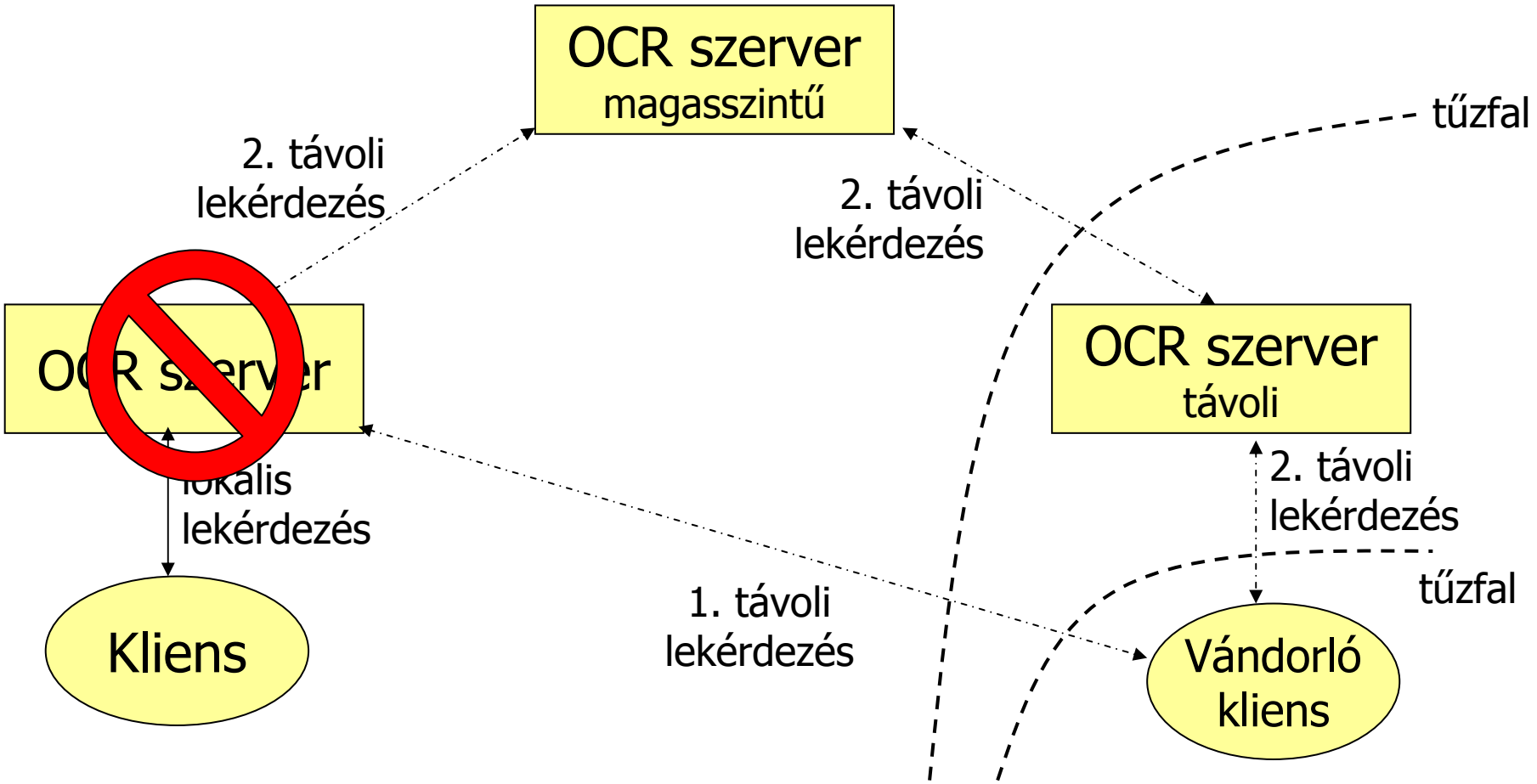
- PS: proxy tanúsítvány és titkos kulcs
- RS: korlátozott jogosultságú proxy tanúsítvány és titkos kulcs

Visszavont tanúsítványok kezelése



- CRL: Certificate Revocation List, Tanúsítvány-visszavonó lista

Adatbázisreplikák, vándorló kliensek



OCR implementáció I.

- Kódolatlan kulcsadatbázis
  - Az OCR szerver kódolatlanul tárolja a titkos kulcsokat
  - Lehetővé teszi a jelszótovábbítás nélküli autentikációt, illetve az alternatív autentikációs eljárásokat (OTP, Kerberos)
  - Az OCR számára dedikált gépet kell elkülöníteni
- Háttéradatbázis moduláris implementációja
  - Kis felhasználóbázis esetén az egyszerű szövegfájl is elég
  - Nagy terhelés: hasítótábla
  - Nagy terhelés, nagy felhasználóbázis: valódi adatbáziskezelő

OCR implementáció II.

- Kliens interfész
  - Adott lejárati idejű proxy tanúsítvány igénylése
  - Korlátozott jogosultságú proxy tanúsítvány igénylése
- Adminisztratív felület
  - Új felhasználó létrehozása, felhasználó letiltása és törlése
  - Autentikációs eljárás kiválasztása, jelszóbeállítás
  - Új tanúsítvány feltöltése, tanúsítvány letiltása és törlése
  - Automatikus kulcspárgenerálás, és továbbítás a helyi hitelesítő hatósághoz
- CRL gyorsítótár implementálása a hálózat terhelésének csökkentésére

Köszönöm a figyelmüket

További információ: www.lpds.sztaki.hu