



# ***Integrating PKI and Kerberos Authentication services***

Emmanuel Ormancey, Alberto Pace



# Authentication Methods

- ◆ **Two technologies for authentication**
  - ◆ **Kerberos and X.509 Certificates (PKI)**
- ◆ **Today at CERN**
  - ◆ **Kerberos is used in Windows Domains and AFS**
  - ◆ **PKI is used in all Grid related projects, with multiple certification authorities**
  - ◆ **Both technologies here to stay**
- ◆ **Multiple scenarios exist to integrate and interoperate the two technologies**



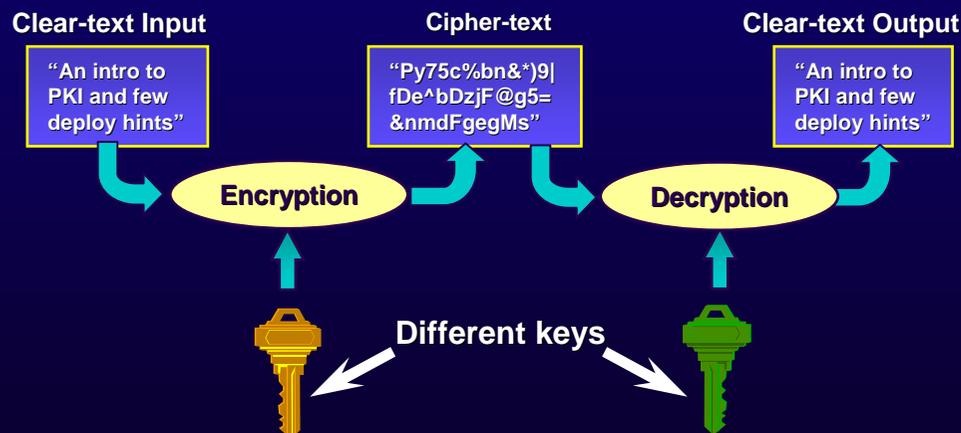
# Kerberos vs PKI ?

- ◆ Both technologies have weak and strong points
  - ◆ Distributed versus centralized management
  - ◆ Forwardable authentication
  - ◆ Offline authentication
- ◆ Technology is different
  - ◆ Asymmetric encryption with public/private key pairs versus symmetric encryption and shared secrets
- ◆ Some details follows ...



# PKI basics

- ◆ PKI provides, among other services, an authentication protocol relying on *asymmetric encryption*.
- ◆ One of the keys is kept private, the other is made public. Public keys are distributed using certificates which are digitally signed by trusted authorities





# PKI: Obtaining a Certificate



User generates  
a key pair



Public key is  
submitted to CA  
for certification

Certificate is  
sent to the user

User identity verified,  
Digital signature added,  
Certificate produced



Certification Server





# PKI: Authentication with Certificates



Alice



Encrypt using private key

Certificate is sent for authentication



Bob



Decrypt using public key in certificate and compare

Bob verifies the digital signature on the certificate

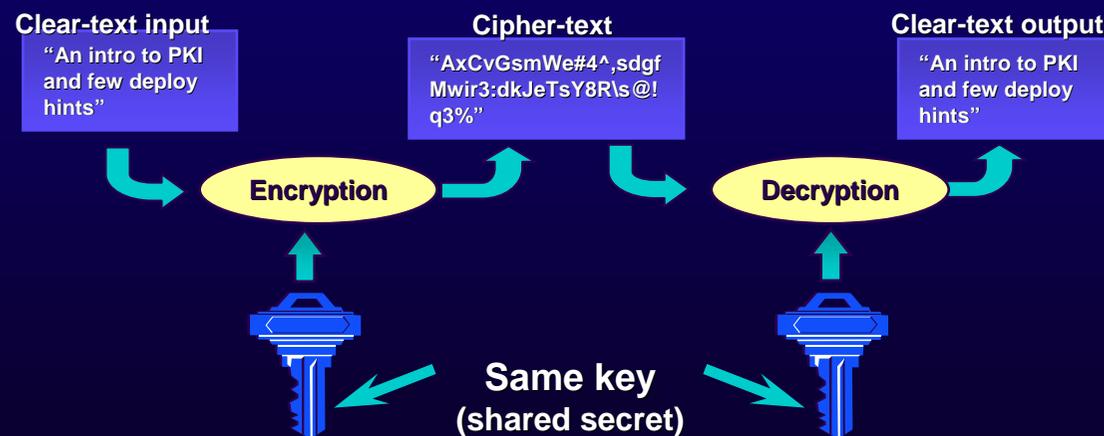
He can trust that the public key really belongs to Alice, but is it Alice standing in front of him ?

Bob challenges Alice to encrypt for him a random phrase he generated



# Kerberos Differences

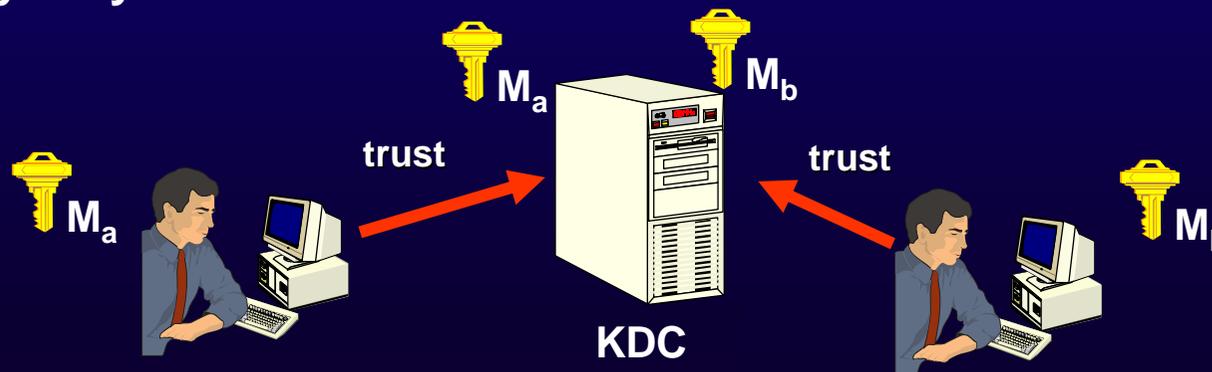
- ◆ Kerberos is an authentication protocol relying on *symmetrical* cryptographic algorithms that use the same key for encryption as for decryption
  - ◆ Different from PKI !





# Kerberos: Basic principles

- ◆ There is a trusted authority known as the Key Distribution Center (KDC) which is the keeper of secrets.
- ◆ Every user shares a secret password with the KDC
  - ◆ technically the KDC doesn't know the password but rather a one way hash, which is used as the basis for a cryptographic "master key".
- ◆ The secret master key is different for each user
  - ◆ As two users don't know each other master key they have no direct way of verifying each other's identity
- ◆ The essence of Kerberos is key distribution. The job of the KDC is to distribute a unique session key to each pair of users (security principals) that want to establish a secure channel.
  - ◆ Using symmetric encryption
- ◆ Everybody has to trust the KDC





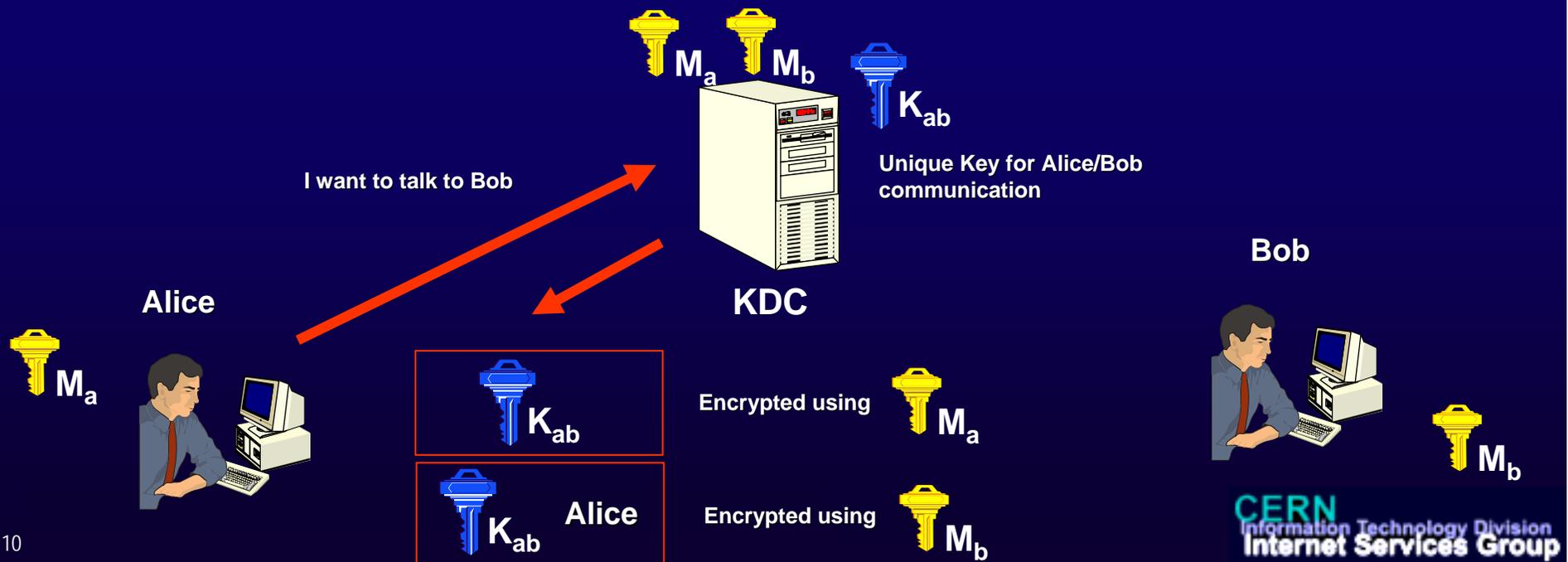
# Breakthrough of a (simplified) Kerberos session

- ◆ Alice wants to communicate with Bob
  - ◆ **bob could be a server or a service**
- ◆ Alice can communicate securely with the KDC, using symmetric encryption and the shared secret (Master Key)
- ◆ Alice tells the KDC that she wants to communicate with Bob (known to the KDC)



## (simplified) Kerberos session 2

- ◆ The KDC generates a unique random cryptographic key for Alice and Bob to use (call this  $K_{ab}$ )
- ◆ He sends back two copies of  $K_{ab}$  back to Alice.
  - ◆ The first copy is for her to use, and is sent to her along with some other information in a data structure that is encrypted using Alice's master key.
  - ◆ The second copy of  $K_{ab}$  is packaged along with Alice's name in a data structure encrypted with Bob's master key. This is known as a "ticket".





# What is the ticket ?

- ◆ The ticket is effectively a message to Bob that only BOB can decrypt
  - ◆ "This is your KDC. Alice wants to talk to you, and here's a session key that I've created for you and Alice to use. Besides me, only you and Alice could possibly know the value of  $K_{ab}$ , since I've encrypted it with your respective master keys. If your peer can prove knowledge of this key, then you can safely assume it is Alice."



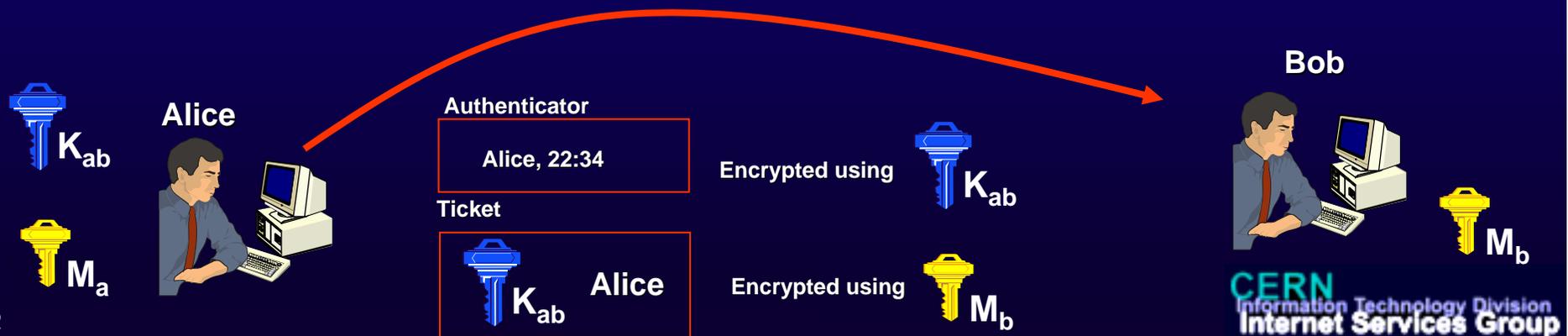
Encrypted using





# Kerberos authentication

- ◆ Alice must send the ticket to Bob
  - ◆ with proof that she knows  $K_{ab}$
  - ◆ and she must do it in a way that allows Bob to detect replays from attackers listening on the network where Alice, Bob, and the KDC are conversing.
- ◆ The ticket is sent to Bob, with an authenticator (her name and the current time, all encrypted with the session key  $K_{ab}$ )
- ◆ Bob takes the ticket, decrypts it, and pulls  $K_{ab}$  out. Then decrypts the authenticator using  $K_{ab}$ , and compares the name in the authenticator with the name in the ticket
  - ◆ If the time is correct, this could provide evidence that the authenticator was indeed encrypted with  $K_{ab}$





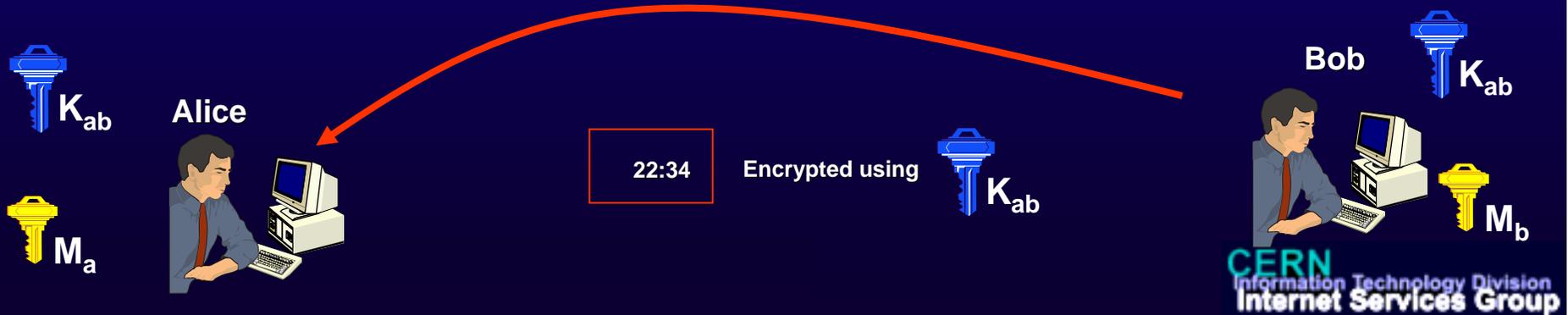
# Kerberos authentication

- ◆ It time is incorrect, bob reject the request
  - ◆ with a hint of what his time is (Bob time isn't a secret)
- ◆ If the time is correct ...
  - ◆ ... it's probable that the authenticator came from Alice, but another person might have been watching network traffic and might now be replaying an earlier attempt. However, if Bob has recorded the times of authenticators received from Alice during the past “five minutes”, he can defeat replay attempts. If this authenticator yields a time later than the time of the last authenticator from Alice, then this message must be from Alice
  - ◆ This is why time synchronization is essential in kerberos and all KDC provides also time synchronization services
- ◆ You can see this as a “challenge” on the knowledge of the shared secret ( $K_{ab}$ ):
  - ◆ “prove that you know  $K_{ab}$  by encrypting the current time for me”



# Mutual authentication

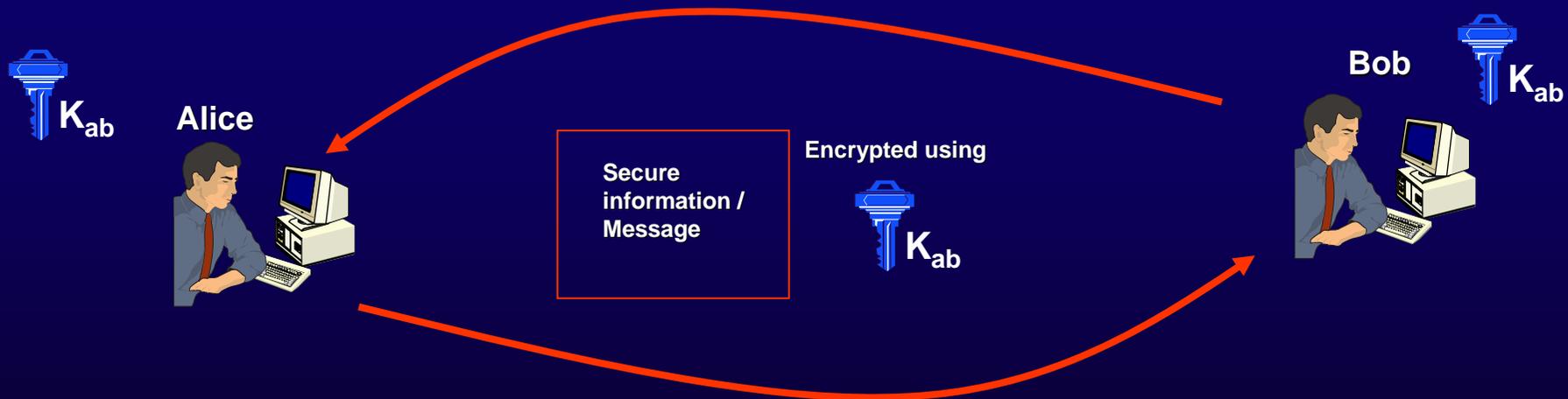
- ◆ Alice has proved her identity to Bob
- ◆ Now Alice wants Bob to prove his identity as well
  - ◆ she indicates this in her request to him via a flag.
- ◆ After Bob has authenticated Alice, he takes the timestamp she sent, encrypts it with  $K_{ab}$ , and sends it back to Alice.
- ◆ Alice decrypts this and verifies that it's the timestamp she originally sent to Bob
  - ◆ She has authenticated Bob because only Bob could have decrypted the Authenticator she sent
  - ◆ Bob sends just a piece of the information in order to demonstrate that he was able to decrypt the authenticator and manipulate the information inside. He chooses the time because that is the one piece of information that is unique in Alice's message to him





# Kerberos Secure Communication

- ◆ Alice and Bob share now a unique secret  $K_{ab}$  that they use to communicate





# Real life is more complicated

- ◆ Real Kerberos includes several extra steps for additional security
- ◆ When Alice first logs in, she actually asks the KDC for what is called a "ticket granting ticket", or TGT.
- ◆ The TGT contains the session key ( $K_{ak}$ ) to be used by Alice in her communications with the KDC throughout the day.
  - ◆ This explains why when the TGT expires you have to renew it
- ◆ So when Alice requests a ticket for Bob, she actually sends to the KDC her TGT plus an authenticator with her request.
- ◆ The KDC then sends back the Alice/Bob session key  $K_{ab}$  encrypted with  $K_{ak}$ 
  - ◆ as opposed to using Alice's master key as described earlier
- ◆ See various Kerberos references for details



# **CERN Certification authority, PKI and Kerberos integration**



# Authentication Services at CERN

- ◆ **For Kerberos:**
  - ◆ **Two KDC in production, one for Windows computers (cern.ch domain) one for AFS (cern.ch cell)**
  - ◆ **Account and passwords planned to be synchronized**
- ◆ **For the grid**
  - ◆ **CERN Certification authority**
    - ◆ <http://cern.ch/service-grid-ca>
- ◆ **Plan for 2006 / 2007**
  - ◆ **Migrate to a new certification authority integrated with the kerberos services**



# New CERN Certification authority

- ◆ Aim to issue certificates
  - ◆ Recognized by the entire grid community
  - ◆ Valid to obtain kerberos ticket automatically
- ◆ Separate Root CA and Issuing CA
- ◆ Offline Root CA:
  - ◆ Run on Virtual PC, Server image on removable disks
  - ◆ Root trusted by default inside CERN.
- ◆ Online Issuing CA
  - ◆ Issues all certificate, online
  - ◆ Web site <http://cern.ch/ca>



# CA services planned

- ◆ **Issuing User certificates**
  - ◆ 'software' client certificates
  - ◆ **Certifies the identity of a persons**
    - ◆ Current identification based on knowledge of Kerberos account password
- ◆ **Issuing Host certificates**
  - ◆ **Typically for all web servers requiring, for example, https services**
  - ◆ **Can certify any host in the cern.ch domain from the CERN network database**
  - ◆ **Manual procedure for host outside the cern.ch domain**
- ◆ **Service certificates foreseen**
- ◆ **Issuing SmartCards**
  - ◆ **Certificate and private key in a HW token**
- ◆ **Allow users to map existing certificates issued by trusted CA (for example existing Grid certificates) to their account.**



# DEMO: User Certificate Request

The screenshot shows the CERN Certification Authority website. The main navigation bar includes 'CERN Home', 'IT Department', 'IT/IS Group', 'Mail Services', 'Web Services', and 'Win Services'. The page title is 'CERN IT/IS Certificates Services' and the user is identified as 'ormancey'. The 'Certificates administration' section is highlighted with a red circle and contains the following links:

- Request user certificate using Internet Explorer
- Request user certificate using Mozilla browser
- Request user certificate manually
- Revoke certificate

Other sections visible include 'Trust CERN Certification authority', 'Certificate mappings', 'Advanced users', 'Host Certificates', and 'Administration'.

Internet Explorer or Mozilla browsers can handle automatically certificate request.

**Request a user certificate**

Press **Submit** button to proceed with your request. Internet Explorer will create the request for you, some warning popups will appear, please click on **Yes** button.

A manual procedure with OpenSSL is also provided.

**Request user certificate**

To submit a saved request to the CA, paste a base-64-encoded CMC or PKCS #10 certificate request or PKCS #7 renewal request generated by an external source (such as an OpenSSL) in the Saved Request box.

**Sample command line with OpenSSL:**

- Run: `openssl req -new -out myrequest.csr`
- When asked, type in a PEM password and don't forget it !
- No need to fill other information, simply press enter for each field.
- 2 files will be created, `newcert.cer` and `myrequest.csr`, do not delete them !
- Copy/Paste the content of `myrequest.csr` in the field below and press Submit.

**Base-64-encoded certificate request (CMC or PKCS #10 or PKCS #7):**



# DEMO: Host Certificates

- ◆ Users can request Host certificates for CERN Hosts they manage, and any non-CERN host.

CERN Home | IT Department | IT/IS Group

**CERN Certification Authority**

CERN IT/IS Certificates Services

Home

**Certificates administration**

- Manage user certificates
  - Request user certificate using Internet Explorer
  - Request user certificate using Mozilla browser
  - Request user certificate manually
  - Revoke certificate
- Trust CERN Certification authority
  - On a CERN Domain managed Windows machine, the CERN Root Certificate is trusted, so any CERN Certificate will be verified correctly, no specific action is required.
  - On any other platform, the CERN Root Certificate needs to be trusted manually to allow CERN Certificate verification. To do this, install the Root Certificate using one of the following methods.
    - Install Root certificate using Internet Explorer
    - Install Root certificate using Mozilla browser
    - Install Root certificate using Safari browser
- Certificate mappings
  - Map an existing Certificate to your account
- Advanced users
  - [Show advanced options]
- Host Certificates**
  - Manage Host Certificates
- Administration
  - Internals

## Manage Host Certificates

**Current Host Certificates**

Hostname	Certificate Date
pcwebc03.cern.ch	Tuesday, December 31, 2002 2:15 PM

**Request Host Certificates**

**CERN Hosts managed by account ormancey**

- [Select] pcwebc03.cern.ch
- [Select] pcitis18.cern.ch
- [Select] pcitis35.cern.ch
- [Select] pcitis24.cern.ch
- [Select] pcitis61.cern.ch
- [Select] pcitis56.cern.ch

**Non-CERN host**

Type in the fully qualified domain name for the host (host.domain.ext) :



# DEMO: Certificate mapping to Existing Account

- ◆ Users can map an existing certificate to their Kerberos account for authentication
  - ◆ Typically for owners of Grid certificates not issued by the CERN CA

The screenshot shows the CERN Certification Authority website interface. The left sidebar contains a menu with 'Certificate mappings' circled in red. A red arrow points from this menu item to the main content area, which is titled 'Map an existing certificate to your NICE Account'. The main content area includes sections for 'Current alternate certificate mappings', 'Why map an existing Certificate', and 'Upload your public key'.

**Map an existing certificate to your NICE Account**

**Current alternate certificate mappings**

Issuer: C=ZA,O=Thawte Consulting (Pty) Ltd.,CN=Thawte Personal Freemail Issuing CA  
Subject: CN=Thawte Freemail Member,E=emmanuel.ormancey@cern.ch

Issuer: O=Root CA,OU=http://www.cacert.org,CN=CA Cert Signing Authority,E=support@cacert.org  
Subject: CN=CACert WoT User,E=emmanuel.ormancey@cern.ch

**Why map an existing Certificate**

If you already have a Certificate issued by another authority than CERN, and you would like to use it to authenticify to CERN Website, you need to map this certificate to your account.

To achieve this, you need to upload your public key in DER or Base64 format.  
[Click here if you need more help on how to export the public key.](#)

Currently only Certificate from these authorities are supported:

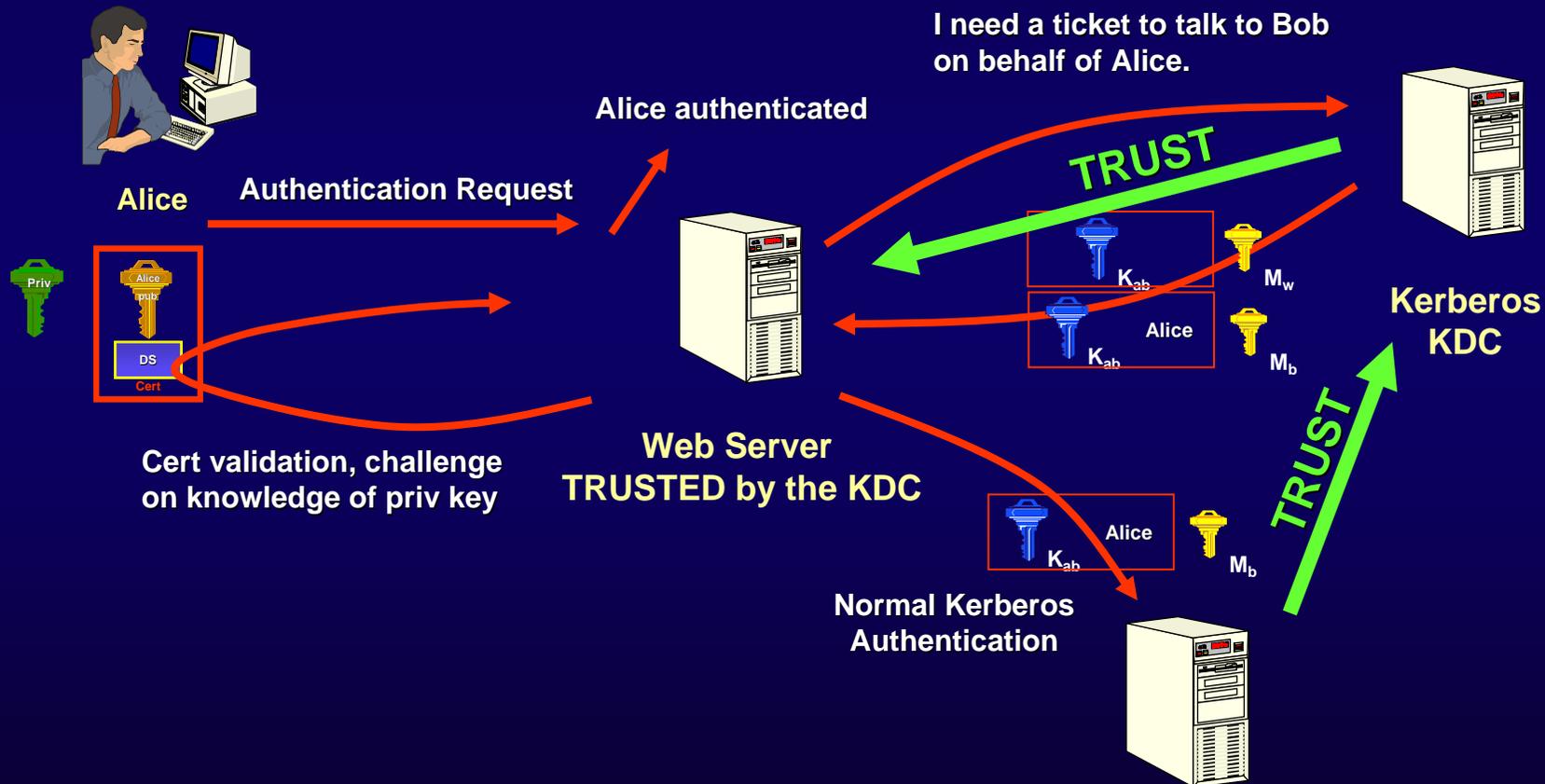
- CA Cert Signing Authority (<http://www.cacert.org>)
- Thawte Personal Freemail Issuing CA (<http://www.thawte.com>)

**Upload your public key**

Upload the DER or Base64 format file containing the public key of your certificate:



# PKI / Kerberos integration



See: Kerberos Constrained Delegation



# Roadmap for 2006

- ◆ Obtain accreditation from the European Grid Policy Management Authority ([www.eugridpma.org](http://www.eugridpma.org))
  - ◆ Obtain approval of the new Certificate Policy and Certification Practice (CP/CPS)
    - ◆ See <http://www.eugridpma.org/members/>
  - ◆ From offline issuing CA to online issuing CA with FIPS Hardware module
    - ◆ <http://www.eugridpma.org/guidelines/IGTF-AP-classic-20050930-4-0.pdf>
- ◆ Verify interoperability with Windows and Linux Desktops
  - ◆ Desktop login requires Active Directory path to match Certificate Distinguished Name
  - ◆ Alternate user mapping possible
  - ◆ Usage of Smartcard on linux requires further investigation



# Certificate usage, Interoperability

- ◆ **Once a certificate is installed in the client computer**
  - ◆ **Can authenticate to CERN Websites (Win, Web, Mail, Terminal services, etc...)**
    - ◆ Not all CERN web sites yet, but planned
    - ◆ Best example of PKI / Kerberos interoperability
  - ◆ **Can participate in any grid activity, worldwide**
    - ◆ Certificate recognized worldwide within the grid community
  - ◆ **Secure e-mail possible**
  - ◆ **Provide a common authentication interface for CERN services: sort of Single Sign On**
- ◆ **Medium to long term:**
  - ◆ **Have the CERN certificates trusted worldwide, not only within the grid community**
  - ◆ **Support Windows and Linux desktop authentication using Smartcard certificates.**
  - ◆ **Combine together SmartCards and CERN Access cards.**



# Web Authentication example

## Opening a website

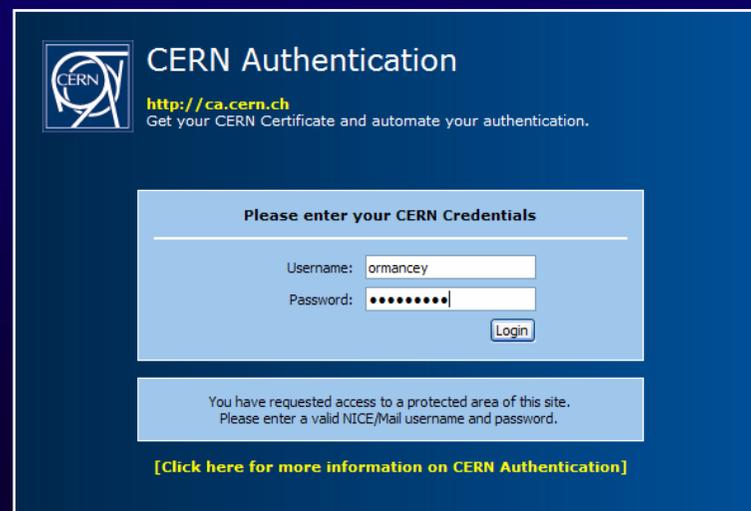
If several client certificates matching server requirements are found, browser asks to choose.



→ Certificate authentication complete. →



Cancelled or no certificate installed





# Technology not platform specific

**User Identification Request**

This site has requested that you identify yourself with a certificate:  
websvc02.cern.ch  
Organization: "CERN"  
Issued Under: "RSA Data Security, Inc."

**Choose a certificate to present as identification:**

ormancey CaCert [00:9B:AE]

Details of selected certificate:

Issued to: E=emmanuel.ormancey@cern.ch,CN=CAcert User Cert  
Serial Number: 00:9B:AE  
Valid from 07/07/2004 11:06:09 to 07/07/2005 11:06:09  
Purposes: Client,Server,Sign,Encrypt  
Issued by: E=support@cacert.org,CN=CA Cert Signing Authority,OU=http://www.cacert.org,O=Root CA  
Stored in: Software Security Device

OK Cancel Help

Client Certificate - Mozilla

File Edit View Go Bookmarks Tools Wind

Back Forward Reload Stop https://web Search Print

Home Bookmarks Red Hat Network St

CERN Home IT Department IT/ Mail Services Web Services Win Services

WinServices

CERN IS WinServices User: **ormancey** (Certificate authentication [details])

Home

**User account**

Login: ormancey  
Name: Emmanuel Ormancey  
Email: Emmanuel.Ormancey@cern.ch  
Phone: 71057 (mobile: 160821)  
Building: Bld. 31 Room R-017

**Client certificate**

Certificate: CN=CAcert User Cert, E=emmanuel.ormancey@cern.ch  
Issued by: O=Root CA, OU=http://www.cacert.org, CN=CA Cert Signing Authority, E=support@cacert.org  
Certificate is valid until: 7/7/2005 11:06:09 AM  
Authenticated user: CERN\ormancey (authentication type: SSL/PCT)

**User certificate mappings**

Issuer: O=Root CA,OU=http://www.cacert.org,CN=CA Cert Signing Authority,E=support@cacert.org  
Subject: CN=CAcert User Cert,E=emmanuel.ormancey@cern.ch

Issuer: C=ZA,O=Thawte Consulting (Pty) Ltd.,CN=Thawte Personal Freemail Issuing CA  
Subject: CN=Thawte Freemail Member,E=emmanuel.ormancey@cern.ch

WinServices Site CERN IT/IS Group - 2005

Done



# Example: Email signing

## ◆ In Outlook:

Absence - Message (Plain Text)

From: Rafal Otto Sent: Thu 4/28/2005 12:50 PM  
To: it-dep-is (Members of the group IT/IS)  
Cc:  
Subject: Absence

Signed By: Rafal.Otto@cern.ch

I'll be off this afternoon. My GSM will be on.

Cheers,  
Rafal

Untitled Message

File Edit View Insert Format Tools Table Window Help

Send [Icons] Options... HTML [Encrypt Message] [Digitally Sign]

To...  
Cc...  
Subject:

Digital Signature: Valid

Subject: Absence  
From: Rafal Otto

The digital signature on this message is Valid and Trusted.

For more information about the certificate used to digitally sign the message, click Details.

Warn me about errors in digitally signed e-mail before message opens.

Details...  
Close

Message Security Properties

Subject: Absence

Messages may contain encryption and digital signature layers. Each digital signature layer may contain multiple signatures.

**Security Layers**  
Select a layer below to view its description.

- ✓ Subject: Absence
  - ✓ Digital Signature Layer
    - ✓ Signer: Rafal.Otto@cern.ch

Description:  
OK: Signed message.

Click any of the following buttons to view more information about or make changes to the selected layer:

Edit Trust... View Details... Trust Certificate Authority...

Warn me about errors in digitally signed e-mail. Close



# Managing Certificates

- ◆ Software certificates expire and must be renewed
  - ◆ Typically once a year
  - ◆ Renewing a certificate is more complicated than a password change
- ◆ Looking towards automating request, distribution and installation of Client certificates
  - ◆ For PCs member of a Windows domain, the CERN certificate can be pushed to the client as a domain policy
  - ◆ Its renewal can be handled automatically (allowing short validity periods)
  - ◆ Users do not need to understand, be aware, be informed. 100 % transparent.
- ◆ Similar automation levels exist for Linux and Mac OS systems, but require the computers to be centrally managed
- ◆ Otherwise, Smartcards are a possible solution
  - ◆ Much easier for the user to understand
  - ◆ Longer certificate validity



## CERN Access Card

- ◆ **Use the same SmartCard for:**
  - ◆ **Windows desktop (and laptop)**
    - ◆ Logon and Browser authentication
  - ◆ **Linux desktop**
    - ◆ Browser authentication
  - ◆ **Mac OS X desktop**
    - ◆ Browser authentication
  - ◆ **Remote windows**
    - ◆ Windows Terminal Services
  - ◆ **Remote Linux**
    - ◆ Putty (to be defined, possible with OpenSC)
    - ◆ OpenSSH (to be defined, possible with OpenSC)
    - ◆ Exceed (to be confirmed)





# Conclusion

- ◆ **CERN is improving its Certificate Authority service to**
  - ◆ **issue certificates useable within the grid community**
  - ◆ **Further automate certificate issuing procedures**
  - ◆ **Automatically map Certificates to Kerberos accounts (when possible)**
- ◆ **In addition, Certificates issued by other trusted CA can be mapped to Kerberos accounts**
- ◆ **This should provide a good PKI/Kerberos interoperability**