Subject: **Authentication on the DataGrid Test Bed**
Author: **David Groep**
Information: **This document is tutorial for EDG and LCG authentication and registration**

1. AUTHENTICATION

1.1. GETTING A CERTIFICATE

1.1.1. What is a certificate?

While you are using computer systems that are scattered all over the world, the administrators of all those machines will want to know who is using their machines and disk. In the past, you had to contact each site administrator separately, and you would get a username and a password for every new site. By providing this combination, the administrator could be sure who was using the system. But the user was obliged to remember as many passwords as there were sites. This cumbersome way of working is not suitable for the Grid, where you will be accessing many different sites without you even knowing.

On the Grid, you will be using a *certificate*. This certificate binds together your identity (name, affiliation, etc.) and a unique piece of digital data called a "public key" that is explained below. A third party that is trusted by all sites in the EDG test bed digitally signs the combination of your name and the public key.

The use of a public key to authenticate yourself is based on a special mathematical trick, called "asymmetric cryptography". If you would pick two large (prime) numbers and multiply them, it is virtually impossible to factorise the product into the two numbers again. The individual prime numbers are used to generate an *encryption* and a *decryption* function and the product of the two, and then the two numbers are destroyed. If you only have the encryption function, it is impossible to derive the decryption functions from it (and vice versa). So, if you distribute the encryption function – called *public key* – widely (e.g. you put it on the web) but keep the decryption function private, everyone can send you encrypted messages, but only you can read them – and even the sender cannot get the message back!

This method is quite useful if you want to authenticate yourself to a remote site without revealing any personal information: if the remote site knows your public key, it can encrypt a *challenge* (e.g. a random number) using this key and ask you to decrypt it. If you can, you obviously own the private key and therefore you are who you say you are – but still the remote site has to know all the public keys of every one of its customers.

It all becomes simpler if we introduce a trusted third party, a human that can authenticate people in persons – called a *certification authority* or CA. When you go to a CA you bring along your public key and an *identifier* – your full name and possibly an affiliation. Now the CA has to make sure by some other means that you are indeed who you claim to be. The CA may ask for a passport or drivers license, it could contact your boss to verify your affiliation, make a phone call to your office, etc. When the CA is reasonably convinced of your identity, it will take your public key and your identifier and put those together in a *certificate*. As a proof of authentication, the CA will then calculate a digest (hash) of the combination of the two and encrypt it with the *private key* of the CA. Everyone can re-calculate the digest, decrypt the signature using the public key of the CA and verify that these two are the same. If you show up at a remote site that only knows your name (*identifier*) and trust the CA that you got your certificate from, the site known that whoever can decrypt the challenge sent corresponds to the name they have in their list of allowed users.

1.1.2. Getting a certificate yourself

In reality, applying for a certificate may take you a day of two – remember that it requires action by real human beings. For that reason you already have a working certificate and a corresponding private key installed in your home directory on the User Interface machine for use during the tutorial. When you look in the directory, you will see:

```
[davidg@tbn08 davidg]$ ls -l $HOME/.globus
total 16
-rw-r--r--    1 davidg   users        5047 Feb 26  2002 usercert.pem
-rw-------    1 davidg   users         963 Feb 26  2002 userkey.pem
-rw-r--r--    1 davidg   users        2043 Feb 26  2002 userrequest.pem
```

Note the protection set on your private key file – userkey.pem. They are very restrictive and are set thus for a reason: your possession of the private key is the only proof remote sites have that they are indeed taking to you. If you would give that key to someone else (or if it gets stolen), you will be held liable for any damage that may be done to the remote site! In any case, if the user key is world readable or worse, is cannot be used by the Grid.

This private key should also be protected with a *pass phrase* (a difficult name for a password of arbitrary length). Make the pass phrase difficult to guess, and keep it safe. You can change the pass phrase anytime you like.

You can always see what is in a certificate using the `openssl` command. This is a toolkit for handling certificates, keys and requests. The table below lists a few useful commands:

`openssl x509 -text -noout -in certificatefile`

show the contents of a certificate

`openssl req -text -noout -in certificatefile`

show the contents of a certificate request

`openssl verify -CAfile trusted_ca_certificate.0 certificatefile`

verify that a certificate is (still) valid

`openssl rsa -in private_key_file -des3 -out new_private_key_file`

writes a new copy of the private key with a new pass phrase

To make the tutorial a bit more realistic, in this section we are going to create a certificate request and apply for a certificate to the EDGtutorial Certification Authority. The exact procedure is different for every CA – and there are 13 CAs one per country. On the DataGrid web site (www.eu-datagrid.org) you can find links to your national CA information. For regular use of the Grid from the Netherlands, you need a "medium-security CA" certificate from the same DutchGrid CA.

For use with the national grid projects and the EU projects (EGEE, DEISA, CrossGrid), DutchGrid is running the CA. The web site for this CA is http://www.dutchgrid.nl/ca and on this page you find a link to a web form that will help you to generate a certificate request as shown in the figure below (nearly all CAs have such a web form). When you fill all information and make your way through the certification details, you can in the end download a shell script that you can run on the user interface machine. The shell script is called "`makerequest.sh`" by default and is usually written to your home directory.

The direct link to the user request pages is at http://certificate.nikhef.nl/userhelp.html, read the document and fill the request forms similar to the one shown above.

When you run the shell script (run it only once!), it will generate a new, unique public and private key and write a *certificate request* to a file in your .globus directory. It is this request that you have to submit to the CA for certification. A regular certificate request is mailed automatically to the CA, so make sure that your machine can actually send mail.

If for some reason you cannot send mail directly, copy-and-paste the file '`certreqXXX.txt`' into your favourite mail client and send the mail to ca@dutchgrid.nl. The mail looks like this:

```
Certificate request for medium certification

  From: David Groep
  Email address: davidg@nikhef.nl
  Contact info: NIKHEF, room H157, Kruislaan 409, Amsterdam, +31 20 592 2179
  Date: 20031120-1020
  Dir: .
  Pwd: /user/davidg/.globus-lcg

Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: O=dutchgrid, O=users, O=nikhef, CN=David Groep
…
aPznCjlI0WAUCrnP47Hj+P5RTx9PVZNTA95H0B/foF1HwXL46wfkwc4Y8QqGAcuG
B99JoIZx9ZXGVAwYb7eU1r2sl3VC8fsCh6PwWX4gMy6wFl9xlCL9EpFI+wLzC/oK
6fE3EQm+oEqA489G0FwHjlWnFFFFaA==
-----END CERTIFICATE REQUEST-----
```

After a short while, you get a certificate back from the CA. Some CAs send the certificate by e-mail to you, others request you retrieve it yourself from a web site. The normal DutchGrid CA will mail it back to you, but the EDGtutorial CA is of the latter type. In any case, you store it in a file called "`usercert.pem`", in the same directory where you found the "`userrequest.pem`" file.

If you lost the mail, go to the web address and download your certificate:

```
http://certificate.nikhef.nl/medium/certlist.html
```

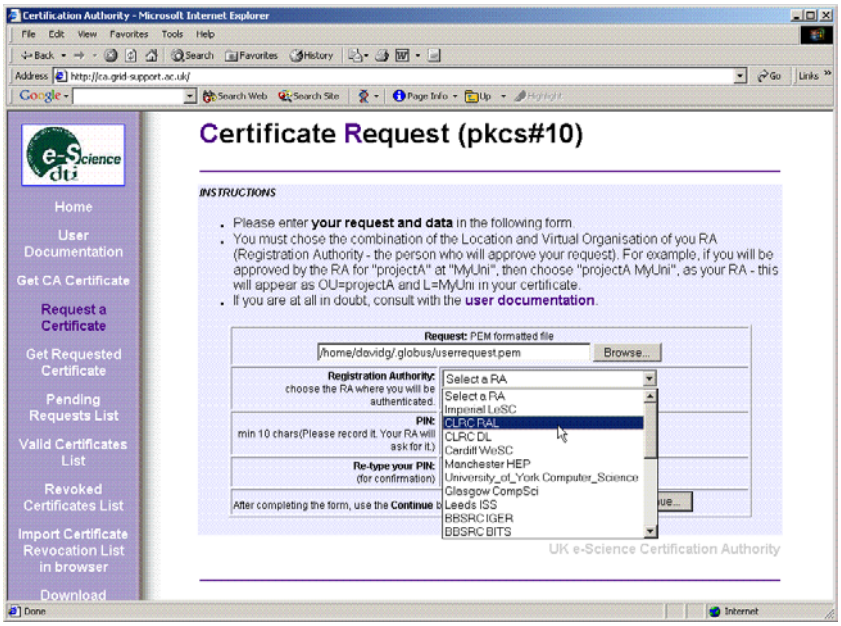
It does not matter how much "bogus" is in this file, as long as you keep the fragment between "BEGIN CERTIFICATE" and "END CERTIFICATE" intact:

```
-----BEGIN CERTIFICATE-----
MIIE1DCCA7ygAwIBAgIBQjANBgkqhkiG9w0BAQQFADBSMQswCQYDVQQGEwJOTDEP
MA0GA1UEChMGTklLSEVGMTIwMAYDVQQDEylOSUtIRUYgbWVkaXVtLXNlY3VyaXR5
…
YjNS8HW/xZ+BvK0hHiIneVccvotJhl35u/qITZK0ExehHIu4UTr1YgaYxOpieIbg
wzUZncH+lVaDME4JcFAOgc5xrA5q+RJeLg8rmbtTvViiK7VEZxyOeg==
-----END CERTIFICATE-----
```

1.1.3. Registration Authorities, do I need one?

For large CAs, it is very difficult to contact everyone personally. Therefore, the task of authenticating people has been devolved onto *Registration Authorities*, or RAs. Like a CA, a RA is a real person, maybe the head of your personnel department, or your team leader. The RAs do not sign certificates themselves, but tell a CA that a particular person belongs to a particular certificate and that they should sign the request. The task of an RA is simple, and many RAs can be appointed for one CA. On the other hand, running a proper CA is a complex task, requiring a secure environment and personnel.

When you request a certificate via the web, you may have to specify which RA is closest to you. For example, every major Grid centre in the UK has such a Registration Authority. When you upload your certificate request using your web browser, you also select the RA for your own lab.



1.1.4. Exercises

1) Look in your certificate directory, and look inside your certificate using the openssl command. What is your "subject name"?
2) Make sure that the files in your `.globus.original` directory are the same as in your `.globus` directory afterwards.
3) Remove the files in your `.globus` directory and copy the original ones from `.globus.original`.
4) Change the pass phrase on your private key.. Then, delete the files in your `.globus` directory and request a certificate from the EDGtutorial CA. What is the subject name in your certificate request?
5) Store your tutorial certificate in the `.globus` directory and try the exercises in the section "Getting a Proxy". Remember to come back here

1.2. REGISTERING IN A VIRTUAL ORGANISATION FOR EGEE, LCG, AND DATAGRID

If you want to use the EGEE or DataGrid Grid for real, you should register with a Virtual Organisation (VO). This may be your experiment (LHCb, Babar) or your community (dteam, EarthOb). Also you thereby agree to the Acceptable Use Policy (of course you do, but realise that you are now legally responsible for your actions ☺). To do this, you must authenticate with your certificate to a web site, and thus you would have your certificate available *inside your web browser*.

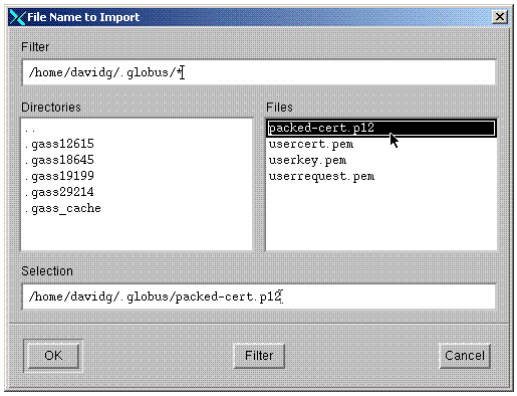
The file you have on disk is suitable for Grid use, but needs to be converted to a different format for web browsers. This format is called PKCS#12, and files have the extension ".p12". This format is special in the sense that the file contains both your public and your private key, and the combination is again protected with a pass phrase (here called "export password").

The openssl programme is again used to convert between the different formats:

```
cd $HOME/.globus
openssl pkcs12 -export -in usercert.pem -inkey userkey.pem -out packed-cert.p12
```

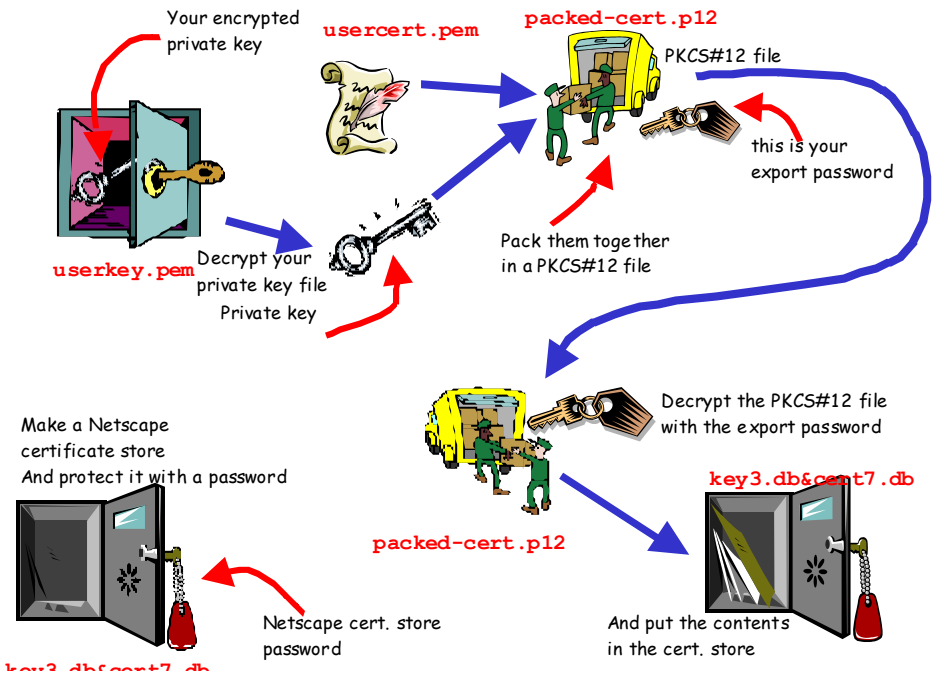
The file packed-cert.p12 now contains both your certificate and your private key, and can be imported in Netscape or Internet Explorer – in this tutorial we will use Netscape 4.75, but Internet Explorer will work as well. The certificate in Netscape can be reached from the button "security", and then "Certificates/yours". You can now import your certificate by pressing the "import certificate" button.

Netscape will protect its certificate store with a password as well. Enter a good password in the dialog, as shown in the pictures below. In the file browser window you will subsequently get, go to your .globus directory and select the packed-cert.p12 file.



Again, you will have to provide a password, this time the "export password" you gave to openssl when you created the PKCS#12 file. You also have to think of a "nickname" for this identity. We suggest you use your username on the User Interface machine.

You have now successfully imported your certificate and you can close the Netscape security window by pressing "OK" at the bottom. Maybe the picture below clarifies all the different passwords you have to provide.



1.2.1. Requesting your account

You are now ready to sign the Guidelines and apply for an account. You can get to the registration page from the main LCG web site http://lcg-registrar. cern.ch/ and selecting **Registration**, or go directly to

```
https://lcg-registrar.cern.ch/cgi-bin/register/account.pl
```

Press OK whenever asked (the web site is protected with a certificate from the CERN CA, which is not recognised by default in Netscape). Using your personal certificate, you can authenticate to the web site.

You will see that all the data from your certificate is already filled in. In real life, you would now have selected your affiliation to DataGrid and apply for a real account … but that has already been done for you by the tutors, so do *not press on the agree button* but instead quit your browser.

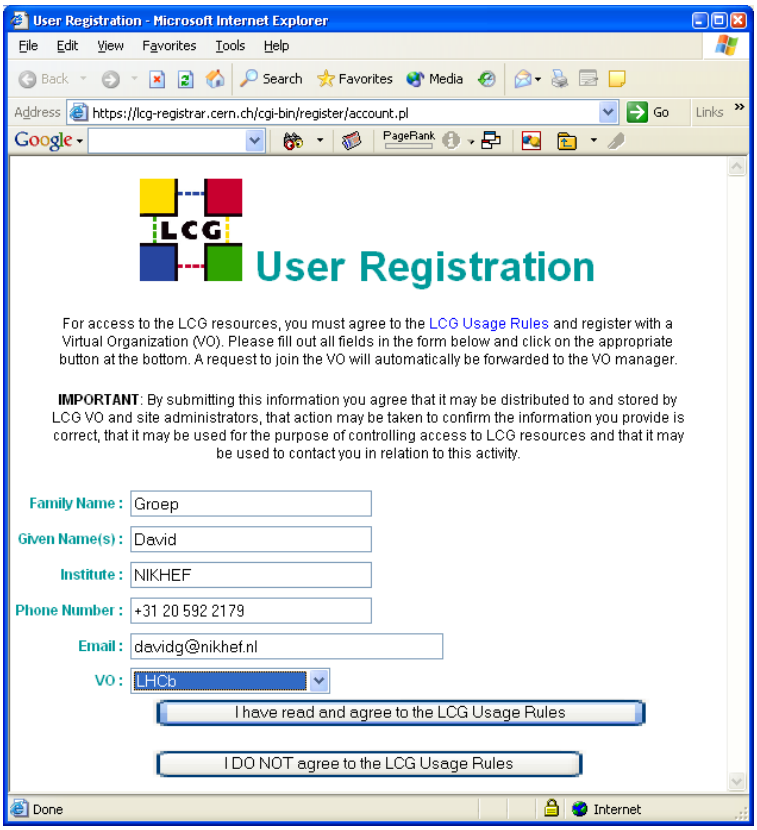
1.3. REGISTERING IN OTHER VIRTUAL ORGANISATIONS

The web registration above only applies for EGEE, EDG and LCG. In you joined a national project like VL-E, please contact the VO directly, e.g., using the mail address given on the project web site.

For VL-E, use the DutchGrid Support system, for the time being, at

```
http://www.dutchgrid.nl/Support/
```

```
support@dutchgrid.nl
```



1.3.1. Excercises

1) Convert your certificate and private key into a PKCS#12 file.
2) Can you get to the VO registration page?