



ID: INFNGRID20040216-1700

Version: v1.0.0

Date: November 18, 2008

A Brief Guide to Certificate Management

Marisa L. Luvisetto

INFN-Bologna (marisa.luvisetto <at> bo.infn.it)

Abstract: *The present documentation is intended as a brief and possibly simple guide to certificate management for Grid access by INFN users.*

Contents

1	Introduction: Concepts, Passphrase	3
2	Certificates: File Format & Conversion	4
3	Personal Certificates released by INFN	6
4	Loading Certificates into Browsers	7
4.1	Loading Certificates with Netscape 4.x	7
4.2	Mozilla & Netscape 7.x	7
4.3	Konqueror	7
4.4	Opera 6.12	9
4.5	Opera 7.50 & 9.62	9
4.6	Windows Firefox	9
4.7	Linux Firefox	10
5	Certificate Passphrases	10
6	Server Certificates	11
7	Certificate Verification	12
7.1	X509 Certificates	12
8	Requesting Server Certificates	14
9	Miscellaneous OpenSSL Commands	15
9.1	Certificate Fingerprint	15
9.2	Certificate Hash Value	15
9.3	General Purpose Commands	15

1 Introduction: Concepts, Passphrase

Certificates are the way users authenticate themselves in network activities that perform identity verifications. The certification method is supported by SSL/TLS (Secure Socket Layer/Transport Layer Security) that implements authentication through the exchange of certificates based on public/private keys according to the X509 standard.

The present documentation is intended as a brief and possibly simple guide to certificate management for Grid access by INFN users. Usage of certificates in Grid computing is described in [4].

Grid activities make use of two types of certificates:

- **personal** certificates, released by a Certification Authority (CA) that recognizes the certificate owner as a individual acknowledged by the Registration Authority (RA) of the user organization
- **server** certificates, released to a Grid Node managed by a site manager, i.e. a user already registered at the CA with a valid personal certificate. Server certificates are required only for main server nodes, like main server nodes, like the computing and storage nodes of a grid unit.

The basic certificate management tasks are:

- manage a personal certificate
- load the certificate into Web Browsers for authentication in intercommunication activities, like registration in a Grid User Group ¹, electronic signature, etc.
- manage server certificates for grid nodes

As certificates are electronic identification documents, they have a validity period. Therefore users must renew both personal and server certificates before expiration date. An expired certificate prevents access to grid activities as the authentication procedure fails.

¹Grid User Groups are managed as VOs, i.e. Virtual Organizations

Access to certificates is private and is granted by user defined **passphrases**. A passphrase is the same as a password but it may have embedded blank spaces:

```
MyNewPass10      password example
My New Pass10    passphrase example
```

Any operation that makes use of certificates needs the correct passphrase, therefore users **must choose** an appropriate passphrase when applying for personal certificates and they **must remember** it for any future use, included certificate renewal.

2 Certificates: File Format & Conversion

The user must also be aware that OpenSSL supports several certificate formats. Certificates are based on the DSA signature algorithm and the RSA algorithm for public-key cryptography according to PKCS algorithms, as described in [7].

The certificate format depends on the application, as there is no agreement on file format standards.

Private keys are usually available in the PEM and DER format. The related files have names of the following type:

```
*key-rsa.pem      for pem files
*key-rsa.der      for der files
```

For OpenSSL applications, the PEM format should suffice. For Java applications, the DER format might be more suitable for importing the private key and certificates.

For **certificates**, the available formats are PEM, DER and PKCS12 with file names of the following type:

```
*cert.pem        for pem files
*cert.der        for der files
*cert.p12        for pkcs12 files
```

In general, the PEM formats are mostly used in the Unix world, PKCS12 in the Microsoft world and DER in the Java world.

Certificate files are ASN.1-encoded objects that may be encrypted according to DES (Data Encryption Standard). The files can optionally be encrypted using a symmetric cipher algorithm, such as 3DES.

An **unencrypted** PEM file might look something like this:

```
-----BEGIN CERTIFICATE-----
MB4CGQDUoLoCULb9LsYm5+/WN992xxbiLQ1EuIsCAQM=
-----END CERTIFICATE-----
```

The string beginning with MB4C . . . is the Base64-encoded, ASN.1-encoded object.

An **encrypted** file would have headers describing the type of encryption used, and the initialization vector:

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4, ENCRYPTED
DEK-Info: DES-EDE3-CBC,C814158661DC1449
AFAZFbnQNrGjzJ/ZemdVSoZa3HWujxZuvBHzHNoesxeyqqidFvnydA==
-----END RSA PRIVATE KEY-----
```

The two headers Proc-Type and DEK-Info declare the type of encryption, and the string starting with AFAZ . . . is the Base64-encoded, encrypted, ASN.1-encoded object.

As web browsers make use of Java applications, they import/export certificates in pkcs12 file format, i.e. public and private keys are packed in one single file using the PKCS#12 algorithm. Other applications require the pem format with

unpacked public and private keys, thus the user must remember the appropriate file format for each application and must perform format conversions as appropriate.

The following tables report a summary of formats used for INFN-Grid applications and two simple scripts with format conversion commands.

INFN-Grid Certificates Format Summary	
Certificate Type	Certificate Format
CA Authority Certificate	DER
Personal Certificate from CA	PKCS12
Grid Access Certificate	PEM

CONVERT pkcs12 to pem
<pre>#!/bin/sh # echo "copy your cert to cert.p12 - then run this script" # openssl pkcs12 -clcerts -nokeys -in cert.p12 -out usercert.pem openssl pkcs12 -nocerts -in cert.p12 -out userkey.pem</pre>
CONVERT pem to pkcs12
<pre>#!/bin/sh # echo "Verify that you are using the correct certificate pair (key/cert)" # openssl pkcs12 -export -out cert.p12 -inkey ./userkey.pem -in ./usercert.pem</pre>

3 Personal Certificates released by INFN

To get a **new personal certificate**, perform the following operations:

- install the CA certificate into your browser
 - open a web browser that allows certificate previewing like netscape or mozilla and go to the site:
<https://security.fi.infn.it/CA/mgt/getCA.php>
The CA download page is opened with user instructions ² and the certificate fingerprint
 - verify that the format selection button at page end is set to DER
 - click the **red** download button Scarica Certificato. An acknowledge window is opened if your certificate is already loaded, otherwise a dialog window is opened that allows certificate preview depending on the browser you are using
 - click on Examine Certificate to verify certificate fingerprint
 - if the fingerprints match, close the preview window and choose OK to download the certificate in your browser
- contact the site RA and ask for a certificate ID request.
INFN RA information is available at the link:
<http://security.fi.infn.it/CA/>
- request your personal certificate as follows:
 - open the link:
<https://security.fi.infn.it/CA/mgt/restricted/ucert.php>
 - fill the form and submit it (**red** submit button at form end)
 - in 48 hours you will receive by e-mail the web link of your certificate
 - connect to the link and download your certificate

If you already have a personal certificate, you will receive an e-mail warning a fortnight before expiration date. To **renew the certificate**:

- connect to the CA site:
<https://security.fi.infn.it/CA/mgt/restricted/rencrt.php>
- select the certificate you want to renew in case you have more than one
- type the certificate *passphrase*
- submit the renewal form

²The INFN CA instructions are available in Italian only

4 Loading Certificates into Browsers

Personal certificates should be loaded into at least one browser when user authentication is required in network operations like registration, certificate renewal, reserved access, electronic signature, etc.

The loading method depends both on browser type and on browser version. In the following sections are reported the loading instructions for the most frequently available browsers on Linux platforms.

4.1 Loading Certificates with Netscape 4.x

Start the netscape browser and act as follows:

- In the upper menu bar select Security, the security window is opened with a menu list on the left side, go to the Certificates item and select Yours. The right side window prints the header of any already loaded certificate
 - click the Import a Certificate.. button
 - type the certificate DataBase *passphrase*, a file viewer is opened
 - move to the directory where the *.p12 file(s) resides and select the file you want to import
 - when the file is imported, select it with a mouse click, then verify the certificate using either the View or the Verify button on the right side of the page
- click on OK to terminate the certificate upload

4.2 Mozilla & Netscape 7.x

Start the browser ³ and act as follows:

- go to the edit menu in the top upper menu bar
- select preferences → privacy & security
- either click the [+] symbol (for mozilla) or the ▷ one (for netscape) or double click on privacy & security (for both) to open the security option list
- select certificates → manage certificates
- the Certificate Manager window is opened with a list of any loaded certificate. If no certificate is present select Import in the option menu at page bottom. If other certificates are already loaded, use the Restore button. After selection, a file viewer window is opened
 - browse the viewer and select the certificate you want to load
 - click Import or Restore as appropriate. you are asked for *passphrases*:
 - * the one you want to use inside the browser (confirmation is queried at database creation)
 - * the one you used when you asked for the certificate (creation passphrase of the certificate)
- the certificate is imported and it is listed in the certificate manager window with an option list of certificate items; the most relevant item is the certificate expiration date
- check the certificate as follows:
 - click on the the certificate and then select View from the option menu
 - double click on the certificate to display a summary of certificate informations

4.3 Konqueror

Start the konqueror browser and act as follows:

- go to the Settings menu in the upper menu bar and select Configure Konqueror
- the KDE Control Module is openend with a menu window on the left side, and a operation window on the right side
- Browse the menu items and select Crypto, the certificate management tool appears in the operation window with a menu bar on top, select Your certificates
- at the right margin of the window is displayed an option menu, select Import
- a file viewer is opened, double click over directories to search for your certificate, double click on the certificate and type the certificate *passphrase*

³The Netscape Version no. is printed in the home page of the browser

- when the certificate is imported, you can check it as follows:
 - double click on the certificate
 - click on *Verify* in the menu window at the left side
 - type the passphrase of your certificate, data about your certificate are displayed; the most relevant informations are the release and expiration dates of your certificate

4.4 Opera 6.12

Start the Opera browser⁴ and act as follows:

- select File in the upper menu bar
- select Preferences → Security, a window is opened with tab options on the top side
- select Personal
- click on Import, a file browser is opened; browse the directories to select the one with your certificate, type the certificate file name
- you are asked for the certificate *passphrase*, your certificate is shown in the window, click on OK to import the certificate
- you are asked for a certificate *passphrase* for the opera browser, the passphrase must be at least six characters long with at least one digit

4.5 Opera 7.50 & 9.62

Start the Opera browser. The Opera version is either printed in the Opera window border (version 7.50) or shown with details from the Help menu of the “About Opera” item (both versions). You may also use `rpm -qa` to get the browser version.

To add your personal certificate you must first generate a special .p12 file from the .pem files created as described in Section 2, then you must create the `usercert.p12` file with the following openssl command:

```
# openssl pkcs12 -export -descert -inkey userkey.pem -in usercert.pem \  
-out usercert.p12
```

Then import the certificate as follows:

- open the Opera browser and load the certificate of your CA (e.g. INFN) as described in Section 3. Opera download area is usually locate at `~/opera/download`
- select Tools in the upper menu bar
- select Preferences → Advanced → Security, a window is opened with security options on the right side
- select Manage Certificates and Authorities to verify the CA certificate, then select Personal
- click on Import, a file browser is opened; browse the directories to select the one with your certificate, type the certificate file name
- you are asked for the certificate *passphrase*, your certificate is shown in the window, click on OK to import the certificate
- you are asked for a certificate *passphrase* for the opera browser, the passphrase must be at least six characters long with at least one digit

4.6 Windows Firefox

Start the firefox and act as follows:

- select Tools in the upper menu bar
- select Options → Advanced, a window is opened with tab options on the top side
- select Encryption and in the opened window select View Certificates
- select Authorities to verify the authority certificate, e.g. INFN CA certificate
- select Your Certificates to manage personal certificates
- click on Import, a file browser is opened; browse the directories to select the one with your certificate, type the certificate file name
- you are asked for the certificate *passphrase*, your certificate is shown in the window, click on OK to import the certificate
- you are asked for a certificate *passphrase* for the firefox browser, the passphrase must be at least six characters long with at least one digit

⁴the opera version is printed on the window border. You may also use `rpm -qa` to get the browser version

4.7 Linux Firefox

Start the firefox and act as follows (firefox-2.0.0.3-4.fc7:

- select Edit in the upper menu bar
- select Preferences → Advanced, a window is opened with tab options on the top side
- select Encryption and in the opened window select View Certificates
- select Authorities to verify the authority certificate, e.g. INFN CA certificate
- select Your Certificates to manage personal certificates
- click on Import, a file browser is opened; browse the directories to select the one with your certificate, type the certificate file name
- you are asked for the certificate *passphrase*, your certificate is shown in the window, click on OK to import the certificate
- you are asked for a certificate *passphrase* for the firefox browser, the passphrase must be at least six characters long with at least one digit

5 Certificate Passphrases

The passphrases involved in certificate management are:

- the **browser** passphrase, that you create when you first populate the certificate database of the browser or you change using the browser password option
- the **p12** passphrase that you create when you download or import *.p12 certificates
- the **pem** passphrase that you create when you convert the .p12 certificate into the .pem cert-key pair

You may use a different passphrase for each case, but you must pay attention to use the appropriate one, otherwise you will not be able to access your certificates unless stored on a safe backup media.

To change passphrases with **Netscape 4.x**:

select Security → Passwords and click on the Change Password button

To change passphrases with **Mozilla & Netscape 7.x**:

select Edit → Preferences → Privacy & Security → Master Passwords and click on the Change Password button.

Double click on Privacy & Security to display the full list option.

6 Server Certificates

When a site sets up a grid environment with computing nodes, the site manager needs the signed certificate at least for the queue manager node, i.e. the CE Computing Element in grid terminology. A certificate is also needed for Mass Storage Nodes, i.e. grid SE Storage Elements. In this case the site manager needs the CA certificate(s) for the node(s). The certificate request steps are the following:

- login into the node for which you need the certificate
- start your preferred web browser and go to the link:
`https://security.fi.infn.it/CA/`
- read the certificate management tutorial
- download the INFN SSL configuration file for INFN at the link:
`https://security.fi.infn.it/CA/mgt/restricted/host.cnf`
- create the request certificates for the node with the following commands:

```
$ openssl req -new -nodes \  
    -out hostreq.pem -keyout hostkey.pem \  
    -config host.cnf  
$ chmod 600 hostkey.pem  
$ chmod 644 hostreq.pem
```

the configuration file `host.cnf` sets default parameters and mandatory variable fields, thus you must answer to the following openssl questions:

 - the name of INFN site, i.e.
Nome Struttura (ad es. Pisa) []:Bologna
 - the fully qualified distinguished IP name of the host, i.e.
FQDN del Server []:myserver.bo.infn.it
 - the e-mail of the server manager in the form Name.FamilyName@bo.infn.it
- openssl creates the two files:
`hostkey.pem` for the private key and
`hostreq.pem` for the certificate signing request
- verify the `hostreq.pem` file using one of the commands:
 - verify the certificate:
`$ openssl req -in req-boalichel.pem -verify`
 - verify and view the certificate:
`$ openssl req -in req-boalichel.pem -text -verify | more`
check that the node name in the certificate and the e-mail address are the correct ones
- save the files in a safe place in a subdirectory that reflects the nodename
- send via e-mail the `hostreq.pem` file to your RA authority. If you are using Netscape 4.x operate as follows:
 - start Netscape that must be configured for e-mail operations
 - open the choice window under Communicator in the upper menu bar
 - choose the Messenger option
 - click on New Msg
 - * fill the To: field with the RA e-mail address and with your own address to keep track of the request
 - * fill the Subject field with the node IP name and a comment, like in the example:
`mynewnode.bo.infn.it cert-request`
 - * click on Attach and select File in the sub-menu, a file viewer window is opened
 - * select the `hostreq.pem` file you want to attach and click on the Attach button of the file viewer window
 - * click on Options to check that the Signed option is active
 - * click on Send to send the request
- in about 48 hours you will receive the certificate by e-mail, the certificate is in text format and must be extracted from the e-mail message and stored with a name of type `hostcert.pem`
- verify the certificate as described in Section 7
- install the certificate in the computing node as described in the application guide

To install the certificate in CE/SE nodes you must perform the following operations:

- login as root into the node in which you want to install the certificate
- verify that in /root are present both hostkey.pem that you created when you have made the request and hostcert.pem that you received from the CA
- verify that hostcert.pem is the node certificate:
openssl x509 -in hostcert.pem -noout -subject
- copy the certificates to /etc/grid-security with the correct protections as shown in the following example:
cd /etc/grid-security/
cp -p /root/hostkey.pem .
cp -p /root/hostcert.pem .
chmod 400 hostkey.pem
chmod 644 hostcert.pem

7 Certificate Verification

7.1 X509 Certificates

To verify your X509 certificates, you may use `ssl-vfy-cert.sh`, a simple certificate verification tool. The tool is a menu driven procedure that shows relevant information about your certificate and operates as follows:

- the tool operates on public .pem files.
- a file name is mandatory as tool argument
- a new file name is entered using the File option.
- supported menu entries are listed by the help option
- the sh option forks a new Bash shell, type `exit` to return to the main menu

Users should always check the certificate validity period with the `date` option.

To check whether the certificate is a personal one or a server one, use the `subject` option and verify the subject parameters OU and CN. For **personal** certificates the parameters are:

OU=Personal Certificate

CN=Mary Jones, i.e. the certificate owner name.

For **Server** certificates the parameters are:

OU=Host

CN=mynode.bo.infn.it, i.e. the fully qualified IP name of the node.

The tool may print error messages as follows:

- if the supplied file name is a private key, the `openssl` query fails with an error message of the following type:
unable to load certificate
5876:error:0906D06C:PEM routines:PEM_read_bio:no start line:pem_lib.c:662:
Expecting: TRUSTED CERTIFICATE
- if the file does not exist, `openssl` error message is:
mynewcert.pem: No such file or directory
unable to load certificate
5965:error:02001002:system library:fopen:No such file or directory:
bss_file.c:245:fopen('mynewcert.pem','r')
5965:error:20074002:BIO routines:FILE_CTRL:system lib:bss_file.c:247:

CERTIFICATE VERIFICATION TOOL ssl-vfy-cert.sh

```
#!/bin/bash
#
CERT=$1
if [ -z $1 ]
then
    echo "cert.pem file name needed - abort"
    exit
fi
#
while echo -n 'Command: '
    read cmd
do
    case $cmd in
    end|quit ) break
        ;;
    help )
        echo " "
        echo "   The supported commands are:"
        echo " "
        echo "   help   type procedure menu"
        echo "   date   type Certificate Validity Dates"
        echo "   subj   type Certificate Subjext"
        echo "   file   set   new Certificate file name"
        echo "   purpose type Certificate Purpose"
        echo "   text   type Certificate full Text"
        echo "   sh     enter bash shell - type exit to reenetr procedure"
        echo " "
        ;;
    date )
        openssl x509 -in $CERT -noout -dates
        ;;
    subj )
        openssl x509 -in $CERT -noout -subject
        ;;
    text )
        openssl x509 -in $CERT -noout -text
        ;;
    purpose )
        openssl x509 -in $CERT -noout -purpose
        ;;
    file )
        echo -n "New Cert File Name: "
        read CERT
        echo verify CERT file: $CERT
        ;;
    sh )
        bash
        ;;
    * )
        echo "Invalid Command"
        ;;
    esac
done
exit
```

8 Requesting Server Certificates

To create and/or verify your request for a server certificate `hostreq.pem` you may use the simple interactive script `server-cert.sh` reported below.

```
CERTIFICATE REQUEST SCRIPT server-cert.sh
#!/bin/bash
#
#-----exec the following openssl command to create .pem info
#-----for your node
#
echo ""
echo "BEFORE STARTING, DO THE FOLLOWING:"
echo "-----"
echo "1) check the HOWTO at https://security.fi.infn.it/CA/"
echo "2) download the new configuration file at:"
echo "      https://security.fi.infn.it/CA/mgt/restricted/host.cnf"
echo "3) N.B. Type your e-mail address in the form John.Griglia@bo.infn.it"
echo " "
#
echo -n 'Create certificate (Y/N)?'
read OK
case $OK in
  Y|y ) openssl req -new -nodes \
    -out hostreq.pem -keyout hostkey.pem \
    -config host.cnf
    if test -f hostkey.pem
    then
      chmod 400 hostkey.pem
    fi
    if test -f hostreq.pem
    then
      chmod 644 hostreq.pem
    fi
  ;;
esac
#
echo "verifying certificate signing requests hostreq.pem"
echo -n 'Verify certificate (Y/N)?'
read OK
case $OK in
  Y|y ) openssl req -in hostreq.pem -text -verify | more
  ;;
esac
exit
```

9 Miscellaneous OpenSSL Commands

9.1 Certificate Fingerprint

A fast certificate verification is the comparison of the fingerprint between a trusted certificate and a certificate you have imported. The trusted certificate fingerprint is available from the web. For INFN CA the CA certificate fingerprint is printed in the download page. For the imported certificate the fingerprint is available with the following SSL command:

```
$ openssl x509 -noout -fingerprint -in public-cert.pem
MD5 Fingerprint=43:FF:27:D0:68:81:AF:E1:7D:2A:D7:D7:E4:FE:CF:6C
```

9.2 Certificate Hash Value

You may store as many certificates as you like in a OpenSSL dedicated directory addressing the certificate via the hash value computed by:

```
$ openssl x509 -noout -hash -in public-cert.pem
6f51b6a8
```

and making a symbolic link between the certificate pem file and the target file 6f51b6a8.0 as follows:

```
$ cd ~/open-ssl-base/certs/
$ ln -s ~/mycert/my-cert.pem 6f51b6a8.0x
```

9.3 General Purpose Commands

The following commands are reported for the sake of completeness, but will be of limited need for generic users

- To remove the Bag Attributes, i.e the header part of a cert.pem key:
openssl x509 -in cert.pem -out certout.pem
- To convert a cert.pem key from PEM to DER format:
openssl x509 -in cert.pem -outform DER -out certout.der
- To output the public part of a cert.pem key:
openssl x509 -in cert.pem -pubkey -out pub.pem

The resulting pub.pem file will look like the following:

```
-----BEGIN PUBLIC KEY-----
MIGfMA0GC.....I
voSJ7VK5E.....A
TPR6lS6m9.....H
yv39C693U.....B
-----END PUBLIC KEY-----
```

- To generate a RSA private key with .pem file name imapd.pem and key size = 1024, do:

```
$ openssl genrsa -out imapd.pem 1024
Generating RSA private key, 1024 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001)
```

References

- [1] OpenSSL Tutorial
<http://www.eclectica.ca/howto/ssl-cert-howto.php>
- [2] INFN Certification Authority
<https://security.fi.infn.it/>
- [3] LCG tutorial for Certificates
http://lcg-registrar.cern.ch/load_certificates.html
- [4] Grid Experience using EDG2 and LCG-1, for certificate handling go to Section: Certificates, Accounts, Proxy and Renewal
<http://www.bo.infn.it/alice/introgrd/edg2/index.html>
- [5] The DES Encryption
<http://www.tropsoft.com/strongenc/des.htm>
<http://www.signalguard.com/encryption/triple-des.htm>
- [6] The Digital Signature Standard: National Institute of Standards and Technology. FIPS Pub 186: Digital Signature Standard. 19 May 1994
- [7] RSA Cryptography Specifications
<http://www.zvon.org/tmRFC/RFC2437/Output/index.html>
- [8] Globus Certificates and Key Algorithms (in Italian)
<http://www.to.infn.it/grid/seminari/18042001/>

Contents