Proxy Certificates

Last review date	Reviewer
2009-09-15	Marco Bencivenni
	Enrico Fattibene

Table of Contents

Proxy Certificates

What proxy certificates are, what information they contain, and how to create them to access the grid infrastructure.

What is a Proxy?

To interact directly with a remote service a certificate can be used to prove identity. However, in the grid world it is often necessary for a remote service to act on a user's behalf, e.g. a job running on a remote site needs to be able to talk to other servers to transfer files, and it therefore needs to prove that it is entitled to use the user's identity (this is known as *delegation*).

One could imagine sending the private key to remote services, however this is *very insecure* because it would delegate full rights to the remote service, would be valid for a long period of time (typically one year) and increases the chance that the private key could be misappropriated.

On the grid, a *proxy* allows limited delegation of rights. Strictly speaking, a proxy is also a certificate, but usually the unqualified term "certificate" is reserved for something issued by a certificate authority (CA). To make a proxy, a new public/ private key pair is created and a new certificate is built containing the public key using a name with the form of the following example:

```
/C=UK/O=eScience/OU=CLRC/L=RAL/CN=john smith/CN=proxy
```

It is signed with the certificate's long- term private key. Proxies normally have a rather short lifetime, typically 12 hours.

When a job is submitted, the proxy certificate, the private key for the proxy and the normal certificate (but not the long- term private key) are sent with it. When the job wants to prove its delegated identity to another service, it sends it the proxy certificate and the standard certificate, but not the proxy private key. This information is sufficient to prove that the remote service has the right to use the delegated identity. If the remote service needs to further delegate rights to the identity to other services, it may create a new proxy based on the first one and give that proxy to the other services, lengthening the certificate validation chain.

In terms of security, a proxy is a compromise. Since the private key is sent with it, anyone who steals it can impersonate the owner, so proxies need to be treated carefully. There is no mechanism for revoking proxies, so in general, even if someone knows that one has been stolen, there is little they can do to stop it being used. On the other hand, proxies usually have a lifetime of only a few hours, so the potential damage is fairly limited.

A proxy, which includes its own private key, is a file that can be stored anywhere. By default, it is stored in a file like `/ tmp/x509up_u1234` where `1234` is the user ID (UID). Like the certificate key, a proxy must only be readable by the owner. However, there is no pass phrase protection. Note that / tmp is a local area, so when using interactive farms (like lxplus at CERN), sessions running on different machines may use different proxies. The default location can be changed by setting the environmental variable `X509_USER_PROXY` to the full file name to use for the proxy; this can be on a shared file system.

Information Contained in Proxies

The proxy, as explained above, contains a public and private key pair that is signed by the original certificate. These contain information about the identity of the user, that is, the users Distinguished Name (DN). The proxy may also contain information about membership in particular VOs.

A system called *Virtual Organization Membership Service (VOMS)* is used in the Grid infrastructure to manage information about the roles and privileges of users within a virtual organization (VO). This information is presented to services via an extension to the proxy. At the time the proxy is created, one or more VOMS servers are contacted, and they return an *Attribute Certificate (AC)* that is signed by the VO and contains information about group membership and any requested roles within the VO.

To create a VOMS proxy, the ACs are embedded in a standard proxy, and the whole thing is signed with the private key of the parent certificate. Services can then decode the VOMS information and use it as required, e.g. a user may only be allowed to do something if he has a particular role from a specific VO. One consequence of this method is that VOMS attributes can only be used with a proxy, they cannot be attached to a CA- issued certificate.

One other thing to be aware of is that the proxy and each AC has its own lifetime. Typically each AC has the same expiration time as the proxy as a whole, but it is possible that they may be different depending on VO policies and on the times specified when the proxy is created. VOMS servers usually limit the AC lifetime to a maximum of 24 hours, although a higher limit has been agreed in some cases. Differing expiration times often causes authorization problems on the grid.

Groups and Roles

The VOMS system allows a proxy to have extensions containing information about the user's VO membership including information about what groups include the user and what roles the user is entitled to have.

In VOMS terminology, a group is a subset of the VO containing members who share some responsibilities or privileges in the project. Groups are organised hierarchically like a directory tree, starting from a VO- wide root group. A user can be a member of any number of groups, and a VOMS proxy contains the list of all groups the user belongs to. When the VOMS proxy is created the user can choose one of these groups as the "primary" group.

A role is an attribute which typically allows a user to acquire special privileges to perform specific tasks. In principle, groups are associated to privileges that the user always has, while roles are associated to privileges that a user needs to have only from time to time. Note that roles are attached to groups, i.e. roles in different groups with the same role name are distinct.

The groups and roles are defined by each VO; they may be assigned to a user at the initial registration, or added subsequently. Groups and roles are identified by *Fully Qualified Attribute Names (FQAN)*. The format is:

```
FQAN = <group name>[/Role=<role name>]
```

for example, `/ cms/ HeavyIons/ Role=production`. Resource providers use these FQANs to authorize users to perform specific tasks in the provided services.

Managing a proxy

The `voms- proxy- init` command generates a grid proxy and optionally contacts one or more VOMS servers, retrieves the user attributes, and includes them in the proxy. If used without arguments, it creates a standard proxy, as shown above, without any VO information.

Creating a VOMS proxy with VO information

To create proxy that contains VO membership information but without requiring any special role or primary group, the following format is used:

```
$ voms-proxy-init -voms cms
```

where `<vo>` is the VO name. The output is similar to:

```
Your identity: /C=CH/O=CERN/OU=GRID/CN=John Doe
Enter GRID pass phrase:
Creating temporary proxy .............................................. Done
Contacting  lcg-voms.cern.ch:15002 [/C=CH/O=CERN/OU=GRID/CN=host/lcg-voms.cern.ch]
"cms" Done
Creating proxy ................................................................ Done
Your proxy is valid until Thu Mar 30 06:17:27 2006
```

The proxy will be written in a path following the template `/ tmp/ x509up_u<uid>` where `<uid>` is the UID of the user, unless the environment variable `X509_USER_PROXY` is defined, in which case its value is taken as the proxy file path.

If the user gives a wrong pass phrase, the proxy file cannot be written, or the private key has incorrect permissions, an error will be written to the console and the proxy will not be generated. In all cases, using the `- debug` option will give more detailed reasons for the failure.

By default, the proxy has a lifetime of 12 hours. To specify a different lifetime, the `- valid H:M` option can be used (the proxy is valid for `H` hours and `M` minutes - default is 12:00). When a proxy has expired, it becomes useless and a new one has to be created with `voms- proxy- init`. However, longer lifetimes imply bigger security risks and the Grid Acceptable Use Policy generally limits proxy lifetimes to 24 hours; some services may reject proxies with lifetimes which are too long.

There are two steps into the creation of a proxy: a simple grid proxy is created first and used to authenticate to the VOMS server, and the full VOMS proxy is then created using information returned by it. If a valid proxy already exists the `- noregen` option can be used to avoid the first step, including typing the passphrase.

One clear advantage of VOMS proxies over standard proxies is that the middleware can find out to which VO the user belongs from the proxy, whilst when using a simple proxy, the VO has to be explicitly specified.

To create a proxy with a given role (e.g. `production`) and primary group (e.g. `/ cms/ HeavyIons`), the syntax is:

```
$ voms-proxy-init -voms <alias>:<group name>[Role=<role name>]
```

where `<alias>` specifies the server to be contacted, usually just the name of the VO. For example:

```
$ voms-proxy-init -voms cms:/cms/HeavyIons/Role=production
```

The command `voms- proxy- init` requires a user- level configuration file, whose path can be specified in several ways; if the path is a directory, the files inside it are concatenated and taken as the actual configuration file. Usually this configuration is done by the system administrator of the user interface machine. If you see errors about the configuration not being found for your VO, you may need to create a configuration file yourself.

Use the option `- help` for a full listing of the available options.

Printing VO information in a proxy

The `voms- proxy- info` command is used to print information about an existing VOMS proxy. Two useful options are `- all`, which prints everything, and `- fqan`, which prints the groups and roles in FQAN format. For example:

```
$ voms-proxy-info -all
```

which prints out something similar to the following:

```
subject   : /C=CH/O=CERN/OU=GRID/CN=John Doe/CN=proxy
issuer    : /C=CH/O=CERN/OU=GRID/CN=John Doe
identity  : /C=CH/O=CERN/OU=GRID/CN=John Doe
type      : proxy
strength  : 512 bits
path      : /tmp/x509up_u10585
timeleft  : 11:59:58
=== VO cms extension information ===
VO        : cms
subject   : /C=CH/O=CERN/OU=GRID/CN=John Doe
issuer    : /C=CH/O=CERN/OU=GRID/CN=host/lcg-voms.cern.ch
attribute : /cms/Role=NULL/Capability=NULL
timeleft  : 11:59:58
```

There are separate times to expiry for the proxy as a whole and the VOMS extension, which may be different.

The spurious warning can be ignored, if you have not requested any VOMS proxy extensions (i.e. used the `- voms` option of `voms- proxy- init`). If a proxy does not exist, the output is:

```
ERROR: Couldn't find a valid proxy.
Use -debug for further information.
```

Destroying a proxy

To destroy an existing proxy before its expiration, it is enough to do:

```
$ voms-proxy-destroy
```

If no proxy exists, the result will be:

```
ERROR: Proxy file doesn't exist or has bad permissions
Use -debug for further information.
```