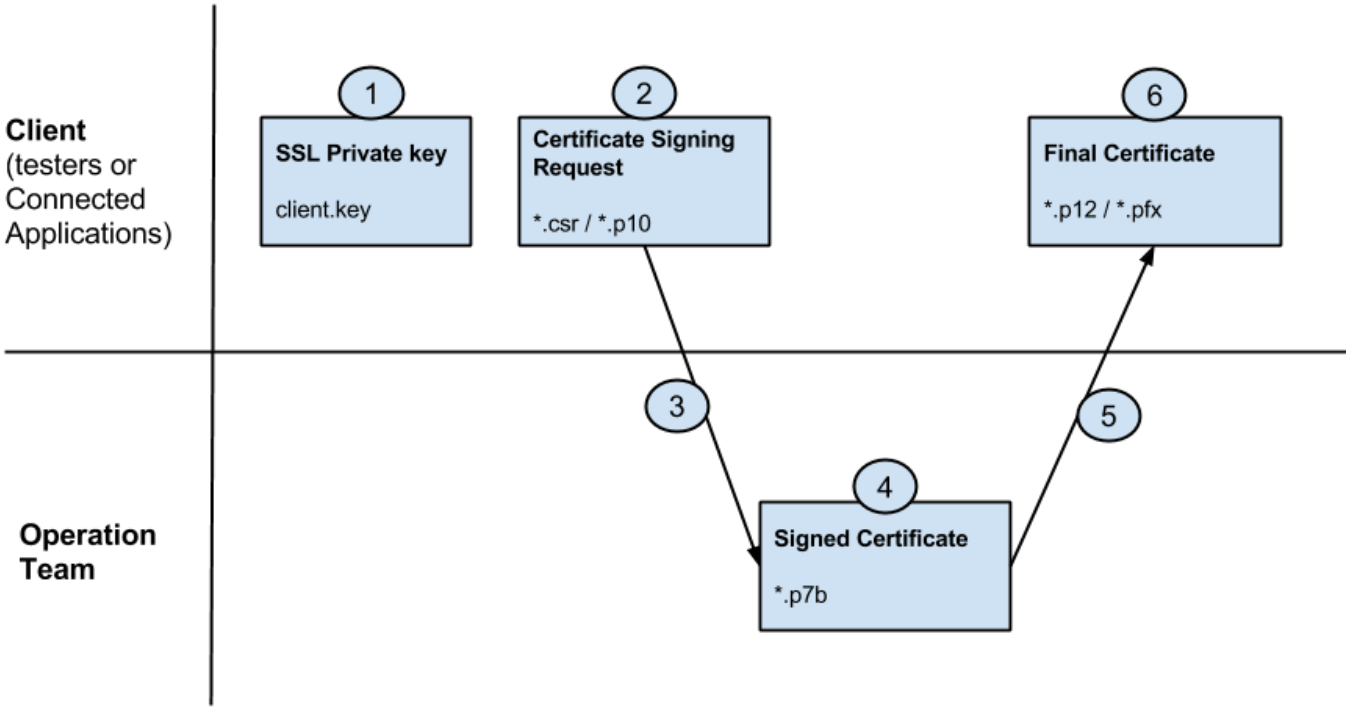
2 Way SSL Tutorial

The following is an explicit list of action to perform by Client (who configure a client application to access protected serverdata) and the OperationTeam (who provides the server application that need to be accessed securely)



Client :

1 - Generate an SSL Private key .key file (or re-use the one located in your ~/.ssh/id_rsa)

```
$ openssl genrsa -out oreo_sqe_skmUser.privateKey 2048
c:\> openssl genrsa -out oreo_sqe_skmUser.privateKey 2048
```

2 - Generate a “Certificate Signing Request” .p10 file (replace <description> by :
Your name or login if the cert is meant to be used in a browser

```
$ openssl req -new -key oreo_sqe_skmUser.privateKey -out $(date
'+%Y%m%d')_oreo_sqe_skmUser.p10
c:\> TODO (windows command)
```

You will have to fill the following form (note : accents or special chars are prohibited):

```
Country Name (2 letter code) [XX]:FR                    # your organization’s country
State or Province Name (full name) []:rhonealpes        # region or state of your org.
Locality Name (eg, city) []:Grenoble                    # Your organization’s city
Organization Name (eg, company) []:Edifixio             # your organization’s name
Organizational Unit Name (eg, section) []:appli env     # appli and env you want to access
```

```
Common Name (e.g. server FQDN or YOUR name) []:XXX    # your name / your server-name
Email Address []:                                     # not needed

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:                              # not needed
An optional company name []:                          # not needed


- Possible Values for Organizational Unit Name :
oreo int
oreo sqe
oreo ppr
oreo prd
```

3 - give the .p10 file to OperationTeam.
NOTE: the private key must remain private. Do NOT send the private key to anyone, keep it in a safe location.

- go to : https://itop.operations.edifixio.com/pages/UI.php (ask account creation to eleonore.dauvergne@edifixio.com)
- go to : HelpDesk (Left Panel) > New user request
- Fill the form :
    - Organization : Schneider_Infra
    - Call : you
    - Origin : portal
    - Application Solution : BSL
    - Service: Add/Change/Remove
    - Service Subcategory : user & right
    - Qualification Imp : One user
    - Qualification Urgency : Medium
    - Title : BSL https certificate for XXX

Content :
Hi,
Can you generate .p7b certificate file for the .p10 file provided as attachment ?

OperationTeam :
4 - Sign the .p10 file using procedure described on this page:
https://operations.ebusiness.schneider-electric.com/w/index.php/BEM/2ways_SSL
5 - Sends back to the client the signed cert as a .p7b file

Client :
6 - generate a .p12 file using the .p7b send by OperationTeam and the following 2 commands :

```
$ openssl pkcs7 -in 20160915_oreo_sqe_skmUser.p7b -inform PEM -out
20160915_oreo_sqe_skmUser.pem -print_certs
$ openssl pkcs12 -export -inkey oreo_sqe_skmUser.privateKey -in 20160915_oreo_sqe_skmUser.pem
-name "skmUser" -out 20160915_oreo_sqe_skmUser.p12
c:\> TODO (windows command)
```

(you might need to supply a password and confirm it to protect your p12 file.).

- to configure your **Browser** you will need the finalCert.p12 file. You will need the previous
- to configure your **Soapui** you will need to put your finalCert.p12 file in a .jks file (Java Key Store file) :

```
$ JAVA_HOM/bin/keytool -v -importkeystore -srckeystore 20160915_oreo_sqe_skmUser.p12
-srcstoretype PKCS12 -destkeystore myUserName.jks -deststoretype JKS
c:\> TODO (windows command)
```

- to configure your server you may need the following files : `20160915_oreo_sqe_skmUser.p12, oreo_sqe_skmUser.privateKey, 20160915_oreo_sqe_skmUser.pem, 20160915_oreo_sqe_skmUser.pem.p7b`

OpenSSL for Windows 64bit : you need to install
http://slproweb.com/download/Win64OpenSSL-1_0_1g.exe
http://www.microsoft.com/downloads/details.aspx?familyid=bd2a6171-e2d6-4230-b809-9a8d7548c1b6

Sample files

Sample .p10 file

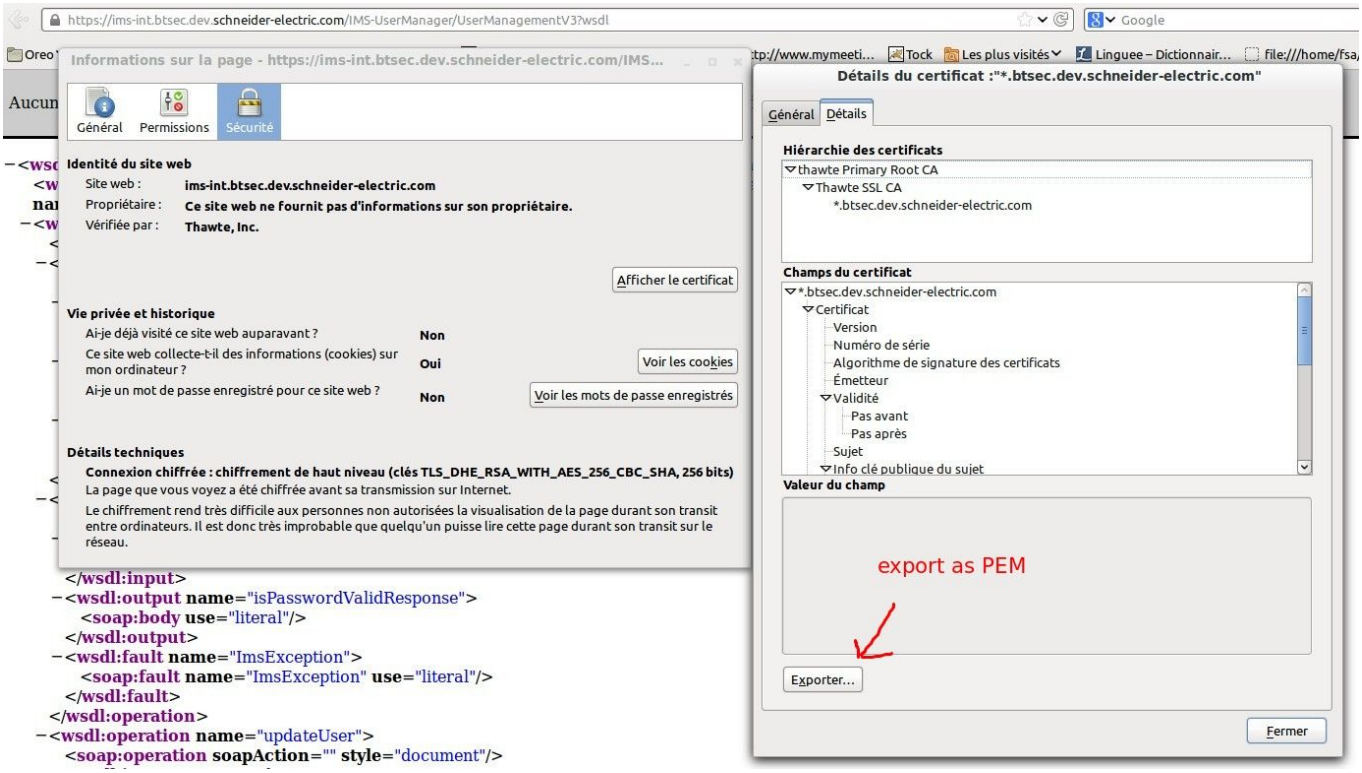
```
-----BEGIN CERTIFICATE REQUEST-----
MIICujCCAaICAQAwdzEcMBoGA1UEAwwTVFJBSU4wMl9DZXJ0aWZpY2F0ZTEMMAoG
A1UECwwDYkZPMRswGQYDVQQKDBJTY2huZWlkZXIgRWxlY3RyaWMxDjAMBgNVBAcM
BVBhcmlzMQ8wDQYDVQQIDAZGcmFuY2UxCzAJBgNVBAYTAkZSMIIBIjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEAmqJ5vUiiSgm72gwc5HIv+jgK5LJ91i+7RTdG
i/T3xCUIoXh6QtPD0znrUHPT4IdJZWdHAfi43wyRfFZThRXVYqfK0zXj7MGrAvx5
W2MUBHZ3xVxbk68mYMP8IXgXe9cMjkoM6KkTAsXv4TqfwSNa7AYqhd0/OyFpzExz
ciR6anYmUxAB5CIgW2FQYZNnfHna8fiCCWeyBuLir4eazCNCof/bkL3gh/hVrxbY
gJnoHJ3TMD46EwJt8yeJ/AC0+vpryZSdhoVDn1n++gv655yPdhwL9uNsTD+QT+mM
TPIYaZCZusp+cx4yq/uhSaeU1GZPd0Cn1VtXZWQx5syPeJixuQIDAQABMA0GCSqG
SIb3DQEBBQUAA4IBAQCUIP7rkKilMaDo2ZEKD8kiNoUUX3y9Hb7VWAGhcWFuACV/
u8Y4EHXMVn7s+V/PTegB9/FhrvrxUAts/r89AFbNLHj+Z164+0P16DJyts4lDuyf
37PDgu4/rChTRZagOP1si8v7VGg86dnZFsNqaYjx6CyJ3xKBeE6UbMaSMbMqbmGc
ZsPOAUTeoxVdzSXf7MEhfhlIifJAhbU7NEEAaDTOWEiHfU3rbcATLOvrd60joOASh
sgiV+2yQM+cXodCyRfIJrR5bFe6SoXYYoHSTHqIxp1FIXzs/6zcfOvREZqFwtbFC
fmZQ7tvdPbHIIfdIt94dBzZdgAxUbFEwYtf7O+7A
-----END CERTIFICATE REQUEST-----
```

Sample resulting .p7b file

```
-----BEGIN PKCS7-----
MIIIXQYJKoZIhvcNAQcCoIIITjCCCEoCAQExADALBgkqhkiG9w0BBwGggggwMIID
5jCCAs6gAwIBAgIBBjANBgkqhkiG9w0BAQUFADBzMQswCQYDVQQGEwJGUjEOMAwG
A1UECBMFUGFyaXMxDjAMBgNVBAcTBVBhcmlzMRswGQYDVQQKExJTY2huZWlkZXIt
RWxlY3RyaWMxJzAlBgNVBAMTHnRybi1iZW0uc2NobmVpZGVyLWVsZWN0cmljLmNv
bTAeFw0xMjEwMDMwOTIxNThaFw0xMzEwMDMwOTIxNThaMH0xCzAJBgNVBAYTAkZS
MQ8wDQYDVQQIDAZGcmFuY2UxDjAMBgNVBAcMBVBhcmlzMRswGQYDVQQKDBJTY2hu
ZWlkZXIgRWxlY3RyaWMxDDAKBgNVBAsMA2JGTzEiMCAGA1UEAwwZVFJBSU4wM2Jl
bXRybl9DZXJ0aWZpY2F0ZTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
AJMglvsIQgHQSZ7T5AQVpxZ4zJHQs9wLfSqGpNdUc2PCZH1rYsQ5I1Vru22Hxpl6
eC0PhHNFXGp9JFCbACVidnXSdMyC6I2MNtzua2Z+7iipeY3XXp6VKaD94a9AU5ID
07CpD0UUJ5tMr4cAl6XDQZFZf5OYH8dQnrMAaJoY8wyEzi/hfZvGpPkVkKpJWOXv
w/cy/prQYf8LqRyCgG30i3ApcNF3S/0LOXgKTASj86fgfLy3YaIcsQGNTow5WGfj
Z7jrZN7cQj9sfaDqYjq/a5lnbgzuprveO8X77iKA3QBxtgWlkmApkyqbJMqPfLVP
GLhhOO7tETtIeK1s0nu7QBUCAwEAAaN7MHkwCQYDVR0TBAIwADAsBglghkgBhvhC
AQ0EHxYdT3BlblNTTCBHZW5lcmF0ZWQgQ2VydGlmaWNhdGUwHQYDVR0OBBYEFK/b
ZN+goo12KLVdC8C1Fza+FjoNMB8GA1UdIwQYMBaAFCxfojCBy8lYqkprykrag/7l
zm2rMA0GCSqGSIb3DQEBBQUAA4IBAQCMUhmbsbEMxdt72eUxCWQ27ZSrl86ZEzVM
Ejcebbo4sp1zZRn2ZLBNSeuL0ZD1Yxl6E+ljtOp9pnVpdbr4coY/Jv9uBlFsXqt7
jSxe664JCRbpcH7SNuwOfOnOtEm8QwCqyyg0yNiUJ6YLZQh1cpyBygLOpgKqvfeh
AYEvzMTF7qbRxk4d+7Z4Bda47g3RBTGkUNIG83dzqnHYtVnQ2Up8Ex0Hzh4HQIcc
eME/QFus/I8UNTph8av4HJLXE7Ve8wgkjuVzeIjPyn8aNYEvY+Ti5ivtwYn9Hkdi
2gLlTQOm7GFizEfbyxpO9G3kEemQLqZDvn9Afn1WuS4vUhRFD7LuMIIEQjCCAyqg
AwIBAgIJAKj8e4QGCySqMA0GCSqGSIb3DQEBBQUAMHMxCzAJBgNVBAYTAkZSMQ4w
DAYDVQQIEwVQYXJpczEOMAwGA1UEBxMFUGFyaXMxGzAZBgNVBAoTElNjaG5laWRl
ci1FbGVjdHJpYzEnMCUGA1UEAxMedHJuLWJlbS5zY2huZWlkZXItZWxlY3RyaWMu
Y29tMB4XDTEyMDcxOTEyMTMyMVoXDTE3MDcxODEyMTMyMVowczELMAkGA1UEBhMC
RlIxDjAMBgNVBAgTBVBhcmlzMQ4wDAYDVQQHEwVQYXJpczEbMBkGA1UEChMSU2No
bmVpZGVyLUVsZWN0cmljMScwJQYDVQQDEx50cm4tYmVtLnNjaG5laWRlci1lbGVj
dHJpYy5jb20wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDD/PUERiVg
aMmKjZKoG653HM702HfA1D02KmJ+xs6XDV+xMow5ZadNsWOMR5m+f7Ae3U6AH0WS
nY88Wtsv1JnNU/EqOldRHBKZRgkZMDBZq9gQ6D/wqNYjCeO6CnjZU/jscuEwR5Db
uZaibwH1hIsk37KjRdLSHq0EtrE2GZS3dkwtbzsaRAKn2btfgA9/upikzpDaBKhd
gl4aJP5o8GvsPTZ/H5ilYkYKTBG9mwXFgXRQBkJ8N1EXB6SBeoOhHA36Lwad602Q
rwGpfwW6nBPmnDab9JzArUMXb+5F2MDiScYsyGbdTylLQjBLcoGGCeKpKJNrY3rB
```

```
1UalpNySuFHpAgMBAAGjgdgwgdUwHQYDVR0OBBYEFCxfojCBy8lYqkprykrag/7l
zm2rMIGlBgNVHSMEgZ0wgZqAFCxfojCBy8lYqkprykrag/7lzm2roXekdTBzMQsw
CQYDVQQGEwJGUjEOMAwGA1UECBMFUGFyaXMxDjAMBgNVBAcTBVBhcmlzMRswGQYD
VQQKExJTY2huZWlkZXItRWxlY3RyaWMxJzAlBgNVBAMTHnRybi1iZW0uc2NobmVp
ZGVyLWVsZWN0cmljLmNvbYIJAKj8e4QGCySqMAwGA1UdEwQFMAMBAf8wDQYJKoZI
hvcNAQEFBQADggEBAJ3eny/a58dp+8dfbK2mUmqpw7uAy7Tig7BNeXAHuOmuKmBl
ZDRCLyZ/9BzWMb/yXaRoAIbJYpPRcvfsHK+AMo0BHHS6nlGmfo2NptJcgZ2tY947
s9qeM04AfDXyyncx/Lx+HNHaSxC3bUghQj5lhf2ofnAuVQNWCA3ZKtvoqgDDh30a
Nf5aaAsQmDPJ/gNc1oNoqictc/50CwiwyEY160dJWUR+zFAn9swJ+QQNyISjM/sU
i1iHqDbCPpAiD8He7xLisgMkCkgBqu8HUliyuJYrL+Kp4sDum8UTVi9Lm4mV38AW
LBfFI6a8b4SApeduwUZ2QSZYNsxcK86mr889yA+hADEA
-----END PKCS7-----
```

How to import website certificate in a keystore (1 way SSL)
fetch the certificate from the site to access :



run the command :
/bin/keytool -import -trustcacerts -file ims_int_fromSite.pem -keystore ims_sqe.jks

View the content of a p10
openssl req -in 20131125_04.bsl_prd_fsa.p10-noout -text

Definitions:

Certificate signing request: a **certificate signing request** (also **CSR** or **certification request**) is a message sent from an applicant to a certificate authority in order to apply for a digital identity certificate. The most common format for CSRs is the PKCS #10 specification and another is the Signed Public Key and Challenge Spkac format generated by some Web browsers

Client Certificate:

In cryptography, PKCS #12 defines an archive file format for storing many cryptography objects as a single file. It is commonly used to bundle a private key with its X.509 certificate or to bundle all the members of a chain of trust.[1]

A PKCS #12 file may be encrypted and signed. The internal storage containers, called "SafeBag"s, may also be encrypted and signed. A few SafeBags are predefined to store certificates, private keys and CRLs. Another SafeBag is provided to store any other data at individual implementer's choice. [2][3]

PKCS #12 is one of the family of standards called Public-Key Cryptography Standards (PKCS) published by RSA Laboratories.

The filename extension for PKCS #12 files is ".p12" or ".pfx".[4]

GCR viewer (on Ubuntu)
gcr-viewer file.p7
Site en ligne :
https://certlogik.com/decoder/