

---

# HSM Proxy Administrator Manual

Version 0.5.1

Frank Cornelis

Copyright © 2013 FedICT

## Abstract

This manual describes the administrator procedure for the HSM Proxy product.

1. Administrator privileges .....	1
2. Creation of PKCS#12 key stores .....	1
3. Creation of eToken PKCS#11 key store .....	2
4. Key Store configuration .....	5
5. Application Credentials .....	5
A. HSM Proxy Administrator Manual License .....	5
B. HSM Proxy Project License .....	5

## 1. Administrator privileges

The administrator portal authentication is based on the Belgian eID card. The first time that the HSM Proxy service starts, it doesn't know any administrator. Hence the first user that logs in into the administrator portal, will received administrator rights. Subsequent logins using another eID card will fail. However, these other users are being registered as pending administrators and can be approved by actual administrators.

## 2. Creation of PKCS#12 key stores

This section described how to create PKCS#12 key stores using OpenSSL. These PKCS#12 key stores can be used as software key stores within the HSM Proxy service.

Create a 1024 bit RSA key pair with default public exponent via:

```
openssl genrsa -out key.pem -F4 1024
chmod og-rw key.pem
```

Create a new self-signed certificate via:

```
openssl req -config openssl.conf -new -x509 -key key.pem -out cert.pem -verbose
-days 365
```

with the configuration file `openssl.conf` containing the following:

```
[req]
distinguished_name = req_distinguished_name
prompt = no
x509_extensions = req_x509_extensions

[req_distinguished_name]
commonName=Test HSM

[req_x509_extensions]
```

View the new X509 certificate via:

```
openssl x509 -noout -text -in cert.pem
```

Create a PKCS#12 key store via:

```
openssl pkcs12 -export -out keystore.p12 -inkey key.pem -in cert.pem -name alias
```

View the content of the PKCS#12 key store via:

```
openssl pkcs12 -info -in keystore.p12
```

Convert a certificate from PEM format to DER format via:

```
openssl x509 -in cert.pem -out cert.der -outform DER
```

View the content of the PKCS#12 via the Java `keytool` tool:

```
keytool -list -storetype PKCS12 -keystore keystore.p12
```

### 3. Creation of eToken PKCS#11 key store

This section describes the creation of SafeNet eToken PKCS#11 key stores. Via eTokens you can emulate production HSM PKCS#11 key stores.

Install the SafeNet Authentication Client version 8.1+.

Check whether the smart card reader is available via:

```
pcsc_scan
```

We will first initialize the eToken PRO using the SafeNet Authentication Client (SAC) Tool. In the SAC click "Advanced View". Select the token and click "Initialize Token". For token name you pick "HSM Proxy". Set a token password and uncheck "Token Password must be changed on first login". Click "Advanced". Check "2048-bit RSA key support". Next click "Start" to initialize the token.

The location of the PKCS#11 module on 32 bit systems is `/usr/lib/libeTPkcs11.so`. The location of the PKCS#11 module on 64 bit systems is `/usr/lib64/libeTPkcs11.so`.

Retrieve the PKCS#11 slot list index of the token via:

```
pkcs11-tool --module /usr/lib/libeTPkcs11.so --list-slots --show-info
```

The `pkcs11-tool` tool is part of the `opensc` package. The slot list index should be used as parameter for `--slot` in the following commands.

Create a new 2048 bit RSA key pair on the token via:

```
pkcs11-tool --module=/usr/lib/libeTPkcs11.so --keypairgen --key-type rsa:2048 --  
login --id 45 --label HSM --slot 0
```

Create a new self-signed certificate based on the private key of the token via:

```
openssl req -config openssl-hsm-proxy.conf -engine pkcs11 -new -x509 -key 45 -  
keyform engine -out hsm-proxy-cert.pem -verbose -days 356
```

with `openssl-hsm-proxy.conf` containing:

```
openssl_conf = openssl_def

[openssl_def]
engines = engine_section

[engine_section]
pkcs11 = pkcs11_section

[pkcs11_section]
engine_id = pkcs11
dynamic_path = /usr/lib/openssl/engines/engine_pkcs11.so
```

```
MODULE_PATH = /usr/lib/libeTPkcs11.so
init = 0

[req]
distinguished_name = req_distinguished_name
prompt = no
x509_extensions = req_x509_extensions

[req_distinguished_name]
C=BE
L=Brussels
O=FedICT
OU=eID
commonName=HSM Proxy

[req_x509_extensions]
```

View the created certificate via:

```
openssl x509 -noout -text -in hsm-proxy-cert.pem
```

Convert the certificate from PEM to DER format via:

```
openssl x509 -inform PEM -in hsm-proxy-cert.pem -outform DER -out hsm-proxy-
cert.der
```

Write the X509 DER certificate to the token via:

```
pkcs11-tool --module /usr/lib/libeTPkcs11.so --write-object hsm-proxy-cert.der
--type cert --login --label HSM --id 45 --slot 0
```

List the content of the token via:

```
pkcs11-tool --module /usr/lib/libeTPkcs11.so --list-objects --login --slot 0
```

Check whether the token is accessible from within Java via:

```
keytool -keystore NONE -storetype PKCS11 -providerClass
sun.security.pkcs11.SunPKCS11 -providerArg etoken.config -list -v
```

with the file `etoken.config` containing:

```
name=eToken  
library=/usr/lib/libeTPkcs11.so  
slotListIndex=0
```

## 4. Key Store configuration

Via the HSM Proxy administrator console you can configure the different HSM key stores.



### Key store passwords

Keep in mind that the configured key store passwords are available within the database in plain text. Thus the HSM Proxy should have at least a dedicated database table space.

## 5. Application Credentials

Application credentials can be any X509 certificate, including self-signed certificates. Application credentials should be uploaded in DER encoded format. A given application credential can be registered for only one application at the same time.

The HSM Proxy trust model is based on the fingerprint of the X509 certificate application credential. The HSM Proxy trust model does not perform any validity check on these certificates. Hence it is up to the administrator to properly manage application credentials.

## A. HSM Proxy Administrator Manual License



This document has been released under the [Creative Commons 3.0](http://creativecommons.org/licenses/by-nc-nd/3.0/) [http://creativecommons.org/licenses/by-nc-nd/3.0/] license.

## B. HSM Proxy Project License

The HSM Proxy Project source code has been released under the GNU LGPL version 3.0.

This is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License version 3.0 as published by the Free Software Foundation.

This software is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this software; if not, see <http://www.gnu.org/licenses/>.