

Utilisation d'un certificat cachet d'établissement à l'université de Rouen



Il est possible d'utiliser un certificat cachet d'établissement (personne morale). Le POC mené à Rouen repose sur l'utilisation d'un certificat cachet eIDAS (RGS**) obtenu auprès de Certinomis au format clé cryptographique.

La clé est de marque Feitian et la carte à puce de type Gemalto.

Depuis la version 1.27, il est possible d'utiliser OpenSC en lieu et place du driver safenet.



L'installation se passe coté serveur ce qui implique de connecter la clé usb sur le serveur hébergeant esup-signature. Cela implique des contraintes particulière par rapport aux serveurs virtuels.

Deux solution sont possibles :

- Un port mappé sur le serveur hébergeant esup-signature. De ce fait, le serveur ne peut pas changer d'hyperviseur sans que la clé ne soit débranchée. Le débranchement de la clé est géré coté esup-signature pour permettre une continuité de service (en mode dégradé) lors des manipulations sur l'infra serveur.
- Un hub USB (type AnywhereUSB) permettant de virtualiser le port usb et éviter les contraintes liées à la première solution

- [Pré-requis](#)
- [Installation avec le driver SafeNet](#)
- [Installation avec OpenSC](#)
- [Configuration](#)
- [Mode d'accès](#)

Pré-requis

Installation des paquets :

- pcsc-tools
- libpcsclite1
- libpcsclite-dev

Installation avec le driver SafeNet

Pour le matériel de type clé Feitian et carte sim Gemalto (comme fournis par Certinomis par exemple), les pilotes sont à télécharger ici : <https://support.globalsign.com/ssl/ssl-certificates-installation/safenet-drivers#Linux%20Debian>

Une fois le paquet installé, on obtient le fichier `/lib/pkcs11/libIDPrimePKCS11.so` qui est le pilote qui sera utilisé par esup-signature ci-après...

La configuration à mettre dans le fichier `application.yml` :

```
seal-certificat-type: PKCS11
seal-certificat-driver: /lib/pkcs11/libIDPrimePKCS11.so
seal-certificat-pin: *****
```

Le premier permet de préciser le type de certificat à utiliser, le deuxième pointe pilote de la clé USB, le dernier correspond au code pin du certificat



Il est possible d'utiliser un certificat PKCS12 en guise de certificat cachet. Pour cela **seal-certificat-type** doit être PKSC12 et il faudra utiliser **seal-certificat-file** pour préciser l'emplacement du fichier .p12

Installation avec OpenSC

Afin d'éviter l'utilisation d'un pilote spécifique vous pouvez passer par OpenSC. L'installation et la vérification sont décrites ici : [OpenSC](#)

Configuration

à mettre dans le fichier application.yml :

```
seal-certificat-type: OPENSC  
seal-certificat-pin: *****
```

Mode d'accès

Cette signature électronique est disponible de trois manières :

- Les utilisateurs ayant obtenu le ROLE_SEAL ont la possibilité de signer avec le certificat d'établissement.
- Il est possible de configurer esup-signature pour signer automatiquement **tous** les documents en fin de circuits
- Enfin, il est possible de configurer la signature cachet automatique, circuit par circuit



Depuis la version 1.21, les utilisateurs ont la possibilité de signer avec un niveau de signature supérieur à celui exigé initialement pour signer le document