Technical Note: HSM

Version 1.02

System Requirements, Setup and Trouble Shooting

Contact: pdfsupport@pdf-tools.com

Owner: **PDF Tools AG**
Kasernenstrasse 1
8184 Bachenbülach
Switzerland
www.pdf-tools.com

Copyright © 2001-2014

1 Table of Content

2 Overview

2.1 Purpose

This document acts as a guide to help with the following subjects:

1. Prepare a HSM so it can be used with software of PDF Tools AG, such as the 3-Heights™ PDF Security or the 3-Heights™ PDF to PDF/A Converter.
2. In case the issue occurs during the installation, configuration or in production, there is a short trouble shooting guide.

2.2 HSM Vocabulary

Signature Cryptographic procedure to ensure the integrity and / or authenticity of a document. The signature is embedded in the PDF document in the form of a CMS (PKCS#7) message.

Certificate A certificate is an electronic confirmation of the identity of a natural or legal person.

Key The certificate contains a public key for the validation of the signature. The public key must match a private key, which is used for the creation of the signature.

Token A “container” (part of HSM, USB stick, smart card, etc.) that contains cryptographic objects or protects against unauthorized access.

Slot A “plug-in position” inside the HSM that holds a token. The Token must not be physically present instead it may be part of the HSM.

PIN A secrete number, which is required for the access of the token.

2.3 Abbreviations

CA	Certification Authority
CMS	Cryptographic Message Syntax
CRL	Certificate Revocation List
CSP	Cryptographic Service Provider
HSM	Hardware Security Module
OCSP	Online Certificate Status Protocol
PKCS	Public Key Cryptography Standards
QES	Qualified Electronic Signature
TSA	Time Stamp Authority
TSP	Time Stamp Protocol
PIN	Personal Identification Number

3 System Requirement

This chapter describes the requirements for the use of a HSM with the PDF Tools AG signature software. It is best to check them in the described order.

3.1 HSM

Certificate

For the creation of a signature a valid signing certificate must be installed. The certificate shall not be marked as private (*CKA_PRIVATE=FALSE*).

The German Federal Network Agency requires the following algorithms and key strengths for qualified digital signatures, valid until the year 2017.

Hash: SHA-256

RSA: 2048 Bit

We suggest therefore using certificates that meet these requirements.

Private Key

For the creation of a signature a private key is required. The key must be associated to the certificate (same value for *CKA_ID*). The private key must be marked as private (*CKA_PRIVATE=TRUE*), to ensure the signature can only be applied in combination with providing a PIN. The key must be suitable for signature creation (*CKA_SIGN=TRUE*).

The private key is not required for signature validation.

Trust Chain

Embedding of the trust chain in the signature requires all certificates of the issuer (certificate authority) up to and including the root certificate. They must be installed in the same slot as the signing certificate itself. Certificates must not be marked private (*CKA_PRIVATE=FALSE*) and the subject attribute (*CKA_SUBJECT*) should be set to the certificate's subject.

For signature validation all certificates in the keystore are regarded as trusted.

Slot

The number of the slot must be known. It will be required by the signature software.

Test Certificate

A test certificate should be available, which can be used to test the signature software.

Example: Setup SafeNet Luna SA Using the CMU Tool

Generate a new key pair:

```
cmu gen -modulusBit=2048 -publicExp=65537 -sign=1 -verify=1 -labelPublic="Public Key
XXX" -labelPrivate="Private Key XXX"
```

Create a new CSRs. Use the handles of the newly created keys:

```
cmu requestCert -privatehandle=XXX -publichandle=XXX -C=CH -OU="My Unit" -O="My
Organization " -CN="Signing Certificate x" -outputFile=cert.req
```

Submit the CSR to your certificate authority (CA). Download your new certificate and all certificates of the trust chain.

Import the signing certificate into the keystore:

```
cmu import -inputFile=certificate.cer -label="Signing Certificate XXX"
```

List all objects and their handles:

```
cmu list -display=handle,class,label
```

Associate the signing certificate with its private key using the *CKA_ID* attribute. First, a unique ID must be chosen. For example, the certificate’s fingerprint is suitable:

```
openssl x509 -in certificate.cer -fingerprint -noout | sed -e 's/://g'
```

Set the ID of the certificate:

```
cmu setAttribute -handle=XX -id=3c62ddcee426701f1fae3fdc690f7d89ffe18326
```

Set the ID of the private key:

```
cmu setAttribute -handle=XX -id=3c62ddcee426701f1fae3fdc690f7d89ffe18326
```

The private key is not needed anymore and may be deleted:

```
cmu delete -handle=XX
```

Import all certificates of the trust chain:

```
cmu import -inputFile=ca1.cer -label="CA Certificate XX"
cmu import -inputFile=ca2.cer -label="CA Certificate XX"
```

3.2 Installation PKCS#11 Client

PKCS#11 Interface

The HSM must support the PKCS#11 interface. The manual of the manufacturer states if this interface is supported. The interface is normally provided in the form of a library (DLL on Windows, or shared object on Unix systems) as part of the client software.

Installation Client Software

The client software of the HSM must be installed on the same computer as where the signature software is used. The client software also installs a DLL for the PKCS#11 interface. The name of the library, e. g. cryptoki.dll and the path on the file system must be known for the configuration of the signature software.

3.3 HTTP Access, Proxy Server, Firewall

HTTP Access

For the application of a time stamp or an online verification of certificates, the signature software requires access to the server of the issuer (e. g. *http://ocsp.quovadisglobal.com* or *http://platinum-qualified-g2.ocsp.swisssign.net/*) via HTTP. The URL for verification is stored in the certificate; the URL for time stamp services is provided by the issuer. In case these functions are not configured, no access is required.

Proxy Server

In organizations where a web proxy is in used, it must be ensured that the required MIME types are supported. These are:

```
application/ocsp-request

application/ocsp-response

application/timestamp-query

application/timestamp-reply
```

Firewall

In case no web proxy server is used, it must be ensured the HTTP requests and responses can pass the firewall.

3.4 Configuration of the Signature Software

Slot

The number of the slot that holds the certificate must be known prior to the configuration.

PCKS#11 DLL

The name and path of the library (DLL) must be known prior to the configuration (see also Installation HSM Client Software).

Provider

The parameter Provider is a string consisting of three sub parts, which are separated by semi columns. Example:

```
"C:\Program Files\SafeNet\Protect Toolkit C SDK\bin\hsm\cryptoki.dll;0;123456"
```

The first part defines the name including path of the PKCS11 library (DLL), the second is the slot number and the third is the PIN. The first two values are always required. The PIN is only required for applying signatures and can be omitted with verifying signatures.

Online Verification

In order to execute an online verification, the parameter `EmbedRevocationInfo` must be set to true (default). This parameter only takes effect if the certificate supports OCSP and / or CRL requests.

Time Stamp

In case a time stamp is required, the parameter `TimeStampURL` und optionally `TimeStampCredentials` must be set to values that are provided by the issuer of the certificate.

Web Proxy

In an environment where a web proxy server is in use and an online verification or time stamp is required, the parameter `ProxyURL` and optionally `ProxyCredentials` must be set to values valid for the organization.

4 Trouble Shooting

4.1 Return Values versus Error Codes

Many functions in the PDF Tools AG' software return a Boolean value.

- The return value indicates whether the function generally completed successfully or not.
- The property `ErrorCode` provides additional information about the result of the previous function call.

The return value is more important than the error code. A function that returns false always indicates an error. If the return value is true and in error code is different from 0, the error code describes a warning, which does not invalidate the process per se (e. g. applying a digital signature), but may still be of importance (e. g. OSCP server not available).

Return values should be verified in particular for the following functions:

```
BeginSession

AddSignature

SaveAs
```

The error code is an enum. A complete list of all errors is available as part of the documentation of the software.

C	`pdferror.h`
Java	`NativeLibrary.java.ERRORCODE`
.NET	`libpdfNET.Pdftools.Pdf.PDFErrorCode` (nur Liste)

Error messages and possible reasons in relation with HSM are described in the next chapter.

4.2 Error Codes and Possible Reasons

SIG_E_SESSION (0x8A130001)

- PKCS#11 library (e.g. DLL) not found
- The platform of the library is different to the application's
- The library does not have a PKCS#11 interface
- Initialization of the library failed due to too many applications and / or threads access the library concurrently
- The slot number is invalid
- The PIN is incorrect

SIG_E_STORE (0x8A130002)

- This error does not occur in combination with PKCS#11 (MS CryptAPI only)

SIG_E_CERT (0x8A130003)

- Certificate not found in the defined slot number

SIG_E_OCSP (0x8A130004), SIG_E_TSP (0x8A130005)

- Failed to establish an HTTP connection (see requirements)
- The server of the issuer is not available

SIG_E_PRIVKEY (0x8A130006)

- The private key is not installed in the slot number or does not match the certificate
- Die PIN is incorrect
- The signature algorithm in the certificate is unknown

PDF_E_SIGVAL (0x85410002)

- The provider name is invalid when starting the session