3-Heights™ Signature Creation and Verification Service

Version 4.5

User Manual

Contact: pdfsupport@pdf-tools.com

Owner: **PDF Tools AG**

Kasernenstrasse 1
8184 Bachenbülach
Switzerland

http://www.pdf-tools.com

1 Table of Content

2 Introduction

2.1 Overview

The *3-Heights™ Signature Creation and Verification Service* provides HTTP protocol based remote access to cryptographic providers such as smartcards, USB tokens, and other cryptographic infrastructure such as HSMs. By means of this service the tokens can be hosted centrally and used by any client computer which has access to the service.

The service is configurable to handle multiple tokens and is secured via credentials. While the service is running on a Windows computer, its clients can access it also from other platforms such as UNIX.

PKCS#11 is a widely used standard for providing extensive support in the area of digital signatures, including cryptographic algorithms and storage for certificates and keys.

The *3-Heights™ Signature Creation and Verification Service* relies on the PKCS#11 infrastructure for creating and verifying digital signatures. It constitutes the preferred infrastructure when dealing with hardware tokens and hardware security modules (HSMs).

2.2 Advantages

Using the *3-Heights™ Signature Creation and Verification Service* has several advantages over the direct use of client software:

Hosted Tokens

By means of the *3-Heights™ Signature Creation and Verification Service* personal tokens of employees may be hosted in a secure location and can be used remotely from any client computer which has access to the service by using individual credentials. The tokens may also be stored in a hardware security module (HSM).

Platform support

The *3-Heights™ Signature Creation and Verification Service* uses a HTTP interface. This enables signature support for platforms that are otherwise not supported by the cryptographic infrastructure.

Restricted Intranet Access

The creation of a digital signature requires access to the servers of the certificate authority (CA) to be able to query the status of a certificate (OCSP or CRL) and optionally access to the servers of a time stamp authority (TS) to create trusted time stamps (TSP).

With the *3-Heights™ Signature Creation and Verification Service* these functions are centralized on a server and are not performed by the client any more. Thus, internet access is not required by the client computers and may be restricted to a dedicated server.

Robustness

The fact that the signature creation and verification is done in a separate process greatly increases the robustness of the client application.

If the cryptographic middleware produces a crash, only the respective worker process is terminated. The *3-Heights™ Signature Creation and Verification Service* and the client application remain untouched.

3 Installation and Configuration

3.1 Requirements

Operating system

The *3-Heights™ Signature Creation and Verification Service* is available for the following operating systems:

- Windows XP, Vista, 7, 8, 8.1 - 32 and 64 bit
- Windows Server 2003, 2008, 2008 R2, 2012, 2012 R2 - 32 and 64 bit

PKCS#11 Cryptography Provider

The middleware of the cryptographic infrastructure (USB Token, HSM) must be installed on the same computer as where the *3-Heights™ Signature Creation and Verification Service* runs. The middleware 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.

The following providers have been tested for interoperability with the *3-Heights™ Signature Creation and Verification Service*:

- SafeNet Protect Server (cryptoki.dll)
- SafeNet Luna (cryptoki.dll)
- SafeNet Authentication Client (eTPKCS11.dll)
- CryptoVision (cvp11.dll)
- Siemens CardOS
- IBM OpenCrypTokI (opencryptoki.dll)

Client Software

The *3-Heights™ Signature Creation and Verification Service* can be used by any signature-aware *3-Heights™* client software in particular with the following client software:

- 3-Heights™ Security Tool
- 3-Heights™ PDF to PDF/A Converter
- 3-Heights™ Document Converter

3.2 Installation

Two Windows Installer kits are available for 32-bit and 64-bit systems. Select the kit that matches your platform architecture. The following steps apply to the 64-bit and are similar for the 32-bit variant.

1. Download the ZIP archive e.g. *SIGSVC450x64.zip* from your download account at www.pdf-tools.com
2. Extract the file *3-Heights(TM) Signature Creation and Verification Service (x64).msi* from the ZIP archive.
3. Double-click the MSI file to start the installation wizard.
4. Follow the installation wizard. There are no installation options.

The installation automatically adds the *3-Heights™ Signature Creation and Verification Service* and sets it to automatic start. After the installation the service must be started manually, however. Upon un-installation, the service is stopped and removed.

3.3 Service Configuration

Configuration files

The service configuration of the *3-Heights™ Signature Creation and Verification Service* is done by editing the configuration files *TokenConfig.xml* and *SignatureService.exe.config*. The files must reside in the same directory where the executable *SignatureService.exe* is. The first file is used to configure the cryptographic tokens and the latter to configure the properties of the service itself.

XML structure of TokenConfig.xml:

- **<configuration>**
  - **ID**: The unique identifier of the cryptographic provider.
  - **ProviderString**[1]: A string to identify and access a cryptographic token. The attributes in the provider string are separated by a semicolon. The attributes are:
    - location of the PKCS#11 interface DLL
    - slot number
    - user PIN
  - **Password**: The password which is used by the client software to access the token.

Example of TokenConfig.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <add ID="0001" ProviderString="c:/Program Files (x86)/SafeNet/Protect Toolkit
C SDK/bin/sw/cryptoki.dll;0;123456" Password="pass01"/>
  <add ID="0002" ProviderString="cvp11.dll;1;123456" Password="pass02"/>
</configuration>
```

[1] A more detailed description of the ProviderString can be found in the manual of the 3-Heights™ PDF Security API in the description of the property Provider of the interface PdfSignature.

XML structure of SignatureService.exe.config:

- **`<configuration>`**
- **`<appSettings>`**
    - **`add`**: Add a key / value pair to the property bag. The following keys are supported.
        - **`Port`**: The IP port number on which the service is listening.
        - **`MaxResponseLenght`**: The maximum buffer size for response data.
        - **`RequestBufferSize`**: The buffer size for receiving request chucks.
        - **`LogFile`**: The path to a verbose log which is written by the service. If empty logging is disabled.
        - **`TokenConfigFile`**: The path to the XML configuration file. If empty, the server looks for a file named *TokenConfig.xml* in the installation directory.

Example of SignatureService.exe.config

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <add key="Port" value="8080"/>
    <add key="MaxResponseLength" value="20000"/>
    <add key="RequestBufferSize" value="4096"/>
    <add key="LogFile" value=""/>
    <add key="TokenConfigFile" value="" />
  </appSettings>
</configuration>
```

3.4 Client configuration

Once you have the service configured and running, it can be accessed from any signature-capable 3-Heights™ product by specifying a provider string of the form

```
“http://server.mydomain.com:8080/0001;pass01"
```

- *server.mydomain.com* is the network name of the computer hosting the service
- *8080* designates the TCP/IP port that is configured the *SignatureService.exe.config* file
- *0001* designates the *"ID"* entry in the *TokenConfig.xml* file for the selected token
- *pass01* stands for the password that is configured for the selected token

3.5 Service Execution

The service is registered as a Windows service during installation. However, there is no obligation to execute the service as a Windows service. It can also run in a command line window. Either way has its advantages and disadvantages, depending on the following criteria:

- Console: you can easily verify that the smartcard infrastructure is available. This may be quite difficult in the service environment. Also, you can easily monitor the activities of the service.
- Service: the service will automatically start up when the computer is started, without the need to perform an interactive login

When deciding for interactive use, change the startup mode of the windows service to "manual" or "disabled".

4 Glossary

4.1 Technical Terms

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

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

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

Private Key The private key is used to create the digital signature. It is contained on a cryptographic token and is protected against unauthorized access.

Token A “container” (part of HSM, USB stick, smart card, etc.) that contains cryptographic objects such as certificates and private keys which are protected against unauthorized access.

Slot A logical address of a USB-Token or 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 secret number, which is required to access the token. There are User PINs and Administrator PINs. The first allows for creating digital signatures and the latter for managing the cryptographic objects in the token.

4.2 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 Standard
QES	Qualified Electronic Signature
TSA	Time Stamp Authority
TSP	Time Stamp Protocol
PIN	Personal Identification Number

5 Trouble Shooting

5.1 Additional Documentation

There are two technical notes which cover the following special topics:

- Technical Note on HSMs: *www.pdf-tools.com/public/downloads/manuals/TechNoteHSM.pdf*
- Technical Note on PKCS#11: *www.pdf-tools.com/public/downloads/manuals/TechNotePKCS11.pdf*

5.2 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.

5.3 Usage of certificates from the Windows Certificate Store

Soft certificates and other certificates stored in the Windows Certificate Store can be used with the *3-Heights™ Signature Creation and Verification Service* as well. For this, a token can be used with a *ProviderString* configuration of the Microsoft Crypt API provider. The default for which is the empty string *ProviderString=""*.

Clients using the Crypt API token must set the provider session property *MessageDigestAlgorithm* to *SHA-1*.

Special care must be taken that the *3-Heights™ Signature Creation and Verification Service* a session and under a user that has access to the signing certificate (see chapter

Service Execution)

5.4 Error Codes and Possible Reasons

SIG_E_SESSION (0x8A130001)

- PKCS#11 library (e.g. DLL) not found
- 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
- Die slot number is invalid
- Die PIN is incorrect

SIG_E_STORE (0x8A130002)

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

SIG_E_CERT (0x8A130003)

- No certificate 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
- The message digest algorithm sent by the client is not supported by the token

PDF_E_SIGVAL (0x85410002)

- The provider name is invalid when starting the session