Venafi
CodeSign Protect

Integration Guide

Utimaco HSM

utimaco®

Imprint

Copyright 2020	Utimaco IS GmbH Germanusstr. 4 D-52080 Aachen Germany
Phone	AMERICAS +1-844-UTIMACO (+1 844-884-6226) EMEA +49 800-627-3081 APAC +81 800-919-1301
Internet	https://support.hsm.utimaco.com/
e-mail	support@utimaco.com
Document Version	1.0.0
Date	06/10/2025
Status	PUBLSIHED
Document No.	IG-2025-0024

Table of Contents

1 Introduction

This guide is part of the information and support provided by Utimaco. Additional documentation produced to support your Utimaco SecurityServer product can be found in the document directory of the Utimaco SecurityServer product bundle. All Utimaco SecurityServer product documentation is available from Utimaco's web site at https://utimaco.com/

1.1 About This Guide

This guide provides an integration explaining how to integrate an Utimaco CryptoServer Hardware Security Module (HSM) with Venafi CodeSign Protect.

1.1.1 Target Audience for This Guide

This guide is intended for administrators of Venafi CodeSign Protect and of Utimaco HSMs.

1.1.2 Contents of This Guide

After the introduction this guide is divided up as follows:

Chapter 2 Overview

Chapter 3 Integration Requirements and Prerequisites

Chapter 4 Software Download and Installation

Chapter 5 PKCS#11 Configuration

Chapter 6 Integrating Venafi Platform with Utimaco HSM

Chapter 7 Troubleshooting *Chapter 8* Further Information

1.1.3 Document Conventions

The following conventions are used in this guide:

Convention	*Use*	*Example*

Bold	Items of the Graphical User Interface (GUI), e.g., menu options	Press the OK button.
`Monospaced`	File names, folder and directory names, commands, file outputs, programming code samples	You will find the file example.conf in the /exmp/demo/ directory.
Italic	References and important terms	See Chapter 3, "Sample Chapter", in the *CryptoServer - csadm Manual* or [CSADMIN].

Table 1: Document conventions

Special icons are used to highlight the most important notes and information.

Here you find important safety information that should be followed.

Here you find additional notes or supplementary information.

1.1.4 Abbreviations

The following abbreviations are used in this guide:

Abbreviation	*Meaning*
API	Application Programming Interface
CA	Certificate Authority

CD	Compact Disc
CSADM	CryptoServer Command-line Administration Tool
CSAR	Cloud Service Architecture
CSP	Cryptographic Service Provider
DLL	Dynamic-Link Library
DN	Domain Name
GP HSM	General Purpose Hardware Security Module
Abbreviation	*Meaning*
GUI	Graphical User Interface
HSM	Hardware Security Module Hardware Security Module
IIS	Internet Information Services
IP	Internet Protocol
KSP	Key Storage Providers
LAN	Local Area Network
MBK	Master Backup Key

PCIe	PCI Express Interface
PIN	Personal Identification number
PKCS#11	Public-Key Cryptography Standard #11
REST	Representational State Transfer
RSA	Rivest, Shamir, Adleman (cryptosystem)
SO	Security Officer
TLS	Transport Layer Security
TPP	Trust Protection Platform
URL	Uniform Resource Locator

Table 2: List of Abbreviations

1.2 Utimaco CryptoServer HSM

CryptoServer is a hardware security module developed by Utimaco IS GmbH. CryptoServer is a physically protected specialized computer unit designed to perform sensitive cryptographic tasks and to securely manage as well as store cryptographic keys and data. It can be used as a universal, independent security component for heterogeneous computer systems.

2 Overview

2.1 Venafi Code Signing

Software runs your business and represents your brand. Protect your company's software with a secure code signing process that is fast and easy for your developers to use. Venafi CodeSign Protect secures your code signing private keys, automates approval workflows, and maintains an irrefutable record of all code signing activities.

CodeSign Protect helps you ensure that private keys never leave a secure location. You can store all your enterprise's private keys in the Utimaco HSMs and restrict access to authorized users and use cases.

CodeSign Protect helps you manage all code signing private keys and automatically enforce fast, easy code signing processes for your development teams. It reduces the risk of your code signing credentials getting into the hands of cybercriminals.

3 Integration Requirements and Prerequisites

Ensure the system environment you will be using meets the following hardware and software requirements.

This guide assumes that the user has already installed and configured required Software.

3.1 Tested Versions

The integrations that have been successfully tested with the Utimaco HSM with Venafi Code Signing.

Operating System	*Venafi Code Sign Protect Version*	*Utimaco Security Server Version*	*Utimaco HSM*
Windows Server 2016	Venafi TPP - 22.1.1.3113	SecurityServer V4.45.3.0	CryptoServer CSe-Series/Se-Series u.trust Anchor Se*k and u.trust Anchor CSAR

Table 3: List of Tested versions

3.2 Software Requirements

Software	*Software Requirements*
Java	Version 8, Update 271 or higher
Venafi Code Signing	Venafi TPP - 22.1.1.3113
HSM Utility	SecurityServer/ CyrptoServer Administration (csadm)

HSM Utility	SecurityServer PKCS#11 Tool (p11tool2)
HSM Interfaces	SecurityServer PKCS#11 Provider

Table 4: List of Software Requirements

3.3 Hardware Requirements

Hardware	*Hardware Requirements*
Utimaco LAN HSM	CryptoServer CSe-Series/Se-Series LAN with firmware SecurityServer 4.45.3.0 or higher u.trust Anchor Se*k and u.trust Anchor CSAR with firmware 4.45.3.0 or higher
Utimaco PCI-e HSM	CryptoServer CSe-Series/Se-Series PCI-e with firmware SecurityServer 4.45.3.0 or higher

Table 5: List of Hardware Requirements

Setup an account on the Utimaco support portal and request download access at the following URL. https://support.hsm.utimaco.com

3.4 Prerequisites

The user must first deploy the Venafi Code Signing system before following the steps in this guide.

- The user with admin privileges to the Venafi Code Signing Server.

- Java version 8 or higher
- CryptoServer Admin should be replaced with new admin
- MBK created
- Utimaco CryptoServer HSM is setup and configured. Refer the CryptoServer documentations to setup the HSM

4 Software Download and Installation

This section describes the process of installing Utimaco HSM software with the PKCS#11 Provider.

4.1 Download Utimaco Software

If you have not already done so, please create and request an Utimaco Support Portal Account. This will allow you to download the software components needed for this installation. Copy the downloaded software at the appropriate location on the Venafi TPP Server.

5 PKCS#11 Configuration

On windows, as part of CryptoServer software installation, cs_pkcs11_R3.cfg will get automatically created and will be available under "C:\ProgramData\Utimaco\PKCS11_R3" folder.

Create `pkcs11.cfg` file at location `C:\ProgramData\Utimaco\PKCS11_R3` and add the contents as listed below

›_ pkcs11.cfg

```
name=CryptoServer
library=C:/Program Files/Utimaco/SecurityServer/Lib/cs_pkcs11_R3.dll
slot=0
attributes=compatibility
attributes(*,*,*) = {
CKA_TOKEN = true
}
```

For more information regarding the commands and command parameters please check the Utimaco documentation. The device may be a CryptoServer (PCIe or LAN) device. The device line will follow one of these patterns, based on the HSM form-factor:

`Device = 288@<HSM IP address> Hardware (LAN) HSM`

OR

`Device = /dev/cs2.0 Hardware (PCIe) HSM`

Example values

cs_pkcs11_R3.cfg

```
[Global]

# Path to the logfile (name of logfile is attached by the API)

# For unix:

# Logpath = /tmp

For windows:

Logpath = C:/ProgramData/Utimaco/PKCS11_R3

# Loglevel (0 = NONE; 1 = ERROR; 2 = WARNING; 3 = INFO; 4 = TRACE)

Logging = 4

[CryptoServer]

# Device specifier

Device = 192.168.10.10
```

To make your testing easier, it would be good to enable the PKCS#11 log file. That can be enabled by editing the Logging *Loglevel. Set the* LogPath *and Logging* Loglevel *to* 1*. For testing you may want to increase it to* 4*.*

The added LogPath *points to a writable directory, not to a file.*

If you encounter problems, check the log file named cs_pkcs11_R3.log *in the* LogPath *defined directory. When you are done testing, you should change Logging to* 1 *or* 2*. This will limit the logging to only critical and important messages.*

6 Integrating Venafi Code Sign Protect with Utimaco HSM

6.1 Enabling the Venafi Advanced Key Protect

To setup the key generation and store the cryptographic keys, the Venafi Advanced Key Protect is used. The steps to enable Venafi Advanced Key Protect are as follows;

1. Open the Venafi Configuration Console, then click on Connectors from the left pane
2. Click Enable Advanced Key Protect from the Actions panel
3. Verify the details in the dialog boxes, Click Confirm
4. Restart the IIS service by selecting the Website from the Product node
5. Select the Venafi Platform service, Click Restart
6. Select Logging service, Click Restart

*For more information refer **Venafi Advanced Key Protect** documentation.*

6.2 Creating the HSM Connector for Utimaco

The steps to create the HSM connector, are shown below

1. Open the Venafi Configuration Console, then click on Connectors from the left pane
2. Click on the Create HSM Connector in the Actions pane
3. Enter the administration credentials for Venafi Trust Protection Platform, click OK
4. Enter the Name for connector, Set the Cryptoki Dll Path as - `C:\Program Files\Utimaco\SecurityServer\Lib\cs_pkcs11_R3.dll`, Select the Slot from dropdown list and, enter the User Type and Pin details from the Create new HSM Connector for Utimaco HSM, then Click Verify

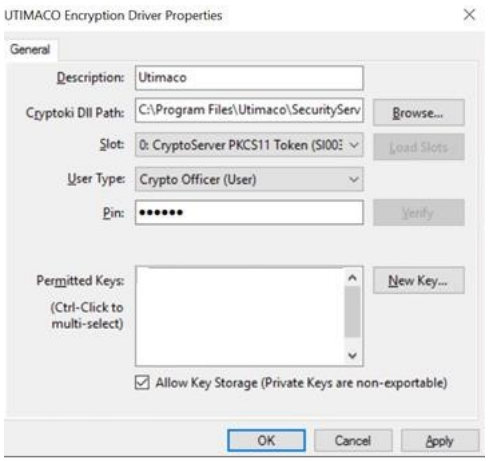

Figure 1: Create HSM Connector Window

5. Check the Allow Key Storage check box then click, Apply and OK
6. Now, under the Platform Connectors pane verify that the HSM connector appears

6.3 Setting up the Code Sign Protect to use Utimaco HSM Connector

Follow the steps to set up the Code Sign protect to use Utimaco HSM Connector.

6.3.1 Assigning the Code Signing Administrator

To assign the Code Signing Administrator:

1. Open the Venafi Configuration Console, click Administrator node
2. Click Add Code Signing Administrator, from the Actions panel
3. Click on the user you want to assign, click Select

6.3.2 Creating the Certificate Authority (CA) template

A CA template is required in each environment in a code-signing project. The user can create a self-signed CA template, a DigiCert CA template, or a Microsoft CA template. For more information refer to the Venafi Documentation.

6.3.3 Creating the Signing Flow

Flows in CodeSign Protect allow you to enforce actions that must take place when using signing keys or when deleting Projects or Environments. Venafi CodeSign Protect supports creating Code Signing Flow approvals based on a defined approver or approver group or based on the Project Owner and Key Use Approver roles associated with the Code Signing Project.

To create the Signing Flow:

1. In Custom Flows node, from Action Panel, click Add a new Code Signing Flow
2. Enter the name of the flow, click Create
3. Configure the flow by adding Approvers. For more information refer to the Venafi Documentation.

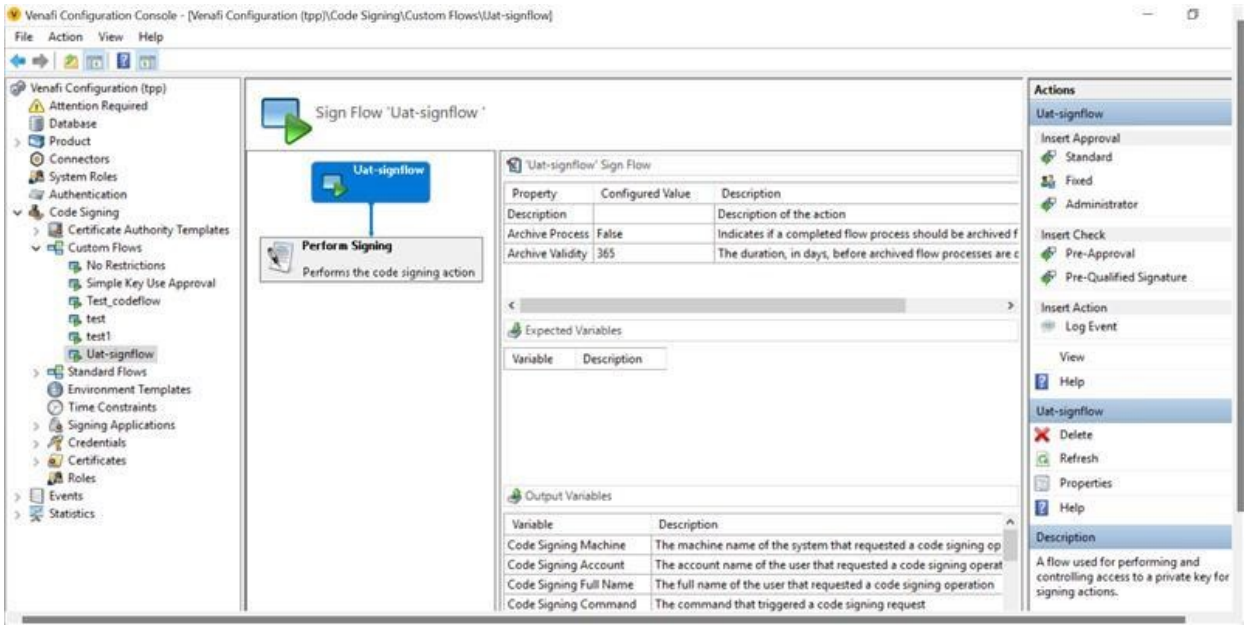

Figure 2: Create Signing Flow Window

6.3.4 Creating the Environment Template

Code Signing Environment Templates allow the Code Signing Administrator to suggest or require specific values to be used in CodeSign Protect Project Environments. Each project will include at least one environment. To create the Environment Template:

1. Select Environment Templates from the Venafi Code Signing node of the Venafi Configuration Console

2. From the Actions panel, Click Add Template
3. Enter the name of the template. The Development Properties wizard gets populated
4. In the Settings, enter the details for Description, Certificate Container and the Signing Flow created
5. In the Certificate Authority tab, enter the details for the CA template created
6. In the Keys tab, select the appropriate RSA key length values. This algorithm and key length appear as part of the certificate
7. In the Key Storage tab, click on the drop-down menu and select the HSM Connector created earlier for Utimaco HSM. Click, Add
8. The user can enter additional details, such as the Subject Domain Name of the certificate, in the remaining tabs

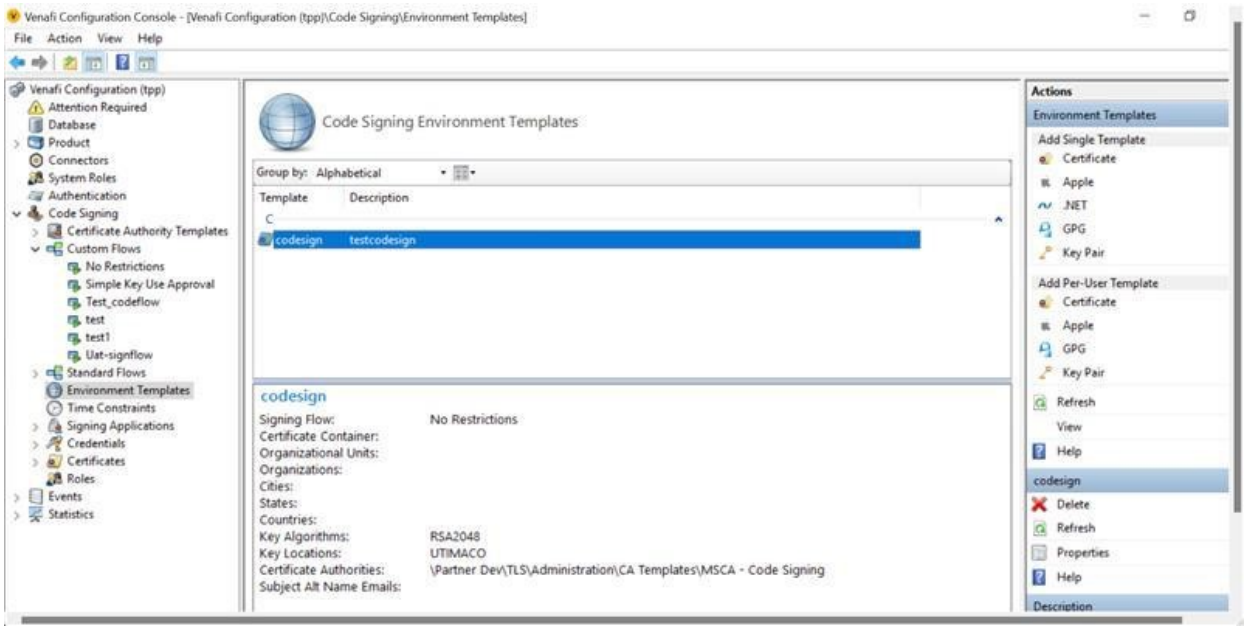

Figure 3: Code Signing Environment Templates Window

6.3.5 Creating the Code Signing Project

To Create the Code Signing Project:

1. Log into Aperture by going to `https://[IP_address_of_Venafi_TPP]/Aperture/codesigning`

2. To open the project configuration wizard, Click Add Project on the project list screen
3. Enter the details for Project Name and Description. Click Next
4. The Project window will be displayed in the Properties tab, add the appropriate individuals or groups to the Key User field
5. Now, in the Environment tab, Click on Add environment and from the drop-down select "Certificates and Keys for Authenticode based signing or any certificate based signing"

Further you can create new keys or certificates, or you can choose the existing ones.

6. To create an environment that generates a new certificate and private key,
    a. In the Environment Type drop-down, select the type of environment
    b. In the Certificate Provider drop-down list, select the appropriate certificate provider to associate with this environment.
    c. If only one certificate provider is assigned to this environment, that provider is automatically selected and the drop-down is not editable
    d. In the Environment Name box, enter a name for this environment
    e. Verify the Key Storage location points to HSM connector created earlier for Utimaco HSM
    f. Verify "Create New" radio button is selected
    g. Enter the remaining details
    h. Click, Create

Alternatively, if you want to use an existing key and certificate, skip step 6.

7. To create an environment that uses the existing key and certificate from the HSM,
    a. In the Environment Type drop-down, select the type of environment
    b. In the Certificate Provider drop-down list, select the appropriate certificate provider to associate with this environment.
    c. If only one certificate provider is assigned to this environment, that provider is automatically selected and the drop-down is not editable

d. In the Environment Name box, enter a name for this environment
e. Verify that the Key Storage location points to HSM connector created earlier for Utimaco HSM
f. Select "Use Existing Key in HSM" radio button
g. Select the Public HSM Key & Private HSM Key from the drop-down
h. Enter the remaining details
i. Click, Save

8. Click, Next
9. To create new certificate and private key on approval, Click Submit for Approval
10. To use existing key or certificate instead, click Save, if the project is already approved

6.3.6 Approving the Code Signing Project

After the code-signing project is submitted for approval, the Code Signing Administrators receive an email informing them that a project is ready for review. The Code Signing Administrators needs to follow these steps for reviewing and approving the code-signing project:

1. Sign into Aperture at `https://[IP_address_of_Venafi_TPP]/Aperture/codesigning`
2. In the Code Signing menu, click on Approvals and then select Pending Approvals
3. Click Approve for the Code Signing Project created. If you have selected to generate new key pair on HSM, the keys are created

This completes the configuration for Venafi Code Signing Project.

6.3.7 Installing and Configuring the Venafi Code Signing Client on Windows

Venafi Code Signing Client provides the Venafi CSP which is one of the ways to link Windows code signing workstations to the Trust Protection Platform server, which securely stores the private code signing keys inside HSM and manages its use. The Venafi CSP communicates with the Trust Protection Platform server over a TLS-encrypted REST API. The Venafi CSP supports both CSP and KSP and currently supports only RSA certificates.

Complete the below steps for installing and configuring the Venafi CSP:

1. Obtain Venafi Code Signing client package for windows from https://download.venafi.com and install it on your machine
2. Navigate to C:\Program Files\Venafi\MMC and Run Venafi Csp Configuration.msc
3. On the Welcome screen, if you already have an answer file, select whether you want to use it for this installation. Click Next.
4. On the Before You Begin screen, verify that you have all the information you need to complete installation
5. On the Host URLs screen, enter the addresses for your Authentication server and your HSM server.
6. Click Next.
7. On the Access Authorization screen, enter your Trust Protection Platform Key User and password. Check whether you want to enable access for the Current User only, Local Machine only, or both.
8. Click Finish
9. Open `C:\Program Files\Venafi CodeSign Protect\MMC\Venafi Csp Configuration.msc` if not opened.

Now the associated certificates to an user are visible in Venafi CSP Configuration Console.

These certificates can be used with signtool, jarsigner or any other tools that uses CAPI/CNG/KSP for signing.

›_ Console

```
signtool.exe sign /n "<certifcate_name>" c:\temp\myfile.exe
```

6.3.8 Installing and Configuring the Venafi Code Signing Client on Linux

To install and configure the Venafi Code Signing Client on Linux Platform and use PKCS#11 driver:

1. Obtain Venafi Code Signing Client Package for Linux from https://download.venafi.com and install it on your machine
2. Once the package is installed, verify all the tools are available in /usr/local/bin

›_ Console

```
# ls -l /usr/local/bin/
```

3. Now set the Venafi TPP Authentication URL and HSM URL with the below command

›_ Console

```
# pkcs11config seturl --authurl=Error! Hyperlink reference not valid. --
hsmurl=Error! Hyperlink reference not valid. --username=<username> --
password=<password> --force
```

4. Create the trust by adding the server TLS certificate to local trust store. This ensures that application is communicating only with the trusted servers

›_ Console

```
# pkcs11config trust --hsmurl=https:// <IP-Venafi-Server>/vedhsm
```

5. To verify trust setting run the below command

›_ Console

```
# pkcs11config trust --show
```

6. Run the below GetGrant command to check if the user credentials are valid. This command queries to Authentication Server and returns Grant, which is further stored in venafipkcs11config file.

›_ Console

```
# pkcs11config getgrant --username=username--password=password
```

7. Further, verify the validity of a grant using the checkgrant command

›_ Console

```
# pkcs11config checkgrant
```

8. To verify the configuration and list the available Code Signing Certificates run the below command.

›_ Console

```
# pkcs11config listobjects
```

This completes the Venafi PKCS#11 driver configuration.

6.3.9 Signing Code with Jarsigner using Venafi CodeSign Protect

For signing the Code using Code Signing Certificate, we will use Jarsigner as a signing tool.

Before proceeding ahead make sure to install and configure - Venafi Code Signing Client on Linux.

Follow the steps to setup the environment for jarsigner.

1. Create `venafipkcs11.conf` file in any directory and add the below entries in the file

›_ pkcs11.cfg

```
name = VenafiPKCS11
library ="/opt/venafi/codesign/lib/venafipkcs11.so"
slot = 0
```

2. Edit the `java.security` file of your installed java application and add the below entries

›_ java.security

```
#
# List of providers and their preference orders (see above):
#
security.provider.1=sun.security.provider.Sun
security.provider.2=sun.security.rsa.SunRsaSign
security.provider.3=sun.security.ec.SunEC
security.provider.4=com.sun.net.ssl.internal.ssl.Provider
security.provider.5=com.sun.crypto.provider.SunJCE
security.provider.6=sun.security.jgss.SunProvider
security.provider.7=com.sun.security.sasl.Provider
security.provider.8=org.jcp.xml.dsig.internal.dom.XMLDSigRI
security.provider.9=sun.security.smartcardio.SunPCSC
security.provider.10=sun.security.pkcs11.SunPKCS11
/root/venafipkcs11.conf
```

3. Now sign the jar file using the below command

›_ Console

```
# jarsigner \
-verbose helloworld.jar Test-uat-certificate \
-keystore NONE \
-storetype PKCS11 \
-certs \
-storepass none \
-providerclass sun.security.pkcs11.SunPKCS11 \
-providerArg /root/venafipkcs11.conf
```

4. To verify the signed jar file run the below command

›_ Console

```
# jarsigner -verify \
-keystore NONE \
-storetype PKCS11 \
-storepass none \
-providerclass sun.security.pkcs11.SunPKCS11 \
-providerArg /root/venafipkcs11.conf  helloworld.jar
```

7 Troubleshooting

Error	*Diagnosis*
The Underlying Connection was closed. Could not establish trust relationship for the SSL/TLS secure channel. (Validating 'https://venafittpserver.com/vedauth/')	1. Check if Certificate expired. 2. Check if there is proxy installed between CSP Machine and Venafi TPP Server.
You are trying to list available keys from signing on CSP Machine and do not get keys in return. User check of any error and below error displays on Venafi Server Log. tppadmin is an administrator and do not have access to Read Private Keys	Assign a different user than administrator to access the private
You are trying to add a Codesign certificate from CodeSign Protect by creating Project and you face below error Failed to add Private Key to KeyStore. Error: Failed to generate key pair due to exception: Call to C_GenerateKeyPair failed [UserNotLoggedIn]	1. Check if the HSM slot is initialized. 2. Check if the user is logged in. 3. Restart the Venafi Configuration Console. 4. Restart the IIS Service.
You are trying to add a Codesign certificate from CodeSign Protect by creating Project and you face below error Failed to add Private Key to KeyStore. Error: The selected engine 'UTIMACO' does not allow key storage (or AKP is not licensed.)	While creating the HSM connector, user need to check the "Allow Key Storage (Priavte Keys are non-exportable)" checkbox.

Table 6: List of Error and its Diagnosis

8 Further Information

This document forms a part of the information and support which is provided by the Utimaco IS GmbH. Additional documentation can be found on the product CD in the Documentation directory.

All CryptoServer product documentation is also available at the Utimaco IS GmbH website:

https://utimaco.com/

9 References

Reference	*Title/Company*	*Document No.*
[CSADMIN]	CryptoServer – csadm Manual/ Utimaco IS GmbH	2009-0003
[CSTrSh]	CryptoServer Troubleshooting/Utimaco IS GmbH	M011-0008-en
[CSADMIN2]	CryptoServer_csadm_Manual_Systemadministrators.pdf	2009-0003
[CSP11Tool2]	CryptoServer_p11tool2_Manual.pdf	2012-0004
[CSPKCSM]	CryptoServer - PKCS#11 P11CAT Manual	M013-0001-en
[CSLAN5]	CryptoServerLAN_Manual_Systemadministrators.pdf	2018-0004