enclaive vHSM

Integration Guide

Utimaco u.trust GP HSM

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	PUBLISHED
Document No.	IG-2025-0023

Table of Contents

1 Introduction

This guide is part of the information and support provided by Utimaco. Additional documentation produced to support your Utimaco u.trust General Purpose HSM product can be found in the document directory of the Utimaco u.trust General Purpose HSM product bundle.

All Utimaco u.trust General Purpose HSM product documentation is available from Utimaco's website at https://support.hsm.utimaco.com/ .

1.1 About this Manual

This manual explains how to integrate an Utimaco u.trust General Purpose Hardware Security Module (GP HSM) with enclaive Virtual HSM (vHSM).

1.1.1 Target Audience for this Manual

This guide is intended for administrators of enclaive vHSM and of Utimaco HSMs.

1.1.2 Contents of this Manual

After the introduction this guide is divided up as follows:

Chapter 2 gives a brief overview about enclaive vHSM and Utimaco HSM.

Chapter 3 describes the requirements and prerequisites for integrating enclaive vHSM with Utimaco HSM.

Chapter 4 shows how to install and confige Utimaco u.trust General Purpose HSM Software.

Chapter 5 guides you through the integration of enclaive vHSM with Utimaco HSM.

Chapter 6 describes how to test vHSM Integration with Utimaco HSM.

Chapter 7 explains how to enable entropy augmentation for random number generation.

Chapter 8 explains how to troubleshoot integration issues.

1.1.3 Document Conventions

Convention	***Use***	***Example***
Bold	Graphical User Interface (GUI), e.g., menu options	Press **OK**
`Monospaced`	Code that is given for explanation or as an example, file paths	`chsm-create`
Italic	References and important terms	See *Sample Chapter* in the *CryptoServer - Sample Manual*

Table 1: Document conventions

We use special icons 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.

This message marks the result expected after the successful execution of an instruction.

1.1.4 Abbreviations

We use the following abbreviations in this manual:

Abbreviation	Meaning
HSM	Hardware Security Module
HSMaaS	HSM as a Service
PKCS#11	Public-Key Cryptography Standard #11
SO	Security Officer
vHSM	Virtual HSM

Table 1: Document conventions

1.2 enclaive Virtual HSM

HSMs are physical devices specifically designed to safeguard cryptographic keys and sensitive data. They offer a higher level of security compared to software-based solutions because they are tamper-resistant and physically protected.

HSMs can have limitations in terms of flexible demand driven scalability. In cloud environments, scalability is a fundamental requirement, and adding more physical HSMs does not align with the dynamic needed in modern business applications.

enclaive Virtual HSM combines hardware security with software agility. vHSMs deliver the same level of trust and security anchored in hardware with the benefit of shifting functionality into enclaves. The trust is rooted in hardware by choosing the Utimaco u.trust GP HSM or HSM as a Service (HSMaaS) as an anchor, verify the integrity with enclaive's confidential boot and attestation technology.

1.3 Utimaco u.trust General Purpose HSM

u.trust General Purpose (GP) HSM is the next generation hardware security module platform developed by Utimaco IS GmbH. u.trust GP HSM is a physically protected specialized computer unit designed for true multi-tenancy and 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.

Utimaco u.trust GP HSM provisions randomness and unseals secrets for enclaive vHSM.

2 Integration Requirements and Prerequisites

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

2.1 Tested Versions

The following integrations of Utimaco u.trust GP HSM with enclaive vHSM have been successfully tested.

Operating System	enclaive vHSM Version	Utimaco u.trust GP HSM	Utimaco u.trust GP HSM firmware version
Ubuntu 24.04	1.4.3-2	u.trust GP HSM Se-Series	4.70.0
RHEL 10	1.4.3-2	u.trust GP HSM Se-Series	4.70.0
Oracle Linux UEK R8	1.4.3-2	u.trust GP HSM Se-Series	4.70.0
SLES 15 SP3	1.4.3-2	u.trust GP HSM Se-Series	4.70.0

Table 3: List of Tested Versions

2.2 Tested Cloud Provider

The integrations of Utimaco HSM with enclaive vHSM have been successfully tested with confidential VMs of the following Cloud provider.

Cloud Provider	**cVM instance**
Microsoft Azure	DCas-series, ECas-series

Cloud Provider	**cVM instance**
AWS	m6a-series, c6a-series
Google Cloud	n2d-series, c2d-series

Table 4: List of Tested Cloud Provider

We recommend confidential VMs with a minimum of 4 vCPUs, 8 GB RAM, and 10 GB persistent volume.

2.3 Software Requirements

The following software has been successfully tested for integrations of Utimaco HSM with enclaive vHSM.

Software	***Software Requirements***
HSM Utility	PKCS#11 Tool Version 2 (p11tool2)
HSM Interfaces	PKCS#11 Provider (cs_pkcs11_R3.so)

Table 5: List of Software Requirements

2.4 Hardware Requirements

The following hardware is supported for integrations of Utimaco u.trust GP HSM with enclaive vHSM.

Hardware	***Firmware Requirements***
u.trust General Purpose HSM Se-Series LAN HSM	Firmware 4.70.0.0 or higher

Hardware	***Firmware Requirements***
u.trust General Purpose HSM Se-Series PCIe HSM	Firmware 4.70.0.0 or higher
General Purpose HSM as a Service on Utimaco Trust as a Service (TaaS) marketplace	Firmware 4.70.0.0 or higher
AMD EPYC 3rd gen or newer	AMD SEV-SNP firmware 1.55.29 or higher
Intel Xeon 4th gen or newer	Intel TDX module version 1.5.06 or higher

Table 6: List of Hardware Requirements

2.5 Prerequisites

Before you begin, please ensure that you have installed/setup:

- Utimaco u.trust GP HSM is setup and configured. Refer the u.trust GP HSM
- documentation to setup the HSM.
- MBK must be created and stored onto the HSM. Refer the u.trust GP HSM documentation to setup the MBK.
- u.trust GP HSM Default Admin should be replaced with a new admin user.
- Operating system, Cloud provider, software and hardware as listed in sections above.
- Download and install Docker runtime engine, or related container runtimes.
- Download and install enclaive vHSM as described in enclaive vHSM documentation.

The steps for installation of enclaive vHSM are out of scope of this document. Please follow the link: https://docs.enclaive.cloud/virtual-hsm/documentation/setup/install for more information about enclaive vHSM installation and configuration.

3 Installing and Configuring Utimaco u.trust GP HSM Software

In the following sections the notation ›_ Console indicates that you have to type the command displayed in the following line after # into your command shell.

3.1 Download and Install Utimaco Software

If you have not already done so, please create and request an Utimaco Support Portal Account at https://support.hsm.utimaco.com/. This will allow you to download the software components needed for this installation.

Install the latest version of the u.trust GP HSM software as described in the administration manual for the HSM. We recommend that you uninstall any older software before installing the new software.

1. Copy the downloaded software at the appropriate location on the server.
2. Create a folder `utimaco` under `/opt` directory and further create two directories named `/opt/utimaco/bin` and `/opt/utimaco/lib`.
3. Copy pkcs11 library libcs_pkcs11_R3.so from Utimaco software to the `/opt/utimaco/lib` directory.
4. Copy the csadm and p11tool2 files from Utimaco software to `/opt/utimaco/bin` directory and make both files executable.

3.2 Update cs_pkcs11_R3.cfg

1. Locate the Utimaco PKCS#11 configuration file cs_pkcs11_R3.cfg in your u.trust GP HSM software in directory `Linux/x86-64/Crypto_APIS/PKCS11_R3/sample`. Copy the configuration file into `/opt/utimaco` directory.
2. Edit the cs_pkcs11_R3.cfg file and make the appropriate changes to the file.

Installing and Configuring Utimaco u.trust GP HSM Software

```
[Global]
# Path to the logfile (name of logfile is attached by the API)
# For Unix:
Logpath = /tmp

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

# Prevents expiring session after inactivity of 15 minutes
KeepAlive = true

# Set the Device to connect with
[CryptoServer]
# Device specifier
Device = <HSM_IP>
```

Figure 1 : cs_pkcs11_R3.cfg

The device may be a u.trust GP HSM PCIe or LAN device or u.trust GP HSMaaS. The device line will follow one of these patterns, based on the HSM form factor. Device = 4001@<HSM IP address> for LAN HSM or HSMaaS

OR

Device = /dev/cs2.0 for PCIe HSM

To make your testing easier, it would be good to enable the PKCS#11 log file. That can be enabled by editing the Logging log level and Logpath. Set Logging to 4 and specify the LogPath. The 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 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.

3.3 Create SO User and Initialize a Slot

You must initialize a slot using p11tool2. The slot may get a custom label by specifying the command line parameter "label=". A default label will be assigned when omitting the "label=" parameter.

First, use p11tool2 to create the SO or Security Officer. Then initialize the slot that you want to use and the slot user as shown below.

```
# p11tool2 slot=<slot_no.> Label=<token_label> Login=ADMIN,<ADMIN.key> InitToken=ask
```

```
# p11tool2 slot=<slot_no.> LoginSO=ask SetPin=ask,ask
```

```
# p11tool2 slot=<slot_no.> LoginSO=ask InitPin=ask
```

```
# p11tool2 slot=<slot_no.> LoginUser=ask SetPin=ask,ask
```

For more information regarding the commands and command parameters please check the Utimaco [p11tool2] documentation.

When integrating a physical HSM, you must use the default administrator key file ADMIN.key as shown in the console commands above, or the currently active administrator key file. When integrating the HSM simulator, you must use the default simulator key file ADMIN_SIM.key as shown in the console output below.

Mind the output of p11tool2. It will instruct you to enter the current or new PIN as needed.

```
C:\Program Files\Utimaco\SecurityServer\Administration>p11tool2 slot=0 label=vHSM Login=ADMIN,ADMIN_SIM.key InitToken=ask
Enter SO PIN:
Repeat SO PIN:
```

```
C:\Program Files\Utimaco\SecurityServer\Administration>p11tool2 slot=0 LoginSO=ask SetPin=ask,ask
Enter SO PIN:
Enter the old PIN:
Enter the new PIN:
Repeat the new PIN:
```

```
C:\Program Files\Utimaco\SecurityServer\Administration>p11tool2 slot=0 LoginSO=ask InitPin=ask
Enter SO PIN:
Enter normal user PIN:
Repeat normal user PIN:
```

```
C:\Program Files\Utimaco\SecurityServer\Administration>p11tool2 slot=0 LoginUser=ask SetPin=ask,ask
Enter normal user PIN:
Enter the old PIN:
Enter the new PIN:
Repeat the new PIN:
```

4 Unsealing enclaive vHSM with Utimaco u.trust GP HSM

In the following sections we describe how to configure the enclaive vHSM to unseal with the Utimaco HSM, rooting the trust in the physical security of the u.trust GP HSM.

enclaive Virtual HSM requires a valid license. To obtain one, please reach out to the enclaive sales team.

4.1 Create vHSM User and Group

1. Create the vhsm group

›_ Console

```
# groupadd vhsm
```

2. Create the vhsm User

›_ Console

```
# useradd --gid vhsm vhsm
```

4.2 Download the Image and the vHSM CLI

To run vHSM in a Docker container, you need to download a precompiled container image from enclaive's registry at https://harbor.enclaive.cloud/. To interact with enclaive vHSM, you'll also need to install the vHSM CLI tool.

1. Pull the latest vHSM container image.

›_ Console

```
# docker pull harbor.enclaive.cloud/vhsm/vhsm-utimaco/vhsm:latest
```

```
root@enclaive ~ # docker pull harbor.enclaive.cloud/vhsm/vhsm-utimaco:latest
latest: Pulling from vhsm/vhsm-utimaco
30a9c22ae099: Pull complete
186747b7b1b3: Pull complete
852f74dd4553: Pull complete
95228a8b75ac: Pull complete
192a5d04b35f: Pull complete
39d73a80dfcf: Pull complete
19a126bffd62: Pull complete
a172dd152953: Pull complete
ea30057401b7: Pull complete
bcb4a9d6cf68: Pull complete
3bf9c3b57aae: Pull complete
ff65128d420e: Pull complete
2587aa35237f: Pull complete
90fd167fd99c: Pull complete
3d9149b60339: Pull complete
Digest: sha256:0f3bd89d8b4d1f54dad8b8007721034f48a061cb5cc32c7862c6625dcea8cc41
Status: Downloaded newer image for harbor.enclaive.cloud/vhsm/vhsm-utimaco:latest
root@enclaive ~ %
```

2. Download the enclave vHSM CLI.

›_ Console

```
# wget https://vhsm.enclaive.cloud/static/vhsm
```

```
root@enclaive ~ % wget https://vhsm.enclaive.cloud/static/vhsm
--2025-02-11 20:08:21--  https://vhsm.enclaive.cloud/static/vhsm
Resolving vhsm.enclaive.cloud (vhsm.enclaive.cloud)... 185.112.181.201
Connecting to vhsm.enclaive.cloud (vhsm.enclaive.cloud)|185.112.181.201|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 255266968 (243M) [application/octet-stream]
Saving to: ‘vhsm’

vhsm                100%[===================>] 243.44M  1.81MB/s    in 2m 13s

2025-02-11 20:10:35 (1.83 MB/s) - ‘vhsm’ saved [255266968/255266968]
```

3. Confirm that vHSM CLI is installed.

›_ Console

```
# vhsm -h
```

```
root@enclaive ~ % vhsm -h
Usage: vhsm <command> [args]

Common commands:
    read        Read data and retrieves secrets
    write       Write data, configuration, and secrets
    delete      Delete secrets and configuration
    list        List data or secrets
    login       Authenticate locally
    agent       Start a vHSM agent
    server      Start a vHSM server
    status      Print seal and HA status
    unwrap      Unwrap a wrapped secret

Other commands:
    audit               Interact with audit devices
    auth                Interact with auth methods
    debug               Runs the debug command
    events
    kv                  Interact with vHSM's Key-Value storage
    lease               Interact with leases
    monitor             Stream log messages from a vHSM server
    namespace           Interact with namespaces
    nitride             Interact with the remote-attestation plugin
    operator            Perform operator-specific tasks
    patch               Patch data, configuration, and secrets
    path-help           Retrieve API help for paths
    pki                 Interact with vHSM's PKI Secrets Engine
    plugin              Interact with vHSM plugins and catalog
    policy              Interact with policies
    print               Prints runtime configurations
    proxy               Start a vHSM Proxy
    secrets             Interact with secrets engines
    ssh                 Initiate an SSH session
    token               Interact with tokens
    transform           Interact with vHSM's Transform Secrets Engine
    transit             Interact with vHSM's Transit Secrets Engine
    version-history     Prints the version history of the target vHSM server
```

4.3 Install the vHSM Enterprise License

To enable HSM functionality you must have a vHSM Enterprise license.

1. Create a directory for the vHSM configuration file, data, and certificates

›_ Console

```
# mkdir /etc/vhsm
```

```
# mkdir /etc/vhsm/data
```

```
# mkdir /etc/vhsm/certs
```

2. Set the value for the environment variable `ENCLAIVE_LICENCE` to the license key.

›_ Console

```
# export ENCLAIVE_LICENCE=<license_key>
```

4.4 Configure vHSM to unseal with Utimaco u.trust GP HSM

To configure the vHSM, you need to set up vHSM first. Outside of development mode, vHSM servers are configured using a persistent storage method. For more information about the configuration parameters, see vHSM configuration.

Create a '`/etc/vhsm/config`' file and add the following contents:

```
# Configure Seal with Utimaco u.trust GP HSM
seal "pkcs11" {
  lib = "/opt/utimaco/lib/libcs_pkcs11_R3.so"
  slot = "<slot_no.>"
  pin = "<slot_PIN>"
  key_label = "hsm_demo"
  hmac_key_label = "demo"
  generate_key = "true"
  r3_config = "/opt/utimaco/cs_pkcs11_r3.cfg"
}
ui = true
disable_mlock = true
# Configure the storage backend
storage "file" {
```

```
  path = "/etc/vhsm/data"
}
listener "tcp" {
  address = "0.0.0.0:8200"
  tls_disable = true
}
```

Update <slot_no.> and <slot_PIN> according to your setup in section Create SO User and Initialize a Slot.

2. Start the enclave vHSM server to run with Utimaco u.trust GP HSM and mount the cs_pkcs11_r3.cfg file to the Docker container.

›_ Console

```
# docker run --cap-add=IPC_LOCK -p 8200:8200 \   -e
ENCLAIVE_LICENCE=$ENCLAIVE_LICENCE \   -v /opt/utimaco/cs_pkcs11_r3.cfg:/opt/
utimaco/cs_pkcs11_r3.cfg \   -v /path/to/your/local/config:/etc/vhsm/config \
harbor.enclaive.cloud/vhsm/vhsm-utimaco:latest \

  server -config /etc/vhsm/config
```

3. Set the environment variable.

›_ Console

```
# export VAULT_ADDR='0.0.0.0:8200'
```

4. Check the status of vHSM server.

›_ Console

```
# vhsm status
```

```
root@enclaive ~ % vhsm status
Key                     Value
---                     -----
Recovery Seal Type      pkcs11
Initialized             false
Sealed                  true
Total Recovery Shares   0
Threshold               0
Unseal Progress         0/0
Unseal Nonce            n/a
Version                 1.4.3-1
Build Date              2025-04-17T20:06:17Z
Storage Type            file
HA Enabled              false
```

5. Initialize vHSM to use the HSM initial token value.

›_ Console

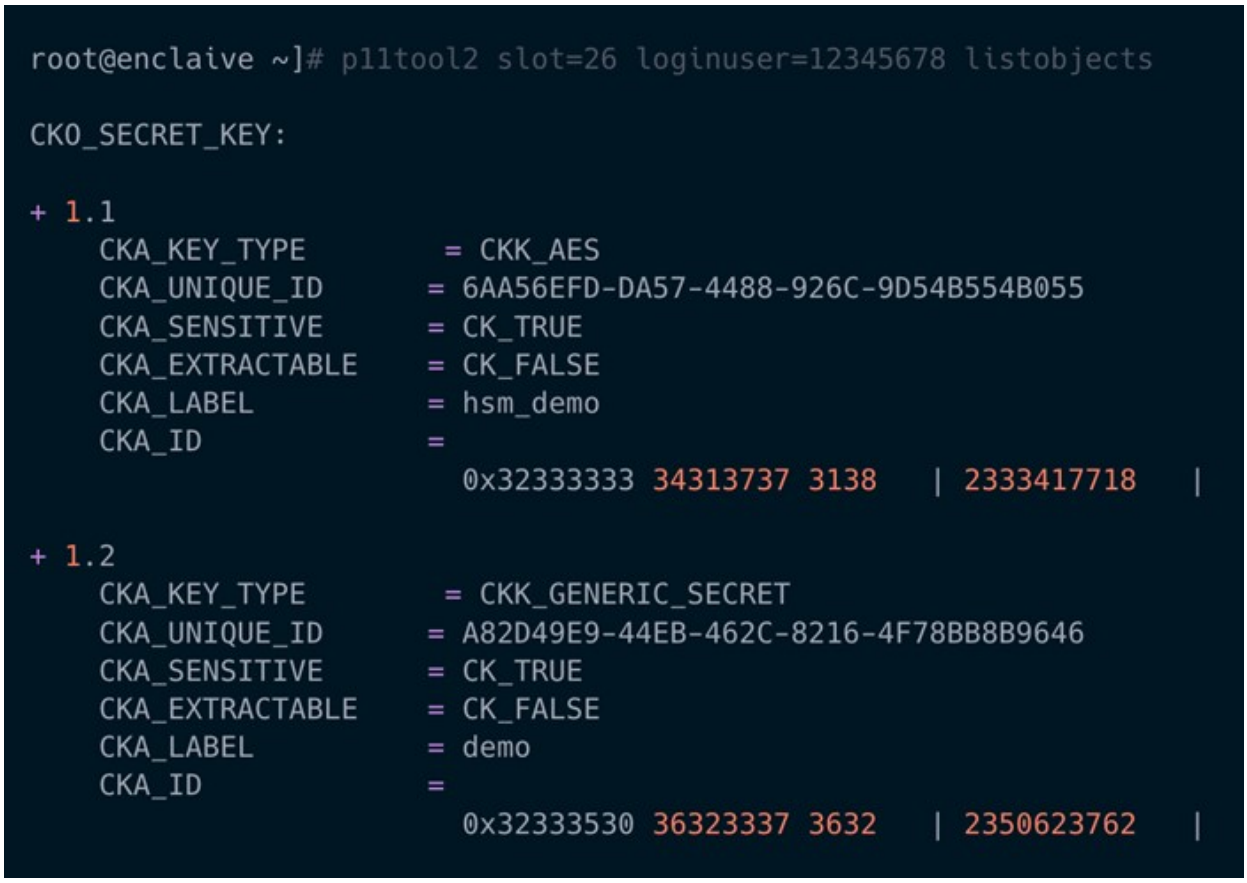
```
# vhsm operator init
```

```
root@enclaive ~ % vhsm operator init
Unseal Key 1: umQpOvos8g+wD5m0vVX6Jic6PEnajHy7xppqhXMiQtL0
Unseal Key 2: dYJQXLHzMKv4qGP3FN0gFhRynEE4LczZdtjOSCPdoFBO
Unseal Key 3: KLkrVV/RivfayjRooRd7Qsf25ddb8wtnMm+UDmmLvCfR
Unseal Key 4: 2CqPu4vuZ2S/b8ryfxiHtq1ZkMYKXGDb34yYIyHPvHDI
Unseal Key 5: KRMFesZpScG6iEbH/drt5h/xkdsImhlrhuvq3CIAbENo

Initial Root Token: hvs.XOazg3cmWJkfbMu4F9UU0HWq

vHSM initialized with 5 key shares and a key threshold of 3. Please securely
distribute the key shares printed above. When the vHSM is re-sealed,
restarted, or stopped, you must supply at least 3 of these keys to unseal it
before it can start servicing requests.

vHSM does not store the generated root key. Without at least 3 keys to
reconstruct the root key, vHSM will remain permanently sealed!

It is possible to generate new unseal keys, provided you have a quorum of
existing unseal keys shares. See "vhsm operator rekey" for more information.
```

6. Verify that the keys got generated into the HSM.

›_ Console

```
# p11tool2 slot=<slot_no.> loginuser=<slot_PIN> listobjects
```



7. Check the status of vHSM again to verify that it is initialized and unsealed.

›_ Console

```
# vhsm status
```

```
rooti@encllaive ~ % vhsm status
Key              Value
---              -----
Seal Type        shamir
Initialized      true
Sealed           false
Total Shares     1
Threshold        1
Version          1.4.3-1
Build Date       2025-04-17T15:41:32Z
Storage Type     inmem
Cluster Name     vault-cluster-ae7dd09c
Cluster ID       ba25adb3-59e2-1ba2-d933-3083c77f3ec6
HA Enabled       false
```

4.5 Login to vHSM using the CLI

Login to vHSM using the initial Root token that you saved.

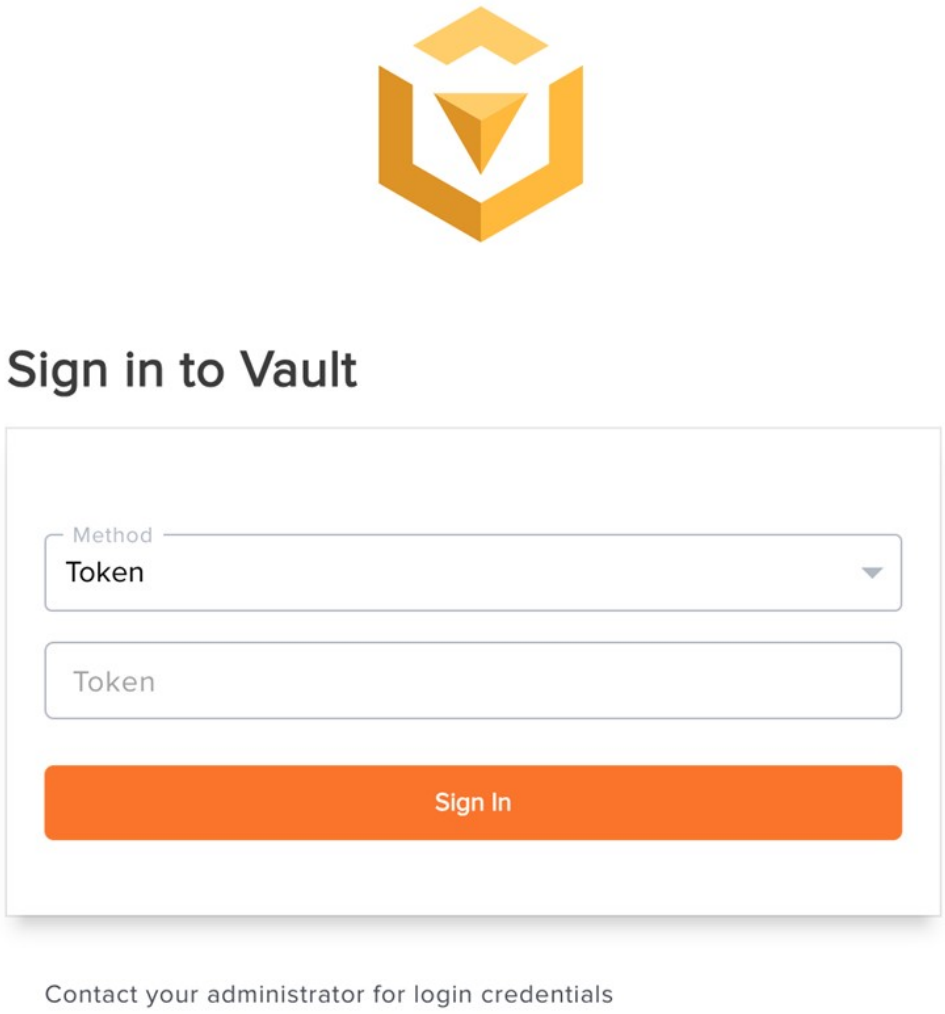
›_ Console

```
# vhsm login <initial_root_token_value>
```

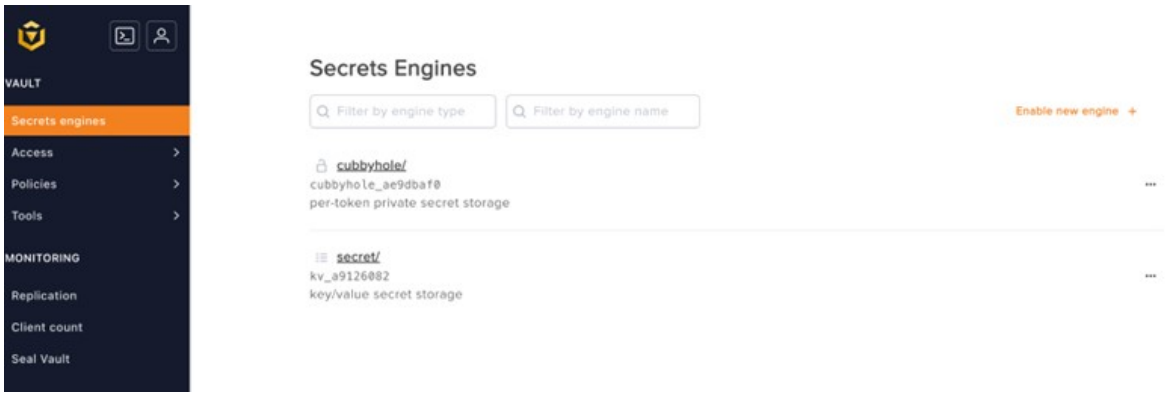
```
rooti@enclaive ~ % vhsm login
Token (will be hidden):
Success! You are now authenticated. The token information displayed below is
already stored in the token helper. You do NOT need to run "vhsm login" again.
Future vHSM requests will automatically use this token.

Key                  Value
---                  -----
token                hvs.hidxumzIlZ6yk044XQCO7bxS
token_accessor       E5leo0H9hmyinJ0n8eqfTtH5
token_duration       ∞
token_renewable      false
token_policies       ["root"]
identity_policies    []
policies             ["root"]

Cluster ID      1abf9eb1-3ae1-1652-7058-e9d73b08d105
HA Enabled      false
```

4.6 Login to vHSM using the Web UI

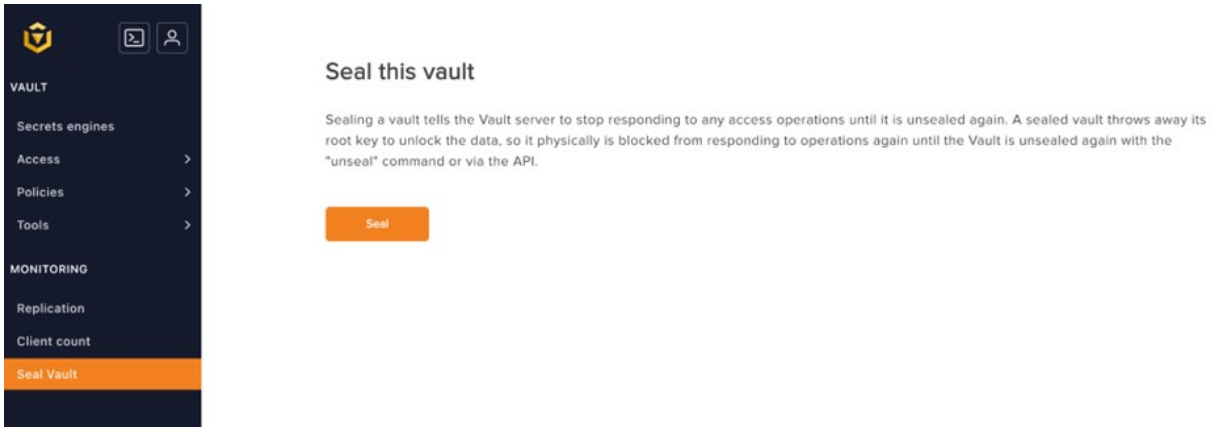
1. Open a browser and go to the URL http://<vhsm_server_ip_address>:82000/ui/vault/auth . The sign-in page loads.
2. Login to vHSM using the Initial Root Token that you saved.



3. After you login the Secrets Engines page is loaded.



4. To seal the vhsm vault go to Seal Vault and click Seal.



5 Test vHSM Integration with Utimaco u.trust GP HSM

After configuring vHSM with Utimaco u.trust GP HSM you can test the integration by viewing the secrets, enabling the Secrets Engine, seal wrapping the secret data, and retrieving the secret data.

1. View the current secrets and the default locations.

›_ Console

```
# vhsm secrets list
```

```
root@enclaive ~ % vhsm secrets list
Path          Type         Accessor              Description
----          ----         --------              -----------
cubbyhole/    cubbyhole    cubbyhole_ae9dbaf0    per-token private secret storage
identity/     identity     identity_55ca36ed     identity store
secret/       kv           kv_a9126082           key/value secret storage
sys/          system       system_620d7ac5       system endpoints used for control, policy and
debugging
```

2. Enable the KV engine.

›_ Console

```
# vhsm secrets enable -version=1 kv
```

```
root@enclaive ~ % vhsm secrets enable -version=1 kv
Success! Enabled the kv secrets engine at: kv/
root@enclaive ~ %
```

3. View the details of the Secrets Engine that you enabled.

›_ Console

```
# vhsm secrets list -detailed
```

```
root@enclaive ~ % vhsm secrets list -detailed
Path          Plugin       Accessor              Default TTL    Max TTL    Force No Cache    Replication    Seal Wrap    External
Entropy Access    Options           Description                                                  UUID
Version    Running Version         Running SHA256    Deprecation Status
----          ------       --------              -----------    -------    --------------    -----------    ---------    -------------
----------    -------           -----------                                                 ----                                    ---
----    ---------------         --------------    ------------------
cubbyhole/    cubbyhole    cubbyhole_c24a9bae    n/a            n/a        false             local          false        false
            map[]             per-token private secret storage                          810b9c6d-c754-2402-5fcc-48c874218565    n/a
      v1.4.3+builtin.vault    n/a               n/a
identity/     identity     identity_ae0ed0e2     system         system     false             replicated     false        false
            map[]             identity store                                            e2cb9b32-f1b9-b8a1-ed2f-e8a8f75d1b56    n/a
      v1.4.3+builtin.vault    n/a               n/a
secret/       kv           kv_7e8217a6           system         system     false             replicated     false        false
            map[version:2]    key/value secret storage                                  cdea4163-df9a-6f10-4b06-8a94f320795c    n/a
      v0.15.0+builtin         n/a               supported
sys/          system       system_324f2e96       n/a            n/a        false             replicated     true         false
            map[]             system endpoints used for control, policy and debugging   3792c10b-7541-09b8-c3e7-9ac42ea73aee    n/a
      v1.4.3+builtin.vault    n/a               n/a
```

4. To test the seal wrap feature, add secret data to the key/value storage of vHSM.›_ Console

```
# vhsm kv put kv/opt/vhsm/secret key=test_secret
```

```
root@enclaive ~ % vhsm kv put kv/opt/vhsm/secret key=test_secret
Success! Data written to: kv/opt/vhsm/secret
root@enclaive ~ %
```

5. Retrieve the secret data from vHSM storage.

›_ Console

```
# vhsm kv get kv/opt/vhsm/secret
```

```
root@enclaive ~ % vhsm kv get kv/opt/vhsm/secret
=== Data ===
Key    Value
---    -----
key    test_secret
root@enclaive ~ %
```

6 Enable Entropy Augmentation

Entropy augmentation allows enclaive vHSM to supplement its system entropy with entropy from an external cryptographic hardware security module. This is designed for environments where compliance with cryptographic regulations such as NIST SP 800-90B is required, or where augmented entropy from external sources—such as hardware true random number generators (TRNGs)—is used to replace or enhance system entropy.

1. Update the ' /etc/vhsm/config ' file and add the following contents:

```
# Configure Seal with Utimaco u.trust GP HSM

seal "pkcs11" {

lib = "/opt/utimaco/lib/libcs_pkcs11_R3.so"

slot = "<slot_no.>"

pin = "<slot_PIN>"

key_label = "hsm_demo"

hmac_key_label = "demo"

generate_key = "true"

r3_config = "/opt/utimaco/cs_pkcs11_r3.cfg"

}

ui = true

disable_mlock = true

# Configure the storage backend

storage "file" {

path = "/etc/vhsm/data"

}

listener "tcp" {

address = "0.0.0.0:8200"

tls_disable = true

}
```

```
# vHSM configuration to use Utimaco u.trust GP HSM for entropy augmentation

entropy "seal" {
mode = "augmentation"
}
```

Update <slot_no.> and <slot_PIN> according to your setup in section 2025-0002 Create SO User and Initialize a Slot

2. Save the config file, then restart the vHSM service.

›_ Console

```
# docker restart <name-of-the-container>
```

3. Log in to vHSM using the initial Root token that you saved.

›_ Console

```
# vhsm login <initial_root_token_value>
```

4. Execute the following command to enable transit secrets engine with external entropy source using the '-external-entropy-access' flag.

›_ Console

vhsm secrets enable -external-entropy-access transit

```
[root@enclaive ~]# vhsm secrets enable -external-entropy-access transit
Success! Enabled the transit secrets engine at: transit/
```

5. List the enabled secrets engine with '-detailed' flag.

›_ Console

```
# vhsm secrets list detailed
```

```
[root@enclaive ~]# vhsm secrets list -detailed
Path          Plugin       Accessor              Default TTL  Max TTL  Force No Cache  Replication    Seal Wrap  External Entropy
Access  Options                      Description                                Version               Running Version       Running
SHA256          Deprecation Status  UUID
----          ------       --------              -----------  -------  --------------  -----------    ---------- --------------------
----  -------                        -----------                                -------               ---------------       -----------
----          ------------------  ----
cubbyhole/    cubbyhole    cubbyhole_13b27c27    n/a          n/a      false           local          false      false
    map[]                          per-token private secret storage       n/a                   n/a                   n/a
          n/a                  89c7f88b-bc6d-40cb-513d-786
ece92afd3     n/a          v1.4.3+builtin.vault n/a          n/a      false           replicated     false      false
    map[]                          system                                 n/a                   n/a                   n/a
          n/a                  24b614e2-7f00-ac1b-1f93-328
identity/     identity     identity_8f414177     n/a          n/a      false           replicated     false      false
    map[]                          identity store                         system                system                system
          false                2bca948c-f41b-0602-5893-3fb
4bd40faa5     n/a          v1.4.3+builtin.vault n/a          n/a      false           replicated     false      false
    map[]                          system                                 n/a                   n/a                   n/a
          n/a                  925f31a3-74da-9266-e91b-245
kv/           kv           kv_768a1b40           n/a          n/a      false           replicated     false      false
    map[version:1]                 n/a                                    n/a                   n/a                   n/a
          false                6818492d-20e5-97bf-1a35-0e2
d522d1479     n/a          v0.14.2+builtin       n/a          n/a      false           replicated     true       false
    map[]                          system endpoints used for control, policy and debugging   n/a                      n/a
     n/a                      supported           n/a
sys/          system       system_3b8d717be      n/a          n/a      false           replicated     false      false
    map[]                          system endpoints used for control, policy and debugging   system                   system
   system                   false               925f31a3-74da-9266-e91b-245
b94872ba6     n/a          v1.4.3+builtin.vault n/a          n/a      false           replicated     false      true
   map[]                           system                                 n/a                   n/a                   n/a
          supported            n/a
transit/      transit      transit_d9b2c2a9      n/a          n/a      false           replicated     false      true
     map[]                           n/a                                    system                system                system
           false                6818492d-20e5-97bf-1a35-0e2
ba6fd9e24     n/a          v1.4.3+builtin.vault n/a          n/a      false           replicated     false      false
    map[]                          system                                 n/a                   n/a                   n/a
          supported            n/a
```

Note that the External Entropy Access is set to true for transit.

6. You can start using the transit secrets engine to encrypt your sensitive data which leverages the HSM as its external entropy source. Now create a new encryption key named "orders".

›_ Console

```
# vhsm write -f transit/keys/orders
```

```
[root@enclaive ~]# vhsm write -f transit/keys/orders
Success! Data written to: transit/keys/orders
```

7. Send a base64-encoded string to be encrypted by vHSM.

›_ Console

```
# vhsm write transit/encrypt/orders plaintext=$(base64 <<< “4111 1111 1111 1111”)
```

```
root@enclaive ~ % vhsm write transit/encrypt/orders plaintext=$(base64 <<< "4111 1111 1111 1111")
Key             Value
---             -----
ciphertext      vault:v1:mBBYBUoICZ/igXKgkb9YPmWA+2b6upmZM1WqQEyiiiyGa6aq6bpqn0Hfqxpi89aJ
key_version     1
```

8. Verify that you can decrypt.

›_ Console

```
# vhsm write transit/decrypt/orders ciphertext=vault:v1:mBBYBUoICZ/
igXKgkb9YPmWA+2b6upmZM1WqQEyiiiyGa6aq6bpqn0Hfqxpi89aJ
```

```
root@enclaive ~ % vhsm write transit/decrypt/orders
ciphertext=vault:v1:mBBYBUoICZ/igXKgkb9YPmWA+2b6upmZM1WqQEyiiiyGa6aq6bpqn0Hfqxpi89aJ
Key         Value
---         -----
plaintext   NDE xMSAxMTE xIDE xMTEgMTE xMQo=
    1
```

9. Decode to get the original value.

›_ Console

```
# base64 --decode <<< NDExMSAxMTExIDExMTEgMTExMQo=
```

```
root@enclaive ~ % base64 --decode <<< NDEzMSAxMTEzIDEzMTEgMTEzMQo=
4111 1111 1111 1111
```

7 Troubleshooting

Error	***Diagnosis***
Error: Failed to attach external HSM client library. Please check if you specified the vendor provided PKCS#11 library path correctly.	PKCS#11 library cannot be loaded. 1. Verify that you have specified the correct Path to PKCS#11 library path in 'config.json' file . 2. Verify that the cs_pkcs11_R3.cfg file is available in the '/etc/utimaco' folder. 3. Verify that the configurations in the cs_pkcs11_R3.cfg file are correct.
C_Login [type=1] returned Error 0x00000102 (CKR_USER_PIN_NOT_INITIALIZED)	PKCS#11 Slot is not initialized. You need to initialize the Slot. See steps 3 and 4 in section **Create SO User and Initialize a Slot**.
Error loading configuration from /etc/vhsm/config.json: stat /etc/vhsm/config.json: no such file or directory	json configuration cannot be loaded. 1. Ensure that the 'config.json' file is available in the docker container. 2. Check that the docker user has the permissions to run the container.
C_Login [type=131] returned Error 0x00000030 (CKR_DEVICE_ERROR)	PKCS#11 No connection to the HSM. 1. Check network and token availability. 2. Provide the correct Admin Key for the device to connect.

Error	***Diagnosis***
[WARN] failed to unseal core: error="stored unseal keys are supported, but none were found"	Ensure that the IP address of the vHSM in Docker is configured correctly.

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 u.trust GP HSM product documentation is also available at the Utimaco IS GmbH support website https://support.hsm.utimaco.com/.

9 References

Reference	***Title/Company***	***Document No.***
[p11tool2]	u.trust Anchor - PKCS#11 p11tool2 Reference Manual, Utimaco IS GmbH	2021-0072

10 Contact

You can reach us from Monday to Friday, 09.00 a.m. to 05.00 p.m., Central European Time (CET).

Utimaco IS GmbH

Germanusstr. 4

52080 Aachen

Germany

RMA Query

If you need to send the device back to Utimaco IS GmbH, please open a new RMA case (Return Merchandise Authorization). We request that you use the following web address. RMA cases cannot be opened by email or phone.

https://support.hsm.utimaco.com/support/rma/new

Other Support Queries

Mail (preferred contact method)

support@utimaco.com

Attach the diagnostic information to your email.

Web portal

https://support.hsm.utimaco.com/support/cases/new/

The diagnostic information will be requested in our response if necessary.

By phone

AMERICAS +1-844-UTIMACO (+1 844-884-6226)

EMEA +49 800-627-3081

APAC +81 800-919-1301

The diagnostic information will be requested in our response if necessary.