

u.trust Anchor FIPS 140-3

Administration Manual



Imprint

Copyright 2024	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 e-mail	https://support.hsm.utimaco.com/ support@utimaco.com
Document Version	1.0.5
Product Version	6.0.0
Date	2024-11-25
Document No.	2023-0027
Status	PUBLISHED

All rights reserved	<p>No part of this documentation may be reproduced in any form (printing, photocopy or according to any other process) without the written approval of Utimaco IS GmbH or be processed, reproduced or distributed using electronic systems.</p> <p>Utimaco IS GmbH reserves the right to modify or amend the documentation at any time without prior notice. Utimaco IS GmbH assumes no liability for typographical errors and damages incurred due to them. Any mention of the company name Utimaco in this documents refers to the Utimaco IS GmbH.</p> <p>All trademarks and registered trademarks are the property of their respective owners.</p>
---------------------	--

Table of Contents

1	Introduction	8
1.1	About this Document.....	8
1.1.1	Target Audience	8
1.1.2	Document Conventions	8
1.1.3	Abbreviations	9
2	Overview	11
2.1	u.trust Anchor Product Line	12
2.2	Architecture.....	14
2.3	The Chain of Trust Model	16
2.4	Keystore (Internal vs External)	18
2.5	Software	19
2.5.1	Boot Process	20
2.5.2	Images.....	21
2.5.3	Firmware Packages	21
2.6	Hardware	22
2.6.1	Hardware Version	23
2.6.2	Shared Resources	24
2.7	Tools and Utilities.....	24
2.7.1	gladm.....	24
2.7.2	cHSM Toolset	25
2.7.2.1	csadm	25
2.7.2.2	cxitool.....	27
2.7.2.3	CAT	28
2.7.2.4	p11tool2	29
2.8	Roles.....	29
2.8.1	Global Initial Administrator ADMIN	31
3	Concepts.....	32
3.1	Device Life Cycle, States and Modes	32
3.2	Quorum Requirements	35
3.3	System Keys.....	37
3.4	Security Mechanisms	40
3.4.1	Secure Messaging	40
3.4.2	Alarm and other Zeroization Events	41

3.4.2.1	Zeroization Event Overview	41
3.4.2.2	Alarm Handling and Environmental Failure Procedure.....	42
3.5	cHSM Management	45
3.5.1	cHSM Operating Modes	45
3.5.2	cHSM States.....	45
3.5.3	cHSM Templates	47
3.5.4	Taking Snapshots.....	48
4	Setup.....	49
4.1	Installing the Host Software.....	49
4.2	Installing the Host Software Without User Interaction.....	51
4.3	Installing gladm	54
4.4	Claiming the Device.....	56
4.4.1	Global Admin Management Tool (gladm)	56
4.4.1.1	Keys for User Authentication.....	59
4.4.2	Specifying a Device	60
4.4.3	Verifying the Authenticity of the Device	62
4.4.4	Changing the Authentication Token of the Global Initial Administrator	65
4.4.5	Importing an Operator Secret	66
4.4.6	Importing an Operator Certificate.....	68
5	Configuration	71
5.1	u.trust Anchor FIPS	71
5.1.1	Configuring the FIPS Approved Mode	71
5.1.2	cHSM Built-in Elliptic Curves FIPS 140-3	74
5.2	Creating a new cHSM	79
5.3	Generating an Operator Secret.....	81
5.4	Copying an Operator Secret.....	82
5.5	Changing Quorum Requirements for Dual Control	84
5.6	cHSM Claiming	85
5.6.1	Creating a Customer CA	85
5.6.2	Claiming the cHSM.....	86
5.6.2.1	Signature Verification.....	87
6	Monitoring and Maintenance	88
6.1	Updating u.trust Anchor	88

6.1.1	From Version 4.47.2 FIPS to 6.0.0-c FIPS	90
6.1.2	From Version 4.80 and 4.90 to 6.0.0	92
6.1.3	From Version 4.80, 4.90 and 6.0.0 to 6.0.0-c FIPS	92
6.2	Downgrading u.trust Anchor.....	93
6.2.1	From Version 4.60 or Higher	93
6.2.2	From Version 4.80 or Higher	94
6.2.3	From Version 6.0.0-c FIPS to 6.0.0	95
6.3	Erase and Clearing Procedures.....	96
6.3.1	Clearing Procedures	96
6.3.1.1	Clearing the Device.....	96
6.3.1.2	Clearing to FACTORY DEFAULT State	97
6.3.2	External Erase.....	98
6.3.2.1	Short External Erase	100
6.3.2.2	Long External Erase.....	101
6.4	Audit Log	103
6.4.1	Audit Log Events	104
6.4.2	Audit Log Hash Chain	106
6.5	Temperature-dependent Behavior of u.trust Anchor	106
6.6	Restarting the Device	107
6.7	License Files.....	108
6.7.1	Updating License Files	109
7	Troubleshooting	111
7.1	How to Identify different Device States.....	111
7.2	Leaving the Recovery Mode.....	113
7.3	Leaving the Alarm State	114
7.4	Leaving the Dead State.....	116
7.5	The License File is Corrupted, Invalid or Missing	116
8	Contact Address for Support Queries	118
9	References	119
10	Appendix.....	120
10.1	Global Administration Management (gladm) Tool.....	120
10.1.1	Introduction to gladm	120
10.1.2	Overview of gladm Commands	121
10.1.3	Managing Certificates and Secrets.....	123
10.1.3.1	Generating an Operator Secret (key-set-operator-secret)	123
10.1.3.2	Listing Operator Secrets (key-list-operator-secrets)	127

10.1.3.3	Importing an Operator Secret (key-import-operator-secret)	127
10.1.3.4	Deleting Operator Secrets (key-delete-operator-secret).....	129
10.1.3.5	Copying an Operator Secret (smartcard-copy-secret)	129
10.1.3.6	Importing a CA Certificate (key-import-cert)	131
10.1.3.7	Retrieving a Wrapping Key (key-get-wrapping-key)	131
10.1.4	Managing Users.....	132
10.1.4.1	Listing Users (user-list)	132
10.1.4.2	Adding a User (user-add).....	133
10.1.4.3	Changing Credentials (user-change-credentials)	134
10.1.4.4	Restoring Users (user-restore-backup)	135
10.1.4.5	Deleting a User (user-delete)	136
10.1.4.6	Getting a User Permissions Template (user-permissions)	136
10.1.4.7	Backing Up Users (user-create-backup)	137
10.1.5	Managing Slots.....	137
10.1.5.1	Listing Slots (chsm-list-slots)	138
10.1.5.2	Freeing a Slot (chsm-free-slot).....	139
10.1.5.3	Setting the Slot Quota (slot-set-quota)	140
10.1.5.4	Getting the Slot Quota (slot-get-quota).....	141
10.1.6	Managing CHSMs	142
10.1.6.1	Creating a new CHSM (chsm-create)	142
10.1.6.2	Cloning a CHSM (chsm-clone)	144
10.1.6.3	Taking a Snapshot (chsm-snapshot).....	145
10.1.6.4	Halting a CHSM (chsm-halt)	146
10.1.6.5	Retrieving a CHSM (chsm-retrieve)	147
10.1.6.6	Restoring a CHSM (chsm-restore)	147
10.1.7	System Commands.....	148
10.1.7.1	Listing Templates (system-list-templates).....	148
10.1.7.2	Retrieving the Chain of Trust (system-get-trust-chain).....	149
10.1.7.3	Retrieving the Audit Log (system-get-audit-log)	150
10.1.7.4	Displaying Device System Information (system-get-info)	151
10.1.7.5	Displaying License Information (system-get-license-info).....	152
10.1.7.6	Reading the System Log (system-fetch-log)	153
10.1.7.7	Getting Device Metrics (system-get-metrics).....	154
10.1.7.8	Setting the Quorum (system-set-quorum).....	158

10.1.7.9	Getting the Quorum (system-get-quorum-requirements).....	159
10.1.7.10	Setting the System Quota (system-set-quota)	159
10.1.7.11	Getting the System Quota (system-get-quota).....	160
10.1.7.12	Setting the Time (system-set-time).....	160
10.1.7.13	Getting the Time (system-get-time)	161
10.1.7.14	Getting the NTP Configuration (system-get-ntp-config)	162
10.1.7.15	Setting the NTP Configuration (system-set-ntp-config).....	163
10.1.7.16	Activating NTP (system-activate-ntp)	166
10.1.7.17	Resetting the Alarm (system-reset-alarm)	168
10.1.7.18	Restarting the Device (device-restart).....	169
10.1.7.19	Restarting the Device (system-restart)	170
10.1.7.20	Clearing the System (system-clear)	170
10.1.7.21	Updating the Device Firmware (system-update)	172
10.1.7.22	Emitting the Bash Completion Script (bash-completion)	173

1 Introduction

Thank you for purchasing our u.trust Anchor security system. We hope you are satisfied with our product. Please do not hesitate to contact us if you have any questions or comments.

1.1 About this Document

This manual provides information about the fundamentals of Utimaco's hardware security module u.trust Anchor and its security mechanisms. It contains guidelines on the administration of the u.trust Anchor with Utimaco's Global Admin Management Tool *gladm*, as well as guidelines for administrating the device according to [FIPS PUB 140-3 Security Requirements for Cryptographic Modules \(p. 119\)](#).

1.1.1 Target Audience

This document is primarily intended for all persons assuming the **Global Administrator** role for u.trust Anchor.

The main tasks of the **Global Administrator** are the initial personalization of the u.trust Anchor device, user management, global configuration, setup, and maintenance of cHSMs.

1.1.2 Document Conventions

We use the following document conventions:

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

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.

1.1.3 Abbreviations

We use the following abbreviations:

Abbreviation	Description
AES	Advanced Encryption Standard
BSI	Bundesamt für Sicherheit in der Informationstechnik (Federal Office for Information Security)
CA	Certificate Authority
CAK	Container Authentication Key
chSM	Containerized Hardware Security Module
CNG	Cryptography API: Next Generation
CSP	Cryptographic Service Provider
CSR	Certificate Signing Request
CXI	Cryptographic eXtended Interface
csadm	CryptoServer command-line administration tool
DAK	Device Authentication Key
DES	Data Encryption Standard
DKMS	Dynamic Kernel Module Support
DMK	Device Master Key
DRBG	Deterministic random bit generator

Abbreviation	Description
DRNG	Deterministic random number generator
DSA	Digital Signature Algorithm
ECDSA	Elliptic Curve DSA
EKM	Extensible Key Management
FIPS	Federal Information Processing Standard
GAK	Global Admin Key
GAAK	Global Admin Authentication Key
GIAK	Global Initial Admin Key
gladm	Global Admin Management Tool
JCE	Java Cryptography Extension
JRE	Java Runtime Environment
MAC	Message authentication code
MBK	Master Backup Key
NTP	Network Time Protocol
P11CAT	PKCS#11 CryptoServer Administration Tool
PCIe	PCI Express Interface
PEM	Privacy-Enhanced Mail
PMU	Platform Management Unit
PRNG	Pseudorandom number generator
RSA	Rivest, Shamir, Adleman (cryptosystem)
SDMK	Sticky Device Master Key
TRNG	True random number generator

Table 2: Abbreviations

2 Overview

The u.trust Anchor HSM is a multi-tenant Hardware Security Module (HSM) platform for payment and general-purpose use cases that enables cloud service providers and enterprises to offer HSM-as-a-Service (HSMaaS).

The u.trust Anchor HSM offers up to 31 cHSMs (containerized Hardware Security Module) and multiple PKCS #11 partitions per cHSM for application separation and key partitioning. Each cHSM instance is isolated; the administrative access and cryptographic functions are limited to the corresponding cHSM user, ensuring the required level of confidentiality for their sensitive data and keys.

A clear separation is maintained between the [Global Administrator \(p. 29\)](#) who orchestrates the physical HSM and the overall setup, and the [cHSM Administrators \(p. 29\)](#) who manage their corresponding virtual cHSMs. See [Architecture \(p. 14\)](#) for further details.

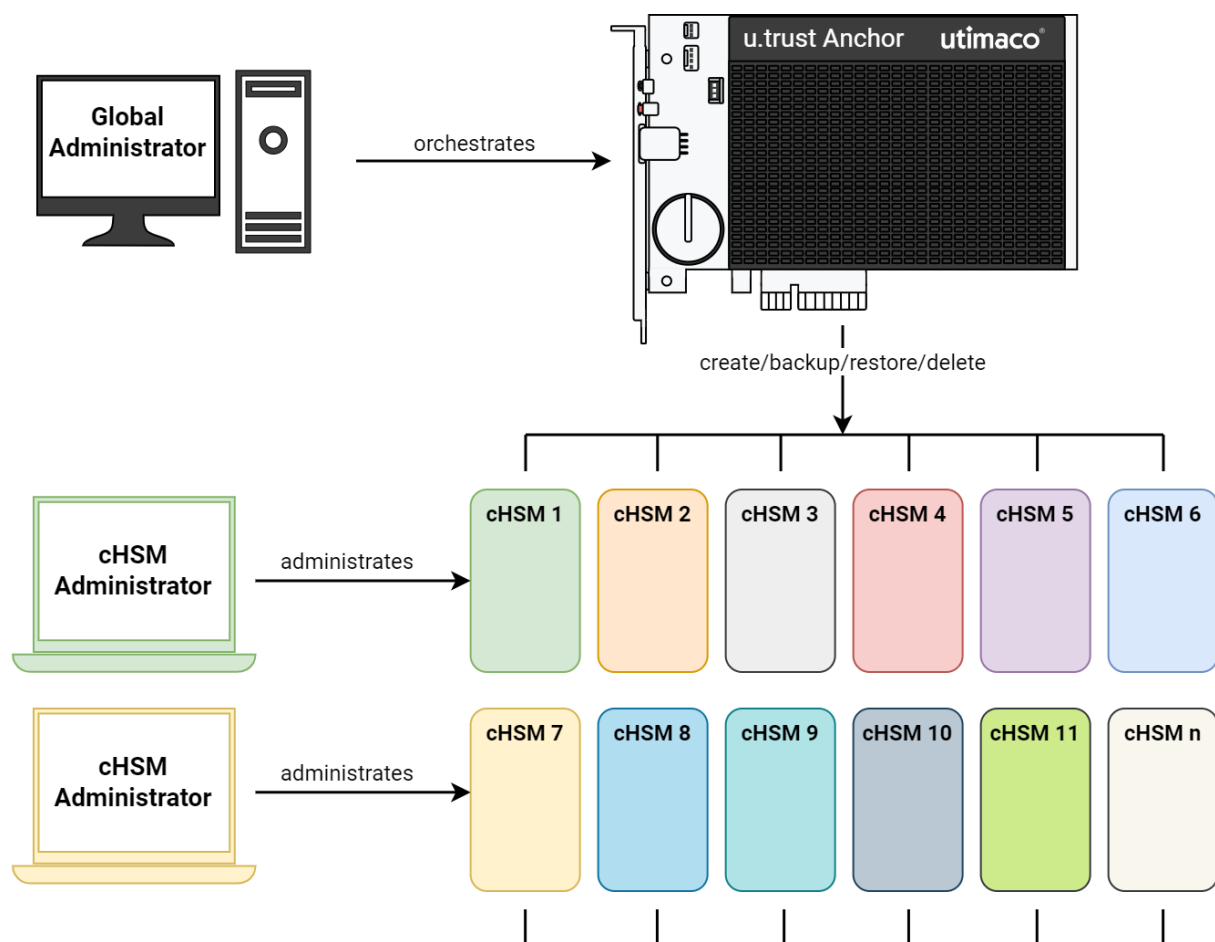


Figure 1 : u.trust Anchor and cHSMs

Various [templates \(p. 47\)](#) are provided for the creation of cHSMs; the cHSMs will have varying attributes and functionalities, depending on the template chosen.

The u.trust Anchor platform offers the following core features:

- Usage of up to 31 cHSMs (containerized Hardware Security Modules)
- Key management
- Key usage
- Certificate management
- Storage
- Encryption

These core functions are performed within a tamper-resistant, hardened environment, guaranteeing the integrity and confidentiality of sensitive data. The critical hardware components of a u.trust Anchor are located on a printed circuit board, completely covered by potting material. This hard, opaque enclosure protects the sensitive u.trust Anchor hardware components from physical attacks. If the sensory controller (powered by the on-board battery) detects a critical tamper event (see Alarm Triggers), all sensitive information is deleted immediately. See also [Alarm Mechanism \(p. 42\)](#) for more details.

2.1 u.trust Anchor Product Line

u.trust Anchor is offered as a single PCIe card or mounted in a LAN appliance.

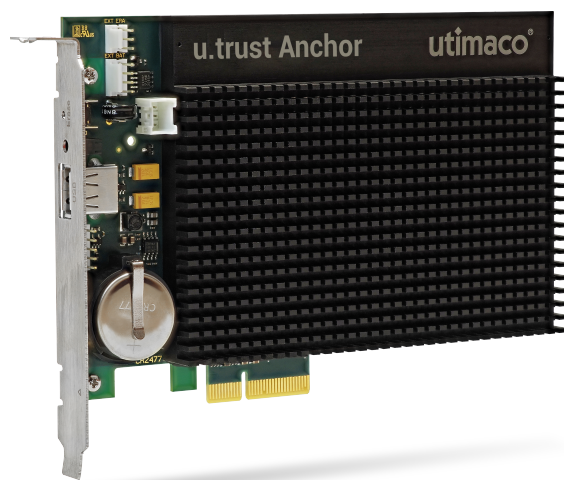


Figure 2 : u.trust Anchor PCIe card



Figure 3 : LAN Appliance Front View

Available models:

Product	Model Number	cHSMs	RSA 2k Performance	Cryptographic Accelerator	Certification	Certification Status
u.trust Anchor Se	Se100	1	100	✗	FIPS 140-3 Level 3	In progress
	Se2k	4	2000	✗	FIPS 140-3 Level 3	In progress
	Se5k	8	5000	✗	FIPS 140-3 Level 3	In progress
	Se15K	4	15000	✓	Common Criteria EAL4+ FIPS 140-2 Level 3 FIPS 140-3 Level 3	Certified (v.4.49.0) Certified (v.4.47.2) In progress
	Se40K	12	40000	✓	Common Criteria EAL4+ FIPS 140-2 Level 3 FIPS 140-3 Level 3	Certified (v.4.49.0) Certified (v.4.47.2) In progress

<i>Product</i>	<i>Model Number</i>	<i>cHSMs</i>	<i>RSA 2k Performance</i>	<i>Cryptographic Accelerator</i>	<i>Certification</i>	<i>Certification Status</i>
u.trust Anchor CSAR	Standard	8	40000	✓	Common Criteria EAL4+ FIPS 140-2 Level 3 FIPS 140-3 Level 3	Certified (v.4.49.0) Certified (v.4.47.2) In progress
	Plus	12	40000	✓	Common Criteria EAL4+ FIPS 140-2 Level 3 FIPS 140-3 Level 3	Certified (v.4.49.0) Certified (v.4.47.2) In progress
	Premium	31	40000	✓	Common Criteria EAL4+ FIPS 140-2 Level 3 FIPS 140-3 Level 3	Certified (v.4.49.0) Certified (v.4.47.2) In progress

Table 3: Hardware Models



For information about upgrading your u.trust Anchor model see [License Files \(p. 108\)](#) section.

2.2 Architecture

The u.trust Anchor has two administration categories:

1. The orchestration of the physical u.trust Anchor HSM, including basic management (creation, backup, restore, and deletion) of all cHSMs (virtual containerized Hardware Security Module)
 - The [Global Administrator \(p. 29\)](#) issues commands via [gladm \(p. 24\)](#) (Global Administration Management tool), which are communicated by the Global Administration (glad) service to [Gladracks \(p. 19\)](#) (container orchestration system) and are executed within the device.
 - The entity operating and owning the physical u.trust Anchor HSM is called the [OPERATOR \(p. 29\)](#).
2. The administration of a virtual cHSM (containerized Hardware Security Module)

- The **cHSM Administrator** (p. 29) issues commands via the **cHSM Toolset** (p. 25), which are executed by the cHSM firmware within the container.
- The programs of the **cHSM Toolset** (p. 25) can either communicate directly with a cHSM (**csadm** (p. 25), **CAT** (p. 28) and **cxitool** (p. 27)) or via a range of standard APIs (PKCS#11, CNG, OpenSSL JCE).
- The entity operating (and owning) the virtual cHSM is called **cHSM TENANT** (p. 29).



The (secret) data stored in each cHSM is isolated within the container and cannot be accessed by the **Global Administrator** (p. 29) or another cHSM.



The **OPERATOR** (p. 29) and the **cHSM TENANT** (p. 29) can belong to the same or different organizations.



The **VENDOR** (p. 29)(Utimaco) can only access the u.trust Anchor during the manufacturing process or in case of a Return Merchandise Authorization (RMA).

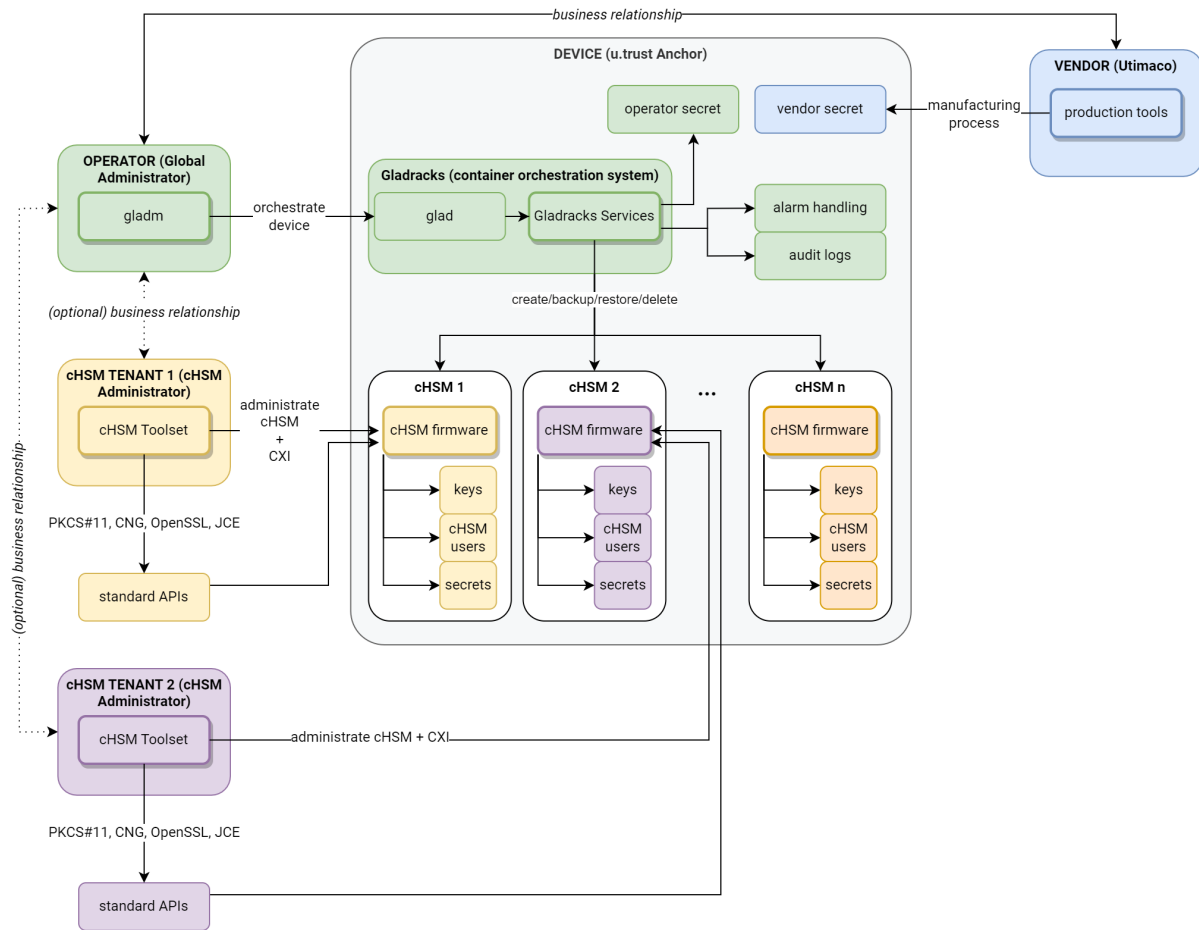


Figure 4 : u.trust Anchor - Architecture

2.3 The Chain of Trust Model

Each u.trust Anchor HSM is delivered with a unique asymmetric Elliptic Curve (EC) secret key generated in the HSM during the manufacturing stage. This key is named the Device Authentication Key (DAK). The DAK public key is signed during production by the Utimaco (Intermediate-) certificate authority (CA) certificate and the resulting DAK certificate is loaded into the HSM. The DAK certificate itself is a CA certificate and allows the device to sign a number of End-Entity certificates.

In the picture below you can see two End-Entity certificates signed by the DAK key: the GLAD Authentication Key (GAK) certificate and Container Authentication Key (CAK) certificate. For simplicity, the Intermediate CA between the Utimaco ROOT CA and the DAK CA is not displayed in the picture. Therefore, there is a verification path from the GAK and CAK certificates up to the Utimaco Root CA and the HSM is in possession of the corresponding GAK/CAK secret keys.

The certificate chain allows the OPERATOR/CUSTOMER to verify that the device has not been tampered with and was produced by Utimaco.



In case of a tamper attempt, the DAK/GAK/CAK secret keys are automatically erased.

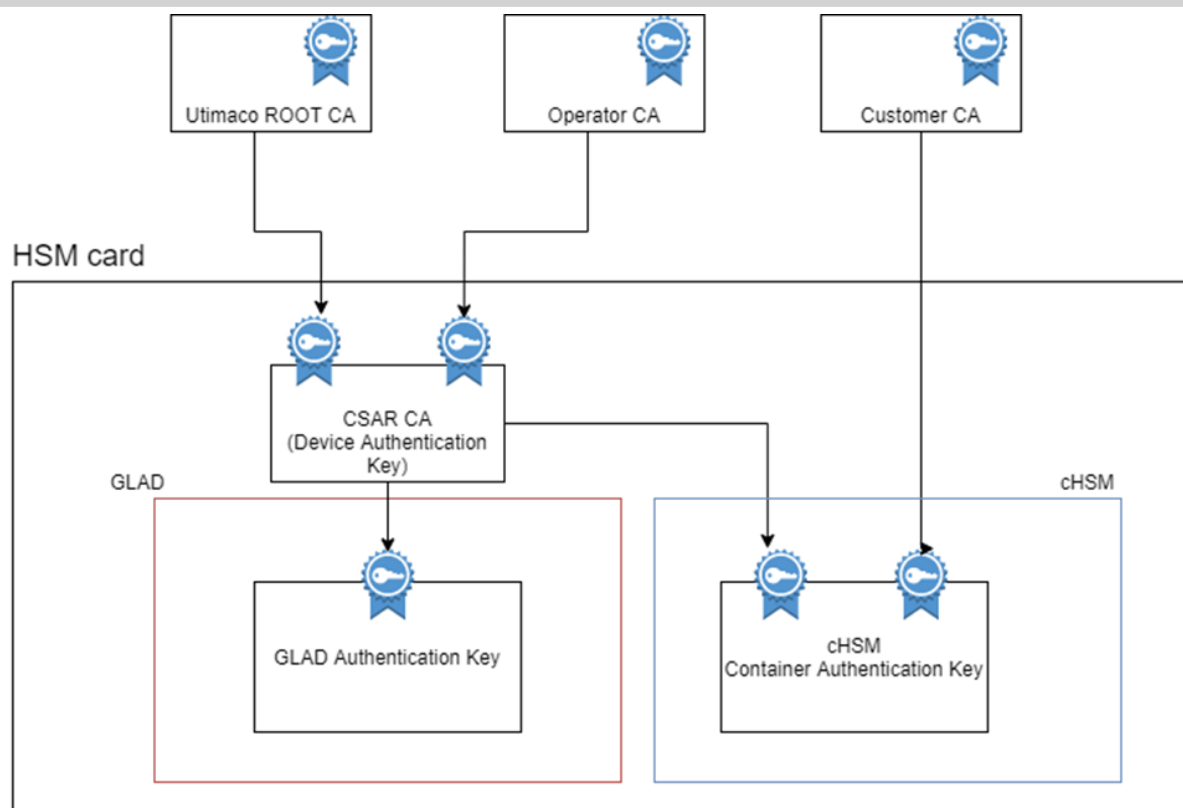


Figure 5 : Chain of Trust model

Upon receiving the device, one of the first steps after the basic installation is to authenticate the device against the Utimaco Root Certificate.

Depending on the system design, it can be useful to validate that the HSM is not just an Utimaco HSM, but that it is operated by a particular OPERATOR (e.g. the IT department of the company or any kind of service provider). In this case, the OPERATOR would establish another chain of trust by signing the DAK public key with their own OPERATOR CA. This process is called device claiming. For more information on device claiming, see the *Importing an Operator Certificate* chapter of the [u.trust Anchor FIPS 140-3- Administration Manual \(p. 119\)](#).

A similar process can be performed on the cHSM level as well. Namely, the cHSM Administrator can claim a cHSM instance by signing the CAK public key of this instance using their own CA. For more information on claiming a cHSM instance, see [t \(p. 16\)](#)he *Claiming a new*

cHSM chapter of the [u.trust Anchor FIPS 140-3 - Containerized Hardware Security Module \(cHSM\) - Administration Manual](#) (p. 119).

Though different Chains of Trust, the cHSM administrator and user can verify:

- that a trusted Utimaco HSM is running the cHSM,
- the HSM is operated by the OPERATOR (owner of Operator CA),
- the cHSM was claimed by the CUSTOMER.

The OPERATOR can verify:

- the HSM device is a trusted Utimaco HSM,
- the HSM was previously claimed by the OPERATOR.

2.4 Keystore (Internal vs External)

u.trust Anchor devices can use either external or internal keystore.

It is possible to adjust the use of the cHSMs based on customer requirements, such as by utilizing some cHSMs for one application and other cHSMs for another application. Each application may use their keystore independently, either internally or externally:

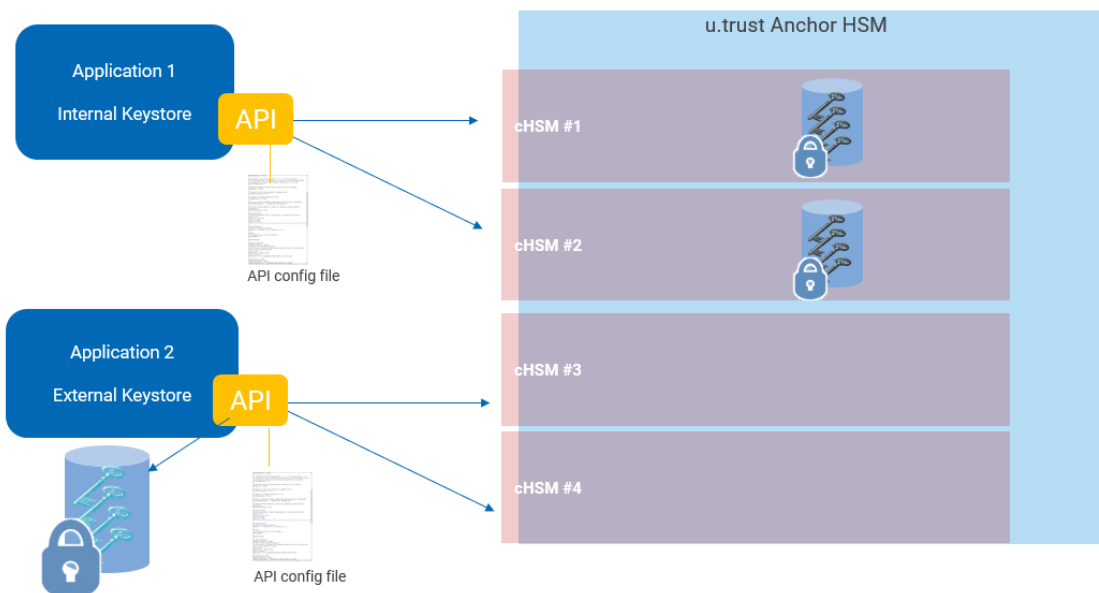


Figure 6 : Internal and External Keystore

Internal Keystore

When keys are generated or imported, they are replicated amongst all cHSMs within a cluster.

The concept of a cluster is defined at the API level within the API configuration file, wherein you can specify how many cHSMs will be part of a given cluster.

External Keystore

In this configuration, the keys are automatically visible to all applications that can access the keystore. The external keystore can be a flat file or an ODBC database, thus permitting access management from multiple applications concurrently.

Therefore, the customer is not limited to a maximum number of keys and key replication is not needed. Keys are encrypted and stored under the Master Backup Key (MBK), and backing up the key simply requires a backup of the flat file or the database content.

2.5 Software

Inside the u.trust Anchor, different kinds of software will run at different times:

- **bootloader**, a special firmware that loads software [images \(p. 21\)](#) from persistent storage into RAM, verifies their correctness, and executes them. The bootloader is the first software started inside the u.trust Anchor after a power-up, reboot, or reset.
- **COSMOS**, the underlying operating system, consisting of two modules:
 - A customized Linux kernel with a software stack supporting the individual cHSM environments. It contains libraries and binaries necessary to launch and support Gladracks, the container orchestration system.
 - Custom drivers, which provide access to the necessary hardware interfaces.
- **Gladracks**, the container orchestration system, which allows the administration of the physical u.trust Anchor HSM, as well the management of the the virtual cHSMs via [gladm \(p. 24\)](#).
- **cHSM**, a containerized Hardware Security Module. Depending on the [template \(p. 47\)](#) used to create the cHSM, different functionalities are available.



The bootloader is an independent firmware that runs only during start-up on the u.trust Anchor, whereas other modules run on the u.trust Anchor during its normal Operational Mode.

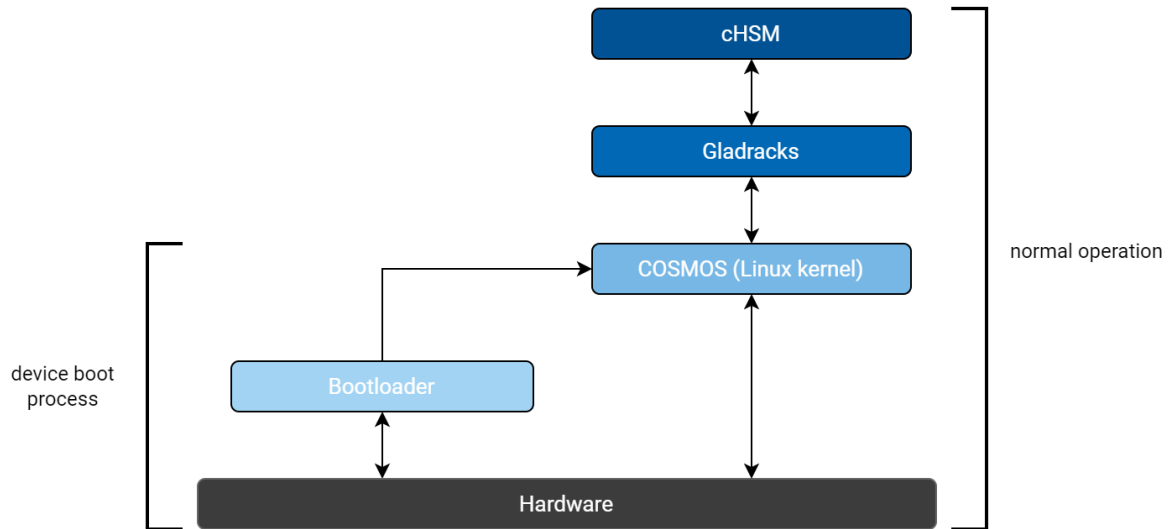


Figure 7 : u.trust Anchor - Subsystems

2.5.1 Boot Process

The bootloader is a special firmware that loads software images from persistent storage into RAM, verifies their correctness, and executes them. The bootloader is the first software started inside the u.trust Anchor after a power-up, reboot, or reset.

If during the boot process the bootloader finds itself initialized (i.e., the u.trust Anchor's public Production Key has been found) and the operating system module COSMOS is present in the u.trust Anchor, the latter will be started by the bootloader.



In case of an error during any of the boot stages, FSBL increments the multi-boot register, prints a message to the console, and resets the board (booting recovery).

A recovery boot is performed in the following cases:

- an alarm has been triggered,
- all operational images on the device are broken,
- after the External Erase button has been pressed for more than 10 seconds,

- or the reboot process has been interrupted several times in short order (a full reboot takes up to 50 seconds).



See also [Leaving the Recovery Mode \(p. 113\)](#).

2.5.2 Images

The u.trust Anchor contains two types of images:

- **The operational image**

The device is booted with the operational image, if the device runs in the **FACTORY DEFAULT** (p. 32) or **INITIALIZED** (p. 32) state.

- **The recovery image**

The device is booted with the recovery image, if a special event has been triggered.

This can be the case when

- an alarm has been triggered,
- all operational images on the device are broken,
- after the External Erase button has been pressed for more than 10 seconds,
- or the reboot process has been interrupted several times in short order (a full reboot takes up to 50 seconds).



See also [Leaving the Recovery Mode \(p. 113\)](#).

See also [How to Identify different Device States \(p. 111\)](#).

2.5.3 Firmware Packages

Different firmware packages are available for u.trust Anchor, depending on the purchased license:

- Standard

- FIPS
- GP CC



In-field license changes are currently not possible. A u.trust Anchor device needs to be returned to Utimaco in order to replace a license file. Please contact [Utimaco Support \(p. 118\)](#) for assistance.

The firmware packages contain different [cHSM templates \(p. 47\)](#) with varying restrictions and features.

See also [Updating u.trust Anchor \(p. 88\)](#) and [Downgrading u.trust Anchor \(p. 93\)](#).

2.6 Hardware

Utimaco's hardware security module (HSM) u.trust Anchor is a physically protected specialized computer unit designed to perform sensitive cryptographic tasks and to securely manage cryptographic keys.

The diagram below shows the hardware components of u.trust Anchor located on the printed circuit board and completely covered by potting material. This hard, opaque enclosure protects the sensitive u.trust Anchor hardware components from physical attacks.

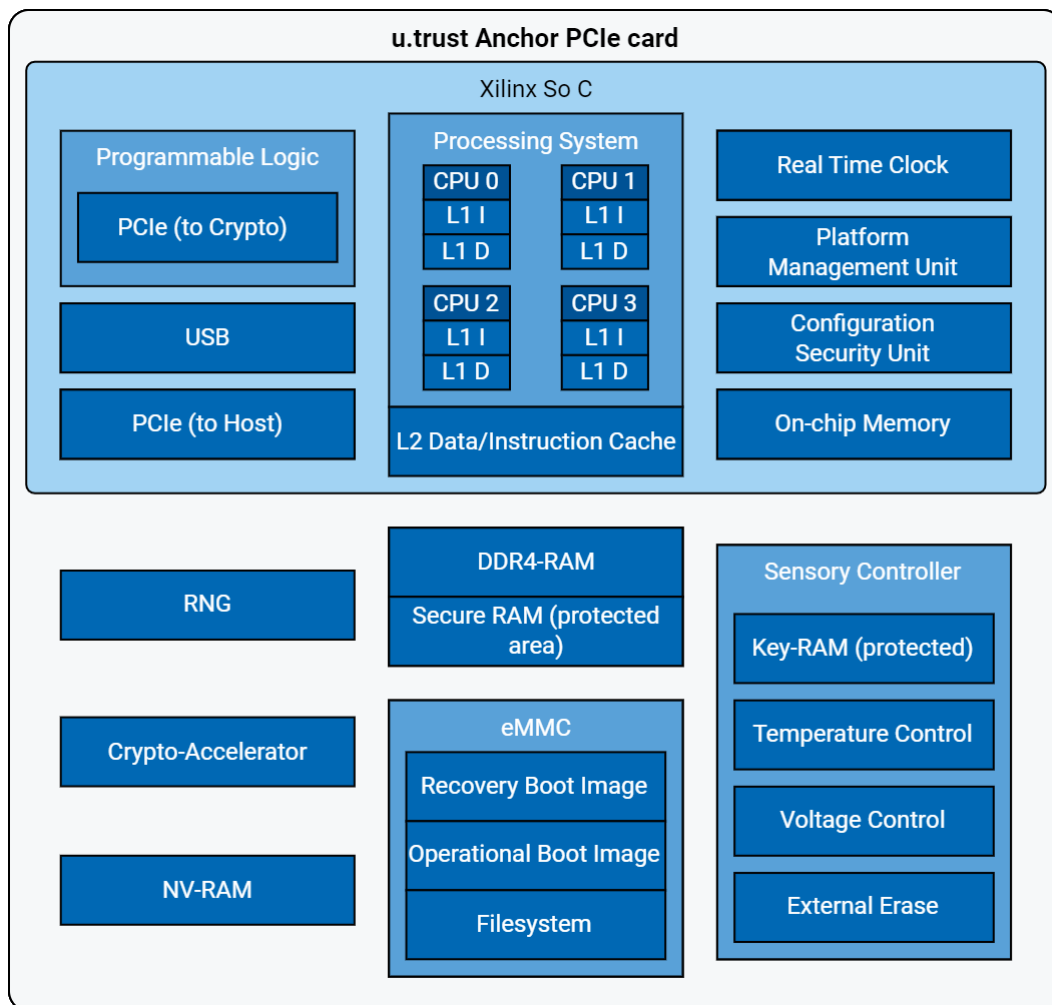


Figure 8 : u.trust Anchor - Hardware

2.6.1 Hardware Version

The hardware version is denoted in `a.bb.c.d` format:

a	Model number which is hardcoded in the bootloader a = 7 for u.trust Anchor series
b	Major hardware version number, hardwired on the PCB
c	Minor hardware version number, hardwired on the PCB
d	Revision number, coded as assembly configuration in PCB

Table 4: Hardware version format

The hardware version numbers `bb`, `c` and `d` are read from the PCB on start-up and combined with the model number `a` hardcoded in software to the hardware version number `a.bb.c.d`.

2.6.2 Shared Resources

The following resources are shared among all cHSMs present on a u.trust Anchor device:

- RAM
- Secure RAM
- NVRAM
- PCIe
- USB
- Cryptographic Accelerator
- Sensory Controller
- Random Number Generator
- UID

2.7 Tools and Utilities

2.7.1 gladm

The Global Administration Management (gladm) tool is a command-line administration tool. It is used by the [Global Administrator \(p. 29\)](#) and the [Global Initial Administrator \(p. 29\)](#) to configure the device and manage cHSMs.

gladm offers the following core functions:

- u.trust Anchor HSM management
 - Device setup
 - Card updates
 - Alarm handling

- Reset to FACTORY DEFAULT
- GLAD user management
- Operator secrets
- Audit logs
- cHSM management
 - Create cHSMs (on behalf of the [cHSM TENANT](#) (p. 29))
 - cHSM backups



gladm's cryptographic functions are limited to authentication operations only.

See also the appendix [Global Administration Management \(gladm\) Tool](#) (p. 120) for more details about gladm commands.

command> can be any of:	
chsm-clone	Clone cHSMs from a snapshot to create a cHSM cluster
chsm-create	Create a cHSM in the specified slot
chsm-free-slot	Free the specified cHSM slot(s)
chsm-halt	Halt the cHSM(s) in the specified slot(s)
chsm-list-slots	List basic information about accessible cHSM slots
chsm-restore	Restore a cHSM from a snapshot in the specified slot
chsm-retrieve	Halt the cHSM and retrieve it
chsm-snapshot	Take a snapshot of the cHSM in the specified slot
key-get-wrapping-key	Obtain a wrapping key for encrypted import of an Operator Secret
key-import-operator-secret	Import a new Operator Secret
key-delete-operator-secret	Delete a stored Operator Secret
key-list-operator-secrets	List all stored Operator Secrets
key-import-cert	Import an X.509 CA certificate for the DAK
slot-get-quota	Get quota values for slots
slot-set-quota	Set quota values for slots
system-get-time	Get the current time of the device
system-get-info	Display device system information
system-list-templates	List the templates available on the device
system-get-audit-log	Get the system audit log
system-fetch-log	Read the system log
system-get-metrics	Get device metrics
system-reset-alarm	Reset the alarm state of the device
system-set-time	Set the current time of the device
system-get-quota	Get quota values for system services
system-set-quota	Set quota values for system services
system-update	Update the device firmware
system-get-trust-chain	Retrieve the trust chain for device

Figure 9 : Screenshot - gladm

2.7.2 cHSM Toolset

2.7.2.1 csadm

csadm is a command-line administration tool. It is used by the [cHSM Administrator](#) (p. 29) to administer containerized Hardware Security Modules (cHSMs).

The tool offers the following core functions:

- Setup
- Status monitoring
- cHSM user management
- Key management



csadm's cryptographic functions are limited to authentication and signing operations only.

Additionally, it can perform advanced administration functions that are exclusively available for customers working with SDK devices that want to extend the standard functionality of the device with self-developed firmware modules providing specific cryptographic functions and commands.

See also *u.trust Anchor - csadm Manual*.

```

Basic Commands:
Help[=]
Sleep=
PrintError=
Cmd=

More
ConnTimeout=
StrError=
CmdFile=

Version
SetTimeout=
CSTerm=

Authentication:
LogonPass=
GetHSMAuthKey

LogonSign=
ShowAuthState

Administration:
GetState
MemInfo[=]
GetAuditConfig
GetAuditLogKey
ClearAuditLogFiles=
Reset[=]
ResetAlarm[=]
RecoverOS
SetAdminMode=
SignConfig=

GetInfo
GetAuditLog
SetAuditConfig=
GetSignedAuditLog=
GetTime
Restart[=]
ResetToBL[=]
Test=
SetStartupMode=

GetBattState
ClearAuditLog[=]
GenerateAuditLogKey=
VerifySignedAuditLog=
SetTime=
GetBootLog
StartOS
GetModel
GetStartupMode

User Key Management:
GenKey=
SaveKey=
GetCardInfo=

ChangePin=
BackupKey=
BackupDatabase=

ChangePassword=
CopyBackupCard=
RestoreDatabase=

User Management:
ListUser
DeleteUser=
SetMaxAuthFails=

AddUser=
BackupUser=
GetMaxAuthFails

Master Backup Key (MBK) Management:
MBKListKeys
MBKCopyKey=
MBKPinChange=

MBKGenerateKey=
MBKCardInfo=
MBKImportKey=
MBKCardCopy=

Firmware Management:
ListFirmware
DeleteFile=
LoadPkg=
Unpack=

ListFiles[=]
ListPkg=
Clear[=]
ModuleInfo=

LoadFile=
CheckPkg=
Pack=

```

Figure 10 : Screenshot - csadm

2.7.2.2 cxitool

The cxitool is a command-line utility tool. It is utilized by cHSM Users to perform cryptographic tasks within a containerized Hardware Security Modules (cHSM).

The utility tool offers the following core functions:

- Key management
 - Generate
 - Delete
 - Back up
 - Restore
- Key usage
 - Encrypt
 - Decrypt
 - Sign
 - Verify
- Certificate management
- External key storage handling

See also *u.trust Anchor - cxitool Manual*.

```
$ cxitool help
Basic Commands:
Help=          Version          Timeout=       LogonPass=
LogonSign=
CXI Firmware Configuration:
GetConfig      SetConfig=     ResetConfig   SecureGroupPass=
Key Management:
ListKeys       KeyInfo        GenerateKey=  DeleteKey
BackupKey      RestoreKey=    ExportPubKeyFile=
Certificate Management:
ExportP10=     ExportCert=    ImportCert=   ImportP12=
SelfSignedCert=
Cryptography:
Encrypt=       Decrypt=       Hash=         Sign=
Verify=
FIPS Specific Commands:
SetFipsUsage=
Use cxitool help=<command> to get further help
```

Figure 11 : Screenshot - cxitool

2.7.2.3 CAT

CAT is a GUI administration tool written in Java. It is used by the [cHSM Administrator \(p. 29\)](#) to administer containerized Hardware Security Modules (cHSMs).

The tool offers the following core functions:

- Setup
- Status monitoring
- cHSM user management
- Key management



CAT's cryptographic functions are limited to authentication and signing operations only.

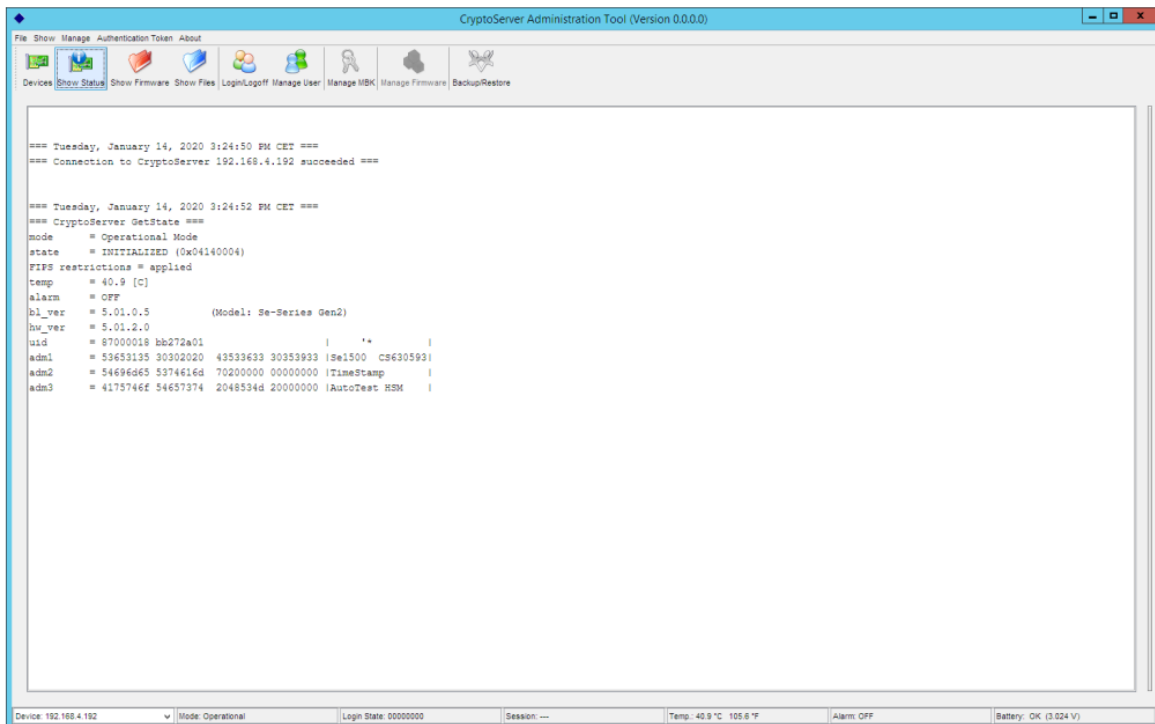


Figure 12 : Screenshot - CAT

2.7.2.4 p11tool2

The PKCS#11 Administration Tool Release 2 (p11tool2) is a command-line utility tool. It is based on the provided PKCS#11 Library R3 and utilized by the cHSM Users to execute PKCS#11 commands within a containerized Hardware Security Modules (cHSM).

p11tool2 offers the following core functions:

- Key management
- Certificate management
- Backup and restore

See also *u.trust Anchor - PKCS#11 p11tool2 - Reference Manual*.

```
$ p11tool2 help
CryptoServer PKCS#11 Administration Tool Release 2

Basic Commands:
  Help[=]          PrintError=      Version

PKCS#11 Commands:
  ListSlots[=]     GetInfo           GetSlotInfo
  GetTokenInfo     InitToken=        LoginSO=
  LoginUser=       Login=            InitPIN=
  SetPIN=          ListObjects       DeleteObject
  ImportP12=       ImportCert=       ExportCert[=]
  ExportP10=       GenerateKey=      GenerateKeyPair=

Backup/Restore Commands:
  GetBackupInfo=   BackupInternalKeys= BackupExternalKeys=
  BackupConfig=    RestoreInternalKeys= RestoreExternalKeys=
  RestoreConfig=   DeleteSO          RecryptExternalKeys=

Configuration Commands:
  ListConfig       GetLocalConfig=   GetGlobalConfig=
  SetGlobalConfig= GetSlotConfig=      SetSlotConfig=
  SecureSlotPass=

Use p11tool2 help=<command> to get further help.
```

Figure 13 : Screenshot - p11tool2



When implementing your own PKCS#11 applications, please also read the *CryptoServer - PKCS#11 R3 - Developer Guide* (document number: 2012-0007) provided within the product bundle at `\Documentation\Crypto_APIs\PKCS11_R3`.

2.8 Roles

Role	Responsible for	Access	Tools
Global Initial Administrator (p. 31)	<ul style="list-style-type: none"> ▪ Setting up (p. 49) and Claiming (p. 56) the device 	u.trust Anchor HSM running in FACTORY DEFAULT state (p. 32)	gladm (p. 24)

Global Administrator	<ul style="list-style-type: none"> ▪ u.trust Anchor HSM management <ul style="list-style-type: none"> • Card updates (p. 88) • Alarm handling (p. 41) • External erases (p. 98) • GLAD users (p. 132) • Operator secrets (p. 81) • Physical access to the device • Audit logs (p. 103) ▪ cHSM management (p. 45) <ul style="list-style-type: none"> • Create cHSMs (p. 142) (on behalf of the cHSM TENANT) • cHSM backups 	u.trust Anchor HSM	gladm (p. 24)
cHSM Administrator	<ul style="list-style-type: none"> ▪ cHSM administration ▪ cHSM users ▪ Database backups 	cHSM	csadm (p. 25)
cHSM User	<ul style="list-style-type: none"> ▪ Keys ▪ Cryptographic operations 	cHSM	cxitool (p. 27) p11tool2 (p. 29) (Standard APIs)
Entity	Responsible for	Access	Tools
VENDOR	<ul style="list-style-type: none"> ▪ Vendor secrets ▪ Manufacturing ▪ Support (p. 118) 	only during manufacturing or RMA	production tools

OPERATOR	<ul style="list-style-type: none"> ▪ u.trust Anchor HSM ownership ▪ u.trust Anchor HSM management <ul style="list-style-type: none"> • Card updates (p. 88) • Alarm handling (p. 41) • External erases (p. 98) • GLAD users (p. 132) • Operator secrets (p. 81) • Physical access to the device • Audit logs (p. 103) ▪ cHSM management (p. 45) <ul style="list-style-type: none"> • Create cHSM (p. 142) (on behalf of the cHSM TENANT) • cHSM backups 	u.trust Anchor HSM	gladm (p. 24)
cHSM TENANT	<ul style="list-style-type: none"> ▪ cHSM administration ▪ cHSM users ▪ Database backups ▪ Keys ▪ Cryptographic operations 	cHSM	cHSM Toolset (p. 25)

Table 5: u.trust Anchor - Role Summary

2.8.1 Global Initial Administrator ADMIN

When the u.trust Anchor device is started for the first time, the role of the Global Administrator is that of the Global Initial Administrator ADMIN. For the credentials of this user, the Global Initial Admin Key (GIAK) is used, which only provides access to a very restricted set of commands. To be able to use the full functionality of u.trust Anchor, the authentication token of the Global Initial Administrator ADMIN has to be changed to create an admin user without restrictions. For the procedure, see section [Changing the Authentication Token of the Global Initial Administrator](#) (p. 65).



When the execution of a command/function (except `user-change-credentials`) that requires authentication is requested by the user and the GIAK is provided, then a request to change credentials is returned and the message "The user authentication key needs to be updated." is displayed.

3 Concepts

3.1 Device Life Cycle, States and Modes

In error-free operation throughout its life cycle, the u.trust Anchor runs through the following states:

FACTORY DEFAULT -> INITIALIZED



The flowchart below depicts the most common state/mode transitions of a u.trust Anchor, but not all possible cases (especially, if an unforeseen condition occurs).

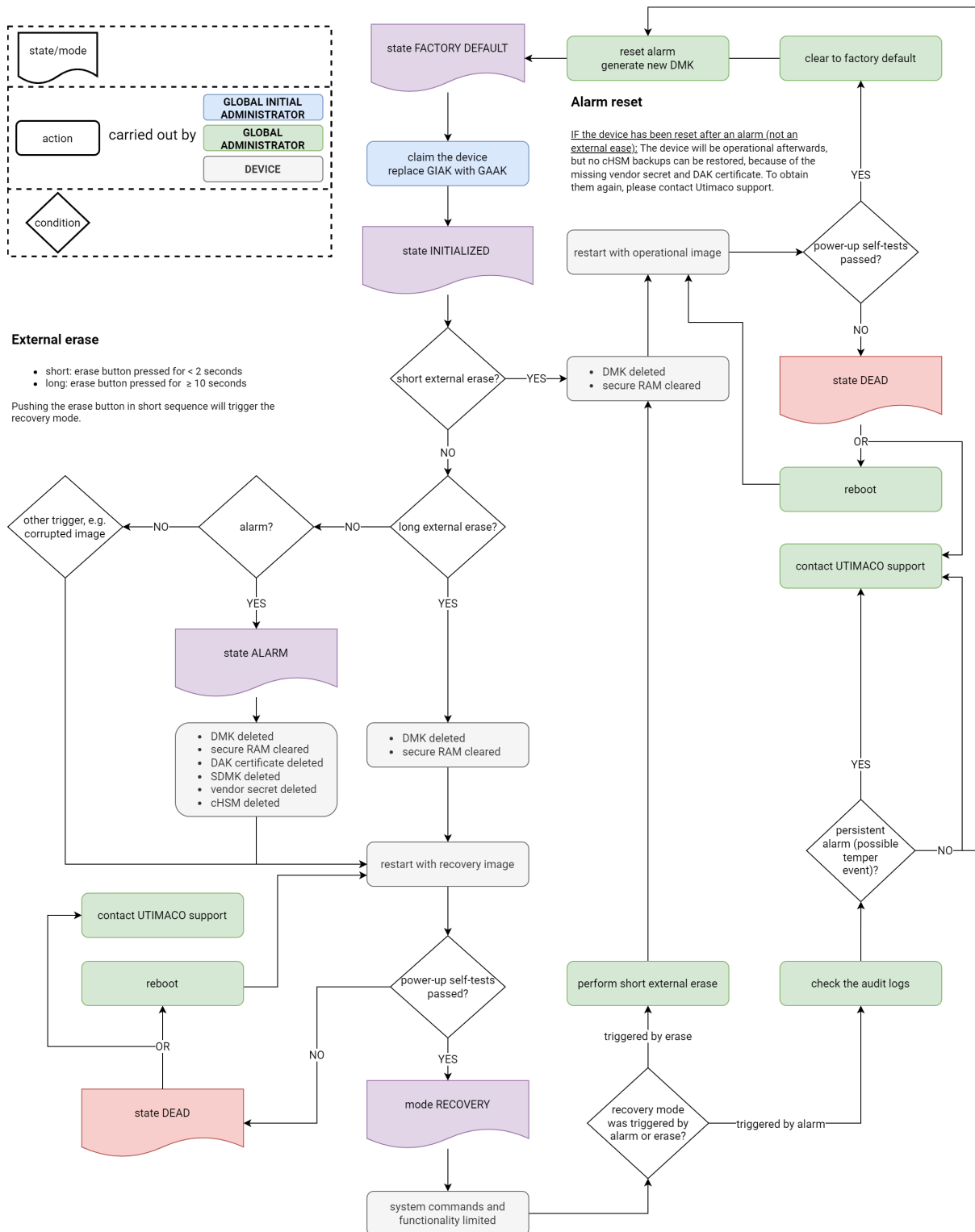


Figure 14 : u.trust Anchor - Device States and Modes Transitions

State	Description
FACTORY DEFAULT	<p>Condition</p> <ul style="list-style-type: none">▪ The device is delivered in the FACTORY DEFAULT state.▪ The <code>global-admin-auth-key-crt</code> and the <code>dak-operator-crt</code> are not present on the device. <p>The device needs to be claimed by the administrator with the Global Initial Admin Key (GIAK) received together with the device to reach full operational mode, see Claiming the Device (p. 56). The FACTORY DEFAULT state can be checked by retrieving the chain of trust via <code>gladm system-get-trust-chain</code>.</p>
INITIALIZED	<p>Condition</p> <ul style="list-style-type: none">▪ The device has been successfully claimed and is operational.▪ The <code>global-admin-auth-key-crt</code> and the <code>dak-operator-crt</code> are present on the device. <p>The INITIALIZED state can be checked by retrieving the chain of trust via <code>gladm system-get-trust-chain</code>.</p>
ALARM	<p>Condition</p> <ul style="list-style-type: none">▪ An alarm has been triggered, see Alarm Mechanism (p. 42) for details. For a full list of commands available in alarm state, see Overview of gladm Commands. <p>Perform the following steps to check whether the device is in the alarm state:</p> <ol style="list-style-type: none">1. Execute the command <code>gladm system-get-info</code>2. If the output returns <code>alarm present</code> and <code>A zeroization event occurred</code>, then the device is in the alarm state. <p>See also Leaving the Alarm State (p. 114).</p>
DEAD	<p>Condition</p> <ul style="list-style-type: none">▪ The device is completely unresponsive or failed the power-up self-tests. <p>See also Leaving the Dead State (p. 116).</p>

MINIMAL OPERATIONAL STATE	<p>Condition</p> <ul style="list-style-type: none"> An error message is returned, when <code>gladm system-get-license-info</code> is executed. <p>If the device detects a corrupted, invalid or missing license file, then only minimal operations are possible. In this state gladm commands can be executed, but no container can be used, i.e. there are no templates and no slots available. To leave this state a valid license file must be loaded via <code>gladm system-update-license</code>. See also The License File is Corrupted, Invalid or Missing (p. 116).</p>
Mode	
RECOVERY	<p>Condition</p> <ul style="list-style-type: none"> The device does not respond to any commands requiring authentication. This can be the case when: <ul style="list-style-type: none"> an alarm has been triggered, all operational images on the device are broken, after the External Erase (p. 98) button has been pressed for more than 10 seconds, or the reboot process has been interrupted several times in short order (a full reboot takes up to 50 seconds). <p>Perform the following steps to check whether the device is in recovery mode:</p> <ol style="list-style-type: none"> Execute the command <code>gladm system-get-info</code>. Check the output of the returned parameter <code>version</code>. If it starts with <code>recovery_</code>, then the device is in recovery mode. <p>See also Leaving the Recovery Mode (p. 113).</p>

Table 6: u.trust Anchor - Device State and Mode Description



See also [How to Identify different Device States \(p. 111\)](#).

3.2 Quorum Requirements

Each command comes with an individual *quorum requirement*. The quorum requirement determines the *eligibility* a user must meet to execute the command. Every time a command execution is requested, it is checked that the device user authenticated in that particular session is eligible for the individual quorum of the command. If the user is not eligible, they cannot execute the command. By default, the initial quorum requirement for most commands

is `1`, which means that a user eligible for the command can execute it on their own. A quorum requirement of `0` means that no authentication is needed to execute the command.



For forward compliance reasons, commands unknown to gladm will be displayed as `command<ID>`.

For some commands, it is required that more than one user eligible for that particular command must be logged in within the same session to execute the command. Since the quorum values are stored in a device-global table of integers, the value returned for each function indicates how many users eligible for contributing to a quorum must be logged in. This means that a command with a quorum requirement of `2` requires two eligible users to be authenticated within the session to execute the command.

Following this logic, each session can authenticate a group of users and execute a command for which they together meet the minimal quorum requirement. To give an example, take a look at the following table:

	Command 1 QR: 1	Command 2 QR: 2	Command 3 QR: 2
User1	1	1	0
User2	0	1	1

If User1 and User2 both authenticate to the same session, the session authentication sums up to (1/2/1) and is allowed to execute command 1 (User1 contributes the required authentication level 1) and command 2 (User1 and User2 both contribute to the required authentication level 2), but not command 3 (only User2 contributes one authentication level, but the command requires two authentication levels).

Configuring Quorum Requirements

A full list of the quorum for each command can be retrieved at any time via gladm `system-get-quorum`. The retrieved `.cfg` file can be adjusted with custom quorum requirements and loaded onto the device via gladm `system-set-quorum`.



The quorum requirements are part of the user backup. When a user backup is restored, the user database is replaced entirely, along with the current quorum requirement configuration from the backup.

Configuring User Eligibility

For each device user, the eligibility to contribute to a quorum can be configured for each specific command by giving a permission file when the user is created via `gladm user-add`.

There are checks in place assuring that the quorum for all commands can always be met by preventing a lockout or soft lock. These checks are performed when deleting a user, restoring a user backup, or changing the quorum requirements. Additionally, the overall user limit is taken into account.

3.3 System Keys

This chapter describes the system keys used in and around u.trust Anchor, how they are generated, who will be responsible for storing them, and the way they are handled and used.

The key hierarchy of the u.trust Anchor device has two root keys: DMK (device master key) and SDMK ("sticky" device master key). The SDMK protects two secrets injected at manufacturing time: the vendor secret (used as part of the cHSM backup process) and the signed Device Attestation Key (DAK) which, along with the vendor certificate, can only be used when the DAK is present. The DMK protects all sensitive data imported onto the device after manufacturing.

Key Name	Abbreviation	Type	Key Use	Generation	Deletion
Device Authentication Key	DAK	NIST P-521	The Device Authentication Key authenticates the Global Admin Key when the u.trust Anchor device is booting. It also authenticates the Container Authentication Key of a cHSM when it is booting.	Present on the device upon first boot Regenerated via <code>gladm system-reset-alarm</code>	Upon alarm occurrence
GLaD Authentication Key	GAK	NIST P-521	The GLaD Authentication Key authenticates GLaD towards the administrator in a secure messaging session. It is itself authenticated by the Device Authentication Key.	Upon first execution of first <code>gladm</code> command	Upon alarm occurrence, External Erase (p. 98) or via <code>gladm system-clear</code>

Key Name	Abbreviation	Type	Key Use	Generation	Deletion
Container Authentication Key	CAK	NIST P-521	The Container Authentication Key authenticates a cHSM towards a cHSM user in a secure messaging session. It is itself authenticated by the Device Authentication Key.	Upon <code>gladm chsm-create</code>	Upon alarm occurrence, External Erase Procedures (p. 98) or via <code>gladm chsm-free-slot</code> or <code>gladm system-clear</code>
Global Initial Admin Key	GIAK	NIST P-521	The Global Initial Admin Key authenticates an initial user towards the device. It is used for unclaimed u.trust Anchor devices to make it possible for the Global Administrator to replace the authentication token, see section Changing the Authentication Token of the Global Initial Administrator (p. 65) .	External Part of the product bundle	Replaced, see Changing the Authentication Token of the Global Initial Administrator (p. 65)
Global Admin Authentication Key	GAAK	ECDSA NIST P-256/521 brainpoolP320t1 RSA >= 2048 bits	The Global Admin Authentication Key authenticates a user towards the device. The public part of the key is imported during the creation of a user via <code>gladm user-add</code> .	External	External Erase (p. 98) or via <code>gladm user-delete</code>
Container Admin Authentication Key	CAAK	RSA >= 512 bit	The CAAK (Container Admin Authentication Key) is the cHSM Admin's public key file provided to the u.trust Anchor Operator after the creation of a cHSM. It is used for user authentication towards the cHSM and verification of cHSM user authentication.	External	Upon alarm occurrence, External Erase (p. 98) or via <code>gladm system-clear</code>

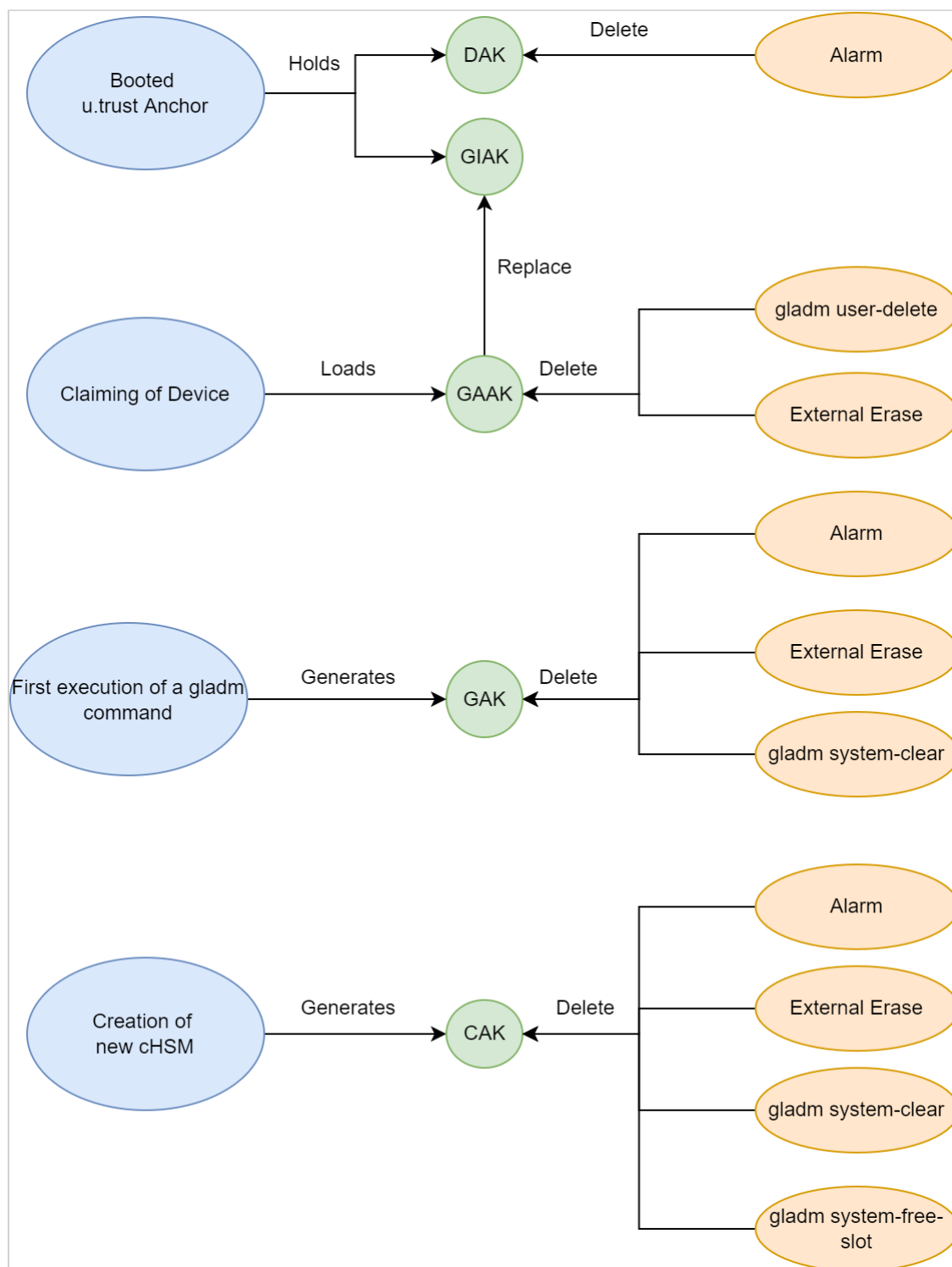


Figure 15 : u.trust Anchor System Keys Life Cycle

3.4 Security Mechanisms

This chapter describes the various security mechanisms of u.trust Anchor, such as user authentication, secure messaging, and treatment of alarms.

Additionally, to ensure secure handling and validity between all involved parties, several security mechanisms are in place to verify that the devices are genuine.

For the u.trust Anchor device, a manufacturer secret is loaded into the device on production and can be used to verify the authenticity of the device before taking it into operation.

See [Verifying the Authenticity of the Device \(p. 62\)](#).

For a u.trust Anchor cHSM, the cHSM Administrator can provide their initial credentials for cHSM creation to the Global Administrator to be used in the creation of the cHSM. This ensures no unwanted access to the cHSM is possible and the cHSM Administrator is the only party with the required authentication key (see [Creating a new cHSM \(p. 79\)](#)).

3.4.1 Secure Messaging

u.trust Anchor supports *Secure Messaging* for the communication between the u.trust Anchor device and the host.

To verify a certificate chain, be sure to include the applicable parameter (i.e. `--vendor=<vendor root certificate>` or `--operator=<operator root certificate>`). If one of the certificates can not be verified, the command will fail. The vendor root certificate `u-trust-Anchor-ROOT-CA-1.cer` is supplied within the product bundle at `/Software/Linux/Administration/key` (for Windows: `\Software\Windows\Administration\key`).

To enable this verification process, the chain of trust has to be established on the device, see section [Claiming the Device \(p. 56\)](#).

Verification of the Operator Certificate

The vendor root certificate and operator root certificates can be verified with the following syntax:

```
gladm -u <username> -k <credentials> --vendor=<vendor root certificate> --operator=<operator root certificate> <gladm command>
```

This procedure can be used to double-check the verification process if, for example, a new operator certificate has been imported, see section [\(1.2.28\) 2020-0035 Importing an Operator Certificate \(p. 68\)](#), to make sure the new operator certificate is used. Giving these

parameters will not change the existing certificates and the command will still fail if the certificates cannot be verified.

Verification of the Vendor DAK Certificate

The vendor DAK certificate can be verified at any point by using the vendor root certificate retrieved from the trust chain of the device via `gladm system-get-trust-chain` with the following openssl command:

```
openssl verify -CAfile vendor_root.pem dak-vendor-chain.pem
```



The verification with openssl can occasionally return some openssl errors. In case this happens, use gladm to verify the certificates.

3.4.2 Alarm and other Zeroization Events

Alarms occur when a physical attack or certain extraordinary physical circumstances are detected that might compromise the security of u.trust Anchor.

Every time an alarm occurs, all secrets within the u.trust Anchor device are actively deleted and the device is automatically restarted in the alarm state with limited command functionality. For a full list of commands available in alarm state, see [Overview of gladm Commands \(p. 121\)](#) in the Global Administration Management (gladm) Tool section of the Appendix.

The execution of an [external erase \(p. 98\)](#) as well as the `gladm system_clear` command also lead to the deletion of sensitive data, for comparison see [Zeroization Event Overview \(p. 41\)](#).

3.4.2.1 Zeroization Event Overview

	Alarm (p. 42)	External Erase (p. 98)	Clearing the Device (p. 96)	Clearing to Factory Default State (p. 97)
Trigger	see Alarm Triggers (p. 43)	physical access	authenticated <code>gladm system_clear</code> command	unauthenticated <code>gladm system_clear</code> command, if preceded by an external erase
GAAK (p. 37)	preserved	preserved	preserved	restores Global Initial Administrator with GIAK
Device Audit Log (p. 103)	preserved	preserved	preserved	preserved

Device Boot Log	cleared	cleared	preserved	cleared, if preceded by an external erase
DMK (p. 37) cHSM cHSM Audit Log cHSM Boot Log	cleared	cleared	cleared	cleared
SDMK (p. 37) DAK (p. 37) Vendor Secret	cleared	preserved	preserved	preserved
Secure RAM	cleared	cleared	unused data is zeroized, other data is preserved	cleared, if preceded by an external erase

Table 7: Zeroization Commands and Events Summary

3.4.2.2 Alarm Handling and Environmental Failure Procedure

To ensure that the security of the u.trust Anchor cannot be compromised by physical attacks and/or by extreme environmental conditions, the cryptographic module implements a tamper detection mechanism and measures for *Environmental Failure Protection (EFP)*.

EFP is provided as part of u.trust Anchor's more general alarm mechanism, which is described in the following section. The u.trust Anchor's behavior in extreme temperatures (which exceeds its reaction according to the alarm mechanism) is described in section [Temperature-dependent Behavior of u.trust Anchor \(p. 106\)](#).

Since an alarm can only occur directly on the u.trust Anchor device and not for the cHSMs running on it, the Global Administrator is the only one responsible for handling alarms. For details on the actions to be taken, see [Alarm Mechanism \(p. 42\)](#).

3.4.2.2.1 Alarm Mechanism

If the sensory controller (powered by the on-board battery) detects a critical tamper event (see [Alarm Triggers \(p. 43\)](#)), then the following steps are taken immediately on powered down and active devices:

1. The alarm is registered as valid.
2. The Sticky Device Master Key (SDMK), Device Master Key (DMK), the Device Authentication Key (DAK), the vendor secret, all cHSMs, the cHSM audit log, the cHSM boot log and the secure RAM are cleared. The deletion of both master keys invalidates all data protected by these keys.
3. The processor is then restarted.

The user can still communicate with the device after a tamper event, and perform a reset of the alarm via `gladm system-reset-alarm` to acknowledge the event. The acknowledging of the event will regenerate the DMK, but will not be able to restore the manufacturer secrets (vendor secret and DAK, including the manufacturer certificate).

The clearing of the Master Keys DMK and SDMK takes place on an active device as well as on a device that is powered down. The clearing of caches and secure RAM can only be addressed on an active module, and therefore has to be performed as part of a power-down process. See also [Secure RAM on System Reset or Power Down \(p. 44\)](#).

If the reason for the alarm did not persist, like in the case of an [external erase \(p. 98\)](#) or a temperature alarm, the alarm only has *occurred* and the Global Administrator can [reset the alarm \(p. 168\)](#) via `gladm system-reset-alarm`. Resetting the global alarm triggers the new generation of the erased Master Keys. Afterward, the device can be set up again.

For more information, see [Resetting the Alarm \(p. 168\)](#).



Even though the u.trust Anchor device will be operational after resetting the alarm, no cHSM snapshots can be loaded and no backups can be restored, because of the missing vendor secret and vendor DAK certificate. To obtain them again, please contact Utimaco to send the device back in for maintenance, see section [Contact Address for Support Queries \(p. 118\)](#).

3.4.2.2.2 Alarm Triggers

Alarms occur directly on the u.trust Anchor device. If an alarm occurs for the u.trust Anchor device, the cHSM is stopped and deleted, along with the Device Master Key, which makes all secrets stored on a u.trust Anchor cHSM non-usable since it was encrypted with the Device Master Key.

<i>Physical alarms noticed by the sensory</i>	
<i>Notation</i>	<i>Explanation</i>
Temperature too low	Temperature too low (below -17 °C (1.4 °F))
Temperature too high	Temperature too high (above 77 °C (170.6 °F))
Power too high	Voltage / Tension above approx. 13.5 V (u.trust Anchor supply voltage)
Power too low	Voltage / Tension below approx. 2.11 V (u.trust Anchor-internal retention voltage)
Upper tamper wire destroyed	Mechanical attack has destroyed the upper tamper wire
Bottom tamper wire destroyed	Mechanical attack has destroyed the bottom tamper wire

External Erase	An external erase has been executed (manually by pressing the Erase push-button). This is not an actual alarm and the handling differs in a few points, see below
Other alarm triggers	
Notation	Explanation
Sensory Controller failed	No reaction from the sensory controller
Power failed	Sensory controller without power The sensory controller, which has its own battery power supply, has been down. If the complete power supply has been down (e.g. battery empty) or if the sensory controller is started for the first time, it starts with this alarm reason set



Resetting to Factory Default with the External Erase Button

The **External Erase** button is the only physically reachable interface to the u.trust Anchor that is always available as long as the device is present, even in case of a non-booting – and hence non-communicative – device. Pressing the **External Erase** button for no more than 2 seconds causes an alarm signal which leads to the erasure of the secure memory with the exception of the Sticky Master Key and all secrets which are encrypted with the Sticky Master Key. These device secrets are only deleted in case of a real alarm.

Pressing the **External Erase** button for 10 seconds or longer will trigger a boot into recovery mode, leaving the device unresponsive to most commands. See [External Erase \(p. 98\)](#) for more information.

3.4.2.2.3 Secure RAM on System Reset or Power Down

If the system is powered down and the boot procedure has not finished, the mechanism for clearing the caches and the secure RAM is not active. To prevent any attacks at one of these states, the clearing procedure for caches and secure RAM has to be applied before any System Reset and Power Down takes place.

System Resets are exclusively controlled by the Platform Management Unit (PMU). Failing of the main power is reported directly to the PMU via interrupt line. In case of failure of the main power (e.g. powering down the system), an adequate number of capacitors on the device ensures that enough power for erasing the secure RAM is still available.

3.5 cHSM Management

u.trust Anchor introduces the concept of a Containerized Hardware Security Module, abbreviated *cHSM*. Depending on the used template to create the cHSM, different functionalities are available. For more details about templates, see [cHSM Templates \(p. 47\)](#).

3.5.1 cHSM Operating Modes

A cHSM can be in one of two operating modes: regular mode or cluster mode.

The mode of a cHSM can not be set directly. Instead, the command used to load the cHSM implicitly sets its mode.

Regular Mode

A cHSM is in regular mode when it is loaded onto the device by either creating a new cHSM via `gladm chsm-create` or by restoring a cHSM from a snapshot via `gladm chsm-restore`.

cHSMs in regular mode operate in a standalone manner and are meant to be used for configuration by the customer. This mode has no functional limitations.

Cluster Mode

u.trust Anchor offers the possibility to operate several cHSMs grouped together as a cluster.

A cHSM cluster is a number of cHSMs that were created from the same snapshot through one or multiple clone actions of a cHSM via `gladm chsm-clone`.

As a result of being loaded into the device through cloning, all of the cHSMs created this way operate in cluster mode. Therefore, they do not allow any operations that would result in their internal states going out of sync. This also restricts action that can be taken by the u.trust Anchor cHSM Administrator. For detailed information on which commands are blocked on clustered cHSMs, see *Cluster Mode* in [u.trust Anchor - Containerized Hardware Security Module \(cHSM\) - Administration Manual \(p. 119\)](#).

To find out if several cHSMs belong to a cluster, their UID has to be compared, which is the identifier for the cluster. The UID of a cHSM is listed when performing `gladm chsm-list-slots`.

3.5.2 cHSM States

During its lifecycle, a cHSM can enter a variety of different states.

The state of a cHSM does reflect its virtual power state and determines which management operations can be carried out.

The state of a cHSM can be displayed by listing all cHSM slots via `gladm chsm-list-slots`.

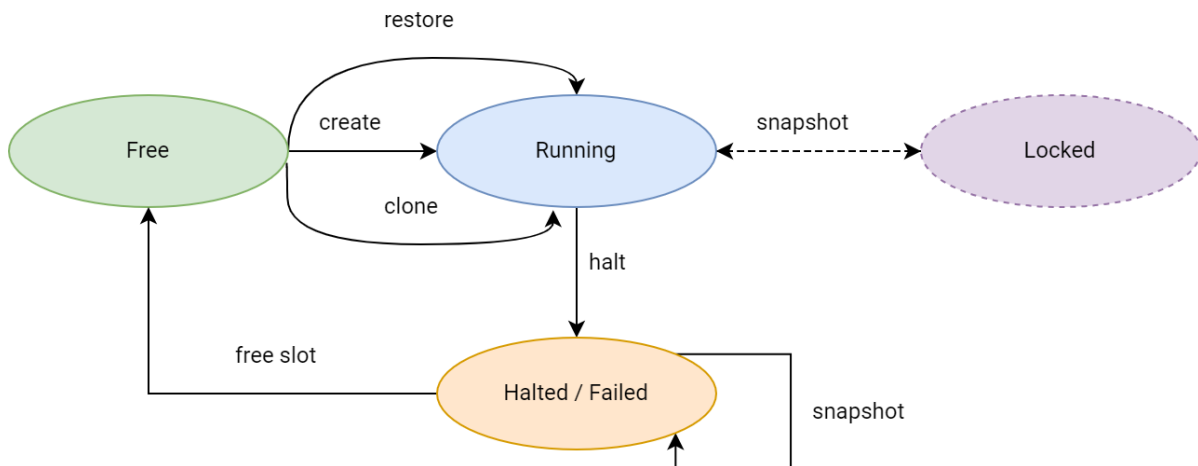



Figure 16 : cHSM States

State	Description
Free / None	The associated cHSM slot is not occupied by a cHSM and therefore free for use to create or clone a cHSM to the slot.
Running	<p>The cHSM is operational and can respond to requests. While it is running, certain metrics or logging information can be retrieved. Depending on its operating mode, see section cHSM Operating Modes (p. 45), it is also possible to take snapshots, see section Taking Snapshots (p. 48). The cHSM can be put in a halted state by halting the cHSM via <code>gladm chsm-halt</code>.</p> <hr/> <p> Currently, a cHSM is displayed in running mode even if it is still booting. Therefore, the booting cHSM might not be available for further requests right away.</p> <hr/>
Halted	<p>The cHSM has been put in halted state via <code>gladm chsm-halt</code> and therefore not running. It will not respond to requests. It is not possible to directly put a cHSM back into a running state, but data can still be obtained through snapshots. Since the cHSM cannot be directly restarted, the retrieved data is guaranteed to reflect the most recent state of the cHSM. Limitations imposed by the operation mode of the cHSM still apply, see section cHSM Operating Modes (p. 45). Once all desired data has been obtained, the associated cHSM slot can be freed via <code>gladm chsm-free-slot</code>.</p>
Failed	<p>The cHSM has encountered an error during or between cHSM management operations and is not running. This state is identical to the Halted state and the slot can be freed via <code>gladm chsm-free-slot</code>.</p>

State	Description
Locked	The cHSM is temporarily locked by an administrative operation. It is not running and will not respond to any requests.

3.5.3 cHSM Templates

The u.trust Anchor provides templates for the creation of cHSMs. Depending on which template is used to create a cHSM, it has different attributes and functionalities. All templates support the usage of an alternative module signature key (AMSK) to load a custom firmware module.

Template	Description	Standard Package	FIPS Package (certifiable)	GP CC Package (certifiable)
SecurityServer	Creates a standard <i>cHSM</i> .	✓	✓	✓
SecurityServer-SDK	Creates an <i>SDK cHSM</i> .	✓	✗	✗
SecurityServer-FIPS	Creates a <i>FIPS cHSM</i> according to the standards defined in FIPS-140-3. FIPS cHSMs block a number of functions, see section <i>Availability of Commands in FIPS mode</i> in <i>u.trust Anchor FIPS 140-3 - Containerized Hardware Security Module (cHSM) - Administration Manual</i> .	✓	✓	✗
SecurityServer-FIPS-SDK	The FIPS-SDK template is the merge between the SDK and FIPS templates. It has the SDK behavior plus FIPS restrictions.	✓	✗	✗
SecurityServerCC	Creates a <i>CC cHSM</i> according to the standards defined in Common Criteria (EAL4+).	✗	✗	✓

Table 8: Templates



SDK templates are only available with a valid u.trust Anchor SDK license.



In this manual, all further mentions of *cHSM* refer to all kinds of cHSMs, and all specific differences are pointed out by stating the respective information for *SDK cHSMs* where applicable.

3.5.4 Taking Snapshots

For the time that a cHSM is not required by the customer, the customer can leave the u.trust Anchor device as an encrypted cHSM snapshot.

A snapshot contains the data of a cHSM stored on disk at the time of taking the snapshot. It contains the user data created after the cHSM was created. Snapshots can be taken of running, halted, or locked cHSMs. When taking a snapshot of a running cHSM, the cHSM is temporarily halted and remains unavailable until the snapshot operation has been completed. Taking snapshots of cHSMs in cluster mode is not supported.

A cHSM snapshot is encrypted by using both the fixed vendor secret and an operator secret, which is a binary string with a length of 32 bytes that can be configured through gladm, see Importing the Operator Secret. Note that if you take a snapshot of a cHSM and afterward load a new operator secret into the u.trust Anchor device, the snapshot taken previously with the old operator secret can not be loaded into the device until the operator secret that was active when the snapshot was taken is active again.

The following gladm commands create snapshots of a cHSM:

- `gladm chsm-snapshot` creates a snapshot of a cHSM
- `gladm chsm-retrieve` halts a cHSM, creates a snapshot, and retrieves the cHSM from the device.

4 Setup

The following chapters describe the installation of tools to manage u.trust Anchor and cHSMs, as well as how to claim the device and set up the chain of trust.



All returned certificates, certificate chains, and CSRs are PEM-encoded.

For detailed information on the keys used in this chapter, see [System Keys \(p. 37\)](#)

4.1 Installing the Host Software



This section applies to Windows operating systems only.

Prerequisites

To run the GUI tools, your Java installation will need to support unlimited crypto. If your system does not have a JRE, download one from <http://openjdk.java.net>. Next, install the corresponding Java security policy files, for example “UnlimitedJCEPolicyJDK11.zip” from the openjdk.java.net¹ website. Extract them and copy the *.jar files to your /lib/security directory.

This section describes the installation using the GUI. If you want to perform a silent installation, see [Installing the Host Software Without User Interaction \(p. 51\)](#).

Procedure

1. In the highest folder level of the product bundle, click the file `SecurityServer-<version number>.msi`.



If necessary, you will be prompted to install the Microsoft runtime environment (VCRedist). Click **OK** to confirm the corresponding dialog box.

2. In the installation wizard, click **Next**.

¹ <http://openjdk.java.net>

3. In the **Select Installation Folder** dialog, use the **Browse...** button to select a different directory for installing the software or confirm the default installation directory.
4. Click **Next**.
The **Installation type** dialog box opens.

Typical

The u.trust Anchor administration tools (gladm, CAT, csadm etc.) and the u.trust Anchor documentations are installed.

Custom

This installation type lets you specify the features to be installed.

Complete

This corresponds to selecting all items of the custom installation except for the PCIe driver and PIN Pad driver.

Select the installation type of your choice.

5. In the case of custom installation, the **Select the features to be installed** dialog box opens. Select the features you want to install. By default, **Administration** and **Documentation** are selected.
6. Click **Next**.
7. In the **Ready to Install** dialog box, click **Install**.
8. After completing the installation, the **Completing the SecurityServer Setup Wizard** dialog box opens. Here you can specify whether the CAT should be launched automatically after successful software installation. By default, the **Launch SecurityServer** option is selected.
9. Click **Finish** to complete the software installation.

You have now finished installing the u.trust Anchor software. CAT starts now by default and the **CryptoServer Devices** dialog box is displayed.

4.2 Installing the Host Software Without User Interaction



This section applies to 64-bit Windows operating systems only.

The installation is done by performing a command with the following syntax:

```
msiexec /i SecurityServer-<version>.msi <parameters> /qn
```

`SecurityServer-<version>.msi` is the installation file you find in the topmost directory of the product bundle. `<parameters>` are optional parameters specifying the optional features to be installed.

Parameter	Description
	<p>If no parameter is used, a default installation is performed. The administration tools (gladm, csadm, CAT etc.) and the documentations are installed.</p> <p>This corresponds to the parameter <code>INSTALLLEVEL= 3</code>.</p> <p>This corresponds to the preselected items Administration (CryptoServer Administration Tools) and Documentation (CryptoServer Documentation) in the GUI of the user-interactive installation, see the step to select features in Installing the Host Software (p. 49).</p>
DOCUMENTATION_ONLY	<p>This corresponds to selecting only Documentation (CryptoServer Documentation) in the GUI of the user-interactive installation, see the step to select features in Installing the Host Software (p. 49).</p>
PKCS11=1	<p>This corresponds to selecting PKCS#11 (PKCS#11 R3 – Cryptographic Token Interface) in the GUI of the user-interactive installation, see the step to select features in Installing the Host Software (p. 49).</p>
CXI=1	<p>This corresponds to selecting CXI > CXI_C (CXI – Cryptographic eXtended Interface) in the GUI of the user-interactive installation, see the step to select features in Installing the Host Software (p. 49).</p>
CXI_JAVA=1	<p>This corresponds to selecting CXI > CXI_Java (CXI– Cryptographic eXtended Interface Java) in the GUI of the user-interactive installation, see the step to select features in Installing the Host Software (p. 49).</p>
CSPCNG=1	<p>This corresponds to selecting CSP/CNG (CSP/CNG – Cryptographic Service Provider for Microsoft Windows) in the GUI of the user-interactive installation, see the step to select features in Installing the Host Software (p. 49).</p>

Parameter	Description
CSPCNGSRV=1	This corresponds to selecting CSP/CNG > CSP Login Service (CSP/ CNG Interactive Log-in Support) in the GUI of the user-interactive installation, see the step to select features in Installing the Host Software (p. 49) . This includes CSP/CNG.
JCE=1	This corresponds to selecting JCE (JCE Provider) in the GUI of the user-interactive installation, see the step to select features in Installing the Host Software (p. 49) .
JCE2=1	This corresponds to selecting JCE2 (JCE2 Provider) in the GUI of the user-interactive installation, see the step to select features in Installing the Host Software (p. 49) .
EKM=1	This corresponds to selecting EKM (EKM – Extensible Key Management) in the GUI of the user-interactive installation, see the step to select features in Installing the Host Software (p. 49) .
OPENSSL=1	This corresponds to selecting OpenSSL (OpenSSL PKCS11 engine) in the GUI of the user-interactive installation, see the step to select features in Installing the Host Software (p. 49) .
PCIEDRV=1	This corresponds to selecting Drivers > CryptoServer (CryptoServer PCIe driver) in the GUI of the user-interactive installation, see the step to select features in Installing the Host Software (p. 49) . This includes the PCIe driver and the CryptoServer driver.
PPDRV=1	Installation of the drivers for the PIN pads „REINER SCT cyberJack“ and „Utimaco cyberJack one“. This corresponds to selecting Drivers > cyberJack (PIN pad driver) in the GUI of the user-interactive installation, see the step to select features in Installing the Host Software (p. 49) . This corresponds as well to performing the instructions in <i>Setting up the PIN Pad for Windows</i> in the <i>u.trust Anchor - Containerized Hardware Security Module (cHSM) - Administration Manual</i> .
PPD=1	This corresponds to selecting PPD (PPD – PIN Pad Daemon) in the GUI of the user-interactive installation, see the step to select features in Installing the Host Software (p. 49) . For details about the PIN pad daemon, see <i>Using a PIN Pad</i> in the <i>u.trust Anchor - Containerized Hardware Security Module (cHSM) - Administration Manual</i> .
INSTALLLEVEL=3	This corresponds to selecting Administration (CryptoServer Administration Tools) and Documentation (CryptoServer Documentation) in the GUI of the user-interactive installation, see the step to select features in Installing the Host Software (p. 49) . These items are selected by default.

Parameter	Description
INSTALLLEVEL=1000	<p>This corresponds to selecting all items except for PCIe Driver and PIN Pad driver in the GUI of the user-interactive installation, see the step to select features in Installing the Host Software (p. 49). This corresponds to a complete installation.</p> <p>This corresponds as well to using the following collection of parameters:</p> <pre>PKCS11=1 CXI=1 CXI_JAVA=1 CSPCNG=1 CSPCNGSRV=1 JCE=1 JCE2=1 EKM=1 OPENSLL=1 PPD=1</pre>
INSTALLLEVEL=1001	<p>This corresponds to selecting all items including PCIe Driver and PIN Pad driver in the GUI of the user-interactive installation, see the step to select features in Installing the Host Software (p. 49).</p> <p>This corresponds as well to using the following collection of parameters:</p> <pre>PKCS11=1 CXI=1 CXI_JAVA=1 CSPCNG=1 CSPCNGSRV=1 JCE=1 JCE2=1 EKM=1 OPENSLL=1 PCIEDRV=1 PPD=1 PPDRV=1</pre>
APPDIR	<p>Absolute installation path of the host software. Default : C:\Program Files\Utimaco\SecurityServer</p> <p>If the installation path contains at least one space, quotation marks at the beginning and the end of the installation path are mandatory. Example:</p> <pre>APPDIR="C:\My SecurityServer path"</pre> <p>Using APPDIR corresponds to selecting the installation path in the GUI of the user-interactive installation, see Installing the Host Software (p. 49).</p>

Table 9: Parameters for an installation without user-interaction

All listed parameters are optional. Any combinations of the parameters are supported. Omit the parameters that should not be used. Additional msixec parameters like `/l` for logging can be used as well.

Examples:

- Default installation (administration tools and documentations)

```
msiexec /i SecurityServer-<version>.msi /qn
```

or

```
msiexec /i SecurityServer-<version>.msi INSTALLLEVEL=3 /qn
```

- Default installation plus the PKCS#11 R3 cryptographic token interface

```
msiexec /i SecurityServer-<version>.msi PKCS11=1 /qn
```

- Complete installation

```
msiexec /i SecurityServer-<version>.msi INSTALLLEVEL=1000 /qn
```

- Complete installation plus the installation of the PCIe driver

```
msiexec /i SecurityServer-<version>.msi INSTALLLEVEL=1000 PCIEDRV=1 /qn
```

- Complete installation plus the installation of the PIN pad driver

```
msiexec /i SecurityServer-<version>.msi INSTALLLEVEL=1000 PPDRV=1 /qn
```

- Complete installation plus the installation of the PCIe driver and the PIN pad driver

```
msiexec /i SecurityServer-<version>.msi INSTALLLEVEL=1001 /qn
```

4.3 Installing gladm

gladm is used to perform administrative tasks on the u.trust Anchor device.

Installing gladm on a computer with a Windows operating system

This section describes how to install gladm on a Windows administration computer.

On an administration computer running a Windows operating system, gladm is installed by default during the installation of the u.trust Anchor software provided in the product bundle.

The following steps describe how to install gladm on a Windows host computer.

You will find the `gladm.exe` file for Windows in the product bundle here:

- For Windows 32-bit operating systems
Not supported
- For Windows 64-bit operating systems
`Software\Windows\Administration\`

1. Copy the `gladm.exe` file to a well-chosen directory.
2. Add this directory to the `PATH` environment variable to be able to call the Administration Tool from any other directory.



gladm has been successfully installed on the administration computer.

Installing gladm on a computer with a UNIX-like operating system

This section describes how to install gladm on a Linux administration computer.

The gladm installation file is provided within the product bundle under `.../Software/Linux/Administration`.

1. Create a `~/bin` directory in your user directory, if there is not one yet:

```
mkdir ~/bin
```

2. Copy the gladm relevant for your operating system into the `~/bin` directory. An example for Linux 64-bit:

```
cp <mount point of the product bundle>/Software/Linux/Administration/gladm  
~/bin
```

3. Ensure that you have write and execute permissions for gladm.

```
chmod -R u+w+x ~/bin
```

4. Add the `~/bin` directory to the path in the user configuration file in the shell that is being used. In this example, a bash is used as the shell, i.e., open the `~/.bashrc` file and add the following line to it:

```
export PATH=$PATH:~/bin
```

5. Save the changes and close the `~/.bashrc` file.



gladm has been successfully installed on the administration computer.

4.4 Claiming the Device

A u.trust Anchor device leaving the manufacturer's secure environment and being delivered to the customer is in *FACTORY DEFAULT* state. The device is handed to the customer together with the Global Initial Admin Key (GIAK) for initial authentication. The public key of the Utimaco HSM root certificate should be provided for additional verification.

In this state, the Global Administrator has to change the Global Initial Admin Key (GIAK) into an individual Global Admin Authentication Key (GAAK) to change the device state to *INITIALIZED*. The Global Administrator can now create a Wrapping Key, wrap the operator secret, and import the wrapped operator secret back to the device.



A u.trust Anchor device can be reset to the *FACTORY DEFAULT* state by performing an external erase and clearing the device via `gladm system-clear`. See section [Clearing to FACTORY DEFAULT State \(p. 97\)](#).

4.4.1 Global Admin Management Tool (gladm)

The Global User Administration Tool *gladm* is used by the Global Administrator to configure the device, and to set up and maintain cHSMs.

For a complete overview of all available gladm commands and their detailed description, please refer to the [Global Administration Management \(gladm\) Tool \(p. 120\)](#) section of the [Appendix \(p. 120\)](#).

This section describes the syntax of gladm and the dependencies of commands and arguments.

The basic usage for gladm can be expressed as follows:

```
gladm -u admin -k /Software/Linux/Administration/key/gaak.pem -d <addr> [-p  
<port no.>] <command> [<args>...]
```

For Windows, replace the path as follows: `\Software\Windows\Administration\key\`

Parameter	Description
<code>gladm</code>	Addressing the gladm application
<code>-u admin</code>	User specifier, with <code>admin</code> being the example for the user name
<code>-k /Software/Linux/Administration/key/gaak.pem</code>	Key file specifier, with <code>/Software/Linux/Administration/key/gaak.pem</code> being the example key file of the user. For Windows, replace the path as follows: <code>\Software\Windows\Administration\key\</code>
<code>-d <addr></code>	Device specifier/address of the u.trust Anchor device. Mandatory for all gladm commands except for the <code>gladm bash-completion</code> command where the device specifier is not needed. The device specifier must be used regardless of whether the u.trust Anchor device is accessed locally or remotely. There is no default value. If no value is specified, the following error message is output: <code>gladm error: No device was given. Please specify the device using the device specifier. Values:</code> <ul style="list-style-type: none"> ▪ Local (Linux): <code>/dev/cs2.0</code> ▪ Local (Windows): <code>PCI:0</code> ▪ Local (Linux/Windows): <code>127.0.0.1</code> or <code>localhost</code> ▪ Remote (Linux/Windows): IP address or hostname
<code>[-p <port no.>]</code>	Optional port number of the u.trust Anchor device. Default if the parameter is not set: 4000. This default value is configurable by the <code>Port</code> parameter in the <code>[ListenerGlad]</code> section of the <code>csard.conf</code> configuration file of the CSAR daemon (csard) or of the <code>/etc/csxlan.conf</code> configuration file of the u.trust Anchor LAN device. If the <code>-d</code> parameter is <code>/dev/cs2.0</code> or <code>PCI:0</code> , the <code>-p</code> parameter is irrelevant. It is ignored.
<code><command></code>	The command to be executed
<code>[<args>...]</code>	The arguments of the command to be executed

Any option given before a command is considered global and may apply to or override any following command. Arguments that are given after a command apply to that specific command.

Required arguments are typically positional and require no flags as an indicator. Optional arguments are commonly expressed as flags and can be given in both a short and a long syntax (e.g. `-v` and `--version`).

For most of the commands available through gladm, user authentication has to be given. The user needs to provide their username and the public part of their credential key.

Example

The user `admin` has the public part of the credential key saved to `/Software/Linux/Administration/key/gaak.pem` and wants to perform the command `system-reset-alarm`. The syntax, including authentication, is as follows:

```
gladm -u admin -k /Software/Linux/Administration/key/gaak.pem -d 123.123.123.123 system-reset-alarm
```



The authentication has to be given at every command execution, except for the commands `system-get-info`, `user-list`, `chsm-list-slots`, `system-get-quorum`, `system-get-time`.

Example

If the quorum requirement of a command requests multiple eligible users to be authenticated, the usernames and credential keys have to be given with the following syntax:

```
gladm -u user1 -k /Software/Linux/Administration/key/gaak_user1.pem -u user2 -k /Software/Linux/Administration/key/gaak_user2.pem -d 123.123.123.123 system-reset-alarm
```



gladm is intended for single-threaded use only. It is not recommended to run multiple instances concurrently since this may result in unexpected behavior. Using the TCP daemon, it is ensured no concurrent commands are issued.

All gladm commands can globally use the following parameters:

Shortcut	Command	Description
-h	--help	Show help
-v	--version	Show version
-d	--device=addr	Device specifier/address
-p	--port=port	Port

Shortcut	Command	Description
-g	--debug	Write debug message to stdout
-q	--quiet	Disable status messages
-t	--timeout-msec=<val>	Timeout for command execution in milliseconds Default if the parameter is not set: 60 msec
	--timeout-transport-msec=<val>	Timeout for command execution in milliseconds Default if the parameter is not set: 60 msec
-u	--users=<val>	Device users
-k	--keyfiles=<val>	Device user keyfiles
-y	--vendor=<val>	Utimaco Root Certificate
-x	--operator=<val>	Operator Root Certificate

Table 10: gladm Global command parameters



To see a list of all available commands, the command `gladm --help` can be used. To list all required and optional arguments for a command, enter `gladm <command> --help`.

4.4.1.1 Keys for User Authentication

To perform gladm commands, the required number of eligible users to authenticate the command must authenticate with each given gladm command.

The authentication for each user is given within the command syntax. For details, see [Global Admin Management Tool \(gladm\) \(p. 56\)](#).

For authentication credentials, users can use the following keys as encrypted key files:

All keys can be stored as encrypted key files. To generate such a key as an encrypted key file, the csadm command `csadm GenKey` (see *u.trust anchor - Containerized Hardware Security Module (CHSM) - Administration Manual*) can be used.

RSA keys and EC keys with brainpoolP320t1 can also be stored on smartcards.

Alternatively, the openssl command `genpkey` can be used for key generation. In this case, the key can not be stored on a smartcard.

Example for a key generation with openssl:

```
openssl genpkey -algorithm ec -pkeyopt ec_paramgen_curve:secp521r1 -aes-256-cbc
-out admin_gaak_enc.key
openssl ec -pubout -outform pem -in admin_gaak_enc.key -out admin_gaak_pub.key
```

4.4.2 Specifying a Device

The u.trust Anchor device to which a command is to be sent is specified by the global `-d` argument (device specifier/address). This argument is mandatory for all gladm commands except for the `gladm bash-completion` command where the device specifier/address is unnecessary because, in this case, a connection to the u.trust Anchor PCIe card is not needed. The device specifier must be used regardless of whether the u.trust Anchor device is accessed locally or remotely or whether the u.trust Anchor PCIe card is mounted in a Linux or a Windows computer or in a u.trust Anchor LAN device.

There is no default value for the device specifier. If no value is specified, the following error message is output:

```
gladm error: No device was given. Please specify the device using the device specifier.
```

The device specifier may have the following values:

- Local access

- Local access via the device node

In this case, no port is used because the communication is not transmitted via the network. The `-p <port>` argument is not needed. No default value of this argument is used, i.e., the `Port` parameter in the `[ListenerGlad]` section of the `csard.conf` configuration file or the `/etc/csxlان.conf` configuration file is not used. If the `-p` argument is set in the gladm command (example:

```
gladm ... -d /dev/cs2.0 -p 5555 ...), it is ignored.
```

- Specifying a u.trust Anchor PCIe card mounted in a local Linux computer or a local u.trust Anchor LAN device:

```
gladm ... -d /dev/cs2.0 ...
```

- Specifying a u.trust Anchor PCIe card mounted in a local Windows computer:

```
gladm ... -d PCI:0 ...
```

- Local access via the network

Specifying a u.trust Anchor PCIe card mounted in a local Linux/Windows computer or a local u.trust Anchor LAN device

Examples:

- `gladm ... -d 127.0.0.1 ...`
 - `gladm ... -d 127.0.0.1 -p 4000 ...`

- `gladm ... -d 127.0.0.1 -p 5555 ...`
- `gladm ... -d localhost ...`
- `gladm ... -d localhost -p 4000 ...`
- `gladm ... -d localhost -p 5555 ...`

▪ Remote access

Specifying a u.trust Anchor PCIe card mounted in a remote Linux/Windows computer or a remote u.trust Anchor LAN device

- Using the IP address of the remote computer or LAN device:

Examples:

- `gladm ... -d 123.123.123.123 ...`
- `gladm ... -d 123.123.123.123 -p 4000 ...`
- `gladm ... -d 123.123.123.123 -p 5555 ...`

- Using the hostname of the remote computer or LAN device

Examples:

- `gladm ... -d myPC ...`
- `gladm ... -d myPC -p 4000 ...`
- `gladm ... -d myPC -p 5555 ...`
- `gladm ... -d myUtaLan ...`
- `gladm ... -d myUtaLan -p 4000 ...`
- `gladm ... -d myUtaLan -p 5555 ...`

The optional `-p` argument indicates the port that is used on the computer or LAN device. Default if this argument is not set: 4000. If you use a u.trust Anchor PCIe card mounted in a Linux/Windows computer, this default value is configurable by the `Port` parameter in the `[ListenerGlad]` section of the `csard.conf` configuration file of the CSAR daemon (csard). If you use a u.trust Anchor LAN device, this default value is configurable by the `Port` parameter in the `[ListenerGlad]` section of the `/etc/csxlan.conf` configuration file of the u.trust Anchor LAN device.

Example of the configuration file:

```
...
[CryptoServerGlad]
Label = GladCS1
Timeout = 30000
Device = /dev/cs2.0
```

```
[ListenerGlad]
Port = 4000
Keepalive = 1
Route_To = GladCS1
...
```

Do not confuse this `-p` argument indicating a port with the following `-p` arguments in the following gladm commands:

- `gladm ... user-add [-p <val>] ...`
- `gladm ... chsm-create ... [-p <directory_path>] ...`
- `gladm ... system-get-trust-chain [-p <directory path>]`

The `-p` argument indicating a port is identified by gladm by the position of the `-p` argument within the gladm command. For example, a command

```
gladm -d <device specifier/address> -p <port> ... user-add [-p <val>] ...
```

is supported, but a command

```
gladm ... user-add [-p <val>] ... -d <device specifier/address> -p <port>
```

is not.

4.4.3 Verifying the Authenticity of the Device

When setting up the device for the first time, you should perform the following steps to ensure the authenticity of the device before uploading your certificates.



Note that this step is optional. It is necessary if you want to verify the HSM is genuine and manufactured by Utimaco. This step is expressly recommended from a safety perspective.

To perform the following commands, the device needs to be addressed correctly as described in *Specifying a Device* in the *u.trust Anchor - Administration Manual*, where the placeholder `<device>` is given.

1. Get the system information via the `gladm system-get-info` command, see the *Displaying Device System Information (system-get-info)* chapter of the *u.trust Anchor - Administration Manual*.

```
gladm -d <device> system-get-info
```

The output should contain the following lines:

```
Initial user credentials unchanged
No alarm present
No zeroization event occurred
Vendor Secret is present on the device
Vendor DAK Certificate is present on the device
```

2. Use the Utimaco root certificate `u-trust-Anchor-ROOT-CA-1.cer` that is delivered within the product bundle at `/Software/Linux/Administration/key` (for Windows: `\Software\Windows\Administration\key`) as an additional verification parameter in a `gladm` command, for example with the command `gladm system-get-info`.

```
gladm -d <device> --vendor=u-trust-Anchor-ROOT-CA-1.cer -u admin -k giak.pem
system-get-info
```

By using the `--vendor=` parameter with the Utimaco root certificate in combination with the Global Administrator authentication credentials, the command checks the given certificate against the vendor certificate stored on the device. If the command is executed successfully, the certificate is verified. In case the certificate is not verified, an error is returned and the command is not executed. In this case, contact Utimaco immediately.



The device has been verified successfully as a genuine Utimaco u.trust Anchor device that has not been tampered with.

Verification by the operator can always be done by retrieving the relevant certificates via `gladm system-get-trust-chain` and verifying them with `openssl`:

1. Retrieve the chain of trust. For authentication, use the Global Initial Admin Key that is delivered within the product bundle at `/Software/Linux/Administration/key` (for Windows: `\Software\Windows\Administration\key`).

```
gladm -d <device> -u admin -k giak.pem system-get-trust-chain
```

The Certificate Signing Request is returned along with all available certificates.

```
device auth key certificate signing request written to: device-auth-key.csr
operator device auth key certificate chain not present
vendor device auth key certificate chain written to: dak-vendor-chain.pem
```

```
glad auth key certificate written to: glad-auth-key-device-cert.pem
```



In DAK and GAK certificates, Elliptic Curve (EC) point compression is used. This is an optional feature according to the corresponding standards and is not necessarily supported by all tools.

2. Take the DER-coded root certificate `u-trust-Anchor-ROOT-CA-1.cert` supplied to you within the product bundle at `/Software/Linux/Administration/key` (for Windows: `\Software\Windows\Administration\key`) and convert it to a PEM-formatted file named `Utimaco_root.pem` via openssl.

```
openssl x509 -in u-trust-Anchor-ROOT-CA-1.cer -inform DER -out  
Utimaco_root.pem
```

3. Verify the chain by using the converted Utimaco root certificate and the vendor certificate from the trust chain.

```
cat <path>/u-trust-Anchor-ROOT-CA-1.pem dak-vendor-chain.pem | openssl  
verify
```

The following line should be returned:

```
dak-vendor-chain.pem:OK
```



The verification with openssl can occasionally return some openssl errors (e.g. error 20 at 0 depth lookup: unable to get local issuer certificate). In case this happens, use gladm to verify the certificates.

Before starting regular use, make sure to update the device using the latest FIPS certified system image available, see *Updating the Device Firmware (system-update) in the u.trust Anchor FIPS 140-3 - Administration Manual*.

4.4.4 Changing the Authentication Token of the Global Initial Administrator

When the u.trust Anchor device is started for the first time after delivery or after being cleared, the role of the Global Administrator is that of the Global Initial Administrator ADMIN. After the installation of the u.trust Anchor host software from the product bundle, the authentication keyfile `ADMIN.key` for the Global Initial Administrator ADMIN is stored on the host computer used for device administration. For the credentials of this user, the Global Initial Admin Key (GIAK) is used, which only provides access to a very restricted set of commands.

The Global Initial Admin Key (GIAK) is delivered as part of the u.trust Anchor product bundle that can be downloaded from the Utimaco download portal. The GIAK `giak.pem` can be found within the bundle at `/Software/Linux/Administration/key` (for Windows: `\Software\Windows\Administration\key`).



When the execution of a command/function that requires authentication is requested by the user and the GIAK is provided, then a request to change credentials is returned and message "The user authentication key needs to be updated." is displayed.

Only the following commands are exempt:

- `user-change-credentials`
- `system-get-trust-chain`
- `system-reset-alarm`

To acquire full initial admin rights, the GIAK must be replaced with a Global Admin Authentication Key (GAAK).

The openssl command `genpkey` can be used to generate a GAAK. In this case, the key cannot be stored on a smartcard.

Example for a key generation with openssl:

```
openssl genpkey -algorithm ec -pkeyopt ec_paramgen_curve:secp521r1 -aes-256-cbc
-out admin_gaak_enc.key \
openssl ec -pubout -outform pem -in admin_gaak_enc.key -out admin_gaak_pub.pem
```

To replace the GIAK with the GAAK, perform the command `gladm user-change-credentials`, authenticating it with the GIAK.

```
gladm -d <device> -u admin -k giak.pem user-change-credentials admin
admin_gaak_pub.pem
```



The credentials of the Global Administrator have successfully been changed from the GIAK to the GAAK.

For further details on user authentication in gladm and detailed command descriptions, see the [Global Administration Management \(gladm\) Tool \(p. 120\)](#) section of the [Appendix \(p. 120\)](#).

4.4.5 Importing an Operator Secret



This step is not mandatory, however, it is required if you have to instantiate more than one cHSM. The operator secret must be loaded to allow the creation of snapshots and to restore cHSMs.

An operator secret must be imported to enable the taking and restoring of snapshots. All snapshots created on a device with an active operator secret can be restored on all devices that hold this specific operator secret.

The operator secret is imported into the device with the `gladm key-import-operator-secret` or the `gladm key-set-operator-secret` command. It will be used when restoring a cHSM. The last imported operator secret will be set as active and used for subsequent snapshots on this device. If an inactive copy of the operator secret is already present on the device, the inactive copy must be deleted before reimporting.



Make sure to save the Operator Secret locally, since it is erased at any alarm occurrence and every perform of a Clear or an External Erase of the device. Any snapshot can only be restored when the Operator Secret that was present on the device upon snapshot creation is imported again. Otherwise, the snapshots cannot be used anymore.

For a detailed command description, see the 2020-0035 Global Administration Management (gladm) Tool section of the 2020-0035 Appendix.

In this process, a new wrapping key pair is generated and the private key is stored on the device. Then an X.509 certificate for the public key is created and signed by the Device Authentication Key (DAK). The certificate and the unique token of the private wrapping key used to identify the used wrapping key pair are returned, and can then be used to import a wrapped Operator Secret.

1. Generate a new wrapping key pair via `gladm key-get-wrapping-key`.

```
gladm -d <device> -u admin -k admin_gaak.pem key-get-wrapping-key -c  
wrapkey_cert -t wrapkey_token -- 2048
```



If there are already 32 wrapping keys, the oldest one is deleted before storing the new one.

2. Retrieve the DAK by retrieving the trust chain via `gladm system-get-trust-chain`.

```
gladm -d <device> -u admin -k admin_gaak.pem system-get-trust-chain
```

3. Locate the PEM-format vendor root certificate `Utimaco_root.pem` you created when verifying the device or take the DER-coded root certificate `u-trust-Anchor-ROOT-CA-1.cert` supplied to you within the product bundle at `/Software/Linux/Administration/key` (for Windows: `\Software\Windows\Administration\key`) and convert it to a PEM-formatted file named `Utimaco_root.pem` via openssl.

```
openssl x509 -in u-trust-Anchor-ROOT-CA-1.cert -inform DER -out  
Utimaco_root.pem
```

4. Verify the certificate using the chains and vendor root certificate to ensure that the wrapping key was generated for this purpose on a genuine u.trust Anchor device.

```
openssl verify -CAfile Utimaco_root.pem -untrusted dak-vendor-crt  
wrapkey_cert:
```

The following should be returned:

```
wrapkey_cert: OK
```

Optionally, repeat verification with operator certificate and operator root.

5. Extract the public key from the certificate and wrap the Operator Secret with RSAES-OAEP, SHA-256 label digest and MGF1-SHA-256:

```
openssl pkeyutl -encrypt -certin -inkey wrapkey_cert -pkeyopt  
rsa_padding_mode:oaep -pkeyopt rsa_oaep_md:sha256 -pkeyopt  
rsa_mgf1_md:sha256 -in plain_secret -out wrapped_secret
```

6. Import the wrapped Operator Secret, providing the unique token used to identify the wrapping key (here `token`) and the Operator Secret (here `operator_secret`) via `gladm key-import-operator-secret`.

```
gladm -u admin -k admin_gaak.pem key-import-operator-secret wrapkey_token  
wrapped_secret
```



The wrapped Operator Secret has been unwrapped and stored as the new active Operator Secret.

4.4.6 Importing an Operator Certificate

The final step to claiming the device is to import the operator certificate. The PEM-encoded DAK CSR obtained in *Verifying the Authenticity of the Device* will be signed with a CA key.

It is important that the resulting certificate includes the certificate extensions requested in the CSR. For example, since it is used to sign other certificates on the device such as the GAK or CAK certificates, the DAK certificate should be marked as CA. For this reason, the Basic Constraints and Key Usage extensions shall be set accordingly. The following signature algorithms are supported:

- sha224WithRSAEncryption
- sha256WithRSAEncryption
- sha384WithRSAEncryption
- sha512WithRSAEncryption
- ecdsa-with-SHA224
- ecdsa-with-SHA256
- ecdsa-with-SHA384

- ecdsa-with-SHA512

The following steps guide through the process of importing an operator certificate. While there are multiple ways to do this, the following example uses openssl.

1. Create a file `v3.ext` that contains a section `v3_intermediate_ca` with the following X.509v3 extensions.

```
cat > v3.ext <<EOF
[v3_intermediate_ca]
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer
basicConstraints = critical,CA:true,pathlen:0
keyUsage = critical,keyCertSign
EOF
```

2. Create the self-signed operator root certificate `ca.crt` with the generated key `ca.key`. The subject of the certificate must include the commonName field, `CN=.`

```
openssl req -x509 -nodes -newkey rsa:4096 -keyout ca.key -out ca.crt
-days 365 -subj '/CN=CSAR Test CA/'
```

3. The PEM-encoded CSR for the public part of the DAK is now signed with the generated key `ca.key` and the root certificate `ca.crt`, using the extensions of section `v3_intermediate_ca` contained in the extension file `v3.ext` and saving the certificate as `device-auth-key.crt` in PEM-encoding.

```
openssl x509 -req -in device-auth-key.csr -CA ca.crt -CAkey ca.key \
-extfile v3.ext -extensions v3_intermediate_ca -set_serial 01 -days 30 \
-outform pem -out device-auth-key.crt
```

4. The operator certificate `device-auth-key.crt` can now be loaded onto the device. If Intermediate-CAs are used in the OPERATOR chain, the complete chain must be loaded into the HSM and not just the DAK certificate.

```
gladm -d <device> -u admin -k admin_gaak_enc.key key-import-cert device-
auth-key.crt
```



The device has been claimed successfully. The steps taken guarantee that the device is a genuine Utimaco device that has not been tampered with since it has been manufactured.

At this stage, the versions of different components on the card and its system state can be retrieved via `gladm system-get-info`. The device is ready for cHSM creation, snapshot, and restore operations.

5 Configuration

5.1 u.trust Anchor FIPS

FIPS restrictions can be applied to the u.trust Anchor, their scope depends on the available firmware package.

	Scope	Requirement	gladm system-get-info (p. 151) return	csadm GetState return
FIPS Mode	u.trust Anchor device	FIPS firmware package	Device version number has suffix <code>-c</code>	
	cHSM	SecurityServer FIPS template (p. 47)		FIPS mode = ON
FIPS restrictions applied	cHSM	SecurityServer FIPS template (p. 47)		FIPS restrictions = applied

Table 11: FIPS Overview

To update the device from non-FIPS firmware package to a FIPS approved firmware package, please follow the instructions in section Updating from Version 4.80, 4.90 and 6.0.0 to 6.0.0-c FIPS.

5.1.1 Configuring the FIPS Approved Mode

Global Administrators use the [gladm \(p. 24\)](#) tool to access the Global Administration application glad. The cHSM Administrators use the [csadm \(p. 25\)](#) tool to access their cHSM. The [Global Administration Management \(gladm\) Tool \(p. 120\)](#) section of this manual and the [u.trust Anchor FIPS 140-3- csadm Manual \(p. 119\)](#) describe the gladm and csadm commands in more detail.



If a u.trust Anchor device is in in FIPS certified state, then:

- The Global Administration application glad is always operated in FIPS mode.
- u.trust Anchor cHSMs can be operated in FIPS Approved mode or in non-FIPS Approved mode.

Please note that commands must be performed by appropriately authenticated operators.



If the u.trust Anchor device is in FIPS certified state every service in the Approved mode is an 'Approved service'. Every service in non-Approved mode is a 'non-Approved service' in the sense of [FIPS140-3] (p. 119).



In case the device was running with an image that was non-FIPS and then updated with a FIPS image, a ClearToFactoryDefault must be performed before the device can be used to ensure the exclusiveness of critical security parameters between the different images. For this, perform an external erase, then a `gladm system-clear` command and afterwards a `gladm system-reset-alarm` command authenticated with the initial credentials of the Global Initial Administrator.

Verify, if the device is in FIPS certified state:

1. Check that a valid certificate exists <https://csrc.nist.gov/projects/cryptographic-module-validation-program/validated-modules/search>

Compare the device identifier version as stated in the certificate with the return of the `gladm system-info` command:

```
$ gladm Dev=<device_address> system-info
Device system version 6.0.0.0-c
Sensory Controller software version 3.02.0.8
Hardware revision number 7.03.0.3
UID <device_uuid>
Initial user credentials unchanged
Vendor Secret is present on the device
Vendor DAK Certificate is present on the device
OK
```

The Global Administrator should verify that the following line items are included in the output and that the listed version numbers align with these entries:

```
Device system version 6.0.0.0-c
Sensory Controller software version 3.02.0.8
Hardware revision number 7.03.0.3
```

or

```
Device system version 6.0.0.0-c  
Sensory Controller software version 3.02.0.7  
Hardware revision number 7.03.0.3
```



If the device version number contains the suffix `-c`, then the device has loaded a FIPS firmware image.



Verify that the device system version does not contain `recovery`. If `recovery` is indicated, the module is in an error state. If `recovery` is still indicated after reboot see [Leaving the Recovery Mode \(p. 113\)](#).

To operate the cHSM in the approved mode, perform the following steps:

1. List the existing cHSM templates as follows:

```
gladm Dev=<device_address> system-list-templates  
SecurityServer  
SecurityServer-FIPS
```

Verify that there is

- a template for a FIPS cHSM called `SecurityServer-FIPS`, and there is
 - a template for a non-FIPS cHSM called `SecurityServer`.
2. Global Administrators can create cHSMs that operate in Approved mode (FIPS cHSMs), and cHSMs that operate in non-Approved mode (non-FIPS cHSMs). Log in as a Global Administrator and create cHSMs as described in [Creating a new cHSM \(p. 79\)](#) (`gladm chsm-create` command):
 - To create a cHSM in Approved mode, use the FIPS template `SecurityServer-FIPS`.
 - To create a cHSM in non-Approved mode, use the non-FIPS template `SecurityServer`. The cHSM mode cannot be changed after creation.
 3. Each cHSM's operating mode is shown in the output of the `csadm GetState` command:

```
$ csadm Dev=<device address, cHSM slot number> GetState  
mode = Operational Mode  
state = INITIALIZED (0x00140004)
```

```
FIPS mode = ON
temp = -
alarm = OFF
...
```

Verify that each cHSM is in `Operational Mode`, in `INITIALIZED` state, and the alarm state is `OFF`.

The addressed cHSM is in FIPS approved mode, if the device is in FIPS certified state and the following line is contained in the output:

```
FIPS mode = ON
```




5.1.2 cHSM Built-in Elliptic Curves | FIPS 140-3

The CXI firmware module of your device can use elliptic curves for ECDSA signature creation, EdDSA signature creation and ECDH key agreement. Each curve is specified by elliptic curve domain parameters and is identified by a name.

The following table lists curves actually approved/allowed for FIPS 140-3. The table shows what is blocked in FIPS mode and what is not implemented.

Built-in elliptic curves are listed by their name, denotes the bit size of their domain parameters (i.e. the key length), references the specification where the curve and its domain parameters are defined, and shows what they can be used for.

On high-performance HSM models, numerous common elliptic curves are implemented in hardware.

	approved / allowed
	blocked
	Not implemented

Type	Curve	Size	sign	verify	deriveKey	agreeSecret	ECDSA user authentication	generateKey (pair)	Defined in
Weierstrass Curves	P-224 P-256 P-384 P-521	224	✓	✓	✓ in FIPS mode only ECDH_COF with kdf	-	✓	✓	[FIPS186-4], [ANSI-X9.62], [SEC2]
		256							[FIPS186-4], [ANSI-X9.62], [SEC2]
		384							[FIPS186-4], [ANSI-X9.62], [SEC2]
		521							[FIPS186-4], [ANSI-X9.62], [SEC2]
									[FIPS186-4], [ANSI-X9.62], [SEC2]
	P-192 K-163 B-163	192	-	✓	-	-	-	-	[FIPS186-4], [ANSI-X9.62], [SEC2]
		163							[FIPS186-4], [ANSI-X9.62], [SEC2]
		163							[FIPS186-4], [ANSI-X9.62], [SEC2]
		163							[FIPS186-4], [ANSI-X9.62], [SEC2]
		163							[FIPS186-4], [ANSI-X9.62], [SEC2]

Type	Curve	Size	sign	verify	deriveKey	agreeSecret	ECDSA user authentication	generateKey (pair)	Defined in
Weierstrass Curves									[FIPS186-4], [ANSI-X9.62], [SEC2]
									[FIPS186-4], [ANSI-X9.62], [SEC2]
		2							[FIPS186-4], [ANSI-X9.62], [SEC2]
		3							[FIPS186-4], [ANSI-X9.62], [SEC2]
		3							[FIPS186-4], [ANSI-X9.62], [SEC2]
		2							[FIPS186-4], [ANSI-X9.62], [SEC2]
		8							[FIPS186-4], [ANSI-X9.62], [SEC2]
		3							[FIPS186-4], [ANSI-X9.62], [SEC2]
		4							[FIPS186-4], [ANSI-X9.62], [SEC2]
		0							[FIPS186-4], [ANSI-X9.62], [SEC2]
	K-233	9							[FIPS186-4], [ANSI-X9.62], [SEC2]
	K-283	5							[FIPS186-4], [ANSI-X9.62], [SEC2]
	K-409	7							[FIPS186-4], [ANSI-X9.62], [SEC2]
	K-571	1	⊖	✓	⊖	⊖	⊖	⊖	[FIPS186-4], [ANSI-X9.62], [SEC2]
	B-233	2	⊖		⊖	⊖	⊖	⊖	[FIPS186-4], [ANSI-X9.62], [SEC2]
	B-283	3							[FIPS186-4], [ANSI-X9.62], [SEC2]
	B-409	3							[FIPS186-4], [ANSI-X9.62], [SEC2]
	B-571	2							[FIPS186-4], [ANSI-X9.62], [SEC2]
		8							[FIPS186-4], [ANSI-X9.62], [SEC2]
		3							[FIPS186-4], [ANSI-X9.62], [SEC2]
		4							[FIPS186-4], [ANSI-X9.62], [SEC2]
		0							[FIPS186-4], [ANSI-X9.62], [SEC2]
		9							[FIPS186-4], [ANSI-X9.62], [SEC2]
		5							[FIPS186-4], [ANSI-X9.62], [SEC2]
		7							[FIPS186-4], [ANSI-X9.62], [SEC2]
		1							[FIPS186-4], [ANSI-X9.62], [SEC2]

Type	Curve	Size	sign	verify	deriveKey	agreeSecret	ECDSA user authentication	generateKey (pair)	Defined in
Montgomery Curves	curve25519 curve448	255 448	✗	✗	—	—	✗	—	[RFC 7748] with Errata ID 4730 [RFC 7748] with Errata ID 4730
Twisted Edwards Curves	Edwards25519 Edwards448	255 448	✓ only EdDSA	✓ only EdDSA	✗	✗	✗	✓	[RFC 7748] with Errata ID 4730 [RFC 7748] with Errata ID 4730

Type	Curve	Size	sign	verify	deriveKey	agreeSecret	ECDSA user authentication	generateKey (pair)	Defined in
Brainpool Curves	brainpoolP224r1	224							
	brainpoolP256r1	256							
	brainpoolP320r1	320							
	brainpoolP384r1	384							[BP]
	brainpoolP512r1	512							[BP]
	brainpoolP224t1	224							[BP]
	brainpoolP256t1	256							[BP]
	brainpoolP320t1	320							[BP]
	brainpoolP384t1	384							[BP]
	brainpoolP512t1	512							[BP]
	twisted variants		✓	✓	✓	✗	✓	✓	[BP]
	rainpoolP224t1	224							[BP]
	rainpoolP256t1	256							[BP]
	rainpoolP320t1	320							[BP]
	rainpoolP384t1	384							[BP]
assuming the module verifies the domain parameter validity to ensure the correct curves are used (i.e., not its quadratic twist)*									
Curve secp256k1 (Koblitz, security strength of 128 bits)	secp256k1	256	✓ only for use with block chain applications	✓ only for use with block chain applications	✗	✗	✗	✓ only for use with block chain applications	[SEC2]

Type	Curve	Size	sign	verify	deriveKey	agreeSecret	ECDSA user authentication	generateKey (pair)	Defined in
French Curve	FRP256 v1	256	—	—	—	—	—	—	[ANSSI]

Table 12: List of built-in elliptic curves

5.2 Creating a new cHSM

Prerequisites

Have the public part of the Container Administration Authentication Key (CAAK) at hand that is supplied by the cHSM TENANT who will use the cHSM.

If CUSTOMERs do not yet have a key, they must generate a key pair and save the public part in a place where it can be accessed.

The most secure and therefore recommended way to generate a private key is to use `csadm GenKey` to generate an RSA key pair (private and public key) on a smartcard and then save the public key with `csadm SaveKey`. For details see, *GenKey* and *SaveKey* in the *u.trust Anchor csadm Manual*.

```
csadm GenKey=:cs2:cjo:USB0,"cHSM User"
Generates an RSA key pair on the smartcard.

csadm PubKey=:cs2:cjo:USB0 SaveKey=C:\keys\pubkey1.key
Extracts the RSA public key from a smartcard and stores it in a file.
```

Procedure

1. Execute the `gladm chsm-create` command. The only required parameters are the admin key file and the slot ID within the u.trust Anchor device. If no other parameters are given, the new cHSM will be created using the default template. The resulting certificates will be stored in the working directory.

Example to create a new cHSM in slot 15 with the public key part `myADMIN.pub` used as initialization data:

```
gladm -d <device> -u admin -k admin_gaak.pem chsm-create myADMIN.pub 15
Creating cHSM...
```

```
slot: 15  
init_data: myADMIN.pub
```

Depending on the device type you use, the specified cHSM slot might not exist (in the example: slot 15). In this case, the following error message is shown:

```
Operation failed: CHAI_SLOT_INVALID: The requested cHSM slot is invalid or  
unavailable
```

In this case, perform a `gladm chsm-list-slots` command to show the available cHSM slots and repeat the `gladm chsm-create` command using one of the available cHSM slots.

2. When the command is performed, u.trust Anchor creates a Container Authentication Key (CAK), which is then signed with the Device Authentication Key (DAK), allowing the cHSM to prove it runs on genuine hardware.

While executing this action, u.trust Anchor writes the following files:

- the cHSM creation receipt, containing the Container Administration Authentication Key (CAAK) used to set up the cHSM along with a timestamp and the cHSM's UUID;
- a CSR for the Container Authentication Key (CAK) to be signed by the customer;
- the certificates for the vendor signed certificate of the Device Authentication Key (DAK);
- the DAK-signed certificate of the CAK and the GlAD Authentication Key (GAK), and
- the operator-signed certificate of the DAK.

Output pattern

```
cHSM creation receipt... .txt  
cHSM receipt signature...  
cSHM auth key certificate... .csr  
device cHSM auth key certificate written to... .pem  
glad auth key cert... .pem  
vendor device auth key... written to... .pem  
operator device... written too... .pem
```

3. Provide the u.trust Anchor cHSM Administrator with these files so that they are able to claim the u.trust Anchor cHSM.



The u.trust Anchor cHSM has been successfully created and can be verified and claimed by the u.trust Anchor cHSM Administrator.

5.3 Generating an Operator Secret

Operator secrets are used alongside a vendor secret to derive keys used for the encryption of snapshots. This approach prevents either party from decrypting any of the created snapshots.

A snapshot contains the data of a cHSM stored on disk at the time of taking the snapshot. It contains the user data created after the cHSM was created. Snapshots can be taken of running, halted, or locked cHSMs. When taking a snapshot of a running cHSM, the cHSM is temporarily halted and remains unavailable until the snapshot operation has been completed. Taking snapshots of cHSMs in cluster mode is not supported.

An operator secret consists of 32 bytes and is generated and stored securely by the operator. u.trust Anchor additionally offers the possibility to import wrapped operator secrets.

An operator secret must be loaded onto the device to create multiple cHSMs, and at least one active operator secret is necessary to create snapshots. The device can store multiple operator secrets, of which at most one can be active. The active secret is used when new snapshots are created. The other secrets are used to import snapshots that were created by using those secrets.



Global Administrators should use a dedicated smartcard for operator secrets (and not use the same smartcard for MBKs, for example).

Procedure:

1. Connect the smartcard reader to the device and insert the smartcard.
2. Generate an operator secret via `gladm key-set-operator-secret`. In this example -g generates the operator secret, splits it into 3 shares out of which 2 are needed to recover the secret. The share is saved on position 7 on the smartcard.

```
gladm -u admin -k admin_gaak.pem key-set-operator-secret -g 2,3 -r 7 :cs2:auto:USB0
```



The command cannot be executed if an operator secret is already present on the smartcard. Use `[-f]` to overwrite the existing record on the smartcard.

```
gladm -u admin -k admin_gaak.pem key-set-operator-secret -g 2,3 -f -r 7 :cs2:auto:USB0
```

3. The command is executed once and a message on the PIN pad will be displayed, telling you when to press OK and when to insert the next card.
4. The shares are saved on the smartcards. A wrapping key is obtained from the device, and the operator secret is wrapped and imported into the u.trust Anchor.
5. Check the operator secret list via `gladm key-list-operator-secrets`.

```
gladm -u admin -k admin_gaak.pem key-list-operator-secrets
```

6. Upon successful execution of the command, gladm returns a list of the available operator secrets, along with the file date. The new operator secret will become the active operator secret. The previous operator secret will remain available for loading user backups and snapshots.

```
646d4110: 2021-06-24T12:26:21+0000 - active
fecf521a: 2021-06-24T12:26:16+0000
da1bcbb3: 2021-06-24T12:26:12+0000
```

5.4 Copying an Operator Secret

This command allows a Global Administrator to copy an operator secret from smartcard to smartcard, or to transfer an operator secret from a key file to a smartcard. Therefore, the source of the secret may be a key file or a smartcard specifier, whereas the target must be a smartcard specifier.

Furthermore, the Global Administrator may specify the exact record number on the smartcard from and into which a key share shall be read or written, as well as the number of key shares.

If smartcards are used: A PIN pad including a smartcard reader can be connected to the computer where gladm is running (USB port) or to another computer. Watch the display of the PIN pad for instructions on further command processing.

If key files are used: We strongly recommend using encrypted key files. If encrypted key files are used, we strongly recommend using a hidden password entry.



Global Administrators should use a dedicated smartcard for operator secrets (and not use the same smartcard for MBKs, for example).

Procedure for copying from a smartcard to a smartcard:

1. Connect the smartcard reader to the device and insert the smartcard.
2. Perform the command `gladm smartcard-copy-secret`. The following example copies an operator secret from smartcard to smartcard, reading and writing from and into the default record number 7.

```
gladm smartcard-copy-secret :cs2:cjo:USB0 :cs2:cjo:USB0
```

3. The command is executed and a message on the smartcard reader will be displayed, prompting the user when to press **OK** and when to insert the next smartcard.
4. The copied operator secret share is stored in the target smartcard using the default record number 7.



Upon successful copying of the operator secret, a message that the operation has been successfully completed is displayed.

Procedure for copying from a key file to a smartcard:

1. Connect the smartcard reader to the device and insert a smartcard.
2. Perform the command `gladm smartcard-copy-secret`. The following example copies an operator secret, creates 2 out of 2 shares from a file and puts them on smartcards in record number 6.

```
gladm smartcard-copy-secret -t 6 -k 2 secret_key_file :cs2:cjo:USB0
```

3. The command is executed and a message on the smartcard reader will be displayed, prompting the user when to press **OK** and when to insert the next smartcard.
4. The copied operator secret shares are stored in the target smartcards using record number 6.



Upon successful copying of the operator secret, a message that the operation has been successfully completed is displayed.



An operator secret share stored on a smartcard cannot be deleted, it can only be overwritten. If another share is already stored in the specified record, it is overwritten by the share to be generated by using the override flag option. See the *Global Administration Management (gladm) Tool* section of the *Appendix* for details.



For more information on smartcard specifiers, please see the [Generating an Operator Secret \(p. 81\)](#) chapter in the *Global Administration Management (gladm) Tool* section of the *Appendix*.

5.5 Changing Quorum Requirements for Dual Control



This change is only needed, if you plan to establish dual control.

First extract the current quorum requirements with via gladm:


```
gladm -d 10.17.72.24 system-get-quorum-requirements > quorum.txt
```

The last part "`> quorum.txt`" is used to write the output directly into a `quorum.txt` file.

Example lines from the `quorum.txt` (below only part of the list is provided as example):

```
[quorum requirements]
user_add = 1
user_delete = 1
user_list = 0
user_restore_backup = 1
user_create_backup = 1
...
```

The number at the end represent the number of user with this right needed to execute the command.

A user can only have a maximum of 1 for a command.

Commands can be set inside the `quorum.txt` to 2 for dual control.



Please be aware, that you can lock yourself out of commands, if you make commands unavailable for you.

Example for for setting the new quorum requirements:

```
gladm -d 10.17.72.24 -u admin -k C:\keys\se40k.key system-set-quorum-
requirements quorum.txt
Setting quorum requirements with parameters:
quorum configuration: quorum.txt
```

5.6 cHSM Claiming

5.6.1 Creating a Customer CA

Creating a CUSTOMER CA is optional, but it allows you to load your own certificate in the cHSMs and authenticate them as your own cHSMs. You can always authenticate the HSM against the Utimaco Root CA and – if established – against the OPERATOR CA.

The example steps below will provide you with a dummy PKI using openssl for demonstration.

1. Create the CA and its parameters:

```
cat > CustomerCA.ext <<EOF

authorityKeyIdentifier = keyid, issuer

extendedKeyUsage = critical, serverAuth

basicConstraints = critical, CA:FALSE

keyUsage = critical, digitalSignature

EOF
```

2. Create the key set for the customer CA:

```
openssl req -x509 -nodes -newkey rsa:4096 -keyout CustomerCA.key -out
CustomerCA.crt -days 365 -subj '/CN=CUSTOMER Test CA/'

Generating a RSA private key
.....++++
.....
.....
.....++++
writing new private key to 'CustomerCA.key'
-----
```

5.6.2 Claiming the cHSM

Use the CAK certificate signing request obtained after the instantiation of the cHSM (from the OPERATOR) and sign it with the CUSTOMER CA's private key to get the CAK certificate. After that, load it into the cHSM to claim it:

```
openssl x509 -req -in chsm-auth-key_001_20220117082933.csr -CAkey
CustomerCA.key -CA CustomerCA.crt -CAform pem -CAcreateserial -out
CustomerCAK.crt -outform der -extfile CustomerCA.ext
Signature ok
subject=0 = Utimaco IS GmbH, OU = u.trust Anchor, C = DE, serialNumber =
b9947191-8877-40f0-8f4f-26f2910eb983
```

Getting CA Private Key

Load the certificate into the cHSM:

```
csadm dev=... LogonSign=ADMIN, CustomerIAK.key LoadCertificate=CustomerCAK.crt
```

5.6.2.1 Signature Verification

Verify the CAK signed by the DAK signed by the OPERATOR CA:

```
openssl verify -CAfile OperatorCA.crt -untrusted dak-operator-  
chain_001_20220117082933.pem chsm-auth-key-device-cert_001_20220117082933.pem  
  
chsm-auth-key-device-cert_001_20220117082933.pem: OK
```

Verify the CAK signed by the DAK signed by the VENDOR (Utimaco) CA:

```
openssl verify -CAfile vendor_root.pem -untrusted dak-vendor-chain.pem dak-  
vendor-chain_001_20220117082933.pem  
  
dak-vendor-chain_001_20220117082933.pem: OK
```

Verify the CAK signed by the CUSTOMER CA:

```
openssl verify -CAfile CustomerCA.crt CustomerCAK.pem  
  
CustomerCAK.pem: OK
```

If needed, convert the CAK certificate from pem to der format with the following openssl command:

```
openssl x509 -inform der -in CustomerCAK.crt -out CustomerCAK.pem
```

6 Monitoring and Maintenance

6.1 Updating u.trust Anchor

The update of a u.trust Anchor device is done through `rauc` bundles together with a manifest inside a `squashfs` file. The `rauc` bundle is signed using a Cryptographic Message Syntax (CMS/PKCS#7) signature. Updates are A/B slotted, which means there is a standby slot that is updated while the other one is booted. To perform a full update, the procedure has to be repeated as described below.



If a FIPS-certified image version is running on the device and the user plans to update to an image version that is not FIPS-certified, a [clear to FACTORY DEFAULT \(p. 97\)](#) must be performed to ensure the exclusiveness of critical security parameters between the different images.

See also [updating from version 4.80, 4.90 and 6.0.0 to 6.0.0-c FIPS \(p. 92\)](#) for more details.

In order to update the device, follow these steps:

1. Back-up existing CHSMs via `gladm chsm-retrieve` ([p. 147](#)).

```
gladm -u <username> -k <credentials> -d <addr> chsm-retrieve [-f <val>]  
<slot id>
```

2. Install the latest system image using your current version of `gladm` via the `gladm system-update` ([p. 172](#)) command. Adjust the path and filename as necessary. The `--force` flag allows an update to be performed that may potentially overwrite existing data. For a new device, this command can safely be used.

```
gladm -d <device> -u admin -k admin_gaak.pem system-update --force  
Firmware/u.trust_Anchor/u.trust_Anchor-4.50.0.0.raucb
```

3. Reboot the device by restarting the host, this action must be run as root:

```
gladm -d <device> device-restart
```



The command `gladm device-restart` (p. 169) can only be executed locally or via SSH.

4. Verify that the update was successful by querying the device version via `gladm system-get-info` (p. 151).

```
gladm -d <device> system-get-info
```

5. Perform the update again after the successful verification to have both A/B slots updated.

```
gladm -d <device> -u admin -k admin_gaak.pem system-update --force Firmware/  
u.trust_Anchor/u.trust_Anchor-4.50.0.0.raucb
```

6. Restore cHSM backups via `gladm chsm-restore` (p. 147).

```
gladm -u <username> -k <credentials> -d <addr> chsm-retrieve [-f <val>]  
<slot id>
```



`gladm` should not be fully replaced before it has been verified that the device update was successful. A new version of `gladm` and other related non-device software might be needed in order to use new features.



Before a device is updated from version 4.70 or older to version 4.80 or higher, it is recommended to backup all existing databases.

Version 4.70 introduced a new database record structure with the use of unpredictable IVs.

If the device has been updated to version 4.80 or higher, all databases have been migrated automatically to the new structure. Once the database has been migrated, there is no downgrade of the database possible.

To return to a previous version the device needs to be erased after a firmware downgrade and the databases need to be restored from existing backups.

6.1.1 From Version 4.47.2 FIPS to 6.0.0-c FIPS



After updating to 6.0.0-c FIPS, a downgrade to 4.47.2 FIPS cannot be done.



Only key backups made with firmware version 4.47.2 FIPS can be restored directly with 6.0.0-c FIPS.

User and other database backups made with firmware version 4.47.2 FIPS cannot be restored directly with 6.0.0-c FIPS.

To be able to restore all backups made with firmware version 4.47.2 FIPS, the database backups need to be restored with a [non-FIPS](#) cHSM first.

Steps for a direct update to 6.0.0-c FIPS with limited database restore (only key backups)

1. Update the device with the 4.47.3 firmware version, see [Updating u.trust Anchor \(p. 88\)](#).
2. Update the device with the 6.0.0-c firmware version, see [Updating u.trust Anchor \(p. 88\)](#).
3. [Reboot the system \(p. 170\)](#).
4. Verify the firmware version via `gladm system-get-info` ([p. 151](#)), the the Device system version should include a `-c` suffix.

```
Device system version 6.0.0.0-c
```

Steps for a update to 6.0.0-c FIPS with full database restore

1. Access the FIPS cHSM and backup the database via `csadm BackupDatabase`.

Example: Backup the cryptographic key database

```
csadm LogonSign=ADMIN,:cs2:cjo:USB0 BackupDatabase=CXIKEY.db
```

2. Perform the `csadm MBKListKeys` command to determine which Master Backup Key (MBK) is currently in use in MBK slot 3.

```
csadm [Dev=<device>] <Authentication> Key=<keyspec> MBKImportKey=<slot_no>
```

3. Copy the MBK via `csadm MBKCopyKey`.

```
csadm Key=<keyspec_source> MBKCopyKey=<keyspec_target>
```

4. Update the device with the 4.47.3 firmware version, see [Updating u.trust Anchor \(p. 88\)](#).
5. Update the device with the 6.0.0-c firmware version, see [Updating u.trust Anchor \(p. 88\)](#).
6. [Reboot the system \(p. 170\)](#).
7. Verify the firmware version via `gladm system-get-info` ([p. 151](#)), the the Device system version should include a `-c` suffix.

```
Device system version 6.0.0.0-c
```

8. [Create a new cHSM \(p. 79\)](#) with the [non-FIPS SecurityServer template \(p. 47\)](#).
9. Verify and claim the cHSM, see *u.trust Anchor - Containerized Hardware Security Module (cHSM) - Administration Manual*.
10. Access the [non-FIPS](#) cHSM and import the previously saved MBK via `csadm MBKImportKey`.

```
csadm [Dev=<device>] <Authentication> Key=<keyspec> MBKImportKey=<slot_no>
```

11. Restore the database via `csadm RestoreDatabase`.

Example: Restore the cryptographic key database

```
csadm LogonSign=ADMIN,:cs2:cjo:USB0 RestoreDatabase=CXIKEY.db
```

12. Back-up the database again via `csadm BackupDatabase`.

Example: Backup the cryptographic key database

```
csadm LogonSign=ADMIN,:cs2:cjo:USB0 BackupDatabase=CXIKEY.db
```

13. [Create a new cHSM \(p. 79\)](#) with the [FIPS SecurityServer template \(p. 47\)](#).

14. Verify and claim the cHSM, see *u.trust Anchor - Containerized Hardware Security Module (cHSM) - Administration Manual*.
15. Access the FIPS cHSM and import the previously saved MBK via `csadm MBKImportKey`.

```
csadm [Dev=<device>] <Authentication> Key=<keyspec> MBKImportKey=<slot_no>
```

16. Restore the database via `csadm RestoreDatabase`.

Example: Restore the cryptographic key database

```
csadm LogonSign=ADMIN,:cs2:cjo:USB0 RestoreDatabase=CXIKEY.db
```

6.1.2 From Version 4.80 and 4.90 to 6.0.0

Steps

1. Update the device with the 6.0.0 firmware version, see Updating u.trust Anchor.
2. Verify the firmware version via `gladm system-get-info` (p. 151).

```
Device system version 6.0.0.0
```

6.1.3 From Version 4.80, 4.90 and 6.0.0 to 6.0.0-c FIPS



To switch from a non-FIPS firmware version to a FIPS firmware version a reset to **FACTORY DEFAULT** (p. 97) is necessary. This procedure will delete all cHSMs. It is therefore recommended to back up all cHSMs before updating.

Steps

1. Update the device with the 6.0.0-c firmware version, see [Updating u.trust Anchor \(p. 88\)](#).
2. Execute a [Short External Erase \(p. 100\)](#) on the device.

3. Clear the device system via `gladm system-clear`, see [Clearing the System to FACTORY DEFAULT \(p. 97\)](#).

Example: system-clear performed after an external erase
`gladm -d /dev/cs2.0 system-clear`

4. Reset the device alarm via `gladm system-reset-alarm`, see [Resetting the Alarm \(system-reset-alarm\) \(p. 168\)](#).

Example: system-alarm-reset performed after an external erase and a system-clear
`gladm -d /dev/cs2.0 -u admin -k /tmp/giak.pem system-reset-alarm`

5. The device is cleared into delivery state and the default administrator ADMIN is restored with his initial user authentication key (`ADMIN.key`). Set the device up again, see [Setting up the device \(p. 49\)](#).
6. [Reboot the system \(p. 170\)](#).
7. Verify the firmware version via `gladm system-get-info` ([p. 151](#)), the the Device system version should include a `-c` suffix.

Device system version `6.0.0.0-c`

6.2 Downgrading u.trust Anchor

To perform a downgrade of a u.trust Anchor device to a previous version, follow the procedure described in section [Updating u.trust Anchor \(p. 88\)](#). Some restrictions may apply based on the currently used firmware version.

6.2.1 From Version 4.60 or Higher



A u.trust Anchor with version 4.60 or higher cannot be downgraded below version 4.60.

6.2.2 From Version 4.80 or Higher



Version 4.80 introduced a new database record structure with the use of unpredictable IVs.

If the device has been updated to version 4.80 or higher, all databases have been migrated automatically to the new structure. Once the database has been migrated, there is no downgrade of the database possible.

To return to a previous version the device needs to be erased after a firmware downgrade and the databases need to be restored from existing backups.

To downgrade a u. trust Anchor from version 4.80 or higher to an older version proceed as follows:

Precondition

- Database backups made with u.trust Anchor version 4.70 or older

Steps

1. Update the device with the desired firmware version, see [Updating u.trust Anchor \(p. 88\)](#).
2. Execute a [Short External Erase \(p. 100\)](#) on the device.
3. Clear the device system via `gladm system-clear`, see [Clearing the System to FACTORY DEFAULT \(p. 97\)](#).
4. Reset the device alarm via `gladm system-reset-alarm`, see [Resetting the Alarm \(system-reset-alarm\) \(p. 168\)](#).
5. The device is cleared into delivery state and the default administrator ADMIN is restored with his initial user authentication key (`ADMIN.key`). Set the device up again, see [Setting up the device \(p. 49\)](#).
6. Restore the databases via `csadm RestoreDatabase` and `csadm RestoreUser`, [u.trust Anchor - csadm Manual \(p. 119\)](#).

6.2.3 From Version 6.0.0-c FIPS to 6.0.0



To switch from a FIPS firmware version to a non-FIPS firmware version a reset to **FACTORY DEFAULT** (p. 97) is necessary. This procedure will delete all cHSMs. It is therefore recommended to back up all cHSMs before updating.

1. Execute a Short External Erase on the device.
2. Clear the device system via `gladm system-clear`, see Clearing the System to FACTORY DEFAULT.

Example: system-clear performed after an external erase
`gladm -d /dev/cs2.0 system-clear`

3. Reset the device alarm via `gladm system-reset-alarm`, see Resetting the Alarm (system-reset-alarm).

Example: system-alarm-reset performed after an external erase and a system-clear
`gladm -d /dev/cs2.0 -u admin -k /tmp/giak.pem system-reset-alarm`

4. The device is cleared into delivery state and the default administrator ADMIN is restored with his initial user authentication key (`ADMIN.key`). Set the device up again, see Setting up the device.
5. Update the device with the 6.0.0 firmware version, see Updating u.trust Anchor.
6. Verify the firmware version via `gladm system-get-info` (p. 151).

Device system version `6.0.0.0`

6.3 Erase and Clearing Procedures

6.3.1 Clearing Procedures

6.3.1.1 Clearing the Device

The u.trust Anchor device can be cleared via `gladm system-clear`, see [Clearing the System \(system-clear\) \(p. 170\)](#) . It must be authenticated by the Global Administrator upon execution.

The following data will be cleared:

	<i>Clearing the Device</i>
GAAK	preserved
Device Audit Log	preserved
Device Boot Log	preserved
DMK cHSM cHSM Audit Log cHSM Boot Log	cleared
SDMK DAK Vendor Secret	preserved
Secure RAM	unused data is zeroized, other data is preserved

Table 13: Cleared Data Overview




Before executing the `gladm system-clear` command, it is recommended to back up the existing cHSMs using `gladm chsm-snapshot` .

Perform the following steps:

1. Clear the system.
2. Retrieve the device system information via `gladm system-get-info` , the output put should read like example below:

```
Device system version 4.70.0.0
Sensory Controller software version 3.02.0.8
Hardware revision number 7.03.0.3
UID ce00001d44b36701
```

```
SN CS850654
Device Type u.trust Anchor Se100
Vendor Secret is present on the device
Vendor DAK Certificate is present on the device
```

 Compare deviating outputs with the overview in section [How to Identify different Device States](#).


6.3.1.2 Clearing to FACTORY DEFAULT State

The u.trust Anchor device can be set it back to the FACTORY DEFAULT state via `gladm system-clear`, see [Clearing the System \(system-clear\) \(p. 170\)](#). It must be preceded by a [Short External Erase \(p. 100\)](#).

The following data will be cleared:

	<i>Clearing to Factory Default State after Performing an External Erase</i>
GAAK	restores Global Initial Administrator with GIAK
Device Audit Log	preserved
Device Boot Log	cleared, if preceded by an external erase
DMK cHSM cHSM Audit Log cHSM Boot Log	cleared
SDMK DAK Vendor Secret	preserved
Secure RAM	cleared, if preceded by an external erase

Table 14: Reset to FACTORY DEFAULT - Cleared Data Overview

 Before executing the `gladm system-clear` command, it is recommended to back up the existing cHSMs using `gladm chsm-snapshot`.

Perform the following steps:

1. Perform a [Short External Erase \(p. 100\)](#), after that the device should restart with the operational image.
2. [Clear the system \(p. 170\)](#) without authentication.

3. Perform an [alarm reset \(p. 168\)](#) using the GIAK.
4. Retrieve the device system information via `gladm system-get-info`, the output put should read:

```
Initial user credentials unchanged
Vendor Secret is present on the device
Vendor DAK Certificate is present on the device
```



Compare deviating outputs with the overview in section [How to Identify different Device States \(p. 111\)](#).

5. The device can be set up again, using the GIAK, see Claiming the Device.

6.3.2 External Erase

The u.trust Anchor can be cleared via an pushing the **Erase** button outside the device .

The following data will be cleared:

	External Erase
GAAK	preserved
Device Audit Log	preserved
Device Boot Log	cleared
DMK cHSM cHSM Audit Log cHSM Boot Log	cleared
SDMK DAK Vendor Secret	preserved
Secure RAM	cleared

Table 15: Data cleared by External Erase

Location of the Erase button

- u.trust Anchor LAN V5



Figure 17 : u.trust Anchor LAN V5 - front panel

- u.trust Anchor PCIe card

(1) LED flash light

(2) Erase button



Figure 18 : u.trust Anchor PCIe card - slot plate

Types of External Erase

	Short External Erase	Long External Erase
<i>Erased button pushed for</i>	<2 seconds	≥ 10 seconds
<i>Device reboots with</i>	operational image	recovery image
<i>Data cleared</i>	no difference	

Table 16: External Erase Comparison

For procedure details, see sections:

- [Short External Erase \(p. 100\)](#)
- [Long External Erase \(p. 101\)](#)



Pushing the erase button/rebooting the device in short sequence will trigger the recovery mode!
Always wait for the reboot to finish!
A complete reboot takes up to 50 seconds.

6.3.2.1 Short External Erase

Perform the following steps to execute a short external erase and reset the device back to FACTORY DEFAULT:



Before executing an external erase, it is recommended to back up the existing cHSMs using `gladm chsm-snapshot`.

1. Push the **Erase** button for 2 seconds by using an appropriate screwdriver.



Figure 19 : u.trust Anchor LAN V5 - front panel



Figure 20 : u.trust Anchor PCIe card - slot plate

PCIe card only: The LED flash light (1) flashes up red to confirm the activation of the **Erase**-button.

2. The u.trust Anchor will reboot with the operational image.
3. [Clear the system \(p. 170\)](#) via `gladm-system-clear` without authentication.
4. Perform an [alarm reset \(p. 168\)](#) via `gladm-reset-alarm`, using the GIAK.
5. Retrieve the device system information via `gladm system-get-info`. The output put should read:

```
Initial user credentials unchanged
Vendor Secret is present on the device
Vendor DAK Certificate is present on the device
```



Compare deviating outputs with the overview in section [How to Identify different Device States \(p. 111\)](#).

6. The device can be set up again using the GIAK, see [Claiming the Device \(p. 56\)](#).



Pushing the erase button/rebooting the device in short sequence will trigger the recovery mode!
Always wait for the reboot to finish!
A complete reboot takes up to 50 seconds.

6.3.2.2 Long External Erase

Perform the following steps to execute a long external erase, enter the recovery mode and reset the device back to FACTORY DEFAULT:



Before executing the an external erase, it is recommended to back up the existing cHSMs using `gladm chsm-snapshot`.

1. Push the **Erase**-button for 10 seconds by using an appropriate screwdriver.



Figure 21 : u.trust Anchor LAN V5 - front panel




Figure 22 : u.trust Anchor PCIe card - slot plate

PCIe card only: The LED flash light (1) flashes up red to confirm the activation of the **Erase**-button.

2. The u.trust Anchor will reboot with the recovery image.
3. Execute the command `gladm system-get-info`.
4. Check the output of the returned parameter `version`. If it starts with `recovery_`, then the device is in recovery mode.
5. To leave the recovery mode perform a short external erase by pushing the **Erase**-button for 2 seconds.
6. The u.trust Anchor will reboot with the operational image.
7. **Clear the system** (p. 170) via `gladm-system-clear` without authentication.
8. Perform an **alarm reset** (p. 168) via `gladm-reset-alarm`, using the GIAK.
9. Retrieve the device system information via `gladm system-get-info`. The output put should read:

```
Initial user credentials unchanged
```

Vendor Secret is present on the device
Vendor DAK Certificate is present on the device

 Compare deviating outputs with the overview in section [How to Identify different Device States](#) (p. 111).

10. The device can be set up again, using the GIAK, see [Claiming the Device](#) (p. 56).



Pushing the erase button/rebooting the device in short sequence will trigger the recovery mode!
Always wait for the reboot to finish!
A complete reboot takes up to 50 seconds.

6.4 Audit Log

The system audit log contains events concerning the operator interface and the hardware of the u.trust Anchor device. This includes command execution through gladm and pre-alarms. It does not cover events that occur within any of the cHSMs that might be running on the u.trust Anchor device.

The audit log can be retrieved via the `gladm system-get-audit-log` command, which saves the current audit log. Each entry in the audit log is linked to the previous through a hash. When the `gladm system-get-audit-log` command is performed, the last hash is returned. The resulting hash chain ensures the integrity of the system audit log.

Optionally, the `gladm system-get-audit-log` command can return a trimmed audit log. When trimmed, the first entry of the trimmed audit log continues the hash chain and protocols the trimming operation itself.



Once the system audit log is full, all commands that would otherwise be written to the audit log will fail with ``CHAL_SYSTEM_AUDIT_LOG_FULL``. Commands that fail for this reason are **not** logged.

The audit log must be trimmed before normal operation can resume. Therefore, it is advised to trim the audit log regularly. The fill status of the audit log can be retrieved via `gladm system-get-metrics`.

6.4.1 Audit Log Events

The following events are written to the audit log:

<i>Audit Log Event</i>	<i>Entry Syntax</i>
Successful and failed attempts to execute commands Some exceptions apply	<code>type="command"</code>
Authentication failures	<code>command="session_open"</code>
Temperature pre-alarms or alarms	<code>type="pre-alarm" or "alarm"</code>

Table 17: Audit Log Events

A snippet from the audit log may look like this. There is one entry per line:

```
{ "count": 31132, "timestamp": "2021-07-05T12:15:16+0000", "type": "command",
  "command": "chsm_halt", "users": ["admin"], "result": "CHAI_OK", "hash":
  "f77fb763caba299a7bf8a2bef7f7c863d268a0347ed31a31e7ff9af6d11f3438" }
{ "count": 31133, "timestamp": "2021-07-05T12:15:17+0000", "type": "command",
  "command": "chsm_free_slot", "users": ["admin"], "result": "CHAI_OK", "hash":
  "abb21ddfc81a5ed6661ceeca9856bf5db1da80e7a29de82e803c42feb9d72388" }
```

The following fields can be contained in the audit log entries. Some fields are restricted to certain entry types:

<i>Field</i>	<i>Description</i>	<i>Type</i>
count	Entry counter Preserved between trimming operations	All
timestamp	Timestamp at which the entry was written It adheres to the format <code>yyyy-mm-ddThh:mm:ss+0000</code> , whereas <code>T</code> is a literal	All
type	Type of the logged event	All
hash	Hash over the previous entry's hash value and the fields of the current entry	All
command	Name of the command executed or the attempt thereof	command
users	List of users within the same session that executed a command, can be empty	command

<i>Field</i>	<i>Description</i>	<i>Type</i>
result	Contains <code>CHAI_OK</code> if successful or the respective error name otherwise.	command
alarms	List of alarms	pre-alarm or alarm

Table 18: Regular Audit Log Fields

The following commands are only written to the audit log if they were **not** successful:

- `chsm_list_slots`
- `system_get_info`
- `system_monitor`
- `system_get_metrics`
- `system_get_time`
- `system_list_templates`
- `slot_get_quota`
- `system_get_quota`
- `user_list`
- `key_list_operator_secrets`
- `system_get_quorum_requirements`
- `system_get_trust_chain`

If gladm issues multiple commands to the device as part of a single call, these are logged individually.



Only commands received by the device are logged. This means that commands that do not interact with the device or fail due to connection issues are not logged. Commands that fail because the audit log has reached its maximum size (`CHAI_SYSTEM_AUDIT_LOG_FULL`) are exempt from the audit log.

6.4.2 Audit Log Hash Chain

Each entry in the audit log is linked to the previous through a hash. When the `gladm system-get-audit-log` command is performed, the last hash is returned. The resulting hash chain ensures the integrity of the system audit log.

The hash field of an entry is a SHA-256 hash over a concatenation of the hash field of the previous entry and the fields current entry. The quotation marks of the previous hash are not included. Neither are the outer curly brackets and their trailing/leading spaces, nor the hash field of the current entry, including the preceding comma and space.

Example Audit Log Entry:

```
{ "count": 31133, "timestamp": "2021-07-05T12:15:17+0000", "type": "command",
  "command": "chsm_free_slot", "users": ["admin"], "result": "CHAI_OK", "hash":
  "abb21ddfc81a5ed6661ceeca9856bf5db1da80e7a29de82e803c42feb9d72388" }
```

Hash Key:

```
f77fb763caba299a7bf8a2bef7f7c863d268a0347ed31a31e7ff9af6d11f3438"count": 31133,
  "timestamp": "2021-07-05T12:15:17+0000", "type": "command", "command":
  "chsm_free_slot", "users": ["admin"], "result": "CHAI_OK"
```

The following example uses the interactive Python shell to calculate the hash from the key above:

```
>>> import hashlib
>>> key =
'f77fb763caba299a7bf8a2bef7f7c863d268a0347ed31a31e7ff9af6d11f3438"count":
31133, "timestamp": "2021-07-05T12:15:17+0000", "type": "command", "command":
"chsm_free_slot", "users": ["admin"], "result": "CHAI_OK"'
>>> hashlib.sha256(key.encode()).hexdigest()
abb21ddfc81a5ed6661ceeca9856bf5db1da80e7a29de82e803c42feb9d72388
```

6.5 Temperature-dependent Behavior of u.trust Anchor

The u.trust Anchor device is fully operational only if its internal temperature does not exceed or fall below a well-defined operational temperature range. The next table shows how the u.trust Anchor's behavior changes depending on its internal temperature.



The operating temperature for u.trust Anchor ranges between -10 °C to 60 °C (50 °F to 140 °F).

Internal Temperature	Behavior of u.trust Anchor
Below -17 °C (1.4 °F)	Alarm is triggered, all sensitive data in the u.trust Anchor is deleted, and the device is restarted.
-17 °C to 77 °C (1.4 °F to 170.6 °F)	Normal operation
Above 77 °C (170.6 °F)	Alarm is triggered, all sensitive data in the u.trust Anchor is deleted, and the device is restarted.

Table 19: Operational temperature of u.trust Anchor

All temperature values in the table are approximate values. The exact temperature values may vary a little because of tolerances of the electronic components and the use of a hysteresis by the comparators.



Resetting the u.trust Anchor has no effect if its internal temperature still exceeds or is below the operational temperature range. In case of u.trust Anchor's constant high temperature, we recommend switching off the power supply for some time to cool down the u.trust Anchor.

Note that only the internal temperature of the u.trust Anchor device is relevant, not the environmental temperature. The actual value of the inner temperature can be retrieved via `gladm system-get-info`.

For details about the remaining data after the alarm, see section [Alarm Mechanism \(p. 42\)](#).

6.6 Restarting the Device

The device can be restarted by either restarting the (virtual) machine via `gladm device-restart` or alternatively by sending a `REBOOT` to the Linux driver.

```
echo REBOOT > /proc/driver/cs2.0
```

The confirmation for a successful reset can be found in `dmesg`. The driver may still need to reload.



The command `gladm device-restart` or sending a `REBOOT` to the Linux driver can only be executed locally or via SSH.



Rebooting the device in short sequence will trigger the recovery mode!
Always wait for the reboot to finish!
A complete reboot takes up to 50 seconds.

6.7 License Files

The license file determines:

- Number of cHSMs
- Number of CBKs
- Available templates

Two types of performance packs are available for u.trust Anchor:

- without cryptographic accelerator (Se100/2k/5k)
- and with cryptographic accelerator (Se15k/40k/CSAR8, CSAR16., CSAR31)

The following performance upgrade combinations are possible:

<i>Initial Configuration</i>	<i>Possible Model Upgrade</i>	<i>cHSM Upgrade</i>	<i>RSA 2k Performance Upgrade</i>	<i>Cryptographic Accelerator</i>
Se100	Se2K	4	2000	✗
	Se5K	8	5000	✗
Se2K	Se5K	8	15000	✗
Se15K	Se40K	12	40000	✓
	CSAR8 (Standard)	8	40000	✓

<i>Initial Configuration</i>	<i>Possible Model Upgrade</i>	<i>cHSM Upgrade</i>	<i>RSA 2k Performance Upgrade</i>	<i>Cryptographic Accelerator</i>
	CSAR16 (Plus)	12	40000	✓
	CSAR31(Premium)	31	40000	✓
Se40K	CSAR8 (Standard)	8	40000	✓
	CSAR16 (Plus)	12	40000	✓
	CSAR31 (Premium)	31	40000	✓
CSAR8 (Standard)	CSAR16 (Plus)	12	40000	✓
	CSAR31 (Premium)	31	40000	✓
CSAR16 (Plus)	CSAR31 (Premium)	31	40000	✓

Table 20: u.trust Anchor - License Upgrade Options



Please contact Utimaco Support to extend the license for a purchased product.

6.7.1 Updating License Files

Precondition:

- The license file can only be updated if a new license has been purchased. Please contact [Utimaco Support \(p. 118\)](#) to extend the license for a purchased product.
- A license file cannot be exchanged between different devices. Each license file is linked to the device serial number.



The license file update process should not be interrupted, otherwise the license file may be corrupted, invalidated or lost.

If the device detects a corrupted, invalid or missing license file, then only minimal operations are possible. In this state gladm commands can be executed, but no container can be used, i.e. there are no templates and no slots available. To leave this state a valid license file must be loaded via `gladm system-update-license`.

Procedure:

1. Check the current device license via [gladm system-get-license-info \(p. 152\)](#).

```
gladm -d 123.123.123.123 -u user01 -k my_key_priv.pem system-get-license-info license file
```

```
version: 3
Product name: u.trust Anchor CSAR Plus
Serial number: CS999999
Number of cHSMs: 12
Number of CBKs: 12
Included templates in license:
SecurityServer
```

2. Update the license file via `gladm system-update-license`.
3. Reboot the system.
4. Validate the new license via [gladm system-get-license-info \(p. 152\)](#).

```
gladm -d 123.123.123.123 -u user01 -k my_key_priv.pem system-get-license-info license file
```

```
version: 3
Product name: u.trust Anchor CSAR Premium
Serial number: CS999999
Number of cHSMs: 31
Number of CBKs: 31
Included templates in license:
SecurityServer
```

7 Troubleshooting

This chapter provides solutions to some possible situations when using u.trust Anchor.

7.1 How to Identify different Device States

This section deals with the situation where it is not known whether the u.trust Anchor device works at all, and in which the mode/state is unknown. The following steps should be systematically performed to check the u.trust Anchor's operativeness and, if possible, to get the u.trust Anchor working again.


Precondition

- Make sure that the u.trust Anchor and the host PC, whereon the administration tool gladm is installed, are properly connected to the network (try to 'ping' the u.trust Anchor from the host PC).

Procedure

1. Retrieve the device information and compare answers listed in the following table.

Command	Result	Explanation/Reason/Adjustment
gladm system-get-info	Device system version 4.70.0.0 Sensory Controller software version 3.02.0.8 Hardware revision number 7.03.0.3 UID ce00001d44b36701 SN CS850654 Device Type u.trust Anchor Se100 Vendor Secret is present on the device Vendor DAK Certificate is present on the device	The u.trust Anchor device is in state INITIALIZED . The device has been successfully claimed and is operational. The <code>global-admin-auth-key-crt</code> and the <code>dak-operator-crt</code> are present on the device.
gladm system-get-info	Initial user credentials unchanged Vendor Secret is present on the device Vendor DAK Certificate is present on the device	The u.trust Anchor device is in state FACTORY DEFAULT . The device needs to be claimed by the administrator with the Global Initial Admin Key received together with the device to reach full operational mode, see Claiming the Device (p. 56) .

Command	Result	Explanation/Reason/Adjustment
gladm system-get-info	No answer	Dead state or power down mode Perform the following steps to resolve the problem: 1. Restart the device, see section Restarting the Device (p. 107) . 2. If the device restarts and boots without errors, the problem is solved. Otherwise, please contact Utimaco, see section 2020-0035 Contact Address for Support (p. 118) .
gladm system-get-info	A zeroization event occurred / alarm occurred Vendor secret is missing on the device Vendor DAK certificate is missing on the device	An alarm has occurred on the u.trust Anchor device. See Leaving the Alarm State (p. 114) . <div> Though the u.trust Anchor device will be operational after resetting the alarm, no cHSM snapshots can be loaded and no backups can be restored because of the missing vendor secret and vendor DAK certificate. To obtain these again, please contact Utimaco to send the device back in for maintenance (see section Contact Address for Support Queries (p. 118)).</div>
gladm system-get-info	external_erase zeroization event(s) / alarm occurred Initial user credentials unchanged Vendor Secret is present on the device Vendor DAK Certificate is present on the device	An external erase has been executed and triggered an alarm. See Leaving the Alarm State (p. 114) .
gladm system-get-info	No zeroization event occurred Vendor secret is present on the device Vendor DAK certificate is present on the device	The u.trust Anchor device is fully accessible. → Continue to step 2.

Command	Result	Explanation/Reason/Adjustment
<code>gladm system-get-info</code>	Device system version 6.0.0-c	If the device version number contains the suffix <code>-c</code> , then the device has loaded a FIPS firmware image.
<code>gladm system-get-license-info</code>	Operation failed: CHAI_CONNECTION_CLOSED: The connection was closed by the other side. The connection limit may have been reached.	The license file is missing or invalid. See The License File is Corrupted, Invalid or Missing (p. 116)
<code>gladm system-get-license-info</code>	Operation failed: CHAI_BAD_REPLY: The client received a reply with bad parameters.	The license file is corrupted. See The License File is Corrupted, Invalid or Missing (p. 116)

- Retrieve the chain of trust via `gladm system-get-trust-chain`. Compare answers listed in the following table.

Result	Explanation/Reason/Adjustment
Missing file: global-admin-auth-key-crt Missing file: dak-operator-crt	The u.trust Anchor device is in <code>Factory Default</code> mode and needs to be claimed by the Global Administrator to generate the needed certificates. → Set up the u.trust Anchor according to section Claiming the Device (p. 56) to reach the state <code>INITIALIZED</code> .
No missing files reported	The u.trust Anchor device is in state <code>Initialized</code> and is fully operational.

7.2 Leaving the Recovery Mode

Condition

The u.trust Anchor device does not respond to any commands requiring authentication.

Perform the following steps to check whether the device is in recovery mode:

- Execute the command `gladm system-get-info`.
- Check the output of the returned parameter `version`. If it starts with `recovery_`, then the device is in recovery mode.

Reason

This can be the case when:

- an alarm has been triggered,
- all operational images on the device are broken,
- after the [External Erase \(p. 98\)](#) button has been pressed for more than 10 seconds,
- or the reboot process has been interrupted several times in short order (a full reboot takes up to 50 seconds).

Solution

If the recovery mode has been triggered by a [long external erase \(p. 101\)](#):

1. Perform a [short external erase \(p. 100\)](#), after that the device should restart with the operational image.
2. Clear the system.
3. Perform an [alarm reset \(p. 168\)](#).
4. Afterwards, the device can be set up again, using the GIAK, see Claiming the Device.

If the recovery mode has been triggered by an [alarm \(p. 42\)](#):

1. See [Leaving the Alarm State \(p. 114\)](#).

If the recovery mode has been triggered by any other reason:

1. [Restart the device. \(p. 107\)](#)
2. If the device restarts and boots without errors, the problem is solved. Otherwise, please contact [Utimaco Support \(p. 118\)](#).

7.3 Leaving the Alarm State

Condition

An alarm has been triggered, see [Alarm Mechanism \(p. 42\)](#) for details. For a full list of commands available in alarm state, see [Overview of gladm Commands \(p. 121\)](#).

Perform the following steps to check whether the device is in the alarm state:

1. Execute the command `gladm system-get-info`.
2. If the output returns `alarm present` and `A zeroization event occurred` or `zeroization event(s) / alarm occurred`, then the device is in the alarm state.

Reason

See [Alarm Triggers \(p. 43\)](#).

Solution

If the output of `gladm system-get-info` returns `alarm present` and `A zeroization event occurred`:

1. A tamper event has likely occurred. Execute `gladm system-get-audit-log` to verify the triggered alarm, see also [Audit Log Events \(p. 104\)](#).
2. [Contact Utimaco Support \(p. 118\)](#) to send the device back to Utimaco. See also (1.2.28) 2020-0035 Returning a Defective u.trust Anchor.

If the output of `gladm system-get-info` returns `zeroization event(s) / alarm occurred`:

1. An alarm occurred, but did not persist, e.g. temperature alarms or external erase. Execute `gladm system-get-audit-log` to verify the triggered alarm, see also [Audit Log Events \(p. 104\)](#).
2. Condition:
 - a. If the alarm has been triggered by an short external erase (erase button pressed for >2 seconds): [Clear the system \(p. 170\)](#) and continue with step 3.
 - b. If the alarm has been triggered by an long external erase (erase button pressed for <10 seconds): Perform the steps for [Leaving the Recovery Mode \(p. 113\)](#).
 - c. If the alarm was triggered by other reasons (see [Alarm Triggers \(p. 43\)](#) for other possible reasons): Continue with step 3.
3. [Reset the alarm \(p. 168\)](#).

4. Condition:

- a. If the alarm was triggered by an external erase, the device can be set up again using the GIAK, see [Claiming the Device \(p. 56\)](#).
- b. If the alarm was triggered by other reasons (see [Alarm Triggers \(p. 43\)](#) for other possible reasons): The u.trust Anchor device will be operational after resetting the alarm, no cHSM snapshots can be loaded and no backups can be restored, because of the missing vendor secret and vendor DAK certificate. [Contact Utimaco Support \(p. 118\)](#) to replace them.

7.4 Leaving the Dead State

Condition

The device is completely unresponsive.

Reason

This can be the case when:

- the power-up self-tests failed,
- an unknown internal error occurred,
- all images on the device are broken,
- or the reboot process has been interrupted several times in short order (a full reboot takes up to 50 seconds).

Solution

1. [Restart the device \(p. 107\)](#).
2. If the device restarts and boots without errors, the problem is solved. Otherwise, please [contact Utimaco Support \(p. 118\)](#).

7.5 The License File is Corrupted, Invalid or Missing

Condition

An error message is returned, when `gladm system-get-license-info` is executed:


```
Operation failed:  
CHAI_CONNECTION_CLOSED:  
The connection was closed by the other side.  
The connection limit may have been reached
```

```
Operation failed:  
CHAI_BAD_REPLY:  
The client received a reply with bad parameters.
```

Reason

The license file is corrupted, invalid or missing. This could be caused, if the [license file update](#) (p. 109) has been interrupted.

Solution

1. Load a valid license file via `gladm system-update-license`.
2. Reboot the system.
3. Check the license via `gladm system-get-license-info`.

```
Example  
gladm -d 123.123.123.123 -u user01 -k my_key_priv.pem system-get-license-  
infoLicense file  
  
version: 3  
Product name: u.trust Anchor CSAR Premium  
Serial number: CS999999  
Number of cHSMs: 31  
Number of CBKs: 31  
Included templates in license:  
SecurityServer
```

4. If the device returns no errors, the problem is solved. Otherwise, please contact Utimaco Support.

8 Contact Address for Support Queries

If an error occurs while operating the u.trust Anchor device, read [Troubleshooting \(p. 111\)](#).

If the error still occurs, prepare diagnostic information in a `.txt` file with the information retrieved via `gladm system-get-info`.

If you have any further questions on u.trust Anchor, feel free to contact us.

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.com²
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.

² <mailto:support@utimaco.com>

9 References

Reference	Title/Company	Document Number
	u.trust Anchor FIPS 140-3 - Containerized Hardware Security Module (cHSM) - Administration Manual / Utimaco IS GmbH	2023-0028
	u.trust Anchor FIPS 140-3 - Containerized Hardware Security Module (cHSM) - User Manual / Utimaco IS GmbH	2023-0025
	u.trust Anchor FIPS 140-3 - csadm Manual / Utimaco IS GmbH	2023-0037
	u.trust Anchor LAN V5 FIPS 140-3 - Operating Manual / Utimaco IS GmbH	2024-0010
	u.trust Anchor PCIe FIPS 140-3 - Operating Manual / Utimaco IS GmbH	2024-0009
	u.trust Anchor LAN V5 FIPS 140-3 - Quick Start Guide / Utimaco IS GmbH	2024-0016
[EMMC]	TN-FC-32: eMMC Device Health Report / Micron Technology ³	
[FIPS140-3]	FIPS PUB 140-3, Security Requirements for Cryptographic Modules / National Institute of Standards and Technology (NIST), March 2019	
[FIPS186-4]	FIPS PUB 186-4F, Digital Signature Standard (DSS) / National Institute of Standards and Technology (NIST), July 2013	
[BRP-IP320]	RFC 5639: Elliptic Curve Cryptography ECC Brainpool Standard - Curves and Curve Generation, March 2010, including Errata, http://tools.ietf.org/html/rfc5639	

³ <https://dtsheet.com/doc/1384294/tn-fc-32--e.mmc-device-health-report>

10 Appendix

10.1 Global Administration Management (gladm) Tool

10.1.1 Introduction to gladm

The Global User Administration Tool *gladm* is used by the Global Administrator to configure the device, and to set up and maintain cHSMs.

This manual provides a detailed description of all available *gladm* commands with the following information:

- **Command**

The command in correct syntax with all available arguments;

- **Arguments**

All arguments available for the command listed in their available syntax, including a detailed description, possible values and dependencies, optional arguments are given in square brackets;

- **Example**

An example of the command in a use case is provided where possible;

- **Return**

The return of the command upon successful execution. If a command does not have any return parameters, nothing will be returned upon successful execution.

Inline Help

In addition to these descriptions, an inline help can be called at any time, providing a list of all available commands. It can be accessed via the following command:

```
gladm -h
```

It can also be used following a specific command, providing a description of the command along with a list and short description of the available arguments:

```
gladm chsm-create -h
```

10.1.2 Overview of gladm Commands

Command	Section	Available in Alarm State
Certificates and Secrets		
key-set-operator-secret	Generating an Operator Secret	✗
key-list-operator-secrets	Listing Operator Secrets (p. 127)	✗
key-delete-operator-secret	Deleting Operator Secrets (p. 129)	✗
key-import-cert	Importing a CA Certificate (p. 131)	✗
key-import-operator-secret	Importing an Operator Secret (p. 127)	✗
key-list-operator-secrets	Listing Operator Secrets (p. 127)	✗
key-get-wrapping-key	Retrieving a Wrapping Key (p. 131)	✗
User Management		
user-list	Listing Users (p. 132)	✓
user-add	Adding a User (p. 133)	✗
user-change-credentials	Changing Credentials (p. 134)	✗
user-permissions	Getting a User Permissions Template (p. 136)	✗
user-create-backup	Backing Up Users (p. 137)	✗
user-restore-backup	Restoring Users (p. 135)	✗
user-delete	Deleting a User (p. 136)	✗
Slot Management		
chsm-list-slots	Listing Slots (p. 138)	✓
chsm-free-slot	Freeing a Slot (p. 139)	✗
slot-get-quota	Getting the Slot Quota (p. 141)	✗
slot-set-quota	Setting the Slot Quota (p. 140)	✗
cHSM Management		

chsm-create	Creating a new cHSM (p. 142)	✗
chsm-clone		✗
chsm-snapshot		✗
chsm-halt	Halting a cHSM (p. 146)	✗
chsm-retrieve		✗
chsm-restore		✗
System		
system-list-templates	Listing Templates (p. 148)	✗
system-get-trust-chain	Retrieving the Chain of Trust (p. 149)	✗
system-get-audit-log	Retrieving the Audit Log (p. 150)	✓
system-get-info	Displaying Device System Information (p. 151)	✓
system-fetch-log	Reading the System Log (p. 153)	✗
system-get-metrics	Getting Device Metrics (p. 154)	✗
system-get-quorum-requirements	Getting the System Quorum (p. 159)	✓
system-set-quorum-requirements	Setting the System Quorum (p. 158)	✗
slot-get-quota	Getting the Slot Quota (p. 141)	✗
slot-set-quota	Setting the Slot Quota (p. 140)	✗
system-get-time	Getting the Time (p. 161)	✓
system-set-time	Setting the Time (p. 160)	✗
system-get-ntp-config	Getting the NTP Configuration (p. 162)	✗
system-set-ntp-config	Setting the NTP Configuration (p. 163)	✗
system-activate-ntp	Activating NTP (p. 166)	✗

system-reset-alarm	Resetting the Alarm (p. 168)	✓
device-restart	Restarting the Device (p. 169)	✗
system-restart	Restarting the Device (p. 170)	✗
system-update	Updating the Device Firmware (p. 172)	✗
system-clear	Clearing the System (p. 170)	✓
bash-completion	Emitting the Bash Completion Script (p. 173)	✗

10.1.3 Managing Certificates and Secrets

Operator secrets are used alongside a vendor secret to derive keys used for the encryption of snapshots. This approach prevents either party from decrypting any of the created snapshots.

A snapshot contains the data of a cHSM stored on disk at the time of taking the snapshot. It contains the user data created after the cHSM was created. Snapshots can be taken of running, halted, or locked cHSMs. When taking a snapshot of a running cHSM, the cHSM is temporarily halted and remains unavailable until the snapshot operation has been completed. Taking snapshots of cHSMs in cluster mode is not supported.

An operator secret consists of 32 bytes and is generated and stored securely by the operator. u.trust Anchor additionally offers the possibility to import wrapped operator secrets.

An operator secret must be loaded onto the device to create multiple cHSMs, and at least one active operator secret is necessary to create snapshots. The device can store multiple operator secrets, of which at most one can be active. The active secret is used when new snapshots are created. The other secrets are used to import snapshots that were created by using those secrets.

10.1.3.1 Generating an Operator Secret (key-set-operator-secret)

This command allows a Global Administrator to generate a new operator secret or to import an operator secret. The new/imported operator secret will become the active operator secret. The previous operator secret will remain available for loading user backups and snapshots. A temporary wrapping key will be obtained with which the operator secret will be encrypted using RSAES-OAEP with MGF1-SHA256 without label.

To manually encrypt the operator secret, use the command `gladm key-get-wrapping-key` to obtain a wrapping key. Set the encrypted operator secret with this command and add the wrapping key token file.



Global Administrators should use a dedicated smartcard for operator secrets (and not use the same smartcard for MBKs, for example).

Command

```
gladm -u <username> -k <credentials> -d <addr> key-set-operator-secret [-g k[,n]] [-f] [-r <record>] [-t <token_file>] [-s <size>] [--] <secret_path>
```

Argument	Description	Required
-g, --generate	The operator secret is generated, split acc. to Shamir's Secret sharing and stored in smartcards.	O
k	Number of shares needed to recover the secret. Default is 2.	O
n	Number of shares to be created. Default is 2.	O
-f, --force	Force overwrite, if the record already exists on smartcard.	O
-r, --record=<record>	The record number specifies the location where to store the key share on the smartcard. Value range: 1...16. Default is 7.	R
-t, --token_file=<token file>	The file name of the token identifying the wrapping key.	O
-s, --size=<size>	Key size in bits, minimum size is 2048. Default size is 2048.	O
<secret_path>	Plain or encrypted operator secret or smartcard specifier.	R

Example**Automatic generation of operator secret**

In this example `-g` generates the operator secret, splits it into 3 shares out of which 2 are needed to recover the secret.

The shares are saved on the smartcards. A wrapping key is obtained from the device, the operator secret is wrapped and imported into the u.trust Anchor.

The command is executed once and a message on the pinpad will be displayed, telling you when to press OK and when to insert the next card.

```
gladm key-set-operator-secret -g 2,3 -r 7 :cs2:auto:USB0
```



The command can not be executed, if an operator secret is already present on the smartcard. Use `[-f]` to overwrite the existing record on the smartcard.

```
gladm key-set-operator-secret -g 2,3 -f -r 7 :cs2:auto:USB0
```

Importing operator secret from smartcard by reading key shares from smartcards

When the command `gladm key-set-operator-secret` is called with a smartcard specifier as argument, then the shares of the operator secret are read from the smartcards one after the other and the operator secret is reconstructed. A wrapping key is obtained from the device, the operator secret is wrapped and imported into the u.trust Anchor.

```
gladm key-set-operator-secret -r 7 :cs2:auto:USB0
```

Importing operator secret from a file

The following example imports an operator secrets from the file `operator_secrtet_01` into the device.

A wrapping key is obtained from the device, the operator secret is wrapped and imported into the u.trust Anchor.

```
gladm key-set-operator-secret operator_secrtet_01
```



This option should only be used on secure host systems.

Importing operator secret with -t token_file

The following example imports an operator secret `operator_secrtet_01` with the wrapping key token given in the file `my_token`.

```
gladm key-set-operator-secret -t my_token operator_secrtet_01
```



This operation requires the execution of `gladm key-get-wrapping-key` as a preliminary step.

Return

Upon successful generation of the operator secret, gladm displays a message that the operation has been successfully completed.



u.trust Anchor Smartcard Specifiers

A smartcard specifier always starts with a colon and consists of three strings separated by colons (for example, `:cs2:cjo:USB0`):

- The first string identifies the type of the smartcard.
- The second string identifies the type of the PIN pad (smartcard reader with keyboard and display).
- The last string is the name of the USB device the reader is connected to.

Currently, only the smartcards of type TC30 and JavaCard, with the identifier cs2, are supported:
The following types of PIN pads are supported:

<i>PIN Pad Identifier</i>	<i>PIN pad type</i>
cyb	REINER SCT cyberJack (COM) or REINER SCT cyberJack (USB)
cjo	Utimaco cyberJack one
auto	Automatic detection of the PIN pad type

Supported PIN pads

The following port types are supported:

<i>Port Identifier</i>	<i>Description</i>
USBn where n = {0, 1, ...}	USB port No. n+1
USBn[@<port>]@<IP address>[/<password>] [#<smartcard PIN>] where n = {0, 1, ...}	With PIN Pad Daemon

Key Specifier Examples

Key Specifier	Description
:cs2:cjo:USBn or :cs2:auto:USBn where n = {0, 1, 2, ...}	Key from a smartcard using a PIN pad Utimaco cyberJack one (USB) connected to a USB port of a Windows/Linux computer

10.1.3.2 Listing Operator Secrets (key-list-operator-secrets)

This command lists all available operator secrets. They are each identified by a 4-byte fingerprint, displayed as an 8-character hexadecimal stream in the output list.

Only the most recent operator secret is active.

Command		
<code>gladm -u <username> -k <credentials> -d <addr> key-list-operator-secrets</code>		
Argument	Description	Required
No arguments are given for input.		
Return		
Upon successful execution of the command, gladm returns a list of the available operator secrets, along with the file date.		
<pre>646d4110: 2021-06-24T12:26:21+0000 - active fecf521a: 2021-06-24T12:26:16+0000 da1bcbb3: 2021-06-24T12:26:12+0000</pre>		
In case of a failure, an error code is returned.		

10.1.3.3 Importing an Operator Secret (key-import-operator-secret)



For a detailed description on importing wrapped operator secrets, see *Importing an Operator Secret* in the *u.trust Anchor - Administration Manual*.

This command imports a wrapped operator secret which has been wrapped with a unique wrapping key, which can be retrieved as described in 2020-0035 Retrieving a Wrapping Key (key-get-wrapping-key).



`gladm key-import-operator-secret` is deprecated, please use `gladm key-set-operator-secret` instead.

The operator secret is identified by a unique token. When executing this command, the wrapping key that was used to wrap the operator secret is deleted from the device to ensure every wrapping key is only used once. The imported operator secret is then unwrapped and stored as the new active operator secret on the device.



If an inactive copy of the operator secret is already present on the device, it has to be deleted before reimport.

Command

```
gladm -u <username> -k <credentials> -d <addr> key-import-operator-secret <token> <secret>
```

Argument	Description	Required
<token>	The file name of the wrapping key token as returned when the wrapping key was retrieved	R
<secret>	The file name of the encrypted operator secret	R

Example

The following example imports an operator secret `OP_S` with the wrapping key token given in the file `token`:

```
gladm key-import-operator-secret token OP_S
```

Return

Upon successful import of the operator secret, gladm returns the 4-byte fingerprint of the operator secret, displayed as an 8-character hexadecimal stream in the output list. In case an inactive copy of the operator secret is present on the device, the import will fail and an error message will be returned. In case of other failures, an error code is returned.

10.1.3.4 Deleting Operator Secrets (key-delete-operator-secret)

This command deletes an operator secret from the device. The operator secret is specified by its fingerprint, given for input as an 8-character hexadecimal stream.

If the deleted operator secret was active, then no operator secret is active after the deletion, which disables the ability to take snapshots. To enable it again, an operator secret must be imported again.

Deleting an operator secret that was previously used to take a snapshot makes the restoring of that specific snapshot impossible. To be able to restore it, the related operator secret has to be loaded again.

Command		
<code>gladm -u <username> -k <credentials> -d <addr> key-delete-operator-secret <os-fingerprint></code>		
Argument	Description	Required
<os-fingerprint>	The fingerprint of the operator secret to be deleted, given as an 8-character hexadecimal stream	R
Example		
The following example deletes the operator secret with the fingerprint 227c9ba1: <code>gladm key-delete-operator-secret 227c9ba1</code>		
Return		
Upon successful deletion of the operator secret, gladm returns no parameters or messages. In case of a failure, an error code is returned.		

10.1.3.5 Copying an Operator Secret (smartcard-copy-secret)

This command allows a Global Administrator to copy an operator secret from smartcard to smartcard, or to transfer an operator secret from a key file to a smartcard. Therefore, the source of the secret may be a key file or a smartcard specifier, whereas the target must be a smartcard specifier.

Furthermore, the administrator may specify the exact record number on the smartcard from and into which a key share shall be read or written, as well as the number of key shares.

If smartcards are used: A PIN pad including a smartcard reader can be connected to the computer where gladm is running (USB port) or to another computer. Watch the display of the PIN pad for instructions on further command processing.

If key files are used: We strongly recommend using encrypted key files. If encrypted key files are used, we strongly recommend using a hidden password entry.



Global Administrators should use a dedicated smartcard for operator secrets (and not use the same smartcard for MBKs, for example).

Command

```
gladm smartcard-copy-secret [-s <source-record>] [-t <target-record>] [-k <val>]
[-n <val>] [-f] [--] <source_secret> <target_secret>
```

Argument	Description	Required
-s, --source-record=<source-record>	The record of the smartcard where the source_secret is stored. The default is 7. It is applicable only to smartcard specifiers.	0
-t, --target-record=<target-record>	The record of the smartcard where the target_secret is stored. The default is 7. It is applicable only to smartcard specifiers.	0
-k <val>	Number of shares needed to recover the secret, only applicable if source is a file. Default is 2.	0
-n <val>	Number of shares to be created, only applicable if source is a file. Default is 2.	0
-f, --force	Force overwrite, if the record already exists on smartcard.	0
<source_secret>	A file or smartcard specifier from where the source secret is read.	R
<target_secret>	A smartcard specifier to where the target secret is stored.	R



An operator secret share stored on a smartcard cannot be deleted, it can only be overwritten. If another share is already stored in the specified record, it is overwritten by the share to be generated by using the override flag option.

Example

Copying an operator secret from smartcard to smartcard

The following example copies an operator secret from smartcard to smartcard, reading and writing from and into the default record number 7:

```
gladm smartcard-copy-secret :cs2:cjo:USB0 :cs2:cjo:USB0
```

Copying an operator secret from a file to a smartcard

The following example copies an operator secret, creates 2 out of 2 shares from a file and puts them on smartcards in record number 6:

```
gladm smartcard-copy-secret -t 6 -k 2 secret_key_file :cs2:cjo:USB0
```

Return

Upon successful copying of the operator secret, a message that the operation has been successfully completed is displayed.



For more information on smartcard specifiers, please see chapter [Generating an Operator Secret \(key-set-operator-secret\)](#) (p. 123).

10.1.3.6 Importing a CA Certificate (key-import-cert)

This command imports an X.509 CA certificate in DER or PEM encoding for the Device Authentication Key (DAK) into the device. It is also possible to import a DER- or PEM-encoded chain of certificates. Every certificate within the chain has to be signed by the following certificate, with the exception of the last certificate of the chain.

The certificates are verified by the device before they are stored. This process is relevant in context of setting up the chain of trust.

Command

```
gladm -u <username> -k <credentials> -d <addr> key-import-cert <certificate filename>
```

Argument	Description	Required
<certificate filename>	The name of the file that holds the certificate or certificate chain	R

Example

The following example imports a certificate from the file `certificate_01` into the device:

```
gladm key-import-cert certificate_01
```

Return

Upon successful importing of the certificate, gladm returns no parameters or messages. In case of a failure, an error code is returned.

10.1.3.7 Retrieving a Wrapping Key (key-get-wrapping-key)

This command obtains a temporary wrapping key for the encrypted import of an operator secret. This key can be used to encrypt an operator secret using RSAES-OAEP with MGF1-SHA256, no label.

To use the obtained key to wrap an operator secret, openssl can be used as stated in the inline help of this command.



If there are already 32 wrapping keys present, the oldest one is deleted before storing the new one.

Command		
gladm -u <username> -k <credentials> -d <addr> key-get-wrapping-key [-w <filename>] [-t <filename>] <key_size>		
Argument	Description	Required
<key_size>	RSA key size in bits, min. 2048	R
-w <filename>	The file name of the wrapping key If no filename is specified, a file wrapping-key_YYYYMMDDhhmmss.crt will be created.	0
-t <filename>	The file name of the token identifying the wrapping key If no filename is specified, a file wrapping-key-token_YYYYMMDDhhmmss will be created.	0

Example
The following example retrieves a wrapping key w_key identified by the token given in token. gladm key-get-wrapping-key -w W_key -t token

Return
Upon successful retrieving of the wrapping key, gladm returns the certificate and the ID of the private wrapping key. In case of a failure, an error code is returned.

10.1.4 Managing Users

10.1.4.1 Listing Users (user-list)

This command lists all stored device users.



No user authentication is needed to perform this command.

Command

```
gladm -d <addr> user-list [-v]
```

Argument	Description	Required
-v --verbose	If this parameter is set, the return includes detailed information about the user's command permissions.	0

Return

Upon successful execution of the command, gladm returns a list of all users, optionally with additional command permission information.

10.1.4.2 Adding a User (user-add)



For details on the encrypted key files that can be used for user authentication, see *Keys for User Authentication* in the [u.trust Anchor - Administration Manual](#) (p. 119).

This command adds a device user who can then carry out administrative tasks on the device, either on their own or together with other eligible users for quorum-controlled operations.

Command

```
gladm -u <username> -k <credentials> -d <addr> user-add [-p <val>] <name>  
<keyfile>
```

Argument	Description	Required
-p <val>	The filename of the permission configuration to be used for quorum authentication If no file is given, the added user will not have any permissions.	0
<name>	The name of the user to be added User names may include any combination of alphanumeric characters and certain special characters (., @, #, -). They can be between 1 and 32 characters long and must not start with "-".	R

Argument	Description	Required
<keyfile>	The user credentials of the new user given in form of the public part of the key The credentials will be stored on the device. When authenticating, the user will have to sign a challenge with the private counterpart of the credentials.	R

Example

The following example adds a user with the name `new-user`, the keyfile `newuserkey_pub.pem` and the permission configuration file `permission.cfg`:

```
gladm user-add -p permission.cfg new-user newuserkey_pub.pem
```

Return

Upon successful creation of the new user, gladm returns no parameters or messages. In case of a failure, an error code is returned.



If `gladm user-add` is executed using a smartcard, then the EC key is automatically picked.

10.1.4.3 Changing Credentials (user-change-credentials)

This command changes the credentials of a device user.

If the target user name is not authenticated within the current session, the command will fail. On success, all active sessions for which the user was authenticated are closed. A result message indicating success will still be sent before the current session is closed. Additionally, if the user had pre-defined credentials (i.e., the user was an initial user), their current set of commands to which they are eligible to contribute and their authentication are replaced with an extended set.



Upon successful execution of the command, the session for the user whose credentials have changed is terminated.

Command

```
gladm -u <username> -k <credentials> -d <addr> user-change-credentials <name>  
<keyfile>
```

Argument	Description	Required
<name>	The name of the user	R
<keyfile>	The new user credentials given in form of the public part of the key The credentials will be stored on the device. When authenticating, the user will have to sign a challenge with the private counterpart of the credentials.	R

Example

The following example changes the credentials of the user `example-user` with the information contained in `new_pub.pem`:

```
gladm -u example-user -k old_key.pem user-change-credentials example-user new_pub.pem
```

Return

Upon successful change of credentials of the user, gladm returns no parameters or messages. In case of a failure, an error code is returned.

10.1.4.4 Restoring Users (user-restore-backup)

This command restores user data from a file containing user data that has previously been backed up.



The user backup contains the quorum requirement configuration. When the user backup is restored, the user database is replaced entirely, along with the quorum requirement configuration from the backup. The current session is terminated for all users from the database.

Command

```
gladm -u <username> -k <credentials> -d <addr> user-restore-backup <user-backup filename>
```

Argument	Description	Required
<user-backup filename>	The filename of the backup from which the user data is to be restored	R

Example

The following example restores the user data from a file with the name `user_backup.bkp`:

```
gladm user-restore-backup user_backup.bkp
```

Return

Upon successful restoring of the user data, gladm returns no parameters or messages. In case of a failure, an error code is returned.

10.1.4.5 Deleting a User (user-delete)

This command deletes a device user.

There are checks in place to avoid a lockout situation.



Upon successful execution of the command, the session for the deleted user is terminated.

Command

```
gladm -u <username> -k <credentials> -d <addr> user-delete <name>
```

Argument	Description	Required
<name>	The name of the user to be deleted	R

Example

The following example deleted the user with the name `user1`:

```
gladm user-delete user1
```

Return

Upon successful creation of the new user, gladm returns no parameters or messages. In case of a failure, an error code is returned.

10.1.4.6 Getting a User Permissions Template (user-permissions)

This command provides a template file containing user permissions for quorum authentication used when adding a new user.

Command

```
gladm -u <username> -k <credentials> -d <addr> user-permissions [-f <val>]
```

Argument	Description	Required
-f <val>	The name of the file for the permission template to be written to. If no filename is given, the permissions will be stored as <code>user_permissions.cfg</code> . If a file with the given filename already exists, it will be overwritten.	0

Example

The following example saves the user permissions template as `permission_template.cfg`:

```
gladm user-permissions -f permissions_template.cfg
```

Return

Upon successful saving of the permission template, the destination of the file will be returned. In case of a failure, an error code is returned.

10.1.4.7 Backing Up Users (user-create-backup)

This command backs up all user data of the device to be later restored again.



The user backup contains the quorum requirement configuration. When the user backup is restored, the user database is replaced entirely, along with the quorum requirement configuration from the backup.

Command

```
gladm -u <username> -k <credentials> -d <addr> user-create-backup [-f <val>]
```

Argument	Description	Required
-f <val>	The filename of the user backup to be written If no filename is specified, a file <code>user-backup_YYYYMMDDhhmmss.bkp</code> is created	0

Example

The following example backs up the user data and saves it in a file with the name `user_backup.bkp`:

```
gladm user-create-backup -f user_backup.bkp
```

Return

Upon successful creation of a backup of the user data, gladm returns no parameters or messages. In case of a failure, an error code is returned.

10.1.5 Managing Slots

Within a u.trust Anchor device, cHSMs are organized into slots.

A cHSM slot ID is a numerical identifier in the range of 1 to the maximum number of cHSMs available for your u.trust Anchor model, mirroring the maximum of cHSMs running simultaneously on a single u.trust Anchor device. To learn more about the available cHSM models, see section [u.trust Anchor Product Line \(p. 12\)](#).

<i>Model</i>	<i>cHSMs</i>
u.trust Anchor CSAR Premium	31
u.trust Anchor CSAR Plus	16
u.trust Anchor CSAR Standard	8

Table 21: u.trust Anchor CSAR models

The identifier must be provided by the operator for most cHSM management operations. Each cHSM occupies a given slot during its lifetime on the device. A snapshot of the cHSM may be restored to a different slot than the slot of its original creation. This way, the cHSM slot ID can be reused by a cHSM belonging to a different customer when it is loaded into that particular original cHSM slot.

When directly communicating with a cHSM on the host operating system, the slot ID may be used as a suffix for the path to the PCIe device mode. For example, `/dev/cs2.0.3` (Linux only) refers to slot 3 on the device `dev/cs2.` Different port numbers are mapped to different configurable slots. Another example is `4001@194.168.4.107`⁴.

For each slot, a quota can be set, determining the CPU distribution across the cHSMs on the u.trust Anchor device, as well as additional attributes for each slot (see [Setting the Slot Quota \(slot-set-quota\)](#) (p. 140)).



The quota of a slot can only be set when the slot is empty.

10.1.5.1 Listing Slots (`chsm-list-slots`)

This command lists all accessible cHSM slots within the card.

For each occupied slot, the slot number, unique cluster identifier and the template name are returned, along with the state of the cHSM.



No user authentication is needed to perform this command.

⁴ <mailto:4001@194.168.4.107>

Command		
gladm -d <addr> chsm-list-slots		
Argument	Description	Required
No arguments are given for input.		
Return		
<p>Upon successful execution, all accessible slots are listed by slot number. Empty slots hold no additional information. Slots that are occupied by a cHSM give the following information:</p> <ul style="list-style-type: none">▪ unique cluster identifier▪ template name▪ operation mode in square brackets: <code>regular</code> or <code>cluster</code>▪ state of the cHSM <div><pre>1: e8e389d7-4a91-4205-3464-dd403d45f452 SecurityServer [regular] - running 2: a6d875d9-5b08-7857-5897-ca204f54a439 SecurityServer-FIPS [cluster] - running [...] 32:</pre></div> <p>In case of a failure, an error code is returned.</p>		

10.1.5.2 Freeing a Slot (chsm-free-slot)

This command frees a specified cHSM slot.

Only slots that are occupied by cHSMs in a halted or failed state can be freed. It is not possible to free a slot that holds a cHSM in running state. A running cHSM must be set to a halted state first.



The cHSM in the specified slot is removed from the device along with its associated data and cannot be recovered. It is advised to take a snapshot before freeing the slot.

Command
gladm -u <username> -k <credentials> -d <addr> chsm-free-slot <slot id>

Argument	Description	Required
<slot_id>	The ID of the slot that is to be freed	R


Example
The following example frees slot 3 : gladm chsm-free-slot 3


Return
Upon successful freeing of the slot, gladm returns no parameters or messages. In case of a failure, an error code is returned.

10.1.5.3 **Setting the Slot Quota (slot-set-quota)**

This command sets the quota for an empty slot. The quota determines the distribution of CPU across all cHSMs present on the device, as well as additional cHSM specific attributes.

The quota is valid for any cHSM that is created in the slot. While no parameter besides the slot ID is mandatory, executing the command without any additional parameter will have no effect.

 The quota can only be set for empty slots.

 To always distribute the maximum available amount for a parameter, set it to 0 .

Command
gladm -u <username> -k <credentials> -d <addr> slot-set-quota [-c <val>] [-t <val>] [-m <val>] [-f <val>] [-i <val>] [-x <val>] [-r <val>] <slot_id>

Argument	Description	Required
<code>-c</code> or <code>--cpu=</code>	CPU weight in percent The CPU available on the u.trust Anchor device is distributed across all cHSMs present on the device according to the relation of the slot quota. Example: If slot 1 is set to <code>--cpu=100</code> and slot 2 is set to <code>--cpu=50</code> , the cHSM in slot 1 will receive 2/3 of the available CPU, while the cHSM in slot 2 will receive 1/3 of the available CPU.	0
<code>-t</code> or <code>--time=</code>	CPU time in percent If the CPU time of several cHSMs is set to 100%, they will always get the maximum of CPU time available distributed across cHSMs at that time.	0
<code>-m</code> or <code>--memory=</code>	Maximal memory in bytes Mem max: 134217728 Mem min: 0 Flash max: 67108864 Flash min: 0	0
<code>-f</code> or <code>--flash=</code>	Maximal Flash space in bytes	0
<code>-i</code> or <code>--io=</code>	Flash read/write rate in bytes per second	0
<code>-x</code> or <code>--xrs=</code>	Accelerator rate in point operations per second	0
<code>-r</code> or <code>--rngs=</code>	RNG rate in blocks per second	0
<code><slot_id></code>	The ID of one or several slots the quota is to be set for.	R

Example

The following example sets the quota for slot 3 with a CPU distribution of 20% and an accelerator rate of 32768 op/s:

```
gladm slot-set-quota -c 20 -x 32768 3
```

Return

Upon successful setting of the quota, gladm returns a message with the parameters and their set values. In case of a failure, an error code is returned.

10.1.5.4 Getting the Slot Quota (slot-get-quota)

This command returns the quota of a slot. The quota determines the distribution of CPU across all cHSMs present on the device, as well as additional cHSM specific attributes.

Command

```
gladm -u <username> -k <credentials> -d <addr> slot-get-quota <slot_id>
```

Argument	Description	Required
<slot_id>	The ID of the slot the quota should be returned for	R

Example

The following returns the quota for slot 3 :
`gladm slot-get-quota 3`

Return

Upon successful getting of the quota, gladm returns the quota set for the following attributes:

- CPU distribution in percent
- CPU time in percent
- Maximal memory in bytes
- Maximal Flash space in bytes
- Flash read/write rate in bytes per second
- Accelerator rate in point operations per second
- RNG rate in blocks per second

In case of a failure, an error code is returned.

10.1.6 Managing cHSMs

10.1.6.1 Creating a new cHSM (chsm-create)



Elliptic Curve (EC) keys that are used with this function on FIPS cHSMs have to be generated either over NIST P-256 or P-521, see [\[FIPS186-4\]](#) (p. 119).

This command creates a new cHSM.

The only required arguments are the admin key file and the slot ID. Optionally, other data can be specified like a template or the creation directory. If no additional Init Data is given, default values are used for each argument.

After the cHSM has been created, it will provide the PEM-encoded DAK signed certificates of the CAK and of the IAK. Finally, a PEM-encoded CSR is created for the CAK to be signed by the customer. These certificates, the CSR and the certificate chains for the DAK should be provided to the customer by the operator so that they can establish their chain of trust with their cHSM.

Command
<code>gladm -u <username> -k <credentials> -d <addr> chsm-create [-t <name>] [-v <version>] [-p <directory_path>] [--] <admin-key-file> <slot></code>

Argument	Description	Required
-t --template-name=<name>	The name of the template to be used. If this argument is not given, the default template <code>SecurityServer</code> is used.	0
-v --template-version=<version>	The version of the template to be used. If this argument is given, the argument -t has to be given as well. If this argument is not given, the most recent version of the template is used.	0
-p --path=<directory_path>	The directory where the certification files are to be written. If this argument is not given, the current working directory is used.	0
<admin-key-file>	The cHSM initialization data contained in an admin key file. The end customer that is to use the cHSM should give the public part of their Default Administration Key to the Global Administrator to be used as initialization data. This key should be generated via openssl.	R
<slot>	The slot id where the cHSM is to be created.	R

Example
<p>The following example creates a new cHSM with the default <code>SecurityServer</code> template using a given admin key file in slot <code>1</code>:</p> <pre>gladm chsm-create myADMIN.key 1</pre>

Return

Upon successful creation of the cHSM, gladm returns the slot number used for the new cHSM along with the location of authentication and initialization keys.

Creating cHSM with parameters:

```
slot: 4
initialization data: customer_chsm.key
cHSM creation receipt written to: chsm-receipt_004_20210706151326.txt
cHSM receipt signature written to: chsm-signature_004_20210706151326
cHSM auth key signature certificate signing request written to: chsm-auth-
key_004_20210706151326.csr
device cHSM auth key certificate written to: chsm-auth-key-device-
cert_004_20210706151326.pem
glad auth key certificate written to: glad-auth-key-device-
cert_004_20210706151326.pem
vendor device auth key certificate chain written to: dak-vendor-
chain_004_20210706151326.pem
operator device auth key certificate chain written to: dak-operator-
chain_004_20210706151326.pem
```



After executing the command, the successful creation of the new cHSM can additionally be verified by listing the cHSM slots and checking the respective slot number.

10.1.6.2 Cloning a cHSM (chsm-clone)

This command clones a cHSM from a snapshot to either one slot, or to multiple slots to create a cHSM cluster. The snapshot is verified by the device before the cHSM is cloned.

If the snapshot has been used for cloning before, the new cHSMs will be added to the existing cluster. The slot used for cloning must be free and the new cHSMs will be started with their mode set to the mode of the cluster. The cloned cHSMs will not allow any commands that would cause their state to go out of sync with the other cHSMs of the respective cluster.



The snapshot can only be decrypted if both the Manufacturer Secret and the Operator Secret used for deriving the encryption key are present on the device.

Command

```
gladm -u <username> -k <credentials> -d <addr> chsm-clone <snapshot filename>
<slot_id>
```

Argument	Description	Required
<snapshot filename>	The name of the file containing the snapshot	R
<slot id>	The ID of the single or multiple slots to clone the cHSM to Multiple slots are given separated by one space. The given slots must be free. If multiple slots are given and one of them is occupied, no clones will be created at all.	R

Example

The following example uses the snapshot chsm_snapshot001 to clone the cHSM to the slots 2, 8 and 19 :

```
gladm chsm-clone chsm_snapshot001 2 8 19
```

Return

Upon successful cloning of the cHSM, gladm returns no parameters or messages. In case of a failure, an error code is returned.

10.1.6.3 Taking a Snapshot (chsm-snapshot)

This command takes a snapshot of a cHSM and stores it so that it can be restored at a later point.

A snapshot contains the data of a cHSM stored on disk at the time of taking the snapshot. It contains the user data created after the cHSM was created. Snapshots can be taken of running, halted or locked cHSMs. When taking a snapshot of a running cHSM, the cHSM is temporarily halted and remains unavailable until the snapshot operation has been completed. Taking snapshots of cHSMs in cluster mode is not supported.



The snapshot can only be decrypted if both the Manufacturer Secret and the Operator Secret used for deriving the encryption key are present on the device.

Command

```
gladm -u <username> -k <credentials> -d <addr> chsm-snapshot -f snapshot_filename <slot_id>
```

Argument	Description	Required
-f --filename	The name of the file the snapshot should be written to. If no filename is specified, a file chsm-snapshotnnn, with nnn being the slot ID, is created.	0

Argument	Description	Required
<slot_id>	The slot of the cHSM of which the snapshot is to be taken.	R

Example

The following example takes a snapshot of the cHSM in slot 1 and stores it as `chsm_snapshot001` :
`gladm chsm-snapshot -f chsm_snapshot001 1`

Return

Upon successful creation of the cHSM, gladm returns no parameters or messages. In case of a failure, an error code is returned.

10.1.6.4 Halting a cHSM (chsm-halt)

This command sets one or more cHSMs to a halted state.

Setting a cHSM to a halted state will make it unresponsive to all requests, and makes it possible to free the slot occupied by the cHSM.



A halted cHSM cannot be directly set back to a running state. The data of the halted cHSM can still be obtained by taking a snapshot.

Command

```
gladm -u <username> -k <credentials> -d <addr> chsm-halt <slot id>
```

Argument	Description	Required
<slot_id>	The ID of one or multiple slots in which the contained cHSMs are to be halted Multiple slots are given separated by one space. The cHSMs in the slots must be in running state. If multiple slots are given and one of the cHSMs is not in running state, the ineligible cHSM will not be halted, but not influence the successful halting of the eligible cHSMs.	R

Example

The following example halts the cHSMs in the slots `6`, `8` and `14` :
`gladm chsm-halt 6 8 14`

Return

Upon successful creation of the cHSM, gladm returns no parameters or messages. In case of a failure, an error code is returned.

10.1.6.5 Retrieving a cHSM (chsm-retrieve)

This command sets the cHSM to a halted state, takes a snapshot and retrieves it from the device. Taking snapshots of cHSMs in cluster mode is not supported.

After the execution of this command, the cHSM will no longer be available on the device. The cHSM can still be restored or cloned using the snapshot.

Command

```
gladm -u <username> -k <credentials> -d <addr> chsm-retrieve [-f <val>] <slot id>
```

Argument	Description	Required
-f -- filename=<val>	The name of the file the snapshot is to be written to If no file name is given, the snapshot is saved to chsm-snapshot<slot id>	O
<slot id>	The slot of the cHSM that is to be retrieved	R

Example

The following example halts the cHSM in slot 3 and retrieves it to the file `chsm_retrieved`:

```
gladm chsm-retrieve -f chsm_retrieved 3
```

Return

Upon successful halting and retrieving of the cHSM, gladm returns no parameters or messages. In case of a failure, an error code is returned.

10.1.6.6 Restoring a cHSM (chsm-restore)

This command restores a cHSM from a snapshot to a free slot.

The snapshot is verified by the device before the cHSM is restored. The cHSM will be started in running state.



The snapshot can only be decrypted if both the Manufacturer Secret and the Operator Secret used for deriving the encryption key are present on the device.

Command		
<pre>gladm -u <username> -k <credentials> -d <addr> chsm-restore -m <snapshot filename> <slot_id></pre>		
Argument	Description	Required
<pre>-m</pre> or <pre>--migrate</pre>	<p>Must be used to update the firmware module versions of snapshots when they are lower than the versions used in the template on the device. The general rule:</p> <ul style="list-style-type: none"> Module version in snapshot > module version in template on device: restoring the snapshot fails. Module version in snapshot = module version in template on device: it works - no migration needed Module version in snapshot < module version in template on device : the snapshot needs to be migrated 	0
<code><snapshot filename></code>	The name of the file containing the snapshot.	R
<code><slot_id></code>	The slot where the cHSM is to be restored to. The slot must be free. It is not possible to overwrite a cHSM that is occupying a slot.	R
Example		
<p>The following example restores the cHSM from the snapshot file <code>chsm_snapshot001</code> to slot <code>5</code> :</p> <pre>gladm chsm-restore chsm_snapshot001 5</pre>		
Return		
<p>Upon successful creation of the cHSM, gladm returns no parameters or messages. In case of a failure, an error code is returned.</p>		

10.1.7 System Commands

10.1.7.1 Listing Templates (system-list-templates)

All cHSM instances are originally created from a cHSM template.

Each template includes the software run inside the cHSM. The available templates are determined by the firmware that has been flashed to the card.

cHSMs that are created from a template do not contain any secrets or credentials, and must be further configured in the process of creating the cHSM.

Command		
<code>gladm -u <username> -k <credentials> -d <addr> system-list-templates</code>		
Argument	Description	Required
No arguments are given for input.		
Return		
The return includes all available templates, determined by the firmware that has been flashed to the card. For each template, the name and version are stated. The default templated is marked with a '*'.		
<pre>SecurityServer 4.48.0 * SecurityServer-FIPS 4.47.0</pre>		
In case of a failure, an error code is returned.		

All cHSM instances are originally created from a cHSM template.

Each template includes the software run inside the cHSM. The available templates are determined by the firmware that has been flashed to the card.

cHSMs that are created from a template do not contain any secrets or credentials, and must be further configured in the process of creating the cHSM.

Command		
<code>gladm -u <username> -k <credentials> system-list-templates</code>		
Argument	Description	Required
No arguments are given for input.		
Return		
The return includes all available templates, determined by the firmware that has been flashed to the card. For each template, the name and version are stated. The default templated is marked with a '*'.		
<pre>SecurityServer 0.16.0 * SecurityServer-FIPS 0.16.0</pre>		
In case of a failure, an error code is returned.		

10.1.7.2 Retrieving the Chain of Trust (system-get-trust-chain)

This command requests the Certificate Signing Request (CSR) for the DAK given upon initialization of the device.

Upon execution, the command writes the CSR as well as the currently available certificates. This process is relevant in context of setting up the chain of trust.

Command		
<code>gladm -u <username> -k <credentials> -d <addr> system-get-trust-chain [-p <directory path>]</code>		

Argument	Description	Required
<code>-p</code> <code>--path=<directory path></code>	The directory to which the certificate files should be written If no directory is specified, the files are written to the current directory. If the specified directory does not exist yet, it will be created.	0

Example		
Example The following example requests the CSR with the path for the certificate files specified as <code>trust_chain</code> :		
<pre>gladm system-get-trust-chain -p trust_chain</pre>		

Return		
During the execution of the command, gladm displays several working messages. In case of operating a newly created device, the listing of missing files is to be expected, as those files still need to be imported. Upon successful creation of the cHSM, gladm returns which files were written and their location. device auth certificate signing request written to: device-auth-key.csr operator device auth key certificate chain written to: dak-operator-chain.pem vendor device auth key certificate chain written to: dak-vendor-chain.pem glad auth key certificate written to: glad-auth-key-device-cert.pem		

10.1.7.3 Retrieving the Audit Log (system-get-audit-log)

This command retrieves the audit log with all available entries.

Command		
<code>gladm -u <username> -k <credentials> -d <addr> system-get-audit-log [-f <val>] [--trim]</code>		

Argument	Description	Required
<code>-f</code> <code>--filename</code>	File name of the audit log to be written If no file name is specified, the file will be saved as <code>system_audit_log</code> .	0
<code>--trim</code>	Trim system audit log once received	0

Example

The following example retrieves the audit log and saves it in trimmed version as `auditlog`.

```
gladm system-get-audit-log -f auditlog --trim
```

Return

Upon successful retrieving of the audit log, gladm returns the name of the saved audit log file.

```
Retrieving system audit log...
```

```
System audit log contents written to: auditlog
```

```
Last hash:
```

```
4569a00134fcef9193d85df66cd6d242fc82538cd1f8efd16261bff424488093
```

10.1.7.4 Displaying Device System Information (system-get-info)

This command returns different values of the device system information, like the state of the device system, present and occurred alarms and versions of the active software components. It does not include information about the device host. The information can be expected to remain unchanged between calls, unless the system on the device is updated or boots into recovery.



No user authentication is needed to perform this command.

Command

```
gladm -d <addr> system-get-info
```

Argument**Description****Required**

No arguments are given for input.

Return**Example**

Device system version	6.0.0
Sensory Controller software version	3.02.0.8
Hardware revision number	7.03.0.3
UID	7300001d44f1b908

Return	Example
Serial Number	CS800359
Device Type	u.trust Anchor CSAR Standard
Alarms present, if applicable	zeroization event(s) / alarm occurred
Presence or absence of vendor secret on the device	Vendor Secret is present on the device
Presence or absence of DAK certificate on the device	Vendor DAK Certificate is present on the device



In case of a failure, an error code is returned.



If the device version number contains the suffix `-c`, then the device has loaded a FIPS firmware image.

10.1.7.5 Displaying License Information (system-get-license-info)

This command displays details about the product license.

Command
<code>gladm -d <addr> -u <user> -k <credentials> system-get-license-info</code>

Argument	Description	Required
No arguments are given for input.		

Example
<code>gladm -d 123.123.123.123 -u user01 -k my_key_priv.pem system-get-license-info</code>

Return

```
License file version: 3
Product name: u.trust Anchor CSAR Premium
Serial number: CS999999
Number of cHSMs: 31
Number of CBKs: 31
Included templates in license:
SecurityServer
SecurityServer-SDK
```




In case of a failure, an error code is returned.

10.1.7.6 Reading the System Log (system-fetch-log)

This command retrieves the system log data and stores it either in the specified location or in the default `daemon.log`.

Command

```
gladm -d <addr> system-fetch-log [-s <val>] [-f <val>]
```

Argument	Description	Required
-s --size=<val>	The maximum size of log data to retrieve in byte If no maximum size is given, all log data is retrieved.	0
-f --filename=<val>	The filename of the log to be written If no filename is given, the log is saved to <code>daemon.log</code> . <div>  If the given filename already exists, the existing file will be overwritten upon execution of the command without further warning </div>	0

Example
The following example retrieves the system log data with a maximum size of 300 bytes and stores it in the file <code>logdata.log</code> : <code>gladm system-fetch-log -s 300 -f logdata.log</code>

Return
Upon successful retrieval of the log, gladm returns the log destination. In case of a failure, an error code is returned.

10.1.7.7 Getting Device Metrics (system-get-metrics)

This command returns the metrics of the device system.

Command
<code>gladm -d <addr> system-get-metrics</code>

Argument	Description	Required
No arguments are given for input.		

Example
<code>gladm -d -PCI:0 system-get-metrics</code>

Return

Upon successful execution of the command, gladm returns the following device system metrics:

Metric	Description
<code>csar_service_up{service="glad"}</code>	Indicates whether the glad service is running: 0: No 1: Yes
<code>csar_system_memory_total_bytes</code>	The total memory of the system in bytes.
<code>csar_system_memory_free</code>	The free memory of the system in bytes. Free memory is unallocated memory that is immediately available.
<code>csar_system_memory_available</code>	The available memory of the system in bytes. Available memory is the free memory and the previously allocated memory that can be used for new processes.
<code>csar_system_memory_cached</code>	The cached memory of the system in bytes. Cached memory is currently used for accessing temporarily stored memory.
<code>csar_system_memory_active</code>	The active memory of the system in bytes. Active memory is currently occupied by data.
<code>csar_boot_time_seconds</code>	Gives the time at which the system booted, in seconds since the Unix epoch.
<code>csar_service_up{service="syssvc"}</code>	Indicates whether the syssvc service is running: 0: No 1: Yes
<code>csar_service_up{service="updater"}</code>	Indicates whether the updater service is running: 0: No 1: Yes
<code>csar_hwmon_temp_celsius{sensor="processor"}</code>	The temperature of the device measured via the processor sensor.
<code>csar_system_disk_total_bytes</code>	The total space on the system disk in bytes.
<code>csar_system_disk_free_bytes</code>	The free space on the system disk in bytes. Free disk space is unallocated space that is immediately available.
<code>csar_system_disk_avail_bytes</code>	The available space on the system disk in bytes. Available disk space is the free disk space and the previously allocated disk space that can be used for new processes.
<code>csar_crypto_accelerator_operations_xr9200</code>	The number of crypto accelerator operations on the xr9200 chip.
<code>csar_crypto_accelerator_operations_silex</code>	The number of crypto accelerator operations on the silex core.
<code>csar_crypto_accelerator_operations_total</code>	The total number of crypto accelerator operations.
<code>csar_crypto_accelerator_operations_total{slot="x"}</code>	The number of crypto accelerator operations for the cHSM in slot x.

Return	
Metric	Description
<code>csvar_hwmon_temp_celsius{sensor="accelerator"}</code>	The temperature of the device measured via the accelerator sensor.
<code>csar_service_up{service="crypto_accelerator"}</code>	Indicates whether the crypto accelerator service is running; 0: No 1: Yes
<code>csar_chsm_rng_operations_total{slot="x"}</code>	The number of rng operations in slot x.
<code>csar_trng_status{oscillator="x"}</code>	Indicates whether the oscillator x is running.
<code>csar_trng_operations{oscillator="x"}</code>	The total number of operations of the True Random Number Generator using oscillator x.
<code>csar_trng_operations_total</code>	The total number of operations of the True Random Number Generator.
<code>csar_service_up{service="random_generator"}</code>	Indicates whether the random generator service is running; 0: No 1: Yes
<code>csar_chsm_smem_pages_used{slot="x"}</code>	The number of secure memory pages used by the chsm in slot x.
<code>csar_smem_free_pages</code>	The total number of free pages of secure memory.
<code>csar_mmc_pre_eol_info{state="normal"}</code>	The lifetime by average reserved blocks: 1: "normal" 2: "warning" (80% of reserved block consumed) 3: "urgent" (90% consumed)
<code>csar_mmc_slc_device_lifetime</code>	Percentage of estimated lifetime, single level cell: 1: <10% 2: 10-20% 3: 20-30%, 4: 30-40% 5: 40-50% 6: 50-60% 7: 60-70% 8: 70-80% 9: 80-90% 10: 90-100% 11: Exceeds estimated lifetime

Return	
Metric	Description
<code>csar_mmc_mlc_device_lifetime</code>	The percentage of estimated lifetime, multi level cell: 1: <10% 2: 10-20% 3: 20-30%, 4: 30-40% 5: 40-50% 6: 50-60% 7: 60-70% 8: 70-80% 9: 80-90% 10: 90-100% 11: Exceeds estimated lifetime
<code>csar_main_battery</code>	The voltage of the main battery.
<code>csar_ext_battery</code>	The voltage of the external battery.
<code>csar_hwmon_temp_celsius{sensor="sensory_controller"}</code>	The temperature of the sensory controller (msp) in Celsius .
<code>csar_msp_alarm_present</code>	Indicates whether the sensory controller (msp) has an active alarm.
<code>csar_zeroization_event_occurred</code>	Indicates whether any alarm or clear was not yet reset.
<code>audit_log_usage_ratio</code>	Displays the fill status of the audit log in a range of 0 (empty) to 1 (full). Example for audit log filled for 0.03%: <code>audit_log_usage_ratio 0.000306248664855957</code>
<code>csar_service_up\{service="container_manager"}</code>	Indicates whether the container manager service is running.
<code>csar_chsm_cpu_seconds_total\{mode="system",slot="x"}</code>	CPUu time of the chsm in slot x spent in kernel space.
<code>csar_chsm_cpu_seconds_total\{mode="user",slot="x"}</code>	CPUu time of the chsm in slot x spent in user space.
<code>csar_chsm_memory_bytes\{slot="x",type="anon"}</code> <code>type="file"</code> <code>type="slab"</code>	The memory of the specified type of the chsm in slot x in bytes.
<code>csar_chsm_disk_total_bytes\{slot="x"}</code>	The total disc space of the chsm in slot x in bytes.
<code>csar_chsm_disk_free_bytes\{slot="x"}</code>	The free disc space of the chsm in slot x in bytes.
<code>csar_chsm_disk_avail_bytes\{slot="x"}</code>	The available disc space of the chsm in slot x in bytes.
<code>csar_chsm_user_count\{slot="x"}</code>	The number of users added to the chsm in slot x.

Return	
Metric	Description
<code>csar_chsm_session_count\{slot="x"}</code>	The number of open sessions of the chsm in slot x.
<code>csar_chsm_operation_count\{slot="x"}</code>	The number of operations of the chsm in slot x.

In case of a failure, an error code is returned.

10.1.7.8 Setting the Quorum (system-set-quorum)

This command sets the quorum requirements of the device by loading them from a given file.



A full list of commands and the connected quorum requirements in the needed format can be returned via `system-get-quorum-requirements`. You can save this list in a `.cfg` file, adjust the quorum values as needed, and then set the quorum with this file given as input.

Command
<code>gladm -d <addr> system-set-quorum-requirements <config filename></code>

Argument	Description	Required
<code><config filename></code>	<p>The file holding the quorum requirements</p> <p>For each command, the command identifier (which is the function name with the hyphen replaced by an underscore) must be given along with the new quorum requirement value in the following format:</p> <p><code><command_identifier> = <value></code></p> <p>A value of <code>1</code> means that one eligible user needs to be authenticated in the active session to execute the command, a value of <code>2</code> means that two eligible users need to be authenticated, etc.</p>	R

Example
<p>The following example sets the quorum requirements based on the file <code>quorum.cfg</code>:</p> <p><code>gladm system-set-quorum-requirements quorum.cfg</code></p>

Return
Upon successful setting of the quorum requirements, gladm returns no parameters or messages. In case of a failure, an error code is returned.

10.1.7.9 Getting the Quorum (system-get-quorum-requirements)

This command returns the quorum requirements for all commands.



No user authentication is needed to perform this command.

Command

```
gladm -d <addr> system-get-quorum-requirements
```

Argument

Description

Required

No arguments are given for input.

Return

Upon successful execution, gladm returns a full list of available commands and their quorum value. In case of a failure, an error code is returned.



The list contains the entry `command25`. This command is the equivalent of `csadm GetBootLog` but from Global Administrator side. `csadm GetBootLog` returns the boot log file. The boot log file is located in memory and is not permanently saved. The content of the previous boot log file is cleared every time the operating system starts.

10.1.7.10 Setting the System Quota (system-set-quota)

This command sets the quota values for system services.

Command

```
gladm system-set-quota [-c <val>] [-t <val>]
```

Argument

Description

Required

Both arguments are optional. If no argument is given, the command will be successfully executed, but no changes will be made to the quota.

Argument	Description	Required
-c -- cpu=<val>	The scheduler weight of the CPU for system services relative to the cHSMs. The weight of the CPU for the cHSMs is 100. The weight for system services can be set to any number from a minimum of 10 up to a maximum of 1000. Example: If 110 is given for input, the total weight of the CPU will be 210, of which system services get 110 (52%) and the cHSMs get 100 (48%). If the value 0 is given, the maximum is used.	0
-t -- time=<val> x	Sets the maximum CPU time (in %) that the system services can use	0

Example

The following example sets the quota with a CPU scheduler weight of 120 and the time of the CPU being 85 percent:

```
gladm system-set-quota -c 120 -t 85
```

Return

Upon successful setting of the quota, gladm returns no parameters or messages. In case of a failure, an error code is returned.

10.1.7.11 Getting the System Quota (system-get-quota)

This command returns the quota values for system services.

Command

```
gladm system-get-quota
```

Argument	Description	Required
No argument are given for input.		

Return

Upon successful execution, the command returns the maximum amount of CPU time available to system services, along with the weight of the CPU. In case of a failure, an error code is returned.

10.1.7.12 Setting the Time (system-set-time)

This command sets the current time of the device.

Command

```
gladm -d <addr> system-set-time [-t <val>] [-l <val>] [-u <val>]
```

Argument	Description	Required
-t	The time in milliseconds since Epoch (Jan 1st 1970). If no time is given for input, the local system time is used. (<i>remains as-is</i>)	0
-l	The time in the format YYYYMMDDHHmmSS. The timezone is retrieved from the local system.	0
-u	The time in the format YYYYMMDDHHmmSS. The timezone is UTC.	0

Example

The following example sets the time to 1595931773400 milliseconds:
`gladm system-set-time -t 1595931773400`

Return

Upon successful setting of the time, gladm returns no parameters or messages. In case of a failure, an error code is returned.



When the command `gladm system-set-time` is called with no parameters, the time is set to the local system time.

10.1.7.13 Getting the Time (system-get-time)

This command gets the current time of the device.

Command

```
gladm -d <addr> system-get-time [--format=<val>]
```

Argument	Description	Required
--format=<val>	The format specifier for the output time, following C++ strftime ⁵ . If no format is specified, the default format <code>YYYY-MM-DD HH:MM:SS <timezone></code> is used.	0

⁵ <http://www.cplusplus.com/reference/ctime/strftime/>

Example

The following example requests the return of the current time of the device with the specified format `YYYY-MM-DD` :

```
gladm system-get-time --format %Y-%m-%d
```

The following example requests the return of the current time of the device with the specified format `YYYYMMDD_HHMMSS` :

```
gladm system-get-time --format %Y%m%d_%H%M%S
```

Return

Upon successful execution of the command, gladm returns the time in the specified format. In case of a failure, an error code is returned.

10.1.7.14 Getting the NTP Configuration (system-get-ntp-config)

This command gets the NTP configuration from the u.trust Anchor device and writes it into the specified file.

This command is only available if the u.trust Anchor PCIe card is mounted in a u.trust Anchor LAN.

Consider that the NTP adjustment cannot be applied before it has been enabled using the `gladm system-activate-ntp` command.

A `gladm system-clear` command and an alarm reset the NTP settings that have been set by the `gladm system-set-ntp-config` command and the `gladm system-activate-ntp` command. A `gladm system-clear` command and an alarm do not reset the settings in the `[NTPClient]` section in the `/etc/csxlan.conf` file.

Command

```
gladm -d <IP address> [-p <port>] -u <username> -k <credentials> system-get-ntp-config -f </path/to/file>
```

Argument	Description	Required
<code>-f <file></code> <code>--filename=<file></code>	Path and filename of the text file where the NTP configuration is to be stored. Typical file extension: <code>cfg</code>	R

Example

```
gladm -d 123.123.123.123 -p 4000 -u user01 -k my_key_priv.pem system-get-ntp-config -f /my/path/ntp.cfg
```

Example for the content of the file to be created:

```
[ntp]
max_delta_per_day = 30000
ntp enabled = true
max_delta_per_op = 3000
```

Return

Returns nothing, if no error occurred.

10.1.7.15 Setting the NTP Configuration (system-set-ntp-config)

This command sets the NTP configuration of the u.trust Anchor device to the values specified in a file.

This command is only available if the u.trust Anchor PCIe card is mounted in a u.trust Anchor LAN.

Consider that the NTP adjustment cannot be applied before it has been enabled using the `gladm system-activate-ntp` command.

A `gladm system-clear` command and an alarm reset the NTP settings that have been set by the `gladm system-set-ntp-config` command and the `gladm system-activate-ntp` command. A `gladm system-clear` command and an alarm do not reset the settings in the `[NTPClient]` section in the `/etc/csxlan.conf` file.

Command

```
gladm -d <IP address> [-p <port>] -u <username> -k <credentials> system-set-ntp-config </path/to/file>
```

Argument	Description	Required
</path/to/file>	<p>Path and filename of the text file where the NTP configuration is stored in the [ntp] section. Typical file extension: <code>cfg</code></p> <p>The following parameters are used:</p> <ul style="list-style-type: none"> ▪ <code>ntp_enabled</code> This parameter indicates whether NTP is activated or deactivated. Setting this parameter is equivalent to using the <code>gladm system-activate-ntp</code> command. <code>ntp_enabled</code> does not indicate whether the NTP client has been started. <ul style="list-style-type: none"> • <code>true</code> - NTP activated • <code>false</code> - NTP not activated ▪ <code>max_delta_per_op</code> - Maximum time delta in msec per operation (default: 3000) ▪ <code>max_delta_per_day</code> - Maximum time delta in msec (default: 30000) per calendar day (0h-24h) <p>In this file, comments are not supported.</p>	R

Example

```
gladm -d 123.123.123.123 -p 4000 -u user01 -k my_key_priv.pem system-set-ntp-config /my/path/ntp.cfg
```

Example for the content of the configuration file:

```
[ntp]
max delta per day = 30000
ntp_enabled = true
max_delta_per_op = 3000
```

Return

Example output (1 - true; 0 - false):

```
Reading NTP-Config file: /home/cslagent/ntp.cfg
NTP-Configuration:
ntpEnabled: 1
maxDeltaPerDay: 30000
maxDeltaPerOperation: 3000
```

Configuring Time Synchronization between the u.trust Anchor LAN and the u.trust Anchor PCIe Card

When the time between the NTP server and the u.trust Anchor LAN (host time; system time) has been synchronized, it is important to synchronize the time between the u.trust Anchor LAN and the u.trust Anchor PCIe card mounted into the u.trust Anchor LAN. If the NTP client is enabled, it verifies every `LoopTime` seconds whether the time difference between the time

on the u.trust Anchor LAN and the time on the u.trust Anchor PCIe card is greater than `Deviation` milliseconds. If this is the case, it transfers the time on the u.trust Anchor LAN to the u.trust Anchor PCIe card.

The following parameters are used for time synchronization:

- `LoopTime`

Verification time interval in seconds

Default value: `LoopTime = 3600` (i.e., once per hour)

We recommend not to change the default value. Do not set a value higher than 86400 (i.e., one day).

`LoopTime` is a parameter of the NTP client on the u.trust Anchor LAN. It is configured in the `/etc/csxlan.conf` file on the u.trust Anchor LAN. See the steps below to perform this configuration.

- `Deviation`

Time deviation in milliseconds between the u.trust Anchor LAN and the u.trust Anchor PCIe card for which the time on the u.trust Anchor PCIe card is to be corrected

Default value: `Deviation = 500`

Recommended value range: `1 - 2500`

A value below 1 is automatically set to 1, and a value higher than 2500 is automatically set to 2500.

`Deviation` is a parameter of the NTP client on the u.trust Anchor LAN. It is configured in the `/etc/csxlan.conf` file on the u.trust Anchor LAN. See the steps below to perform this configuration.

If you want to change the `LoopTime` value or the `Deviation` value, perform the following steps.

1. Log in remotely to the u.trust Anchor LAN, see *Logging in Remotely to the u.trust Anchor LAN* in the *u.trust Anchor LAN V5 - Administration Manual*.
2. Go to the `/etc` directory. Here you will find the `csxlan.conf` configuration file (`/etc/csxlan.conf`).
3. Open the `csxlan.conf` file with a text editor.
In our example, the time synchronization (`LoopTime`) should be performed every 3600 seconds (one hour), and for any time variation (`Deviation`) of more than 2500 milliseconds (2,5 seconds).
To do so, you should adjust the following entries in the `[NTPClient]` section of the

`csxlan.conf` configuration file:

```
[NTPClient]
```

```
Deviation = 2500
```

```
LoopTime = 3600
```

4. Save and close the `csxlan.conf` configuration file.
5. Make the changes in the `csxlan.conf` configuration file effective by performing the following substeps.
 - If you use the remote access to the u.trust Anchor LAN instead, perform the following substep.
 - i. Perform the following commands in exactly this order.

```
set_ntpclient_config.sh no
```

```
set_ntpclient_config.sh yes
```
 - ii. Shut down your SSH client.



Time synchronization has successfully been configured.

10.1.7.16 Activating NTP (system-activate-ntp)

This command activates or deactivates the NTP time adjustment of the u.trust Anchor device. If this NTP feature is activated (default: deactivated), it verifies after a certain time period whether the time difference between the time on the u.trust Anchor LAN and the time on the u.trust Anchor PCIe card is greater than a certain threshold. If this is the case, it transfers the time on the u.trust Anchor LAN to the u.trust Anchor PCIe card.

This command is only available if the u.trust Anchor PCIe card is mounted in a u.trust Anchor LAN.

Consider that the NTP adjustment cannot be applied before

- it has been activated by using the `gladm system-activate-ntp` command and

- the NTP client has been started on the u.trust Anchor LAN by performing the `/etc/init.d/ntpclient start` command.
The configuration of the NTP client is done in the `[NTPClient]` section in the `/etc/csxlan.conf` file.

Consider that the NTP configuration can be read by using the `gladm system-get-ntp-config` command and it can be set by using the `gladm system-set-ntp-config` command.

A `gladm system-clear` command and an alarm reset the NTP settings that have been set by the `gladm system-set-ntp-config` command and the `gladm system-activate-ntp` command. A `gladm system-clear` command and an alarm do not reset the settings in the `[NTPClient]` section in the `/etc/csxlan.conf` file.

Command		
<code>gladm -d <IP address> [-p <port>] -u <username> -k <credentials> system-activate-ntp -f <1 0></code>		
Argument	Description	Required
<code>-f <flag></code> <code>--</code> <code>flag=<flag></code>	<code>1</code> - Activate <code>0</code> - Deactivate	R
Example		
Example of activating NTP adjustment: <code>gladm -d 123.123.123.123 -p 4000 -u user01 -k my_key_priv.pem system-activate-ntp -f 1</code> Example of deactivating NTP adjustment: <code>gladm -d 123.123.123.123 -p 4000 -u user01 -k my_key_priv.pem system-activate-ntp -f 0</code>		
Return		
Returns nothing, if no error occurred.		

General Procedure

- Get the current NTP configuration from the u.trust Anchor device using the `gladm system-get-ntp-config` command.
- If necessary, change the NTP configuration using the `gladm system-set-ntp-config` command.
- Activate the (changed) NTP configuration using the `gladm system-activate-ntp` command.

4. Log in remotely to the u.trust Anchor LAN, see *Logging in Remotely to the u.trust Anchor LAN* in the *u.trust Anchor LAN V5 - Administration Manual*.
5. Verify the NTP client configuration on the u.trust Anchor LAN (`[NTPClient]` section in the `/etc/csxlan.conf` file).
6. Start the NTP client on the u.trust Anchor LAN (`set_ntpclient_config.sh yes` command).

10.1.7.17 Resetting the Alarm (system-reset-alarm)

This command resets the alarm state of the device, if the cause of the alarm is no longer present. A new SDMK and DMK are generated, if they are missing.

Some reasons for an alarm can be:

- Temperature too high/low
- Power too high/low
- Damaged sensory wire
- External erase (manual deletion)



Condition:

If the `system-alarm-reset` follows an external erase and a `system-clear`, then `system-alarm-reset` must be authenticated by the GIAK.



Command		
<code>gladm -d <addr> system-reset-alarm</code>		
Argument	Description	Required
No arguments are given for input.		
Example		
system-alarm-reset performed alone <code>gladm -d /dev/cs2.0 -u admin -k admin_gaak_enc.key system-reset-alarm</code> system-alarm-reset performed after an external erase and a system-clear <code>gladm -d /dev/cs2.0 -u admin -k /tmp/giak.pem system-reset-alarm</code>		
Return		
Upon successful resetting of the alarm state, gladm returns no parameters or messages. In case of a failure, an error code is returned.		

10.1.7.18 Restarting the Device (device-restart)

This command restarts the device.



Do not confuse the command `gladm device-restart` with the command `gladm system-restart`. Both commands restart the device.

- `gladm device-restart`

The command `gladm device-restart` can only be performed locally or via SSH. This command does not need any authentication. It reboots the Linux driver of the PCIe card, i.e., it is equivalent to performing the following command as the `root` user: `echo REBOOT > /proc/driver/cs2.0`

- `gladm system-restart`

The command `gladm system-restart` can only be performed remotely. This command needs authentication. It shuts down and reboots the Linux subsystem.

Command		
<code>gladm -d <addr> device-restart</code>		
Argument	Description	Required
No arguments are given for input.		

Return

Upon successful execution of the command, the device restarts. In case of a failure, an error code is returned.

10.1.7.19 Restarting the Device (system-restart)

This command restarts the device.



Do not confuse the command `gladm system-restart` with the command `gladm device-restart`. Both commands restart the device.

- `gladm system-restart`

The command `gladm system-restart` can only be performed remotely. This command needs authentication. It shuts down and reboots the Linux subsystem.

- `gladm device-restart`

The command `gladm device-restart` can only be performed locally or via SSH. This command does not need any authentication. It reboots the Linux driver of the PCIe card, i.e., it is equivalent to performing the following command as the `root` user: `echo REBOOT > /proc/driver/cs2.0`

Command

```
gladm -d <IP address> [-p <port>] -u <username> -k <credentials> system-restart
```

Argument**Description****Required**

No arguments are given for input.

Example

```
gladm -d 123.123.123.123 -p 4000 -u user01 -k my_key_priv.pem system-restart
```

Return

Upon successful execution of the command, the device restarts. In case of a failure, an error code is returned.

10.1.7.20 Clearing the System (system-clear)

The command clears sensitive system data. Which data was deleted depends on whether an external erase was carried out beforehand.

	Clearing the Device	Clearing to Factory Default State after performing an External Erase
Authentication by	Global Administrator	not required
GAAK	preserved	restores Global Initial Administrator with GIAK
Device Audit Log	preserved	preserved
Device Boot Log	preserved	cleared, if preceded by an external erase
DMK cHSM cHSM Audit Log cHSM Boot Log	cleared	cleared
SDMK DAK Vendor Secret	preserved	preserved
Secure RAM	unused data is zeroized, other data is preserved	cleared, if preceded by an external erase

Table 22: gladm system-clear - Comparison

Since this function does not imply a device restart, the system boot log is still present after executing the command.

After the clear, `gladm system-reset-alarm` has to be executed to regenerate the DMK.

Command		
<code>gladm -d <addr> system-clear</code>		
Argument	Description	Required
No arguments are given for this function.		
Example		
system-clear performed alone <code>gladm -d /dev/cs2.0 -u admin -k admin_gaak_enc.key system-clear</code> system-clear performed after an external erase <code>gladm -d /dev/cs2.0 system-clear</code>		
Return		
The system alarm state must be reset before the device can be used		



In case of a failure, an error code is returned.

10.1.7.21 Updating the Device Firmware (system-update)

This command updates the firmware of the device from a given firmware image.



The command will delete existing cHSMs during the update process. Use `gladm chsm-retrieve` to back-up existing cHSMs and execute `gladm chsm-restore` to restore the backups after the system update.

See also the section *Updating u.trust Anchor* in the *u.trust Anchor - Administration Manual*.

Command

```
gladm -d <addr> system-update [-f] <image filename>
```

Argument	Description	Required
<code>-f</code> <code>--force</code>	This argument skips checks that may prevent an update. In case any cHSMs are still present on the device when the command is executed, this flag will lead to their deletion upon execution of the command. If the command is executed without this flag, the command will fail if there are any cHSMs present on the device.	O
<code><image filename></code>	The filename of the device firmware image	R

Example

The following example updates the device with the `--force` option from the firmware image `firmware.raucb`:

```
gladm -u admin -k gaak.pem system-update -f firmware.raucb
```

Return

Upon successful update of the device, gladm returns no parameters or messages. In case of a failure, an error code is returned.

10.1.7.22 Emitting the Bash Completion Script (bash-completion)

This command emits the bash completion script.

Move the returned file to a standard location (e.g. `/etc/bash_completion.d`) to enable gladm completion systemwide, or it can be sourced via `.bashrc` to enable it for a single user.



The `gladm bash-completion` command does not need the `-d` parameter (device specifier/address).

Command

```
gladm bash-completion
```



By entering `gladm bash-completion > gladm_bash_completion`, the script is automatically saved to a named file which can then be moved to a standard directory as described above.

Argument	Description	Required
No arguments are given for input.		

Return

Upon successful execution, gladm returns the bash completion script. In case of a failure, an error code is returned.