

i2km

KeyMirror

1.1

Integration Guide

u.trust GP HSM

6.5.0.0

Imprint

Copyright 2026	Utimaco IS GmbH Krefelder Straße 220 52070 Aachen Germany
Phone	AMERICAS +1-844-UTIMACO (+1 844-884-6226) EMEA +49 800-627-3081 APAC +81 800-919-1301
Internet	https://support.hsm.utimaco.com/
e-mail	support@utimaco.com
Document Version	1.0.0
Date	2026-10-16
Status	PUBLISHED
Document No.	IG-2026-0099
All rights reserved	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. 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. All trademarks and registered trademarks are the property of their respective owners.

Table of Contents

1	About This Guide	5
1.1	Target Audience	5
1.2	Purpose of the Integration	5
1.3	Abbreviations	5
1.4	Document Conventions	6
2	Product Overview	8
2.1	Overview of KeyMirror	8
2.2	Overview of Utimaco u.trust GP HSM Se-Series.....	8
2.3	Joint Value Proposition.....	8
3	Integration Requirements and Prerequisites	9
3.1	Tested Versions.....	9
3.2	Hardware and Software Requirements.....	9
3.3	Prerequisites	10
4	Installing and Configuring u.trust GP HSM.....	11
5	Installing and Configuring KeyMirror	13
5.1	Installing the KeyMirror Agent.....	13
5.2	Installing the OpenTelemetry receiver plugin	13
6	Integration Steps on Utimaco u.trust GP HSM	14
7	Integration Steps on KeyMirror	15
7.1	BPF mode configuration.....	15
7.2	Shim mode configuration	15
7.3	Viewing the Metrics and Tracing	15
8	Verification and Testing	16
8.1	Viewing Metrics and Telemetry in Grafana.....	16
8.1.1	Overview	16
8.1.2	Topology.....	16
8.1.3	Application Errors.....	17
8.1.4	Key Usages.....	18
8.1.5	Client Applications	18
8.1.6	Session Traces	18
8.1.7	Operations.....	19

8.1.8 Quantum Readiness 20

8.1.9 Client Applications 20

9 Contact and Support Information.....22

10 Appendices 23

10.1 OpenTelemetry Details 23

1 About This Guide

This guide describes the integration of KeyMirror with the Utimaco u.trust GP HSM. It provides an overview of the installation, configuration, and validation procedures required to enable telemetry and observability for PKCS#11-based applications using the Utimaco GP HSM.

1.1 Target Audience

This guide is intended for system administrators and security professionals responsible for deploying and managing KeyMirror with the Utimaco u.trust GP HSM. It assumes a basic understanding of PKCS#11, HSMs, Linux systems, and observability technologies.

1.2 Purpose of the Integration

The purpose of this integration is to enable KeyMirror to provide telemetry, metrics, and tracing for applications using the Utimaco u.trust GP HSM. By collecting PKCS#11 operation metadata and exporting it through OpenTelemetry, the integration provides enhanced observability, performance monitoring, and troubleshooting capabilities for cryptographic operations without requiring changes to the client applications.

1.3 Abbreviations

Abbreviation	Meaning
HSM	Hardware Security Module
PKI	Public Key Infrastructure
TDE	Transparent Data Encryption
PKCS	Public Key Cryptography Standards
PKCS#11	PKCS Part 11: The Cryptographic Token Interface Standard

Abbreviation	Meaning
SO	The PKCS#11 cryptographic slot Security Officer

Table 1: Abbreviations

1.4 Document Conventions

The following conventions are used in this guide:

Convention	Use	Example
Bold	Items of the Graphical User Interface (GUI), e.g., menu options	Press 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 2: Document conventions

We use special icons to highlight the most important notes and information.



Here you will find important safety information that should be followed.



Here you will find additional notes or supplementary information.



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

2 Product Overview

2.1 Overview of KeyMirror

KeyMirror is an observability platform that provides telemetry, metrics, and tracing for applications using PKCS#11 libraries. KeyMirror integrates between client applications and the Utimaco PKCS#11 library, collecting operational metadata such as function calls, latency metrics, and execution outcomes without requiring modifications to the monitored applications. The collected data is converted into OpenTelemetry-compatible formats and can be exported to standard observability platforms for monitoring, analysis, and troubleshooting.

2.2 Overview of Utimaco u.trust GP HSM Se-Series

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

2.3 Joint Value Proposition

The integration of KeyMirror with the Utimaco u.trust GP HSM provides enhanced visibility into cryptographic operations by combining hardware-backed key protection with modern observability capabilities. KeyMirror collects telemetry, metrics, and tracing data from applications using the Utimaco PKCS#11 library and exports the information to OpenTelemetry-compatible monitoring platforms. This enables organizations to monitor cryptographic workloads, analyze performance, troubleshoot issues, and gain operational insights without requiring changes to existing applications.

3 Integration Requirements and Prerequisites

3.1 Tested Versions

Operating System	KeyMirror Version	Utimaco SecurityServer Version	Utimaco HSM
Linux (Kernel $\geq$ 5.8 for BPF mode, Kernel $\geq$ 4.0 for Shim mode)	1.1	6.5.0.0	u.trust GP HSM CSe-Series/Se-Series

Table 3: Tested versions

3.2 Hardware and Software Requirements

Hardware	Hardware Requirements
Utimaco LAN HSM	u.trust GP HSM Se-Series LAN with firmware SecurityServer 6.5.0 or higher
Utimaco PCI-e HSM	u.trust GP HSM Se-Series PCI-e with firmware SecurityServer 6.5.0 or higher

Table 4: List of hardware requirements

Software	Software Requirements
HSM Interface	SecurityServer PKCS#11 Provider
HSM Utility	SecurityServer PKCS#11 Tool (p11tool2)
KeyMirror	KeyMirror Client Agent version 1.1

Software	Software Requirements
Operating System	Linux with kernel version 5.8 or later for BPF mode, or kernel version 4.0 or later for Shim Mode

Table 5: List of software requirements

3.3 Prerequisites

Before you begin, please make sure that you have:

- Installed/set up the operating system listed in [Tested Versions](#).
- Installed/set up the u.trust GP HSM listed in [Tested Versions](#).
- Replaced the default HSM administrator account with a new administrator account.
- Set up and configured the u.trust GP HSM. See the u.trust GP HSM documentation for information about setting up the HSM.
- Installed and configured the PKCS#11 library according to your environment.
- A valid KeyMirror license.
- Installed an OpenTelemetry Collector is ensured it is accessible from the KeyMirror client system.
- Ensured the required network connectivity is available between the KeyMirror client agent and the OpenTelemetry Collector.
- Ensured that root or appropriate administrative privileges are available to install and configure the KeyMirror client agent.
- Identified the required KeyMirror operating mode:
 - **BPF mode** requires a supported Linux kernel version of 5.8 or later.
 - **Shim mode** requires a supported Linux kernel version of 4.0 or later.

4 Installing and Configuring u.trust GP HSM

The following section outlines the procedures required to configure the Utimaco SecurityServer Software.

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

1. Copy the downloaded software to the appropriate location on the KeyMirror client system.
2. Create a utimaco folder under the `/opt` directory and create two directories `/opt/utimaco/bin` and `/opt/utimaco/lib`.

```
# mkdir -p /opt/utimaco/bin
# mkdir /opt/utimaco/lib
```

3. Copy the pkcs11 library file `libcs_pkcs11_R3.so` from the Utimaco CryptoServer software to the `/opt/utimaco/lib` directory and make the file executable.

```
cp ~/path_to_application_folder/lib/libcs_pkcs11_R3.so /opt/utimaco/lib
chmod +x /opt/utimaco/lib/libcs_pkcs11_R3.so
```

4. Copy the csadm and p11tool2 files from the Utimaco CryptoServer software to the `/opt/utimaco/bin` directory and make both files executable.

```
# cd ~/path_to_application_folder
# cp csadm p11tool2 /opt/utimaco/bin
# chmod +x /opt/utimaco/bin/csadm /opt/utimaco/bin/p11tool2
```

5. Create the directory `/etc/utimaco`. Locate the Utimaco PKCS#11 configuration file in your SecurityServer directory, `Software\Linux\Crypto_APIs\PKCS11_R3\sample`. Copy the Utimaco PKCS#11 configuration file `cs_pkcs11_R3.cfg` to the `/etc/utimaco` directory.

```
# mkdir /etc/utimaco
# cd ~/path_to_application_folder/Software/Linux/Crypto_APIs/PKCS11_R3/sample
```

```
# cp cs_pkcs11_R3.cfg /etc/utimaco
```

Edit the `cs_pkcs11_R3.cfg` file and make the appropriate changes to the file.

```
[Global]
# For unix:
Logpath = /etc/utimaco
# Loglevel (0 = NONE; 1 = ERROR; 2 = WARNING; 3 = INFO; 4 = TRACE)
Logging = 3
Keepalive = true
# Set the Device to connect with
[CryptoServer]
# Device specifier
Devices = <HSM_IP>
```



For more information regarding the commands and command parameters, please see the Utimaco u.trust GP HSM documentation. The device may be a PCIe or LAN device. The device line will follow one of these patterns, based on the HSM formfactor: Device = 288@<HSM IP address> Hardware (LAN) HSM or Device = /dev/cs2.0 Hardware (PCIe) HSM



To simplify your testing process, it's recommended that you enable the PKCS#11 log file by adjusting the logging settings. Specifically:

- Set the **LogPath** to a writable directory (not a specific file).
- Set the **Logging** log level to 1 for basic logging. Increase it to 4 for more detailed output during testing.

This will generate a log file named `cs_pkcs11_R3.log` within the specified **LogPath** directory. Reviewing this log can help with troubleshooting if you encounter issues. Once testing is complete, it's advisable to reduce **Logging** log level to 1 or 2 to limit output to only critical or important messages.

5 Installing and Configuring KeyMirror

Installing KeyMirror agents and monitoring in an estate comprises two essential stages and an optional stage:

1. Agent install: On each client machine, install a KeyMirror client agent (e.g. installation of `keymirror.deb` in Debian-based distributions)
2. Configure and License KeyMirror's OpenTelemetry Collector Plugin: OpenTelemetry traffic from KeyMirror client agents is encrypted and OpenTelemetry collectors require a custom receiver to decrypt the traffic from KeyMirror client agents. Installing KeyMirror's OTel receiver plugin allows this traffic to be decrypted when a valid license is applied.
3. Optional: install the KeyMirror Grafana App: For ease of use, we publish a Grafana plugin app which allows enhanced viewing, alerting, etc. of KeyMirror telemetry via Grafana.

5.1 Installing the KeyMirror Agent

KeyMirror client agent packages for your architecture can be downloaded from the KeyMirror client portal (<https://portal.i2km.com>). Transfer the package (e.g. `keymirror.deb`) to the target machine and install (e.g. with `sudo dpkg -i keymirror.deb`). If further configuration changes are required (beyond the defaults configured in the client portal), edit the agent configuration file at `/etc/keymirror/config.json` and restart the agent.



The client portal allows configuration options to be specified as defaults, e.g. your OpenTelemetry collector URL, libraries to monitor, etc., removing the need to manually configure these on each agent install.

5.2 Installing the OpenTelemetry receiver plugin

Download the OpenTelemetry receiver plugin from the KeyMirror client portal and follow the instructions on the portal to apply your license and then add the licensed receiver to your OpenTelemetry collector.



A valid OpenTelemetry endpoint must be specified. It is recommended that an IP address is used to avoid potential DNS issues. Consult the KeyMirror Technical Manual for further details on configuration options including mutual TLS.

6 Integration Steps on Utimaco u.trust GP HSM

The following section outlines the procedures required to configure both the Utimaco HSM and KeyMirror client system components for seamless integration.

Create users SO (Security Officer) and USR (the Crypto user) and initialize a slot.

The slot must be initialized using the **p11tool2**.

First, create the **SO** using *p11tool2*. Then, using the *p11tool2* command, initialize the Slot you want to use and the slot user, as shown below.

- `# ./p11tool2 slot=<slot no.> Label=<token label> Login=ADMIN,ADMIN.key
InitToken=<SO pin>`

Initialize the SO user.

- `# ./p11tool2 slot=<slot no.> LoginSO=<SO pin> InitPin=<Cryptouser pin>`

Ensure the Utimaco GP HSM is accessible from the KeyMirror client system.

7 Integration Steps on KeyMirror

Once a KeyMirror client agent has been installed and configured, integrating it with HSM application is a single-step operation. Following configuration as below, the client agent will gather metadata from the application at runtime, process it and transfer to your Open- Telemetry collector.

7.1 BPF mode configuration

In BPF mode, edit `/etc/keymirror/config.json` to reflect the application/library path(s) you wish to monitor, followed by restarting the KeyMirror client agent.

For example, to instrument `libcs_pkcs11`, the configuration file would be edited to contain:

```
"files_to_hook": ["/opt/utimaco/lib/libcs_pkcs11_R3.so"]
```

7.2 Shim mode configuration

In shim mode, launch your application with the environment variable:

```
LD_PRELOAD=/usr/local/lib/liblshim.so
```

7.3 Viewing the Metrics and Tracing

After this has been carried out, the KeyMirror client agent will collect metadata from your application/library at run time, process it, then transfer it to the configured OpenTelemetry receiver. The resulting traces and metrics can then be viewed in Grafana, Datadog etc. For ease of use, we provide a Grafana plugin application, along with example Grafana dashboard files, highlighting the main metrics and tracing behavior.

8 Verification and Testing

8.1 Viewing Metrics and Telemetry in Grafana

This section covers example screenshots from the KeyMirror Grafana plugin app, along with an example Grafana dashboard.

8.1.1 Overview

The overview section displays information regarding all PKCS#11 sessions estate-wide including their number and activity related to each session. Various indicators such as a heatmap of session activity and display of sessions by activity are displayed. Idle and abandoned sessions are also shown, along with alerts relating to failed token logins estate-wide.

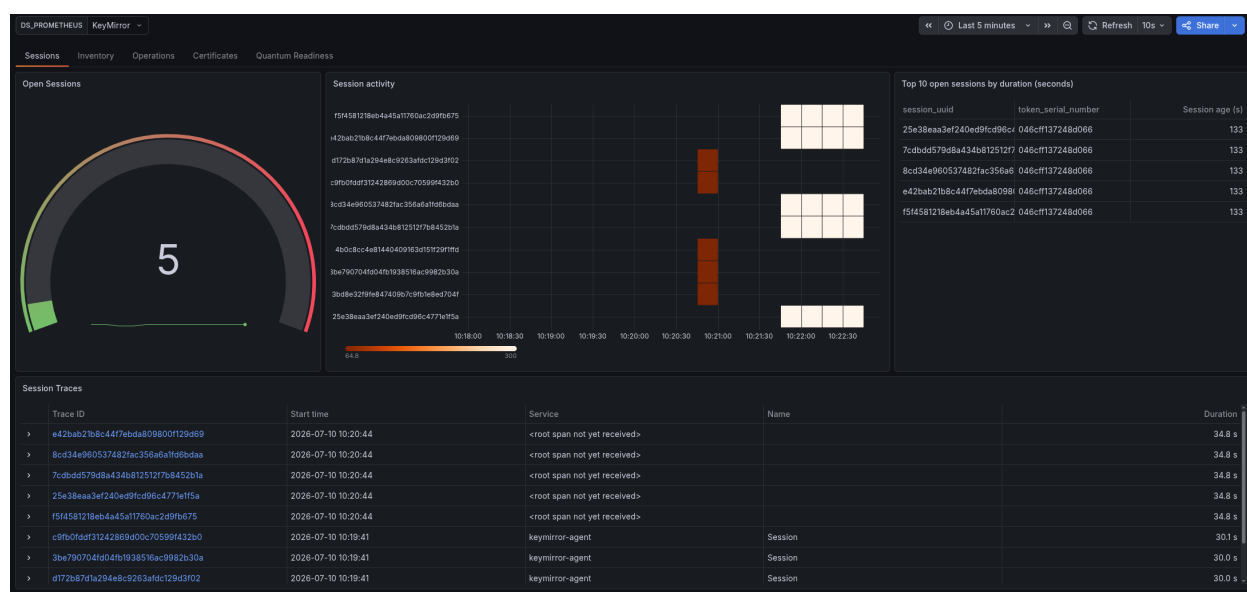


Figure 1 : Dashboard - Sessions

8.1.2 Topology

The topology section of the Grafana example dashboard displays an estate-wide graph capturing the relationships, metrics volume and error rates relating to KeyMirror client agents and HSM tokens.

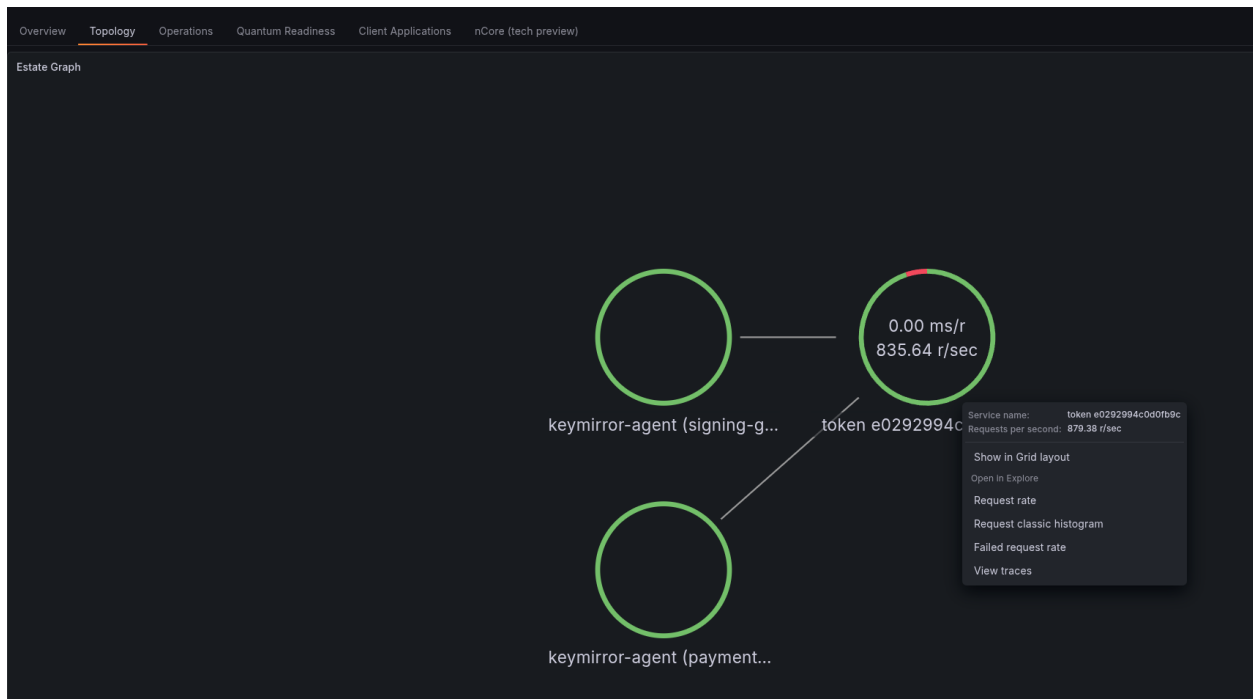


Figure 2 : Dashboard - Topology

8.1.3 Application Errors

The application errors section of the Grafana app presents listing of all application errors which have occurred estate-wide. It categorizes these by session and application, displaying the IP address and binary name, along with the number and type of errors. Clicking the "trace" button for a session will navigate to a trace of the session, allowing the user to view the sequence of events which led to the error(s).

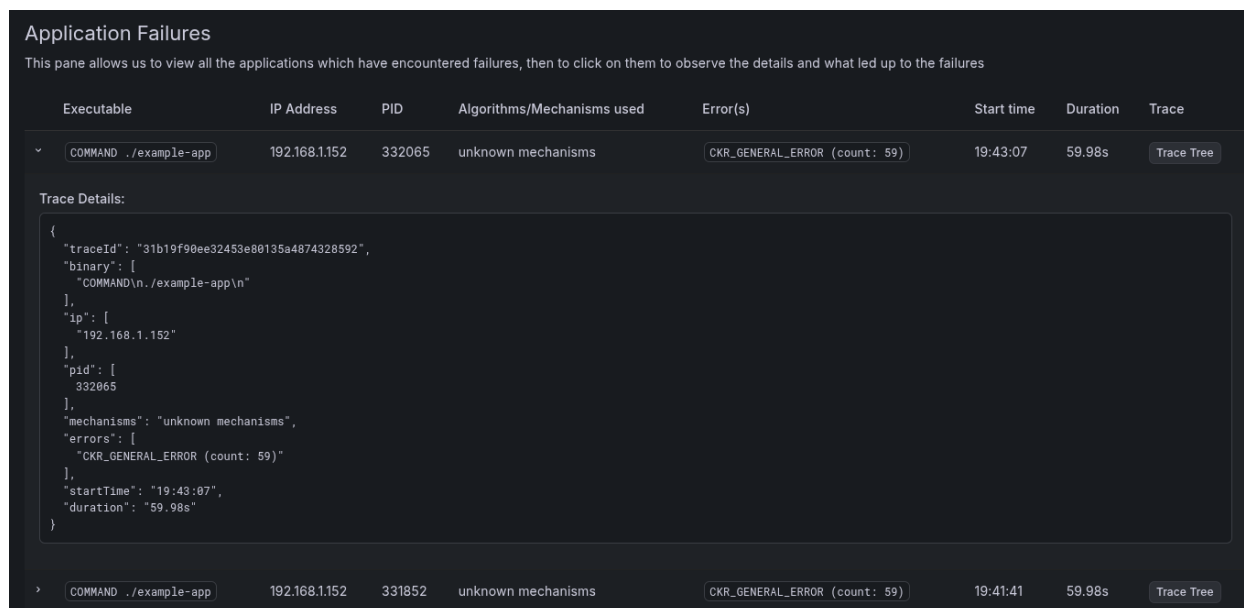


Figure 3 : Dashboard - Session Failures

8.1.4 Key Usages

The key usages section of the Grafana app presents listings of all keys used estate-wide in a given timeframe, allowing ranking by usage frequency, latency and algorithm. It allows easy determination of the top-X keys used estate-wide, with the details button providing breakdowns of all applications and client machines accessing a given key, along with providing tracing links to the relevant sessions.

8.1.5 Client Applications

The client applications of the Grafana plugin app and example dashboard allows examination of real-time memory and CPU usage by PKCS#11 applications on client machines, allowing early detection of potential issues and attribution in the event of client failures.

8.1.6 Session Traces

To examine a particular session, click the relevant link in the Session Traces pane. This will display session details and a detailed session trace from the bundled Grafana Tempo service. By expanding the root session trace, information such as the client binary path, client IP address, process ID and operating system can be displayed.

Individual cryptographic operations can also be examined, including their latency and outcome. For example, the following screenshot displays a session trace featuring a number of failed AES-GCM decryption operations.

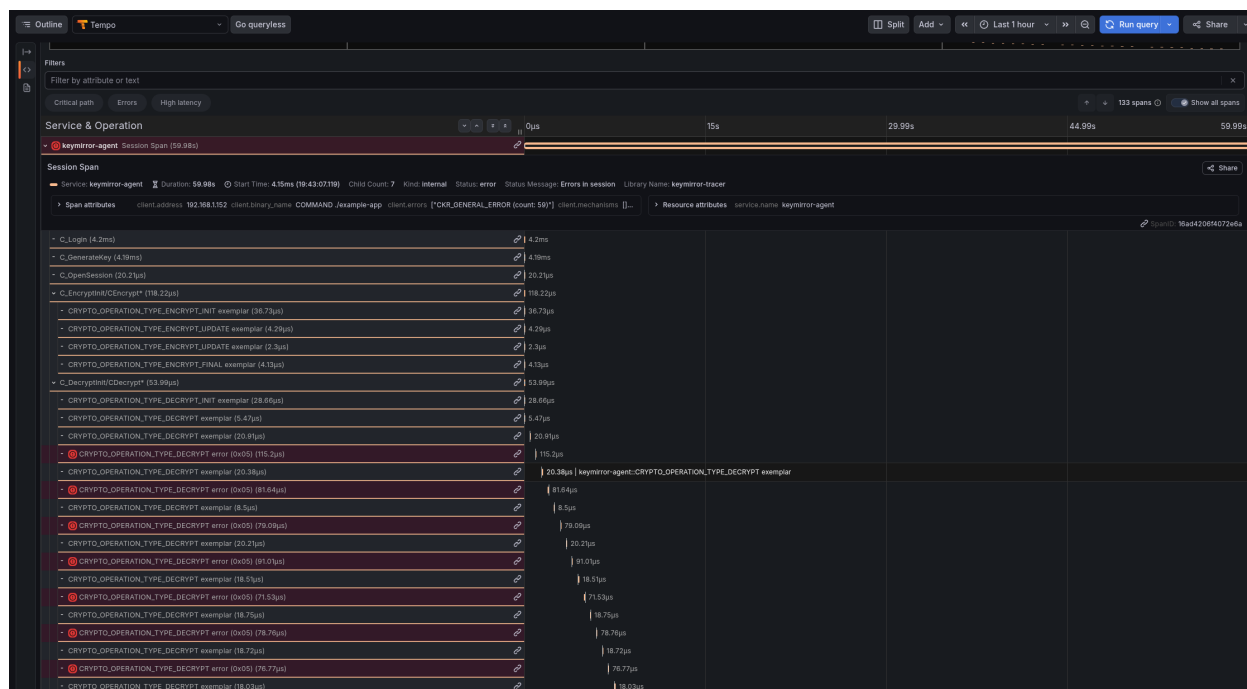


Figure 4 : Grafana dashboard - example session trace

8.1.7 Operations

The Operations tab contains details about the operations, including algorithms in use, keys in use, instantaneous operations rates per second (success and failure), total operations in the time range, a heatmap of key usage by type and a latency histogram.

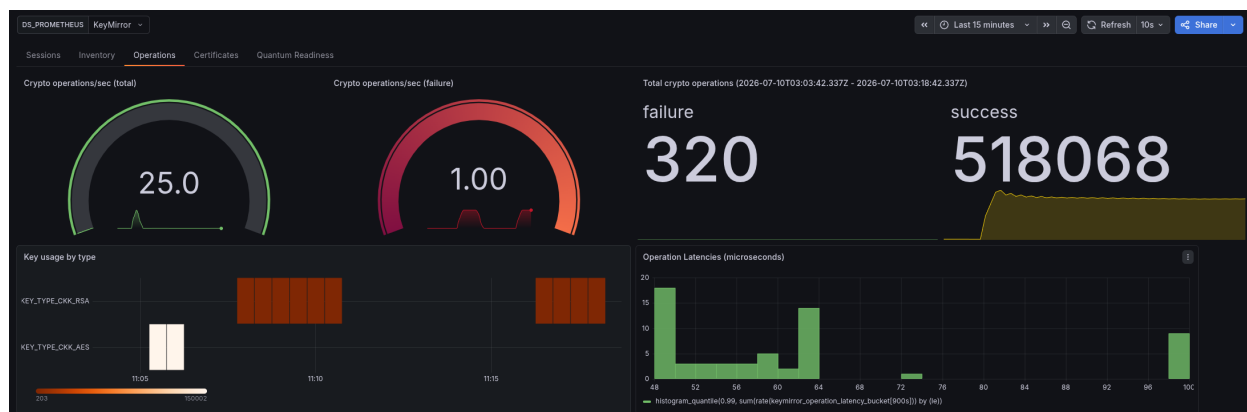


Figure 5 : Dashboard - Operations

8.1.8 Quantum Readiness

The Quantum Readiness tab contains basic details regarding the number of quantum-vulnerable keys generated and used in the time range, allowing basic tracking of quantum posture over time.

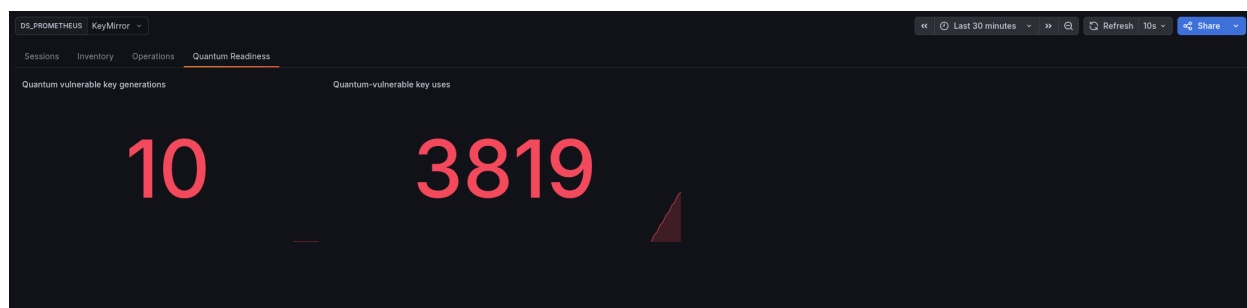


Figure 6 : Dashboard - Quantum Readiness

8.1.9 Client Applications

The Client Applications tab contains basic details regarding the applications using the PKCS#11 libraries, including monitoring the memory consumption of each process. The underlying metrics allow detection of memory leaks and alarms to be raised.



Figure 7 : Dashboard - Client Applications (memory leak detection)

9 Contact and Support Information

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

Utimaco IS GmbH
Krefelder Straße 220
52070 Aachen
Germany

RMA Query

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

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

Other Support Queries

- Mail (preferred contact method)
support@utimaco.com
Attach the diagnostic information to your email.
- Web portal
<https://support.hsm.utimaco.com/support/cases/new/>
The diagnostic information will be requested in our response if necessary.
- By phone
AMERICAS +1-844-UTIMACO (+1 844-884-6226)
EMEA +49 800-627-3081
APAC +81 800-919-1301
The diagnostic information will be requested in our response if necessary.

10 Appendices

10.1 OpenTelemetry Details

The following is a selection of the OpenTelemetry metrics which KeyMirror produces.

OpenTelemetry Metric List	
Name	Description
keymirror_algorithm_usages	Covers the number of algorithm usages by HSM token and client application
keymirror_latencies	Histograms covering crypto operation latencies by HSM token, session and algorithm
keymirror_open_sessions	Covers the number of open sessions by HSM token and client application IP address
keymirror_abandoned_sessions	Covers the number of abandoned sessions by HSM token and client application IP address
keymirror_idle_sessions	Covers the number of idle sessions by HSM token and client application IP address
keymirror_closed_sessions	Covers the number of closed sessions by HSM token and client application IP address
keymirror_key_generations	Covers the number of key generations by HSM token, client application IP address, session and algorithm
keymirror_key_destructions	Covers the number of key destructions by HSM token, client application IP address, session and algorithm

OpenTelemetry Metric List	
keymirror_logins	Covers the number of session logins by HSM token, client application IP address and result, allowing failed logins to be monitored and attributed
keymirror_logouts	Covers the number of session logouts by HSM token and client IP address
keymirror_crypto_operations	Covers the number of cryptographic operations by token, session and algorithm
keymirror_session_ages	Covers the age of open sessions by HSM token, session and client application
keymirror_open_find_operations	Covers the number of open find operations by HSM token, session and client application, allowing misconfigured applications to be detected
keymirror_client_memory	Covers the memory consumption of monitored client applications, allowing centralized detection of client memory leaks
keymirror_client_cpu	Covers the CPU utilization of monitored client applications, allowing attribution of bottlenecks
keymirror_application_errors	Covers the rate of client application errors by client application

Table 6: OpenTelemetry metrics