Apache

HTTP Server

v2.4 and higher

Integration Guide

CryptoServer

v4.5 and higher

utimaco®

Imprint

Copyright 2020	Utimaco IS GmbH
	Germanusstr. 4
	D-52080 Aachen
	Germany
Phone	AMERICAS +1-844-UTIMACO (+1 844-884-6226)
	EMEA +49 800-627-3081
	APAC +81 800-919-1301
Internet	https://support.hsm.utimaco.com/
e-mail	support@utimaco.com
Document Version	1.0.0
Date	06/10/2025
Status	**PUBLISHED**
Document No.	IG-2025-0001

Table of Contents

1 Introduction

Thank you for purchasing our CryptoServer security system. We hope you are satisfied with our product. Do not hesitate to contact us if you have any complaints or comments.

1.1 About This Manual

This manual provides guidance to secure an Apache Web Server private key, using an

Utimaco HSM. By moving the private key used by the Apache Web Server for https inside a HSM and running the https cryptographic operation with that HSM key, a hacker cannot get the private key, even if the hacker gets root access to the server where Apache is hosted.

1.1.1 Target Audience for This Manual

This manual is intended for the system administrators running an Apache Web Server with https mode and wanting to secure the private key of the Apache Web Server.

1.1.2 Document Conventions

We use the following document conventions:

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

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.

Certifications

Chapters with certification-specific content are marked accordingly at the beginning of the chapter, e.g. **FIPS 140-3**.

1.1.3 Abbreviations

We use the following abbreviations in this manual:

Abbreviation	***Meaning***
HSM	Hardware Security Module
PKI	Public Key Infrastructure
SSL	Secure Sockets Layer
URI	Uniform Resource Identifier
TCP	Transmission Control Protocol

2 Overview

The CryptoServer is the hardware security module (HSM) of Utimaco IS GmbH. Developed as a specialized physical-only processing unit, it performs sensitive cryptographic functions and ensures safe management of cryptographic keys and data. In a CryptoServer safety system, safety-relevant action is taken and security-related information is stored. It can be used as a universal, independent security component for heterogeneous computing systems.

The current use of inadequate transport layer security enables easy interception of communications of a web application or web server to a web browser from untrusted third parties. If an unencrypted transport layer - such as HTTP - is used for the transport of business-critical information, it can be easily compromised or intercepted. In modern web applications, SSL/TLS connections are used to secure the HTTP communications. In this case, the communication is encrypted with the help of symmetric cryptographic keys, which had previously been negotiated by an asymmetric cryptographic key exchange process. Foundations of these methods are always the public key of the participating parties or a proof of identity in the form of certificates. In addition to the storage of public keys, cryptographic certificates include normally more information to help identify the owner or a trusted machine. Public internet web servers usually use certificates that are issued by a trusted certificate authority, to prove their trusted identity.

The Apache Web Server is a web server such as Microsoft IIS and provides the web pages of a web site to the clients. It has the ability to transport information using encryption, in the form of the HTTPS protocol.

This document describes the essential steps to establish an SSL connection with the Apache Web Server based on a CryptoServer as a certificate store. Usually, the Apache Web Server is installed with a file-based certificate and private key in context of SSL. Even if this private key is protected by a password, potential attackers can for example simply copy the file to a portable medium or over a network and this can lead to a well taken serious identity theft. A hardware security module (HSM) as the CryptoServer ensures that a logical and physical access to the private key is only possible to trusted individuals or applications. Copying the private key and sensitive information is not possible when using the CryptoServer for key storage.

3 Supported Systems

System	***Version***
Linux	> 3
Apache Web Server	> 2.4
OpenSSL	> 1.1
CryptoServer	> 4.45

4 Installation

Before proceeding with the integration, ensure that you have installed an Apache Web Server and OpenSSL on your Linux and that the CryptoServer is available on the network. The following assumptions for the naming of installation directories, file names or network locations are provided for further description:

- CryptoServer: Available on the network at IP address 0.2.2, on TCP port 3001.
- PKCS#11 R3 configuration file: `/etc/utimaco/cs_pkcs11_R3.cfg`
- PKCS#11 library: `/usr/lib/libcs_pkcs11_R3.so`
- Environment variable: `CS_PKCS11_R3_CFG=/etc/utimaco/cs_pkcs11_R3.cfg`

Check that the environment variable for PKCS#11 was permanently added to the system and that the configuration file for PKCS#11 is present at the location given by the `CS_PKCS11_R3_CFG` variable:

```
$less $CS_PKCS11_R3_CFG
```

Check that you can browse the content of the configuration file (`cs_pkcs11_R3.cfg`). Verify that the setting for Device in the section `[CryptoServer]` in the configuration file points to the current location of the CryptoServer in the network. One way to test this is to establish a connection with telnet at the IP address and the port specified for the device e.g.

```
$telnet IP_address port
```

A connection should be established with CryptoServer and after supplying some characters followed by Enter, the server should terminate the connection.

Note the settings `Logpath` and `Logging` (log level). At the specified location, a log with the PKCS#11 operations called `cs_pkcs11_R3.log` is created. The log is useful for debugging and testing the PKCS#11 operations.

The script that starts Apache erases all environment variables, so for the Apache, the CS_PKCS11_R3_CFG environment variable has to be set again. To set this variable for Apache, add in the file `envvars` this line:

```
export CS_PKCS11_R3_CFG=/etc/utimaco/cs_pkcs11_R3.cfg
```

The location of the `envvars` file is in the root directory of the Apache Web Server. The path to root directory for the Apache can be set at runtime with the `-d` option or in the configuration file. The default location for root directory can be found when apache is launched with `-V` option under `HTTPD_ROOT`.

For Ubuntu , the envvars location is: `/etc/apache2/envvars`

Depending on the version of the Apache Web Server, there are two main scenarios that can be implemented to have the private key stored securely inside the HSM:

1. The Apache Web Server version is at least 2.4.42, or the system administrator can upgrade Apache to a version beyond 2.4.42. In this case, the Apache Web Server works with the HSM out of the box. What needs to be done in this case is to install an engine for OpenSSL supporting PKCS#11 and configure OpenSSL to work with that engine. After that, the private key used by the Apache Web Server for https has to be moved (imported) inside the HSM. The last step is to configure the Apache Web Server to use the private key stored inside the HSM.
2. The Apache Web Server version is earlier than 2.4.42 and the Apache Web Server cannot be upgraded to a later version for various reasons. In this case, a special PKCS#11 engine for OPENSSL needs to be used. This engine has the capability to use the private key from HSM for cryptographic operation needed by https protocol. For this to work, the private key is imported into the HSM and then a special reference key file will be created. This file looks like a normal private key, with a PEM extension, but does not contain the actual key, instead it contains a reference to the private key inside the HSM.

In both scenarios after importing the private key inside the HSM, the private key should be deleted from the filesystem. This ensures that hacker are not able to steal the private key, even if they get root access to the server.

4.1 New Apache (>=2.4.42)

This scenario can work using from Utimaco only the PKCS#11 library: libcs_pkcs11_R3.so.

4.1.1 Setting up OpenSSL to Use the PKCS#11 Engine

4.1.1.1 Installing the PKCS#11 Engine

For Debian-based distributions e.g., Ubuntu, the package for the engine is `libenginepkcs11-openssl1.1` , for Red Hat-based distributions the package is `openssl-pkcs11` . To install it on Ubuntu, perform this command:

```
$sudo apt install openssl libengine-pkcs11-openssl1.1
```

4.1.1.2 Configuring OpenSSL

To make OpenSSL work with the PKCS#11 engine, the configuration file of OpenSSL has to be modified. The configuration file location can be determined using this command:

```
$openssl version -d
```

In the path given by the previous command should be the configuration file for the OpenSSL, usually called `openssl.cfg` . In this file, some items need to be added, see https://github.com/OpenSC/libp11#pkcs-11-module-configuration . At the beginning of the file, add this line before any other section:

openssl_conf = openssl _init

Add these lines at the bottom of the file:

```
[openssl_init]

engines=engine_section
```

```
[engine_section]

pkcs11 = pkcs11_section
```

```
[pkcs11_section]

engine_id = pkcs11

dynamic_path = /usr/lib/x86_64-linux-gnu/engines-1.1/libpkcs11.so

MODULE_PATH = /usr/lib/libcs_pkcs11_R3.so
```

init = 0

The default path where the engines for OpenSSL are located (used for `dynamic_path`) can be found with this command:

```
$openssl version -e
```

For Red Hat-based distributions the dynamic path is usually this: `/usr/lib/ssl/engines/libpkcs11.so`

4.1.1.3 Testing OpenSSL Operation With the Engine

```
$openssl engine pkcs11 -t

(pkcs11) pkcs11 engine

    [available]
```

4.1.2 Importing the Private Key Into the HSM

The private key used for https in Apache Web Server can be moved inside the HSM with the help of `pkcs11-tool`. This tool can be installed with the package `opensc`.

```
$sudo apt install opensc

$sudo pkcs11-tool --module /usr/lib/libcs_pkcs11_R3.so -l --pin 1234--write-object prv_key.key --type privkey --id 313133 --label my_label
```

`prv_key.key` is the private key used by apache server for the https cryptographic operations. It can be found in the Apache configuration file as the value for setting: `SSLCertificateKeyFile`. For Ubuntu this is in the file `/etc/apache2/sites-available/default-ssl.conf`. For Red Hat, this is in the file `/etc/httpd/conf.d/ssl.conf`.

The ID has to be provided in hexadecimal, so the ID of the previous example (313233) represents the ASCII code sequence 123.

Another method of importing the private key (more securely) is described in section "Importing the Private Key Inside the HSM". This method requires the `cs_pkcs11.so` engine to be installed.

4.1.3 Configuring Apache

4.1.3.1 Editing the Configuration File

The configuration file location is `/etc/apache2/sites-available/default-ssl.conf` for Ubuntu. For the SSLCertificateKeyFile setting, replace the path to the private key with the URI to the private key inside the HSM e.g. `SSLCertificateKeyFile` "`pkcs11:token=CryptoServer%20PKCS11%20Token;object=private_key_label;pin-value=0000`" (see https://datatracker.ietf.org/doc/html/rfc7512 for the URI format details).

4.1.3.2 Restarting Apache

```
$sudo service apache2 restart
```

4.1.3.3 Testing Apache's Operational Readiness

Launch a browser pointing to the Apache Web Server with https protocol e.g. `https://localhost`. If the default page of the Apache Web Server is shown in the browser, Apache is ready for operation.

4.2 Old Apache (< 2.4.42)

For this scenario, a specific PKCS#11 engine provided by Utimaco is needed.

4.2.1 Setting up OpenSSL to Use the PKCS#11 Engine (< 2.4.42)

4.2.1.1 Installing the PKCS#11 Engine (< 2.4.42)

Copy the engine `library cs_pkcs11.so` from the Utimaco product bundle to the engine's path. The engine's path can be determined by running the following command:

```
$openssl version -e
```

For Ubuntu, the engine's path is this: `/usr/lib/x86_64-linux-gnu/engines-1.1/`

4.2.1.2 Configuring OpenSSL For Engine Operation (< 2.4.42)

Edit the configuration file for OpenSSL. The path of the configuration file can be found with this command:

```
$openssl version -d
```

For Ubuntu, the configuration file is: `/etc/ssl/openssl.cfg`

Add this line at the beginning of the file, before any section starts:

`openssl_conf = openssl_init`

Add the following lines at the end of the file:

`[openssl_init]`

`engines=engine_section`

`[engine_section]`

`pkcs11 = pkcs11_section`

`[pkcs11_section]`

`engine_id = pkcs11`

`dynamic_path = /usr/lib/x86_64-linux-gnu/engines-1.1/cs_pkcs11.so`

```
MODULE_PATH = /usr/lib/libcs_pkcs11_R3.so
```

```
init = 0
```

For the `dynamic_path`, use the engine path and the engine file used when the `cs_pkcs11.so` engine was installed.

For Red Hat-based distributions, the dynamic path is this: `/usr/lib/ssl/engines/cs_pkcs11.so`

4.2.1.3 Testing OpenSSL Operation With the Engine (< 2.4.42)

```
$openssl engine pkcs11 -t
```

```
(pkcs11) pkcs11 engine
```

```
    [available]
```

4.2.2 Importing the Private Key Inside the HSM (< 2.4.42)

To import the key safely inside the HSM, another RSA key pair has to be used for encryption during the transfer. If such a key pair is not available, a new one can be generated using p11tool2 from the product bundle:

```
$p11tool2 LoginUser=0000PubKeyAttr=CKA_LABEL="transfer",CKA_ID=0x313233PrvKeyAttr=CKA_LABEL="transfer",CKA_ID=0x313233GenerateKeyPair=RSA
```

For more details about p11tool2, see [CS_PKCS11T2].

The private key needs to be in a specific format before it can be imported: pkcs8 DER format, protected with a passphrase encoded with AES 128 ECB cipher. To transform it into that format, you can perform this command:

```
$openssl pkcs8 -inform PEM -outform DER -topk8 -v2 aes-128-ecb -passout pass:prv_key_der_pass -in prv_key.key -out prv_key.der
```

Having an RSA key pair inside the HSM and the private key in the correct format, you can import by performing the following command:

```
$openssl pkey -engine cs_pkcs11 -inform engine -in
"import:token=CryptoServer%20PKCS11%20Token;kek-label=transfer;kek-
id=123;key-label=apache;key-id=123;key-type=rsa;key-file=rsa-private-
enc-2048.der;key-passphrase=prv_key_der_pass;pin-value=1234"-noout
```

`key-file` is the private key used by apache server for the https cryptographic operations. It can be find in the apache configuration file as the value for setting: `SSLCertificateKeyFile`. For Ubuntu this is in the file `/etc/apache2/sites-available/default-ssl.conf`. For Red Hat this is in the file `/etc/httpd/conf.d/ssl.conf`.

`key-label` and `key-id` are the label and the ID for the private key after it was moved inside the HSM. `kek-label` and `kek-id` are the label and the ID of the key used for encryption of the private key during the transfer.

The ID is provided in hexadecimal for p11tool2 and it is provided in ASCII code for the import command. So, in the previous examples (0x313233) represents the ASCII code sequence: 123.

Another method of importing the private key (less secure) is described in section "Importing the Private Key Into the HSM".

4.2.3 Generating the Reference Key File (< 2.4.42)

To generate the private key, the cs_pkcs11 engine is used. A command like this has to be performed:

```
$openssl pkeyutl -engine cs_pkcs11 -keyform ENGINE -inkey
"keyref:token=CryptoServer%20PKCS11%20Token;id=%31%32%33;object=apache_priv
_key_inside_HSM;pin-value=1234;file=/tmp/keyref.pem"
```

The command uses the URI to locate the private key inside the HSM and writes a reference into a file that looks like a key, the reference to the private key. This file containing the reference can be used in the configuration of Apache, so that Apache uses for https the real private key stored safely inside the HSM.

The URI syntax is like in rfc7512, except that it begins with `keyref` : instead of `pkcs11` : and if you want to specify the file where the reference key is generated, this has to be the last parameter with the name `file` . If parameter `file` is not specified, the reference key will be created in `/tmp/ref.key.pem.`

4.2.3.1 Testing Apache's Operational Readiness (< 2.4.42)

Launch a browser pointing to the Apache Web Server with https protocol e.g., `https://localhost` . The default page of Apache Web Server should be seen on the browser.

4.2.3.2 Configuring Apache (<2.4.42)

4.2.3.2.1 Editing the Configuration File (< 2.4.42)

The configuration file for https can be found at this path for Ubuntu: `/etc/apache2/sites-available/default-ssl.conf`

Before the line with `<VirtualHost` , add a new line with

```
SSLCryptoDevice cs_pkcs11
```

For `SSLCertificateKeyFile` , replace the path to the private key with the path to the reference key.

4.2.3.2.2 Restarting Apache (< 2.4.42)

```
$sudo service apache2 restart
```

4.3 Deleting the Private Key

When you are confident that the private key is safely stored inside the HSM and can be used from within the HSM, you should delete the private key from the filesystem and other locations that may be accessible to a hacker.

4.4 Further Information

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

All CryptoServer product documentation is also available at the Utimaco IS GmbH website: http://hsm.utimaco.com.

5 References

Reference	***Title/Company***	***Document No.***
[CSADMIN]	CryptoServer – csadm Manual/ Utimaco IS GmbH	2009-0003
[CSTrSh]	CryptoServer Troubleshooting/Utimaco IS GmbH	M011-0008-en
[CS_PKCS11T2]	CryptoServer –PKCS#11 p11tool2 Reference Manual/Utimaco IS GmbH.	2012-0014

6 Contact Address for Support Queries

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

Utimaco IS GmbH
Germanusstr. 4
52080 Aachen
Germany

RMA Query

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

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

Other Support Queries

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