Integration Guide
OpenSSL and PKCS11 Engine
Linux
Integraton Guide: OpenSSL and PKCS11 Engine

Imprint

copyright 2016 Utimaco IS GmbH
Germanusstrasse 4
D-52080 Aachen
Germany

phone +49 (0)241 / 1696-200
fax +49 (0)241 / 1696-199
web http://hsm.utimaco.com
email support-cs@utimaco.com
document version 1.0.4
date June 2016
author M. Sc. Ariano-Tim Donda
document no. IG_OpenSSL_and_PKCS11_Engine

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.
## Contents

1. Introduction ................................................. 4  
2. Overview .................................................. 4  
3. Requirements ............................................. 5  
4. Setup PKCS#11 Engine ...................................... 6  
   4.1 Dynamic .................................................. 6  
      4.1.1 OpenSSL CLI ....................................... 7  
   4.2 Configuration .......................................... 11  
5. Example ................................................... 13  
6. Further Information ....................................... 14
1 Introduction

The CryptoServer is a hardware security module developed by Utimaco IS GmbH, i.e. a physically protected specialized computer unit designed to perform sensitive cryptographic tasks and to securely manage cryptographic keys and data. In a CryptoServer security system security-relevant actions can be executed and security relevant information can be stored. It can be used as a universal, independent security component for heterogeneous computer systems.

2 Overview

OpenSSL is a general purpose cryptography library that provides an open source implementation of the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols.

Starting OpenSSL version 0.9.6, a new component was introduced to support alternative cryptography implementations, most commonly for interfacing with external cryptography devices (e.g. CryptoServer). This new component is generally called engine. OpenSSL does provide several kinds of engines. For this article we provide instructions how to use the PKCS11 engine to work with the CryptoServer PKCS11 interface. There are two options how to use the PKCS11 engine with the application OpenSSL:

**Dynamic** This option enables OpenSSL application to load the PKCS11 engine at runtime.

**Configuration** This option loads the PKCS11 engine based on the configuration `openssl.cnf` automatically when OpenSSL application is used.
3 Requirements

This article assumes that you have a CryptoServer PKCS#11 environment setup and PKCS#11 token already initialized. Further information on how to setup a CryptoServer PKCS11 environment and to initialize a PKCS#11 token can be found in QuickStart Guide PKCS#11 R2 Linux.

**Software- and Hardware Requirements**

<table>
<thead>
<tr>
<th>HSM Model</th>
<th>Utimaco CryptoServer CSe-Series/Se-Series/Se-Gen2-Series PCI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Utimaco CryptoServer CSe-Series/Se-Series/Se-Gen2-Series LAN</td>
</tr>
<tr>
<td></td>
<td>Utimaco CryptoServer Simulator</td>
</tr>
<tr>
<td>HSM Firmware</td>
<td>Utimaco SecurityServer 4.0.0</td>
</tr>
<tr>
<td>Software</td>
<td>OpenSSL 1.0.1f</td>
</tr>
<tr>
<td></td>
<td>libengine-pkcs11-openssl 0.1.8-3</td>
</tr>
<tr>
<td></td>
<td>p11tool2 from product package Utimaco SecurityServer 4.0.0</td>
</tr>
<tr>
<td></td>
<td>Ubuntu 14.04 LTS</td>
</tr>
</tbody>
</table>

To test a PKCS#11 engine successful integration with the CryptoServer generate a RSA key pair (4096bit) for PKCS#11 token slot zero.

```
> Console

p11tool2 slot=0 LoginUser=123456
   PubKeyAttr=CKA_ID="Example",CKA_MODULUS_BITS=4096
   PrvKeyAttr=CKA_ID="Example" GenerateKeyPair=RSA
```

This RSA key will be later used to run a typical OpenSSL cryptographic operation using the PKCS#11 engine.
4 Setup PKCS\#11 Engine

By default the PKCS\#11 engine for OpenSSL is not installed. To install the PKCS\#11 engine for OpenSSL run:

```console
$ sudo apt-get install libengine-pkcs11-openssl
```

4.1 Dynamic

The dynamic method requires to load the PKCS\#11 engine before it can be used further OpenSSL cryptographic operations. You run OpenSSL as CLI to provide loading of the PKCS\#11 engine manually.

```console
utimaco@utimaco:/$ openssl
OpenSSL> engine -t dynamic -pre SO_PATH:/usr/lib/engines/engine_pkcs11.so
   -pre ID:pkcs11 -pre\ LIST_ADD:1 -pre LOAD -pre
   MODULE_PATH:/usr/lib/libcs_pkcs11_R2.so
(dynamic) Dynamic engine loading support
[Success]: SO_PATH:/usr/lib/engines/engine_pkcs11.so
[Success]: ID:pkcs11
[Success]: LIST_ADD:1
[Success]: LOAD
[Success]: MODULE_PATH:/usr/lib/libcs_pkcs11_R2.so
Loaded: (pkcs11) pkcs11 engine
   [ available ]
OpenSSL>
```
4.1.1 OpenSSL CLI

To check whether it’s now possible to access the RSA key we have previously generated for our PKCS#11 token on slot zero, create a self signed certificate file as shown next.

>_ Console

```
utimaco@utimaco:/$ openssl
OpenSSL>
OpenSSL> engine -t dynamic -pre SO_PATH:/usr/lib/engines/engine_pkcs11.so
   -pre ID:pkcs11 -pre LIST_ADD:1 -pre LOAD -pre
       MODULE_PATH:/usr/lib/libcs_pkcs11_R2.so
(dynamic) Dynamic engine loading support
[Success]: SO_PATH:/usr/lib/engines/engine_pkcs11.so
[Success]: ID:pkcs11
[Success]: LIST_ADD:1
[Success]: LOAD
[Success]: MODULE_PATH:/usr/lib/libcs_pkcs11_R2.so
Loaded: (pkcs11) pkcs11 engine
   [ available ]
OpenSSL> req -engine pkcs11 -new -key id_4578616D706C65 -keyform engine\n   -out certificate.pem -text -x509 -subj "/C=DE/ST=NRW/L=Aachen/O=Utimaco
       IS GmbH/OU=SystemEngineering\ HSM/CN=Max Mustermann"
engine "pkcs11" set.
PKCS11 token PIN: *****
```

Show the content of the certificate.pem file and double check if all information given previously are correct.

>_ Console

```
utimaco@utimaco:/$ openssl x509 -text -noout -in certificate.pem
Certificate:
   Data:
       Version: 3 (0x2)
       Serial Number: 17299328040199222677 (0xf0138f3261077595)
```
Signature Algorithm: sha256WithRSAEncryption

Issuer: C=DE, ST=NRW, L=Aachen, O=Utimaco IS GmbH,
OU=SystemEngineering HSM, CN=Max Mustermann

Validity
Not Before: Mar 12 14:10:51 2015 GMT
Not After: Apr 11 14:10:51 2015 GMT

Subject: C=DE, ST=NRW, L=Aachen, O=Utimaco IS GmbH,
OU=SystemEngineering HSM, CN=Max Mustermann

Subject Public Key Info:
Public Key Algorithm: rsaEncryption

Public-Key: (4096 bit)
Modulus:
00:82:00:aa:84:eb:cd:7d:0d:7c:ce:e0:e2:a5:65:
a5:47:b3:7f:cb:e8:ae:0e:0d:ab:2c:33:83:8f:56:
d3:ba:2b:9e:ff:ea:a8:3b:b2:cb:a4:41:f0:8f:4d:
dc:55:0a:00:77:36:11:10:e7:4b:95:56:a3:7e:bf:
0d:be:ec:02:fb:48:81:a5:ba:0c:13:ac:ae:e8:60:
01:0c:04:05:ab:b2:e9:bc:f1:30:0b:fa:6e:45:12:
74:93:d4:a0:a3:b4:d7:00:e6:00:61:67:1b:0a:68:
d8:17:1a:e5:dd:78:47:52:e7:4c:df:3c:a3:c7:60:
b7:11:7f:39:1f:1e:a3:b9:9d:01:0c:08:00:ed:96:
39:6b:b2:c8:d8:0f:09:4b:11:fc:aa:58:97:0f:7c:
3a:48:cd

Exponent: 65537 (0x10001)

X509v3 extensions:

X509v3 Subject Key Identifier:


X509v3 Authority Key Identifier:


X509v3 Basic Constraints:

CA: TRUE

Signature Algorithm: sha256WithRSAEncryption

cc:59:de:e0:e0:2a:e0:2a:0f:67:0a:5f:76:31:d0:53:da:9f:
4.2 Configuration

Loading the PKCS#11 engine manually each time you need to run a certain openssl command is not handy. By adding some engine information to (openssl.cnf) configuration file, OpenSSL automatically loads the PKCS#11 engine. After adding engine information (openssl.cnf) you can use the engine with the engine parameter referring to the id given in the (openssl.cnf). For our example we used a short (/etc/ssl/openssl.cnf) file.

```bash
[openssl_def]
environments = engine_section

[engine_section]
pkcs11 = pkcs11_section

[pkcs11_section]
engine_id = pkcs11
dynamic_path = /usr/lib/engines/engine_pkcs11.so
MODULE_PATH = /usr/lib/libcs_pkcs11_R2.so
PIN = "123456"
init = 0

[req]
distinguished_name = req_distinguished_name

[req_distinguished_name]
```
To check whether it's now possible to access the RSA key using OpenSSL with the configuration file setup, run the next example.

```
>  Console

utimaco@utimaco:/$ openssl req -engine pkcs11 -new -key\ id_4578616D706C65
   -keyform engine -out certificate.pem -text -x509\ -subj
   "\C=DE/ST=NRW/L=Aachen/O=Utimaco IS GmbH/OU=SystemEngineering\ HSM/CN=Max
   Mustermann"
```

Using OpenSSL like this is much easier and can be used in shell scripts compared to load PKCS#11 engine manually each time you run a certain OpenSSL command.
5 Example

**OpenSSL & PKCS#11**

You cannot do everything with OpenSSL and the PKCS#11 engine. OpenSSL's PKCS#11 support is poor.

For example, you cannot create an RSA key inside the HSM with `genrsa/genpkey`. You have to use `p11tool2` for it like this:

```console
p11tool2 LoginUser=PIN PubKeyAttr=CKA_ID=mykey,CKA_MODULUS_BITS=4096,
PrvKeyAttr=CKA_ID=mykey GenerateKeyPair=RSA
```

Now you can use this key in certain operations. But it won't work to sign data with a private key provided as a file since PKCS#11 expects the private key inside the token (HSM). For the created key, do it as follows:

```console
openssl rsautl -engine pkcs11 -keyform engine -inkey id_6D796B6579\ -sign
-in datatosign.txt -out signature.dat
```

**Key Identification**

"6D796B6579" is the hex value of the String "mykey" (again, that's the way OpenSSL expects it).

To verify the signature, use:

```console
openssl rsautl -engine pkcs11 -keyform engine -inkey id_6D796B6579\ -verify
-in signature.dat
```

You can also replace "sign" by "encrypt" and "verify" by "decrypt" in the commands above.
6 Further Information

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

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

Utimaco IS GmbH
Germanusstraße 4
D - 52080 Aachen
Germany

phone  +49 241 1696 - 200
fax    +49 241 1696 - 199

web    https://hsm.utimaco.com
email  support-cs@utimaco.com