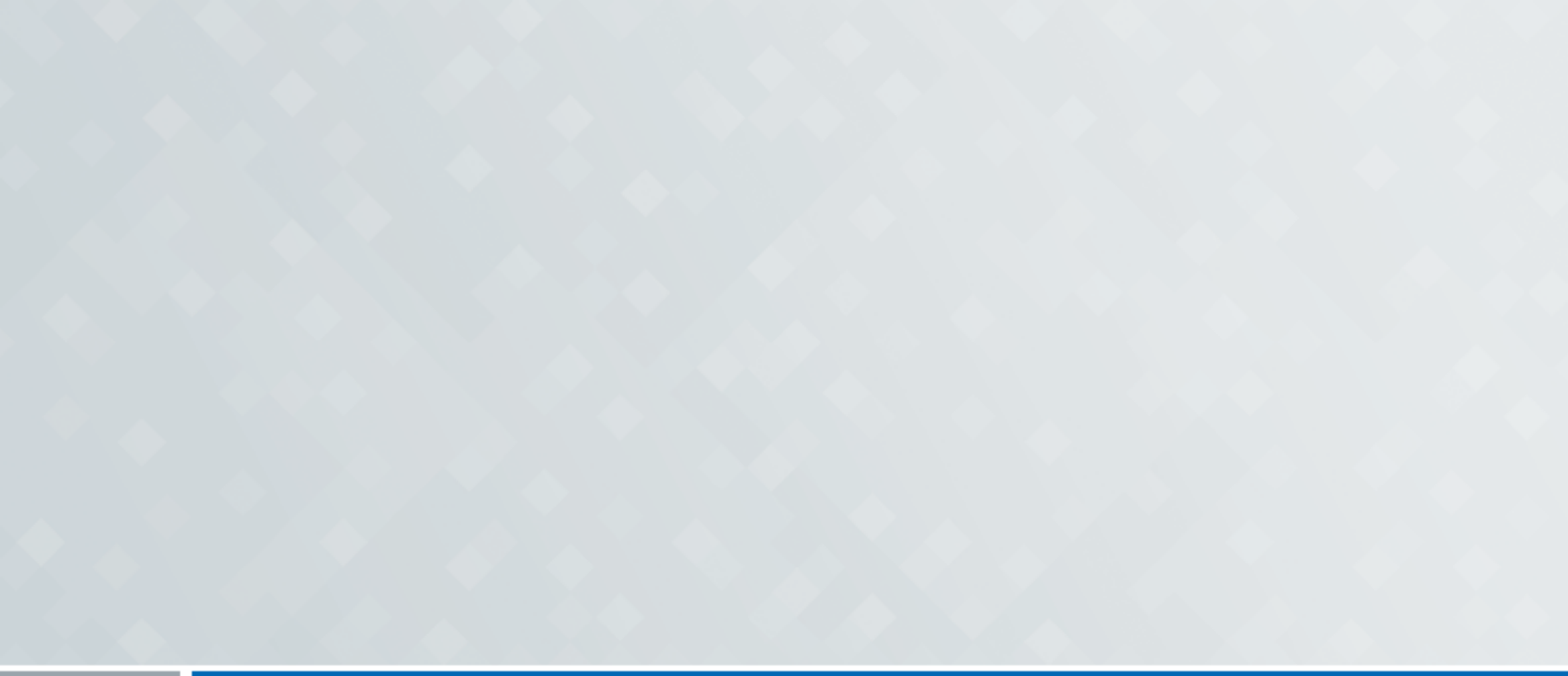




PKCS11Interop API .NET

Integration Guide

CryptoServer HSM

SecurityServer 3.21.0



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	2026-07-20
Status	PUBLISHED
Document No.	IG-2026-0072
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	Introduction	4
1.1	Motivation	4
1.2	Concepts	4
2	Requirements	6
3	Configuration	7
3.1	Create your .NET application.....	7
3.2	Bind the Pkcs11Interop Package to your .NET Application.....	7
3.3	CryptoServer PKCS#11 Library and the Configuration File	8
3.4	Code Skeleton to Establish Connection Between a .NET Application and the HSM	10
4	Further Information	12
4.1	References	12
5	Contact and Support Information.....	13

1 Introduction

This integration guide explains how to use the PKCS#11 cryptography standard with Utimaco Hardware Security Module (HSM) from the PKCS11Interop API .NET.

1.1 Motivation

Utimaco IS GmbH provides its clients with a hardware technology concept; however, it is not always clear if a client's specific application would be able to communicate with the Utimaco HSM properly. In this guide we introduce a solution to one instance of this problem, namely how to use the PKCS#11 standard with the HSM from a C# application using an open source PKCS11Interop interface [\[1\]](#).

1.2 Concepts

PKCS11Interop is a wrapper library written in C# that allows .NET application developers to use full power of PKCS#11 standard. It is an interface between a .NET application and a PKCS#11 vendor library, where the latter directly communicates to the HSM device. The complete communication pipeline between a .NET application and the Utimaco HSM device is explained below.

PKCS11Interop library consists of two sets of APIs: LowLevelAPI and HighLevelAPI. The former has following characteristics:

- Provides a developer with a full power of PKCS#11 C API.
- Requires C style coding.

whereas the latter has these characteristics:

- Built on top of LowLevelAPI.
- Provides a developer with a friendly .NET API.

As already mentioned, PKCS11Interop is considered to be a wrapper library, which implements the conventional programming pattern also known as **Wrapper** or **Adapter**. Whenever any friendly HighLevelAPI method is invoked from a particular .NET application, the corresponding LowLevelAPI method is implicitly invoked. The latter call is followed by the corresponding PKCS#11 C function call, which is the vendor specific function implementation of the cryptography standard. In such a way end-to-end connection between any .NET application and the Utimaco HSM device is set up.

More information about this library can be found under this link: <http://www.pkcs11interop.net/>.

2 Requirements

The following table describes the list of requirements being used to solve the issue:

HSM Model	CryptoServer CS-Series/S-Series/Se-Series LAN CryptoServer Simulator
HSM Firmware	SecurityServer 3.21.0
Software	Microsoft Windows 7 (64-Bit) Microsoft Visual Studio 2013 Professional Pkcs11Interop 2.0.2

Table 1: Software and hardware requirements

3 Configuration

3.1 Create your .NET application

First step is to create an empty .NET application within Visual Studio, for example by:

```
FILE > New > Project... > C# Console Application
```

3.2 Bind the Pkcs11Interop Package to your .NET Application

Since binaries are not available under a download link, but rather in the form of NuGet package, make sure your copy of Visual Studio has the NuGet extension. Detailed information on how to install NuGet Client into Visual Studio can be found under this link <http://docs.nuget.org/consume/installing-nuget>.

After having installed this extension, open the Package Manager Console using the following path within Visual Studio:

```
TOOLS > NuGet Package Manager > Package Manager Console
```

In the opened manager console, you should execute the following command:

› **_ Console**

```
[PM] Install-Package Pkcs11Interop
```

NuGet retrieves the package from the specified package source and installs it in the project that is selected. Files are copied to the solution, references might be added to the project. In Solution Explorer, you can see references that Visual Studio has added for the installed library, as shown in the following figure.

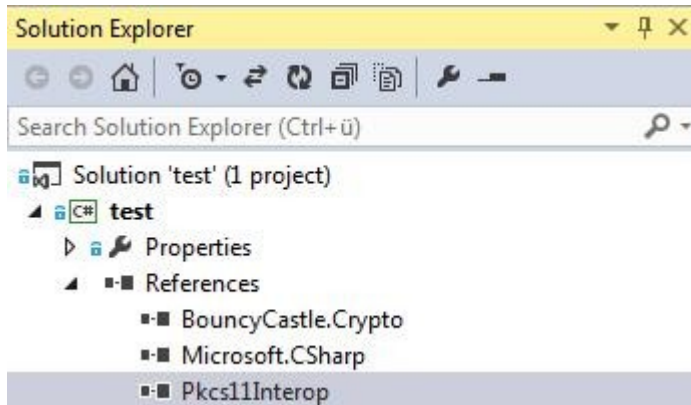


Figure 1 : Added references

You can now use the library in your project.

3.3 CryptoServer PKCS#11 Library and the Configuration File

The application that uses the CryptoServer PKCS#11 interface must be able to load the Utimaco PKCS#11 shared library. Furthermore, the configuration file `cs_pkcs11_R2.cfg` must be created and the CryptoServer PKCS#11 library must be able to find and access the file.

There are several options to tell the CryptoServer PKCS#11 library where the configuration file is located:

- Set the environment variable `CS_PKCS11_R2_CFG` to the correct path and location of the configuration file by:

›_ Console

```
#> set CS_PKCS11_R2_CFG="c:\My Documents\utimaco\cs_pkcs11_R2.cfg"
```

- Place the configuration file in the current working directory, which is useful for development.
- Place the configuration file in the same directory where the application is located.
- The configuration file in a system specific directory.

The CryptoServer PKCS#11 library searches the location of the configuration file `cs_pkcs11_R2.cfg` in the following order:

- First, it will be checked if the environment variable `CS_PKCS11_R2_CFG` is set and if it contains the name and location of the configuration file.
- If not, it will be checked if the configuration file is located in the user's home directory (`%USERPROFILE%`).
- If not, it will be checked if the configuration file is located in the current working directory.
- If not, it will be checked if the file is located in the same directory where the application is located.
- If not, it will be checked if the file is located somewhere in `%PATH%`.
- If not, it will be checked if the file is located in the WINDOWS directory (e.g. `C:\WINDOWS`).



If the CryptoServer PKCS#11 library was installed by SecurityServer, the environment variable `CS_PKCS11_R2_CFG` is set.

The configuration of the CryptoServer PKCS#11 library for the CryptoServer must be done within a file named `cs_pkcs11_R2.cfg`. This file can contain several sections, one `[Global]` section for general configuration one `[CryptoServer]` section for each CryptoServer device that should be addressed by the application and one optional `[Slot]` section for every slot that is in use.

Below the shortest possible configuration is listed, which is sufficient for testing purposes.

> _ Console

```
# sample configuration file
[Global]
# Select the log level (NONE...TRACE)
Logging = 4
# Specifies the path where the logfile shall be created.
Logpath = C:/your_path_to_log_file
# Defines the maximum size of the logfile. If the maximum is reached,
# old entries will be overwritten. Can be defined as value
# in bytes or as formatted text. E.g. value of '1000' means logsize
# is 1000 bytes whereas value of '1000kb' means 1000 kilobytes.
# Allowed formats are 'kb', 'mb' and 'gb'.
Logsize = 10mb
[CryptoServer]
# Device address to connect a CryptoServer device
Device = 10.17.1.35
```

3.4 Code Skeleton to Establish Connection Between a .NET Application and the HSM

After all previous configuration steps have been done, it should be now possible to use Pkcs11Interop interface API to provide the .NET application with an end-to-end connection to Utimaco HSM device. The following shortest code skeleton can be used to establish this bridge:

> _ Console

```
using Net.Pkcs11Interop. ighLevelAPI;

static void Main(string[] args)
{
    // Path to unmanaged library provided by the device vendor
    String path = "cs_pkcs11_R2.dll";

    // Unmanaged PKCS#11 library is loaded by the constructor of Pkcs11 class.
    // Every PKCS#11 library needs to be initialized with C\Initialize method
    // which is also called automatically by the constructor of Pkcs11 class.
    Pkcs11 pkcs11 = new Pkcs11(path, false);

    // find accessible slot of the token
    List<Slot> slots = pkcs11.GetSlotList(true);

    Slot slot = slots[0];

    Session session = slot.OpenSession(false); String pin = "1234";
    session.Login(CKU.CKU_USER, pin); session.Logout(); session.CloseSession();

    // Unmanaged PKCS#11 library is unloaded by Dispose() method. // C_Finalize
    // should be the last call made by an application and it
    // is also called automatically by Dispose() method.

    pkcs11.Dispose();
}
```

4 Further Information

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

4.1 References

1. Pkcs11Interop - developed by Jaroslav Imrich.

5 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 Str. 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.