CryptoServer LAN
CryptoServer

PKCS#11 CryptoServer Administration Tool

Manual for System Administrators

utimaco®

Imprint

Copyright 2017	Utimaco IS GmbH Germanusstr. 4 D-52080 Aachen Germany
Phone	+49 (0)241 / 1696-200
Fax	+49 (0)241 / 1696-199
Internet	http://hsm.utimaco.com
e-mail	hsm@utimaco.com
Document Version	1.3.1
Date	2017-02-14
Status	Final
Document No.	M013-0001-en

Table of Contents

1 Introduction

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

1.1 About this Manual

This manual provides an overview of Utimaco's PKCS#11 R2 implementation, and contains solution-oriented information about how to set up, manage and maintain the PKCS#11 R2 implementation with the PKCS#11 CryptoServer Administration Tool (P11CAT).

1.1.1 Target Audience for this Manual

This manual is primarily designed to be used by administrators who want to use the PKCS#11 CryptoServer Administration Tool to use and administer Utimaco's PKCS#11 R2 implementation.

1.1.2 Document Conventions

We use the following conventions in this manual:

Convention	*Usage*	*Example*
Bold	Items of the Graphical User Interface (GUI), e.g., menu options	Press the **OK** button.
`Monospaced`	File names, folder and directory names, commands, file outputs, programming code samples	You find the file `example.conf` in the `/exmp/demo/` directory.
Italic	References and important terms	See Chapter 3, "Sample Chapter" in the *CryptoServer LAN/CryptoServer CryptoServer Command-line Administration Tool -csadm -Manual for System Administrators*.

Table 1: Document conventions

We use 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.

1.2 Other Manuals

The CryptoServer is supplied as a PCI-Express plug-in card in the following series:

- CryptoServer CSe Series (PCIe)
- CryptoServer Se Series (PCIe)
- CryptoServer Se-Series Gen2

The CryptoServer LAN (appliance) is supplied in the following series:

- CryptoServer LAN CSe Series
- CryptoServer LAN Se Series
- CryptoServer LAN Se-Series Gen2

We provide the following manuals on the product CD for the CSe-, Se-Series, and Se-Series Gen2 of the CryptoServer plug-in card and for the CryptoServer LAN (appliance):

Quick Start Guides

You will find these Manuals in the main folder of the SecurityServer product CD. They are available only in English, do not cover all possible scenarios, and are intended as a supplement to the product documentation provided on the SecurityServer product CD.

- *CryptoServer LAN - Quick Start Guide*
  If you are looking for step-by-step instructions on how to bring the CryptoServer LAN into service, how to prepare a computer (Windows 7) for the CryptoServer administration and how to start administrating your CryptoServer with the Java-based GUI CryptoServer Administration Tool (CAT), read this document.
- *CryptoServer PCIe - Quick Start Guide*
  If you are looking for step-by-step instructions on how to bring the CryptoServer PCIe plug-in card into service, how to install the CryptoServer driver on a computer with minimal

RHEL 7.0 installation and how to start administrating your CryptoServer with the CryptoServer Command-line Administration Tool (csadm), read this document.

Manuals for System Administrators

You will find the manuals on the product CD in the following folder:
`…Documentation\Administration Guides\`

- *CryptoServer - Manual for System Administrators*
  If you need to administer a CryptoServer PCIe plug-in card or a CryptoServer LAN using the CryptoServer Administration Tool (CAT), read this manual. Furthermore, this manual provides a detailed description of the CryptoServer functions, required for the correct and effective operation of the product.
- *CryptoServer LAN - Manual for System Administrators*
  If you need to administer a CryptoServer LAN (appliance), read this manual. Since a CryptoServer plug-in card is integrated into the CryptoServer LAN, please read the *CryptoServer Manual for System Administrators*, as well.
- *CryptoServer/CryptoServer LAN - Troubleshooting*
  If problems occur while you are using a CryptoServer PCIe plug-in card or a CryptoServer LAN (appliance), read this manual.
- *CryptoServer/CryptoServer LAN*
  PKCS#11 CryptoServer Administration Tool (P11CAT) – Manual for System Administrators
  If you need to administer the PKCS#11 R2 interface with the PKCS#11 CryptoServer Administration Tool (P11CAT), read this manual.
- *CryptoServer LAN/CryptoServer*
  CryptoServer Command-line Administration Tool - csadm - Manual for System Administrators
  If you need to administer a CryptoServer PCIe plug-in card or a CryptoServer LAN using the CryptoServer Command-line Administration Tool (csadm), read this manual (only English version available).

Operating Manuals

You will find the manuals on the product CD in the following folder:
`…Documentation\Operating Manuals\`. In these manuals you will find all the necessary information for using appropriately the CryptoServer PCIe plug-in card hardware and the CryptoServer LAN (appliance) hardware.

2 Utimaco's PKCS#11 Implementation

PKCS#11 is a standard that defines a programming interface for cryptographic tokens such as smartcards or hardware security modules (HSMs). From the PKCS#11 point of view, a cryptographic token is a device that stores objects and can perform cryptographic functions. Such objects can be secret, private or public keys and certificates. In addition, the objects can each have different attributes which not only define how you handle them, but may also limit the areas in which they can be used.

2.1 The CryptoServer as a Cryptographic Token

The PKCS#11 standard specifies that a cryptographic token can be a hardware security module (HSM) or a smartcard. From the PKCS#11 point of view, the CryptoServer hardware security module used here is a cryptographic token.

Using the CryptoServer as a cryptographic token has the benefit that cryptographic operations are performed in a secure hardware environment which gives added protection to the PKCS#11 objects.

All the sensitive data in the CryptoServer, including the PKCS#11 objects, is stored within the CryptoServer in encrypted form. If the objects are backed up outside the CryptoServer they are encrypted with a Master Backup Key (MBK) which can only be used for authentication with approval by a second person.

The application PKCS#11 CryptoServer Administration Tool (P11CAT) and the CryptoServer communicate exclusively over a secure connection (secure messaging).

Utimaco's hardware security module provides up to 1000 PKCS#11 slots for use.

The next figure shows how Utimaco's PKCS#11 API can be used for more than one CryptoServers.

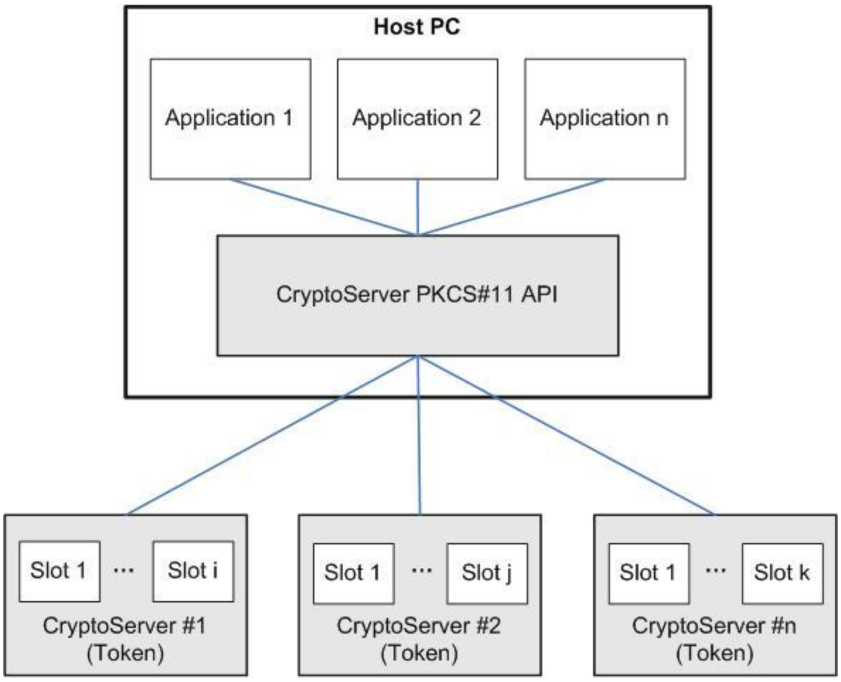


Figure 1: Usage of PKCS#11 API with n CryptoServer

2.2 The P11CAT Users

In Utimaco's PKCS#11 implementation, the following users are responsible for administering the cryptographic token:

- CryptoServer Administrator
  The CryptoServer Administrator is responsible for administering the CryptoServer. Only the default ADMIN user or an administrator with permission 2 in the user group 7 (20000000) is permitted to initialize the tokens (the slots) by initializing the Security Officer's (SO's) authentication (PIN or password).

- Security Officer (SO)
  The Security Officer is responsible for configuring the Cryptographic Token or Slots. The Security Officer (SO) can change the PIN or password they use for authentication, which was initialized by the ADMIN user (or a CryptoServer administrator with min. permission 2 in the user group 7). In addition, only the Security Officer has the right to initialize the PIN or password used to authenticate the User.

- User
  The User can generate, load, delete and use certificates or keys for the cryptographic token. The User can change the PIN or password they use for authentication, which was initialized by the Security Officer (SO). This User can also be an application which uses the cryptographic token.

  - Configurable Key Manager
    The role of the User can be split into a key manager role, with permission (00000020)

to manage cryptographic keys, and a key user role, with the permission (00000002) to use cryptographic keys. The CryptoServer PKCS#11 library should be configured accordingly, if you want to use these two new user roles. See Chapter 4.3, “Setting Up the Configuration Objects” for further details about available configuration objects.

2.3 P11CAT

The dedicated PKCS#11 CryptoServer Administration Tool user interface (referred to simply as P11CAT in this document) is where you configure the PKCS#11 slots. The ADMIN user who is responsible for CryptoServer administration tasks, or an administrator whose authentication status is 20000000 or higher, can log on to the CryptoServer directly via the P11CAT interface and perform token or slot initializations from there.

You can use P11CAT to generate keys and key pairs. You can generate and administer both internal keys, which are used within CryptoServer, and also external keys, which are used outside CryptoServer. You can use the P11CAT to input key attributes manually, select them from a list, or transfer them from a template file.

For security reasons, you cannot use the P11CAT interface to perform a standard PKCS#11 authentication because this is based on an insecure connection to the CryptoServer. Every time the ADMIN user, the Security Officer, or the User, logs on via the P11CAT, the device establishes a secure connection (Secure Messaging) to the CryptoServer.

From version 3.20 of the SecurityServer product CD we supply only one PKCS#11 implementation called PKCS#11 R2.

You can only use the P11CAT to administer the PKCS#11 R2 implementation.

3 Installing P11CAT

If you are using a CryptoServer LAN, you must first have integrated this into your network. In addition, you must have assigned an IP address to this device by entering it directly on the device.

If you are using a CryptoServer plug-in card, you must already have installed this along with the appropriate driver. The CryptoServer PCI/PCIe Operating Manual supplied in printed form to you on delivery, describes how to install the driver for the plug-in card.

3.1 System Requirements

- CPU: Intel x86/x64, AMD x86/x86-64
- Hard disk capacity: at least 120 Mbit
- RAM: more than 12 Mbit
- PCIe expansion slot, if you are using a plug-in card Se-Series, CSe-Series and Se-Series Gen2
- Network card: Ethernet 10/100/1000 Mbit/s for the connection to the CryptoServer LAN
- Operating systems: All operating systems that are currently supported for the host computer, whereon P11CAT shall be installed, are listed in the document `CS_PD_SecurityServer_SupportedPlatforms.pdf` provided on the SecurityServer/CryptoServer SDK product CD in the folder `…\Documentation\Product Details`.
- CD-ROM disc drive
- Current Java versions for Windows/Linux

3.2 Performing Installation for Windows

To use the P11CAT, Oracle's Java Runtime Environment (JRE) must be installed first on your computer.

If you have installed the product CD on your computer, and enabled PKCS#11 R2 – Cryptographic Token Interface in the Select Components window during installation, PKCS#11 R2 is now successfully installed on your computer.

If you click the **Create a desktop icon**, a link to PKCS**#11 CryptoServer Administration** appears on your desktop.

Please do not start the PKCS#11 CryptoServer Administration tool just yet.

You can also start P11CAT by clicking on Windows **Start** menu > **All Programs** > **Utimaco** > **CryptoServer**.

Before starting P11CAT for the first time, you must specify the address of the CryptoServer which PKCS#11 R2 is to access in the `cs_pkcs11_R2.cfg` configuration file.

After the installation of the SecurityServer product CD, you will find this configuration file, by default, on your computer here:

```
C:\Program Files\Utimaco\CryptoServer\Lib
```

1. Open the `cs_pkcs11_R2.cfg` file with an appropriate text editor.
2. Set the CryptoServer device you want to use in the `[CryptoServer]` section.

   - If you are using a CryptoServer LAN, enter its IP address in the file section shown below after `Device =`.

     ```
     [CryptoServer]
     # Device specifier (here: CryptoServer is CSLAN with IP address
     192.168.0.1)
     Device = 192.168.0.1
     ```

From CryptoServer LAN Version 4.2.0 onwards the Internet Protocols IPv4 and IPv6 are supported.

   - If you are using a CryptoServer PCI or a CryptoServer PCIe card, delete the comment # to the left of `Device = PCI:0` and the one to the left of `[CryptoServer]` in the file section shown below.

     ```
     #[CryptoServer]
     ```

```
# Device specifier (here: CryptoServer is internal PCI device)
# For unix:
#Device = /dev/cs2
# For windows:
#Device = PCI:0
```

Comment out the file section for configuring a CryptoServer LAN as the CryptoServer device shown above. The configuration file `cs_pkcs11_R2.cfg` can contain only a single active `[CryptoServer]` section.

3. To change the number of slots, simply enter the number of slots you require after `SlotCount =`. The CryptoServer has from 0 to 1000 slots available.

```
# Maximum number of slots that can be used
SlotCount = 10
```

4. Save the changes and close the configuration file `cs_pkcs11_R2.cfg`.
   You can now start P11CAT.

You will find more details about the `cs_pkcs11_R2.cfg` configuration file in the [CS_PKCS11DEV] provided also on the SecurityServer product CD under `<Installation path of SecurityServer product CD>\Documentation\Crypto_APIs\PKCS11_R2`.

3.3 Setting up P11CAT for Linux

P11CAT can be used on Linux 32- and 64-bit systems.

To use the P11CAT, Oracle's Java Runtime Environment (JRE) must be installed first on your Linux machine.

Use the package manager for your Unix/Linux system to install the Oracle Java Runtime Environment (JRE).

To setup P11CAT, follow these steps:

1. Copy the p11`cat.jar` file into your user directory.

```
cp <path to Product
CD>/Software/All_Supported_Operating_Systems/Administration/p11cat.jar
~/
```

ℹ *Before starting P11CAT for the first time, you must specify the address of the CryptoServer which PKCS#11 R2 is to access in the `cs_pkcs11_R2.cfg` configuration file.*

2. Customize the `cs_pkcs11_R2.cfg` configuration file.

   You will find this configuration file on the product CD here:
   `Software/Linux/x86-32` or `x86-64/Crypto_APIs/PKCS11_R2/sample`

   a) Copy the `cs_pkcs11_R2.cfg` file to your user directory.

   b) Open `cs_pkcs11_R2.cfg`.

   c) If you are using a CryptoServer LAN, enter its IP address in the file section shown below, after `Device =`.

   ```
   [CryptoServer]

   # Device specifier (here: CryptoServer is CSLAN with IP address
   192.168.0.1)

   Device = 192.168.0.1
   ```

ℹ *From CryptoServer LAN Version 4.2.0 onwards the Internet Protocols IPv4 and IPv6 are supported.*

   d) If you are using a CryptoServer PCI or a CryptoServer PCIe plug-in card, delete the comment # to the left of `Device = /dev/cs2` in the file section shown above.

   ```
   #[CryptoServer]

   # Device specifier (here: CryptoServer is internal PCI device)

   # For unix:

   #Device = /dev/cs2

   # For windows:

   #Device = PCI:0
   ```

   Comment out the file section for configuring a CryptoServer LAN as the CryptoServer device shown above. The configuration file `cs_pkcs11_R2.cfg` can contain only a single active `[CryptoServer]` section.

   e) To change the number of slots, simply enter the number of slots you require after `SlotCount =`.

   ```
   # Maximum number of slots that can be used

   SlotCount = 10
   ```

The CryptoServer has from 0 to 1000 slots available.

f) Save the changes and close the configuration file `cs_pkcs11_R2.cfg`.

3. Set the environment variable so that P11CAT is able to find the configuration file.

   Use the following command:

   ```
   export CS_PKCS11_R2_CFG=<path to configuration file>
   ```

   Example:

   ```
   export CS_PKCS11_R2_CFG=~/utimaco/CryptoServer/cs_pkcs11_R2.cfg
   ```

4. Start P11CAT with the following command:

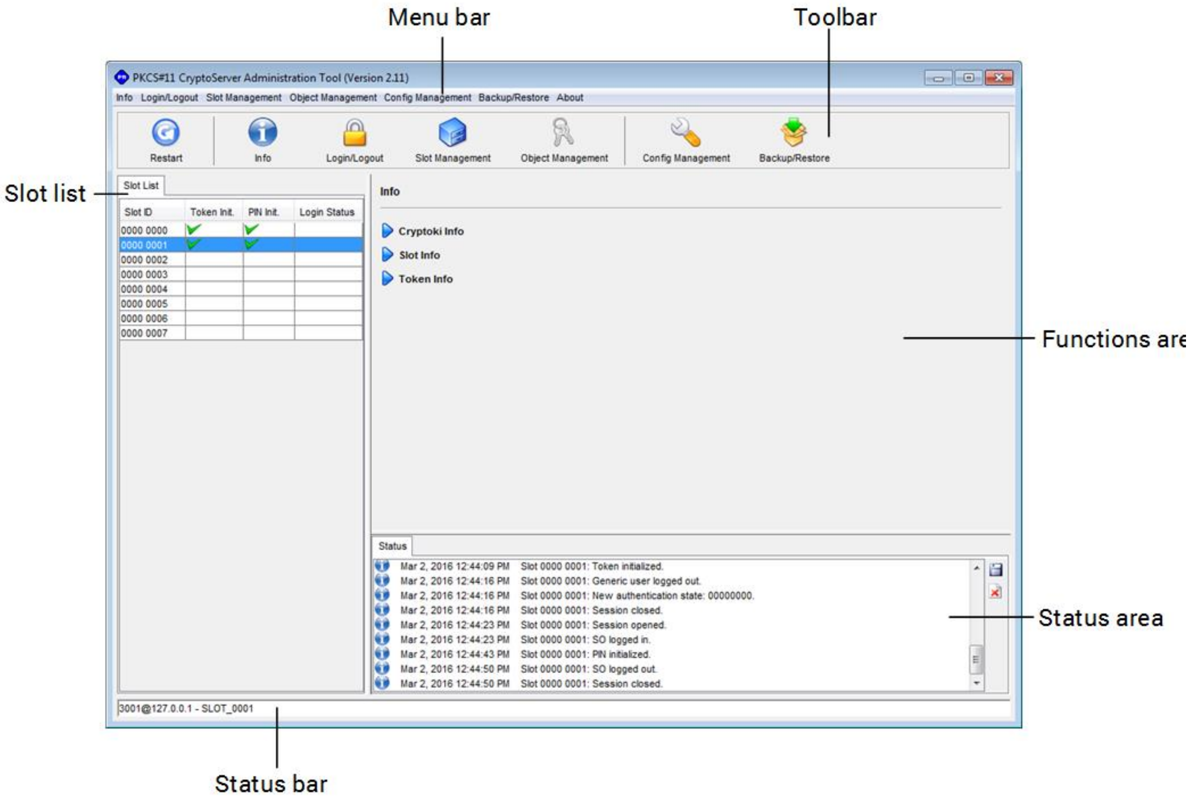
   ```
   java -jar ~/p11cat.jar
   ```

4 Administering PKCS#11 with P11CAT

This chapter uses practical and solution-oriented examples to describe all the configuration steps you need to perform, to use P11CAT to run your Utimaco's PKCS#11 R2 implementation.

4.1 P11CAT – Overview of the Graphical User Interface (GUI)

The P11CAT main window is the main operating control of P11CAT. For example, it could look as follows:



Slot ID	Token Init.	PIN Init.	Login Status
0000 0000	✓	✓	
0000 0001	✓	✓	
0000 0002			
0000 0003			
0000 0004			
0000 0005			
0000 0006			
0000 0007			

```
Mar 2, 2016 12:44:09 PM   Slot 0000 0001: Token initialized.
Mar 2, 2016 12:44:16 PM   Slot 0000 0001: Generic user logged out.
Mar 2, 2016 12:44:16 PM   Slot 0000 0001: New authentication state: 00000000.
Mar 2, 2016 12:44:16 PM   Slot 0000 0001: Session closed.
Mar 2, 2016 12:44:23 PM   Slot 0000 0001: Session opened.
Mar 2, 2016 12:44:23 PM   Slot 0000 0001: SO logged in.
Mar 2, 2016 12:44:43 PM   Slot 0000 0001: PIN initialized.
Mar 2, 2016 12:44:50 PM   Slot 0000 0001: SO logged out.
Mar 2, 2016 12:44:50 PM   Slot 0000 0001: Session closed.
```

3001@127.0.0.1 - SLOT_0001

Figure 2: P11CAT main window and its elements

It contains the following elements:

- Menu bar
  The menu bar is located directly under the title bar in the upper border of the P11CAT main window. It enables you to access all P11CAT functions. For example you can configure the PKCS#11 slots on the CryptoServer (**Slot Management**), generate, export, import and delete keys (**Object Management**), etc.

- Toolbar
  The toolbar is located under the menu bar. It enables you to access quickly all P11CAT functions.
- Slot list
  The **Slot List** is located in the left hand area of the P11CAT main window. All slots of the configured devices are listed here with **Slot ID**, **Token Init**., **PIN Init**., and **Login Status**. The Slot ID contains two areas, e.g., 0000 0003. The first four digits stand for the corresponding device, and the other four digits define the slot number.
- Functions area
  The functions area is located in the right hand area of the P11CAT main window under the toolbar. It remains empty until you have selected an option in the menu bar or you have clicked on a button in the toolbar.
- Status area
  The status area is located directly under the functions area. All status and error messages are shown here.
- Status bar
  The status bar is located on the bottom border of the P11CAT main window. The address of the selected device and the corresponding slot number are shown here, e.g., `10.13.4.8 - SLOT_0001`.

4.2 Setting up a Slot (Init Token)

Before you can use Utimaco's PKCS#11 implementation, the CryptoServer administrator must set up one or more PKCS#11 slots (Init Token). Here you must define the Security Officer's (SO's) authentication for the slot or token that is to be set up.

The specified Security Officer must then log in to this slot and assign a password (PIN) for the User of that slot.

4.2.1 Performing a Generic Login

Before you can initialize a slot or a token as an Administrator, you must have logged in to the CryptoServer with an authentication status of at least 20000000.

1. Select a slot in the slot list.
2. Click the **Login/Logout** button in the toolbar.
   - ▣ If you only want one person to be able to log in to the CryptoServer, select **Login Generic**.
     This person will require an authorization status of at least 20000000.

- ▣ If you want two (two-person rule) or more (n-person rule, where n = 0 to 16) people to be able to log in to the CryptoServer (with approval by a second person), select Login Generic again.
  Two people (for approval by a second person) each require an authorization status of at least 10000000, so that the two people have a total authorization status of 20000000.

3. In the User Name text box, enter the name of the CryptoServer administrator with the required permission of 20000000.
4. Select under Authentication Token the authentication mechanism you require.

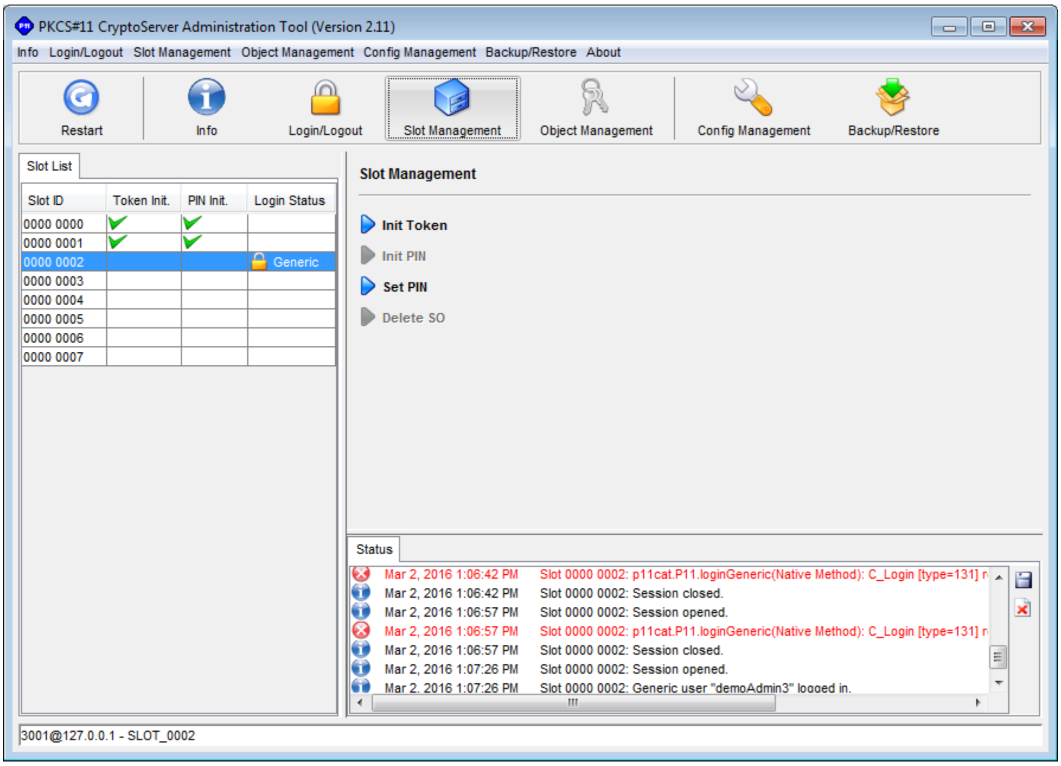
   You can select one of these three authentication mechanisms:

   - ▣ Password
     In this case you must enter the user's password in the Password text box.
   - ▣ Keyfile
     In this case you must input the storage location of the keyfile in the Keyfile text box, and the key file password in the Password text box, if the keyfile is password-protected.
   - ▣ Smartcard
     In this case you must enter the smartcard information in the Smartcard Specifier text box.
     If the REINER SCT PIN pad, supplied by Utimaco, is connected to a USB interface of your computer, and you are using the smartcard (also supplied by Utimaco), you must input the key information as follows: `:cs2:cyb:USB0`.
5. Click the Login button in the functions area.
6. Follow the instructions on P11CAT if you are using a password or a keyfile for authentication. Follow the instructions on the PIN pad if you are using a smartcard for authentication.
   In the Slot List, the status of the current logging on process is indicated by a lock and the entry Generic in the Login Status column.

4.2.2 Setting up a Security Officer (SO)

To set up a Security Officer, you must be logged in to the CryptoServer as an administrator with min. permission 2 in the user group 7 (min authentication status 20000000) or as the ADMIN user. That is, for the selected slot in the Slot List the entry Generic is shown in the Login Status column.

1. Select the required slot in the **Slot List**.
2. Click the **Slot Management** button in the toolbar.



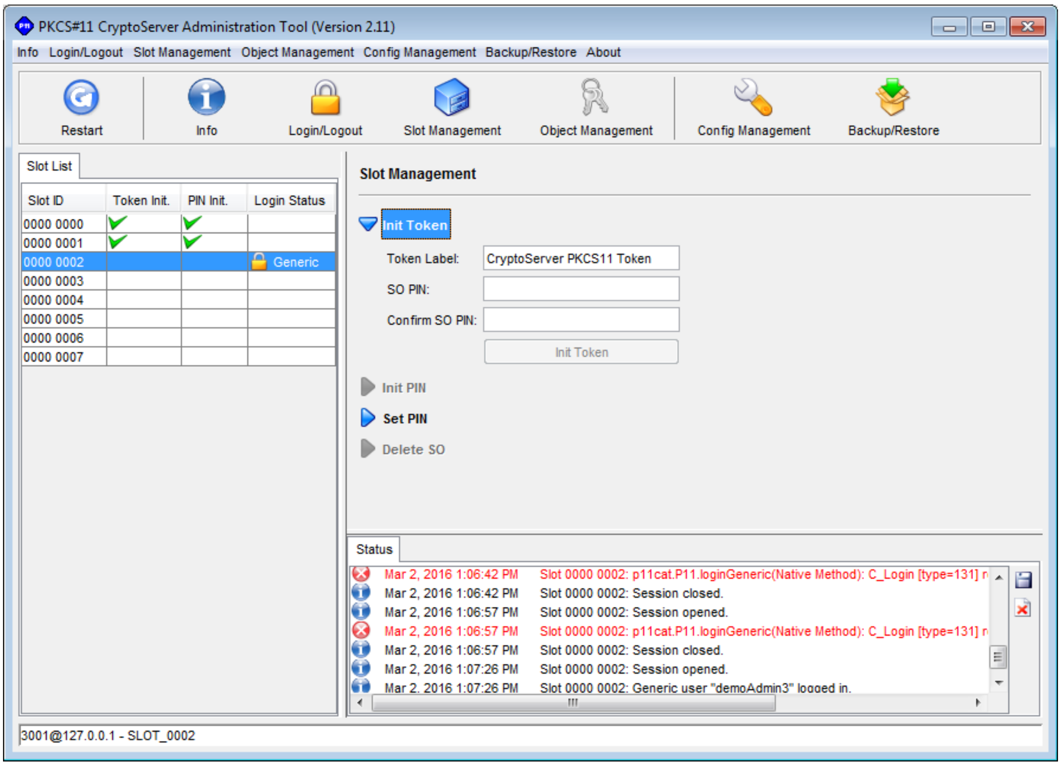
Slot ID	Token Init.	PIN Init.	Login Status
0000 0000	✓	✓	
0000 0001	✓	✓	
0000 0002			Generic
0000 0003			
0000 0004			
0000 0005			
0000 0006			
0000 0007			

```
Mar 2, 2016 1:06:42 PM   Slot 0000 0002: p11cat.P11.loginGeneric(Native Method): C_Login [type=131] r
Mar 2, 2016 1:06:42 PM   Slot 0000 0002: Session closed.
Mar 2, 2016 1:06:57 PM   Slot 0000 0002: Session opened.
Mar 2, 2016 1:06:57 PM   Slot 0000 0002: p11cat.P11.loginGeneric(Native Method): C_Login [type=131] r
Mar 2, 2016 1:06:57 PM   Slot 0000 0002: Session closed.
Mar 2, 2016 1:07:26 PM   Slot 0000 0002: Session opened.
Mar 2. 2016 1:07:26 PM   Slot 0000 0002: Generic user "demoAdmin3" logged in.
```

3001@127.0.0.1 - SLOT_0002

Figure 3: P11CAT main window – Slot Management

3. Click **Init Token** in the functions area.



Slot ID	Token Init.	PIN Init.	Login Status
0000 0000	✓	✓	
0000 0001	✓	✓	
0000 0002			Generic
0000 0003			
0000 0004			
0000 0005			
0000 0006			
0000 0007			

Token Label: CryptoServer PKCS11 Token

SO PIN:

Confirm SO PIN:

Init Token

```
Mar 2, 2016 1:06:42 PM   Slot 0000 0002: p11cat.P11.loginGeneric(Native Method): C_Login [type=131] r
Mar 2, 2016 1:06:42 PM   Slot 0000 0002: Session closed.
Mar 2, 2016 1:06:57 PM   Slot 0000 0002: Session opened.
Mar 2, 2016 1:06:57 PM   Slot 0000 0002: p11cat.P11.loginGeneric(Native Method): C_Login [type=131] r
Mar 2, 2016 1:06:57 PM   Slot 0000 0002: Session closed.
Mar 2, 2016 1:07:26 PM   Slot 0000 0002: Session opened.
Mar 2, 2016 1:07:26 PM   Slot 0000 0002: Generic user "demoAdmin3" logged in.
```

3001@127.0.0.1 - SLOT_0002

Figure 4: P11CAT main window – Define SO PIN

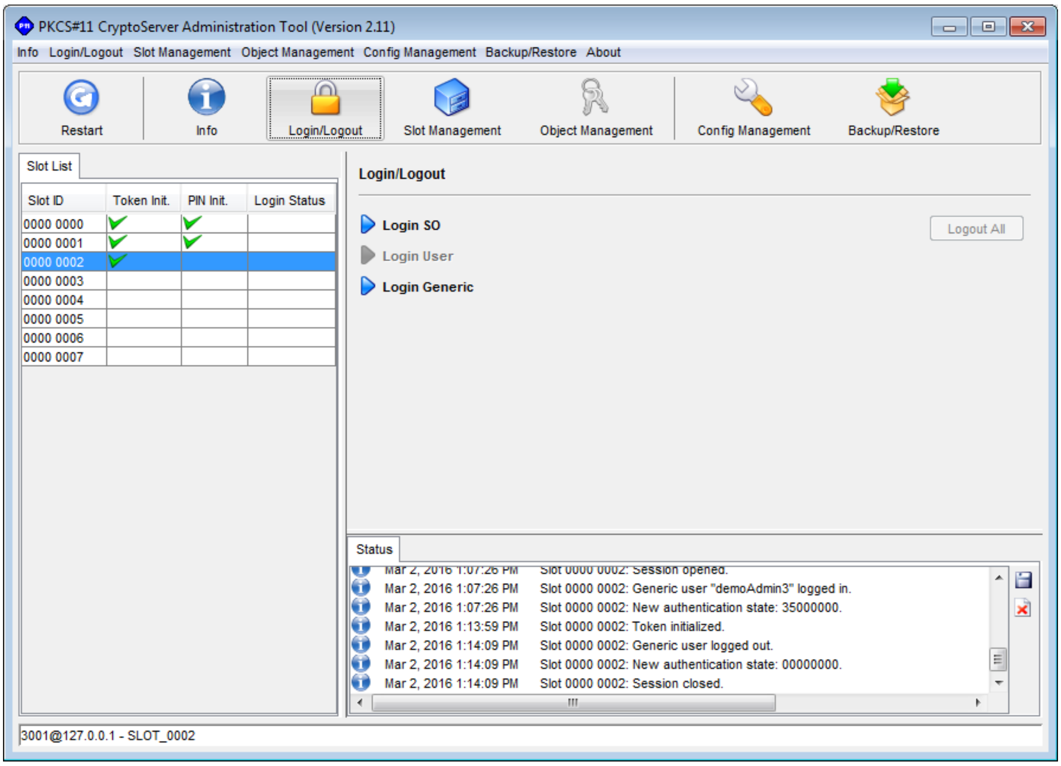
4. Enter, optionally, a unique **Token Label** for the selected PKCS#11 slot. By default, the string **CryptoServer PKCS11 Token** is assigned to each PKCS#11 slot. This label is then automatically assigned to the SO of the slot as the user attribute **L**, and is displayed on execution of the csadm command `ListUser` (see [CSADMIN]).
5. Enter a password for the Security Officer (SO) in the **SO PIN** text box.
6. Repeat the password entry in the **Confirm SO PIN** text box.
7. Confirm the input by clicking the **Init Token** button.

   In the **Slot List** the status of the token initialization is indicated by a green tick in the **Token Init.** column.
8. Log out from the CryptoServer by clicking the **Login/Logout** button in the toolbar, and then clicking the **Logout All** button in the functions area.

4.2.3 Setting up a User

The Security Officer (SO) must assign a password for the User for the slot that is now initialized.

1. In the **Slot List**, click the relevant slot.
2. Click the **Login/Logout** button in the toolbar.



Slot ID	Token Init.	PIN Init.	Login Status
0000 0000	✓	✓	
0000 0001	✓	✓	
0000 0002	✓		
0000 0003			
0000 0004			
0000 0005			
0000 0006			
0000 0007			

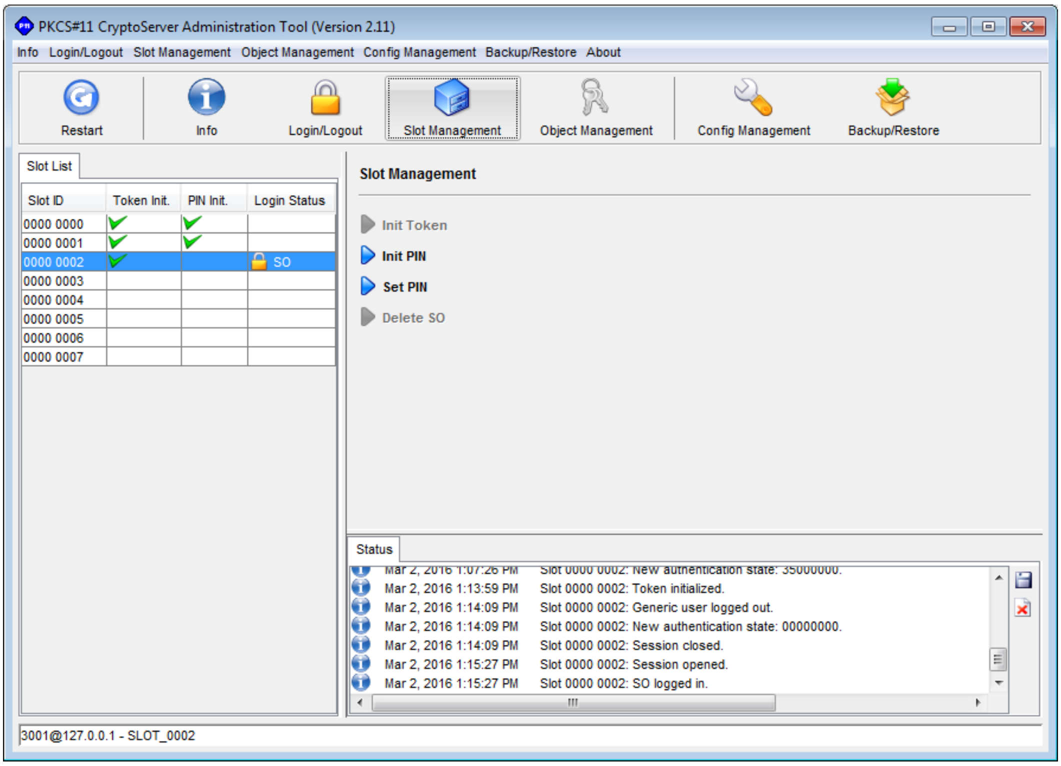
```
Mar 2, 2016 1:07:26 PM   Slot 0000 0002: Session opened.
Mar 2, 2016 1:07:26 PM   Slot 0000 0002: Generic user "demoAdmin3" logged in.
Mar 2, 2016 1:07:26 PM   Slot 0000 0002: New authentication state: 35000000.
Mar 2, 2016 1:13:59 PM   Slot 0000 0002: Token initialized.
Mar 2, 2016 1:14:09 PM   Slot 0000 0002: Generic user logged out.
Mar 2, 2016 1:14:09 PM   Slot 0000 0002: New authentication state: 00000000.
Mar 2, 2016 1:14:09 PM   Slot 0000 0002: Session closed.
```

3001@127.0.0.1 - SLOT_0002

Figure 5: P11CAT main window – Login/Logout area

3. Select **Login SO** in the functions area.
4. Enter the password configured for the Security Officer into the **SO PIN** text box.
5. Click the **Login** button in the functions area.

   In the **Slot List**, the login status is indicated by a lock and the entry `SO` in the **Login Status** column.
6. Click the **Slot Management** button in the toolbar.
7. Click **Init PIN** in the functions area.


PKCS#11 CryptoServer Administration Tool (Version 2.11)
Info Login/Logout Slot Management Object Management Config Management Backup/Restore About
Restart
Info
Login/Logout
Slot Management
Object Management
Config Management
Backup/Restore
Slot List
Slot Management
Init Token
Init PIN
Set PIN
Delete SO
Status
3001@127.0.0.1 - SLOT_0002

Slot ID	Token Init.	PIN Init.	Login Status
0000 0000	✓	✓	
0000 0001	✓	✓	
0000 0002	✓		SO
0000 0003			
0000 0004			
0000 0005			
0000 0006			
0000 0007			

```
Mar 2, 2016 1:07:26 PM    Slot 0000 0002: New authentication state: 35000000.
Mar 2, 2016 1:13:59 PM    Slot 0000 0002: Token initialized.
Mar 2, 2016 1:14:09 PM    Slot 0000 0002: Generic user logged out.
Mar 2, 2016 1:14:09 PM    Slot 0000 0002: New authentication state: 00000000.
Mar 2, 2016 1:14:09 PM    Slot 0000 0002: Session closed.
Mar 2, 2016 1:15:27 PM    Slot 0000 0002: Session opened.
Mar 2, 2016 1:15:27 PM    Slot 0000 0002: SO logged in.
```

Figure 6: P11CAT main window – Init User PIN

8. Enter a password for the slot's User in the Normal User PIN text box.
9. Repeat the password entry in the Confirm Normal User PIN: text box.
10. Confirm the entry by clicking the Init PIN button.

    In the Slot List the status of the token initialization is indicated by a green tick in the PIN Init. column.

11. Log out from the CryptoServer by clicking the Login/Logout button in the toolbar, and then clicking the Logout All button in the functions area.

    You have now set up the slot.

    Repeat these steps to set up all the other slots you require.

4.2.4 Setting up a Key Manager

If you want to have a separate user role for the key usage and key management functions, you have to set up a CryptoServer Key Manager with permission 2 in the user group 1.

1. Use the user management functions in the CAT or the csadm administration tool to create a new Customized User, for example, `KM_0000` for slot 0, with the permission mask 00000020. For details about how to create a new Customized User with CAT see Chapter *"Creating a User"* in [CSMSADM].

2. In P11CAT, select the corresponding slot and log in to the CryptoServer as the administrator with at least permission 2 in the user group 7 as described in Chapter 4.2.1, "Performing a Generic Login".

Alternatively, you can log in to the CryptoServer in the selected slot as the SO if the global configuration attribute CKA_CFG_ALLOW_SLOTS has been set to CK_TRUE. See Chapter 4.3.2, "Changing the Global Configuration" for details.

3. Change the global configuration attribute `CKA_CFG_AUTH_KEYM_MASK` to `0x00000020` as described in Chapter 4.3.2, "Changing the Global Configuration".

4.3 Setting Up the Configuration Objects

A CryptoServer administrator is also responsible for configuring how objects behave. Here, the term objects does not mean the objects in PKCS#11. These are configuration objects that Utimaco has defined, and included in PKCS#11. They can be configured in the following three ways:

- Local configuration

  The local configuration shows where the `cs_pkcs11_R2.cfg` configuration file is stored. This information is read-only and cannot be changed.

- Global configuration

  The global configuration, which can only be changed by the CryptoServer administrator with min. permission 2 in the user group 7 (20000000), can be used to configure the Security Officer (SO) PKCS#11-related permissions for objects, globally, for all slots.

- Slot configuration

  A slot can only be configured by a Security Officer (SO) if the CryptoServer administrator has assigned them the right to configure slots. In this case the Security Officer can override the global slot configuration for each individual slot.

Changes on the attributes of the global and slot configuration objects are stored in the `CXIKEY.db` database. This database is deleted on alarm occurrence and when the `Clear` command is performed. For details about the Clear command, see *Chapter "The Clear Functionality"* in [CSADMIN].

We highly recommend creating a backup of the configuration objects:

- To back up the Slot CryptoServer Configuration Object, see Chapter4.6.3, "Creating a Slot Configuration Backup".

- To back up the Global CryptoServer Configuration Object resp. `CXIKEY.db`, use the csadm command `BackupDatabase` as described in *Chapter "BackupDatabase"* of the [CSADMIN] or CAT, as described in *Chapter "Creating a Database Backup"* of the [CSMSADM].

4.3.1 Attributes for Global Configuration

This chapter describes the available attributes for the global configuration:

- `CKA_CFG_ALLOWS_SLOTS`

  This attribute enables the Security Officer (SO) to configure slots.

  Possible values:

  - `CK_TRUE` - the SO is permitted to configure slots.
  - `CK_FALSE` (default) - the SO is not permitted to configure slots.

- `CKA_CFG_CHECK_VALIDITY_PERIOD`

  This attribute checks the validity period of the key.

  The validity period of a key is only checked, if the following functions are to be performed using the key: `C_SignInit (), C_EncryptInit (), C_DecryptInit (), C_DeriveInit (), C_WrapKey (), C_UnwrapKey ()`

  Possible values:

  - `CK_TRUE` - the validity period of a key is checked, if the key has the attributes `CKA_START_DATE` and `CKA_END_DATE`.
  - `CK_FALSE` (default) - the validity period of a key is not checked.

- `CKA_CFG_AUTH_PLAIN_MASK`

  This attribute defines the permissions required to import and export a key in plain text.

  Default value: 0x00000002 - corresponds to the permissions of the Cryptographic User, who is already set up in the CryptoServer.

  IMPORTANT:
  If you change the default setting, you must also use the CAT or csadm administration tools to set up the corresponding user in your CryptoServer. This user must be assigned the permissions specified here. For step-by-step instructions on how to create a new user with CAT, please read *Chapter "Creating a New User"* in [CSMSADM]. Examples for creating different users with csadm are provided in *Chapter "AddUser"* of the [CSADMIN].

- `CKA_CFG_WRAP_POLICY`

  This attribute applies a key wrapping policy specifying how keys are encrypted so they can be securely exported outside the CryptoServer.

  Possible values:

  - `CK_TRUE` - a strong key (for example, 256-bit AES) cannot be encrypted with a weak key (for example, 1024-bit RSA).
  - `CK_FALSE` (default) - a strong key can be encrypted with a weak key.

- `CKA_CFG_AUTH_KEYM_MASK`

  This attribute defines the authentication status of the Key Manager who, by default, has the same permissions as the User (00000002).

  Default value: 0x00000002 - corresponds to the permission of the Cryptographic User, who is already set up in the CryptoServer.

  You can change this permission for the key manager here to 00000020, and split the User role into two roles: key user and key manager.

  IMPORTANT:
  If you change the default value, you must use the user management functions in CAT or csadm administration tools to set up a key manager in CryptoServer, who is assigned the permission 2 in the user group 1 corresponding to the authentication status 00000020 specified here. For step-by-step instructions on how to create a new user with the Key Manager role with CAT and P11CAT, please read *Chapter "Creating a New User"* in [CSMSADM], and Chapter 4.3.2, “Changing the Global Configuration”. An example for creating a user in the Key manager role with csadm is provided in *Chapter "AddUser"* of the [CSADMIN].

- `CKA_CFG_SECURE_DERIVATION`

This security relevant attribute is only available in P11CAT version 2.12 and later, and requires that CXI firmware module version 2.1.11.1 and later is loaded in the CryptoServer.

This attribute prohibits the use of the following key derivation mechanisms, and prevents Reduced Key Space attacks:

  - `CKM_XOR_BASE_AND_DATA`
  - `CKM_CONCATENATE_DATA_AND_BASE`
  - `CKM_CONCATENATE_BASE_AND_DATA`
  - `CKM_CONCATENATE_BASE_AND_KEY`
  - `CKM_EXTRACT_KEY_FROM_KEY`

For a detailed description of the mechanisms see [PKCS11ICMS].

Possible values:

- ▣ `CK_TRUE` – none of the key derivation mechanisms listed above can be used by the function `C_Derive ()`.
- ▣ `CK_FALSE` (default) – the key derivation mechanisms listed above can be used by the function `C_Derive ()` for key derivation.

- `CKA_CFG_SECURE_IMPORT`

This security relevant attribute is only available in P11CAT version 2.12 and later, and requires that CXI firmware module version 2.1.11.1 and later is loaded in the CryptoServer.

This attribute prevents simple Key Extraction attacks by performing additional strict checks on wrapping keys.

Possible values:

- ▣ `CK_TRUE` – the key wrapping and unwrapping functions perform additional strict checks on wrapping keys. For more details about the additional checks, please read *Chapter "Global CryptoServer Configuration Object"* in [CS_PKCS11DEV] provided on the SecurityServer product CD (and on the CryptoServer SDK CD) under `\Documentation\Crypto_APIs\PKCS11_R2\`.
- ▣ `CK_FALSE` (default) – no additional strict checks on wrapping keys are performed.

- `CKA_CFG_SECURE_RSA_COMPONENTS`

This security relevant attribute is only available in P11CAT version 2.12 and later, and requires that CXI firmware module version 2.1.11.1 and later is loaded in the CryptoServer.

This attribute applies restrictions on the length of the public exponent used for the generation of RSA keys.

Possible values:

- ▣ `CK_TRUE` (default) – new RSA keys cannot be created with very low, smaller than 0x10001, public exponents.
- ▣ `CK_FALSE` – new RSA keys can be created with very low public exponents.

- `CKA_CFG_P11R2_BACKWARDS_COMPATIBLE`

This security relevant attribute is only available in P11CAT version 2.12 and later, and requires that CXI firmware module version 2.1.11.1 and later is loaded in the CryptoServer.

This attribute determines whether keys can be used by default as base keys for key derivation or not.

Possible values:

- ▣ `CK_TRUE` – keys generated by using an ECC scheme or Diffie-Hellman algorithm can be used as base keys for key derivation (PKCS#11 standard non-compliant legacy); may be necessary for some integrations.
- ▣ `CK_FALSE` (default) –newly generated or imported keys cannot be used by default as base keys for key derivation.

- ■ `CKA_CFG_ENFORCE_BLINDING`

This security relevant attribute is only available in P11CAT version 2.13 and later, and requires that CXI firmware module version 2.1.11.4 and later is loaded in the CryptoServer.

This attribute prevents side-channel analysis (SCA) attacks by enabling/disabling CryptoServer-specific software measures for SCA resistance. These software measures imply changing the internal computations of RSA and ECC keys in a way that simple and differential power analysis, as well as electromagnetic and timing analysis measurements on cryptographic keys do not reveal information any longer.

However, the measures for SCA resistance negatively affect the performance of the cryptographic operations on RSA and ECDSA keys. Therefore, they are disabled by default, and can be enabled, if necessary.

Possible values:

- ▣ `CK_TRUE` – software measures for SCA resistance are used for cryptographic operations on RSA and ECDSA keys.
- ▣ `CK_FALSE` (default) – normal (without software measures for SCA resistance) cryptographic operations on RSA and ECDSA keys are used.

- ■ `CKA_CFG_SECURE_SLOT_BACKUP`

This security relevant attribute is available as from SecurityServer/CryptoServer SDK 4.10 and requires that the CXI firmware module version 2.2.1.0 or later is loaded in the CryptoServer.

This attribute enforces the usage of an individual backup key (Tenant Backup Key, TBK) per slot instead of the MBK to protect external keys and key backups. By default, only MBK-protected external key storage and key backup is enabled.

Possible values:

- `CK_TRUE` – use slot-specific backup keys (TBKs) derived from the CryptoServer's MBK to encrypt keys stored in an external database and key backups.
- `CK_FALSE` (default) – use the CryptoServer's MBK to encrypt external keys and key backups.

Make sure you have set this configuration attribute according to your security policy before your CryptoServer production environment gets operational. Changing this attribute for a CryptoServer that is currently in use causes previously generated external keys and their backups to become inaccessible.

To further individualize your slot-individual backup key, you can optionally define a slot specific passphrase to be used for the derivation of that backup key. This is done by setting the `CKA_CFG_SLOT_BACKUP_PASS_HASH` slot configuration attribute prior to enabling the usage of slot-individual backup keys with the `CKA_CFG_SECURE_SLOT_BACKUP` global configuration attribute set to `CK_TRUE`. See Chapter 4.3.3, “Attributes for Slot Configuration” for a detailed description of the `CKA_CFG_SLOT_BACKUP_PASS_HASH` slot configuration attribute and Chapter 4.3.4, “Changing the Slot Configuration” for instructions on how the set the slot specific attributes.

4.3.2 Changing the Global Configuration

To be able to change the global configuration you must have logged in to the CryptoServer with authentication status of at least 20000000.

1. Click the **Login/Logout** button in the toolbar.

   - If you only want one person to be able to log in to the CryptoServer, select **Login Generic**.
     This person will require an authorization status of at least 20000000.

- ▣ If you want two (two-person rule) or more (n-person rule, where n = 0 to 16) people to be able to log in to the CryptoServer (with approval by a second person), select **Login Generic** again.
  Two people (for approval by a second person) each require an authorization status of at least 10000000, so that the two people have a total authorization status of 20000000.

2. In the toolbar, click the **Config Management** button.
3. Click **Global Configuration** in the functions area.

   The following configuration attributes and their default settings are displayed in the table under **Global configuration**:

Attribute	*Data Type*	*Value*
`CKA_CFG_ALLOWS_SLOTS`	`CK_BBOOL`	`CK_FALSE`
`CKA_CFG_CHECK_VALIDITY_PERIOD`	`CK_BBOOL`	`CK_FALSE`
`CKA_CFG_AUTH_PLAIN_MASK`	`CK_ULONG (hex)`	`0x00000002`
`CKA_CFG_WRAP_POLICY`	`CK_BBOOL`	`CK_FALSE`
`CKA_CFG_AUTH_KEYM_MASK`	`CK_ULONG (hex)`	`0x00000002`
`CKA_CFG_SECURE_DERIVATION`	`CK_BBOOL`	`CK_FALSE`
`CKA_CFG_SECURE_IMPORT`	`CK_BBOOL`	`CK_FALSE`
`CKA_CFG_SECURE_RSA_COMPONENTS`	`CK_BBOOL`	`CK_TRUE`
`CKA_CFG_P11R2_BACKWARDS_COMPATIBLE`	`CK_BBOOL`	`CK_FALSE`
`CKA_CFG_ENFORCE_BLINDING`	`CK_BBOOL`	`CK_FALSE`
`CKA_CFG_SLOT_BACKUP`	`CK_BBOOL`	`CK_FALSE`

Table 2: Global configuration attributes

These attributes are explained in more detail in Chapter 4.3.1, “Attributes for Global Configuration”.

4. Click the corresponding entry in the **Value** column to change the attribute value.

To apply your changes for the attributes CKA_CFG_AUTH_PLAIN_MASK and CKA_CFG_AUTH_KEYM_MASK press ENTER.

4.3.3 Attributes for Slot Configuration

Once you have authenticated yourself as the SO with the attribute `CKA_CFG_ALLOW_SLOTS` set to `CK_TRUE`, you can change the configuration of the corresponding slot. You are allowed to change all configuration attributes as for the global configuration listed in Chapter 4.3.1, "Attributes for Global Configuration" except for the `CKA_CFG_ALLOW_SLOT` attribute.

In addition to the attributes described in Chapter 4.3.1, “Attributes for Global Configuration”, the `CKA_CFG_SLOT_BACKUP_PASS_HASH` slot-specific attribute can be configured in a slot configuration object. This attribute stores the SHA-256 hash value of a passphrase which is only used for the derivation of a slot-individual backup key (see `CKA_CFG_SLOT_BACKUP` in chapter 4.3.1).

The `CKA_CFG_SLOT_BACKUP_PASS_HASH` attribute can be set by the SO to any string even if the `CKA_ALLOW_SLOTS` attribute is set to `CK_FALSE`. The default value is the SHA-256 hash of the empty string.

If you want to use an individual passphrase for the derivation of the slot-individual backup key, make sure you have set the CKA_CFG_SLOT_BACKUP_PASS_HASH configuration attribute prior to the activation of the CKA_CFG_SLOT_BACKUP attribute and before your CryptoServer production environment gets operational. Changing this attribute for a slot that is currently in use causes previously generated external keys and their backups to become inaccessible.

4.3.4 Changing the Slot Configuration

If you are working as the Security Officer (SO), the CryptoServer user administrator (default min. permissions 20000000) must change the global `CKA_CFG_ALLOW_SLOTS` attribute in the global configuration from `FALSE` to `TRUE` before you can change the configuration of individual slots. You cannot change the configuration of individual slots until they do this. However, after this you can override the global configuration for all the slots with the slot configuration for the individual slot.

The SO must use his password (PIN) to log in to each individual slot before they can change its configuration.

1. Click the relevant slot in the **Slot List.**
2. Click the **Login/Logout** button in the toolbar.
3. Click **Login SO** in the functions area and enter the password for the Security Officer into the **SO PIN** text box.
4. Click the **Login** button in the functions area.

In the **Slot List**, the login status is indicated by a lock and the entry `SO` in the **Login Status** column.

5. Click **Config Management** in the toolbar.
6. Select **Slot Configuration** in the functions area.

   The following configuration attributes and their default settings are displayed: and can be changed for the selected slot:

Attribute	*Data Type*	*Value*
`CKA_CFG_CHECK_VALIDITY_PERIOD`	`CK_BBOOL`	`CK_FALSE`
`CKA_CFG_AUTH_PLAIN_MASK`	`CK_ULONG (hex)`	`0x00000002`
`CKA_CFG_WRAP_POLICY`	`CK_BBOOL`	`CK_FALSE`
`CKA_CFG_AUTH_KEYM_MASK`	`CK_ULONG (hex)`	`0x00000002`
`CKA_CFG_SECURE_DERIVATION`	`CK_BBOOL`	`CK_FALSE`
`CKA_CFG_SECURE_IMPORT`	`CK_BBOOL`	`CK_FALSE`
`CKA_CFG_SECURE_RSA_COMPONENTS`	`CK_BBOOL`	`CK_TRUE`
`CKA_CFG_P11R2_BACKWARDS_COMPATIBLE`	`CK_BBOOL`	`CK_FALSE`
`CKA_CFG_ENFORCE_BLINDING`	`CK_BBOOL`	`CK_FALSE`
`CKA_CFG_SLOT_BACKUP`	`CK_BBOOL`	`CK_FALSE`
`CKA_CFG_SLOT_BACKUP_PASS_HASH`	`RFC2279 String`	`not set`

Table 3: Configuration attributes for an individual slot

All attributes except for the `CKA_CFG_SLOT_BACKUP_PASS_HASH` can be set/changed for the selected slot. Use the command-line tool p11tool2 (`SecureSlotPass` command) to define a passphrase to be used for the derivation of a slot-individual backup key. The slot-individual backup key is not used by default, but only if the `CKA_CFG_SLOT_BACKUP` attribute is set to `CK_TRUE`. It is very important to first set the required passphrase and then enable the CryptoServer to use slot-individual backup keys for protecting external key databases and their backups created with P11CAT and p11tool2.

ℹ *These attributes are described in more detail in Chapter 4.3.1, "Attributes for Global Configuration". See Chapter 4.3.3, “Attributes for Slot Configuration” for details about the CKA_CFG_SLOT_BACKUP_PASS_HASH attribute.*

7. Click the corresponding entry in the **Value** column to change the attribute value.

To apply your changes for the attributes CKA_CFG_AUTH_PLAIN_MASK and CKA_CFG_AUTH_KEYM_MASK press ENTER.

4.4 Setting up the External Keystore

If you want to use the external keystore you must enable this in the `cs_pkcs11_R2.cfg` configuration file and specify a path for it.

1. Open the `cs_pkcs11_R2.cfg` file for editing.

   You will find this configuration file on your computer, by default, here:

   `C:\Program Files\Utimaco\CryptoServer\Lib`

   In the configuration file section shown below you can specify whether generated keys are to be saved to an external keystore. When the PKCS#11 R2 implementation is supplied, this setting is deactivated.

```
# Created/Generated keys are stored in an external or internal database
KeyExternal = false
```

2. If you want to save keys to the external keystore, set the parameter `KeyExternal` to true.

```
KeyExternal = true
```

3. Next you must either enable or configure the external keystore path.
   If you want to enable the predefined path for the external keystore, delete the comment # for the relevant operating system that stands to the left of the predefined path `KeyStore  =`. You can now enter a different path for the external keystore, for example, for a Windows operating system.

Example:

```
#Path to the external keystore
#If KeyStore is defined the external keystore will be created and used #at
the defined location
# For Unix:
#KeyStore = /tmp/P11.pks
# For Windows:
KeyStore = C:/tmp/P11.pks
```

4.5 Generating Objects

In the PKCS#11 context, the word object is used to refer to data, keys and certificates. This chapter describes how you can use P11CAT to generate keys or key pairs.

You can generate these keys:

- Keys for which you can select the following mechanisms:
  - AES
  - DES
  - DES2
  - DES3

  You can use P11CAT to create your own attribute list for these keys. If you do not create your own attribute list for these keys, the keys will be generated with attributes taken from a standard template.
- Key pairs for which you can select the following mechanisms:
  - RSA
  - ECC

  You can use P11CAT to create your own attribute list for not only the public key but also the private key parts of these keys. If you do not create your own attribute list, the keys will be generated with attributes taken from a standard template.
- Keys for which you have created a template file.
  The mechanism and the attributes are taken from the template file for these keys.
- Key pairs for which you have created a template file.
  The mechanism and the attributes are taken from the template file for these keys.

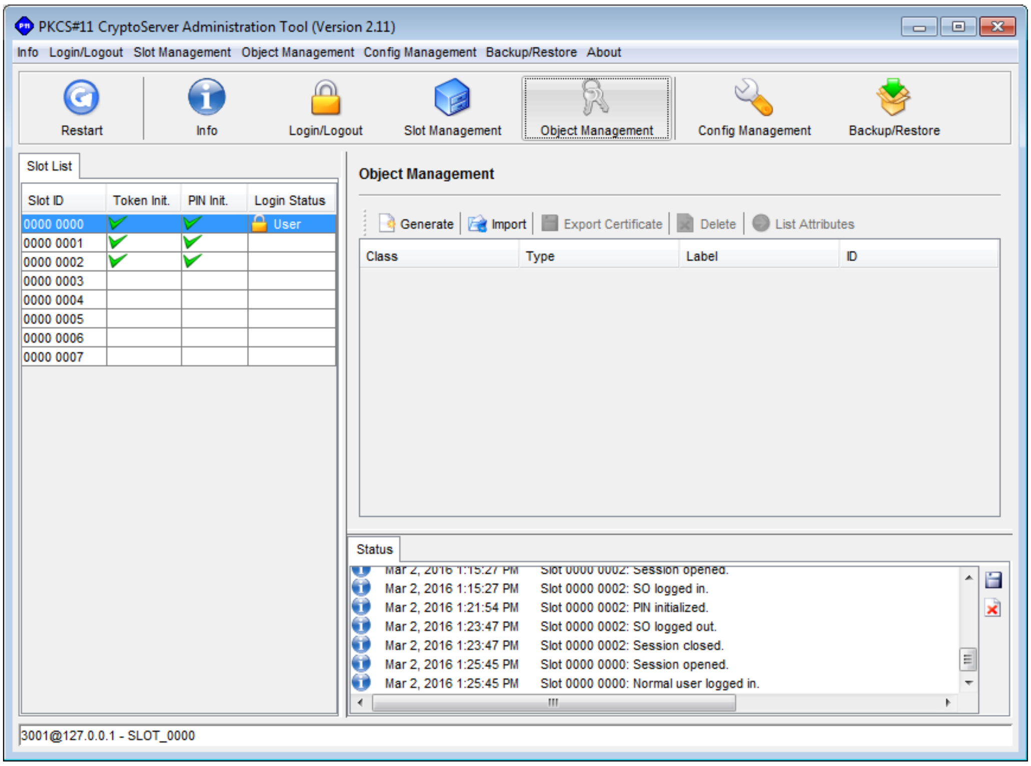
4.5.1 Generating a Key

You must log in to the slot as the User before you can generate a secret key.

This example shows how to generate a key with your own attribute list.

1. Click the relevant slot in the **Slot List**.
2. Click the **Login/Logout** button in the toolbar.
3. Click **Login User** in the functions area of the P11CAT main window.
4. Enter your password in the **Normal User PIN** text box.
5. Click the **Login** button in the functions area.

   The slot's login status is then displayed under **Login Status** with a lock icon and the user (in this case the `User`).
6. Click **Object Management** in the toolbar.



Slot ID	Token Init.	PIN Init.	Login Status
0000 0000	✓	✓	User
0000 0001	✓	✓	
0000 0002	✓	✓	
0000 0003			
0000 0004			
0000 0005			
0000 0006			
0000 0007			

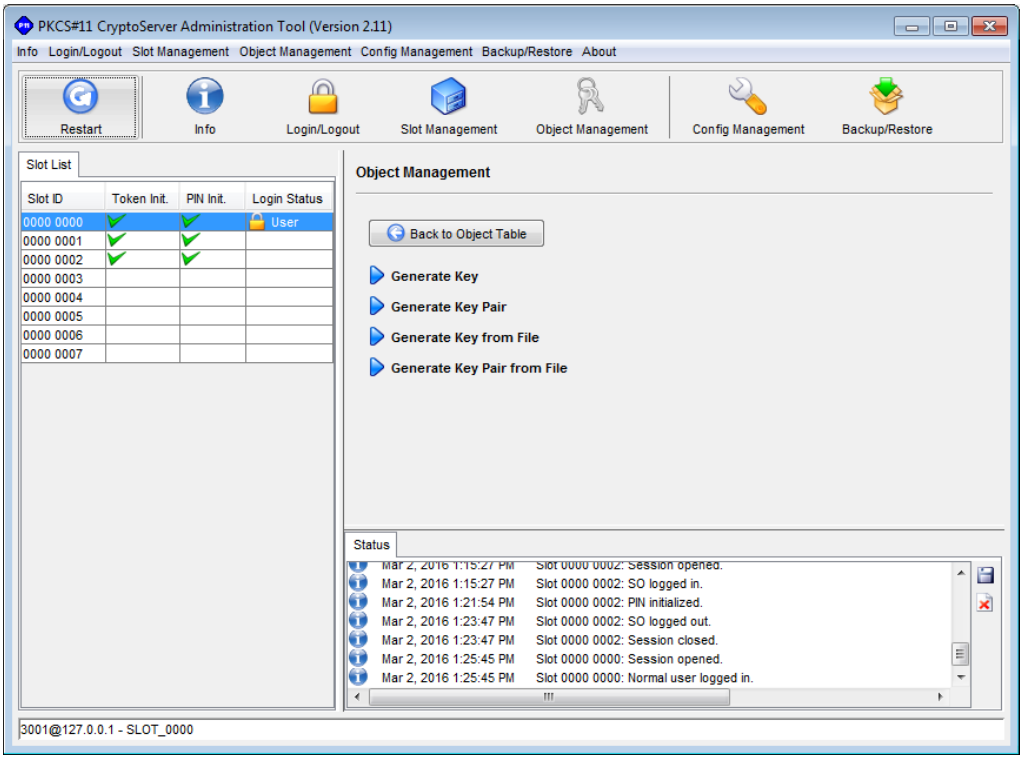
Class	Type	Label	ID

```
Mar 2, 2016 1:15:27 PM   Slot 0000 0002: Session opened.
Mar 2, 2016 1:15:27 PM   Slot 0000 0002: SO logged in.
Mar 2, 2016 1:21:54 PM   Slot 0000 0002: PIN initialized.
Mar 2, 2016 1:23:47 PM   Slot 0000 0002: SO logged out.
Mar 2, 2016 1:23:47 PM   Slot 0000 0002: Session closed.
Mar 2, 2016 1:25:45 PM   Slot 0000 0000: Session opened.
Mar 2, 2016 1:25:45 PM   Slot 0000 0000: Normal user logged in.
```

3001@127.0.0.1 - SLOT_0000

Figure 7: Main P11CAT window – Object Management

7. Click **Generate** in the **Object Management** area.



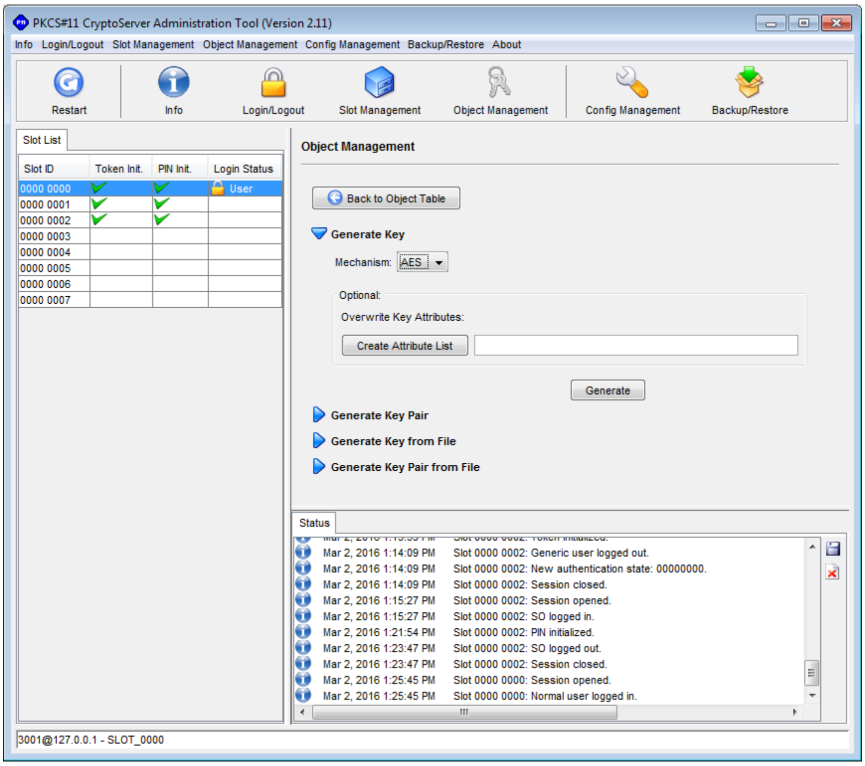
Slot ID	Token Init.	PIN Init.	Login Status
0000 0000	✓	✓	User
0000 0001	✓	✓	
0000 0002	✓	✓	
0000 0003			
0000 0004			
0000 0005			
0000 0006			
0000 0007			

```
Mar 2, 2016 1:15:27 PM   Slot 0000 0002: Session opened.
Mar 2, 2016 1:15:27 PM   Slot 0000 0002: SO logged in.
Mar 2, 2016 1:21:54 PM   Slot 0000 0002: PIN initialized.
Mar 2, 2016 1:23:47 PM   Slot 0000 0002: SO logged out.
Mar 2, 2016 1:23:47 PM   Slot 0000 0002: Session closed.
Mar 2, 2016 1:25:45 PM   Slot 0000 0000: Session opened.
Mar 2, 2016 1:25:45 PM   Slot 0000 0000: Normal user logged in.
```

3001@127.0.0.1 - SLOT_0000

Figure 8: Object Management functions in P11CAT

8. Click **Generate Key**.



Slot ID	Token Init.	PIN Init.	Login Status
0000 0000	✓	✓	User
0000 0001	✓	✓	
0000 0002	✓	✓	
0000 0003			
0000 0004			
0000 0005			
0000 0006			
0000 0007			

Mechanism: AES

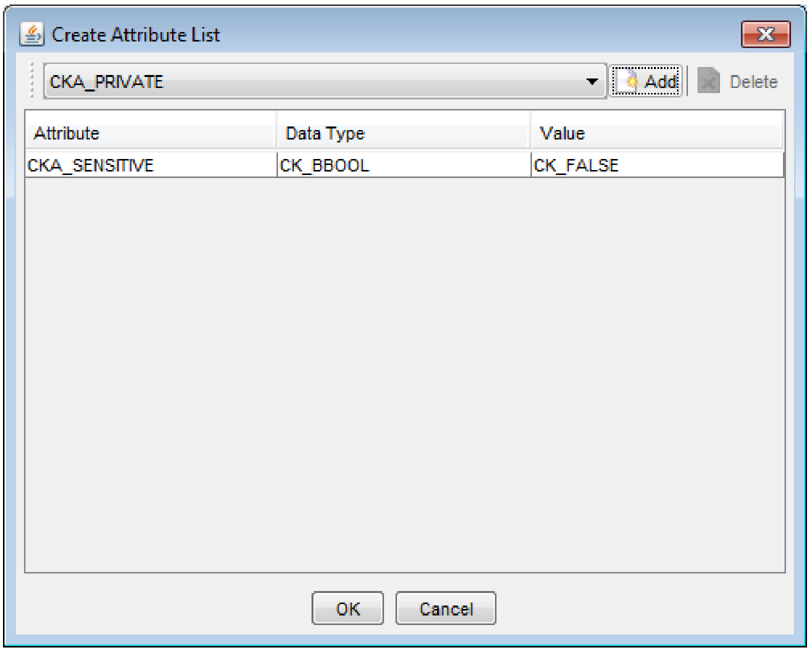
```
Mar 2, 2016 1:14:09 PM    Slot 0000 0002: Generic user logged out.
Mar 2, 2016 1:14:09 PM    Slot 0000 0002: New authentication state: 00000000.
Mar 2, 2016 1:14:09 PM    Slot 0000 0002: Session closed.
Mar 2, 2016 1:15:27 PM    Slot 0000 0002: Session opened.
Mar 2, 2016 1:15:27 PM    Slot 0000 0002: SO logged in.
Mar 2, 2016 1:21:54 PM    Slot 0000 0002: PIN initialized.
Mar 2, 2016 1:23:47 PM    Slot 0000 0002: SO logged out.
Mar 2, 2016 1:23:47 PM    Slot 0000 0002: Session closed.
Mar 2, 2016 1:25:45 PM    Slot 0000 0000: Session opened.
Mar 2, 2016 1:25:45 PM    Slot 0000 0000: Normal user logged in.
```

3001@127.0.0.1 - SLOT_0000

Figure 9: Key generation options in P11CAT

9. Select under **Mechanism** the encryption algorithm you require. The following mechanisms are available: `AES` (default), `DES`, `DES2` and `DES3`.

10. Click the **Create Attribute List** button if you want to change the list of key attributes. This opens a dialog box with the same name.



Attribute	Data Type	Value
CKA_SENSITIVE	CK_BBOOL	CK_FALSE

Figure 10: P11CAT dialog box for selecting the key attributes for a key

Here you can select the attributes you require for the key from the attribute list.

If you are not familiar with the PKCS#11 standard, we recommend using a default standard template with all the necessary attributes instead of an own attribute list.
At the end of this document you find a number of tables listing the attributes used in the standard templates.

11. To add an attribute to the list of key attributes, select the attribute you require in the drop-down box to the left of the **Add** button.
12. Click the **Add** button.
    The selected attribute now appears in the attribute list.

    This list displays the following information:

    - The attribute's name in the **Attribute** column (e.g. `CKA_SENSITIVE`).
    - The attribute's data type in the **Data Type** column (e.g. `CK_BBOOL`).
    - The attribute's value in the **Value** column (e.g., `CK_FALSE`). You can change the attribute value by clicking on the corresponding entry in the **Value** column and selecting the value you require.
13. End processing in this dialog box by clicking **OK**.

This closes the Create Attribute List dialog box. The selected attributes are displayed next to the Create Attribute List button.

Your own attribute list supplements or overwrites the standard attribute templates. If you have not selected enough attributes to meet the PKCS#11 standard for keys, the missing attributes are automatically added from the standard attribute templates. This ensures compliance with the PKCS#11 standard in every case.

14. Click the Generate button.

    This generates the key with the required attributes, which then appears in the Object Management list in the P11CAT main window.

4.5.2 Generating a Key Pair

You must log in to the slot as the User before you can generate a key pair (private and public).

This example shows how to generate a key pair with your own attribute list.

1. Click the slot you require in the Slot List.
2. Click the Login/Logout button in the toolbar.
3. Click Login User in the functions area of the P11CAT main window.
4. Enter your password in the Normal User PIN text box.
5. Click the Login button in the functions area.

   The slot's login status is then displayed under Login Status with a lock icon and the user (in this case the `User`).
6. Click Object Management in the toolbar.
7. Click the Generate button in the in the functions area. See Figure 8 above.
8. Click Generate Key Pair in the functions area.
9. Select in under Mechanism the encryption algorithm you require. The following are available: `RSA` (default) and `ECC`.

   When you generate a key pair you have the option of creating your own attribute list for the public key and for the private key.
10. Click the Create Attribute List button under Overwrite Public Key Attributes.
    This opens the Create Attribute List dialog box.

    In the Create Attribute List dialog box you can select the attributes you require for the public key from the attribute list.

If you are not familiar with the PKCS#11 standard, we recommend using a default standard template with all the necessary attributes instead of an own attribute list.
At the end of this document you find a number of tables listing the attributes used in the standard templates.

11. To add an attribute to the list of key attributes, select the attribute you require in the drop-down combo box to the left of the **Add** button. See Figure 10 above.
12. Click the **Add** button.
    The selected attribute now appears in the attribute list.

    This list displays the following information:

    - The attribute's name in the **Attribute** column (e.g. `CKA_SENSITIVE`).
    - The attribute's data type in the **Data Type** column (e.g. `CK_BBOOL`).
    - The attribute's value in the **Value** column (e.g. `CK_FALSE`).
13. Click the **Create Attribute List** button under **Overwrite Private Key Attributes**, and create an attribute list for the private key in the same way as you created an attribute list for the public key.
14. End processing in this dialog box by clicking **OK**.

    This closes the **Create Attribute List** dialog box. The selected attributes are displayed next to the **Create Attribute List** button.

Your own attribute list supplements or overwrites the standard attribute templates. If you have not selected enough attributes to meet the PKCS#11 standard for keys, the missing attributes are automatically added in from the standard attribute templates. This ensures compliance with the PKCS#11 standard in every case.

15. Click the **Generate** button.

The system generates the key pair, consisting of one private key and one public key, with the required attributes, and then displays it in the **Object Management** list in the P11CAT main window.

4.5.3 Generating a Key from a Template File

You can generate a secret key whose attributes and mechanism are to be taken from a template file.

1. Create the template file.

   This is how to create a template file, which in our example is for an AES key:

   ```
   [Mechanism]
   CK_MECHANISM_TYPE = CKM_AES_KEY_GEN

   [Key]
   CKA_CLASS = CKO_SECRET_KEY
   CKA_KEY_TYPE = CKK_AES
   CKA_TOKEN = CK_TRUE
   CKA_PRIVATE = CK_TRUE
   CKA_DECRYPT = CK_TRUE
   CKA_SIGN = CK_TRUE
   CKA_ENCRYPT = CK_TRUE
   CKA_VERIFY = CK_TRUE
   CKA_WRAP = CK_TRUE
   CKA_VALUE_LEN = 32
   CKA_LABEL = "My AES Secret Key"
   CKA_ID = 0x414553
   ```

2. Save your template file on your computer.

   You must log in to the slot as the User before you can generate a key from a template file.

3. Click the slot you require in the **Slot List**.
4. Click the **Login/Logout** button in the toolbar.
5. Click **Login User** in the functions area of the P11CAT main window
6. Enter your password in the **Normal User PIN** text box.
7. Click the **Login** button in the functions area.

   The slot's login status is then displayed under **Login Status** with a lock icon and the user (in this case the `User`).

8. Click the **Object Management** button in the toolbar.
9. Click the **Generate** button in the table under **Object Management**.
10. Click **Generate Key from File** in the functions area.
11. Click the **Generate Key from File** button to select the template file you created.

    Once you have selected your template file, the system automatically generates the key and then displays it in the **Object Management** list in the P11CAT main window.

4.5.4 Generating a Key Pair from a Template File

You can generate a key pair (private and public) whose attributes and mechanism are to be taken from a template file

1. Create a template file.

   This is how to create a template file, which in our example is for an RSA key:

```
[Mechanism]
CK_MECHANISM_TYPE = CKM_RSA_PKCS_KEY_PAIR_GEN

[PublicKey]
CKA_TOKEN = CK_TRUE
CKA_PRIVATE = CK_TRUE
CKA_ENCRYPT = CK_TRUE
CKA_VERIFY = CK_TRUE
CKA_WRAP = CK_TRUE
CKA_MODULUS_BITS = 1024
CKA_PUBLIC_EXPONENT = 0x010001
CKA_LABEL = "My RSA Public Key"
CKA_ID = 0x525341

[PrivateKey]
CKA_TOKEN = CK_TRUE
CKA_PRIVATE = CK_TRUE
CKA_SENSITIVE = CK_TRUE
CKA_DECRYPT = CK_TRUE
CKA_EXTRACTABLE = CK_TRUE
CKA_SIGN = CK_TRUE
CKA_UNWRAP = CK_TRUE
CKA_LABEL = "My RSA Private Key"
CKA_ID = RSA
```

2. Click the slot you require in the Slot List.
3. Click the Login/Logout button in the toolbar.
4. Click Login User in the functions area of the P11CAT main window.
5. Enter your password in the Normal User PIN text box.
6. Click the Login button in the functions area.

   The slot's login status is then displayed under Login Status with a lock icon and the user (in this case the `User`).
7. Click the Object Management button in the toolbar.
8. Click the Generate button in the table under Object Management.
9. Click Generate Key Pair from File.
10. Click the Generate Key Pair from File button to select the template file you created.

    Once you have selected your template file, the system automatically generates the key pair and then displays it in the Object Management list in the P11CAT main window.

4.5.5 Importing a PKCS#12 File

You can also import a PKCS#12 file. A PKCS#12 file contains a certificate, a private key and a public key, and is password-protected. If you want to import a PKCS#12 file, you can create your own attribute list for the certificate, and for the certificate's private and public keys.

This example shows how to import a PKCS#12 file and how to create your own attribute list for the certificate and for the public and private keys.

If you are not familiar with the PKCS#11 standard, we recommend importing the PKCS#12 file without attribute lists rather than creating your own attribute list for the certificate, and for the private and public keys. If you do not create your own attribute list, standard templates with all the necessary attributes are used instead.
At the end of this document you find a number of tables which list the attributes used in the standard templates.

1. Click the slot you require in the **Slot List**.
2. Click the **Login/Logout** button in the toolbar.
3. Click **Login User** in the functions area.
4. Enter your password in the **Normal User PIN** text box.
5. Click the **Login** button in the functions area.

   The slot's login status is then displayed under **Login Status** with a lock icon and the user (in this case the `User`).
6. In the toolbar, click the **Object Management** button.
7. Click the **Import** button in the functions area.
8. Click **Import PKCS#12**.
9. Click the **Browse** button to select the PKCS#12 file and enter the password if one is used to protect the PKCS#12 file.
10. Click the **Create Attribute List** button which you see under **Overwrite Certificate Attributes**.

    In the **Create Attribute List** dialog box you can now select the attribute you require for the certificate from the attributes list.
11. To do this, select the attribute you require in the combo box (to the left of the **Add** button).
12. Click the **Add** button.

    The selected attribute now appears in the attribute list.

    This list displays the following information:

    - The attribute's name in the **Attribute** column (e.g. `CKA_SENSITIVE`).
    - The attribute's data type in the **Data Type** column (e.g. `CK_BBOOL`).
    - The attribute's value in the **Value** column (e.g. `CK_FALSE`). You can change the attribute value by clicking on the corresponding entry in the **Value** column.

13. Click the Create Attribute List button under Overwrite Public Key Attributes, create an attribute list as you did when you created the certificate.
14. Click OK when you're finished.
15. Click the Create Attributes List button under Overwrite Private Key Attributes, create an attribute list in the same way as you did for a public key.
16. Click OK when you're finished.
17. Click the Import button. This imports the PKCS#12 file which then appears in the functions area of the P11CAT main window. Three entries are displayed: the certificate, the private key, and the public key.

4.5.6 Importing X.509 Certificates

You have the option of importing an X.509 certificate with its associated public key. You can create your own attribute list both for the certificate and also for the public key.

This example shows how to import an X.509 certificate and how to create your own attribute list for the certificate and the public key.

If you are not familiar with the PKCS#11 standard, we recommend importing the certificate without attribute lists rather than creating your own attribute list for the certificate and for the corresponding public key.
If you do not create your own attribute list, standard templates with all the necessary attributes are used instead.
At the end of this document you find a number of tables which list the attributes used in the standard templates.

1. Click the slot you require in the Slot List.
2. Click the Login/Logout button in the toolbar.
3. Click Login User and enter your password in the Normal User PIN text box.
4. Click the Login button in the functions area.

   The slot's login status is then displayed under Login Status with a lock icon and the user (in this case the `User`).
5. In the toolbar, click the Object Management button.
6. Click the Import button in the functions area.
7. Click Import Certificate.
8. Use the Browse button to select the certificate.

9. Click the Create Attribute List button which you see under Overwrite Certificate Attributes.

   In the Create Attribute List dialog box you can now select the attribute you require for the certificate from the attributes list.

10. To do this, select the attribute you require in the combo box (to the left of the Add button).

11. Click the Add button.

    The selected attribute now appears in the attribute list.

    This list displays the following information:

    ▣ The attribute's name in the Attribute column (e.g., `CKA_SENSITIVE`).

    ▣ The attribute's data type in the Data Type column (e.g., `CK_BBOOL`).

    ▣ The attribute's value in the Value column (e.g., `CK_FALSE`). You can change the attribute‘s value by clicking on the corresponding entry in the Value column.

12. Click the Create Attribute List button under Overwrite Public Key Attributes, and create an attribute list as you did when you created the certificate.

13. Click the OK button when you're finished.

14. Click the Import button.

    This imports the certificate and the public key which are now displayed in the table in the functions area of the P11CAT main window. Two entries are displayed: the certificate and the public key.

4.5.7 Deleting Objects from the CryptoServer

You must log in to the slot as the User before you can delete objects from the CryptoServer.

1. Click the slot you require in the Slot List.

2. Click the Login/Logout button in the toolbar.

3. Click Login User, and enter your password in the Normal User PIN text box.

4. Click the Login button in the functions area.

   The slot's login status is then displayed under Login Status with a lock icon and the user (in this case the `User`).

5. Click the Object Management button in the toolbar.

6. In the functions area, click the object you want to delete.

7. Click the Delete button.

4.5.8 Changing Object Attributes

You can change the attributes of the objects that are present in a slot on the CryptoServer.

If you are not familiar with the PKCS#11 standard, we recommend keeping the attributes of individual objects unchanged.

You must log in to the slot as the User before you can change object attributes.

1. Click the slot you require in the **Slot List**.
2. Click the **Login/Logout** button in the toolbar.
3. Click **Login User** in the functions area, and enter your password in the **Normal User PIN** text box.
4. Click the **Login** button.

   The slot's login status is then displayed under **Login Status** with a lock icon and the user (in this case the `User`).

5. Click the **Object Management** button in the toolbar.
6. Select an object in the table of objects displayed in the functions area and then click the **List Attributes** button.

   You can also open the **Attribute List** by double-clicking on the appropriate object.

   You can change the attribute in the **Value** column.

7. When you've finished editing the **Attribute List**, click the **Back to Object Table** button.

4.6 Backup/Restore

You can create a backup of the objects in the slots (data, keys and certificates), and of the slot's configuration, and restore them when required.

The User is responsible for backing up and restoring the objects.

The Security Officer (SO) is responsible for backing up and restoring the slot configuration.

You can display information about a backup.

Backups can be created for the internal and external keys.

The Master Backup Key (MBK) must have been imported to the CryptoServer before a backup of the objects or the slot configuration can be created. All the data, and therefore also all the objects and the slot configuration, are encrypted with the Master Backup Key (MBK) before the data can be copied from the CryptoServer to a backup directory. This applies both to the backups of internal and of external keys.

If you want to restore the objects or the slot configuration, the same MBK as was used to encrypt or create the objects must be loaded in the CryptoServer.

4.6.1 Creating a Backup of All Keys in a Slot

You must log in to the CryptoServer as the User before you can create a backup of all keys in the specified slot.

1. In the **Slot List**, click the slot you require.
2. Click the **Login/Logout** button in the toolbar.
3. Click **Login User** in the functions area, and enter your password in the **Normal User PIN** text box.
4. Click the **Login** button in the functions area.

   The login status for the specific slot is then displayed under **Login Status** with a lock icon and the user (in this case the `User`).
5. In the toolbar, click the **Backup/Restore** button.
6. Then select **Backup/Restore Keys** in the functions area.
7. Click the **Backup Internal Keys** button to create a backup of internal keys or the **Backup External Keys** button to create a backup of external keys.
8. In the **Save Backup of All Internal Keys** or **Save Backup of All External Keys** dialog box, define a unique name for the backup.

   The backup is created and stored on your computer, by default, in the `C:\Program Files\Utimaco\CryptoServer\Administration` directory.

4.6.2 Restoring a Backup of All Keys in a Slot

To ensure, you can restore an object backup successfully, the same MBK as was used to encrypt or back up the objects must be loaded in the CryptoServer.

You have the option of performing a restore either in the internal keystore within the CryptoServer or in the external keystore outside of the CryptoServer. You must log in to the CryptoServer as the User before you can restore the keys.

1. Click the slot you require in the **Slot List**.
2. Click the **Login/Logout** button in the toolbar.
3. Click **Login User** in the functions area, and enter your password in the **Normal User PIN** text box.
4. Click the **Login** button in the functions area of the P11CAT main window.

   The login status for the specific slot is then displayed under **Login Status** with a lock icon and the user (in this case the `User`).
5. In the toolbar, click **Backup/Restore**.
6. Click **Backup/Restore Keys** in the functions area.
7. Click **Restore Key Backup to Internal Key Store** if you want to perform a restore in the CryptoServer's internal keystore. Alternatively, click **Restore Key Backup to External Key Store** if you want to perform a restore in an external keystore.
8. In either the **Select Key Backup to Be Restored to Internal Key Store** or **Select Key Backup to Be Restored to External Key Store** dialog boxes, select the required backup file.
9. Click **Open** to confirm the selection.

   The selected backup is then either loaded into the CryptoServer's internal or the CryptoServer's external memory.

4.6.3 Creating a Slot Configuration Backup

The MBK must have been imported in the CryptoServer before you can back up the slot configuration.

Only a Security Officer (SO) can create a backup of the slot configuration.

1. Click the relevant slot in the **Slot List**.
2. Click the **Login/Logout** button in the toolbar, and then click **Login SO** in the functions area.
3. Enter a password for the Security Officer (SO) in the **SO PIN** text box, and then confirm the input by clicking the **Login** button.

   In the **Slot List**, the login status is indicated by a lock and the entry `SO` in the **Login Status** column.
4. In the toolbar, click the **Backup/Restore** button.
5. Click **Backup/Restore Config** in the functions area.
6. Click the **Backup Slot configuration Object** button.
7. Enter a unique name for the backup in the **Save Slot Configuration Backup As** dialog box.
8. Finish by clicking the **Save** button.

The backup is created and stored on your computer, by default, in the `C:\Program Files\Utimaco\CryptoServer\Administration` directory.

4.6.4 Restoring a Slot Configuration Backup

To ensure you can restore a slot configuration backup successfully, the same MBK as was used to encrypt or back up the slot configuration must be loaded in the CryptoServer.

Only a Security Officer (SO) can restore the slot configuration.

1. Click the relevant slot in the **Slot List**.
2. Click the **Login/Logout** button in the toolbar, and then click **Login SO** in the functions area.
3. Enter a password for the Security Officer (SO) in the **SO PIN** text box, and then confirm the input by clicking the **Login** button.

   In the **Slot List**, the login status is indicated by a lock and the entry `SO` in the **Login Status** column.
4. In the toolbar, click the **Backup/Restore** button and then click **Backup/Restore Config** in the functions area.
5. Click the **Restore Slot Configuration Object** button in the functions area.
6. Select the required backup in the **Select Slot Configuration Backup to Be Restored** dialog box, and then click the **Open** button when you've finished editing in this dialog box.

   The selected backup is imported.

4.6.5 Viewing Backup Information

You can display information about backups of internal and external keys, and also of slot configurations. You do not need to be logged in to the CryptoServer to see this information.

1. In the P11CAT toolbar, click the **Backup/Restore** button.
2. Select **Get Backup Info**.
3. Click the **Browse** button. The **Select a Backup File** dialog box opens.
4. Click the **Open** button to display the backup you require on your computer.
5. Then click the **Show Backup Info** button in the functions area.

   You can display this information for a key or slot configuration backup:

- File version

  Version number of the backup file

- Creation date

  Time at which the backup was created (YYYYMMDD HHMMSS)

- Slot ID
- Slot description

  The address of the device and the slot number, e.g., `3001@127.0.0.1 – SLOT_0000` (displayed in the status bar of the P11CAT main window)

- Object count

  Number and content of the objects involved in the backup [configuration object(s), external key(s), internal key(s)]

4.7 The Restart Button

You can use the **Restart** button in the toolbar to reinitialize the PKCS#11 API. When you do this, all the CryptoServer users (the Administrator, Security Officer (SO) and the User) are logged off from the CryptoServer, the Secure Messaging connection is closed, and the `cs_pkcs11_R2.cfg` configuration file is reimported.

4.8 The Info Button

Click the **Info** button to call up the following information:

- Cryptoki info
  Displays information about which PKCS#11 API (called the Cryptoki in the PKCS#11 standard) is being used.
- Slot Info
  Displays information about the slot.
- Token Info
  Displays information about the token.

4.9 The About Menu Item

Click the **About** menu item in the menu bar and the sub menu item with the same name to call up information about the P11CAT version, the manufacturer, and license information about the icons that are being used. Click the **Version Information** sub menu item to call up more detailed information about P11CAT.

5 Advanced Administration

This chapter describes advanced administration functions. However, not all the administration tasks listed in this chapter can be performed using P11CAT.

5.1 Editing the cs_pkcs11_R2.cfg Configuration File

This chapter will help you familiarize yourself with additional options for configuring the `cs_pkcs11_R2.cfg` configuration file. This is the file you opened after you performed installation.

1. Close P11CAT.
2. Open the `cs_pkcs11_R2.cfg` file for editing.

   You find this configuration file on your computer, by default, here:

   `C:\Program Files\Utimaco\CryptoServer\Lib`

   Under `[Global]`, in this extract from the configuration file, you can see where the logfile is stored in UNIX or Windows systems, if you enable logging and delete the comments (#) that stand to the left of the paths. You have the option of changing the path used for logging, for both operating systems.

```
 [Global]
# Path to the logfile (name of logfile is attached by the API)
# For unix:
#Logpath = /tmp
# For windows:
#Logpath = c:/tmp

# Loglevel (0 = NONE; 1 = ERROR; 2 = WARNING; 3 = INFO; 4 = TRACE)
Logging = 0
# Maximum size of the logfile in bytes (file is rotated with an backup file if
full)
Logsize = 1000000
```

   The default setting for `Loglevel = 0` is `NONE`.

3. If you want to enable logging, you must delete the comment # to the left of the `Logpath` paths for the operating system that is being used.

Example:

```
# For windows:
Logpath = c:/tmp
```

4. You must also enter a log level after `Logging`.

Example:

```
# Loglevel (0 = NONE; 1 = ERROR; 2 = WARNING; 3 = INFO; 4 = TRACE)
Logging = 3
```

In the table below you will find an overview of the different log levels and their individual meanings.

Logging	*Name*	*Description*
0	NONE	No entries are written to the logfile.
1	ERROR	Errors are recorded in the logfile.
2	WARNING	Errors and also warnings are recorded in the logfile.
3	INFO	Errors, warnings, and other additional information, are recorded in the logfile.
4	TRACE	All the events are recorded in the logfile. If you set TRACE as the log level for logging it may, in some situations, cause a system overload. For this reason, we recommend using this log level only for short-term problem analyses.

Table 4: Logging levels

5. You can specify the size of the logfile by using the parameter `Logsize` as required. You can enter values in `kb`, `mb` and `gb`. Examples 10000 kb, 10 mb or 1 gb.

   The configuration file section shown below illustrates how the logfile reacts when it exceeds the defined `Logsize` of `10MB`.

```
# Maximum size of the logfile in bytes (file is rotated with an backupfile if full)
Logsize = 10mb
```

   When the logfile reaches its maximum size it is saved in the same directory as a backup with the file extension `.bak`.

   A second logfile is created. When this second file also reaches its maximum size, it too is saved as a backup and the backup of the first logfile is overwritten.

6. Define whether you want to save generated keys to the external keystore by using the parameter `KeyExternal`. When PKCS#11 R2 is supplied, this setting is deactivated.

   Example (default):

```
# Created/Generated keys are stored in an external or internal database
KeysExternal = false
```

   If you want to save these keys to the external keystore, set `KeyExternal = true`.

7. Next you must either enable or configure the external keystore path.

```
#Path to the external keystore
# If KeyStore is defined the external keystore will be created and used at the
defined location
# For unix:
#KeyStore = /tmp/P11.pks
```

```
# For windows:
#KeyStore = C:/tmp/P11.pks
```

If you want to enable the predefined path for the external keystore, delete the comment # for the relevant operating system that stands to the left of the predefined path, `KeyStore =`. You can now enter a different path for the external keystore, for example, for the operating system you are using.

8. In the configuration file section shown below you can specify the number of slots that should be available to the CryptoServer. The CryptoServer has from 0 to 1000 slots available.

```
# Maximum number of slots that can be used
SlotCount = 3
```

9. In the configuration file section shown below you can see that a connection (Secure Messaging connection) should be kept active for 15 minutes. `False` means that an inactive connection is closed after 15 minutes for security reasons. If you want a connection to be kept active continuously, you must set `KeepAlive = true`.

```
# Prevents session from expiring after 15 minutes inactivity
KeepAlive = false
```

10. In the configuration file section shown below you can see how long an established connection to the CryptoServer is to be kept alive if the CryptoServer does not respond. The default setting is 5000 milliseconds, but you can change this to suit your own requirements.

```
# Timeout of the open connection command in ms
ConnectionTimeout = 5000
```

11. In the configuration file section shown below you can see how long the system is to wait for a response from CryptoServer (in milliseconds) after it receives a command. The default setting is 60000 milliseconds, but you can change this to suit your own requirements.

```
# Timeout of command execution in ms
CommandTimeout = 60000
```

12. In the configuration file section shown below you can enable the device address for the specified operating systems if you have purchased a CryptoServer PCI or PCIe card. You must enter the device address exactly as it is shown here for both operating systems. Simply delete the comment # to the left of `Device =` for the relevant operating system. In the section from the configuration file shown next you must enter a comment # to the left of `Device` (i. e. comment it out) because otherwise two device addresses would be fetched.

```
#[CryptoServer]
# Device specifier (here: CryptoServer is internal PCI device)
# For unix:
#Device = /dev/cs2
# For windows:
```

```
#Device = PCI:0
[CryptoServer]
# Device specifier (here: CryptoServer is CSLAN with IP address 192.168.0.1)
Device = 192.168.0.1
```

13. The configuration file section shown above is an example of a *logical failover device* for a PKCS#11 application with two IP addresses for two CryptoServer LAN.

```
#[CryptoServer]
# Device specifier (here: CryptoServer is logical failover device of CSLANs with IP
address 192.168.0.2 and IP address 192.168.0.3)
#Device = {192.168.0.2 192.168.0.3}
```

The default setting is for failover in the configuration file to be disabled.

If you want to use P11CAT to set up a logical failover device, you must not use the configuration file section shown above.

14. In the configuration file shown above you can see the number of the slot for which the global configuration will be overwritten. In the default setting this section is deactivated. If you want to change the global setting for the specified slot, you must delete the comment # that stands to the left of `SlotNumber = 0`. You can then, for example, define the attributes for this slot in this configuration file. You can also permit and enable global configuration for other slots here.

```
#[Slot]
# Slotsection for slot with number 0
#SlotNumber = 0
```

15. Start P11CAT. The changes you made in the `cs_pkcs11_R2.cfg` configuration file are now applied.

5.2 Setting Up Additional CryptoServers

If you want to use a PKCS#11 application to administer and use a *logical failover device*, you must enter additional CryptoServer addresses in the `cs_pkcs11_r2.cfg` configuration file. P11CAT allows you to use individual CryptoServer in a targeted way. This example illustrates how to set up a second CryptoServer and the issues you need to be aware of when doing so.

From CryptoServer LAN Version 4.2.0 onwards the Internet Protocols IPv4 and IPv6 are supported.

This example uses a CryptoServer LAN with the IP address 192.168.0.2 and, as a second CryptoServer, implemented as a software for simulating the CryptoServer's behavior, and called the CryptoServer simulator with the address 3001@127.0.0.1.

1. Close P11CAT.
2. Open the `cs_pkcs11_R2.cfg` file for editing.

   You will find this configuration file on your computer on this path:

   `C:\Program Files\Utimaco\CryptoServer\Lib`

   You will see this entry in the lower part of the configuration file:

```
[CryptoServer]
# Device specifier (here: CryptoServer is CSLAN with IP address 192.168.0.1)
Device = 192.168.0.1
```

3. Now enter the IP address 192.168.0.2 after `Device`.
4. Copy the section shown above, which includes the IP address for the CryptoServer LAN, and paste it underneath, in the configuration file.

```
[CryptoServer]
# Device specifier (here: CryptoServer is CSLAN with IP address 192.168.0.1)
Device = 192.168.0.2

[CryptoServer]
# Device specifier (here: CryptoServer is CSLAN with IP address 192.168.0.1)
Device = 3001@127.0.0.1
```

5. Now enter the address of the CryptoServer simulator to the right of `Device` in the copied section, as shown above. This address is only used here as an example.
6. Save and close the configuration file.
7. Start P11CAT. The changes you made in the `cs_pkcs11_R2.cfg` configuration file are now applied.

If your P11CAT has not been shut down, simply click the ***Restart*** *button in the toolbar.*

For the CryptoServer LAN with IP address 192.168.0.2 three slots are displayed in the Slot List because of setting `SlotCount = 3` with the following Slot IDs:

`0000 0000`

`0000 0001`

`0000 0002`

For the CryptoServer simulator with address 3001@127.0.0.1 ten slots are displayed in the Slot List, by default, setting `SlotCount = 10` with the following Slot IDs:

`0001 0000`

`0001 0001`

`0001 0002`

You can restrict individually the maximum number of slots per device (`SlotCount`).

The first 4 numbers of the Slot ID identify which CryptoServer is involved. When you enable a slot in the Slot List, the associated CryptoServer's address is displayed in the lower left-hand part of the main window.

The last 4 numbers of the Slot ID identify the slot. The slots are numbered sequentially and identically within the CryptoServer.

5.3 Synchronizing CryptoServer Data

If one of the CryptoServer crashes, the second CryptoServer can be used instead. However, you will need to synchronize the data on the two CryptoServer manually. You should repeat this procedure every time data on one of the two CryptoServer is changed, or when new data is added. At this point you must note the following:

- This function is available from P11CAT version 2.06.
- The same MBK must be loaded in both CryptoServer.
- The Backup/Restore functions are used to synchronize the data in the two CryptoServer.

When you use the Backup/Restore functions to transfer data from one CryptoServer to the another CryptoServer, you must be aware that a backup of a slot with ID 0001 can only be imported as a restore to a slot with ID 0001. In other words, the last 4 numbers of the Slot ID must be identical.

The User is responsible for backing up/restoring keys and the Security Officer (SO) is responsible for backing up/restoring the slot configuration.

1. You must ensure that the same MBK is loaded in the CryptoServer you are using. You will find all the information required to import an MBK to the CryptoServer in the [CSMSADM].
2. You also need to create backups of all the data (key and slot configuration) for the required CryptoServer slots.
   You will find out how to do this in this manual, in sections 4.6.1 "Creating a Backup" and 4.6.3 "Creating a Slot Configuration Backup".

   We recommend giving the backup a name which specifies which slot the backup was prepared from, and what data it contains (key or configuration).

Examples:

`BackupConfig_0000_0001` or `Backup_Key_0000_0001`.

3. Restore the key backups and the configuration backup in the required CryptoServer slot. Ensure that the slot IDs are identical. A backup from a CryptoServer slot with the ID 0000 0001 can only be restored in a CryptoServer slot with the ID 0001 0001. In other words, the last 4 numbers of the Slot ID must be identical.

6 Contact Address for Support Queries

Please feel free to contact us if an error occurs while operating the CryptoServer LAN or the CryptoServer PCIe plug-in card, or if you have further questions on CryptoServer LAN or the CryptoServer PCIe plug-in card.

Utimaco IS GmbH
Germanusstr. 4
52080 Aachen
Germany

You can reach us from Monday to Friday 09.00 a.m. to 05.00 p.m., apart from public holidays and other celebrations, under the following phone/fax number or e-mail address:

Phone:	+49 (0) 241 / 1696-153
Fax:	+49 (0) 241 / 1696-58153
e-mail:	support-cs@utimaco.com

Appendix A Standard Templates for Attributes

The tables below list the standard templates used when certificates are imported, and private and public keys are generated and imported, without a specific attribute list being created.

A.1 Templates for the Generation of Secret Keys

The templates listed below, with their attributes, are used for the available AES, DES, DES2 and DES3 mechanisms when a secret key with attributes taken from a standard template is generated.

A.1.1 AES Keys

Attribute	*Value*
`CKA_TOKEN`	`CK_TRUE`
`CKA_PRIVATE`	`CK_TRUE`
`CKA_DECRYPT`	`CK_TRUE`
`CKA_SIGN`	`CK_TRUE`
`CKA_ENCRYPT`	`CK_TRUE`
`CKA_WRAP`	`CK_TRUE`
`CKA_LABEL`	`"AES Secret Key"`
`CKA_VALUE_LEN`	`32`

Table 5: Standard template for AES keys

A.1.2 DES Keys

Attribute	*Value*
`CKA_TOKEN`	`CK_TRUE`
`CKA_PRIVATE`	`CK_TRUE`
`CKA_DECRYPT`	`CK_TRUE`
`CKA_SIGN`	`CK_TRUE`
`CKA_ENCRYPT`	`CK_TRUE`
`CKA_WRAP`	`CK_TRUE`
`CKA_LABEL`	`"DES Secret Key"`

Table 6: Standard template for DES keys

A.1.3 DES2 Keys

Attribute	*Value*
`CKA_TOKEN`	`CK_TRUE`
`CKA_PRIVATE`	`CK_TRUE`
`CKA_DECRYPT`	`CK_TRUE`
`CKA_SIGN`	`CK_TRUE`
`CKA_ENCRYPT`	`CK_TRUE`
`CKA_WRAP`	`CK_TRUE`
`CKA_LABEL`	`"DES2 Secret Key"`

Table 7: Standard template for DES2 keys

A.1.4 DES3 Keys

Attribute	*Value*
`CKA_TOKEN`	`CK_TRUE`
`CKA_PRIVATE`	`CK_TRUE`
`CKA_DECRYPT`	`CK_TRUE`
`CKA_SIGN`	`CK_TRUE`
`CKA_ENCRYPT`	`CK_TRUE`
`CKA_WRAP`	`CK_TRUE`
`CKA_LABEL`	`"DES3 Secret Key"`

Table 8: Standard template for DES3 keys

A.2 Templates for the Generation of Key Pairs (Private and Public)

The templates listed below, with their attributes, are used for the available RSA and ECC mechanisms when a key pair, consisting of one private key and one public key, is generated with attributes taken from a standard template:

A.2.1 Public RSA Keys

Attribute	*Value*
`CKA_TOKEN`	`CK_TRUE`
`CKA_PRIVATE`	`CK_TRUE`
`CKA_VERIFY`	`CK_TRUE`
`CKA_ENCRYPT`	`CK_TRUE`

Attribute	*Value*
`CKA_WRAP`	`CK_TRUE`
`CKA_MODULUS_BITS`	`1024`
`CKA_PUBLIC_EXPONENT`	`0x010001`
`CKA_LABEL`	`"RSA Public Key"`

Table 9: Standard template for public RSA keys

A.2.2 Private RSA Keys

Attribute	*Value*
`CKA_TOKEN`	`CK_TRUE`
`CKA_PRIVATE`	`CK_TRUE`
`CKA_SENSITIVE`	`CK_TRUE`
`CKA_EXTRACTABLE`	`CK_TRUE`
`CKA_SIGN`	`CK_TRUE`
`CKA_DECRYPT`	`CK_TRUE`
`CKA_UNWRAP`	`CK_TRUE`
`CKA_LABEL`	`"RSA Private Key"`

Table 10: Standard template for private RSA keys

A.2.3 Public ECC Keys

Attribute	*Value*
`CKA_TOKEN`	`CK_TRUE`
`CKA_PRIVATE`	`CK_TRUE`
`CKA_VERIFY`	`CK_TRUE`
`CKA_EC_PARAMS`	`"secp192r1"`
`CKA_LABEL`	`"ECC Public Key"`

Table 11: Standard template for public ECC keys

A.2.4 Private ECC Keys

Attribute	*Value*
`CKA_TOKEN`	`CK_TRUE`
`CKA_PRIVATE`	`CK_TRUE`
`CKA_SENSITIVE`	`CK_TRUE`

Attribute	*Value*
CKA_EXTRACTABLE	CK_TRUE
CKA_SIGN	CK_TRUE
CKA_LABEL	"ECC Private Key"

Table 12: Standard template for private ECC keys

A.3 Templates for the Import of Certificates

The templates listed below, with their attributes, are used to import an X.509 certificate:

A.3.1 X.509 Certificates

Attribute	*Value*
CKA_CLASS	CKO_CERTIFICATE
CKA_TOKEN	CK_TRUE
CKA_PRIVATE	CK_TRUE
CKA_CERTIFICATE_TYPE	CKC_X_509
CKA_LABEL	"X509 Certificate"
CKA_VALUE	Parsed value from file
CKA_SUBJECT	Parsed subject name from file if set or NULL

Table 13: Standard template for X.509 certificates

A.3.2 Public RSA Keys Used in X.509 Certificates

Attribute	*Value*
CKA_CLASS	CKO_PUBLIC_KEY
CKA_TOKEN	CK_TRUE
CKA_PRIVATE	CK_TRUE
CKA_VERIFY	CK_TRUE
CKA_KEY_TYPE	CKK_RSA
CKA_WRAP	CK_TRUE
CKA_LABEL	"RSA Public Key"
CKA_MODULUS	Parsed from file
CKA_PUBLIC_EXPONENT	Parsed from file

Attribute	*Value*
CKA_SUBJECT	Parsed certificate subject name from file if set or NULL

Table 14: Standard template for public RSA keys used in X.509 certificates

A.3.3 Private RSA Keys Used in X.509 Certificates

Attribute	*Value*
CKA_CLASS	CKO_PRIVATE_KEY
CKA_TOKEN	CK_TRUE
CKA_PRIVATE	CK_TRUE
CKA_SIGN	CK_TRUE
CKA_EXTRACTABLE	CK_TRUE
CKA_SENSITIVE	CK_TRUE
CKA_KEY_TYPE	CKK_RSA
CKA_LABEL	"RSA Private Key"
CKA_MODULUS	Parsed from file
CKA_PUBLIC_EXPONENT	Parsed from file
CKA_PRIVATE_EXPONENT	Parsed from file
CKA_PRIME_1	Parsed from file
CKA_PRIME_2	Parsed from file
CKA_COEFFICIENT	Parsed from file
CKA_EXPONENT_1	Parsed from file
CKA_EXPONENT_2	Parsed from file
CKA_SUBJECT	Parsed certificate subject name from file if set or NULL

Table 15: Standard template for private RSA keys used in X.509 certificates

A.3.4 Public DSA Keys Used in X.509 Certificates

Attribute	*Value*
CKA_CLASS	CKO_PUBLIC_KEY
CKA_TOKEN	CK_TRUE
CKA_PRIVATE	CK_TRUE
CKA_VERIFY	CK_TRUE

Attribute	*Value*
CKA_KEY_TYPE	CKK_DSA
CKA_LABEL	"DSA Public Key"
CKA_PRIME	Parsed from file
CKA_SUBPRIME	Parsed from file
CKA_BASE	Parsed from file
CKA_VALUE	Parsed from file
CKA_SUBJECT	Parsed certificate subject name from file if set or NULL

Table 16: Standard template for public DSA keys used in X.509 certificates

A.3.5 Private DSA Keys Used in X.509 Certificates

Attribute	*Value*
CKA_CLASS	CKO_PRIVATE_KEY
CKA_TOKEN	CK_TRUE
CKA_PRIVATE	CK_TRUE
CKA_SIGN	CK_TRUE
CKA_EXTRACTABLE	CK_TRUE
CKA_SENSITIVE	CK_TRUE
CKA_KEY_TYPE	CKK_DSA
CKA_LABEL	"DSA Private Key"
CKA_PRIME	Parsed from file
CKA_SUBPRIME	Parsed from file
CKA_BASE	Parsed from file
CKA_VALUE	Parsed from file
CKA_SUBJECT	Parsed certificate subject name from file if set or NULL

Table 17: Standard template for private DSA keys used in X.509 certificates

A.3.6 Public ECC Keys Used in X.509 Certificates

Attribute	*Value*
CKA_CLASS	CKO_PUBLIC_KEY
CKA_TOKEN	CK_TRUE

Attribute	*Value*
`CKA_PRIVATE`	`CK_TRUE`
`CKA_VERIFY`	`CK_TRUE`
`CKA_KEY_TYPE`	`CKK_EC`
`CKA_LABEL`	`"ECC Public Key"`
`CKA_ECDSA_PARAMS`	Parsed from file
`CKA_EC_POINT`	Parsed from file
`CKA_SUBJECT`	Parsed certificate subject name from file if set or `NULL`

Table 18: Standard template for public ECC keys used in X.509 certificates

A.3.7 Private ECC Keys Used in X.509 Certificates

Attribute	*Value*
`CKA_CLASS`	`CKO_PRIVATE_KEY`
`CKA_TOKEN`	`CK_TRUE`
`CKA_PRIVATE`	`CK_TRUE`
`CKA_SIGN`	`CK_TRUE`
`CKA_EXTRACTABLE`	`CK_TRUE`
`CKA_SENSITIVE`	`CK_TRUE`
`CKA_KEY_TYPE`	`CKK_EC`
`CKA_LABEL`	`"ECC Private Key"`
`CKA_ECDSA_PARAMS`	Parsed from file
`CKA_VALUE`	Parsed from file
`CKA_SUBJECT`	Parsed certificate subject name from file if set or `NULL`

Table 19: Standard template for private ECC keys used in X.509 certificates

References

Reference	*Title/Company*	*Document No.*
[CSADMIN]	CryptoServer LAN/CryptoServer - CryptoServer Command-line Administration Tool – csadm – Manual for System Administrators/Utimaco IS GmbH	2009-0003
[CSMSADM]	CryptoServer Manual for System Administrators/Utimaco IS GmbH.	M010-0001-en
[CS_PKCS11DEV]	PKCS#11 R2 Developer Guide/Utimaco IS GmbH.	2012-0007
[PKCS11ICMS]	"PKCS #11 Cryptographic Token Interface Current Mechanisms Specification Version 2.40," Committee Specification 01, September 16, 2014/OASIS Standard. Available: http://docs.oasis-open.org/pkcs11/pkcs11-curr/v2.40/cs01/pkcs11-curr-v2.40-cs01.html	