



**Charismathics Smart Security Interface™**

**V4.9**

**Manual**

## Contents

1	Preface .....	5
2	About this Manual .....	7
3	Installation.....	8
3.1	Installation Requirements .....	8
3.2	Supported Smart Cards and TPM Chips .....	9
3.3	Tested Card Readers .....	2
3.4	Secure Pin Entry (SPE) .....	11
3.5	Unattended Installation.....	11
4	Administration Tool: Charismathics Security Token Configurator .....	12
4.1	User Interface .....	12
4.1.1	Manager Menu .....	12
4.1.2	Edit menu/ Context menu .....	13
4.1.3	Token menu.....	13
4.1.4	TPM menu .....	14
4.1.5	Key Pair menu .....	15
4.1.6	Certificate menu .....	16
4.1.7	Info menu.....	17
4.2	Changing PINs .....	17
4.3	Unlocking Tokens .....	18
4.4	Generating and Importing Keys.....	19
4.4.1	Generation of a Key Pair .....	19
4.4.2	Importing a Key Pair .....	19
4.4.3	Generation of a Secret Key .....	19
4.4.4	Importing a Secret Key .....	20
4.5	Generating and Importing Certificates.....	20
4.5.1	Generating Self-Signed Certificates and Certificate Requests .....	20
4.5.2	Importing Certificates.....	21
4.6	Creating Profiles .....	21
4.6.1	Smart Card with Profile.....	21
4.6.2	Empty Smart Card .....	21
4.7	Preparing a Token (Initialization and Personalization) .....	22
4.7.1	First Step: Creating a Profile (Initialization) .....	22
4.7.2	Second Step: Creating Keys and Certificates (Personalization) .....	22
4.8	TPM Management .....	22
4.8.1	Operations on TPM without owner .....	23
4.8.2	Changing the current TPM Owner Password.....	23
4.8.3	Creating TPM User .....	23
4.8.4	Deleting TPM User .....	23

---

4.8.5	Inspect TPM User private information .....	24
4.8.6	Changing the TPM User Password .....	24
4.8.7	Importing a key pair from a PFX file .....	24
4.9	Further Functions .....	24
4.9.1	Directory "Certificates" .....	24
4.9.2	Directory "Data" .....	25
4.9.3	Function "Open Token" .....	25
4.9.4	Function "Delete all" and "Delete Object" .....	25
4.9.5	Function "Set Default Container" .....	25
4.9.6	Function "Change Container Label" .....	26
4.9.7	Function "Show Certificate" .....	26
4.9.8	Function "Export Certificate" .....	26
4.9.9	Function "Register Certificate" .....	26
4.9.10	Function "Check Private Key" .....	26
4.9.11	Function "Check Secret Key" .....	27
5	User Tool: Charismathics Smart Security Interface Utility .....	29
5.1	Change PIN .....	29
5.2	Unlock PIN .....	30
5.3	Change Token SO PIN .....	30
5.4	Registration .....	31
5.5	Create TPM User (Optional) .....	31
5.6	Change TPM Password (Optional) .....	32
6	Register Tool .....	33
6.1	Start CSSI security token configurator and Start CSSI Utility .....	33
6.2	Pause/ Continue .....	33
6.3	Settings .....	34
6.4	About .....	34
6.5	Exit .....	34
7	Charismathics Extension Tool .....	35
8	CSP of Charismathics Smart Security Interface .....	36
8.1	General Proceedings .....	36
8.2	Smart Card Login to a Windows 2000 Domain .....	37
8.3	SSL- Authentication with Smart Card over the Internet Explorer .....	37
8.4	Outlook Express with Electronic Signature and Encryption via Smart Card .....	37
8.5	Windows VPN-Login with Smart Card .....	37
9	PKCS#11-Module of Charismathics Smart Security Interface .....	38
9.1	Smart Card Login to a Novell eDirectory (formerly NDS) .....	38
9.2	SSL-Authenticatin with Smart Card Over Mozilla Firefox .....	38

9.3	SSL- Authentication with Smart Card over Netscape .....	40
9.4	Email-Security by Smart Cards and Tokens with Netscape's Messenger .....	41
10	References .....	42
11	Information / Export Restrictions .....	43
Appendix A:	Reference for Developers.....	44
	Functions according to PKCS#11-Standard.....	44
	Synopsis of specific functions .....	45
	C_Finalize.....	45
	C_GetObjectSize.....	45
	C_GetSlotList .....	45
	C_GetTokenInfo .....	45
	C_Initialize .....	46
	C_InitToken .....	46
	C_OpenSession .....	46
	C_WaitForSlotEvent.....	46
	Objects 47	
	Mechanism .....	49
	Sign (RSA):.....	49
	Verify (RSA):.....	49
	Encrypt (RSA):.....	49
	Decrypt (RSA): .....	49
	Digest (Hashfunctions SHA1, MD2, MD5): .....	50
Appendix B:	Non-Standard Functions in PKCS#11 DLL .....	51
Appendix C:	Log Information.....	52
	Convenience Files .....	52
	Registry Settings .....	52
Appendix D:	Certificate Attributes (Key Usage) .....	53

## 1 Preface

Thank you for purchasing the **Charismathics Smart Security Interface (CSSI)**.

**Charismathics Smart Security Interface** provides modules that you need in order to integrate different smart cards, USB tokens and TPM chips into your applications. The functionality ranges from administration of the card to modules supporting the operating system to use the token. The following file structures (profiles) are supported:

- Charismathics corporate profile
- PKCS#15 profile
- Carta Nazionale dei Servizi (CNS) profile
- FINEID profile
- AET profile
- PIV profile
- IAS Profile
- A-Trust Profile
- EDU Profile

**Charismathics Smart Security Interface** is comprised of the following modules:

- the administration tool - Charismathics Security Token Configurator
- the user tool - Charismathics Smart Security Interface Utility
- the CSSI Extension Tool for adding new ATR/OS associations
- the CSP
- the PKCS#11-module
- the TSS module for TPM chips (optional)
- CardOS V4.x minidriver

The user tool **Charismathics Smart Security Interface Utility** allows changing your user PIN and registering your smart card or USB token. You can manage keys and certificates on the token by using the administration tool **Charismathics Security Token Configurator**. You can generate, import or export keys and certificates. Furthermore, you can display information about the contents of a token, change and unlock the PIN of the token and create new profiles.

**Charismathics Smart Security Interface-CSP** enables you to enhance applications and services in a Microsoft environment and their use with a smart card.

**Charismathics Smart Security Interface-PKCS#11** enables you to use additional applications and services that use this standard interface. PKCS#11-Modules are in use by Netscape and in Novell environments.

In particular the following applications can be augmented by **CSSI**:

- Smart card login to Windows Domains or Novell eDirectory
- SSL- Authentication by smart card (Internet Explorer, Mozilla Firefox, ...)
- Email security with cards (PGP, Netscape Messenger, Outlook, Mozilla Thunderbird, Outlook Express, ...)
- VPN with smart cards (Microsoft, Cisco, ...)

This manual is meant for system administrators. Application developers, who develop their own applications that access software modules of **Charismathics Smart Security Interface**, e.g. PKCS#11, will find additional information in the appendices.

**Charismathics Smart Security Interface-BaseCSP** enables you to enhance applications and services in a Microsoft environment and their use with a smart card.

## 2 About this Manual

The **Charismathics Smart Security Interface** Security token configurator tool is described in chapter 4: “Administration Tool: Charismathics Security Token Configurator”. It contains information on how to manage keys and certificates, changing PINs, unlocking, initializing and personalizing smart cards.

The **Charismathics Smart Security Interface** user edition is described in chapter 5: “User Tool: Charismathics Smart Security Interface Utility”. It contains information on how to change PINs and register your smart card.

Furthermore, you will find additional information regarding the Register Tool, CSSI Extension Tool, CSP and PKSC#11 and which applications may be upgraded by hardware tokens. Application developers can find further information on how to access modules (e.g. accessing PKCS#11) of **Charismathics Smart Security Interface** in the appendices, if they intend to develop a proprietary application. Certificate Attributes (Key Usage) is a concise description of the certificate attributes, i.e. information about key employment. However, an explanation on how to configure environments of Microsoft or other producers exceeds the scope of this manual. In these cases, please consult the documentation of the corresponding supplier.

**NOTE:** *To understand this manual you need basic knowledge in IT-security. Especially, you should be familiar with the following notions: certificate, private and public key, secret key, digital signature, PKI, etc. Please consult the glossary on IT security on the Charismathics homepage ([http://charismathics.com/application\\_areas/IT\\_security\\_glossary.php](http://charismathics.com/application_areas/IT_security_glossary.php)) if you want to consolidate your knowledge.*

## 3 Installation

Before you can install **Charismathics Smart Security Interface**, the card reader you intend to use must be installed according to the manufacturer's guidelines and be fully operational. The installation of **Charismathics Smart Security Interface** is run from the program CD. Please execute the file SETUP.EXE as a user with administrator rights. Follow the installation instructions.

### 3.1 Installation Requirements

If not explicitly required otherwise in the following:

- Microsoft Windows XP
- Windows Vista
- Windows 2000
- Windows Server 2003
- Windows Server 2008
- Windows 7

**Note:** During the installation process the CSP Module is registered automatically with the Windows operating system. The following applications are supported:

- Smart card login to a Windows 2000 or 2003 or 2008-Domain:  
ADS, Enterprise CA, Windows 2000 or 2003 or 2008 Server and as Client: Windows 2000 Professional or Windows XP Professional or Windows Vista or Windows 7
- SSL- Authentication with smart card using Internet Explorer:  
Microsoft Internet Explorer Version 5.0, 5.5, 6.0, 7.0, 8.0, or 9.0, High Encryption Pack, SSL V3 with Strong User Authentication
- Outlook with digital signature and encryption via smart card:  
Outlook Express 5.0, 5.5 or 6.0  
Windows Mail  
Outlook 2000, 2003, 2007, 2010
- Lotus Notes with digital signature and encryption via smart card:  
Lotus Notes 6.5 or higher
- Windows VPN-Login with smart card:  
Windows 2000 Server and as Client: Windows 2000 Professional  
Windows 2003 Server and as Client: Windows 2000 or XP  
Windows 2008 Server and as Client: Windows XP or Vista
- Smart card login to Novell eDirectory:  
Netware 5.1 SP3, eDirectory 8.6.1, Novell Client 4.83 SP1, NMAS EE 2.0 (with the included Universal Smartcard Login Method) with NCI 1.5.7 (Server and Client), NMAS 2.1 (with the included Universal Smartcard Login Method) with NCI 2.4.1 (Server and Client) or higher in each case
- Smart card login to Lotus Notes:  
Lotus Notes 6.5 or higher
- SL- Authentication with smart card with Mozilla Firefox:  
Mozilla Firefox Version 1.0.x, 1.5, 2.0, 3.0, 3.6, 3.6 or 4.0
- SSL- Authentication with smart card with Netscape:  
Netscape Navigator 4.72 (High Encryption), 4.73, 4.76, 6.x

- Email-Security via smart cards with Netscape Messenger: Netscape Messenger 4.72 (High Encryption), 4.73, 4.76, 7.x
- Email-Security via smart cards with Mozilla Thunderbird 1.5 and above
- E-Mail-Security via PGP support (PKCS#11): PGP Personal Desktop 8.1 for Windows
- Compatibility/Smart card administration of the Baltimore-PKI (PKCS#11): Token Manager for Betruusted Unicert V5.2 for Windows
- Compatibility/Smart card administration of the Entrust-PKI (PKCS#11): Security Manager Administration 7.0
- Compatibility/Smart card administration of the Ecos-PKI Appliance BB5000 (PKCS#11)

The mentioned products do not require any further client software. Please refer to the manual of your software application if it is not listed above.

Furthermore, CSSI ties in with the following preboot/ harddisk encryption environments:

- CheckPoint or Pointsec
- Utimaco
- McAfee
- PGP
- Secude

### 3.2 Supported Smart Cards and TPM Chips

Charismathics Smart Security Interface supports the following smart cards/tokens:

- ACOS A-Trust Card
- ACOS EMV A03
- ACOS A04
- ACOS A05
- ACOS SMARTMX
- ActivIdentity Card
- Axalto Cyberflex Access V2c
- CardLogix Java 2.2.1
- Feitian FIPCS COS
- Siemens CardOS M4.01(a)
- Siemens CardOS V4.20
- Siemens CardOS V4.2B
- Siemens CardOS V4.2c
- Siemens CardOS 4.2C DI
- Siemens CardOS V4.30
- Siemens CardOS V4.3B
- Siemens CardOS V4.4
- Gemalto EMV – PKI
- Gemalto TPC DM 72K PIV
- Gemalto TOP IM GX4
- GemXpresso Pro R3.2
- GoldKey PIV token
- JCOP 20
- JCOP 21
- JCOP 30
- JCOP 31
- JCOP 41
- JCOP J2
- JCOP J3
- JCOP J4
- jTOP JCX32/36
- KONA 10
- KONA 132
- KONA 25
- KONA 26
- Micardo EC 2.x
- Morpho Orga YPS-ID2
- Morpho YPS-ID3 IAS ECC
- NetKey E4/2000
- NetKey PKS/2000/E4
- Oberthur Cosmo V5.2 PIV
- Oberthur Cosmo v5.4
- Oberthur Cosmopo RSA V5.x
- Oberthur CosmopolIC 64K V5.2
- Oberthur Cosmo ID-One V5.2 PIV
- Oberthur ID-One Cosmo V7.0
- Oberthur ID-One Cosmo V7.0 DI
- Oberthur ID-One Cosmo V7.0 – n
- Oberthur ID-One Cosmo V7.0 - a
- Oberthur ID-One v7 IAS ECC

- PAV Card ABACOS
- Sm@rtCafe Expert 3.1
- StarCOS SPK 3.0
- Privaris PlusID 60,75,90
- Sm@rtCafe Expert 3.2
- TCOS 2.x
- Setec SetCard
- Sm@rtCafe Expert 64k
- Wibu Code Meter Dongle
- Sm@rtCafe Expert 2.0
- StarCOS 3.0
- Watchdata TimeCOSPK
- Sm@rtCafe Expert 2.1
- StarCOS SPK 2.3
- Sm@rtCafe Expert 3.0
- StarCOS SPK 2.4

**Charismathics Smart Security Interface** supports the following TPM chips:

- Atmel TPM
- Infineon TPM
- ST TPM
- Broadcom TPM
- Intel TPM
- Nuvoton TPM

### 3.3 Tested Card Readers

Please make sure your PC/SC smartcard reader has been installed according to the producer's specifications and is fully operational. **Charismathics Smart Security Interface** has been tested with the following card readers:

- ACS38 USB
- charismathics plug'n'crypt
- Eutron cryptoidentity CCID
- Eutron Digipass 860
- Fujitsu Siemens Computer Smartcase KB SCR PRO
- Fujitsu Siemens Computer Smartcase KBPC CX
- Fujitsu Siemens Computer Smartcase SCR USB
- Fujitsu Siemens Computer Smartcase SCR USB internal
- Fujitsu Siemens Computer Smartcase Token USB
- KOBIL KAAAN advanced
- Omnikey Cardman 1010 serial
- Omnikey Cardman 2011 serial
- Omnikey Cardman 2020 USB
- Omnikey Cardman 3021 USB
- Omnikey Cardman 3121 USB
- Omnikey Cardman 3620 USB
- Omnikey Cardman 3621
- Omnikey Cardman 3821
- Omnikey Cardmann CM4040 (PC-Card)
- ORGA Card Mouse USB
- SCM SCR 331 USB
- SCM SCR 3310 USB
- SCM SCR 3340 (Express-Card)
- SCM SCR 532 serial/USB
- SCM SCR241 PCMCIA
- SCM SCR333
- SCM SCR335 USB

Additionally a great number of readers not explicitly mentioned above, but built upon compatible hardware, are supported.

Note:

- Only PC/SC-drivers are supported. There is no support for CT-API-drivers.
- If RSA 2048 bit key shall be used, then the smartcard reader must support the extended APDU.

### 3.4 Secure Pin Entry (SPE)

A number of card readers come equipped with their own PIN-pad. This PIN-pad can be used for SPE if one of the following devices is used:

- Cherry Keyboard G83-6644
- Omnikey 3621 USB
- OmniKey 3821 USB

Please make sure your windows device drivers are up to date if you want to use PC/SC 2.0 with SPE.

To enable SPE, edit the registry setting at

```
[HKEY_LOCAL_MACHINE\SOFTWARE\charismathics\smart security interface]
```

To activate SPE, use: "USE\_PINPAD"=hex:01

To deactivate SPE, use: "USE\_PINPAD"=hex:00

Although CSSI supports alphanumeric PINS in general, SPE obviously only supports digits. Please make sure the PINs used for the card can be entered by using SPE if you intend to use it.

### 3.5 Unattended Installation

Instead of calling setup.exe, the installation can also be started in unattended mode by calling the corresponding msi file from the setup directory

To install the admin edition:

```
msiexec /i "CSSIx.x.msi" /qn
```

To install the user edition:

```
msiexec /i "CSSIx.x.msi" /qn
```

To install the evaluation edition, use

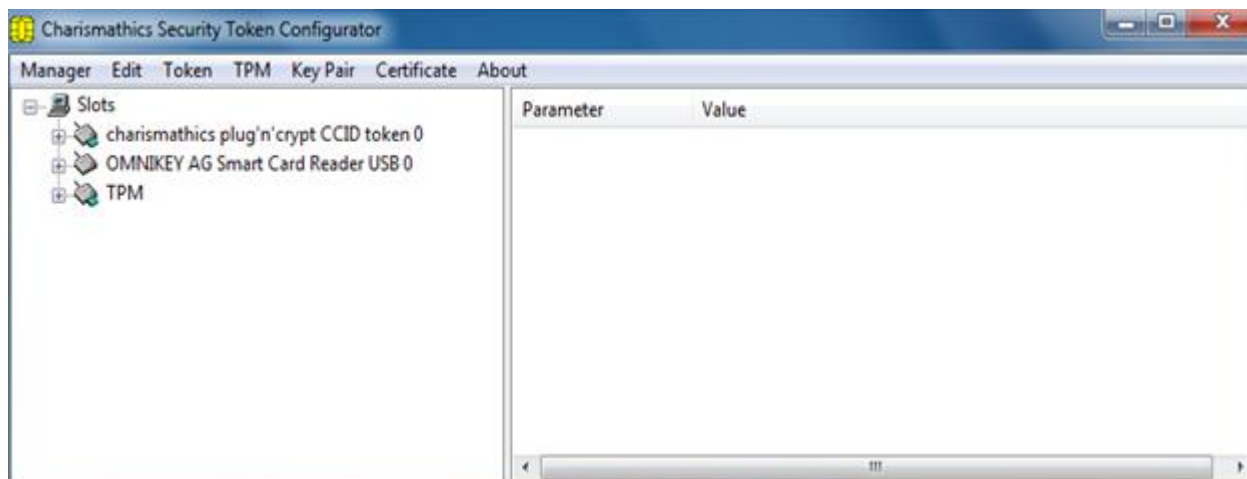
```
msiexec /i "CSSIx.x.msi" /qn
```

## 4 Administration Tool: Charismathics Security Token Configurator

This tool of the admin edition offers the following functions: changing your PINs, unlocking tokens, generating profiles, keys and certificates and so on.

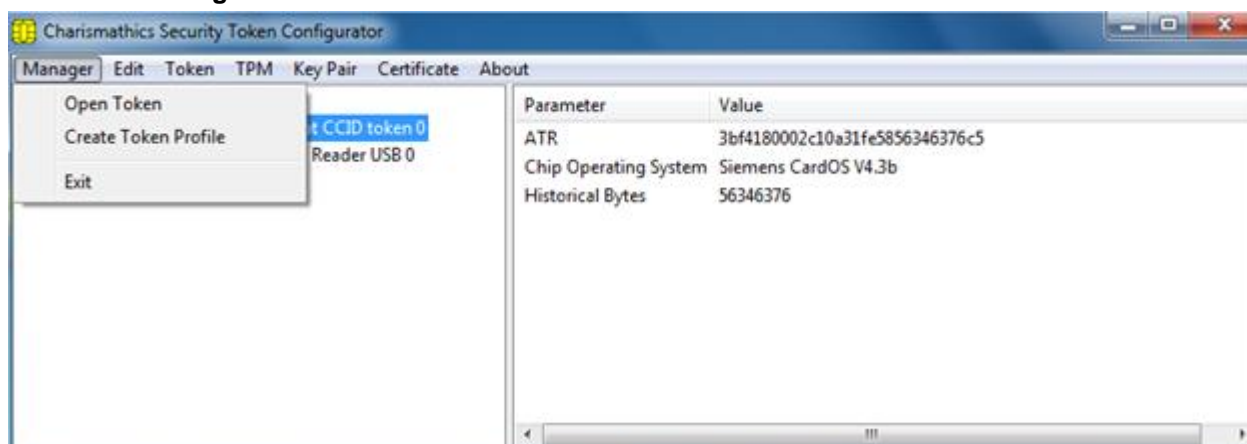
### 4.1 User Interface

After opening the administration tool of **Charismathics Smart Security Interface** you will see the following interface. The “TPM” Menu item is only visible if the optional TSS module has been installed and TPM hardware is present



The left panel displays the list of smart card readers which are connected to the system. Hardware smart card readers, virtual USB token readers and TPM hardware are displayed in the same window. Once a token has been inserted, the hierarchy is extended. Selecting an item in the hierarchy view displays its properties in the right hand panel. The properties are displayed in tabular form with parameter and its associated value.

#### 4.1.1 Manager Menu



- “Open Token”: To view the contents of a token, select the reader which contains the smart card, USB Token or TPM from the hierarchy and select “Open Token” from the “Manager” menu. Clicking the plus-icon in front of the reader to expand the hierarchy serves the same purpose. At first, only public information is available, e.g. label of the token, the profile and free memory. Furthermore, certificates, public keys, container and data are displayed.

- “Create Token Profile”: This option deletes the current profile, if present, and creates a new one on the smart card or USB token. This feature is described in detail in 4.7 “Preparing a Token (Initialization and Personalization)”.

#### 4.1.2 Edit menu/ Context menu

The content and availability of the “Edit” menu changes according to the item selected in the main hierarchy view. Most functions of the “Edit” menu are also accessible by right-clicking an item in the hierarchy. See also chapter 4.9 “Further Functions”

#### 4.1.3 Token menu

For the “Token” menu to contain any active entries, the Token must have been opened in advance e.g. by using “Manager”→“Open Token”.

- “Login”: Prior to operations on the token, the user is required to log in. Logging in requires the User Pin. Once logged in, this option is disabled and additional information becomes available, both within the hierarchy and the properties view. Failing to enter the correct User PIN three times in a row locks the card. See “Unlock User PIN” on how to clear the lock.

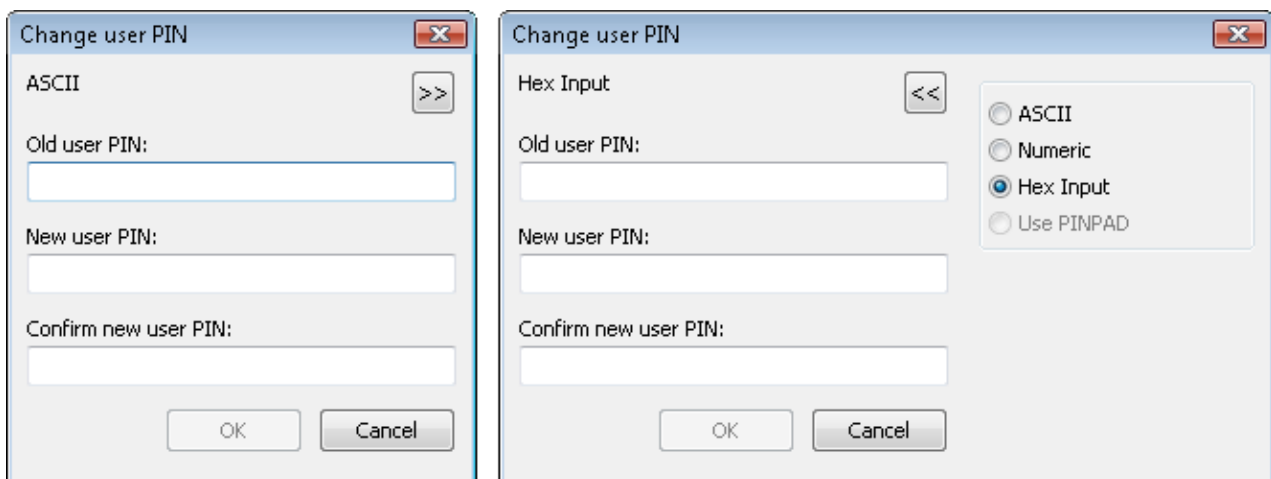
The hardware configuration and user settings determine the initial PIN entry method. To change the method use the dialog extension symbols “>>”.

Supported entry methods are:

- **ASCII:** each character of the PIN needs to be according to the ASCII table
- **Numeric:** each character of the PIN needs to be a digit ('0'...'9'). This can be used to ensure PINPAD compatibility.
- **Hex Input:** the PIN has to be entered in a hexadecimal format. That means the length of the PIN has to be even and only characters '0'-'9' and 'a'-'f' are valid.
- **Use PINPAD:** this option is enabled only when the authentication to the token is possible via secure PIN entry. When this option is selected, the edit text for the PIN will be disabled and the user must input the PIN from the corresponding SPE reader.
- **Use Biometric:** this option is enabled only when biometric authentication is possible by using a corresponding token. When this option is selected, the other PIN types will be disabled and a "Scan" button can be selected in order to start the biometric authentication.

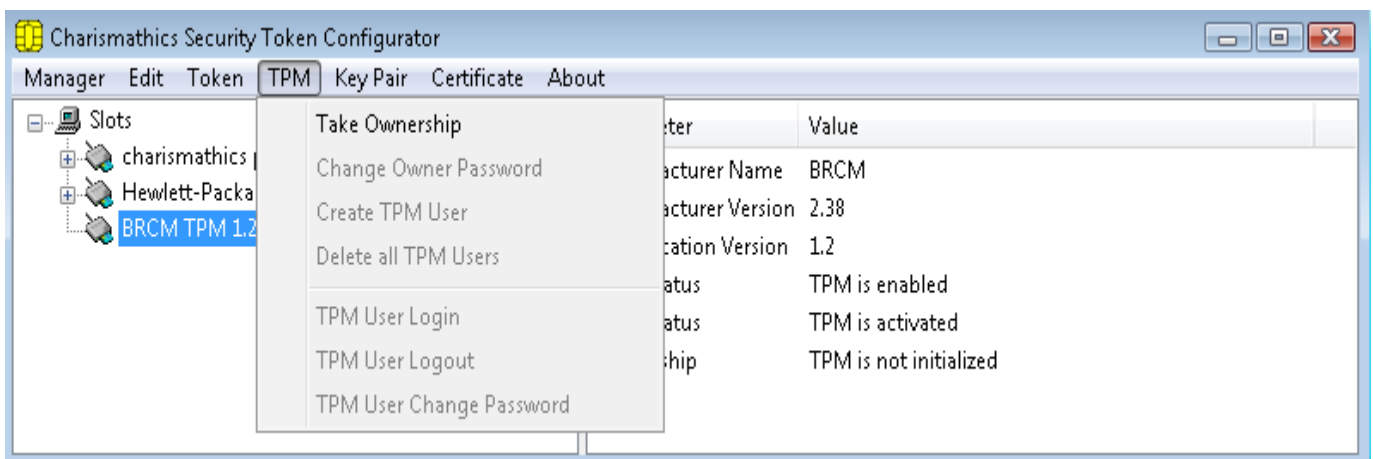
After successfully logging in to the token, certificates on the card can be registered with the Windows certificate store. For each certificate which is not yet registered with the certificate store but stored on the token, a dialog opens asking the user whether the certificate is to be registered.

- "Logout": This item works analogous to the "Login" option.
- "Change User PIN"/ "Change SO PIN"/ "Unlock User PIN"



These functions work very similar to each other. These functions are always available, and all require an authorization PIN to make a change. The changed value has to be entered twice to avoid typographic errors. All values are masked with asterisks to provide privacy. The PIN entry method can be changed the same way as in the login dialog.

#### 4.1.4 TPM menu

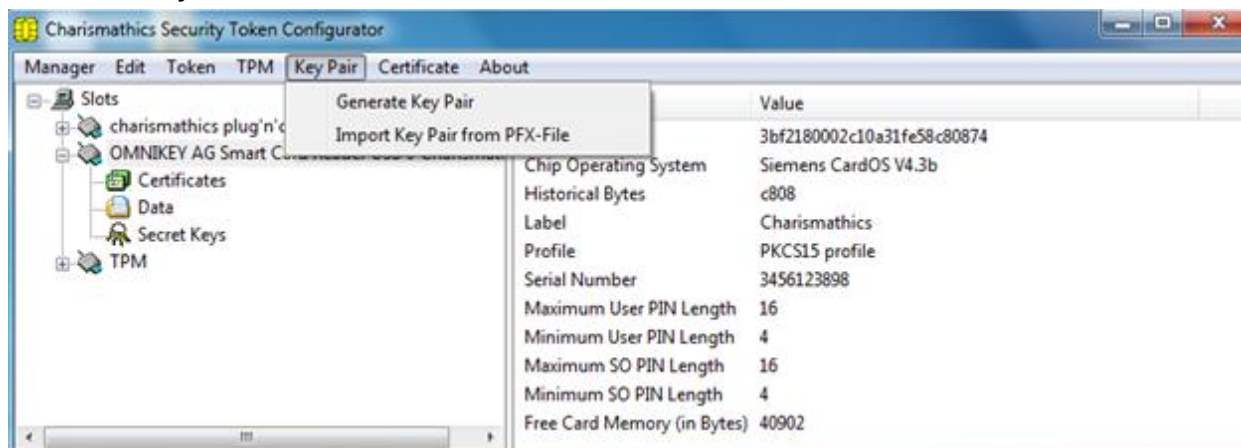


The TPM menu is only visible if the optional TSS module has been installed. The functions of the menu require that TPM hardware has been selected in the hierarchy view. See also chapter 4.8 “TPM Management”

- “Take Ownership”: This option is only available if Ownership of the TPM module has not yet been taken. Ownership is required for functions like TPM User creation to work. Taking Ownership of the TPM requires the User to enter a password. This password is only required for operations concerning the TPM itself. There is no option to give up Ownership from within the CSSI. Refer to the documentation of the TPM regarding this matter.
- “Change Owner Password”: This function asks for the current Owner Password and the new Owner Password, which has to be repeated, to change the Owner Password.
- “Create TPM User”: The CSSI admin editor permits the creation of multiple TPM User accounts. A TPM User is required to have the same name as a Windows account, otherwise the creation will fail. For any user to be able to use the TPM as a secure storage, a TPM User must have been created in advance using this function.
- “Delete all TPM Users”: This option removes all Users after asking for confirmation. Deleting individual users is possible by selecting the TPM User in the hierarchy view and selecting “Delete TPM User” from either the “Edit” menu or the right-click context menu.
- “TPM User Login”: Logging in to the TPM User allows importing key pairs from .pfx files using the “Key Pair” → “Import Key Pair from PFX-File”. Login to the TPM requires the current password of the user.
- “TPM User Logout”: This function works analogous to “TPM User Login”.
- “TPM User Change Password”: Changing the TPM User password requires entering the current password once and the new password twice.

There are no options to unlock TPM passwords since TPM does not use password locking when a password is entered incorrectly regardless of the number of failed attempts.

#### 4.1.5 Key Pair menu



It is possible to generate several key pairs with corresponding certificates on the card. Each set comprised of private key, public key (optional) and certificate (optional) is stored in a separate container.

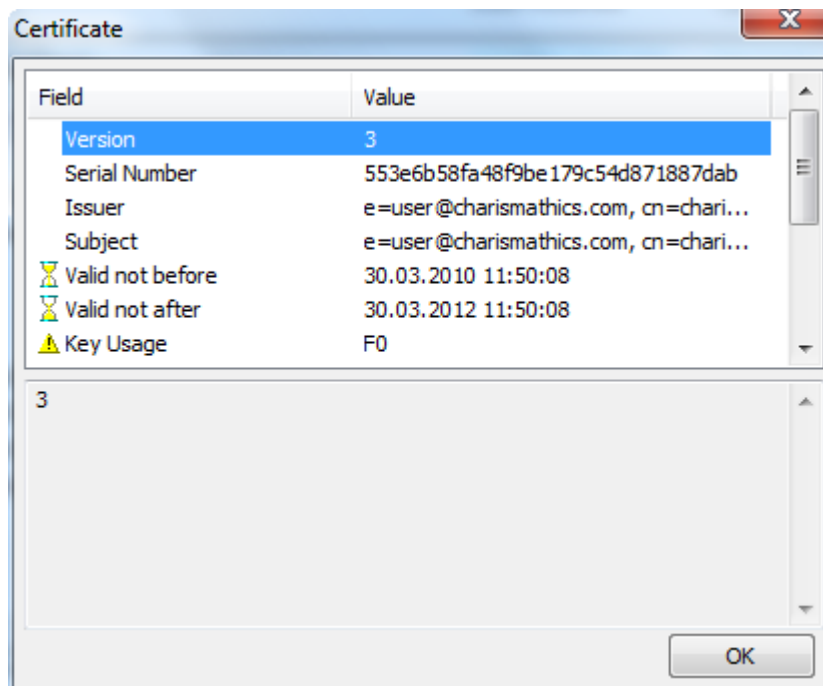
- “Generate Key Pair”: Keys can only be created on the token if the user has logged in before. Once a key pair has been created it can be used for a number of purposes. See also chapters 4.4 “Generating and Importing Keys” and 4.5 “Generating and Importing Certificates”.
- “Import Key Pair from PFX-File”: This item opens a dialog asking for the PFX file to import.

#### 4.1.6 Certificate menu

Most items of the certificate menu are also accessible from the context menu when a certificate, public or private key, an object node, e.g. “Certificates” node in the hierarchy, is right-clicked.



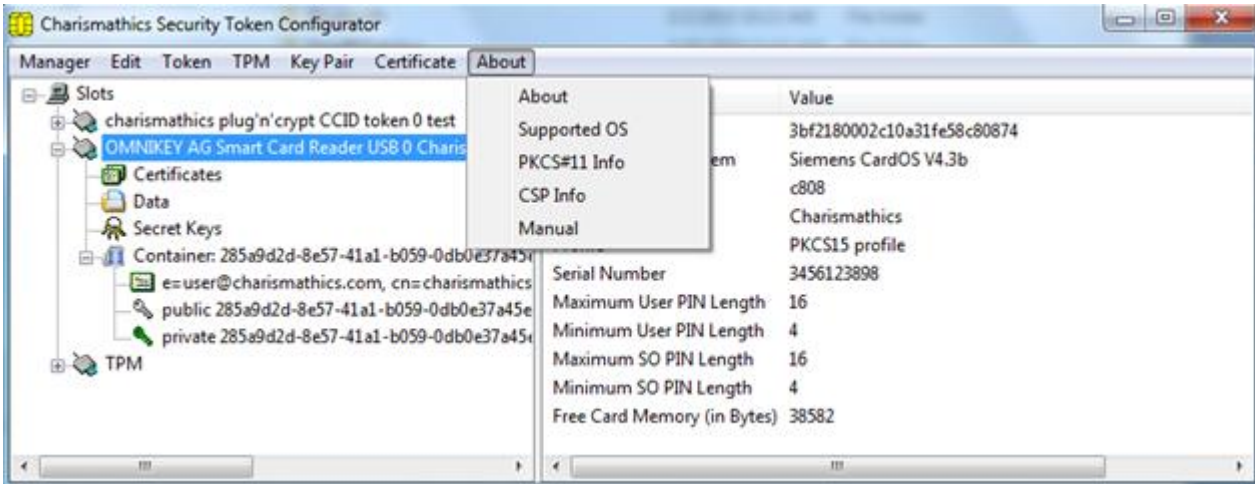
- “Import Certificate”: After selecting this item, pick the certificate to import from the opened dialog. If the certificate can be associated with a private/public key pair, it is automatically inserted into the correct container. Otherwise the certificate is added to the general “Certificates” node in the hierarchy. There is no way of manually associating a certificate with an unrelated key pair.
- “Show Certificate”: Displays all information contained within the certificate. Select a field name in the upper half of the viewer to display the value in the lower half.



- “Export Certificate”: Exports the certificate in either BASE64 or DER format to a file of the users choosing. The association with the key pair is recovered once the certificate is imported again via “Import Certificate”.
- “Register Certificate”: This option registers the certificate with windows, if not already done.
- “Create Certificate Request”: In order to receive a certificate for a private/public key pair, it is possible to prepare a certificate request. This request is stored in a BASE53 or DER encoded file. Refer to 4.5 “Generating and Importing Certificates” for a description of the process.

- “Create self-signed Certificate”: The requesting process is similar to the one in “Create Certificate Request”. However the request is not stored to a file to be processed by a CA, but instead signed by the requesting.

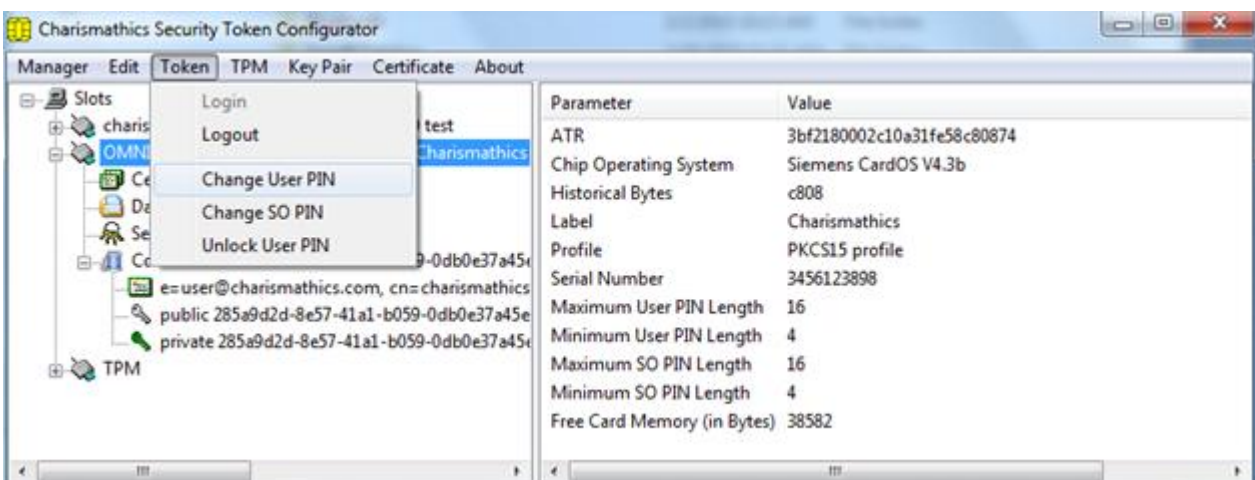
#### 4.1.7 Info menu



- “Info”: Displays general version information about the CSSI admin edition.
- “Supported OS”: Displays the list of smart card operating systems supported by CSSI. This list includes only the predefined associations. Additional associations can be made with the CSSI Extension Tool (7 “Charismathics Extension Tool”).
- “PKCS#11 Info”: Displays Information on the PKCS#11 module, which ships with CSSI.
- “CSP Info”: Information on the CSP.
- “Manual”: This manual.

#### 4.2 Changing PINs

Usually there are 3 PINs on a token: the User PIN, the SO PIN (PIN of the system operator, i.e. system administrator) and the Card PIN. The term Card PIN is used for USB Tokens as well. Please note that not all cards and tokens support changing all PINs. The CSSI supports alphanumeric PINs and is not restricted to numeric digits in general. Refer to Chapter 3.4 “Secure Pin Entry” on limitations. There are different functions to use with these 3 PINs:



The **User PIN** must be entered to write on the card (e.g. key generation, storing a certificate), delete objects or when the cryptographic functions (e.g. signing or decryption) are used. Refer to the table below regarding default User PIN and User PIN length.

**IMPORTANT:** After three consecutive wrong inputs the User PIN will be locked.

A locked User PIN can be unlocked by the **SO PIN**, which is also known as the PUK. Refer to the table below regarding default SO PIN and SO PIN length.

The SO PIN is used exclusively for unlocking the User PIN. There are no other functions associated with the SO-PIN.

**IMPORTANT:** After number of consecutive wrong inputs the SO PIN will be locked. Number of retry depends on the card profile.

With the **Card PIN** it is possible to delete an existing profile on a card and set up a new profile. The Card PIN will be determined during the initialization and can only be changed afterwards by creating a new profile. The length of the Card PIN is exactly ten characters.

**IMPORTANT:** After ten consecutive wrong inputs the PIN is locked and the card cannot be deleted anymore, i.e. if the Card PIN, the SO PIN and the User PIN are locked, the token is useless.

PIN (default)	Charismathics Profile	PKCS#15 Profile	CNS Profile
User PIN (11111111)	4- 8	4- 8	4- 8
SO PIN (1111111111)	8- 10	8- 10	4- 8
Card PIN (0987654321)	10	10	10

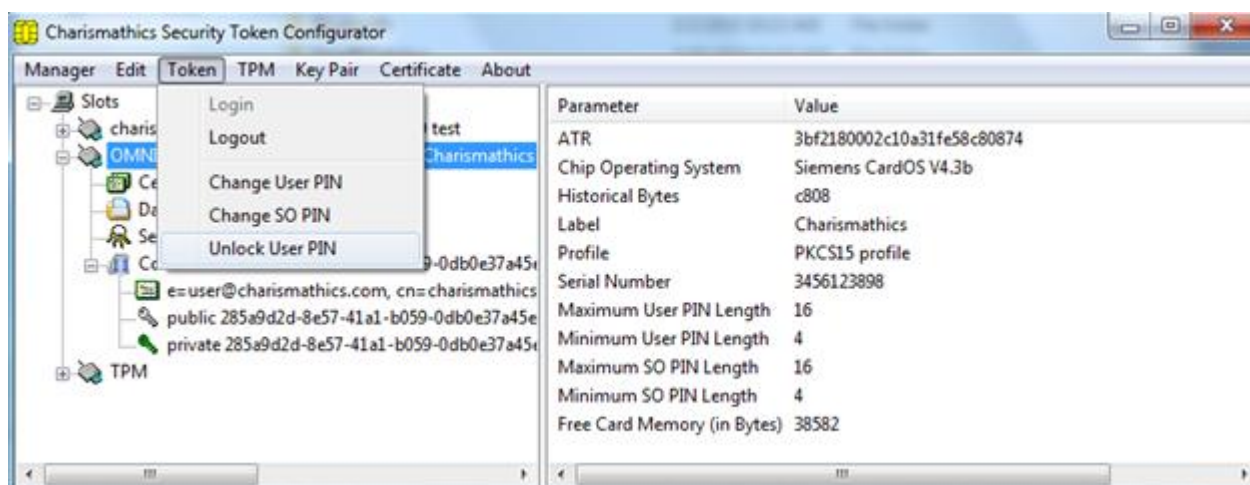
You find all functions concerning User and SO PIN in the menu "Token", as shown in the following figure:

### 4.3 Unlocking Tokens

As a security measure a token will be locked if a user enters a wrong PIN three times in a row. This provides security since an unauthorized person could otherwise check all possible PINs by trial and error if you lost your smart card or USB token, or it has been stolen.

But it might happen that you have entered the wrong PIN three times even as a legitimate owner of the smart card. In this case the smart card will be locked as well. Therefore, you can unlock the smart card with **Charismathics Smart Security Interface**, if you know the SO PIN.

You need the SO PIN to unlock a User PIN. "Unlock User PIN" is available from the "Token" menu, as shown in the following figure:



## 4.4 Generating and Importing Keys

To use the smart card for digital signatures or encryption, you need a key pair, which is composed of a private and a public key. The private key must be securely stored and the public key must be accessible to communication partners by a certificate. These keys and certificates can be generated and managed by the administration tool.

In principle there are two possibilities:

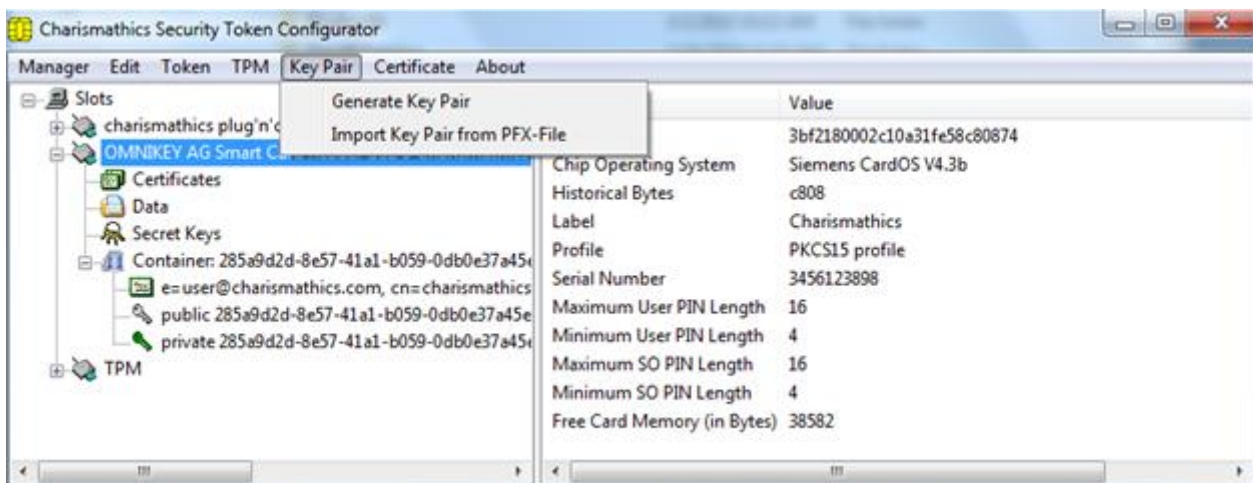
1. You can generate keys (key pairs comprising private and public keys and secret keys) with the administration tool of **Charismathics Smart Security Interface**.
2. You already own a key and/or key pair. Then, you can import the key pair if necessary together with certificate as a PFX-file. You can store Secret Keys by importing them, e.g. with "Copy and Paste".

Use of these functions requires that you are logged into the smart card: From the menu "Token", select the entry "Login" and enter your User PIN.

You can find all the functions to generate and import keys in the menu "Key Pair" and to import certificates in the menu "Certificates".

### 4.4.1 Generation of a Key Pair

The generation of a key pair (private and public key) is started from the menu "Key Pair" via the item "Generate Key Pair". Once the generation process is finished, you see these keys in the administration tool in the corresponding container under "public key" and under "private key".



The type of key and the key strength depend on the available hardware. Make sure your reader has support for Extended APDUs if you want to create and use 2048 bit keys.

### 4.4.2 Importing a Key Pair

If you already own a key pair that you intend to use, you can import it with the menu "Key Pair", selecting the item "Import Key Pair from PFX-File". The method requires you to enter your password for the file.

Remark: The key must be an RSA-key within a .pfx- or .p12-file.

### 4.4.3 Generation of a Secret Key

To generate a secret key for encryption, highlight "Secret Keys" and select the item "Generate Secret Key" in the menu "Edit" or the context menu.

Here, you can generate a Triple-DES-key with 192 bits, a Triple-DES-key with 128 bits or a DES-key with 64 bits.

Algorithm	Key Strength	Note
Triple-DES	192 bit	
Triple-DES	128 bit	
DES	64	Not recommended

**Note:** Algorithms with at least 128 bits (Triple-DES) are recommended. According to present day standards, lesser key lengths cannot be considered secure anymore.

#### 4.4.4 Importing a Secret Key

If you own a secret key that you want to use, you can import it with the menu "Edit" via the menu item "Store Secret Key". The Secret Key must be specified in hexadecimal representation and be of the correct length: 192 or 128 bits for Triple-DES and 64 bits for DES. Please note that a single hex-digit covers 4 bits. Importing takes place by inserting the bits into the field "Secret Key (hexadecimal)", e.g. copy and paste.

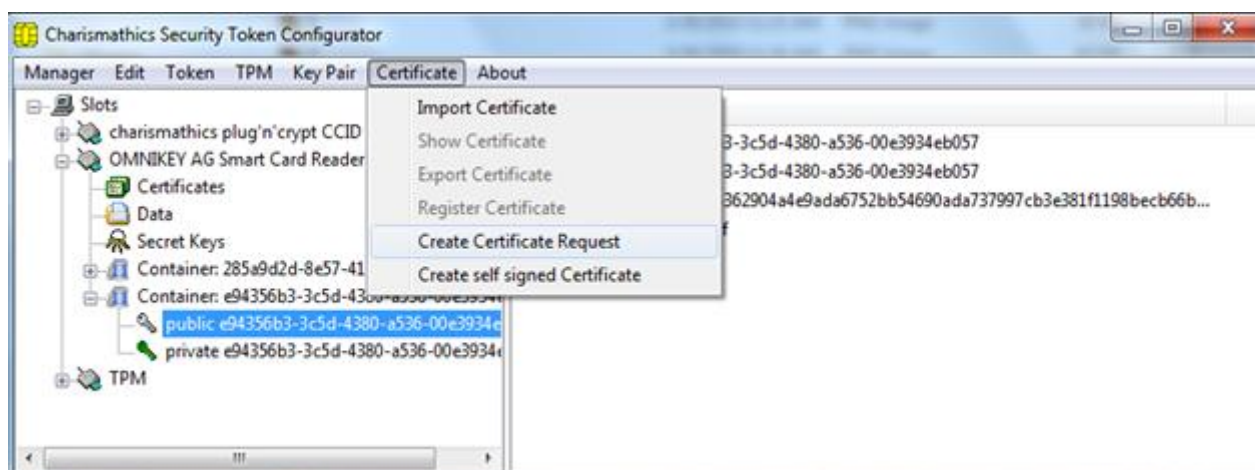
### 4.5 Generating and Importing Certificates

In order to use the smart card for digital signatures or encryption you need a key pair comprised of a private key and a public key. The public key has to be accessible to communication partners via a certificate. These certificates can be generated and managed by the Charismathics Security token configurator tool. These options to create a certificate from within the administration tool are:

1. You can sign the certificate corresponding to a public key by you or make a certificate request, such that another instance e.g. a trust center will authenticate the public key.
2. You already have a key and/or certificates. Then, you can import certificates if needed together with the corresponding key.

#### 4.5.1 Generating Self-Signed Certificates and Certificate Requests

You can generate the certificate belonging to a public key by signing it yourself or make a certificate request such that another instance, e.g. a trust center, authenticates the public key. To this end you highlight the Private Key and select one of the "Create..." entries from the "Certificates" menu.



In order to generate the certificate request you enter the data into the corresponding fields. In case of a certificate request, you create a file to send it to the authority that should sign the certificate (e.g. trust center). Therefore, you store the request as a p10 file in a directory and follow the instructions of the corresponding authority intended to sign the certificate.

Once the certificate has been returned by the issuer, you have to import the certificate using the menu item "Import Certificate".

**Note:** There is an explanation of the certificate attributes and how to employ the keys in the appendix B of this manual.

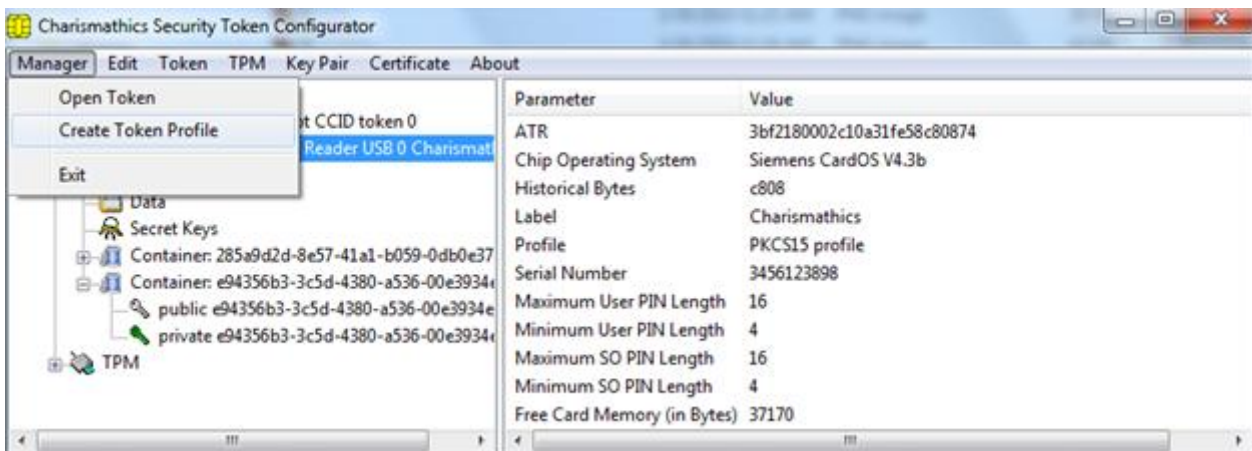
#### 4.5.2 Importing Certificates

In case you already own certificates that you intend to employ, you can import them with the menu "Certificate" under the item "Import Certificate". Certificates which belong to key pairs are directly assigned to the associated "container" after the importation. Certificates without keys - as for example CA certificates - are assigned to the file "Certificates".



#### 4.6 Creating Profiles

If you want to use a smart card, there must be a profile on this smart card. In a first step you have to setup the corporate profile on this smart card. Click the menu "Manager" → "Create Token Profile".



##### 4.6.1 Smart Card with Profile

If there is already a profile on the card and you want to create a new one, the existing one will be deleted as a first step. To this end enter the Card PIN. If you have created the profile yourself, you have to enter the Card PIN you have assigned to the card. The default Card PIN is "0987654321".

The further proceedings are the same as in the following section "...in the case of an empty smart card". Please follow the instructions which are described below.

##### 4.6.2 Empty Smart Card

If the profile is set up (Initialization) on an empty smart card, the "Card PIN", the "SO PIN", the "User PIN" and a Serial Number must be defined. Additionally a Label for the token can be assigned. If multiple profiles are available you can choose one now. Using CSSI tool you can initialize the cards with five profiles (Corporate, PKCS#15, CNS, FineID and PIV) depending upon the smartcards used. For AET profile, you can not initialize the cards with this profile but CSSI middleware is backward compatible with cards that have

been initialized with this profile. Meaning we can read these cards, access data, create data on them, but not initialize the card with this particular profile.

With the help of the Card PIN, the smart card can be deleted later again. With the SO PIN the smart card can be unlocked. Therefore you should not assign "simple" PINs. The input of the SO PIN and the User PIN is also not displayed in plain text but with asterisks (\*) in the input mask. The input must be also confirmed. Further explanations regarding the PINs can be found in the section 4.2.

**NOTE:** PIV card is supported only with the special release, CSSI for PIV.

## 4.7 Preparing a Token (Initialization and Personalization)

In order that a user can employ his smart card, it must be prepared, i.e. the smart card must be initialized and personalized. In a first step you have to setup a profile on the smart card and in a second step setup the keys and certificates on the smart card.

### 4.7.1 First Step: Creating a Profile (Initialization)

As a first step you must setup a profile on an empty smart card. You proceed as described in section 4.6 "Creating Profiles".

### 4.7.2 Second Step: Creating Keys and Certificates (Personalization)

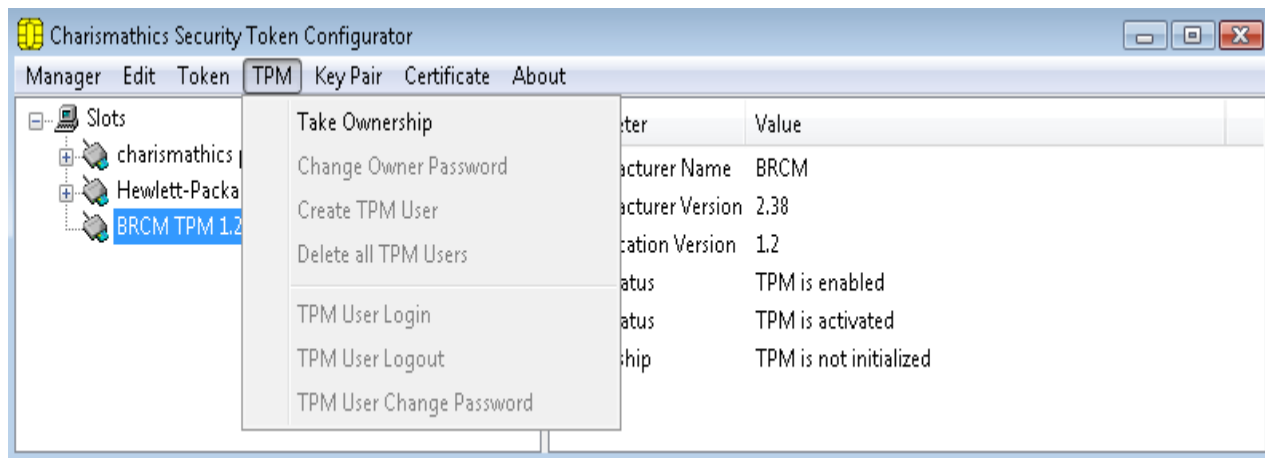
As a second step you must set up for a user key and certificate on the smart card. You have the possibility to either generate keys and certificates or to import them. Refer to section 4.4 "Generating and Importing Keys" and section 4.5 "Generation and Import of Certificates".

## 4.8 TPM Management

TPM management functions are only available if you installed the charismathics TSS module and the computer is equipped with TPM hardware. Please ask charismathics sales for details about the license of charismathics TSS module. The CSSI covers two aspects of the TPM lifecycle: Ownerless Modules and Owned Modules.

#### 4.8.1 Operations on TPM without owner

After the TPM Token has been opened using either the plus sign in front of the label within the main hierarchy view or using “Manager”→”Open Token”, ownership of the TPM can be taken.



Once prompted, enter the password for the TPM Ownership. The TPM owner password is used only for TPM specific but user unrelated operations.

Once ownership has been taken, it can no longer be relinquished using CSSI. Refer to the documentation of your TPM hardware (usually within the BIOS) if you want to give up the ownership.

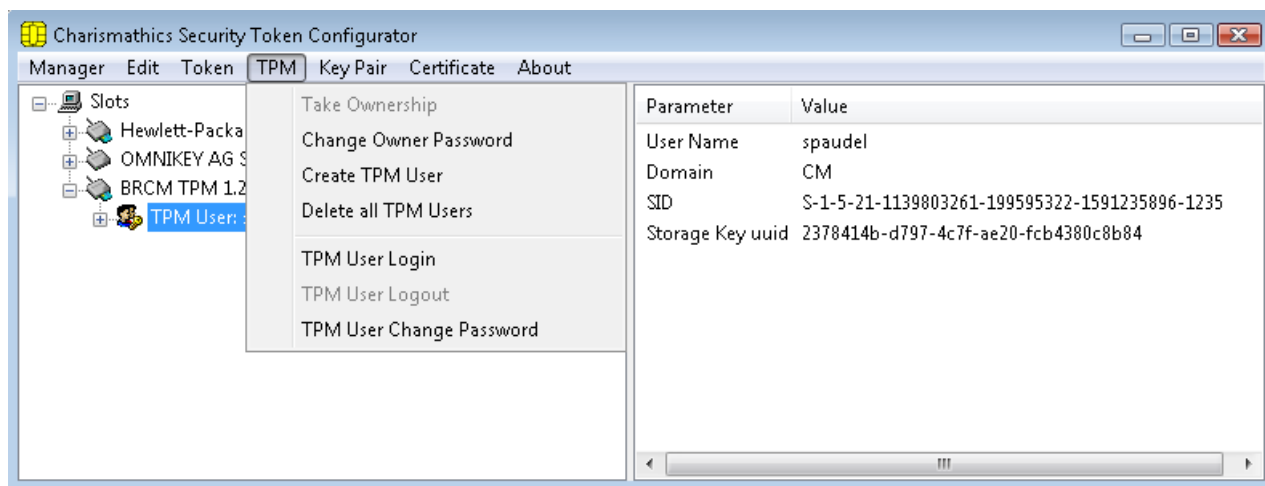
#### 4.8.2 Changing the current TPM Owner Password

The current password can be changed later using “TPM”→”Change Owner Password”

At this point you will be asked to enter the old password and verify the new password by entering it twice.

#### 4.8.3 Creating TPM User

In order to make the TPM usable for a windows user a TPM User has to be created.



The TPM User has to be equal to the name of a windows user account; otherwise the TPM User creation fails.

#### 4.8.4 Deleting TPM User

TPM User can be deleted together or individually. To delete all Users at once, select “TPM”→”Delete all TPM Users” and confirm the following dialog.

Alternatively, it is possible to delete TPM User individually by right-clicking the User account and selecting “Delete TPM User” from either the context menu or the “Edit”. Deleting a user in this fashion requires confirmation as well.

#### 4.8.5 Inspect TPM User private information

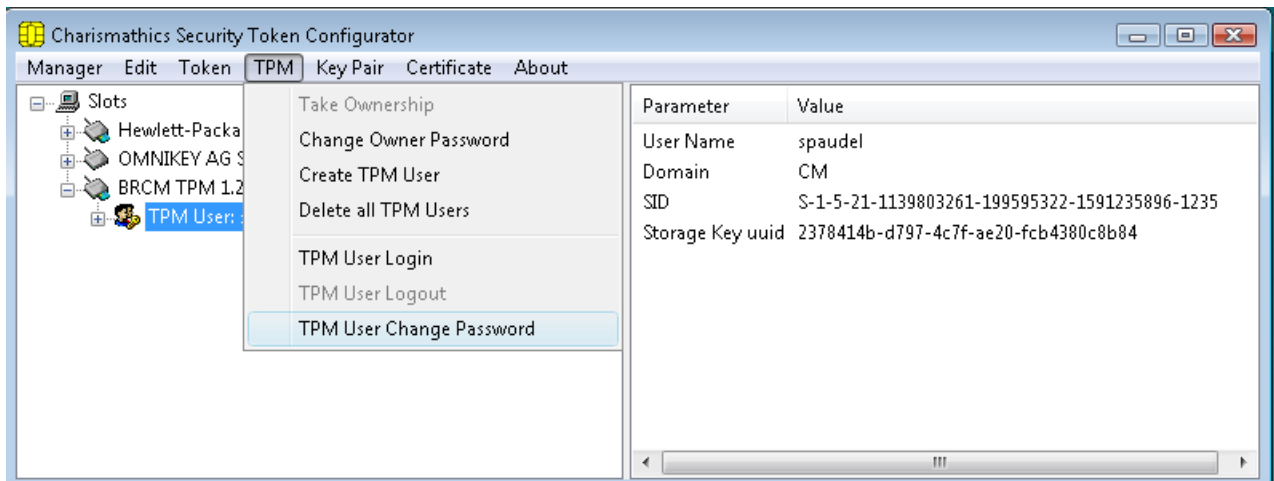
To view the information associated with a user, use “TPM”→”Login”.

Login to the TPM User requires entering the password of the selected account.

TPM does not know a locking mechanism like smartcards do with PINs. Instead TPM hardware increases the delay between login attempts.

#### 4.8.6 Changing the TPM User Password

The password of the selected User can be changed via “TPM”→”TPM User Change Password”.



Confirm the password change by entering the old password and entering the new password twice when asked.

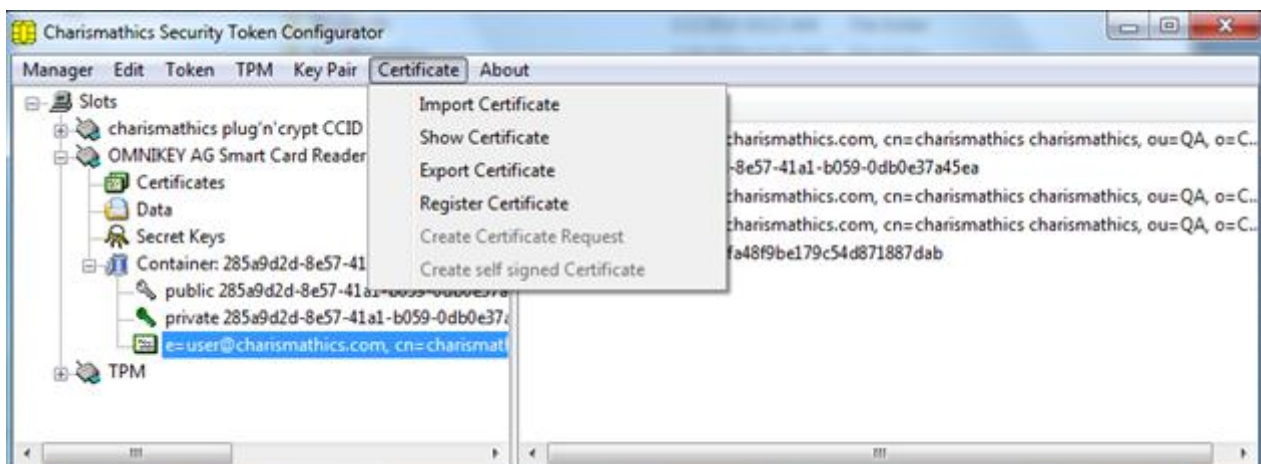
#### 4.8.7 Importing a key pair from a PFX file

Instead of generating the public and private key pair on the TPM, it can also be imported from a PFX file. Select “Key Pair” → “Import Key Pair from PFX File” and select the PFX file using the dialog.

### 4.9 Further Functions

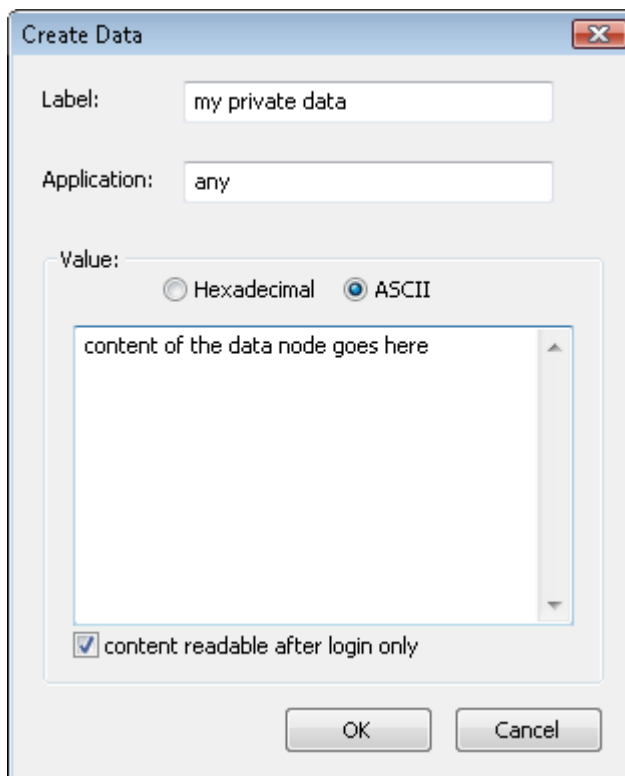
#### 4.9.1 Directory "Certificates"

There is the directory “Certificates” for all certificates that are not directly corresponding to a key. These are intermediate certificates that have to be imported into this directory. For this purpose select the item “Import Certificate” in the menu “Certificate” or choose the context menu using the right mouse button.



#### 4.9.2 Directory "Data"

A smart card is the safest environment for the private key. Furthermore, the smart card is necessary for application with at least daily logins or authentication. Thus, it is often or always carried around. Therefore it makes sense to store sensitive or necessary data on this medium, e.g. a text file with your PINS. To create data highlight "Data" and select the item "Create Data" in the menu "Edit". Then, a further window is displayed for you, where you can create your data:



There you have the possibility to access the actual data only if one is logged on to the smart card. To this end tick the field "content readable after login only". Your existing data can be deleted, updated or exported via the "Edit" menu.

#### 4.9.3 Function "Open Token"

The function "Open Token" of the menu "Manager" transfers data from the smart card to the user interface. This is recommended, if you work with different cards or card readers.

#### 4.9.4 Function "Delete all" and "Delete Object"

You can delete all objects, keys, and certificates with the function "Delete all" of the menu "Edit".

The function "Delete Object" offers you the possibility to remove objects, keys, and certificates. You obtain this second function over the context menu as well: highlight the object that you want to delete, right-click and there chose the item "Delete Object".

#### 4.9.5 Function "Set Default Container"

The function "Set Default Container" of the menu "Edit" is relevant to you only if you use a smart card for login to a Windows-2000 domain via CSP.

If you do not choose a container as Default Container, Windows will take the first key from the list for the login to a Windows-2000 domain via CSP.

If you have chosen a Container as Default Container, it will show in bold face in the interface of the administration tool

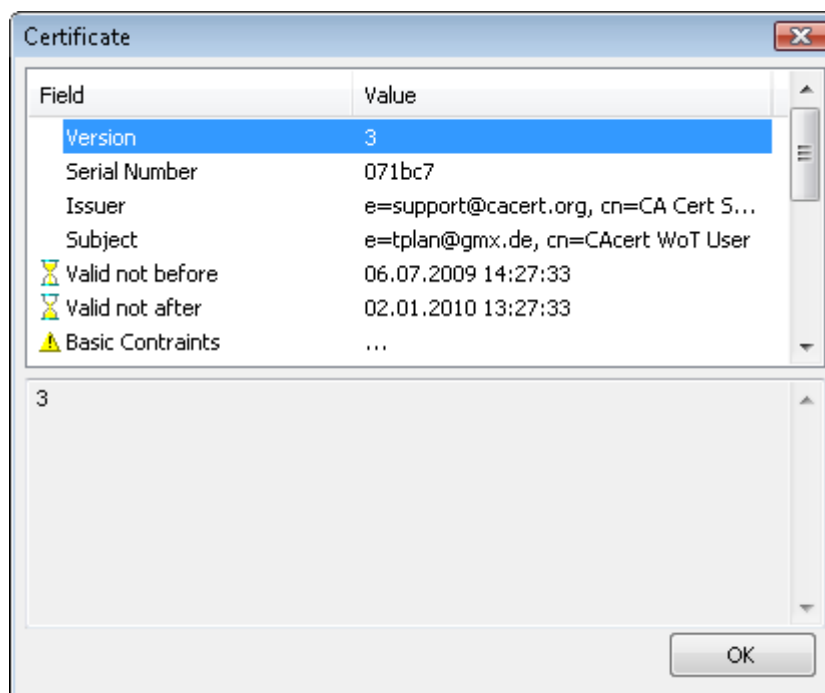
#### 4.9.6 Function "Change Container Label"

The function "Change Container Label" of the menu "Edit" allows the user to change the friendly name of the container. Please be aware that once a certificate was registered in the Microsoft Certificate Store, you need to register the certificate again after renaming the container. Otherwise the certificate cannot be used for any operation via CSP.

#### 4.9.7 Function "Show Certificate"

If you want to display a certificate, use the function "Show Certificate" from the menu "Certificate".

You can access this function over the context menu as well: highlight the certificate that you want to display, right-click and select the item "Show Certificate". Then you obtain the information contained in the certificate:



#### 4.9.8 Function "Export Certificate"

If you want to employ a certificate for other applications, you can export it from the smart card with the function "Export Certificate" from the menu "Certificate". You can also access this function from the context menu: highlight the certificate that you want to export, right-click and choose the item "Export Certificate".

#### 4.9.9 Function "Register Certificate"

The function "Register Certificate" from the menu "Certificate" installs the certificate that you want to register to make it accessible for Windows-applications (like Internet Explorer or Outlook Express).

You can also access this function from the context menu: highlight the certificate that you want to register, right click and choose the item "Register Certificate".

Configuration of the settings regarding the registration is done in the *Register Tool*. Read more about this in chapter 6 Register Tool.

#### 4.9.10 Function "Check Private Key"

With this function you can test generated keys, e.g. for signing or decryption. First you must be logged on, then highlight the private key you want to test and chose the function "Check Private Key" from the menu "Edit" or the context menu.

To test the decryption key, write text in the field "Plaintext" and click on the button "Start". If the decrypted text is the same as the Plaintext, the decryption key works all right.

Check Private Key

Encrypt/Decrypt

Plaintext:  
if the key works, this text will reappear

Ciphertext:  
44817553ebd0cfc011c58042e59a04c597dc4f62bb65f

Decrypted text:  
if the key works, this text will reappear

Start

Sign/Verify

Hash: SHA1

Plaintext:  
if signing works, the result can be verified

Signature:  
6f66a35291467438a84ec7e4a4e19f5c56419bb27d5e

Verify Result:  
True

Start

Close

To test the signing key you can choose the hash algorithm. If the Verify Result is true, the signing key works all right.

#### 4.9.11 Function "Check Secret Key"

With this function you can test generated keys for encryption. First you must be logged on, then highlight the private key you want to test and choose the function "Check Secret Key" from the menu "Edit".

You can choose the cryptographic mode for testing the key. The different versions are the Cipher Block Chaining (CBC) and the Electronic Code Book (ECB). And you can choose ISO or PKCS5 as Padding.

To test the encryption key, write text in the field "Plaintext" and click on the button "Start". If you know the initializing vector you can insert it; otherwise it will be filled with zero. If the Decrypted text is the same as the Plaintext, the encryption key works all right.

The image shows a dialog box titled "Check Encrypt/Decrypt" with a close button (X) in the top right corner. The dialog contains the following elements:

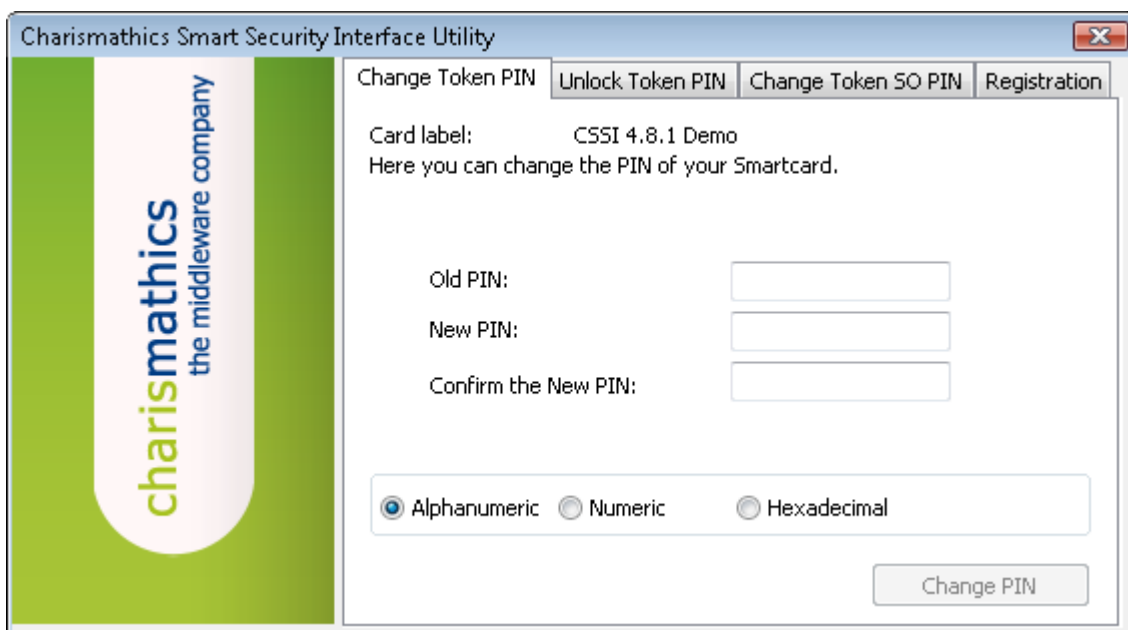
- Mode:** A dropdown menu set to "CBC".
- Padding:** A dropdown menu set to "ISO".
- Plaintext:** A text input field containing the text "if the symmetric encryption works, this text will reapp".
- Initializing vector (hexadecimal):** A text input field containing the hexadecimal string "0123456789abcdef".
- Ciphertext:** A text input field containing the hexadecimal string "0123456789abcdefb301bc29c806de0e232de6978cc9".
- Decrypted text:** A text input field containing the text "if the symmetric encryption works, this text will reapp".
- Buttons:** Two buttons at the bottom: "Start" (highlighted with a blue border) and "Close".

## 5 User Tool: Charismathics Smart Security Interface Utility

This tool exposes all relevant functions if you acquired **Charismathics Smart Security Interface** in the user edition. Changing your pin and the registration of your key/certificates of the smart card are available as well as TPM management functions. Insert your smart card in the reader and open **Charismathics Smart Security Interface** Utility by following the path:

"Programs"->"charismathics "->"smart security interface" ->"smart security interface utility".

### 5.1 Change PIN

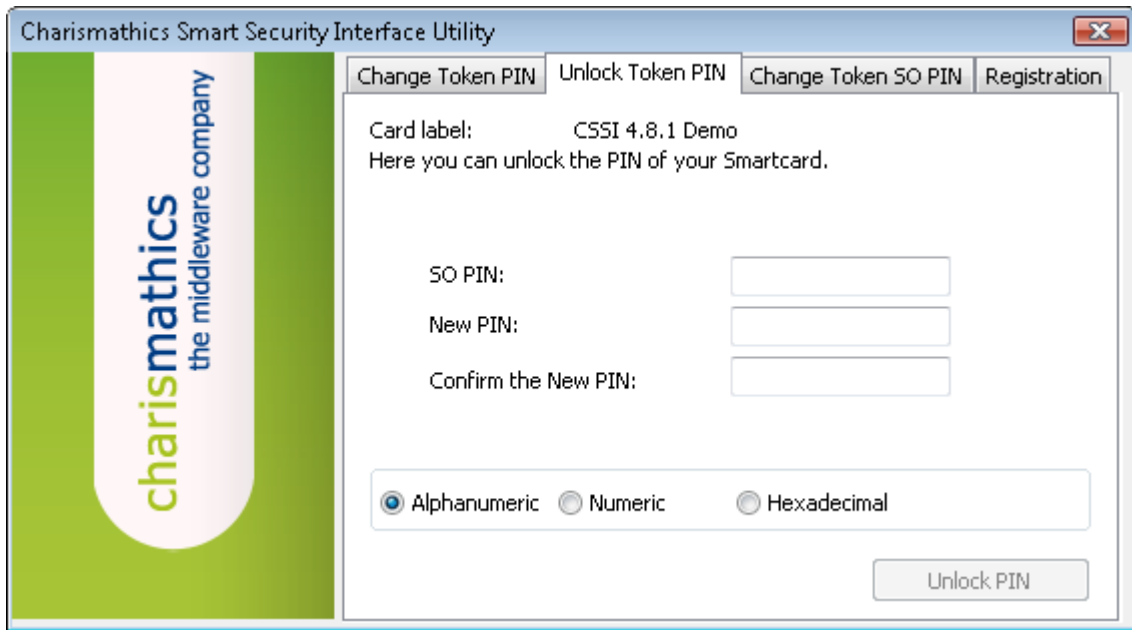


To change your PIN, insert the old PIN followed by the new PIN which must be entered a second time as confirmation. The minimum length of the User PIN is four characters and the maximal length is ten characters.

Click on the button "Change PIN", and you receive a window with the confirmation.

**IMPORTANT:** After three consecutive wrong inputs the User PIN will be locked. Please choose a PIN, which you can remember well, but which cannot be easily guessed. Avoid birthdays or simple sequences of numbers like 1234 or 1111.

## 5.2 Unlock PIN

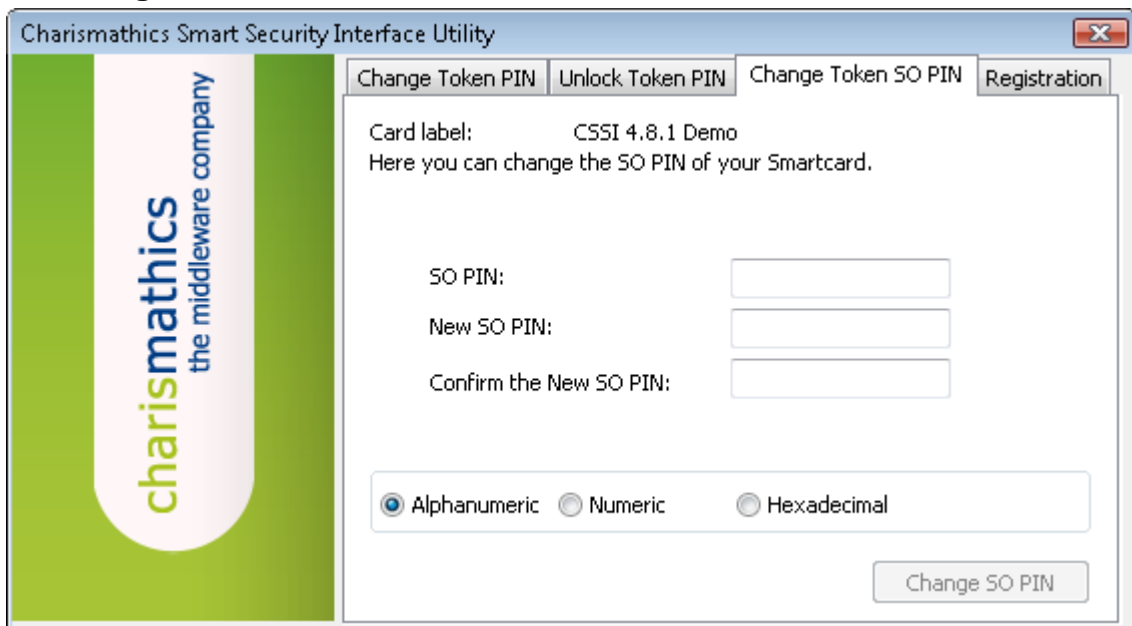


The screenshot shows a window titled "Charismathics Smart Security Interface Utility" with a sidebar logo for "charismathics the middleware company". The "Unlock Token PIN" tab is selected. The card label is "CSSI 4.8.1 Demo". The text reads: "Here you can unlock the PIN of your Smartcard." There are three input fields: "SO PIN:", "New PIN:", and "Confirm the New PIN:". Below these are radio buttons for "Alphanumeric" (selected), "Numeric", and "Hexadecimal". An "Unlock PIN" button is at the bottom right.

To unlock your PIN, enter the SO PIN followed by the new PIN, which must be entered a second time as confirmation. The minimal length of the User PIN is four characters and the maximal length is ten characters.

Click on the button "Unlock PIN" and a confirmation window opens.

## 5.3 Change Token SO PIN

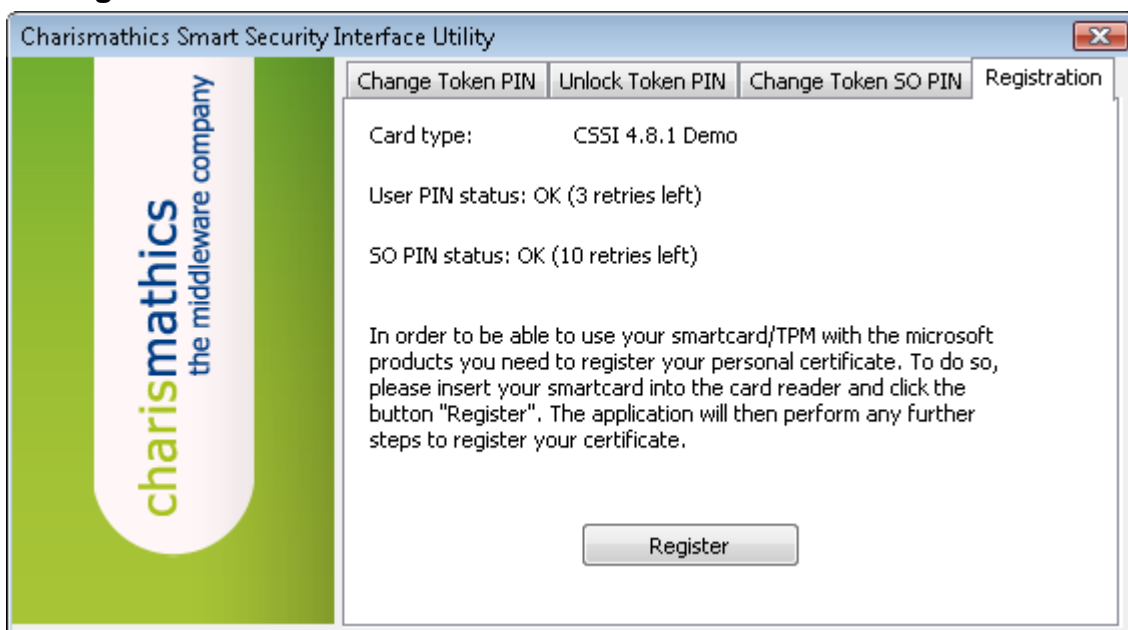


The screenshot shows the same window as in 5.2, but with the "Change Token SO PIN" tab selected. The card label is "CSSI 4.8.1 Demo". The text reads: "Here you can change the SO PIN of your Smartcard." There are three input fields: "SO PIN:", "New SO PIN:", and "Confirm the New SO PIN:". Below these are radio buttons for "Alphanumeric" (selected), "Numeric", and "Hexadecimal". A "Change SO PIN" button is at the bottom right.

To change the Token SO PIN, enter the SO PIN followed by the new SO PIN, which must be entered a second time as confirmation. The minimum and maximum length of the SO PIN is dependent on the card OS.

Click on the button "Change SO PIN" and a confirmation window opens.

## 5.4 Registration



Registration page display the number of retry left for User PIN and SO PIN.

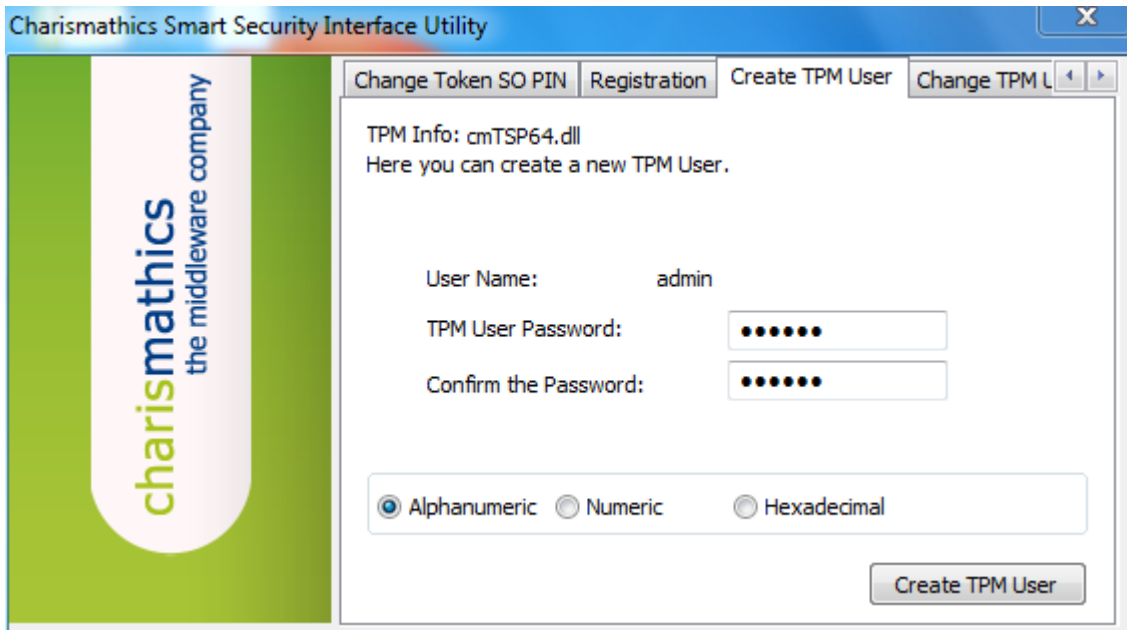
Your smart card may contain multiple certificates and keys. These certificates must be registered once, so that applications can use these. Certificates propagation is done automatically once the card is inserted into the smart card reader. If in case automatic certificate propagation does not work, simply clicking on Register button as shown in the screen shot above will do the job.

**IMPORTANT:** THE REGISTRATION NEEDS TO BE DONE ONLY ONCE FOR EACH CARD.

## 5.5 Create TPM User (Optional)

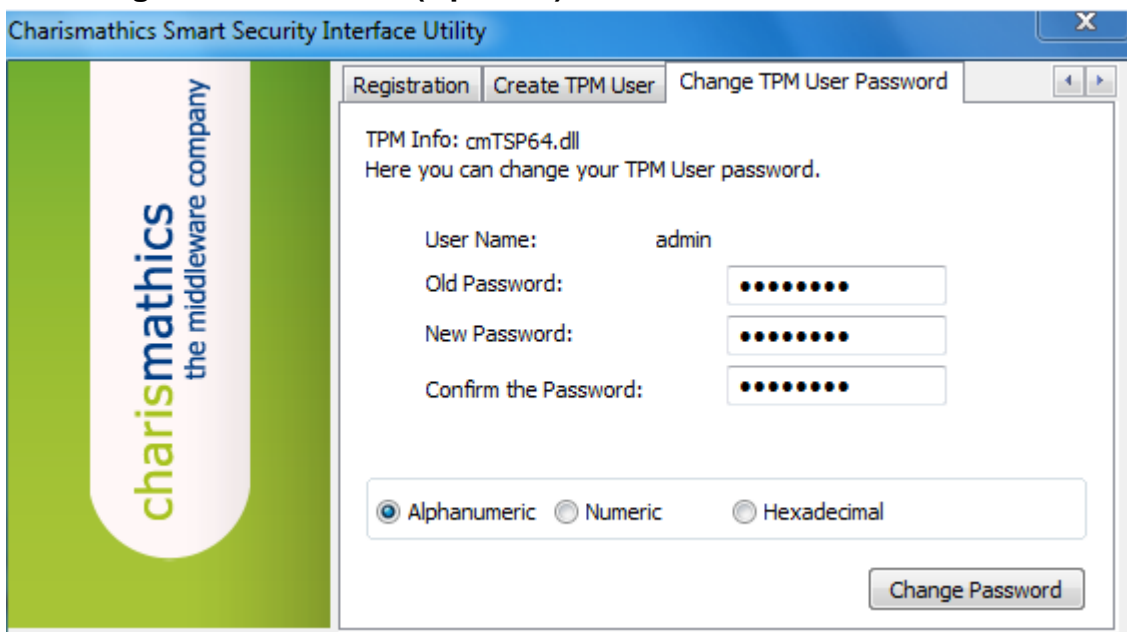
All TPM operations have a number of prerequisites:

- TPM hardware has to be present
- TSS module has to be installed
- TPM Ownership must have been taken, e.g. with CSSI – Security token configurator tool.



Creating a TPM User only works on systems where a TPM User for the current user does not yet exist  
A TPM User can only be created for the currently logged in windows user.

### 5.6 Change TPM Password (Optional)



Changing a TPM User only works on systems where a TPM User for the current user has been created.  
Changing the TPM Password requires entering the old password once and the new password twice in the according fields.

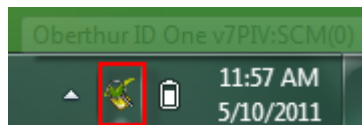
## 6 Register Tool

This tool is installed only on Windows XP. For Vista and above certificate propagation is done automatically using the system services so this tool is not needed.

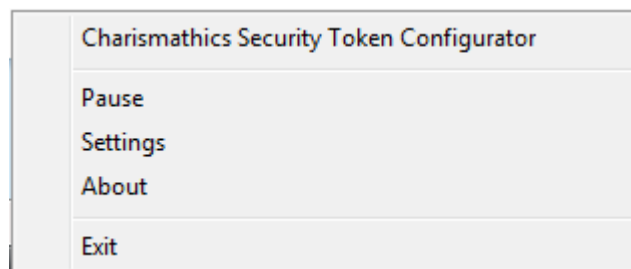
To make certificates accessible for Microsoft Windows applications like Internet Explorer or Outlook Express under Windows XP, you can automatically register the certificates from your smart card in the certificate store of Windows. The settings for this registration are configured in this Register Tool.

The default functionality is as follows: as soon as a smart card is inserted into the card reader, the certificates are automatically registered, as long as the Register Tool is active. On smart card removal, the certificates are **not** automatically unregistered. If this is desired, you can adjust this using the "Settings".

You can call the Register Tool of **Charismathics Smart Security Interface** either over the Start menu or over the tray icon:



Then you get the possibilities of starting the Charismathics Security Token Configurator or the Charismathics Smart Security Interface Utility to pause the Register Tool, to configure Settings, to read information or to terminate the Register Tool, which is now explained.

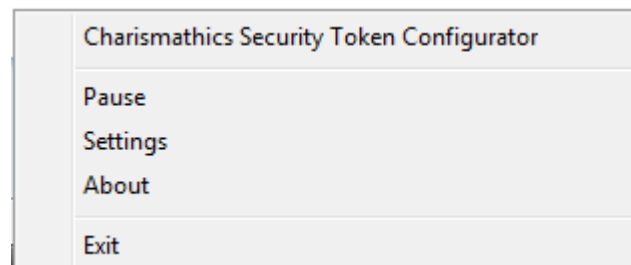


### 6.1 Start CSSI security token configurator and Start CSSI Utility

The context menu of the system tray icon offers the choice of starting either the user edition (Charismathics Smart Security Interface Utility) and the admin edition (Charismathics Security Token Configurator). Refer to chapters 4 and 5 regarding these tools.

### 6.2 Pause/ Continue

If automatic registration of the certificates on the token is not desired, you can pause the Register Tool. Select Pause from the context menu of the system tray icon to temporarily stop automatic registration. Once paused, you can select Continue from the same menu.



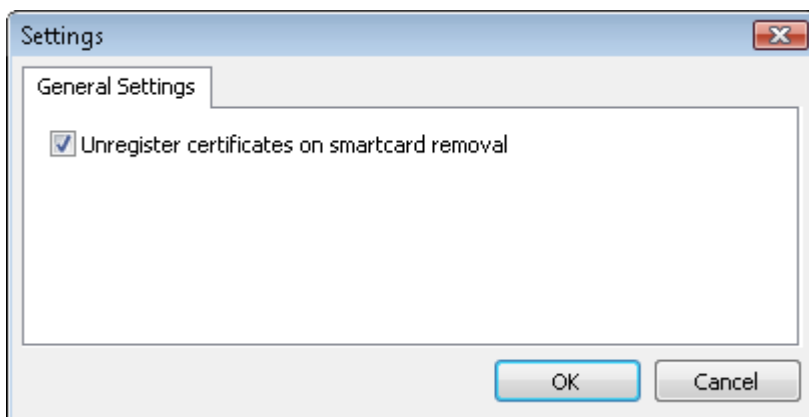
In addition to adding the certificates to the user store, they can also be added to the machine store. Set the registry value at:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\charismathics\smart security interface]
To store certificates in the machine store, set "CSP_RegisterMachineStore"=hex:01.
```

Revert the value to `hex:00` to disable storing the certificates in the machine store.

### 6.3 Settings

The default functionality of the Register Tool is to register certificates automatically as soon as a token is inserted. Once the token is removed the certificates can be unregistered automatically. If this is desired, you can configure this using "Settings".



An alternate way of accessing this option is modifying the registry entry in

[HKEY\_LOCAL\_MACHINE\SOFTWARE\charismathics\smart security interface]

To deactivate automatic unregistering, use: "CSP\_DeactivateUnregister"=hex:00.

To activate automatic unregistering, use: "CSP\_DeactivateUnregister"=hex:01.

### 6.4 About

For information about the version of the Register Tools and the manufacturer charismathics gmbh, select "About" in the menu of the tray icons:

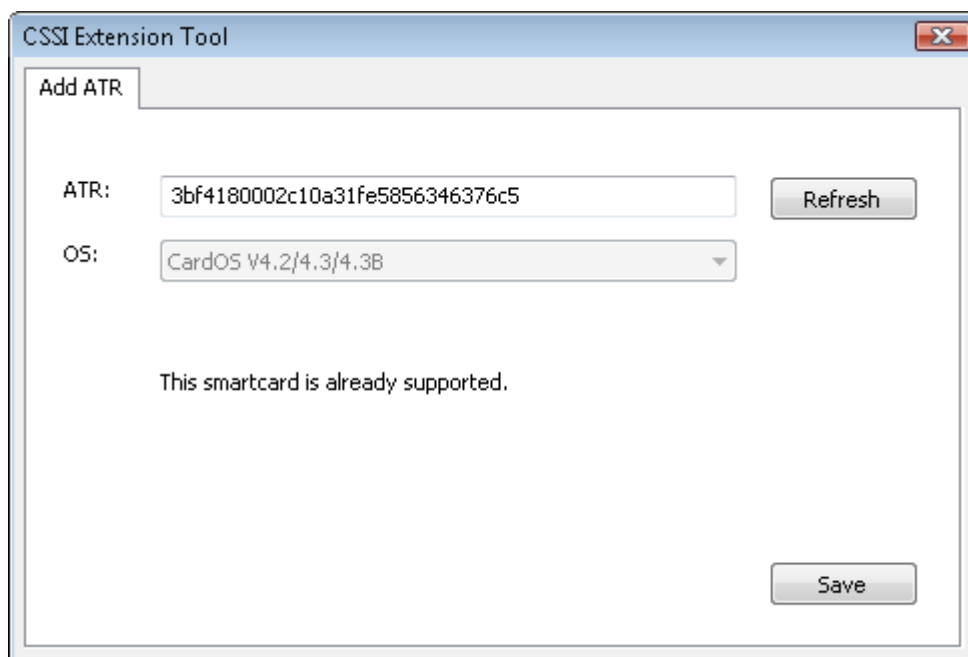


### 6.5 Exit

With "Exit" in the menu of the tray icon you can end the Register Tool.

## 7 Charismathics Extension Tool

The Charismathics Extension Tool can be used to associate smart card operating systems with new ATRs. Without a valid association, correct operation of the smart card can not be guaranteed.



Follow these steps to make a new ATR/Card OS association:

1. Insert the smart card into the reader.  
The card ATR is displayed in the upper field.
  - a. If an OS is associated with the ATR, the OS field is locked and can not be changed while the card is in the reader.
  - b. If no OS is associated with the ATR, select the correct OS or a close as possible match.
2. Press "Save" to store the information.

If the actual Operating System on the card is either unknown or not available, select one that matches the OS most closely, e.g. select the generic "JCOP" OS entry if the exact JCOP xx version number is not known.

**Note: On Windows Vista and above, you must run this tool as administrator.**

## 8 CSP of Charismathics Smart Security Interface

The Windows operating system supports cryptographic functionalities like encryption and digital signature by the so-called Crypto-API. Furthermore, CSPs (Cryptographic Service Providers) enable programs to support smart cards. During the installation of **Charismathics Smart Security Interface** the **Charismathics Smart Security Interface-CSP** (in short cmCSP) has been added.

Using cmCSP enables a number of programs and functions that come with a Windows Operating System, like Outlook Express, Internet Explorer, network login and VPN-login to use smart-cards, USB tokens and TPM. They will be explained in the following.

Charismathics Smart Security Interface includes also a minidriver for any CardOS V4.x smart card. The minidriver corresponds to the BaseCSP specification V6. For more information regarding the description of the minidriver requirements, please visit Microsoft website <http://www.microsoft.com/whdc/device/input/smartcard/sc-minidriver.msp>.

Smart card vendors can write card minidrivers to present a consistent interface to their smart card type to the Microsoft® Smart Card Base Cryptographic Service Provider (CSP) or Crypto Next Generation (CNG) Key Storage Provider (KSP) and to the Smart Card Management Interface. These card minidrivers plug in to Windows® operating system code.

**NOTE:** Here, you will not find a description how to configure your Microsoft environment for the use of smart cards or USB tokens. Please consult the help files for Outlook Express and the Internet Explorer. To configure the network login and the VPN-login for smart cards please consult the documentation of the Windows 2000 Server.

If you need support regarding the implementation or realization, charismathics team can help you. Feel free to contact your account manager.

### 8.1 General Proceedings

In order to use a Microsoft product in connection with the CSP for the first time on a certain computer the certificates on the token must have been registered. Please refer to chapter 6 "Register Tool" and 4.9.9 "Function "Register Certificate"" if you want to learn more about how to register your certificate.

The token must contain keys and certificates. There are several different possibilities to obtain these.

- Generating a key pair and corresponding certificate directly on the smart card with the functions of standard browsers, like Internet Explorer or Netscape. Thereby the token is accessed using the modules of **Charismathics Smart Security Interface**, i.e. correspondingly by cmCSP or cmP11. Point your browser to `http://<Servername of Enterprise-CA>/certsrv` if a certificate server is available.
- Import of existing keys and certificates on the smart card which were generated by other CAs or trust centers.
- Generation of key pair and corresponding self-signed certificate directly on the smart card by the administration tool **Charismathics Smart Security Interface**. Please note that the use of self signed certificates makes sense only in environments without PKI or for testing.

**Note:** If you request a certificate from a trust center, you might be requested to choose a security module, e.g. a token. In this case choose the corporate profile, the cmCSP or the cmP11. Furthermore, your smart card has to be inserted in the card reader, so that certificates can be written on it.

The programs must be configured to work with your keys and certificates. Some programs require root-certificates to be installed in certain directories, others require registering the certificate.

In the following chapters only the special features of the corresponding application will be explained.

## 8.2 Smart Card Login to a Windows 2000 Domain

The following is a brief outline of the steps involved in setting up smart card or USB token login.

- Setup of ADS. Please ensure the correct configuration of the DNS-Server.
- Installation of the Enterprise CA and at least the templates "Enrollment Agent", "Smartcard Logon" and "Smartcard User".
- Then an Enrollment-Agent-Certificate must be generated and registered on the computer where the smart cards should be personalized.
- After that, the smart cards for users may be issued over the Enrollment Station.

## 8.3 SSL- Authentication with Smart Card over the Internet Explorer

To use certificates stored on hardware tokens for SSL connections, the certificate must have been registered with the Windows Certificate Store. This can be done using either the admin or user edition of CSSI and the Register Tool (refer to chapter 6 "Register Tool" or in section 4.8 "Register Certificate").

## 8.4 Outlook Express with Electronic Signature and Encryption via Smart Card

Electronic signing and encryption requires the certificates to be registered the same as for SSL connections. Once this is done, the desired certificate for signing and encryption can be chosen from "Tools → Accounts → E-Mail → Preferences → Security".

Normally, there are pull-down menus in the email windows that you may click encryption and/or signing an email in order to use the security functionalities. The verification of incoming signed emails for instance uses the red "signet" symbol in the right corner of the email window

In order for Outlook Express to automatically recognize the right key and corresponding certificate, the certificate should lie in the address book, i.e. the certificate should be imported into the "Digital IDs". Highlight the name in the address book and choose the tab "Digital IDs" over the context menu. On this tab you can import the certificate for the chosen contact.

## 8.5 Windows VPN-Login with Smart Card

You should generate keys and certificates with the Microsoft Enterprise-CA. Furthermore the certificate must be registered with the administration tool of **Charismathics Smart Security Interface** (refer to chapter 6 "Register Tool" and 4.9.9 Function "Register Certificate").

## 9 PKCS#11-Module of Charismathics Smart Security Interface

The use of software that supports PKCS#11 is enabled by the **Charismathics Smart Security Interface-PKCS#11** (abbreviated cmP11). The matter of applications and functionalities with tokens like network login, SSL, email security with Netscape and other producers are explained briefly.

**NOTE:** There are no description on how to configure each environment to use cmP11. If your application is not covered here, please consult the corresponding documentation that comes with the application.

**IMPORTANT:** The PKCS#11 module is a DLL by the name "cmP11.dll" and is installed in the system directory. Usually this is C:\Windows\system32.

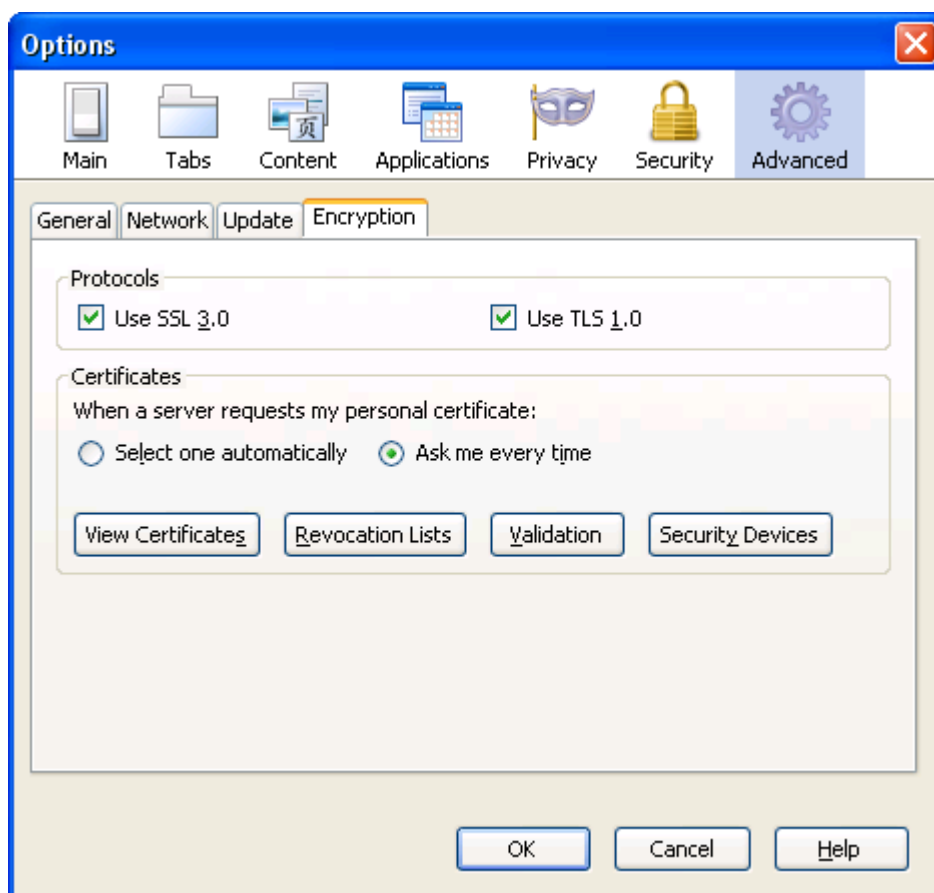
**Remark:** Despite strict measures for the quality of PKCS#11 modules by the different manufacturers, charismathics gmbh can not guarantee the compatibility with each PKCS#11 Module of a third party manufacturer.

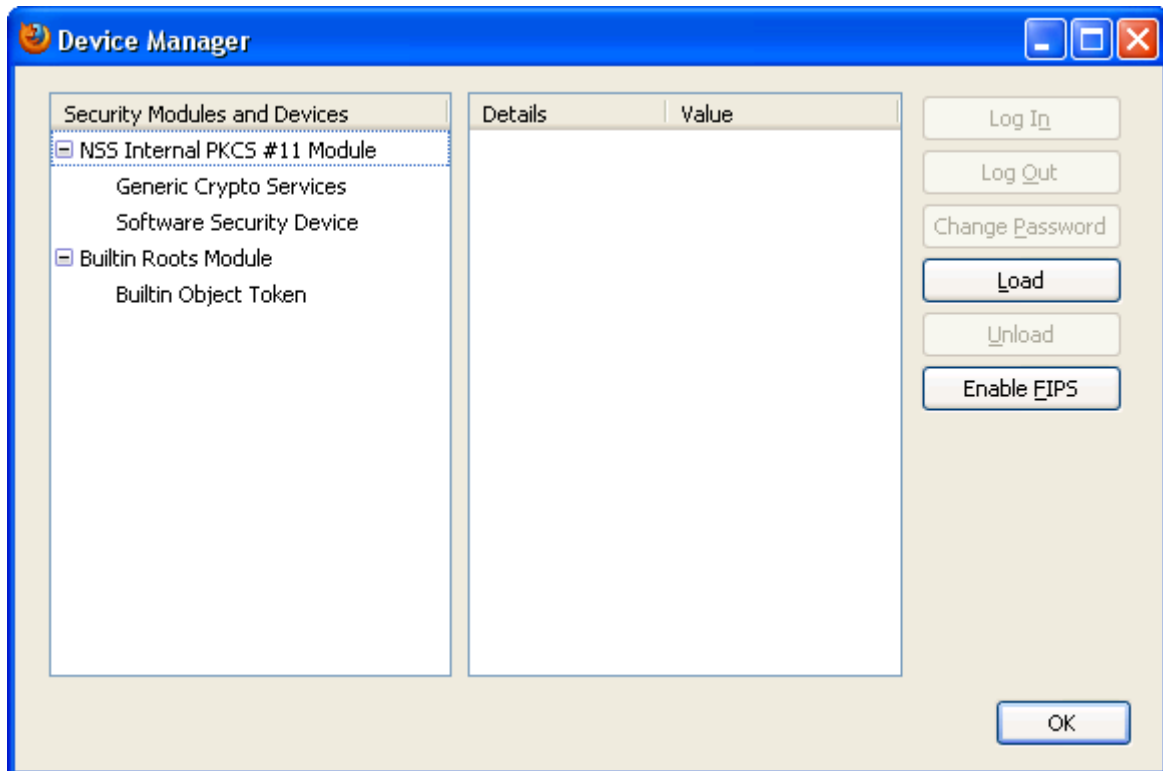
### 9.1 Smart Card Login to a Novell eDirectory (formerly NDS)

This requires a very good understanding of the administration of Novell servers. To enable smart card or USB token login to an eDirectory you explicitly need the product NMAS and the corresponding Universal Smartcard Login Method.

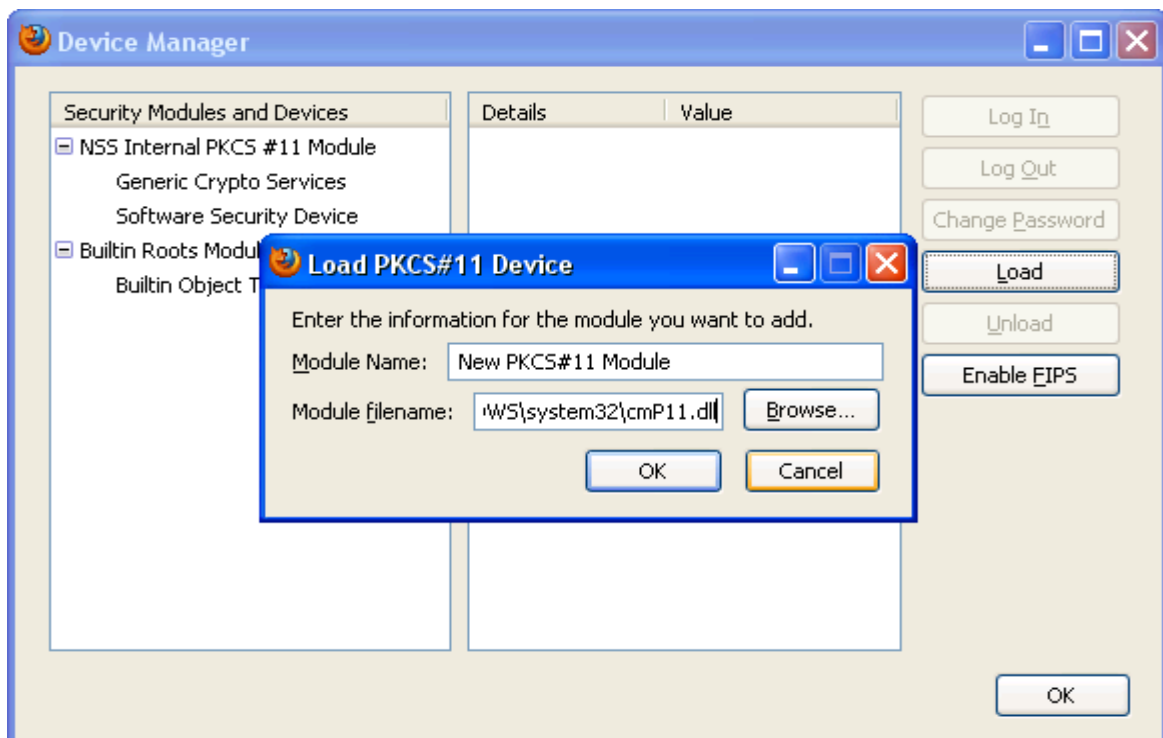
### 9.2 SSL-Authenticatin with Smart Card Over Mozilla Firefox

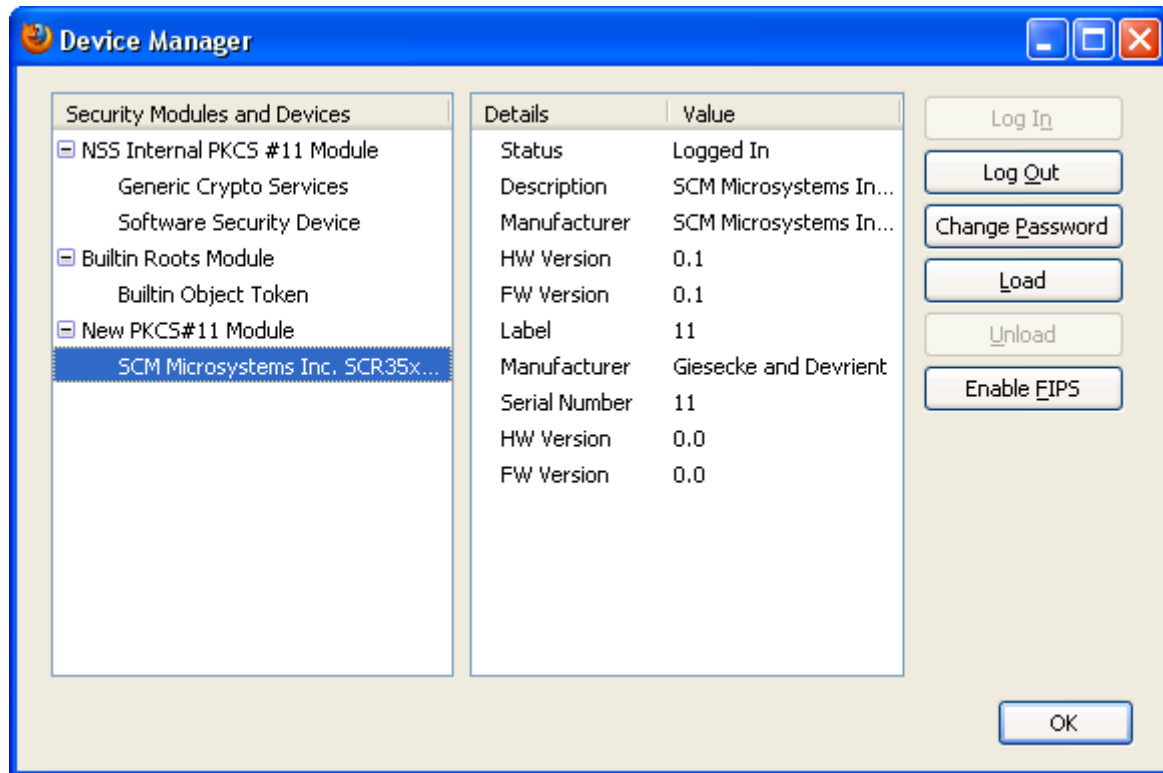
Launch Firefox - Go to Tools – Options – Advanced – Encryption – Security Devices – Load.





Click on Browse and locate cmP11.dll (cmP1164.dll for 64bit) and click on Open. Click on OK.



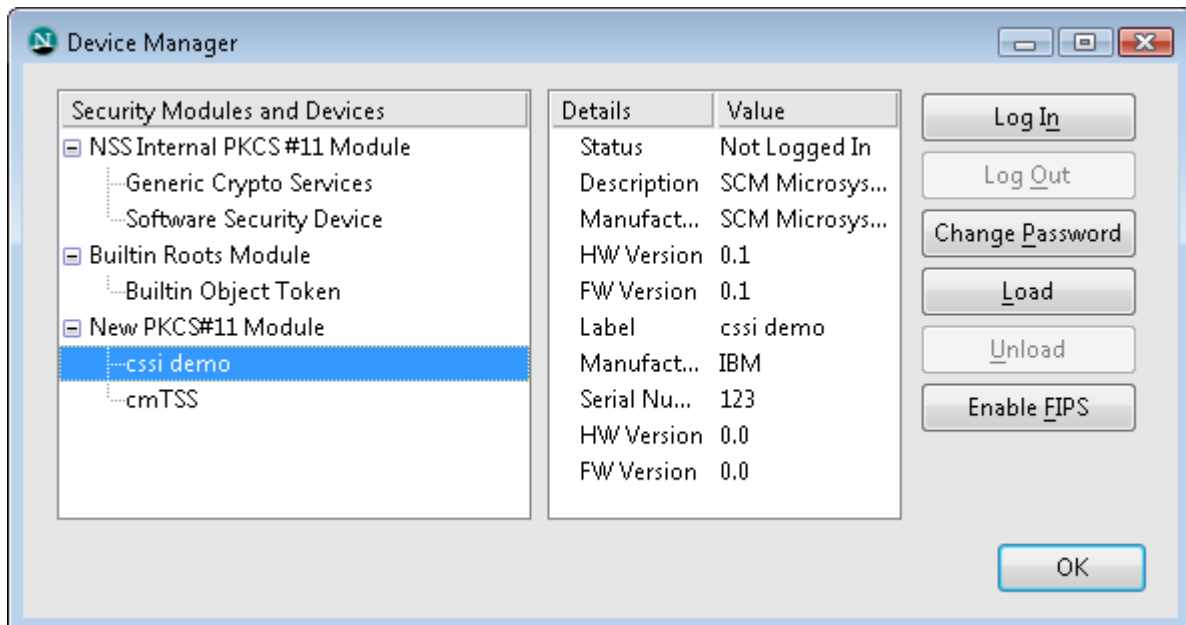


### 9.3 SSL- Authentication with Smart Card over Netscape

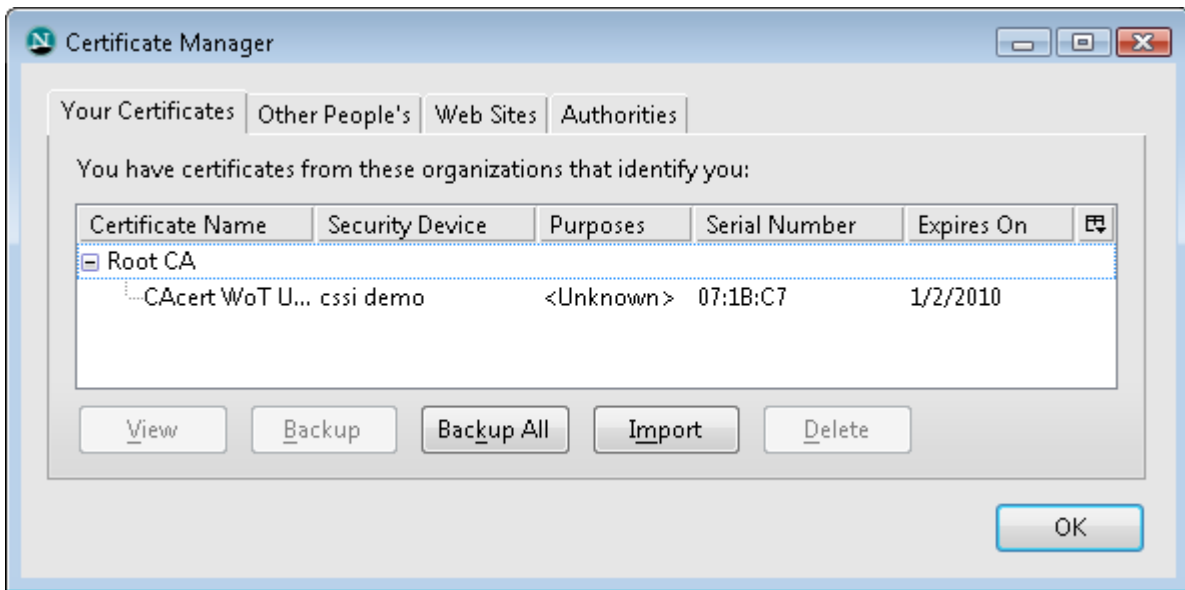
The notes for the configuration of Netscape are presented by example of version 7.

**Example:** Netscape 7.01

You can call "Manage Security Devices" in Netscape 7.01 via "Edit"→"Preferences"→"Privacy & Security"→"Certificates". From here you can load the cmP11, so that SSL and emails can be used with tokens.



Furthermore, you can call the Certificate Manager of Netscape from the same tab by clicking "Manage Certificates..."



#### 9.4 Email-Security by Smart Cards and Tokens with Netscape's Messenger

The notes for the employment of Netscape and screen shots to manage certificates and modules are available in the example of version 7 in the previous section.

Normally, there are pull-down menus in the email windows, where you can tick whether an email should be encrypted and/or signed. Functions for verification of received signed emails and decryption are available as well.

## 10 References

[PKCS#5] <http://www.rsasecurity.com/rsalabs/pkcs/index.html>

[PKCS#11] <http://www.rsasecurity.com/rsalabs/pkcs/index.html>

[MS\_CA] HOW TO: Configure a Certificate Authority to Issue Smart Card Certificates in Windows 2000: <http://support.microsoft.com/default.aspx?scid=kb;en-us;Q313274&sd=tech>

Guidelines for Enabling Smart Card Logon with Third-Party Certification Authorities:  
<http://support.microsoft.com/default.aspx?scid=kb;en-us;Q281245>

[MS\_SC] Windows 2000 Server Documentation, Smart card Administration:  
[http://www.microsoft.com/windows2000/en/server/help/default.asp?url=/windows2000/en/server/help/sag\\_SC\\_admin.htm](http://www.microsoft.com/windows2000/en/server/help/default.asp?url=/windows2000/en/server/help/sag_SC_admin.htm)

## 11 Information / Export Restrictions

Charismathics GmbH  
47 Sendlinger St  
80331 Munich  
Germany

Manual Revision: Mai 10, 2011

### © Copyright Charismathics GmbH 2002-2010

All rights reserved. Without the express prior written consent of charismathics you must not distribute, edit or translate copyrighted material.

### Trade Mark

All mentioned software and hardware names are in most of the cases trade marks and are liable to legal requirements.

### Please observe!

**The product delivered to you is liable to export control. Please observe the legal requirements of specific countries. For export out of the EU an export approval is necessary.**

## Appendix A: Reference for Developers

In this appendix there are detailed specification regarding the supported functions of the PKCS#11-standard, a synopsis of particular functions and a list of objects and mechanisms. This information is useful and necessary for application developers who want to develop their own applications supporting the cmP11.

### Functions according to PKCS#11-Standard

In the following there are three lists of functions according to PKCS#11-Standard. The lists are supported, incompletely supported, and not supported functions by **Charismathics Smart Security Interface**:

Supported Functions	Incompletely Supported Functions / Deviations	Not supported functions
C_CancelFunction <sup>a</sup>	C_GetObjectSize	C_CopyObject
C_CloseAllSessions	C_GetTokenInfo	C_DecryptDigestUpdate
C_CloseSession	C_Initialize	C_DecryptVerifyUpdate
C_CreateObject	C_OpenSession	C_DeriveKey
C_Decrypt	C_SignRecover <sup>b</sup>	C_DigestEncryptUpdate
C_DecryptFinal	C_SignRecoverInit <sup>c</sup>	C_DigestKey
C_DecryptInit	C_WaitForSlotEvent	C_GetFunctionStatus
C_DecryptUpdate		C_GetOperationState
C_DestroyObject		C_SeedRandom
C_Digest		C_SetOperationState
C_DigestFinal		C_SignEncryptUpdate
C_DigestInit		
C_DigestUpdate		
C_Encrypt		
C_EncryptFinal		
C_EncryptInit		
C_EncryptUpdate		
C_Finalize		
C_FindObjects		
C_FindObjectsFinal		
C_FindObjectsInit		
C_GenerateKey		
C_GenerateKeyPair		
C_GenerateRandom		
C_GetAttributeValue		
C_GetFunctionList		
C_GetInfo		
C_GetMechanismInfo		
C_GetMechanismList		
C_GetSessionInfo		
C_GetSlotInfo		
C_GetSlotList		
C_InitPIN		
C_InitToken		
C_Login		
C_Logout		
C_SetAttributeValue		
C_SetPIN		
C_Sign		

<sup>a</sup> returns CKR\_FUNCTION\_NOT\_PARALLEL

<sup>b</sup> use C\_Sign

<sup>c</sup> use C\_SignInit

C\_SignFinal  
C\_SignInit  
C\_SignUpdate  
C\_UnwrapKey  
C\_Verify  
C\_VerifyFinal  
C\_VerifyInit  
C\_VerifyRecover  
C\_VerifyRecoverInit  
C\_VerifyUpdate  
C\_WrapKey

---

## Synopsis of specific functions

### C\_Finalize

Parameter: pReserved (CK\_VOID\_PTR)

Description: Sessions will be closed.  
Slots will be closed.  
Reserved Memory will be freed.

Deviation: pReserved will be ignored.  
C\_Finalize will be called automatically on Finish.  
If C\_Initialize is called n times in succession (without C\_Finalize in between), C\_Finalize will only be carried out after the n time.

### C\_GetObjectSize

Parameter: hSession CK\_SESSION\_HANDLE  
hObject CK\_OBJECT\_HANDLE  
pulSize CK\_ULONG\_PTR

Description: The size of an object will be returned

Deviation: The returned size is the minimum size of an object, which means it does not contain the size for extra attributes like label, or id. The size of private objects is the default value.

### C\_GetSlotList

Parameter: tokenPresent CK\_BBOOL  
pSlotList CK\_SLOT\_ID\_PTR  
pulCount CK\_ULONG\_PTR

Description: Returns a list of identified Slots.  
It might occur that installed but not connected Slots will be in the list.  
The number of Slots may be obtained by passing pSlotList a Null-Pointer.  
If you want only the Slots with an inserted card set tokenPresent to true.

### C\_GetTokenInfo

Parameter: slotID CK\_SLOT\_ID  
pInfo CK\_TOKEN\_INFO\_PTR

Description: Returns whether a card is inserted in a Slot. If the card is not inserted, CKR\_TOKEN\_REMOVED will be returned.

Special Feature: Inserting or removing a card from a Slot is an Event (see C\_WaitForSlotEvent). If C\_GetTokenInfo will be called the Event will be finished, even if the card was removed and C\_GetTokenInfo CKR\_TOKEN\_NOT\_PRESENT has been returned.

### C\_Initialize

Parameter: CinitArg CK\_VOID\_PTR\_PTR  
 Description: Library will be initialized.  
 Slots will be created.  
 Inserted cards are read.  
 Deviation: CinitArg is expected in the format CK\_C\_INITIALIZE\_ARGS. From these the flags are picked out, in particular CKF\_LIBRARY\_CANT\_CREATE\_OS\_THREADS which decides over Multi threading. The rest is ignored. If C\_Initialize is called several times, CKR\_CRYPTOKI\_ALREADY\_INITIALIZED is returned. The number is taken in account (see C\_Finalize).

### C\_InitToken

Parameter: slotID CK\_SLOT\_ID  
 pPin CK\_UTF8CHAR\_PTR  
 ulPinLen CK\_ULONG  
 pLabel CK\_UTF8CHAR\_PTR  
 Description: Token will be initialized. SO pin is given by the parameter pPin, User pin will be reset to default 11111111. The maximal length of SO pin is 10 digits.  
 Special Feature: In Case Init (The token is empty): Card pin will be set to the same value as SO pin. After the token initialization, Card pin cannot be changed by PKCS#11, but SO pin can be changed using the function C\_SetPIN.  
 In Case Re-Init (The token is already initialized): The given SO PIN will be verified firstly. Then the User PIN will be reset to default 11111111, and all objects on token will be deleted. The SO PIN and Card PIN are unchanged.

### C\_OpenSession

Parameter: slotID CK\_SLOT\_ID  
 flags CK\_FLAGS  
 pApplication CK\_VOID\_PTR  
 Notify CK\_NOTIFY  
 phSession CK\_SESSION\_HANDLE\_PTR  
 Description: Opens a new session on the Slot.  
 Deviation: Notify and pApplication are ignored and should be set to NULL\_PTR. Sessions can only be opened, if a card is inserted.  
 Special Feature: If a session is opened and then the card will be removed, all sessions on the Slot will return CKR\_DEVICE\_REMOVED. If there is an error with CKR\_DEVICE\_REMOVED, CKR\_TOKEN\_NOT\_RECOGNIZED or CKR\_TOKEN\_NOT\_PRESENT a pause is automatically produced on this Slot for all sessions.  
 If a paused session is used again, this session will be reopened automatically.  
 If a card is inserted into or removed from a Slot, then this is an Event (see C\_WaitForSlotEvent). If C\_OpenSession is called, the Event will be finished, even if the card has been removed and C\_OpenSession returned CKR\_TOKEN\_NOT\_PRESENT.

### C\_WaitForSlotEvent

Parameter: flags CK\_FLAGS  
 pSlot CK\_SLOT\_ID\_PTR  
 pReserved CK\_VOID\_PTR (= NULL\_PTR)



CKA_PRIME_2	Pkcs12 Format	not readable
CKA_EXPONENT_1	Pkcs12 Format	not readable
CKA_EXPONENT_2	Pkcs12 Format	not readable
CKA_COEFICIENT	Pkcs12 Format	not readable
CKA_MODIFIABLE	TRUE	read only(**)
CKA_LOCAL	TRUE	(**)(***)
CKA_START	<empty>	(***)
CKA_STOP	<empty>	(***)
CKA_EXTRACTABLE <sup>8</sup>	FALSE	read only(**)
CKA_NEVER_EXTRACTABLE <sup>2</sup>	TRUE	read only(**)

(\*) Can only be read if a corresponding certificate exists.

(\*\*) Returns no error on trying to write.

(\*\*\*) Is not supported.

#### CKO\_PUBLIC\_KEY (CKK\_RSA)

Attribute	Value	Access
CKA_CLASS	CKO_PUBLIC_KEY	read only
CKA_LABEL	<alias>	read/write
CKA_ID	<number>	read/write
CKA_KEY_TYPE	CKK_RSA	read only
CKA_TOKEN	TRUE	read only
CKA_PRIVATE	FALSE	read only
CKA_SUBJECT	<alias>	read only(*)
CKA_ENCRYPT	TRUE	read only(**)
CKA_VERIFY	TRUE	read only(**)
CKA_VERIFY_RECOVER	TRUE	read only(**)
CKA_WRAP	FALSE	read only(**)
CKA_MODULUS	pkcs12 Format	read only
CKA_PUBLIC_EXPONENT	pkcs12 Format	read only
CKA_MODIFIABLE	FALSE	read only(**)
CKA_LOCAL	TRUE	(**)(***)
CKA_START	<empty>	(***)
CKA_STOP	<empty>	(***)

(\*) Can only be read, if a corresponding certificate exists

(\*\*) Returns no error on trying to write

(\*\*\*) Is not supported

#### CKO\_DATA

##### General Data

Attribute	Value	Access
CKA_CLASS	CKO_DATA	read only
CKA_LABEL	<alias>	read/write
CKA_VALUE	<data>	read/write
CKA_TOKEN	TRUE	read only
CKA_PRIVATE	FALSE	read only(**)
CKA_APPLICATION	<alias>	read/write
CKA_MODIFIABLE	TRUE	read only(**)

(\*\*) Returns no error on trying to write.

## Mechanism

### Sign (RSA):

Description: Signs data

Order: C\_SignInit, C\_SignUpdate, C\_SignFinal  
or C\_SignInit, C\_Sign  
C\_Sign works as if C\_SignUpdate and then C\_SignFinal were called.  
C\_SignUpdate processes the data immediately.

Special Feature: Order C\_SignInit, C\_Sign(C\_SignUpdate, C\_SignFinal), C\_Sign  
(C\_SignUpdate, C\_SignFinal) where on the first C\_Sign (resp.  
C\_SignFinal) NULL\_PTR will be passed for the signature and only the  
length of the signature will be returned. The signature will be returned on  
the second C\_Sign (resp. C\_SignFinal). If C\_SignUpdate is called for the  
second time, the data must match with the data of the first time. A third call  
is not possible. For another signature C\_SignInit must be called first.

### Verify (RSA):

Description: Verifies a signature. VerifyRecover returns only the data (normally as a  
hash-value)

Order: C\_VerifyInit, C\_VerifyUpdate, C\_VerifyFinal  
or C\_VerifyInit, C\_Verify  
or C\_VerifyRecoverInit, C\_VerifyRecover  
C\_Verify works as if \_VerifyUpdate and then C\_VerifyFinal were called.  
C\_VerifyUpdate stores data only temporarily.  
C\_VerifyRecover returns the signed data.

Special Feature: Order C\_VerifyRecoverInit, C\_VerifyRecover, C\_VerifyRecover, where on  
the first C\_VerifyRecover a NULL\_PTR will be passed as data. It returns  
only the length of the data. The data will be returned on the second  
C\_VerifyRecover. A third call is not possible. For further verifications  
C\_VerifyRecoverInit must be called first.

### Encrypt (RSA):

Description: Encrypts data.

Order: C\_EncryptInit, C\_EncryptUpdate, C\_EncryptFinal  
or C\_EncryptInit, C\_Encrypt  
C\_Encrypt works as if C\_EncryptUpdate and then C\_EncryptFinal were  
called.

Special Feature: C\_EncryptUpdate stores the data temporarily. And you can pick up finished  
data with C\_EncryptUpdate. If you don't do this, you receive with  
C\_EncryptFinal all data at one time. The data is however only once availa-  
ble!

### Decrypt (RSA):

Description: Decrypts data.

Order: C\_DecryptInit, C\_DecryptUpdate, C\_DecryptFinal  
or C\_DecryptInit, C\_Decrypt  
C\_Decrypt works as if C\_DecryptUpdate and then C\_DecryptFinal were  
called.

Special Feature: C\_DecryptUpdate stores the data temporarily. And you can pick up finished  
data with C\_DecryptUpdate. If you don't do this, you receive with  
C\_DecryptFinal all data at one time. The data is however only once availa-  
ble!

**Digest (Hashfunctions SHA1, MD2, MD5):**

Description: A hash value is calculated from the data.

Order: C\_DigestInit, C\_DigestUpdate, C\_DigestFinal  
or C\_DigestInit, C\_Digest  
C\_Digest works as if C\_DigestUpdate and then C\_DigestFinal were called.  
C\_DigestUpdate processes the data immediately.

## Appendix B: Non-Standard Functions in PKCS#11 DLL

Two non-standard functions for the token initialization are added to the PKCS#11 library cmP11.

<b>CK_RV EraseProfile</b>	slotID	CK_SLOT_ID /* ID of the token's slot */
	pCardPIN	CK_BYTE_PTR /* CardPIN value */
	ulCardPINLen	CK_ULONG /* length of CardPIN value */
Description:	Erase the existed profile on a token. In order to erase the profile, the Card-PIN must be verified.	

<b>CK_RV CreateProfile</b>	slotID	CK_SLOT_ID /* ID of the token's slot */
	pProfile	CK_UTF8CHAR_PTR /* profile name, null terminated */
	pSerNum	CK_BYTE_PTR /* serial number */
	ulSerNumLen	CK_ULONG /* length of serial number */
	pCardPin	CK_BYTE_PTR /* CardPIN value */
	ulCardPINLen	CK_ULONG /* length of CardPIN value */
	pSOPIN	CK_BYTE_PTR /* SO PIN value */
	ulSOPINLen,	CK_ULONG /* length of SO PIN value */
	pUserPIN	CK_BYTE_PTR /* UserPIN value */
	ulUserPINLen	CK_ULONG /* length of UserPIN value */
	pLabel	CK_UTF8CHAR_PTR /* 32-byte token label (blank padded) */
	ulUserPINRetry	CK_ULONG /* retry counter of UserPIN */
Description:	Create a profile. The possible profile names are "CORPORATE", "PKCS15", "CNS" and "FINEID". Usually, the token must be empty or the old profile must be erased before the new profile is written to the token.	
Remark:	Not all profiles are supported by all smartcards. CardOS V4.x supports: CORPORATE, PKCS15, CNS CardOS M4.0(a) supports: CORPORATE JavaCards support: CORPORATE, PKCS15, FINEID ACOS supports: CORPORATE.	

## Appendix C: Log Information

Logging information may serve to find and correct errors but impacts performance. In general, logging should be disabled. The logger should only be used by experienced users or when asked to. The log-file format is as follows: Each entry contains the function name, the parameter before and after the function call and the result of the function. Private information is hidden by a static string "[-----]", so only the length is readable.

### Convenience Files

To enable logging with the default settings .reg files can be found in the installation directory.

```
<program files>\Charismathics\smart security interface x.zz\  
CSSI_Param.reg contains logging parameters for PKCS#11 and the CSP.
```

### Registry Settings

Logging is controlled by registry entries stored in

```
[HKEY_LOCAL_MACHINE\SOFTWARE\charismathics\smart security interface]
```

```
"LogFile_mode"=dword:00000001  
Use 1 to enable logging, 0 to disable logging.
```

```
"PKCS11_LogFile_name"="c:\\temp\\cmP11.log"
```

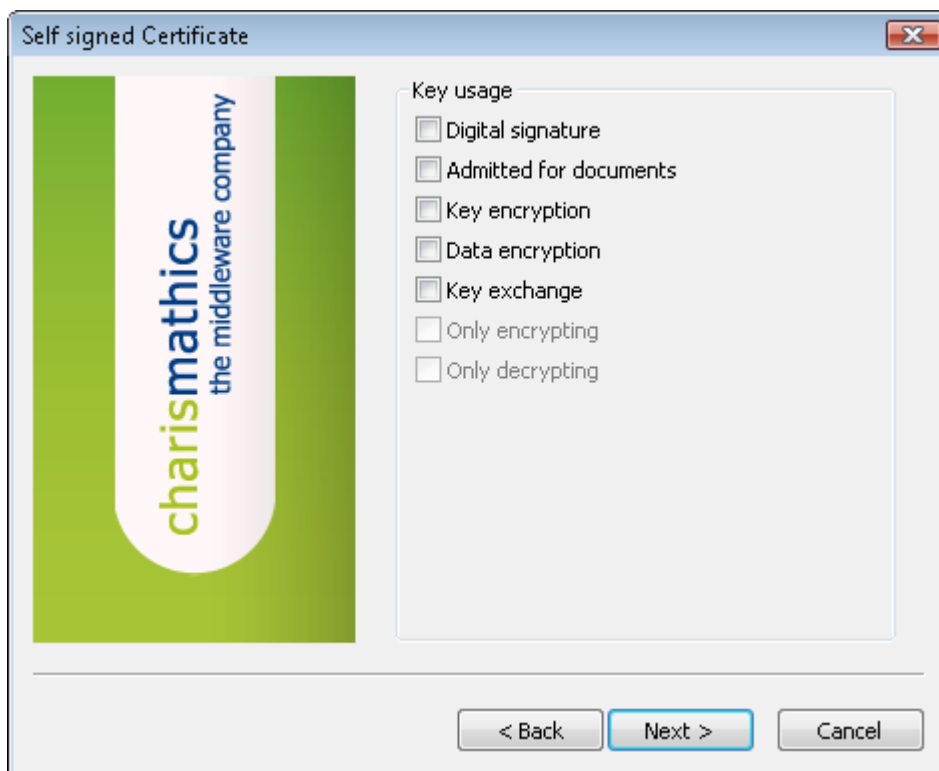
```
"CSP_LogFile_name"="c:\\temp\\cmCSP.log"
```

```
"TSP_LogFile_name"="c:\\temp\\cmTSP.log"
```

```
"TCS_LogFile_name"="c:\\temp\\cmTCS.log"
```

Select a logging file and directory. Use only absolute paths names. Remember to maintain backslash '\\' doubling.

## Appendix D: Certificate Attributes (Key Usage)



A short explanation of the options follows:

1. Digital Signature: The certificate can be used for authentication and digital signature.
2. Admitted for Documents: The certificate verifies signatures that check the liability and bindingness of documents (except signatures of certificates and CRLs of CA).
3. Key encryption: Encryption of keys for the purpose of their transmission.
4. Data encryption: Encryption of data for the purpose of transmission but not of keys.
5. Key exchange: Employment of the key to agree on other keys, e.g. a Diffie-Hellman key.
6. Only encrypting: This option is mutually exclusive with all other options.
7. Only decrypting: This option is mutually exclusive with all other options.