Linux on Z and LinuxONE
Trusted Key Entry 9.1

How to set an AES master key

SC34-7712-00

This edition applies to the TKE version 9.1 and to all subsequent versions and modifications until otherwise indicated in new editions.

Contents

About this document

This publication outlines a procedure how to create an AES master key on a cryptographic coprocessor configured in CCA coprocessor mode (shortly referred to as CCA coprocessor) using the Trusted Key Entry (TKE) workstation. The cryptographic coprocessor is connected to an IBM Z® mainframe that runs an instance of Linux on Z.

What this document describes

There are multiple master key types and different methods how to create and set these keys using TKE capabilities. This document focuses on the creation and setting of an AES master key on a CCA coprocessor.

What this document does not describe

This document does not describe the complete process of setting up a comprehensive security concept, nor does it demonstrate all security features available from the TKE workstation.

Further information

For information about sophisticated features, for example, for using TKE domain groups, refer to the *Trusted Key Entry Workstation User's Guide* (SC14-7511) from the IBM Resource Link or from the IBM Knowledge Center.

Terminology

A cryptographic coprocessor is also called a cryptographic adapter. The TKE graphical user interface uses the term *crypto module* for a cryptographic coprocessor.

Chapter 1. Getting started

As a single point of control, you require a Trusted Key Entry workstation (TKE) to securely manage multiple cryptographic coprocessors with their domains, master keys and operational keys stored inside.

The TKE provides the following secure services:

- Loading and maintaining master and operational keys into the host cryptographic hardware.
- Configuring and managing the cryptographic coprocessors in the host, for example, configure a coprocessor for running in CCA coprocessor mode.
- Providing a host cryptographic hardware migration feature. The TKE allows you to collect data from one host cryptographic coprocessor and apply the data to another host cryptographic coprocessor.
- Managing the involved smart cards.

A TKE comprises the following components:

- the hardware: a workstation with a cryptographic coprocessor which is used for secure communication with connected hosts
- the LIC (Licensed Internal Code) which includes an embedded operating system which supports one single application. No other applications or products can be installed on the TKE.
- smart card readers and smart cards.

Each TKE version is closely associated with the host platforms that it supports.

The TKE implementation uses a zone concept to ensure the secure transfer of key parts. After a short introduction of this concept, this document outlines the process how to initialize a certificate authority smart card (CA smart card) and the required TKE smart cards within a zone. Following this, you are guided through a scenario how to set a master key on a CCA coprocessor.

Note: The Trusted Key Entry workstation is a powerful appliance used to manage IBM Z cryptographic coprocessors. It provides hardware-based key management services with proper encryption strength, dual controls, and security-relevant auditing.

There are multiple ways and methods for setting master keys on a cryptographic adapter (coprocessor). New users of the TKE services may be overwhelmed by all the offered features and functions.

Therefore, the illustrated scenario presents one quick and easy path, but nevertheless a valid real-life approach, to generate and activate an AES master key on a domain of a cryptographic adapter configured in CCA coprocessor mode. This master key is composed from two key parts, which are individually generated and owned by two persons in a Linux on Z installation.

Figure 1 on page 2 depicts the environment in which the TKE workstation applications work. In Linux on Z, the TKE communicates with the host coprocessors using a TKE daemon called `catcher.exe`. This daemon is listening on a certain port for TKE commands.

In addition, a support element (SE) is required for cryptographic configuration. This is a dedicated workstation used for monitoring and operating IBM Z hardware. TKE commands must be permitted on the SE before any commands issued by the TKE workstation can be processed.

A master key is finally set within one or more domains of the attached coprocessors.

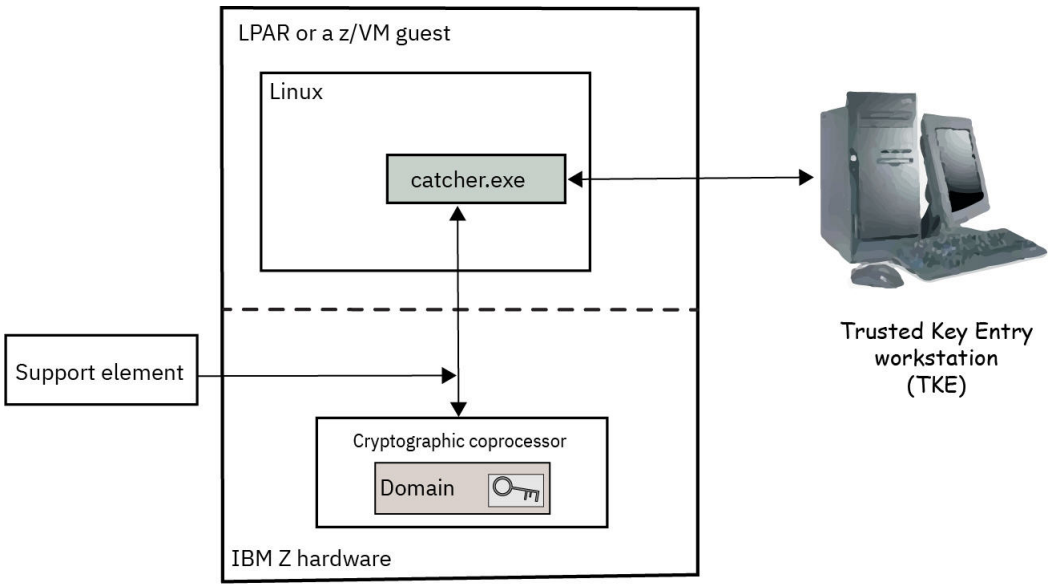

Figure 1. The TKE workstation environment

Scenario

An AES master key, which is composed from two key parts, is created and activated.

In the scenario described in this document, a security environment is established with two authorities for module administrators who manage the cryptographic coprocessor, and another two authorities for key administrators who create and activate a new master key. Only if both authorities work together, the required tasks can be completed. This approach is called dual control security.

The key parts are generated on two separate smart cards by two key administrators with different privileges. The key parts are loaded from the smart cards onto the cryptographic coprocessor.

The first key administrator is authorized to create and load the first key part. The second key administrator is authorized to create and load the second key part. Loading the second key part will automatically create the final master key. Both key administrators will be authorized to activate the new master key.

Prerequisites

On your Linux host system, you must install the CCA RPM or DEB package, which contains a Linux TKE daemon. This daemon must be running and ready to receive TKE requests. Also, you must enable the involved coprocessor to perform TKE commands.

You need to start the `catcher.exe` program, which is the Linux TKE daemon to handle administrative commands between the TKE and the cryptographic coprocessors. You can use the CSUTKEcat system initialization script to handle the daemon via `systemctl`. The `catcher.exe` daemon is automatically started by the CSUTKEcat system initialization script when Linux starts. You can also use this script to start or stop the `catcher.exe` from the command line. To start the `catcher.exe`, use the command:

```
# systemctl start CSUTKEcat.service
```

You must ensure that the firewall of your Linux system allows to access the `catcher.exe` via port 50003, because this daemon listens for TKE commands on this port. These commands are translated into **`ioctl`** commands which communicate with the `zcrypt` device driver.

To verify whether the `catcher.exe` daemon is running on your system, enter the following command and look for the daemon in the output list:

```
# ps ax

...
 9689 ?        Ss     0:11 /opt/IBM/CCA/bin/catcher.exe
...
```

This description shows how to set a master key on a cryptographic coprocessor that is running in CCA mode. From the support element (SE), you must at first enable such coprocessors to perform TKE commands.

Therefore, logon to the appropriate support element. Open **Systems Management**, then select the system with the attached cryptographic coprocessor. In the list of **Tasks**, navigate to **Configuration** and open the **Cryptographic Configuration** dialog (Figure 2 on page 3). Select the appropriate cryptographic coprocessor and press the **TKE Commands...** button. In the upcoming dialog, select the **Permit TKE commands** check box and confirm your request when prompted. This action changes the text entry in the **TKE commands** column in Figure 2 on page 3 from **Denied** to **Permitted**.

Fast path: **Logon SE → System Management → Tasks → Configuration → Cryptographic Configuration**

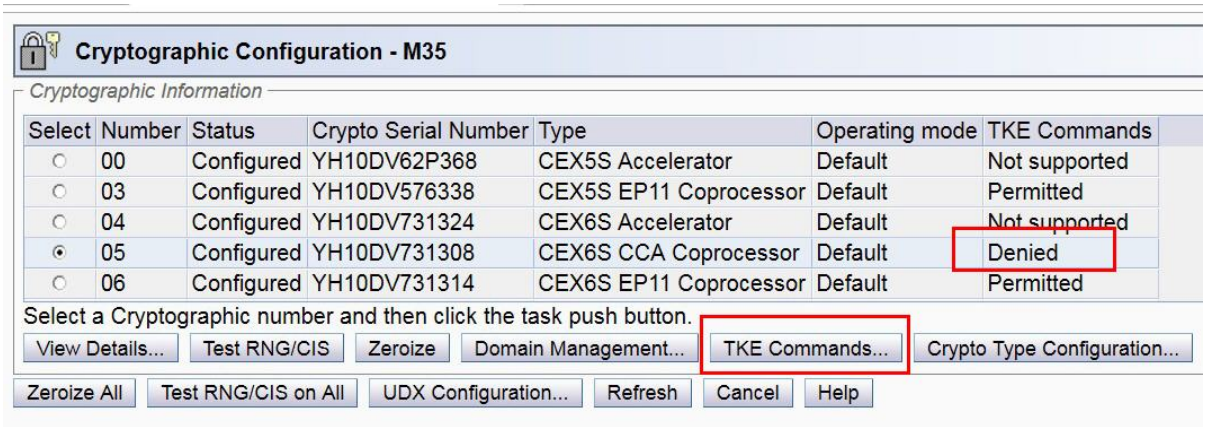

Figure 2. Permit TKE commands

Starting the TKE console

You can perform all tasks required for configuring your smart cards and for setting a new master key on attached cryptographic coprocessors from the TKE console on the TKE workstation.

You can configure that the Trusted Key Entry welcome dialog automatically loads on start-up. Click on **Launch the Trusted Key Entry web application**. Then the initial **Trusted Key Entry Console** (Figure 3 on page 4) opens and offers access to a set of commonly used applications and utilities available on the TKE workstation.

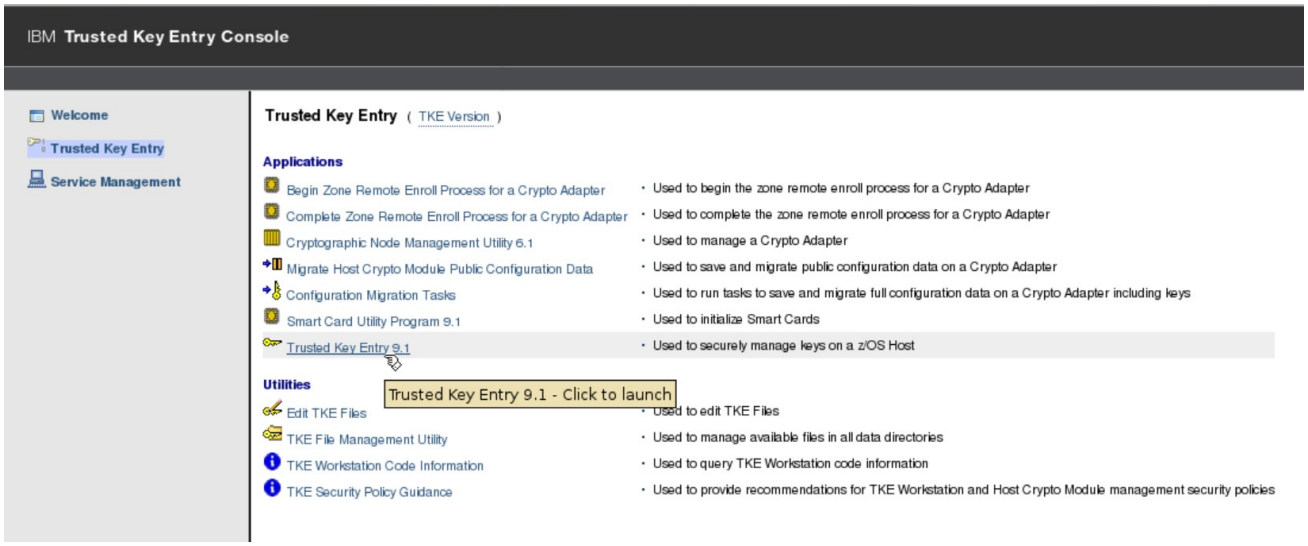

Figure 3. TKE Console - starting the Trusted Key Entry program

What to do next

Before you can create and load a master key, you must have completed all sub-tasks described in Chapter 2, “Establishing the security environment,” on page 5.

Chapter 2. Establishing the security environment

For a key management on the host cryptographic coprocessors using smart cards, you first need to enable the TKE smart card support and establish the connection between the applicable host and the TKE workstation. Also, use the **Setup Module Policy** wizard to create required authorities and assign signature keys and roles to them.

Complete these one-time tasks before you can use the secure Trusted Key Entry features for key management:

- “Enabling the TKE smart card support” on page 5
- “Creating and connecting a host ” on page 11
- “Creating the required roles and authorities” on page 15

Enabling the TKE smart card support

For the scenario described in this document, an enabled smart card environment is required. You can use a wizard to create and initialize the smart cards required for setting up this environment.

Zone concepts

Smart-card support for a TKE environment is designed around the concept of a zone to ensure the secure transfer of key parts between the members of the zone. A zone is created with a zone ID when you use the **Smart Card Utility Program** (SCUP) to create a CA smart card. In other words, a zone is defined by a CA smart card.

In the described scenario, these zone members can communicate with each other:

- a certificate authority (CA) smart card
- a Trusted Key Entry workstation (TKE) that was enrolled in a zone using the CA smart card
- four TKE smart cards, two for the module administrators and two for the key administrators. All TKE smart cards are created using the CA smart card.

Within a zone, you use the CA smart card when you create the required TKE smart cards. The TKE workstation must be enrolled in the same zone as the TKE smart cards.

Smart card support provides the ability to manage master key parts. The enrolled TKE smart cards are used to create and store key parts, or to load the key parts onto the cryptographic coprocessor. In addition, the smart cards can contain a key that is used to sign the load commands.

A CA smart card is protected by two 6-digit PINs. TKE smart cards are secured by one 6-digit PIN. In your environment, you can distinguish between a CA administrator (owning the CA smart card), the module administrators managing access to the domains of a cryptographic adapter, and the key administrators owning the TKE smart cards for key generation.

Initialize a CA smart card and TKE smart cards

A certificate authority (CA) smart card is an entity that establishes a zone using the **Smart Card Utility Program** (SCUP). A CA smart card is protected by two 6-digit PINs. This CA smart card is required to create and initialize TKE smart cards. Use the **TKE Smart Card Wizard** to accomplish the initialization of both card types.

Before you begin

In a real live production environment, this task must be performed in common by all involved administrators.

About this task

You employ a CA smart card to create TKE smart cards. You then use TKE smart cards on which you generate the roles for administrating the cryptographic adapters and the keys.

Procedure

1. On the TKE console in Figure 4 on page 6, open the **Smart Card Utility Program**.

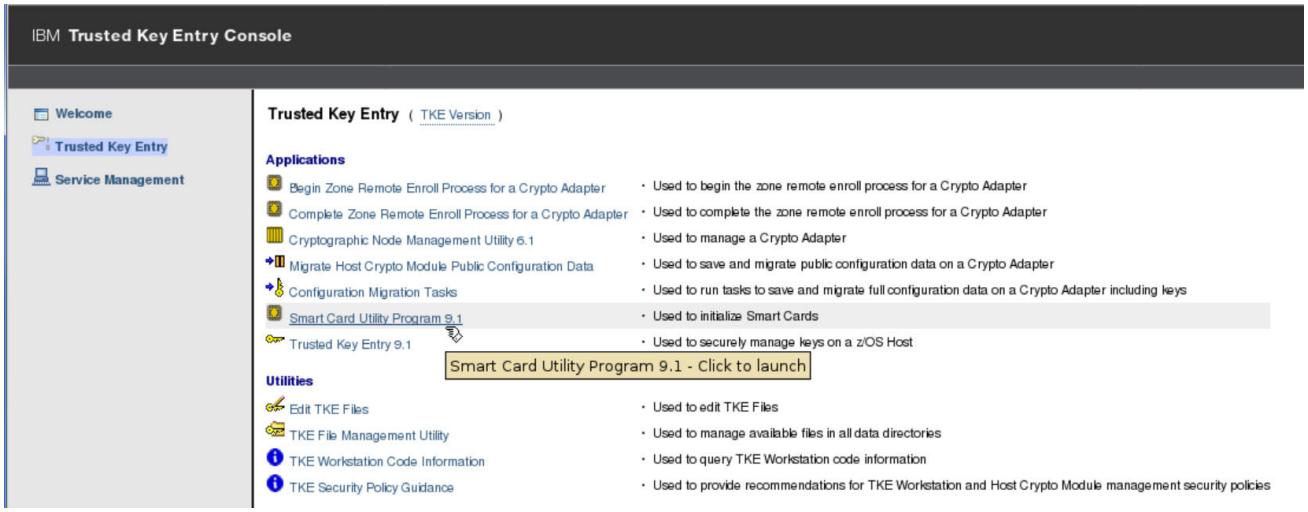

Figure 4. TKE Console - starting the Smart Card Utility Program

2. Log on to the local cryptographic adapter of the TKE workstation.

   You must sign on with a profile that is on the associated cryptographic coprocessor. Therefore, depending on how you have initialized your environment, the **Crypto Adapter Logon** window is displayed with profile IDs that represent a single or group passphrase logon. The individual or group profile you choose must have enough authority to do the functions that need to be performed. In this scenario, use the TKEADM profile to be allowed to initialize a CA smart card and to enroll the TKE workstation. The TKEADM profile is a system-supplied role and profile which is created when the cryptographic adapter on the TKE is initialized. It is intended for a person with the responsibility of initially setting up a TKE, completing migration tasks, or managing the TKE.

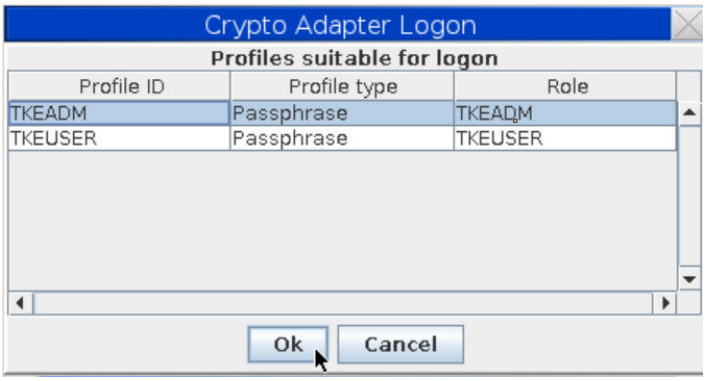

Figure 5. Select the TKEADM profile

   After entering the correct passphrase, the **Smart Card Utility Program** opens. Start the **TKE Smart Card Wizard** from the **File** drop-down menu as shown in Figure 6 on page 7.

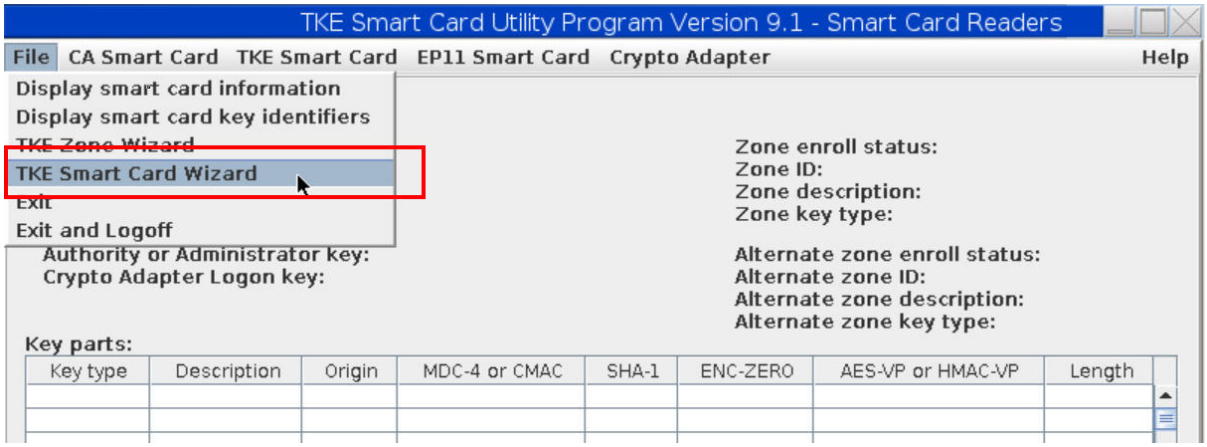

Figure 6. Invoke the TKE Smart Card Wizard

On the welcome panel, read the information, then press **Next**.

3. On the upcoming window, you select the required CA and TKE smart card types as shown in the **TKE Smart Card Wizard** in Figure 7 on page 7. Then press the **Next** button and let the wizard guide you through the creation and initialization of one CA smart card and four TKE smart cards as required for the scenario.

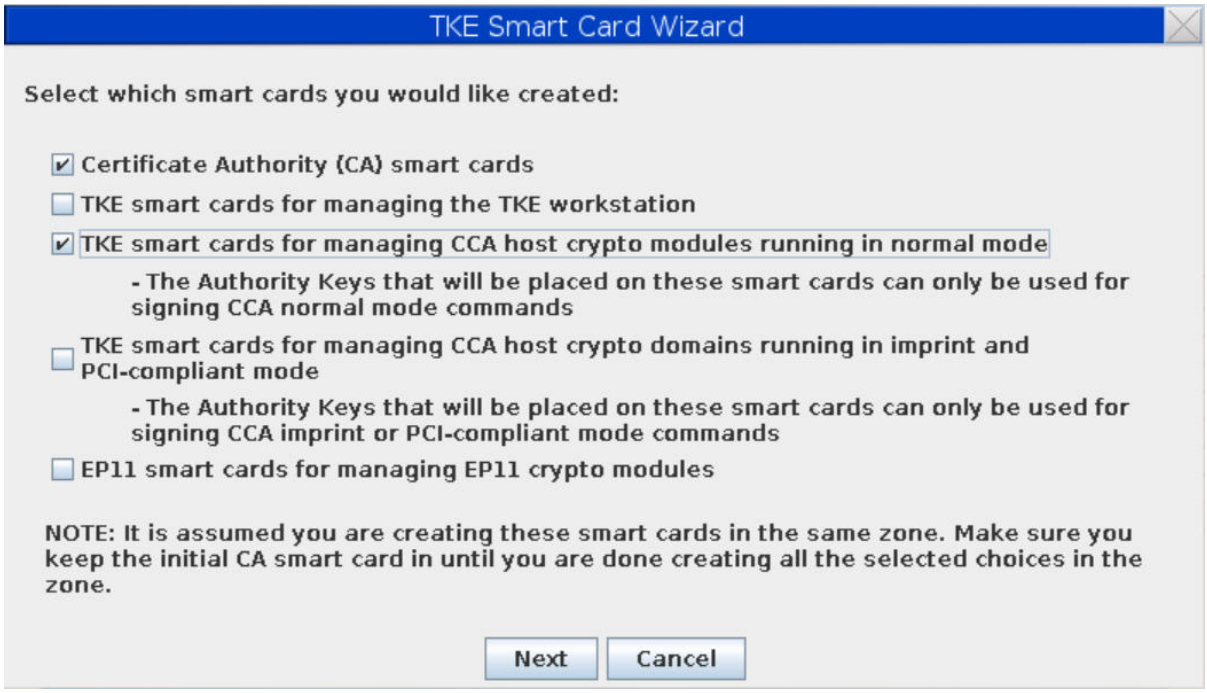

Figure 7. Select smart card types to be created

In an environment where a zone is already defined by a CA smart card, you only need to select the TKE smart cards and you can continue with step “9” on page 8.

4. Press **OK** on the information message that appears.

   When prompted, insert a new smart card to be initialized and personalized as a CA smart card into smart card reader 1. Then press **OK**. An information is displayed that the smart card is being initialized. Press **OK** again.

   If your smart card is not new, you get a warning that all data will be overwritten (not shown here). Press **Cancel**, if you want to keep the card unchanged and insert a new smart card.
5. You are prompted to enter two six-digit PINs. Each PIN must be entered twice.

6. Enter a zone description and a description for the CA smart card.

   For our scenario, enter *testlinux* as the zone description.

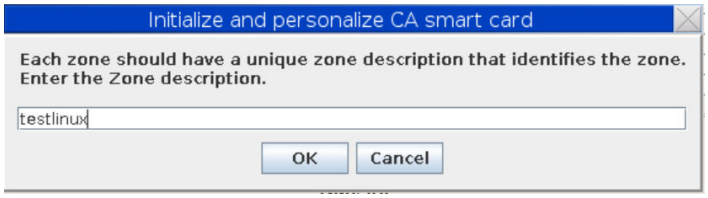

Figure 8. Enter a zone description

   The CA smart card is initialized and personalized.

   Press **OK** to confirm the successful creation. Optionally, you can make a backup of the CA smart card.

7. Press **Yes** on the **Enroll TKE crypto adapter** dialog in Figure 9 on page 8. With this action, you enroll the crypto adapter installed on the TKE workstation, in the *testlinux* zone.

Figure 9. Enrolling the TKE crypto adapter in the testlinux zone

8. Enroll a locally installed cryptographic adapter.

Figure 10. Enroll a locally installed cryptographic adapter

   Pressing **OK** to confirm your selection displays information about the successful enrollment of the cryptographic adapter. The wizard now starts the creation of the TKE smart cards.

9. Create four TKE smart cards, two for module adminivstration, and two for key administration.

Figure 11. Creating TKE smart cards

   After pressing **OK**, the wizard offers you four available TKE smart card types. The first one is for role **MAIss (Module Admin Issue)**:

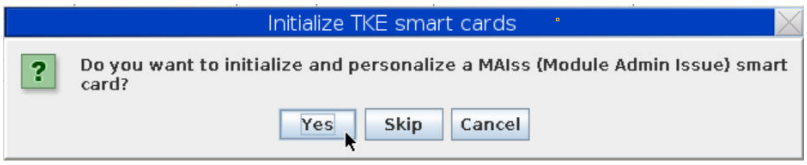

*Figure 12. Creating the **MAIss (Module Admin Issue)** smart card*

Pressing **YES** prompts you to insert the first of the four required smart cards into reader 2 (not shown here). After inserting the smart card into the reader, you are informed about building the TKE smart card for this role.

In a scenario where a CA smart card is already available, and you start with the creation of the TKE smart cards, you are at first prompted to insert this CA smart card into reader 1 and enter the two PINs. Then you are prompted to insert the first TKE smart card into reader 2. Finally, you are informed about the building process.

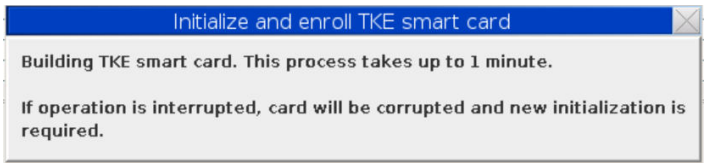

Figure 13. Process information

The first module administrator now must enter a PIN twice to protect the new smart card.

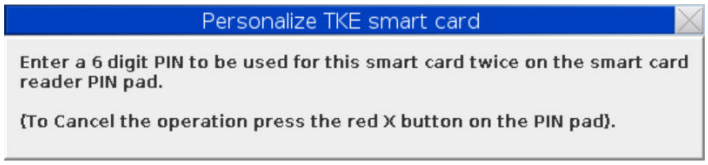

Figure 14. Personalize a TKE smart card

The **TKE Smart Card Wizard** now creates the smart card for the **MAIss** role.

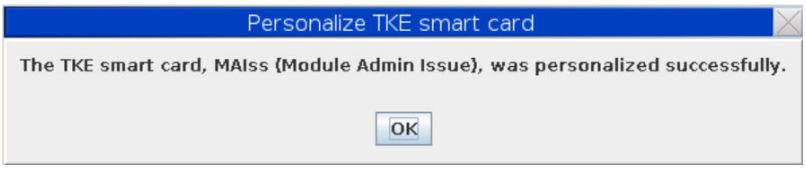

Figure 15. Personalizing the MAIss smart card

Press **OK**. The wizard now asks whether you want to create a smart card for the **MACos** (Module Admin Cosign) role. Answer by pressing **Yes** to let the second module administrator create the second smart card with role **MACos** in the same way.

The onwner of the **MAIss** authority may issue module administration commands, which must be co-signed by the owner of the **MACos** authority.

After creating these two smart card types for module administrating, the wizard asks you for the number of key administrators. For the scenario described here, select **2** to distribute the key parts on two smart cards owned by two key administrators.

Figure 16. Distribute the key parts on two smart cards

10. Initialize two TKE smart cards to be owned by the two key administrators.

    The two roles for the key administrators are called **CCAFst** (CCA First Key Admin) and **CCAMl1** (CCA Middle/Last Key Admin 1).

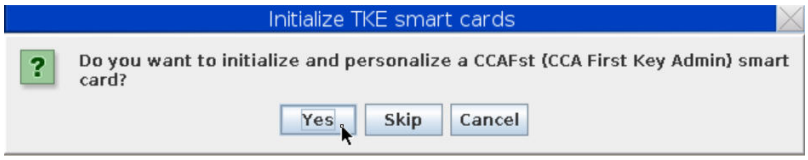

Figure 17. Initialize CCAFst smart card

The process of creating the **CCAFst** and **CCAMl1** smart cards is the same as described for **MAIss** in step "9" on page 8.

After a successful personalization of all four smart cards, the wizard offers you to create another set of TKE smart cards, for example for backup. Select **No** in our scenario.

Results

After completion, you get information that the TKE smart cards have been successfully created. These cards are now ready to accept role and authority information and a signature key.

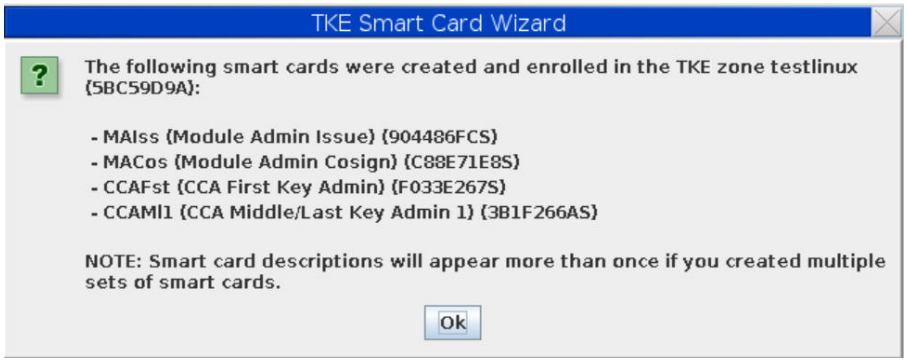

Figure 18. Overview of enrolled smart cards

You can exit the **Smart Card Utility Program** (SCUP) now.

Creating and connecting a host

You need to define and connect the Linux on Z host system to the Trusted Key Entry program (TKE). Then you can create master keys on any attached cryptographic adapters and use them for cryptographic operations.

Before you begin

Defining the host is a one time task. Check the **Host ID** list of the dialog shown in Figure 20 on page 12, whether the host is already available. In our scenario, this list is empty, which may not be the case in a real environment. If your host is already defined to the TKE, you can continue with step “5” on page 12. Else, start at the beginning of the procedure.

Ensure the following:

- The host to which you want to connect the TKE must be an up and running Linux on Z instance with an attached CCA cryptographic coprocessor.
- The `catcher.exe` TKE daemon, listening to port 50003, is started on this host.

Procedure

1. Invoke the **Trusted Key Entry** program from the **Trusted Key Entry Console** in Figure 3 on page 4.
   This initial window provides access to applications and utilities available on the TKE workstation.
2. In the list of applications, click on **Trusted Key Entry**.
   Similar as described in step “2” on page 6 of “Initialize a CA smart card and TKE smart cards” on page 5, whenever you launch a TKE application or utility, you must sign on with a profile that has enough authority to do the functions that are performed by the selected application or utility. The steps described here use the system-provided default TKEUSER user profile.

Figure 19. Crypto Adapter Logon

3. Select the TKEUSER profile (or an existing profile of your choice) and click **OK**.
   In the subsequent **Passphrase Logon** dialog, log-on with the passphrase associated with the TKEUSER profile. A window is displayed (Figure 20 on page 12) that normally shows a list of the host systems defined to the Trusted Key Entry workstation.

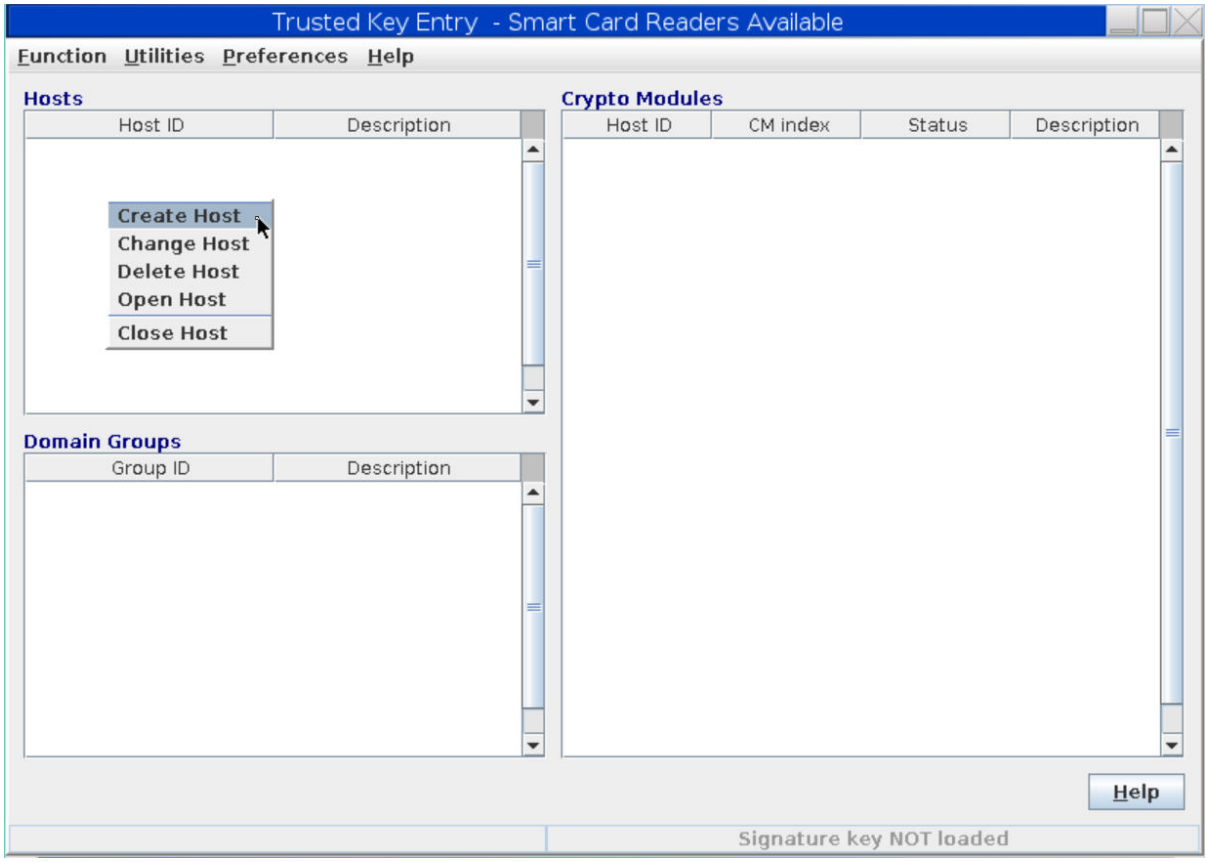

Figure 20. Host systems defined to the Trusted Key Entry

4. To define a new host to the TKE, open the context menu for hosts (Figure 20 on page 12), and select the **Create Host** action. The **Create New Host** dialog opens.

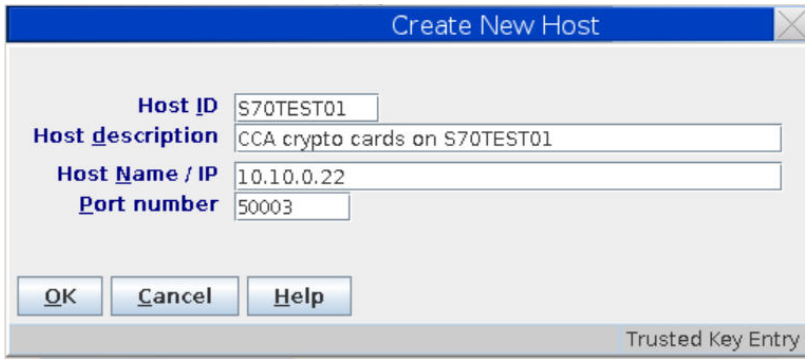

Figure 21. Create a new host

Enter the values of the host to which your cryptographic coprocessor is connected. Input for all entry fields is required, except for an optional host description. It is assumed that this host is a Linux instance on an IBM Z system running the `catcher.exe` TKE daemon, listening to port 50003. After pressing **OK**, the new host is visible within the list of **Host ID**s (Figure 22 on page 13) and the TKE is connected with all detected cryptographic coprocessors attached to this host.

5. Open the new or applicable host and select the attached cryptographic coprocessor on which you want to set the master key on one of its domains.

   From the host's context menu select action **Open Host**.

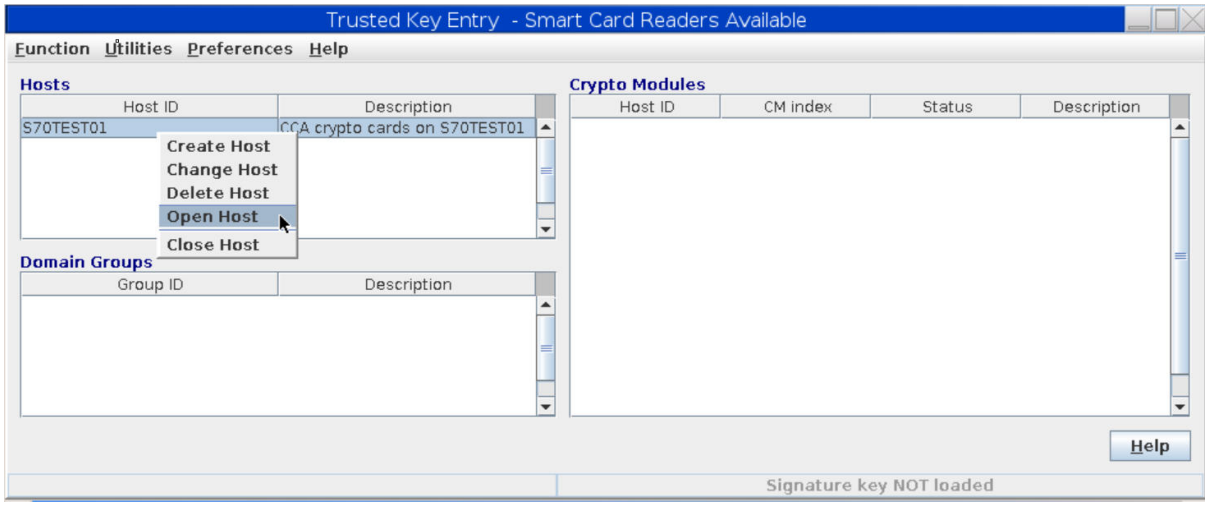

Figure 22. Open the host

a) Log on to the selected host with the appropriate credentials.

Figure 23. Log on to the host

b) If applicable, the TKE now requests a verification of any new detected cryptographic coprocessor. Check the displayed information.

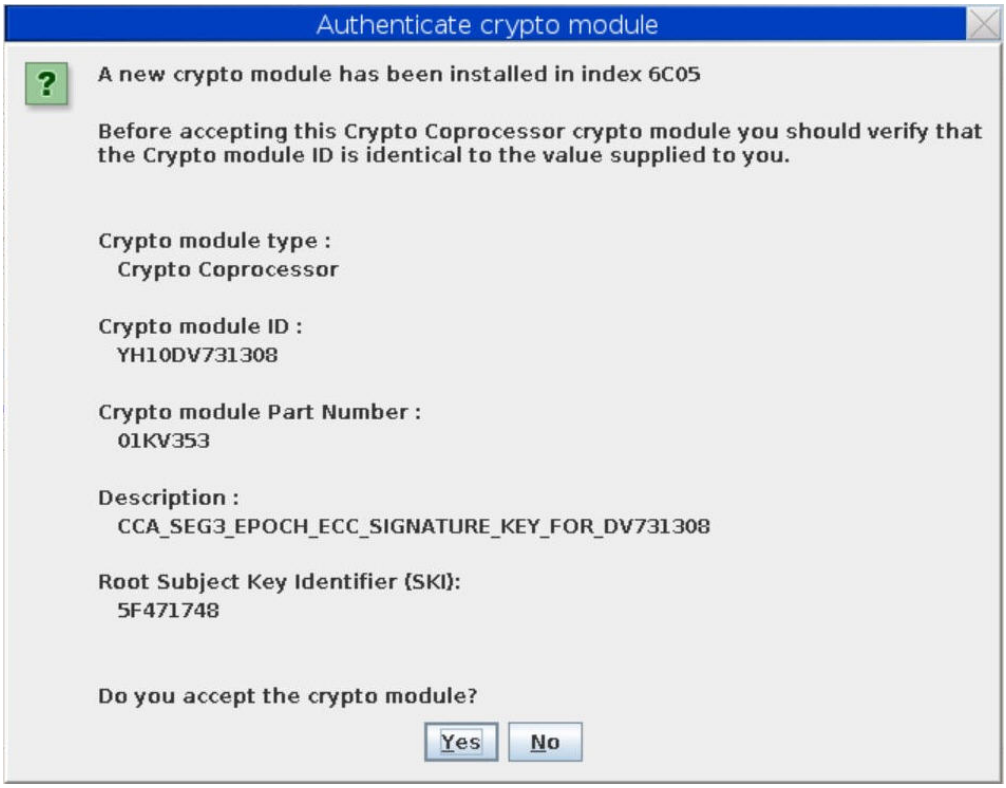

Figure 24. Authenticate crypto module

Press the **Yes** button to continue if you see information about the expected coprocessor.

The **Crypto Modules** view of Figure 25 on page 14 now displays the available adapters. In our scenario, three adapters in CCA coprocessor mode are available on the current host, one CEX6C module and two CEX5C modules.

c) Open the cryptographic coprocessor.

Select a cryptographic coprocessor of your choice and trigger action **Open Crypto Module** from its context menu.

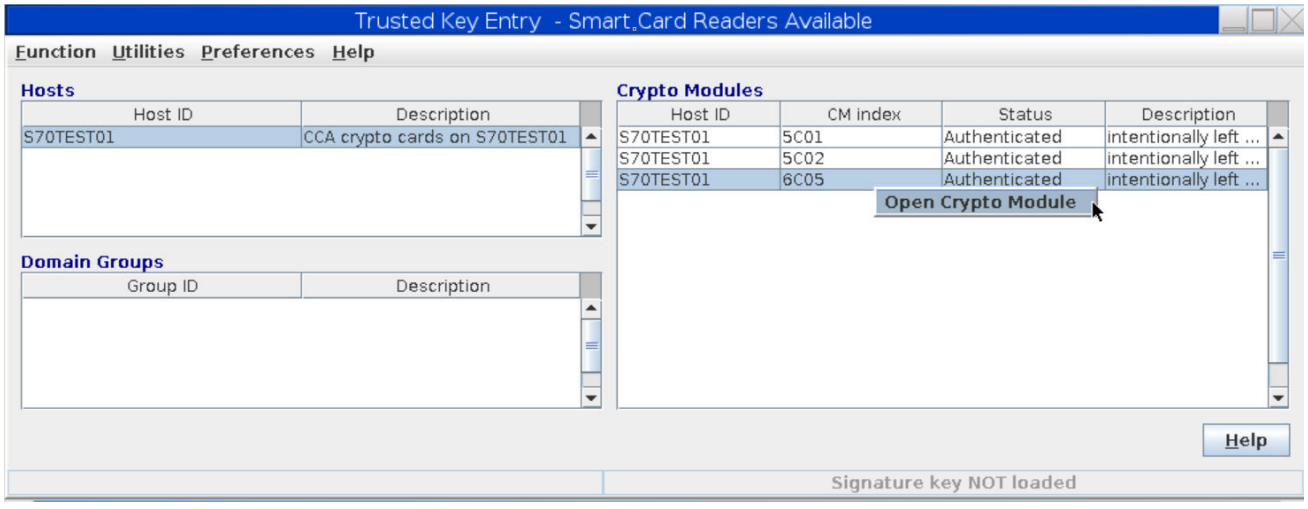

Figure 25. List of crypto modules

The **Crypto Module Administration** dialog for the selected cryptographic adapter opens (see Figure 26 on page 16).

Results
The attached cryptographic adapter is now ready for communication with the Trusted Key Entry workstation.

What to do next
Depending on your current environment, you now need to define the required roles and authorities for module administration and for loading and setting master keys.

Creating the required roles and authorities

Use the **Setup Module Policy** wizard to create the proposed suitable roles and authorities.

Before you begin
You need the four initialized TKE smart cards produced as described in “Initialize a CA smart card and TKE smart cards” on page 5. These cards do not yet contain any signature keys. The generation of these signature keys is part of the described procedure.

About this task

An authority is identified to the host cryptographic coprocessor or domain by the *authority index*. Each authority has an associated role that defines which actions the authority can perform, that is, which signed commands the authority can issue or co-sign.

Each authority is furthermore created together which an authority signature key. This signature key is actually a key pair. An authority sends a command (a request to perform an action) to a host cryptographic coprocessor or domain. It signs its commands using the private key of its signature key pair. The host cryptographic coprocessor or domain verifies the signature by using the public key of the same key pair. If the verification is successful, and the requested action is allowed by the associated role, the command can be performed.

This procedure is described for role **CCAFst**. You need to perform the same steps for roles **CCAMl1**, **MAIss**, and **MACos** accordingly.

Procedure

1. In the **Crypto Module Administration** window, press the **Setup Module Policy** button (see Figure 26 on page 16).

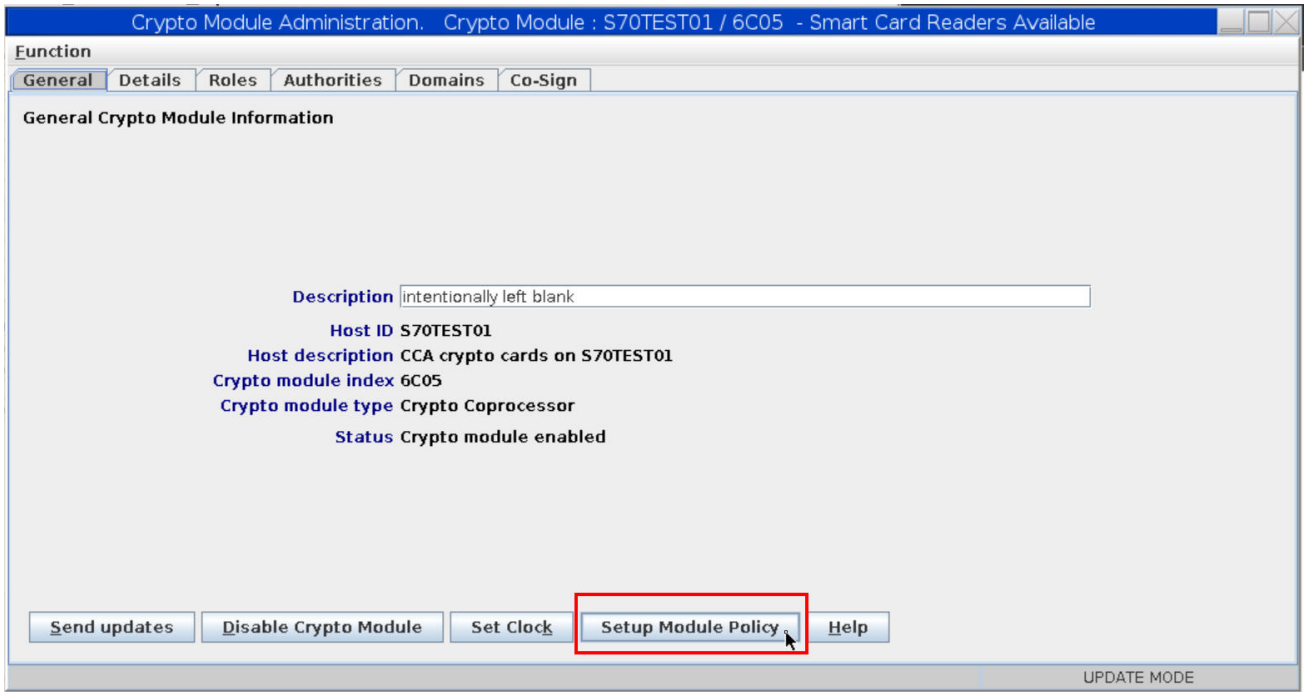

Figure 26. Crypto Module Administration

There are two other paths to invoke the **Setup Module Policy** wizard. From the **Crypto Module Adminstration** window shown in Figure 26 on page 16, you can either select the **Roles** or the **Authorities** tab. In both cases, you then open the context menu from the window's white space and select option **Setup Module Policy**.

The **Setup Module Policy** welcome window opens.

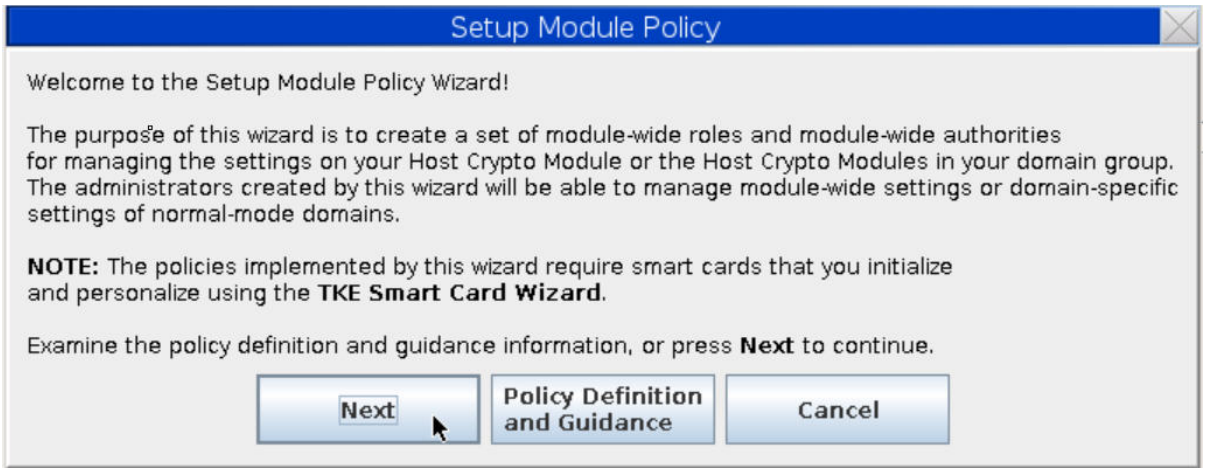

Figure 27. Setup Module Policy welcome window

Pressing **Next** invokes an information message that a set of module-specific roles will be created.

Press **OK** to display Figure 28 on page 17.

2. Select the source of the signature key for the roles.

   For all crypto module types, a default authority with index 0 is created on the crypto module when it is manufactured or reinitialized. Starting with the CEX6C, a default authority with index 99 is additionally created. The default authority for index 99 uses a 512-bit Brainpool ECC key. Select **Default key for index 99** for a CEX6C. For prior versions of the coprocessor choose **Default key for index 0**.

Figure 28. Source of the signature key

Then, in Figure 29 on page 17, select the authority index which is associated with INITADM's signature key and the INITADM role. In the scenario, the authority index 99 is associated with the default signature key for index 99.

Figure 29. Specify authority index for the signature key

Press the **Continue** button in Figure 29 on page 17. On the succeeding confirmation message, press **OK** to display Figure 30 on page 18.

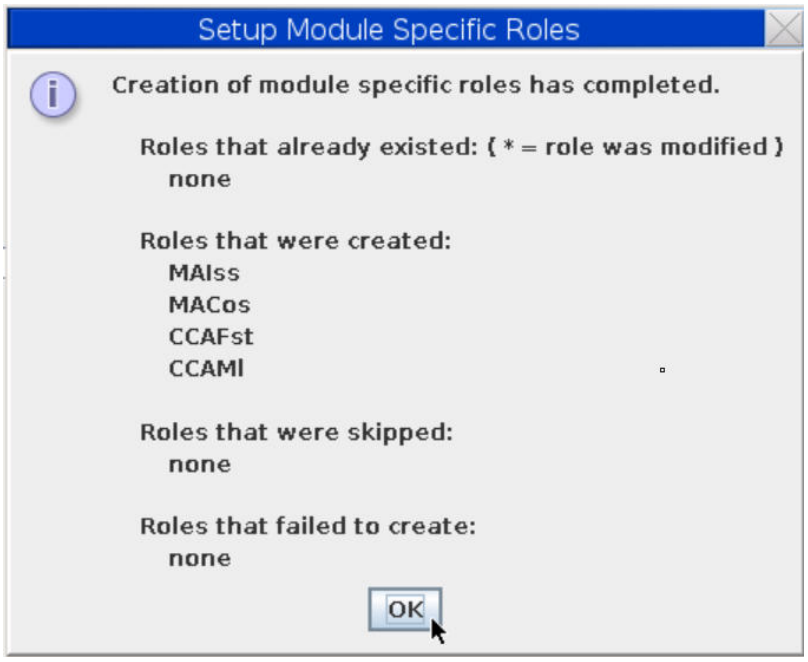

Figure 30. Roles created

3. Let the wizard continue to create the required authorities.
   Press the **Next** button as shown in Figure 31 on page 18.

Figure 31. Starting to create authorities

Insert the TKE smart card that you created with role **CCAFst** into smart card reader 2.

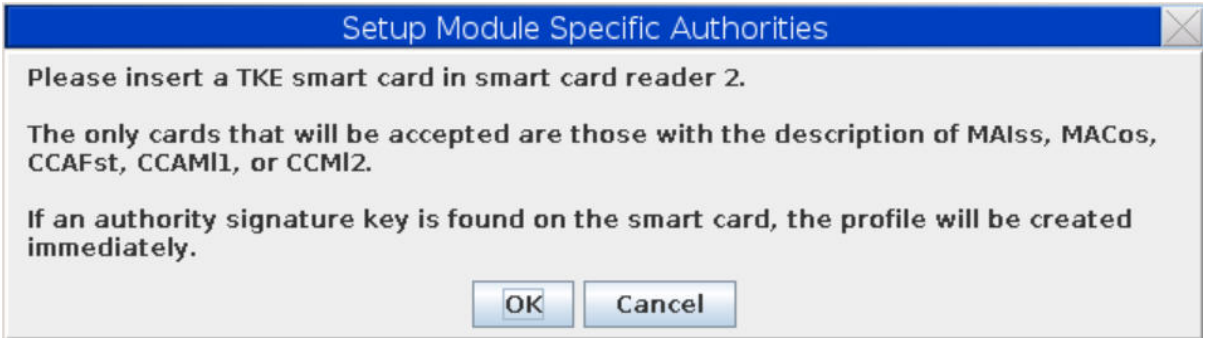

Figure 32. Creating authorities

Press **OK**. The wizard confirms that the smart card is new without a signature key on it.

Figure 33. Generating an authority signature key

Confirm with **OK**. On the **Select Key Type** dialog (Figure 34 on page 19) specify your desired key type for the signature key to be generated. For the scenario, a **320-bit EC key** is used.

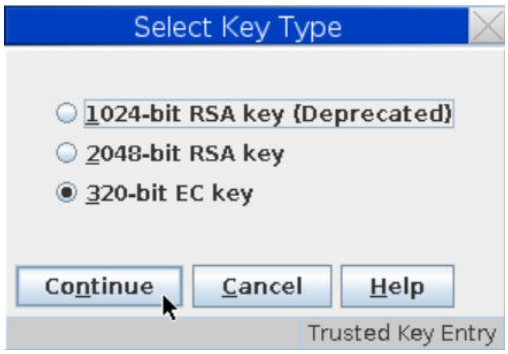

Figure 34. Select signature key type

When prompted, enter the PIN on the smart card. Then the wizard notifies you about the successful creation of the signature key.

Figure 35. Signature key successfully stored

4. Press **OK** on the notification about the generated signature key.

   You get a message about the successful creation of the authority. The wizard creates the following authorities:

   Authority 10 with role **MAIss**
   Authority 11 with role **MACos**
   Authority 20 with role **CCAFst**
   Authority 21 with role **CCAMl**

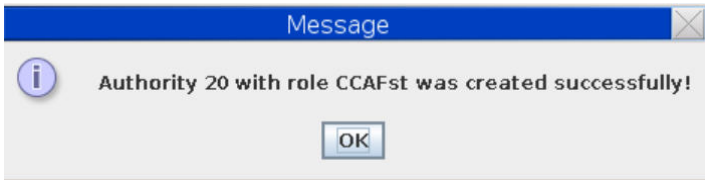

Figure 36. Authority successfully created

5. Press **OK** on the message and continue to create the next required authority, until you finished to handle all four role/authority combinations (profiles).

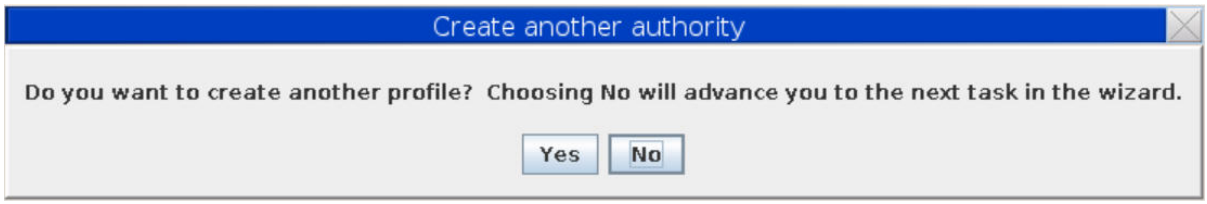

Figure 37. Create another authority

Press **Yes** and then insert the next TKE smart card in reader 2 to create the authority for the next role. Again, select type **320-bit EC key**. After a successful creation of all authorities by all involved administrators, a completion notice is shown (Figure 38 on page 20).

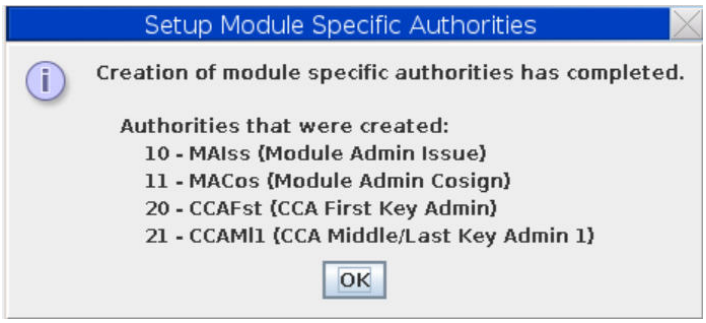

Figure 38. Authority and role completion notice

Press **OK**.

6. Limit the INITADM authority.

   Now that you created all required profiles for your security environment, you must remove all authority from the default INITADM role.

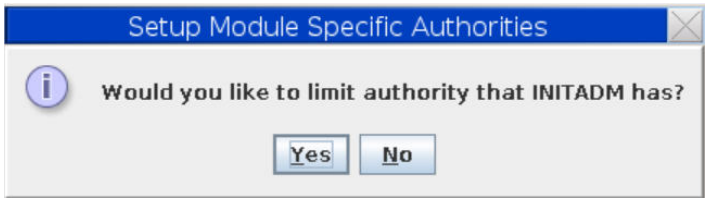

Figure 39. Limit the INITADM authority

   Pressing **Yes** displays a message about the successful command execution. Press **OK** on subsequent information dialogs until the wizard returns to the **Crypto Module Administration** dialog.

7. If required, change the roles **CCAFst** and **CCAMl** to include the privilege for activating the new master key. This is shown for **CCAFst**. This step is not applicable for roles **MAIss** and **MACos**.

   In Figure 40 on page 21, select **Change Role** from the context menu of **CCAFst** and check **Set AES master key** as shown in Figure 41 on page 21.

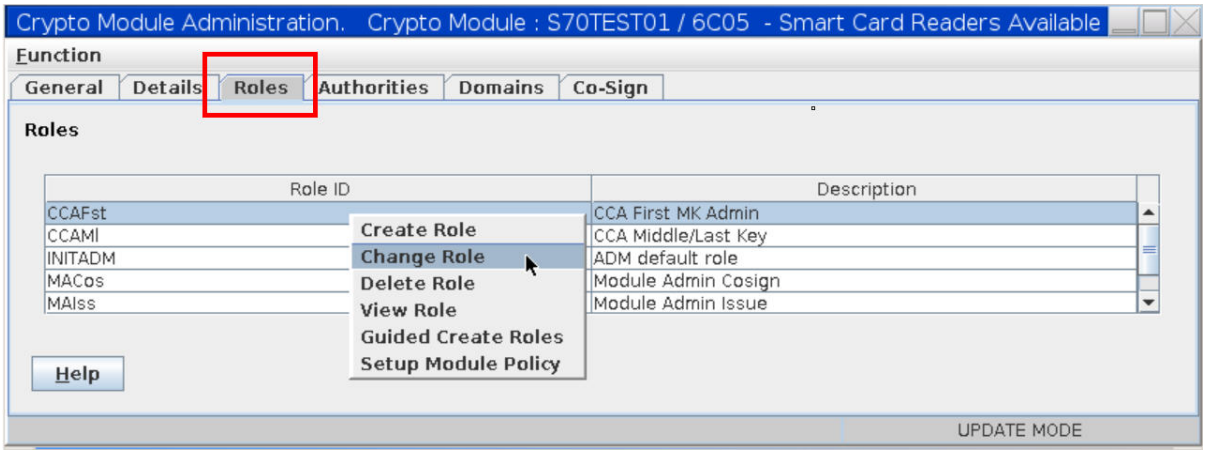

Figure 40. Changing the roles

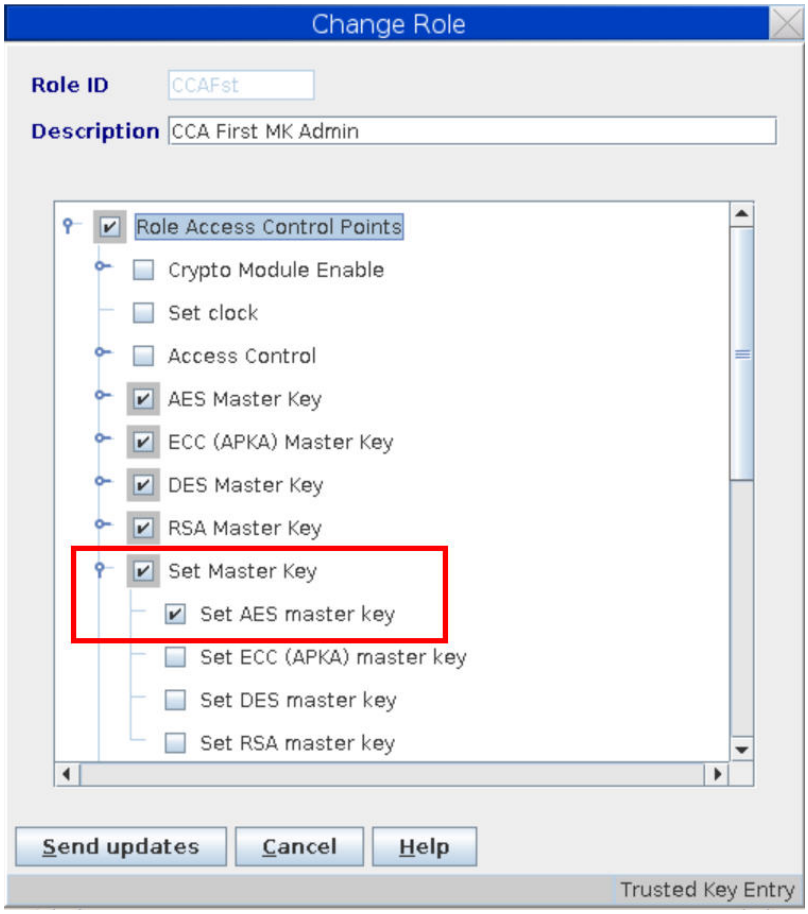

*Figure 41. Including **Set AES master key** for role **CCAFst***

This action requires a signature key and sufficient authority. Use the authority of the **MAIss** role for issuing the role change and use the **MACos** role for co-signing the command. The wizard guides you through this procedure with adequate prompts.

8. Verify your created roles by selecting the appropriate tab as shown in Figure 42 on page 22.

   To verify the privileges allowed for role **CCAFst**, select this role from Figure 42 on page 22, and invoke **View Role** from its context menu to see the result shown in Figure 43 on page 22.

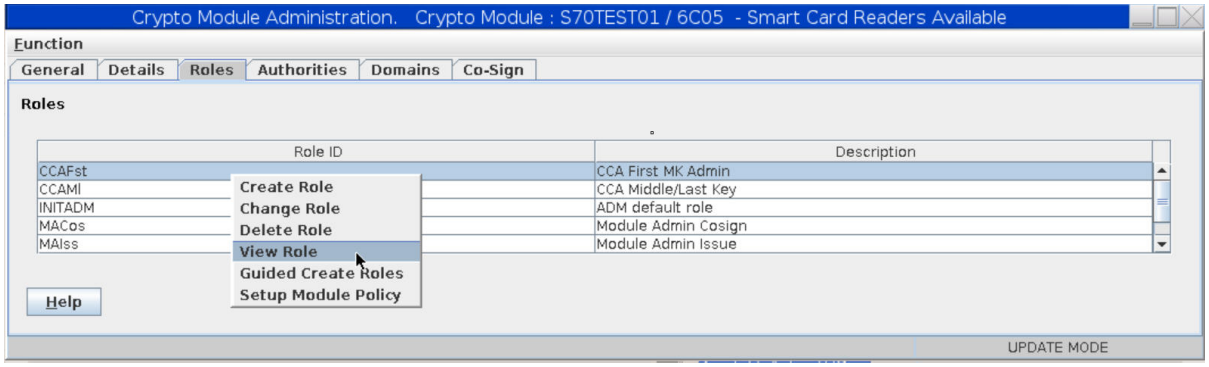

Figure 42. View roles

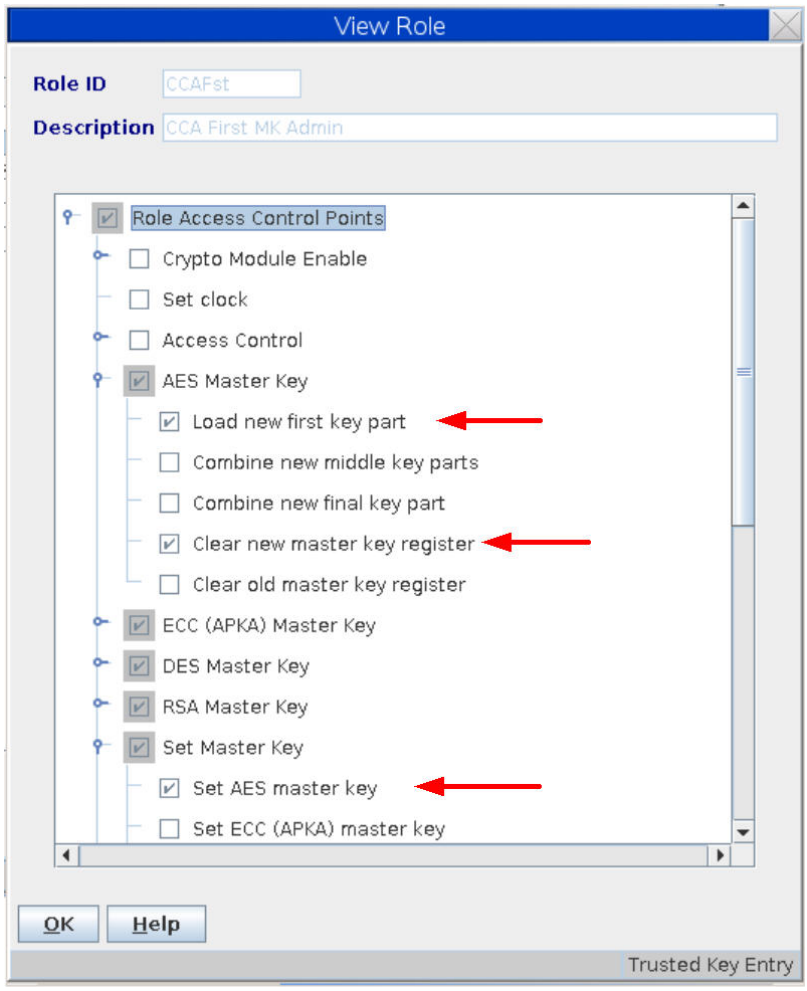

Figure 43. View privileges for role CCAFst

For the **CCAMl** role, you can view the privileges in Figure 44 on page 23.

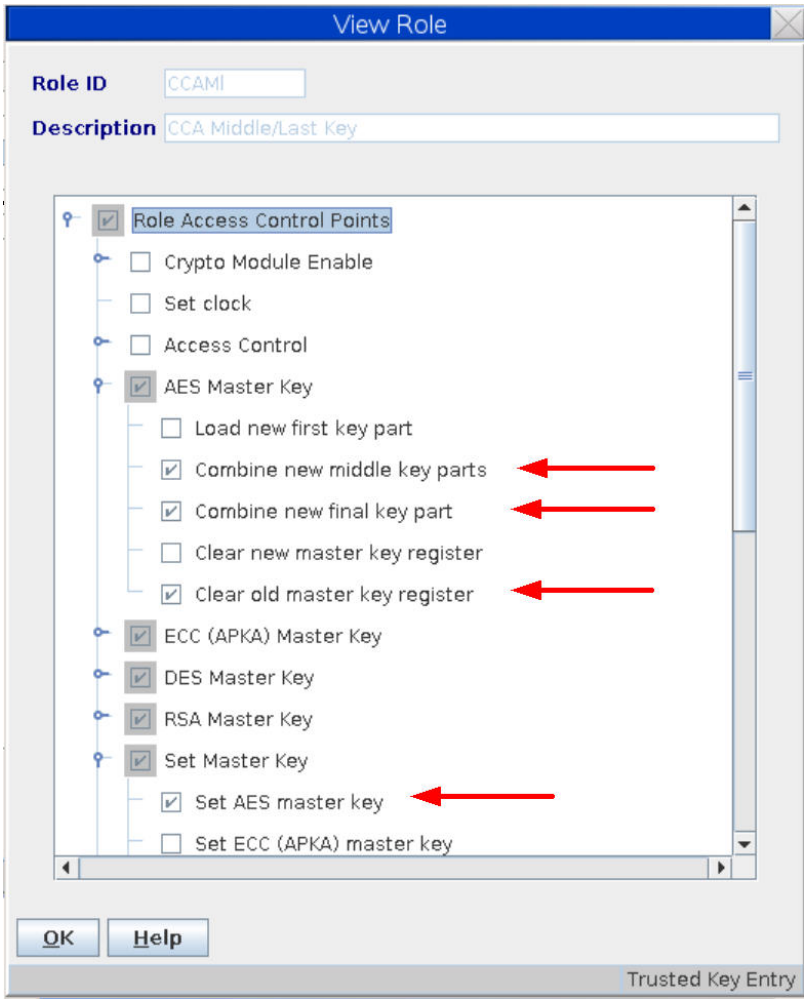

Figure 44. View privileges for role CCAMl

Results

After you completed this procedure for all required roles (**CCAFst**, **CCAMl**, **MAIss**, and **MACos**), your security environment is readily set up. The key administrators owning the pertaining smart cards can now commonly start to generate and load the key parts and activate the new master key to be used as the current master key as described in “Activating the master key” on page 33.

Chapter 3. Setting an AES master key on a CCA coprocessor

After completing the setup of the smart card environment, you can create an AES master key on an attached cryptographic coprocessor running in CCA coprocessor mode.

In contrast to the one-time tasks described in Chapter 2, “Establishing the security environment,” on page 5, a master key change may occur multiple times, depending on your security policies. Therefore, all sub-tasks of this topic must be performed each time a master key change is necessary.

Let two key administrators generate two parts of the master key on two separate smart cards (**CCAFst** and **CCAMl1** smart cards). Both users are authorized to create master key parts on a smart card and to load them onto the host coprocessor. The administrator with role **CCAMl1** is additionally authorized to combine the loaded key parts to create the final master key on the cryptographic adapter. Both users are allowed to activate a loaded master key.

Generating key parts

In the scenario, this task must be performed sequentially by both key administrators using their different smart cards.

About this task

This task must be performed by the involved key administrators holding the pertaining smart cards where to independently store the key parts.

Procedure

1. In the Trusted Key Entry application, open the host and the desired cryptographic coprocessor (actions **Open Host** and **Open Crypto Module**) to reach the **Crypto Module Administration** dialog. Then select the domain where to set the AES master key.

   Click on the **Domains** tab and then click on your domain on the right side (Figure 45 on page 25). In this document, the domain with index 26 is used on the attached coprocessor. Domains are assigned during LPAR definition on the support element (SE).

   When clicking on index 26, the window for domain 26 opens as shown in Figure 46 on page 26.

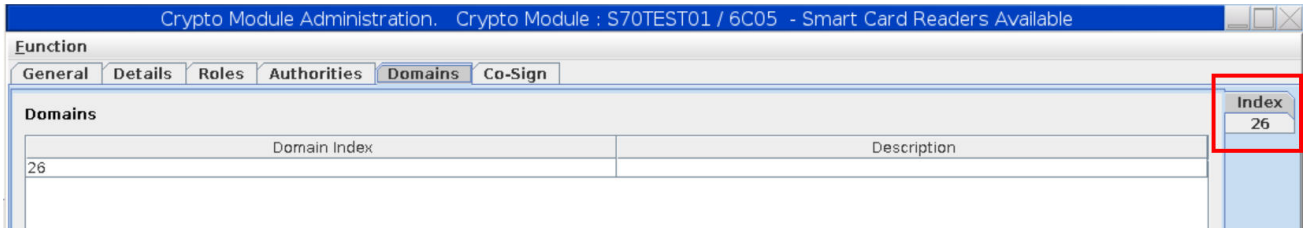

Figure 45. Domain selection for setting the master key

2. Press the **Keys** button at the lower edge of the dialog from Figure 46 on page 26.

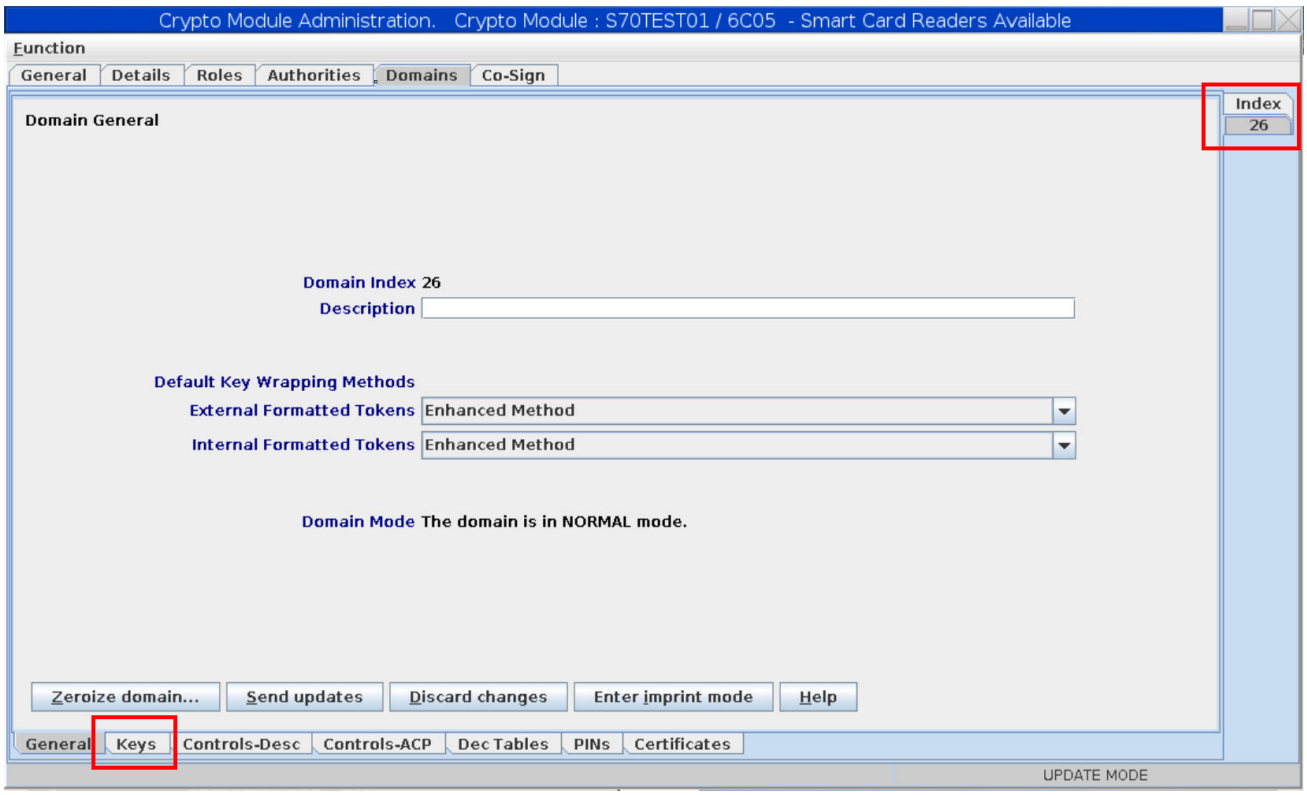

Figure 46. Setting master key on selected domain

The window shown in Figure 47 on page 27 opens. In this scenario, you generate two key parts for an AES master key:

- Key part 1 is generated by the key administrator who owns the TKE smart card for role **CCAFst** with authority 20.
- Key part 2 is generated by the key administrator who owns the TKE smart card for role **CCAMl1** with authority 21.

3. Select the key type **AES Master Key** from Figure 47 on page 27, and from the context menu, select **Generate multiple key parts to... -> Smart card**.

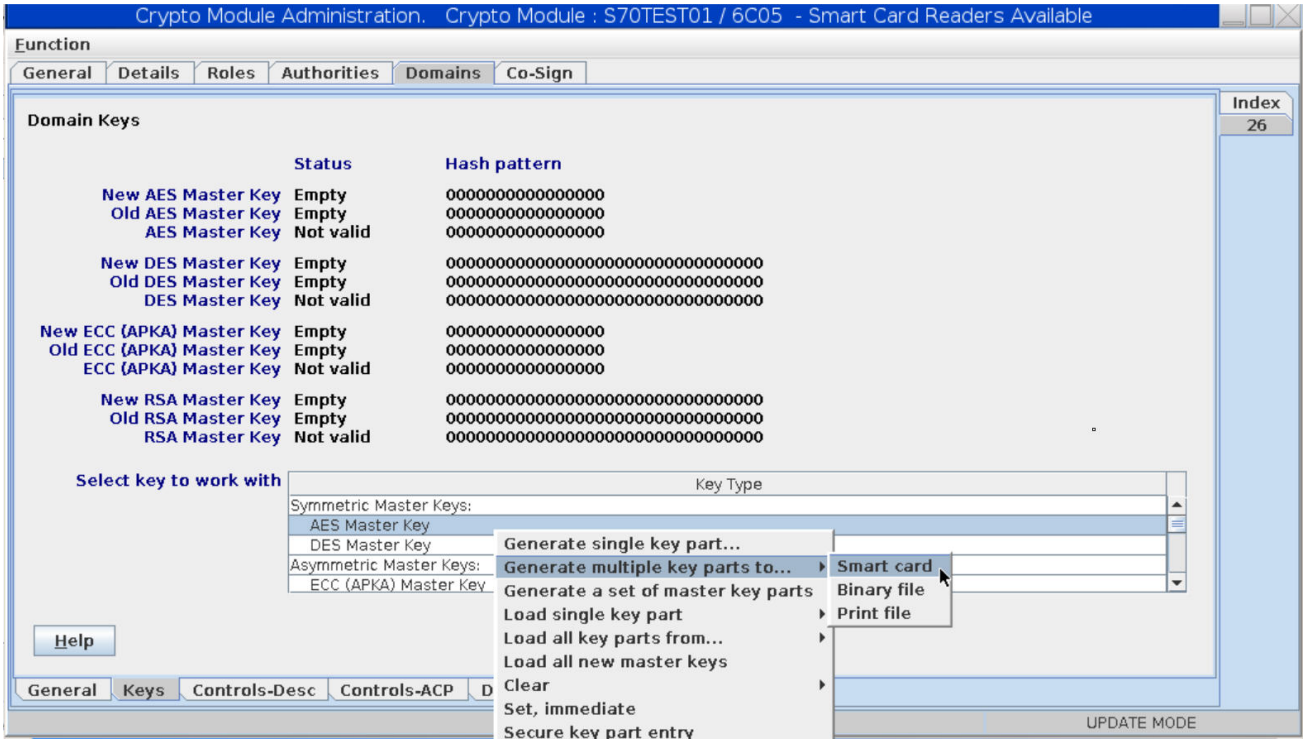

Figure 47. Generate multiple key parts on a smart card

4. You are now guided through the process by a series of prompts.

   a) Enter the number of key parts to be generated for the final master key.

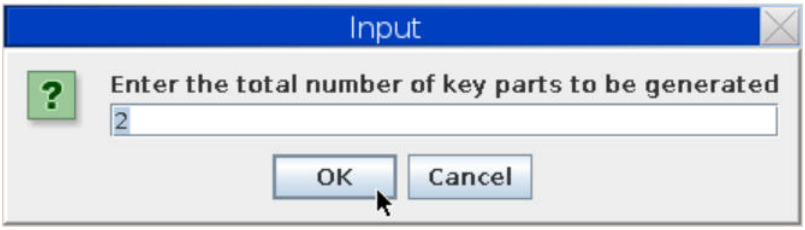

Figure 48. Enter the total number of key parts to be generated

   b) Press **OK** on the **Generate Key Part** message dialog.

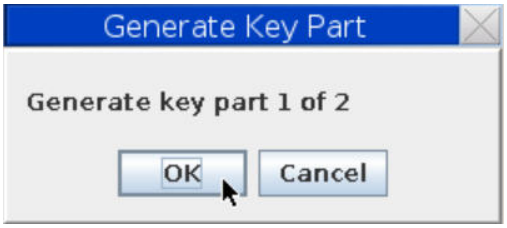

Figure 49. Generate first key part

   c) Create the first master key part and place it on the smart card that has the authority signature key for authority 20 (for role **CCAFst**).

      Insert this smart card into reader 1.

   d) You are now asked if you want to use the same smart card reader for the whole process. Select **No** in Figure 50 on page 28, because in the scenario, you use two different smart card readers.

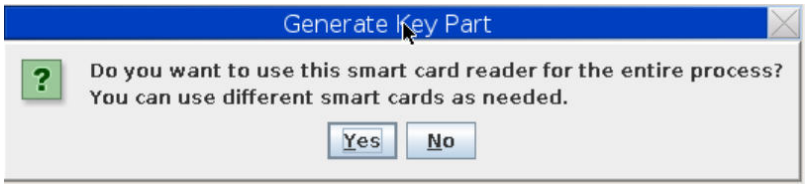

Figure 50. Use different same smart card readers for the entire process

e) Then the wizard prompts you to insert the smart card for key part 1 into reader 1 and to enter the smart card PIN.

f) Enter a key part description.

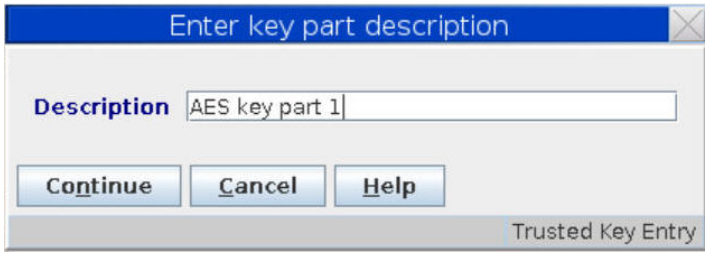

Figure 51. Enter key part description

Pressing **Continue** displays a confirmation about the successful creation of the key part.

g) Now the second key administrator is guided through the same dialogs to create key part 2 on his smart card.

Note: Ensure to use smart card reader 2 for the second and third key part.

After pressing **OK** when saving the second key part, the program returns to the **Crypto Module Administration** dialog.

Results

The smart cards now contain the signature key and key parts needed to perform a key load operation. You can verify the generated key parts on the smart cards using the **Smart Card Utility Program** (**File -> Display smart card information**) .

Loading key parts

After the generation of the key parts on the smart cards you load them onto the cryptographic coprocessor.

About this task

Both key administrators must load their key part. The process for both differs slightly, so both loading actions are described in this procedure.

Procedure

1. On the **Crypto Module Administration** dialog, in the **Key Type** list select the **AES Master Key** entry. Right click to open its context menu and select **Load all key parts from ... Smart card** as shown in Figure 52 on page 29.

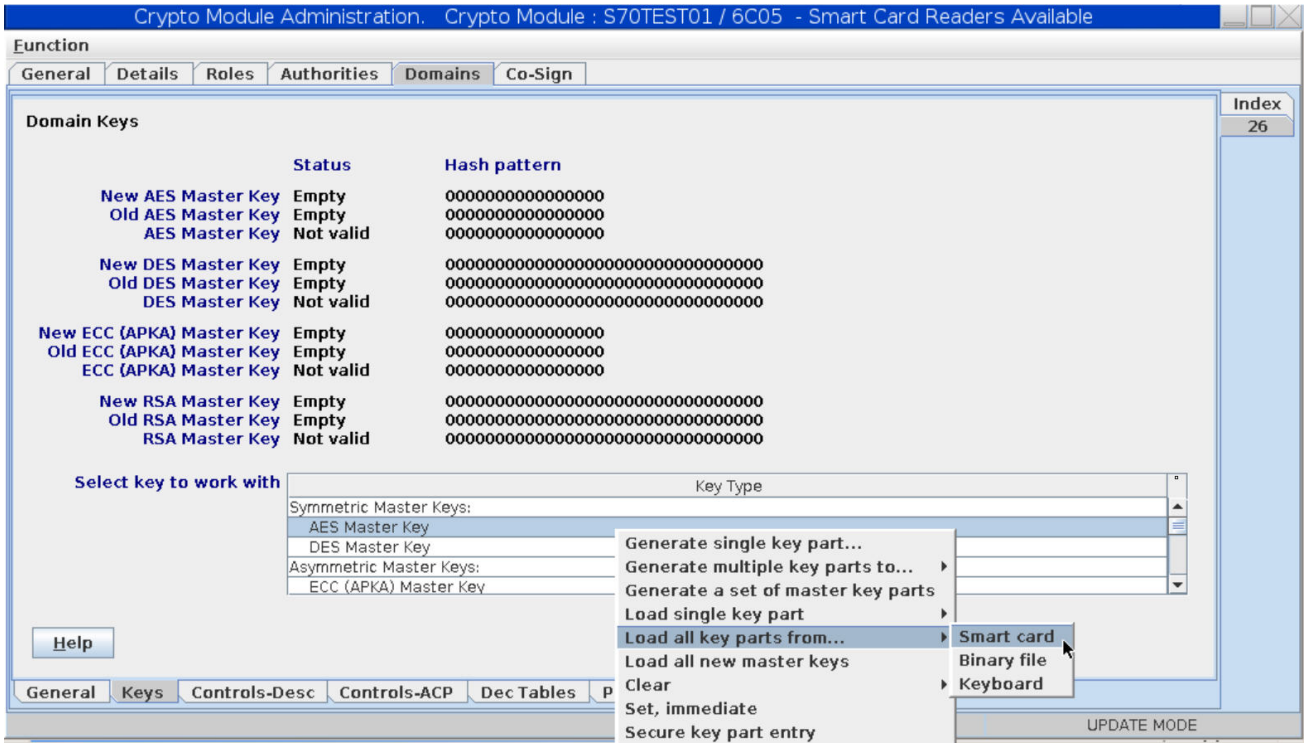

Figure 52. Load all key parts from smart card

a) When prompted, enter the number of key parts (2) to be loaded.

b) When the **New AES Master Key** register is not empty, answer **Yes** to the **Clear Key Register** question (Figure 53 on page 29), because the currently loaded new master key will now be stored in this register.

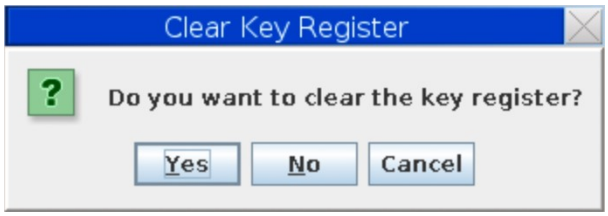

Figure 53. Clear the key register

To clear the **New AES Master Key** register, you need a signature key and a certain level of authority. Authority 20 with the role **CCAFst** is allowed to perform this action. So, when prompted, select smart card reader 1 as the source of the signature key and insert the **CCAFst** smart card.

In the message about the successful clearing of the key register, press the **OK** button.

c) Select the first key part by pressing **OK** in the **Load Master Key** dialog as shown in Figure 54 on page 29.

Figure 54. Select first key part to be loaded

d) In the upcoming prompt, selecting smart card reader 1 as the source for key part 1 and press the **Continue** button.

e) In Figure 55 on page 30, select the shown AES key part 1 and press **OK**.

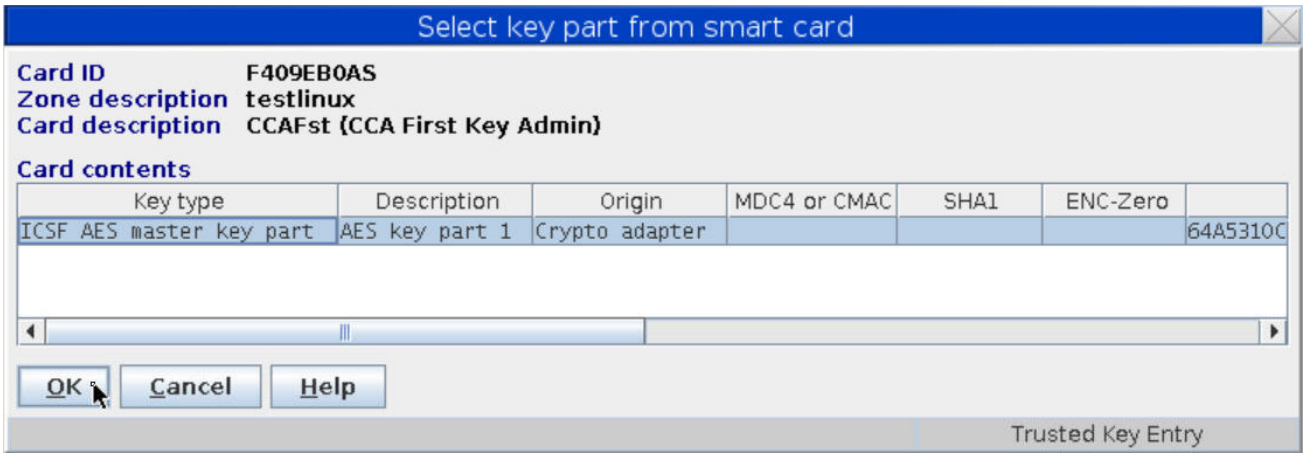

Figure 55. Load first key part

Press **OK** to proceed to the **Key part information** dialog (Figure 56 on page 30).

f) Now press the **Load key** button.

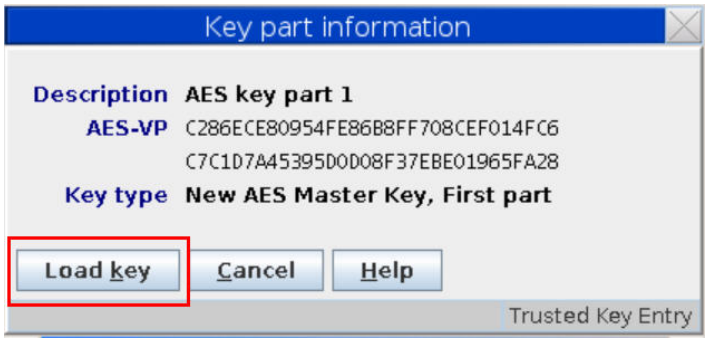

Figure 56. Key part information

Press **OK** in the upcoming confirmation message (`Command was executed successfully`) to proceed.

2. Select the last (second) key part.

Figure 57. Select last (second) key part

Press **OK** to load the second key part.

If the **CCAFst** administrator now tries to continue, an error message is displayed, because the key administrator with authority 20 may only load the first key part. Now the second key administrator needs to continue to process this step.

Figure 58. Error ...

Press **Retry** to use the smart card with key part 2 owned by the key administrator with authority 21. When prompted, select smart card reader 2 as the source of the required signature key and press **Continue**. Then insert the TKE smart card 2 into reader 2 and enter the password for this smart card. As expected, the wizard offers to use authority index 21 in Figure 59 on page 31.

Figure 59. Specify authority index

Press the **Continue** button and then select smart card reader 2 as the source of the missing master key part to be loaded.

3. Now you can load key part 2, the last key part in our scenario.

   You see a view showing information about the AES key part 2 (see Figure 60 on page 32) that you have been previously generated on the inserted smart card.

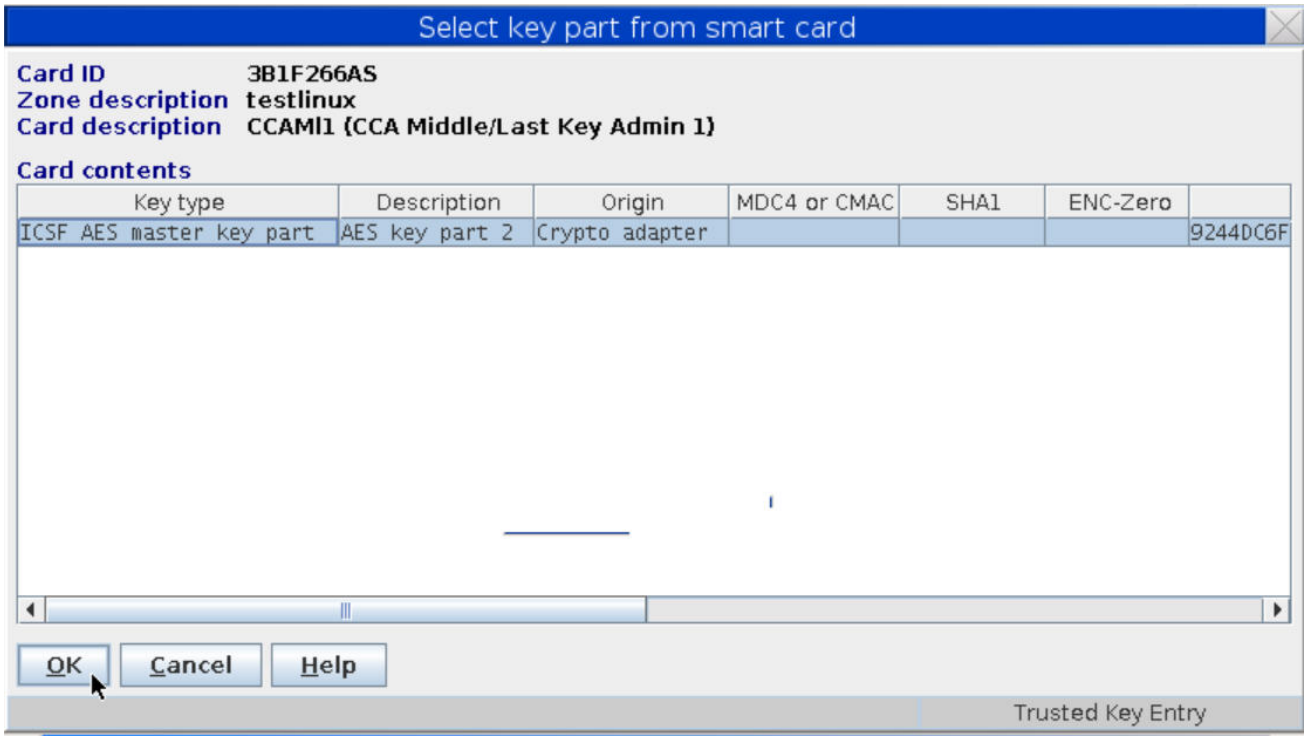

Figure 60. Load key part 2

Now proceed with key part 2 as previously described for key part 1 in Figure 55 on page 30. Press **OK** until you return to Figure 61 on page 32 where you can see that the **New AES Master Key** register is now **Full**.

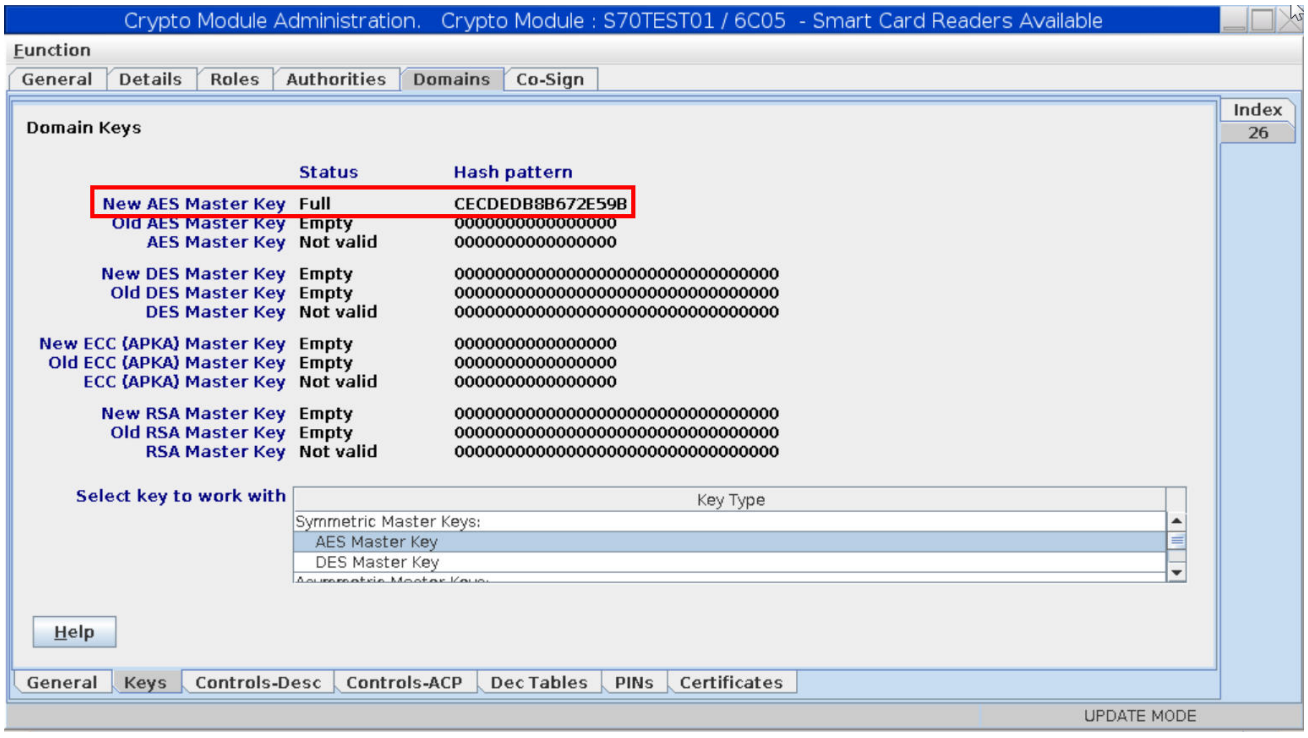

Figure 61. Key parts loaded into New AES Master Key register

Results
Now both key administrators finished loading the complete master key into the **New AES Master Key** register on the cryptographic adapter.

Activating the master key

Finish the AES master key creation by setting the new master key as the current master key. In our scenario, both key administrators are authorized to perform this task.

Before you begin
Most probably, there will be data encrypted by a secure key wrapped by the currently valid master key. Therefore, you should activate a new master key only after ensuring that no data loss will occur. If you maintain a secure key repository, perform this procedure only if the keys in this repository are already re-enciphered under the key you want to make current. For Linux, you find information about re-enciphering utilities for a key repository in *Pervasive Encryption for Data Volumes* available at:

```
https://www.ibm.com/support/knowledgecenter/linuxonibm/com.ibm.linux.z.lxdc/lxdc_linuxonz.html
```

About this task
For each domain on a cryptographic coprocessor, the TKE maintains three master key registers (see Figure 62 on page 34):

New AES Master Key
This register contains the new master key to be set. While the new master key remains in this register, it cannot be used to generate new secure keys until it is activated as described in this topic. Activation transfers the new master key into the **AES Master Key** register and clears the New AES Master Key register.

Old AES Master Key
This register contains the previously used master key. Secure keys enciphered with the master key contained in the OLD register can still be used until the master key is changed again.

AES Master Key
This register contains the currently valid master key.

Procedure

1. Select **Set, immediate** from the AES Master key pull-down choice as shown in Figure 62 on page 34.

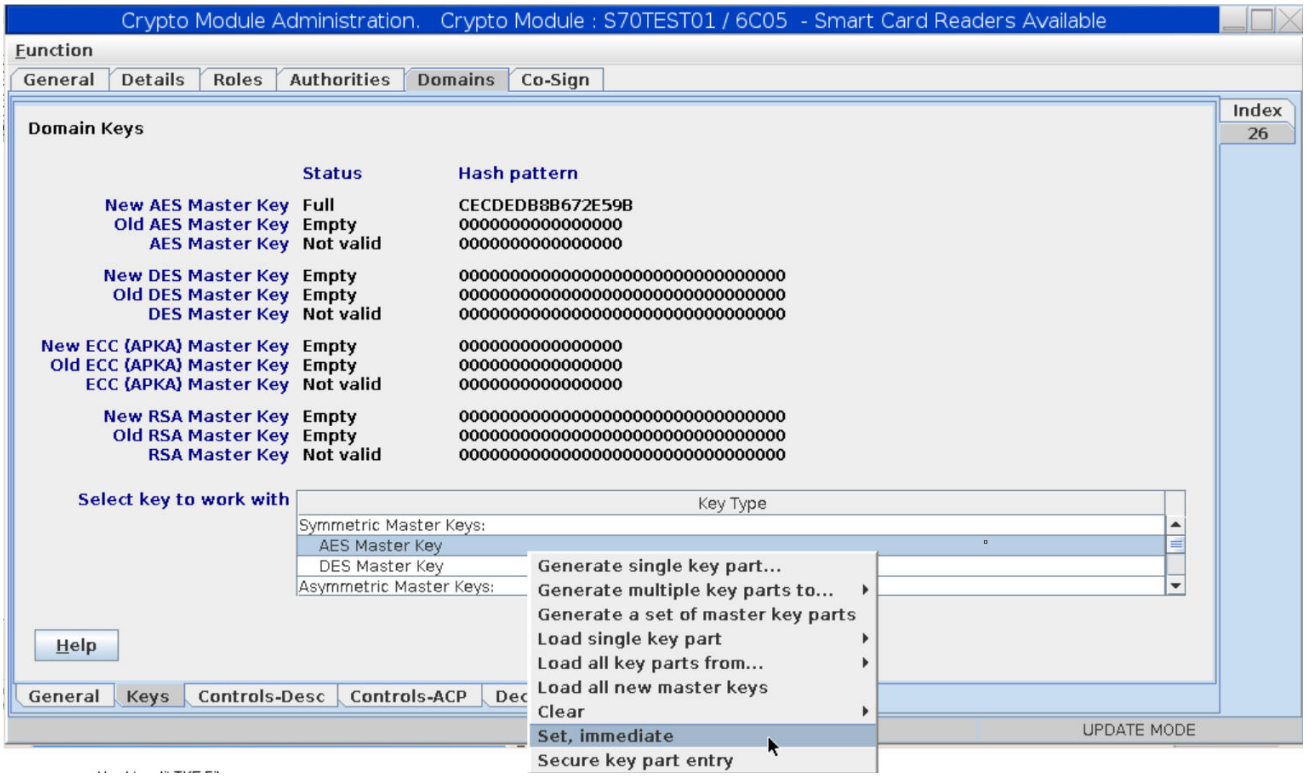

Figure 62. AES master key status: Full

The **Set, immediate** action transfers the master key from the **New AES Master Key** register (CECDE...) to the **AES Master Key** register. This master key will from now on be used to generate a secure key by wrapping the clear key. The **Set, immediate** action also transfers the previous master key from the **AES Master Key** register to the **Old AES Master Key** register.

Important: You can proactively re-encipher existing secure keys with the new master key which is still stored in the **New AES Master Key** register. Such re-enciphered secure keys are not valid until the new master key is activated to become the current master key.

So changing the master key must be coordinated between the persons who change the master key and the persons who own the secure keys, or who own applications that use the secure keys, or who own the data that is encrypted with these secure keys.

If you maintain secure keys in a key repository, or some type of a cryptographic key data set (CKDS), these keys need to be re-enciphered before the new master key is activated. If you implemented the infrastructure for protected volume encryption, using a key repository as described in Pervasive Encryption for Data Volumes, the available utilities support you in a staged re-enciphering of the secure keys in this repository: In a first stage, a secure key is re-enciphered with the key in the **New AES Master Key** register. The re-enciphered key is kept in the repository in a separate file. This ensures that while the new master key has not been set active, the secure keys continue to be valid. Once the new master key has been set active, the second stage of the re-enciphering process replaces the secure key with its re-enciphered version which was previously created during the first stage.

Before you can execute the **Set, immediate** action, you get a warning that informs you about the result of this action.

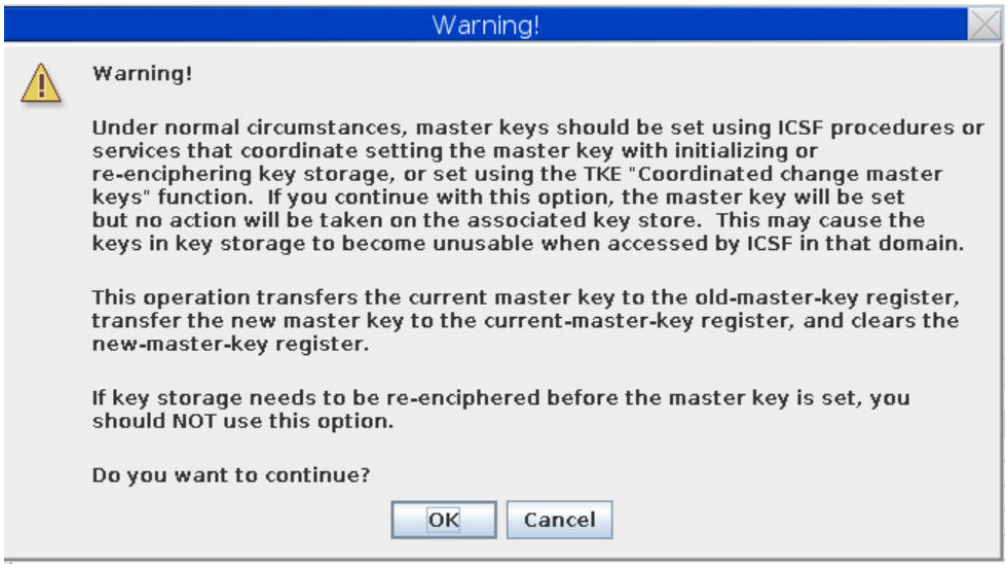

Figure 63. Warning

Important: In our scenario, there is no previous master key in the **Old AES Master Key** register. If this register is not empty in your environment, ensure that you really do no longer need the master key that may be stored in this register, because you will no longer be able to read any data that is encrypted with a secure key created with this old master key. You might consider re-enciphering existing secure keys wrapped by the old master key with the new master key before activating the new master key.

2. If you are sure about your key hierarchy and usage, press the **OK** button at the end of the warning (Figure 63 on page 35).

   You can see the result in Figure 64 on page 35. The master key with a hash beginning with CECDED… is now in the **AES Master Key** register with status **Valid**.

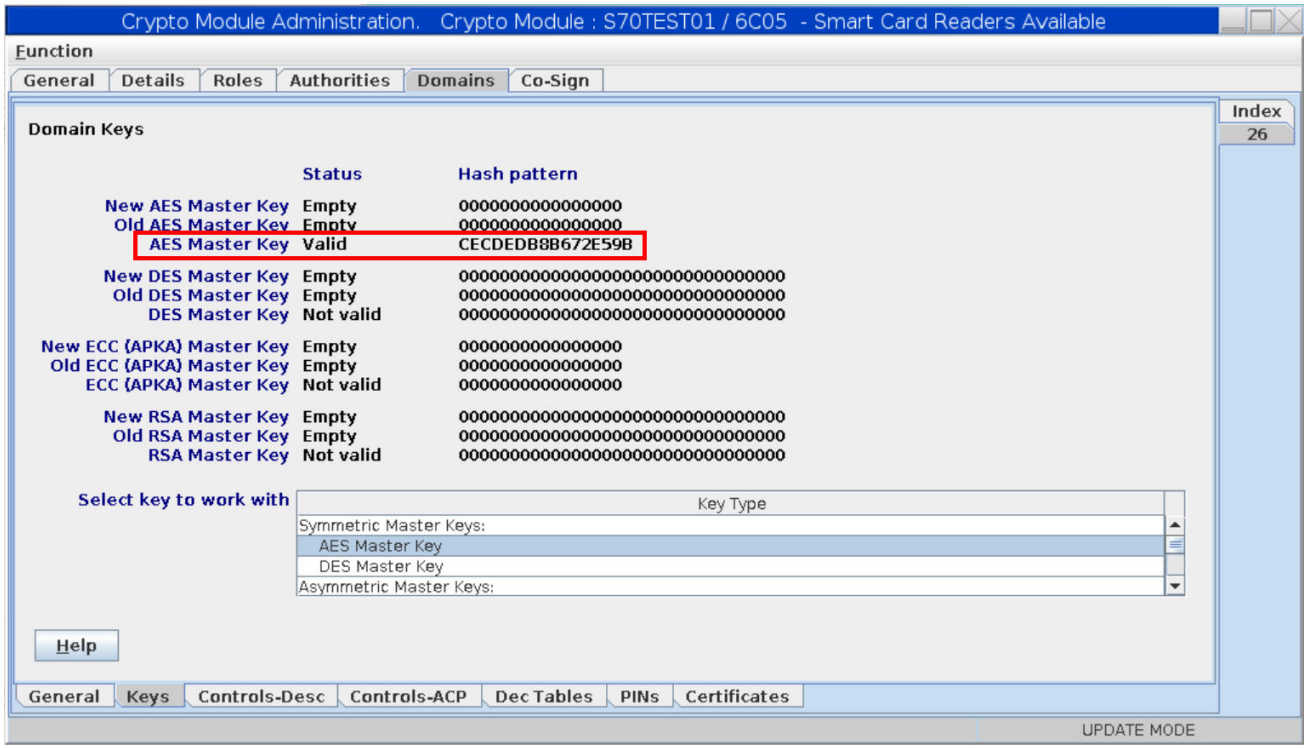

Figure 64. Checking the new valid AES master key

If applicable, a previously current master key is saved on the cryptographic coprocessor in the **Old AES Master Key** register for recovery actions, as long as you do not overwrite this register by generating a new current AES master key.

Results

Now you have loaded and activated one AES master key on one domain on one cryptographic coprocessor. The Linux on Z instance, where the cryptographic coprocessor is attached can use this master key to generate a secure key for data and volume encryption.

More information

- To learn about security products for Linux on Z and LinuxONE, search for the following terms in your web browser:

  ```
  IBM Knowledge Center -> Products -> Linux on IBM Systems -> Linux on Z and LinuxONE ->
  Security
  ```

  Or you can directly click or enter the following URL directly:

  ```
  https://www.ibm.com/support/knowledgecenter/de/linuxonibm/liaaf/security.html
  ```

- For information on the features and advantages of the infrastructure for protected volume encryption, search for this video in your web browser:

  ```
  Pervasive encryption for data volumes video
  ```

  Or you can directly click or enter the following URL directly:

  ```
  https://youtu.be/jDK3ZwEdX4I
  ```

- Documentation on how to establish and exploit the infrastructure for protected volume encryption is provided in the IBM Knowledge Center. Search for the following title in your web browser:

  ```
  (Linux on Z) Pervasive encryption for data volumes ->
  ```

  Or you can directly click or enter the following URL directly:

  ```
  https://www.ibm.com/support/knowledgecenter/linuxonibm/com.ibm.linux.z.lxdc/lxdc_linuxonz.html
  ```

Accessibility

Accessibility features help users who have a disability, such as restricted mobility or limited vision, to use information technology products successfully.

Documentation accessibility

The Linux on Z and LinuxONE publications are in Adobe Portable Document Format (PDF) and should be compliant with accessibility standards. If you experience difficulties when you use the PDF file and want to request a Web-based format for this publication, use the Readers' Comments form in the back of this publication, send an email to eservdoc@de.ibm.com, or write to:

IBM Deutschland Research & Development GmbH
Information Development
Department 3282
Schoenaicher Strasse 220
71032 Boeblingen
Germany

In the request, be sure to include the publication number and title.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

IBM and accessibility

See the IBM Human Ability and Accessibility Center for more information about the commitment that IBM has to accessibility at

```
www.ibm.com/able
```

Notices

This information was developed for products and services offered in the U.S.A. IBM® may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml

Adobe is either a registered trademark or trademark of Adobe Systems Incorporated in the United States, and/or other countries.

SC34-7712-00