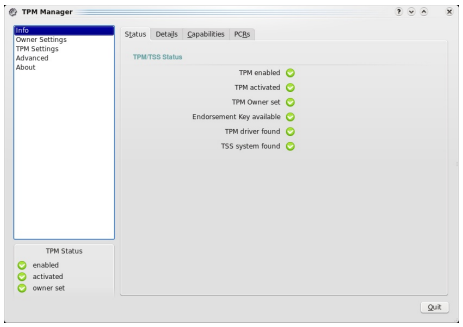
The Turaya Security Framework

The TPM Manager

Requirements, Design, and Implementation

July 1, 2009 Version 0.8



Sirrix AG security technologies

Sirrix AG security technologies
Lise-Meitner-Allee 4
D-44801 Bochum, Germany
http://www.sirrix.com

Author:

Anoosheh Zaerin
a.zaerin@sirrix.com

Advisors:

Ahmad-Reza Sadeghi
ahmad.sadeghi@trust.rub.de

Christian Stüble
c.stueble@sirrix.com

Contents

List of Figures

1 Introduction

1.1 Motivation and Problem Description

The computer industry has recently come up with Trusted Computing (TC), a new generation of computing platforms based on new architectures both in hardware and software. The results of their investigations are the Trusted Computing Group (TCG) initiative that has published the corresponding hardware and software specifications [3]. The basic idea is to embed elementary security functions into the underlying hardware of the computing platform while keeping the assumption on tamper-resistance as weak as possible, and reducing costs by keeping the trusted component as small as possible.

The stated goal of these architectures is to improve the security and trustworthiness of computing platforms. Indeed, these platforms offer many useful functions which can be used to increase a platform's security. They extend the conventional PC architecture by new mechanisms to (i) protect cryptographic keys, (ii) generate random numbers in hardware, (iii) authenticate (the configuration of) a platform (attestation), and (iv) cryptographically bind the data to be encrypted to certain information, e.g., the system configuration and the identifier of the invoking application (sealing).

However, there is still an ongoing public debate about the negative economical, social, and technical consequences of these platforms. People are concerned about the potential dangers of the capabilities of such platforms: They may give vendors and content providers too much control power over personal systems and users' private information. Although most complains about trusted computing are speculative, it is highly important to observe its development carefully, and to improve this technology by putting together the missing pieces for more secure platforms in future. Today, only a few software components supporting TPMs are available, especially in the context of open-source operating systems like Linux. We observed that many people, who would like to experience with a Trusted Platform Module (TPM), have many problems using the provided management software, i.e., libtpm and Trousers. Moreover, many open-source projects in the context of trusted computing lack support of a TPM management.

1.2 Goals

The goal of "TPM Manager" project is the realization of an open source TPM management software providing an easy-to-use graphical user interface. Currently, the TPM Manager can be used under Linux, while later releases should also be usable as a compartment executed on top of a security kernel such as Turaya or OpenTC. Current releases of the TPM Manager

were developed by Sirrix AG and Ruhr-University Bochum. The TPM Manager is published under version 2 of the GNU GPL license.

2 Requirements Specification

This chapter defines the functional and supplementary requirements of the TPM Manager. It defines the target groups, roles and actors and gives an overview of use cases in Section 2.4.

2.1 Target Groups

This section defines the users/other components that wish to use the product.

- Home user (Single-user platform at home)

2.2 Roles and Actors

In this section we define different roles and actors important for the use case model. Actors are parties outside the system that interact with the system; an actor can be a class of users, roles users can play, or other systems. Note that, depending on the use case, some parties or actors may not be involved.

- Owner: The owner of a platform is an entity who defines the allowed configurations of the underlying platform. Note that this also includes certain changes to the platform's configuration. In practice, these changes are patches/updates. The platform owner is also owner of the TPM and thus is aware of the owner authorization information. Typical examples are an enterprise represented by an administrator or an end-user owning a personal platform.
- User: The user of a computing platform is an entity interacting with the platform under the platform's security policy. Examples are employees using enterprise-owned hardware. User and owner might also be identical, e.g., in an end-user environment.

2.3 Functional Requirements (Use Case Model)

Each use case focuses on describing how to achieve a single business goal or task. From a traditional software engineering perspective a use case describes just one feature of the system. For most software projects this means that multiple, perhaps dozens, of use cases are needed to fully specify the new system. The degree of formality of a particular software project and the stage of the project will influence the level of detail required in each use case.

A use case defines the interactions between external actors and the system under consideration to accomplish a business goal. Actors are parties outside the system that interact with the system; an actor can be a class of users, roles users can play, or other systems.

Use cases treat the system as a "black box", and the interactions with the system, including system responses, are perceived as such from outside the system. This is a deliberate policy, because it simplifies the description of requirements, and avoids the trap of making assumptions about how this functionality will be accomplished.
A use case should:

- describe a business task to serve a business goal
- have no implementation-specific language
- be at the appropriate level of detail
- be short enough to implement by one software developer in a single release.

2.4 Use case overview

Figure 2.1 gives on overview of all use cases realized by the TPM Manager. The use cases are separated into different subsets:

- Info: Display information about the TPM that can be read without an authorization.
- Owner Settings: Use cases related to TPM Owner management and SRK management issues.
- TPM Setting: Use cases related to TPM state management.
- Advanced: Critical use cases that may cause permanent changes to the TPM
- Misc: Utility use cases like password dialogs or warning messages.

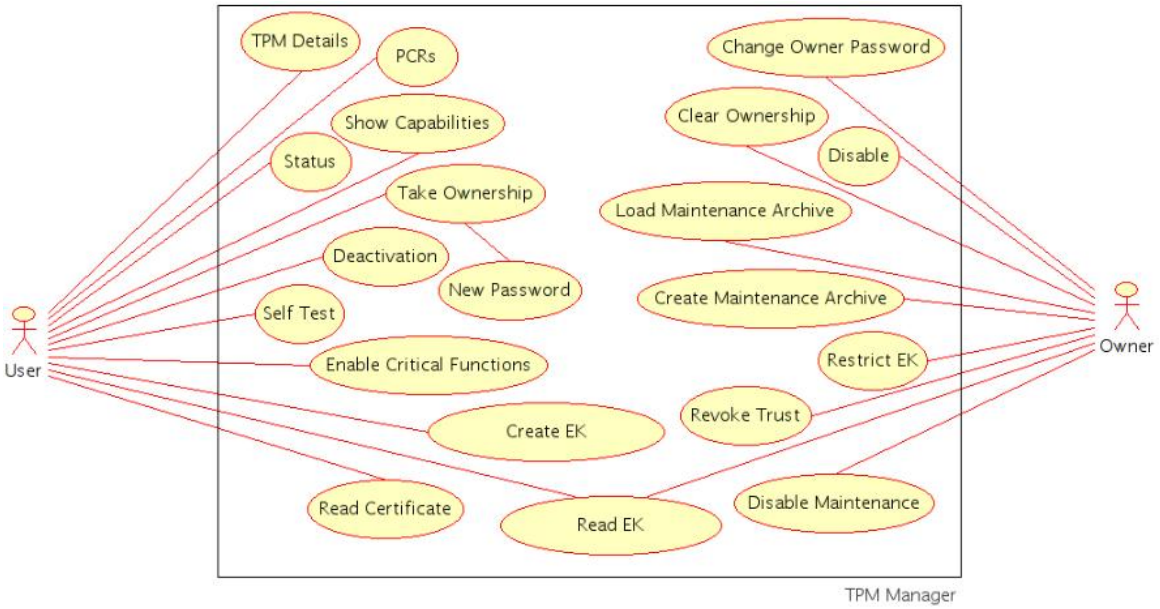


Figure 2.1: Overview of all use cases realized by the TPM Manager.

2.4.1 Info

The use cases described in this section display information about the TPM and can be read without authorization.

Use Case Unique ID	/UC 10/
Title	Info
Description	Show the status of the TPM.
Actors	User
Normal Flow	1. User clicks on the ”Info“ tab 2. The following information will be displayed: • TPM enabled • TPM activated • TPM owner set • Endorsement Key available • TPM driver found • TSS system found

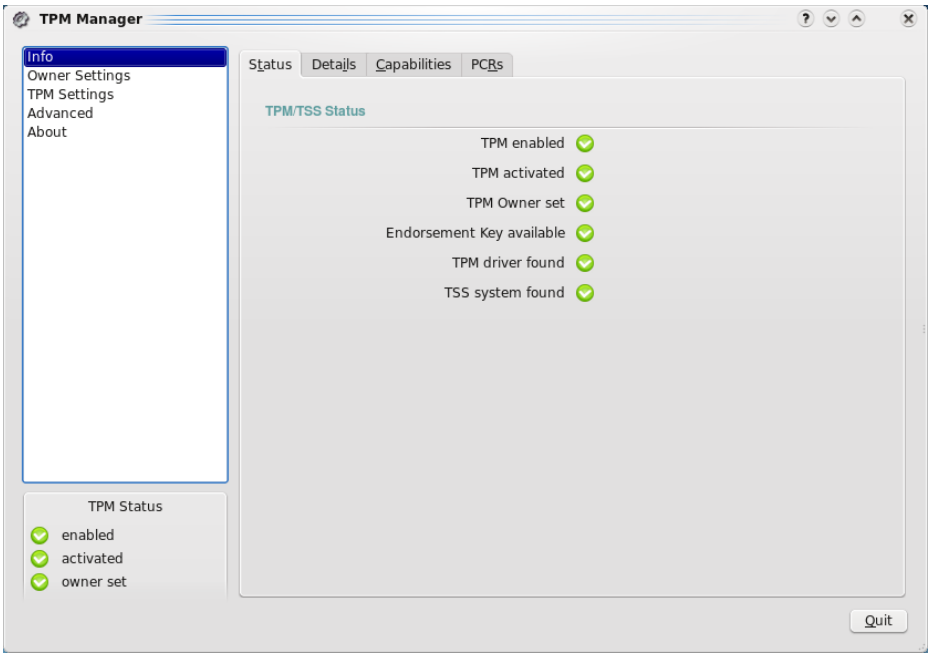


Figure 2.2: Overview of /UC 10/.

Use Case Unique ID	/UC 20/
Title	TPM Details
Description	Show detailed information about the TPM and the TSS
Actors	User
Normal Flow	1. User clicks on the "Details" tab 2. The following information will be displayed: • Information about TPM such as: – Vendor – Version – Firmware • Information about TSS such as: – Vendor – Version

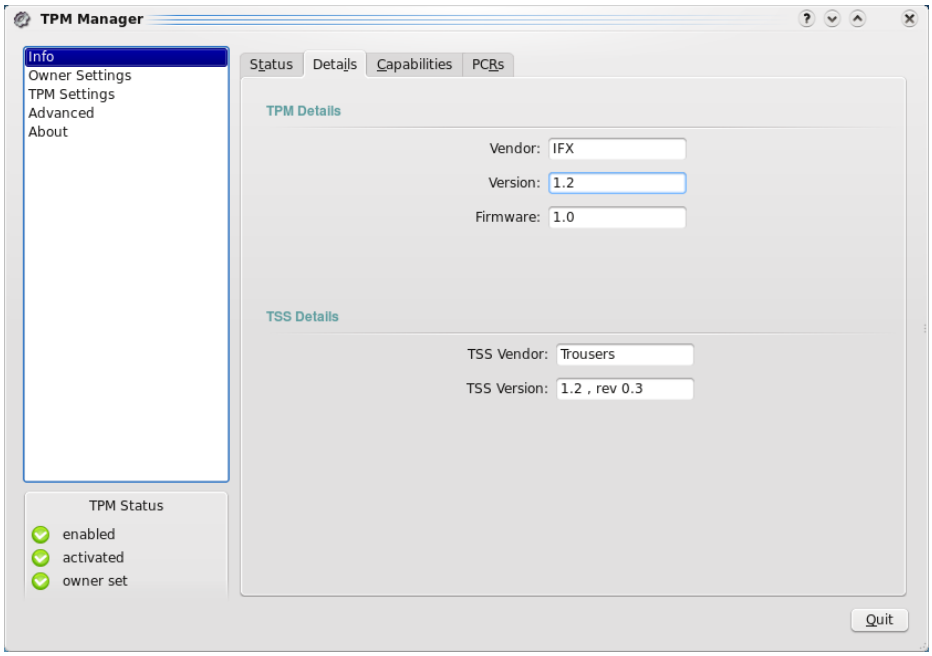


Figure 2.3: TPM and TSS details tab dialog /UC 20/

Use Case Unique ID	/UC 30/
Title	Show TPM Capabilities
Description	Show the capabilities of the TPM
Actors	User
Normal Flow	1. User clicks on the "Capability" tab 2. The following information will be displayed: • Number of PCRs • Available features • etc.

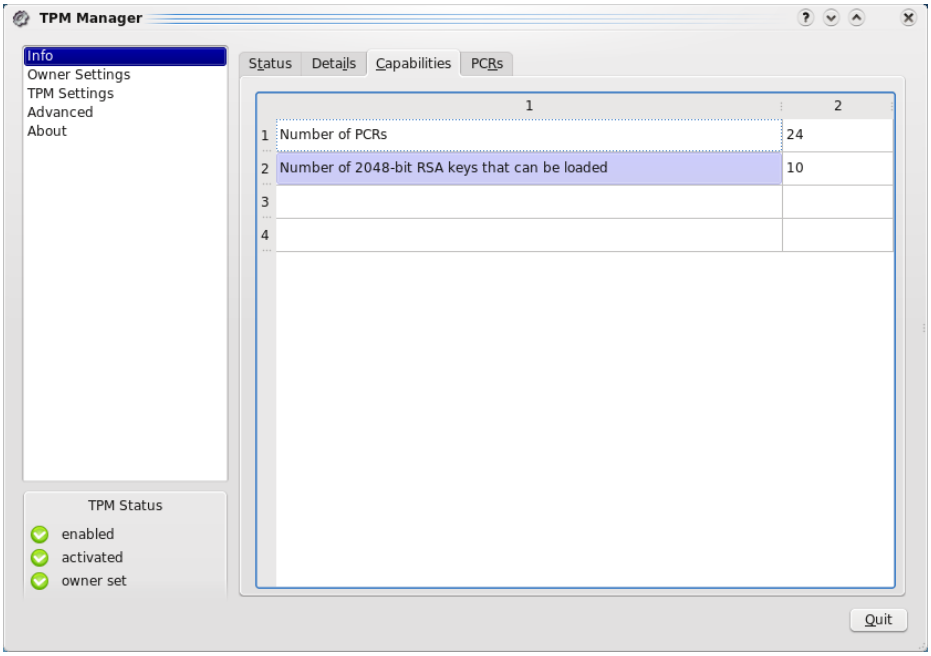


Figure 2.4: Dialog showing the capabilities supported by the TPM /UC 30/.

Use Case Unique ID	/UC 40/
Title	PCRs
Description	Show the current PCR values.
Actors	User
Normal Flow	1. User clicks on the ”PCRs“ tab 2. Display current PCR values
Notes	The PCR view is updated every 1 second.

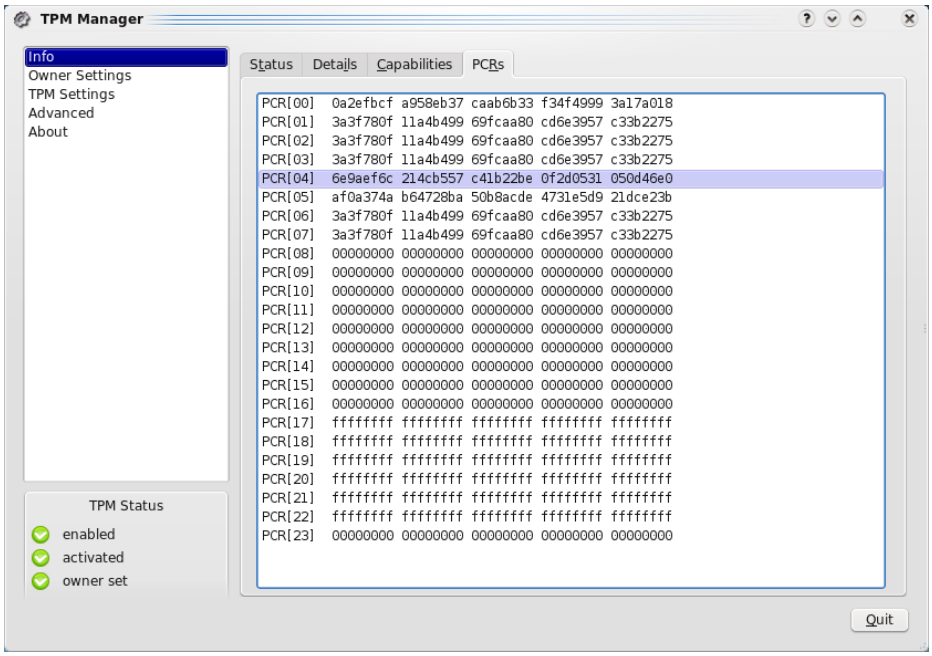


Figure 2.5: PCR value dialog /UC 40/

2.4.2 Owner Settings

The use cases of this section are related to TPM Owner management and Storage Root Key (SRK) management issues.

Use Case Unique ID	/UC 50/
Title	Take Ownership
Description	Configure the TPM owner and set the embedded security features
Actors	Owner
Includes	New Password
Preconditions	No TPM owner set
Postconditions	TPM owner set
Normal Flow	1. User clicks on the "Ownership" tab 2. User hits the "Take" button 3. Displays a dialog to choose the owner password 4. User enters the owner password and confirms his choice 5. TPM Manager displays a dialog to choose the type of SRK password 6. User chooses the default choice (set SRK password to WELL_KNOWN_SECRET) 7. TPM Manager sets the owner and SRK password and displays a dialog whether taking ownership was successful or not
Alternative Flow	1. User clicks on the "Ownership" tab 2. User hits the "Take" button 3. TPM Manager displays a dialog to choose the owner password 4. User enters the owner password and confirms his choice 5. TPM Manager displays a dialog to choose the type of SRK password 6. User chooses the second choice (set SRK password manually) 7. TPM Manager displays a dialog to choose the SRK password 8. User enters SRK password and confirms his choice 9. TPM Manager sets the password and displays a dialog whether taking ownership was successful or not

Use Case Unique ID	/UC 60/
Title	Change Owner Password
Description	Change the password used to authenticate the TPM owner
Actors	Owner
Includes	• Owner authentication /UC 220/ • New Password /UC 210/
Preconditions	TPM owner set
Postconditions	New password for TPM Owner
Normal Flow	1. User clicks on the "Ownership“ tab 2. User hits the corresponding "Change“ button 3. TPM Manager displays a dialog window asking for current owner password 4. TPM owner enters the current owner password 5. TPM Manager display a dialog asking for new owner password 6. TPM owner enters new owner password 7. TPM Manager sets the new owner password and displays a dialog whether the new owner password was successfully set or not

Use Case Unique ID	/UC 70/
Title	Change SRK Password
Description	Change the password used to protect the Storage Root Key (SRK)
Actors	Owner
Includes	• SRK authorization /UC 230/ • New Password /UC 210/
Preconditions	TPM Owner set
Postconditions	New password for SRK authorization
Normal Flow	1. User clicks on the "Ownership" tab 2. User hits the corresponding "Change" button 3. TPM Manager displays a dialog window asking for current SRK password 4. User enters the current SRK password 5. TPM Manager display a dialog asking for new SRK password 6. User enters new SRK password 7. TPM Manager sets the new SRK password and displays a dialog whether the new SRK password was successfully set or not

Use Case Unique ID	/UC 80/
Title	Clear Ownership
Description	Delete TPM ownership and set TPM to factory defaults
Actors	Owner
Includes	• Owner authentication /UC 220/ • Warning /UC 240/
Preconditions	TPM owner set
Postconditions	TPM Disabled, Deactivated and no TPM Owner set
Normal Flow	1. User clicks on the "Ownership" tab 2. User hits the "Clear" button 3. TPM Manager displays a dialog window asking for owner password 4. TPM owner enters the owner password 5. TPM Manager displays a dialog whether the TPM was successfully cleared or not

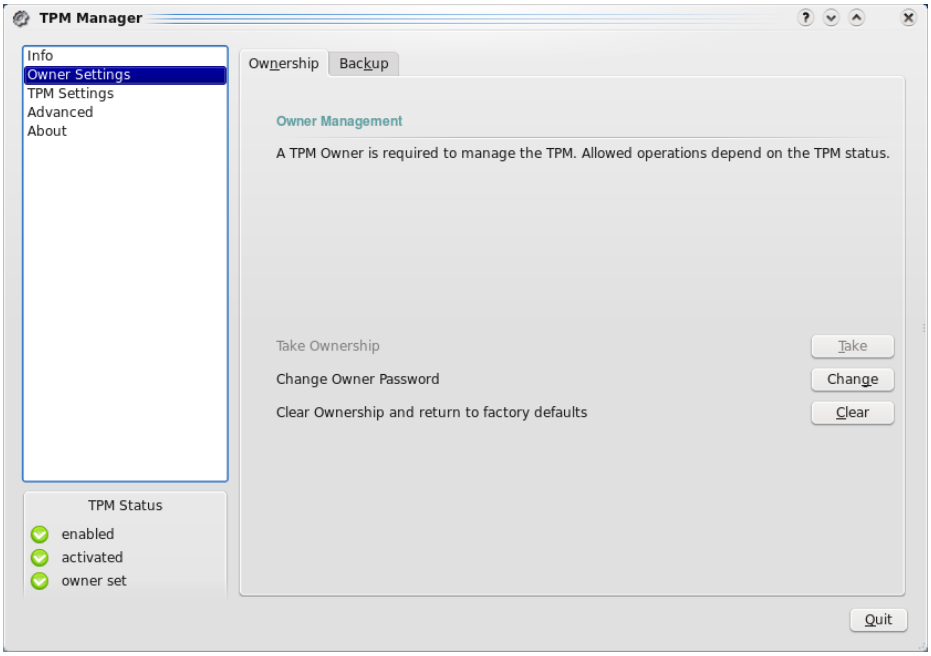


Figure 2.6: "Ownership" tab dialog /UC 50/, /UC 60/, /UC 70/ and /UC 80/

Use Case Unique ID	/UC 90/
Title	Create Maintenance Archive
Description	Create a backup of the SRK
Actors	Owner
Includes	• File selection • Owner authorization
Preconditions	TPM supports maintenance feature
Normal Flow	1. Owner hits “Create” maintenance button 2. Owner selects destination for archive file 3. Owner authorization 4. Owner confirms creation of archive

Use Case Unique ID	/UC 100/
Title	Load Maintenance Archive
Description	Restore the SRK backup archive
Actors	Owner
Includes	• File selection • Owner authorization
Preconditions	SRK archive was created by the TPM manufacturer
Normal Flow	1. Owner hits “Load” button 2. Owner selects maintenance archive file 3. TPM Manager asks for owner authorization 4. Owner confirms loading maintenance archive

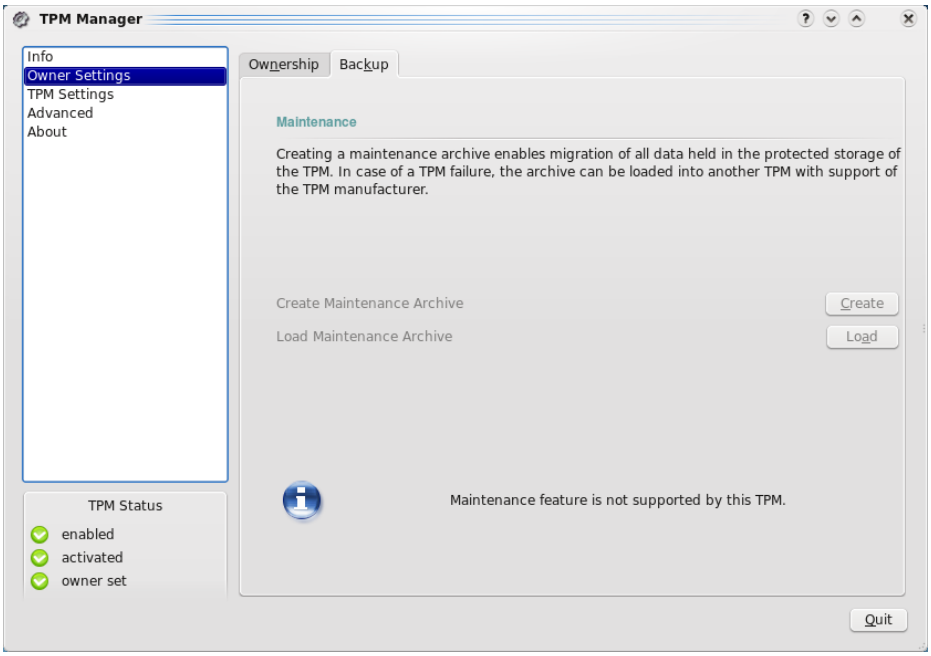


Figure 2.7: ”Maintenance“ tab dialog /UC 90/, /UC 100/

2.4.3 TPM Settings

Use Case Unique ID	/UC 110/
Title	TPM Deactivation
Description	TPM can be switched from activated to temporarily deactivated mode
Actors	User
Preconditions	TPM is activated
Postconditions	TPM is temporarily deactivated
Normal Flow	1. User hits "Deactivate" button 2. TPM Manager displays warning message 3. TPM Manager displays a dialog whether TPM was successfully deactivated or not
Notes	On 1.2 TPMs, temporarily deactivating the TPM requires operator authorization. When trying to deactivate a TPM 1.2 without operator authorization, a "bad physical presence" exception is thrown.

Use Case Unique ID	/UC 120/
Title	Disable (Enable) the TPM
Description	TPM can be switched to enabled or disabled mode
Actors	Owner
Includes	• Owner Authorization /UC 220/ • Warning /UC 240/
Normal Flow	1. Owner hits “Disable“ (“Enable“) button 2. TPM Manager asks for owner authorization 3. TPM Manager displays a warning message 4. TPM Manager displays a dialog whether TPM was successfully disabled (enabled) or not

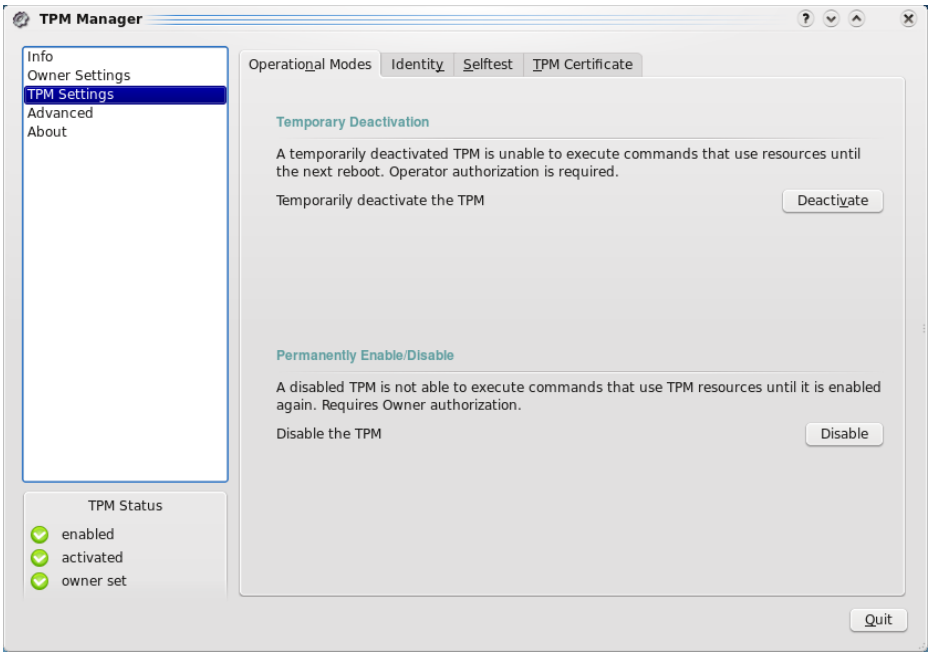


Figure 2.8: TPM operational modes dialog implementing /UC 110/ and /UC 120/

Use Case Unique ID	/UC 130/
Title	Create Endorsement Key
Description	Create Endorsement Key
Actors	User
Includes	• Owner Authorization /UC 220/ • Warning /UC 240/
Preconditions	No Endorsement Key is set by manufacturer
Postconditions	TPM has Endorsement Key

Use Case Unique ID	/UC 140/
Title	Read Public Endorsement Key
Title	Read the public part of EK
Actors	User

Use Case Unique ID	/UC 150/
Title	Store Public Endorsement Key
Title	Store the public part of EK outside the TPM
Actors	User

Use Case Unique ID	/UC 160/
Title	Restrict Public Endorsement Key
Description	Disable reading of EK without owner authorization
Actors	Owner
Includes	• Owner Authorization /UC 220/ • Warning /UC 240/
Preconditions	• Owner set • Endorsement Key is not restricted
Postconditions	Need owner authentication to read Endorsement Key

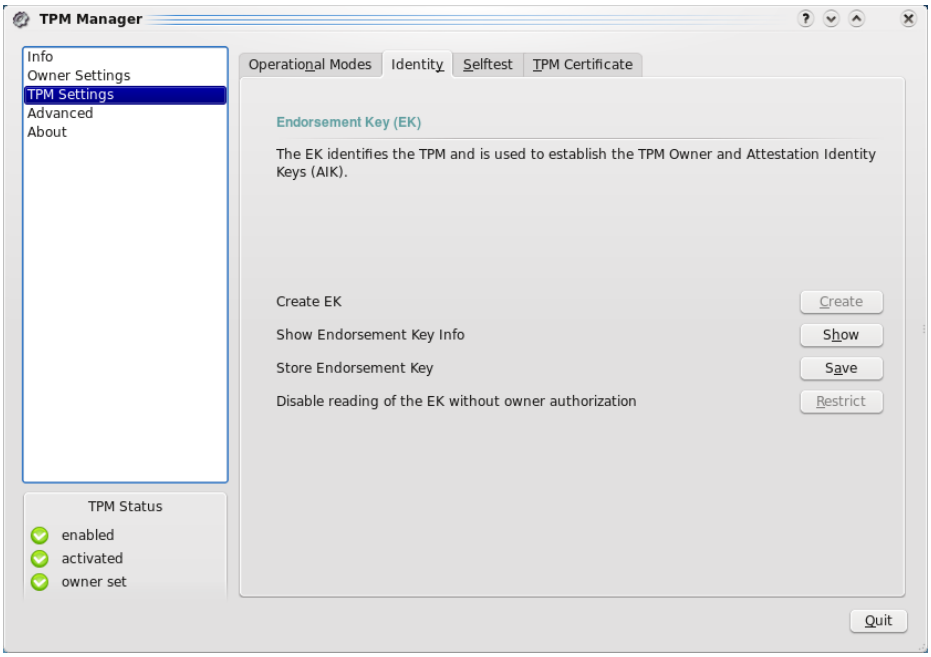


Figure 2.9: "Endorsement" tab dialog /UC 130/, /UC 140/, /UC 150/ and /UC 160/

Use Case Unique ID	/UC 170/
Title	Self test
Description	Perform a full self test of each TPM internal function
Actors	User

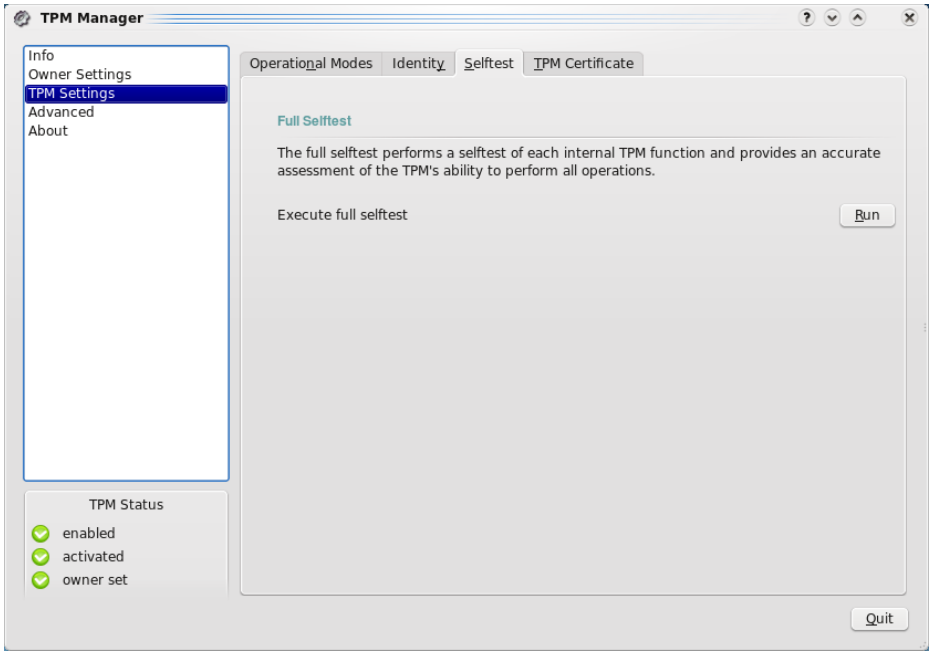


Figure 2.10: Tab dialog to perform a full TPM self test /UC 170/

2.4.4 Advanced

Use Case Unique ID	/UC 180/
Title	Warning Critical Functions
Description	The "Advanced" section warns the user about the criticality of the features provided, because these features can do permanent changes to the TPM
Actors	User

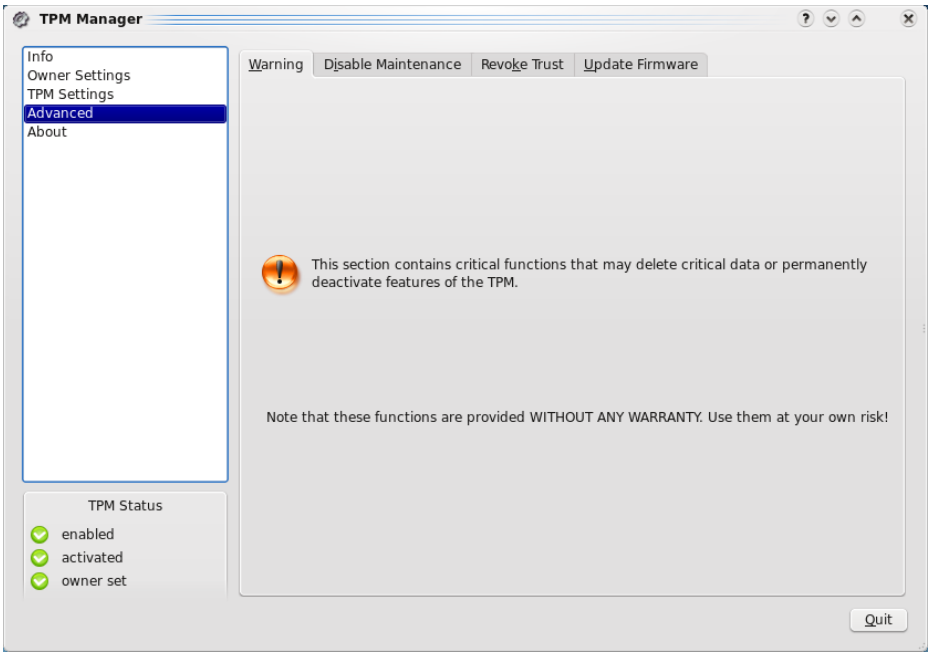


Figure 2.11: Tab dialog to warning about critical features /UC 180/

Use Case Unique ID	/UC 190/
Title	Disable Maintenance Feature
Description	Kill Maintenance feature
Actors	Owner
Includes	Owner authorization

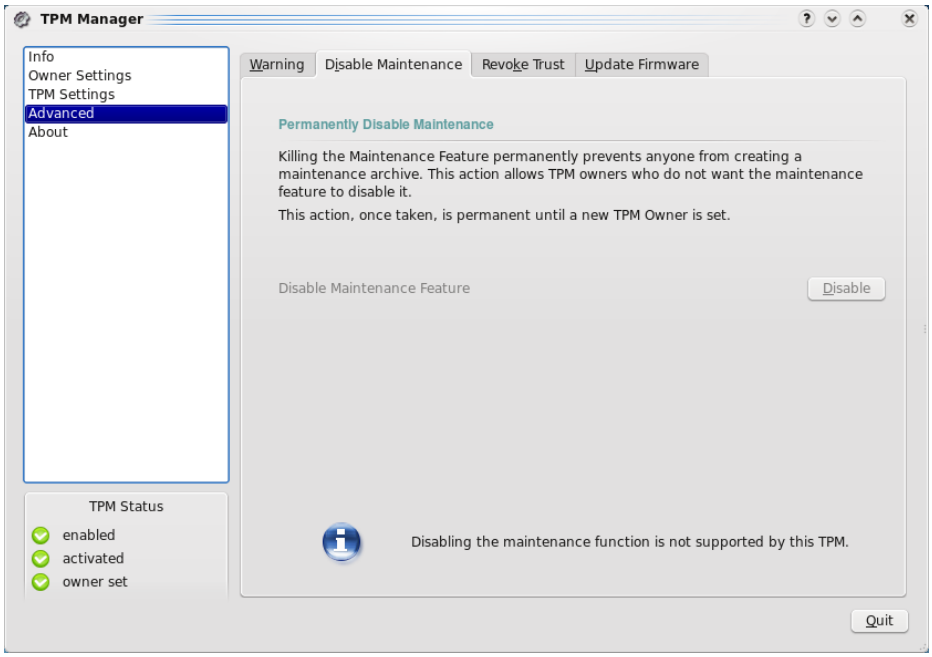


Figure 2.12: Disable Maintenance /UC 190/

Use Case Unique ID	/UC 200/
Title	Revoke Trust
Description	Delete Endorsement Key
Actors	Owner
Includes	• Owner Authorization /UC 220/ • Warning /UC 240/
Preconditions	TPM has Endorsement Key
Postconditions	No Endorsement key

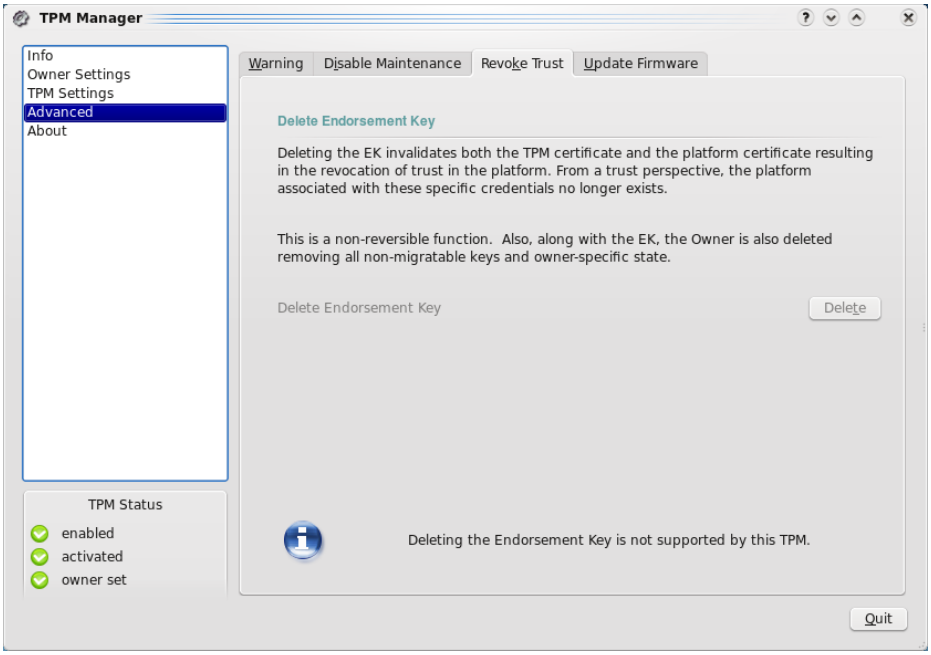


Figure 2.13: "Revoke Trust" tab dialog /UC 200/

2.4.5 Misc.

The uses cases of this section include utility dialogs like password dialogs or warning messages.

Use Case Unique ID	/UC 210/
Title	New Password
Description	User/Owner enters a new passphrase
Actors	Owner, User

Use Case Unique ID	/UC 220/
Title	Owner Authentication
Description	Owner authorization using the owner secret
Actors	Owner

Use Case Unique ID	/UC 230/
Title	SRK Authorization
Description	SRK authorization using the SRK secret
Actors	User

Use Case Unique ID	/UC 240/
Title	Warning Message
Description	System shows a warning message
Actors	Owner, User

2.5 Supplementary Requirements

This section describes obligatory criteria, mandatory for successful completion.

2.5.1 Preconditions

Requirements that have to be fulfilled already, because they were needed for the development process are descriebd in this section.

/PR 10/ TSS

The TPM Manager will depend on an existing TCG Software Stack (TSS) implementation.

/PR 20/ Widget Library

The TPM Manager will depend on an open-source widget library.

/PR 30/ TPM-enabled BIOS

The BIOS has to support the TPM to be able to enable it.

2.5.2 Required Criteria

Mandatory criteria, that are obligatory for successful completion are descriebd in this section.

/RC 10/ Linux support

The realization of the use cases should be based on an Linux-based architecture.

2.5.3 Desired Criteria

Optional criteria, that are not mandatory for successful completion are descriebd in this section.

/DC 10/ Turaya support

The realization of the use cases should be based on a Turaya-based architecture.

2.5.4 Execution Environment

This section specifies software and hardware the user requires at least to run our product successfully.

2.5.4.1 Software

- (required) Linux Distribution based on kernel 2.6.x, including
  - TrouSerS-TSS
  - TPM Driver
  - Widget Library
- (optional) Turaya Architecture, including
  - TPM Driver
  - Widget Library
  - Trust Manager

2.5.4.2 Hardware

- HP Notebook with Infineon TPM 1.1.b
- HP Notebook with Infineon TPM 1.2
- Desktop PC with ST TPM 1.2
- ThinkPad T41p with Atmel TPM 1.1b

2.5.5 Development Environment

This section specifies hard- and software that developers need at least to implement the product successfully.

2.5.6 Software

- Linux 2.6.x
- gcc 4.3.x
- Qt4 libraries 4.4.x
- QDevelop
- Qt4 Designer

2.5.7 Hardware

- HP Notebook
- TPM 1.1b
- TPM 1.2

3 Software Architecture

3.1 Introduction

This chapter provides an overview of the TPM Manager's design. In Section 3.2 the architecturally significant parts of the design model are described, and Section 3.3 illustrates how the TPM Manager actually works by giving a few selected use-case realizations, and explains how the various design model elements contribute to their functionality.

3.2 Logical View

This section describes the architecturally significant parts of the design model, such as its decomposition into subsystems and packages. And for each significant package, its decomposition into classes and class utilities is illustrated.

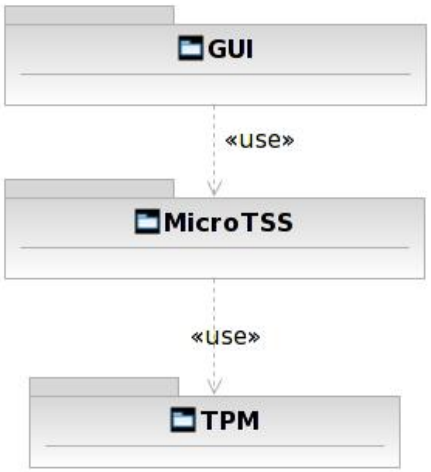


Figure 3.1: The two layers of TPM Manager design

3.2.1 Overview

This subsection describes the overall decomposition of the design model in terms of its package hierarchy and layers.

As illustrated in Figure 3.1, the design of the TPM Manager has two layers, the Graphical User Interface (GUI) layer and the MicroTSS layer. While the GUI layer provides the user interface components, based on widgets of the Qt[1] framework, the MicroTSS layer offers a simple, object-oriented interface to the functions of the underlying TPM.

Although the current implementation of the MicroTSS is based on the TrouSerS TSS, the abstraction provided by the MicroTSS layer allows support from alternative TSS implementations offered by a third party or from the required functions implemented directly. The latter is especially important if the TPM Manager should be used in a security-critical environment, e.g., as a part of a security kernel.

3.2.2 Architecturally Significant Design Packages

The GUI layer. The GUI layer of the TPM Manager offers an interface to the user and uses the MicroTSS layer to access TPM functions. The logic of the user interface layer is realized by the Qt-based class `TPM_Manager`. Its base class, `TPM_ManagerBase`, is created by the Qt Designer, Trolltech's tool for designing and building GUI's. The Qt Designer allows the addition of new functionality without much effort. Figure 3.2 shows an example screenshot of the TPM Manager GUI in use.

[1] http://www.qtsoftware.com/

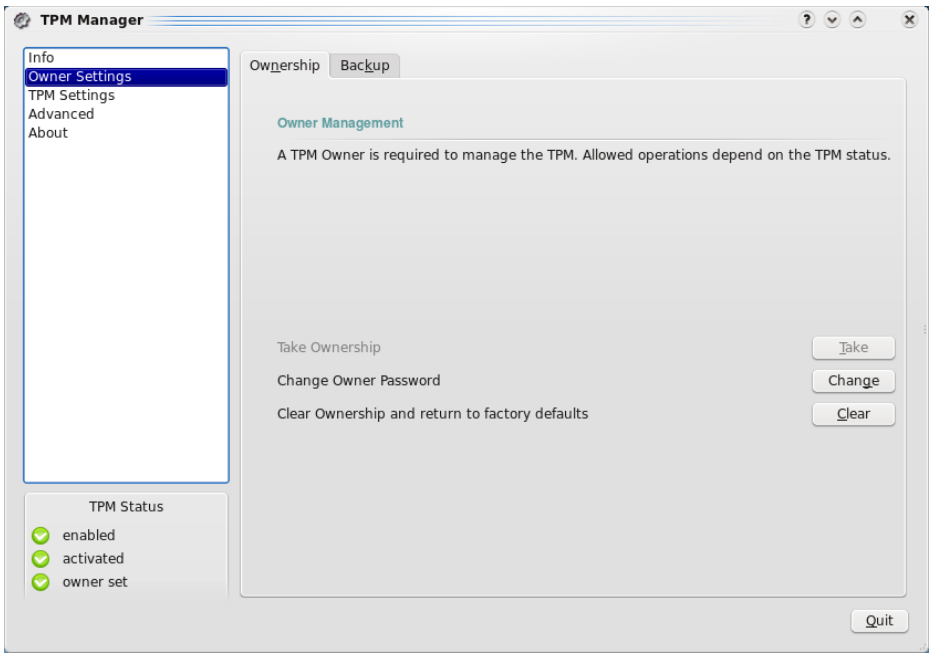


Figure 3.2: “Ownership” tab Dialog

The MicroTSS layer. The MicroTSS layer provides an abstract interface to access the TPM and hides implementation details. Figure 3.3 shows the Unified Modeling Language (UML) model of the public interfaces and components provided by the TPM Manager and the TSS Service Provider Interface (TSPI). The main component is the class `TPM` that implements the TPM management functions. The class `TPM` is created by the class `TSS` managing access to the TSS implementation in use. For example, the class `TSS` checks the availability of a TPM driver before creating an object of type `TPM`. The class `PublicKey` provides information about types and attributes of cryptographic encryption and test keys managed by the `TSS`. Although the MicroTSS interface includes a small number of functions related to version 1.2 of the TSS specification[2], the main portion is based on TrouSerS TSS version 1.1b[1].

The current implementation of the class `TPM` itself uses the TSPI interface provided by the `libtspi.so` library of TrouSerS. Since the TSPI interface includes many functions, in Figure 3.3, only the functions that are required to realize the use case "Take Ownership" are described in Section 3.3.1.

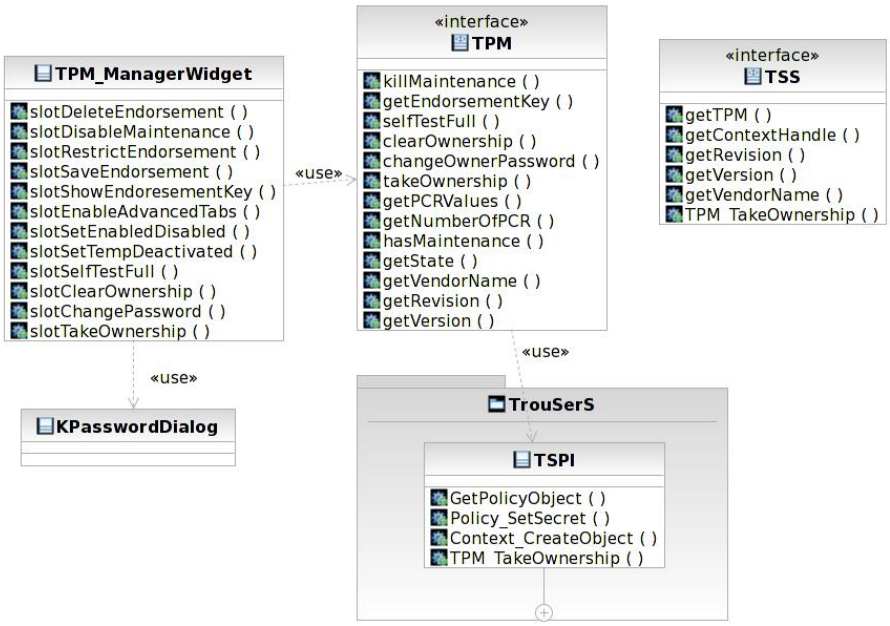


Figure 3.3: Interfaces provided and used by the TPM Manager and the MicroTSS

3.3 Use Case Realization

To illustrate the dynamic behavior of the TPM Manager design packages, the use cases “Take Ownership”, “Enable/Disable“ the TPM and ”PCRs“ are explained in the following.

3.3.1 Use Case Take Ownership

The sequence diagram illustrated in Figure 3.4 describes the message and control flow upon invocation of the “Take Ownership” function in the TPM Manager.

After the user presses the appropriate button of the TPM Manager dialog, the method `on_myTakeOwnership_clicked`() of the class `TPM_Manager` is invoked (step 1). Here, two password dialogs are created for entering the TPM owner password and the SRK passoword (steps 1.1 and 1.3). Next, the MicroTSS, more concretely the method `takeOwnership`() of the class `TPM`, is invoked (step 1.5). This method hides the complexity of the interface of the `libtspi.so` by invoking `TSPI` functions by setting the owner password and the `SRK` password to finally take ownership of the TPM (steps 1.5.1 to 1.5.11). Lastly, a dialog informs the user about the result of the take ownership operation (step 1.7).

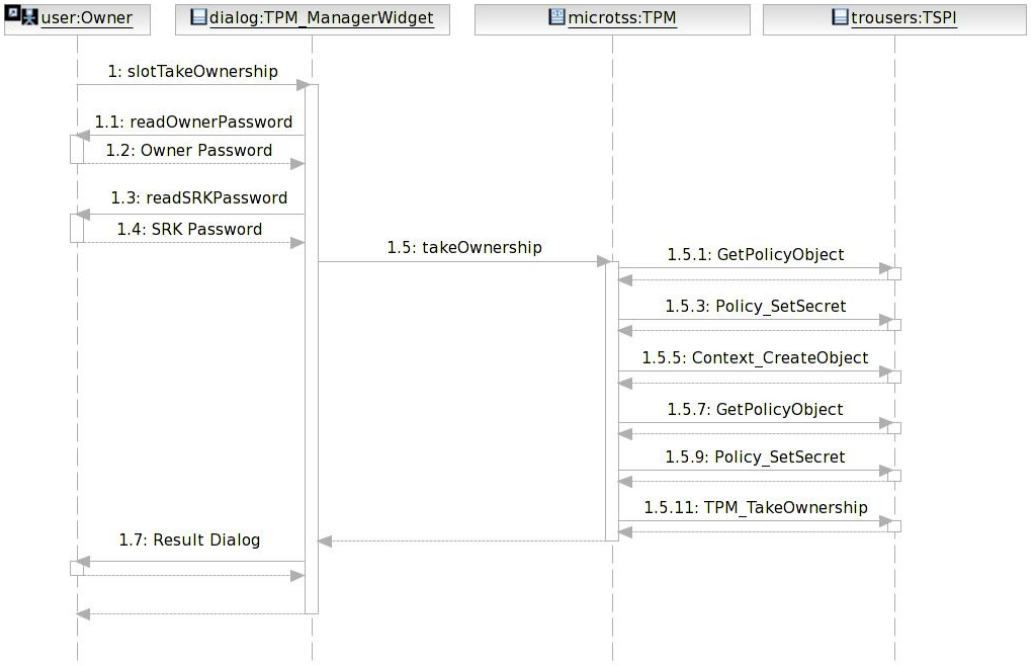


Figure 3.4: Control flow for “Take Ownership“ function in the TPM Manager

3.3.2 Use Case Enable/Disable

The "Enable"/"Disable" button on the "Operational Mode" page of the TPM Manager GUI will be only active if the ownership of the TPM is done successfully. The caption of the button is set by class `TPM_Manager` correspondence to the current status of the TPM. If the TPM is enabled the button is set to Disable and if the TPM is disabled the caption of the button will be Enable.

After the user clicks on the "Enable"/"Disable" button on the TPM Manager dialog, the appropriate method `on_myDisable_clicked`() of the class `TPM_Manager` is called. This method invokes a password dialog that asks for owner password. After the correct owner password is given by the user, the appropriate funtions `setDisabled`()/`setEnabled`() will be called, that are methods of the class `TPM`. Lastly, a dialog informs the user about the result of the Enable/Disable operation.

3.3.3 Use Case PCRs

Choosing "Info" option from the left menu and then choosing the "PCRs" tab the function of `TPM_Manager` will be called. It shows the current values of the PCRs only if the TPM is activated. The values are regularely updated by a signal coming from the system timer every 1 second.

4 User Manual

4.1 Requirements

Since the TPM Manager is based entirely on the Qt UI framework, corresponding header and library files Qt4 should be in the library path. On some linux distributions you have to install the developer version of Qt to have the header files used by TPM Manager.

The required programs to install the TPM Manager are:

- Qt4
- TrouSerS

The required packages for kubuntu are:

- build-essential
- libtspi-dev
- libtspi1
- trousers
- libqt4-dev

To use the features of the TPM Manager you need a running TrouSerS daemon. The TPM Manager has been successfully compiled under Qt 4.4.3 and KDE 4.1 resp. Gnome 2.26.

4.2 Installation and Configuration

The TPM Manager is hosted on sourceforge at http://sourceforge.net/projects/tpmmanager/. The software is based on qmake; therefore, it can be configured and installed as described below. Note that on some systems, qmake points to an older version of Qt: Qt3. Check your version of Qt prior to compiling TPM Manager as described below.

Listing 4.1: Configuring and compiling the TPM Manager:

```
# tar -xzf tpmmanager-0.6.tar.gz
# cd tpmmanager-0.7
# qmake --version
# qmake // if qmake --version returned Qt version 3.x.x, use qmake-qt4 instead
# make
# install bin/tpmmanager /your/install/path
```

Note: e.g. /your/install/path → /usr/local/bin.

4.3 Usage

After successfully installing the TPM Manager you can run the executable on the shell by call `tpmmanager` or you may add it to your system application menu.

4.3.1 Overview

As illustrated in Figure 4.1 there are 3 parts in the TPM Manager GUI. Part 1 is mainly used as a list to select functional areas of the TPM Manager. Clicking on the items as "Info", "Owner Settings", etc. changes the view of Part 2. Part 2 is itself separated into different tabs. Each tab contains a group of functionality. Part 3 shows the current status of the TPM. Figure 4.2 shows on the left side the status of a TPM with no owner and on the right side the status of a disabled TPM and no owner is set.

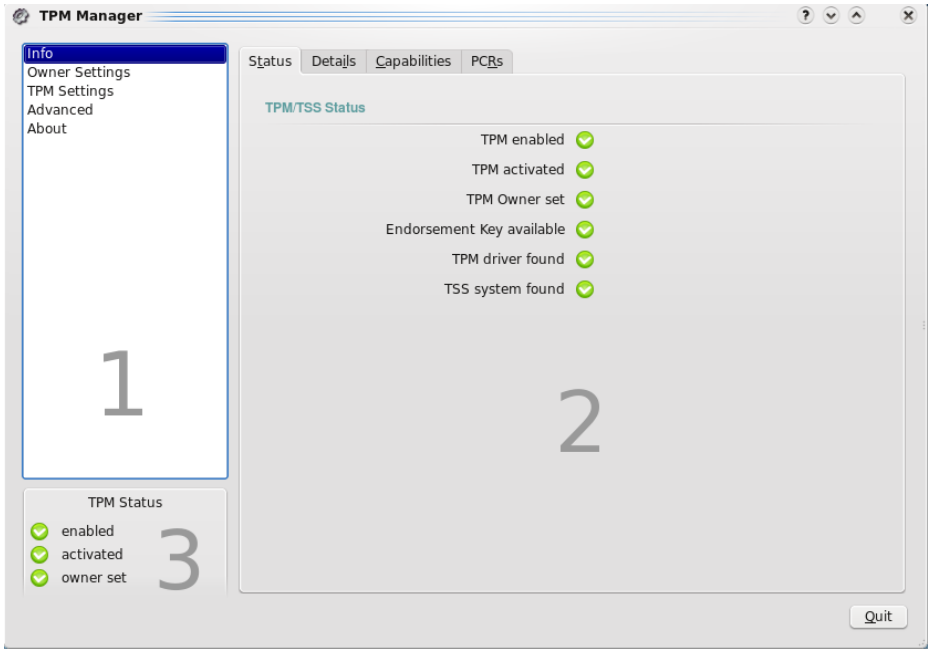


Figure 4.1: 3 parts of TPM Manager GUI

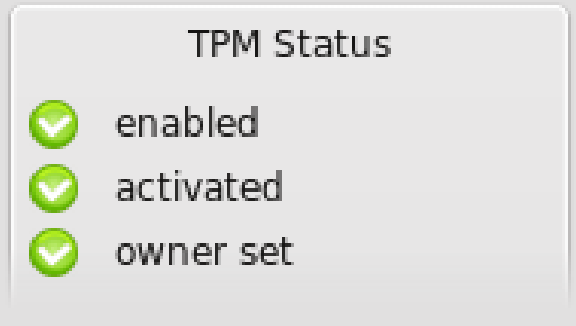


Figure 4.2: Status of the TPM

4.3.2 Info

This functional area provides some general information about the TPM and its status.

4.3.2.1 Status tab

The “Status” tab gives some general information about the status of the TPM. It shows also implicitly if you have a TPM chip on your system, by checking whether the TPM driver is loaded or not. Figure 4.3 shows the ”Status“ tab when a TPM driver but no TSS was found. Figure 4.4 shows the ”Status” tab when the TPM is disabled and no owner is set.

Figure 4.3: Status of the TPM with no running TrouSerS

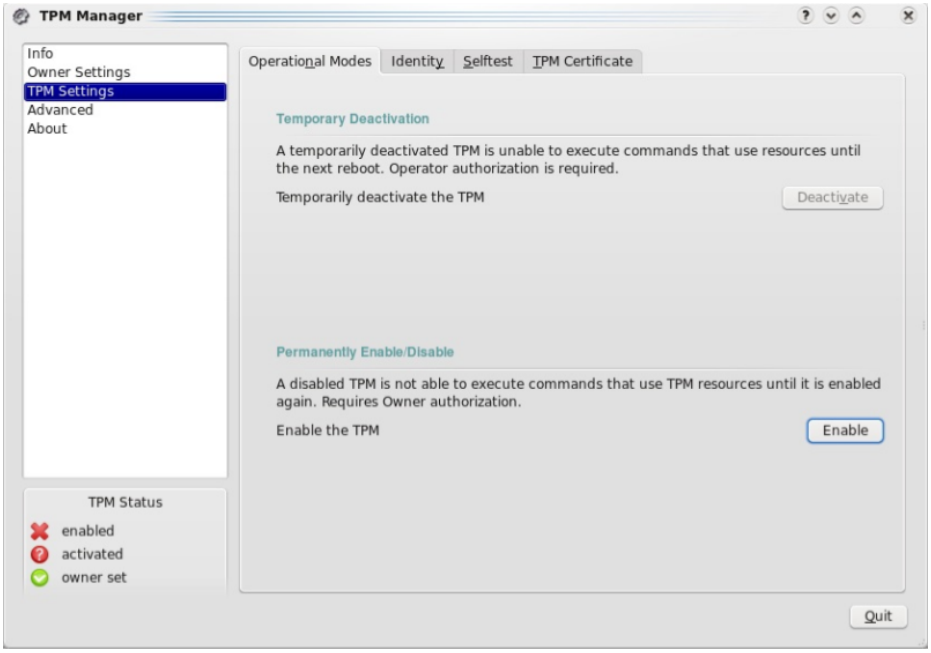


Figure 4.4: Status of a disabled TPM

4.3.2.2 Details tab

This page provides some information about the TPM version, the TPM vendor and also about the TSS.

4.3.2.3 Capabilities tab

This page shows values of some TPM capabilities.

4.3.2.4 PCRs tab

This page shows the current values of all PCRs. The view is refreshed every 1 second.

4.3.3 Owner Settings

This functional area provides an interface to manage and backup the ownership of the TPM, as discussed in the following sections.

4.3.3.1 Ownership tab

This page gives you the possibility to manage the ownership of your TPM.

You can take the ownership of your TPM using the TPM Manager doing the steps described as follows.

- Make sure that your TPM is enabled in the BIOS (See Figure 4.5).
- Choose the "Owner Settings" from the left menu, and click on the "Take" button.
- On the second dialog choose the owner password and confirm it.
- Then choose a SRK password and confirm it, or choose the default password specified by TCG (WELL_KNOWN_SECRET) (See Figure 4.6).
- Finally the TPM Manager will inform you whether taking ownership of the TPM was successfully or not(See Figure 4.7).

After doing a take ownership successfully you can use the corresponding"Change" button of the "Ownership" tab to change the owner password. You can also clear the owner using the "Clear" button on the same page.

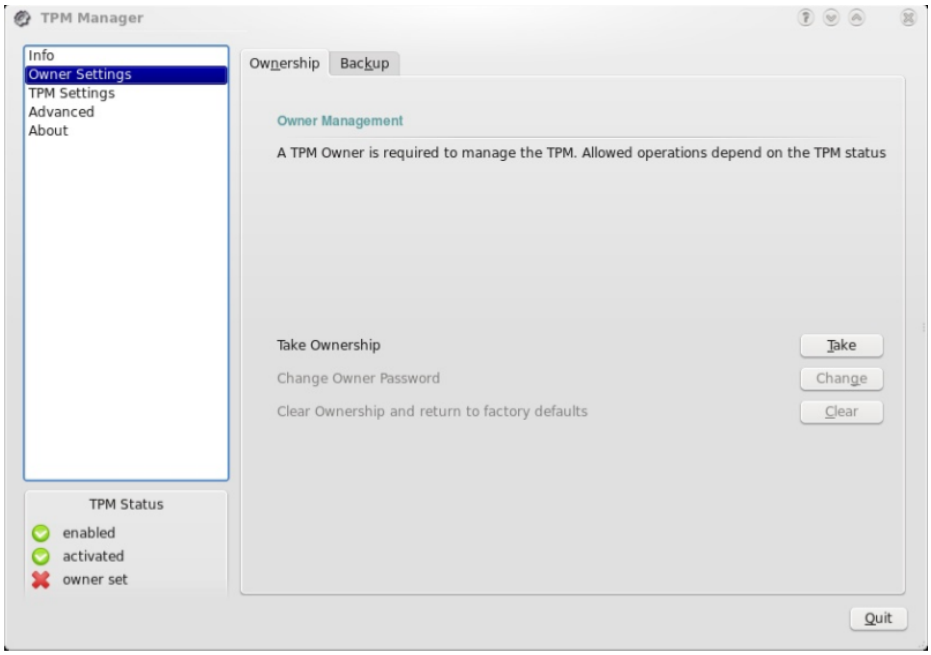


Figure 4.5: Take ownership

Figure 4.6: Dialog for setting SRK passphrase

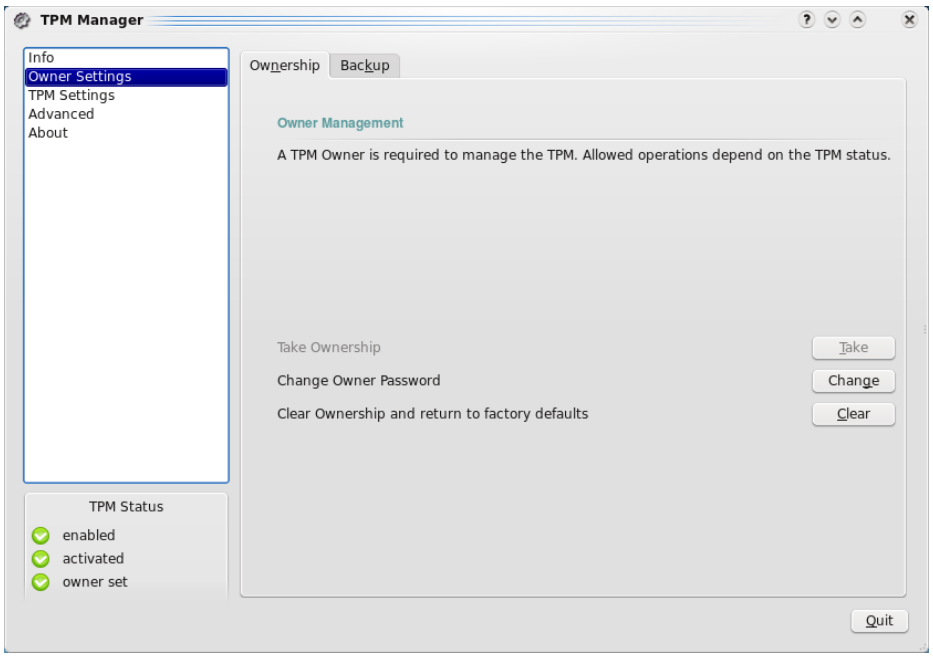


Figure 4.7: After take ownership

4.3.3.2 Backup tab

This functional area provides the possibility to create a new maintenance archive or to load an older archive into the TPM.

4.3.4 TPM Settings

This functional area provides an interface to manage the TPM settings, such as Enable/Disable the TPM on “Operational Modes” tab or manage the Endorsement Key (EK) on the “Identity” tab.

4.3.4.1 Operational Modes tab

The “Deactivate“ button provides the ability to temporarily deactivate the TPM. If the TPM has an owner the button ”Enable“/”Disable“ allows to enable/disable the TPM, using the owner password. See Figure 4.8 for a TPM status that has an owner and is enabled. Following the next steps you can disable the TPM:

- Click in the left menu on the ”TPM Settings“ entry.
- Choose the ”Operational Modes“ tab
- Press ”Disable“ button
- Enter the owner password
- Confirm the next dialog if you really want to disable the TPM
- You will see that the TPM is actually disabled. See Figure 4.9

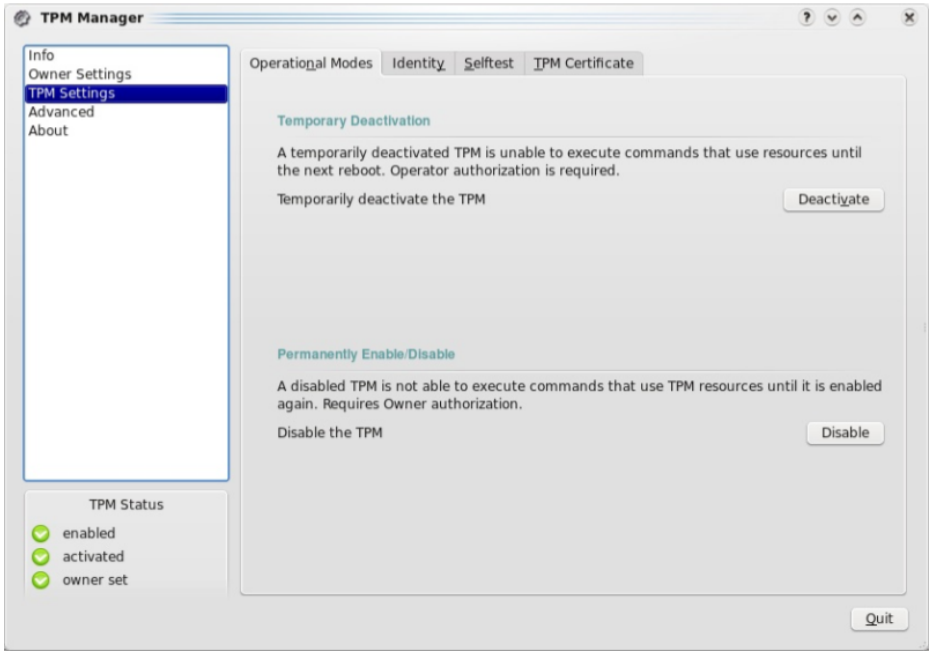


Figure 4.8: Status of an enabled TPM

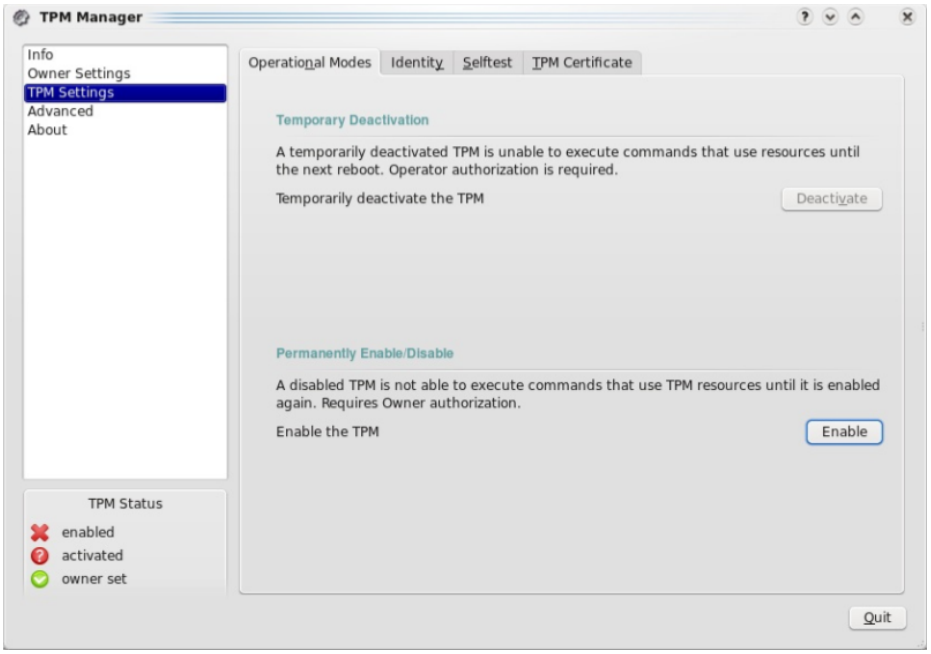


Figure 4.9: Status of the TPM after disabling

4.3.4.2 Identitiy tab

This functional area provides management functions of the EK. The functionalities "Create" and "Restrict" EK are not implemented yet. However, you can see the key information and save the public part of the EK to your PC.

4.3.4.3 Selftest tab

This tab allows you to perform the TPM selftest, a full test of each internal TPM function.

4.3.4.4 TPM Certificate tab

Not implemented yet.

4.3.5 Advanced

This section provides an interface to do some advanced TPM operations, such as disabling maintenance archive or deleting the Endorsement Key. Most functions of this section are not implemented yet.

Bibliography

[1] Trusted Computing Group. TCG software stack specification. http://trustedcomputinggroup.org, August 2003. Version 1.1.

[2] Trusted Computing Group. TCG software stack specification. http://trustedcomputinggroup.org, January 2006. Version 1.2.

[3] Trusted Computing Group. TPM main specification. Main Specification Version 1.2 rev. 103, Trusted Computing Group, July 2007.

List of Acronyms

EK	Endorsement Key
GUI	Graphical User Interface
SRK	Storage Root Key
TCG	Trusted Computing Group
TNC	Trusted Network Connect
TPM	Trusted Platform Module
TSPI	TSS Service Provider Interface
TSS	TCG Software Stack
UML	Unified Modeling Language

Glossary

Endorsement Key

An asymmetric 2048-Bit RSA-Encryption key, which is unique for every TPM. The EK resides inside the TPM permanently and can be used to authenticate a TPM and its platform.

Graphical User Interface

A type of user interface allowing people to interact with a computer or computer-controlled device employing graphical icons, visual indicators or special graphical elements called widgets as well as text, labels or text navigation to represent the information and actions available to a user.

Storage Root Key

An asymmetric 2048-Bit RSA key stored inside the TPM, which is used to encrypt TPM-internal data. The SRK is created by taking ownership of the TPM and resides permanently until the owner is cleared.

TCG Software Stack

The software stack specified by the TCG that is responsible for accessing and using the TPM.

Trusted Computing Group

An industry consortium defining several specifications required to build a trusted computing platform, incl. the TPM specification, the TSS specification and the Trusted Network Connect (TNC) specification.

Trusted Network Connect

The TNC architecture focuses on interoperability of network access control solutions and on the use of trusted computing as basis for enhancing security of those solutions. Integrity measurements are used as evidence of the security posture of the endpoint so access control solutions can evaluate the endpoint's suitability for being given access to the network.

Trusted Platform Module

A hardware device, protected against manipulation and designated for opt-in usage, providing protected capabilities and shielded locations. The TPM is a passive component and contains engines for random number generation, calculation of hash values

and RSA key generation. A TPM generates and stores keys, signs or binds data to the platform and measures the platform's current state.

TSS Service Provider Interface

The interface of the TSP.

Unified Modeling Language

A standardized specification language for the modeling of objects in the context of software engineering; includes a graphical notation used to create an abstract model of a system, referred to as a UML model.