Kanguru Defender Security Target

Version:	**1.11**
Status:	**Release**
Last Update:	**2015-01-16**
Classification:	**Public**

Legal Notice

This document is provided AS IS with no express or implied warranties. Use the information in this document at your own risk.

Revision History

Revision	Date	Author(s)	Changes to Previous Revision
1.11	2015-01-16	Andreas Siegert	Release for firmware 02.05.10, based on v1.10 of the original firmware.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Security Target Identification

Title:	Kanguru Defender Security Target
Version:	1.11
Status:	Release
Date:	2015-01-16
Sponsor:	Kanguru Solutions
Developer:	Kanguru Solutions
Certification Body:	BSI
Certification ID:	BSI-DSZ-CC-0772
Keywords:	Security Target, Common Criteria, USB Storage, Encryption, Protected Storage

1.2 TOE Identification

This ST is applicable to the following TOEs:

- Kanguru Defender Elite 200 with Kanguru Defender Manager Elite 200. Version Firmware 02.05.10, KDME200 v 2.0.0.0-2, KDME200 v 2.0.0.0-3, KDME200 v 2.0.0.0-6.
- Kanguru Defender 2000 with Kanguru Defender Manager 2000. Version Firmware 02.05.10, KDM2000 v 1.2.1.8-2, KDM2000 v 1.2.1.8-3, KDM2000 v 1.2.1.8-6.
- Universal Kanguru Local Administrator. Version 3.2.0.3.
- Kanguru Remote Management Console. Version 5.0.2.6.

1.3 TOE Type

The TOE type is Encrypted USB Storage Device.

1.4 TOE Overview

The Kanguru Defender family provides confidential storage in USB attached devices, where the storage itself is memory chip based. The device contains the necessary crypto logic on a dedicated chip(s) which provide the core of the Defender family functionality. It uses an environment provided host running Linux, MacOS or Windows to run the device specific client software (Kanguru Defender Management (KDM)) to facilitate the interaction between the TOE and the user. The client software is shipped as device specific versions, but is functionally identical.

The client software is used to engage the TOE interface for TOE unlocking and user password management as well as acting as a relay for the remote management functions.

In addition, local and remote management capabilities (initialization/device reset and password reset) for enterprise environments are provided via the Kanguru Local Administrator (KLA) and the Kanguru Remote Management Console (KRMC). The runtime environment for KLA runs is Windows XP or newer. KRMC requires an environment of Windows Server 2003 or Windows Server 2008 with MS SQL Server and IIS. When using the remote management, a protected network for the communication with the KRMC has to be provided by the environment.

Depending on the communications capabilities desired, each of the devices is available with three different versions of KDM, which are identified by the version number suffix. The Cloud version (-2), the Enterprise version (-3) or the standalone (No-Comms) version (-6). The Cloud version is treated like the standalone No-Comms version in this evaluation as the Cloud functionality is not part of the evaluated configuration, both act only locally. The enterprise version is used together with the Kanguru Remote Management Console (KRMC).

1.5 TOE Description

1.5.1 Introduction

Kanguru Defender provides protected USB mass storage. Its purpose is to protect the contents of the mass storage from unauthorized access by attackers with a basic attack potential should the locked storage device fall into the hands of unauthorized entities.

The device is evaluated at EAL2 conforming to the assurance level of the protection profile which is not geared towards highly sensitive data.

When attaching the storage device to a supported version of Linux, MacOS or Windows, client software (KDM) on the public part (CD-ROM area) of the device can be executed to unlock the device. All access control decisions for the device are implemented in the integrated security chip.

The client software is shipped under different names for the different TOE instances, but the functionality is the same for all of them.

The devices can be managed locally via the Universal Kanguru Local Administrator (KLA).

Further management support for large scale roll-outs of the TOE is provided by the KRMC, which allows a central administrator to control the devices in an enterprise environment. To prepare the TOE for use with the KRMC a standalone application is used (KLA).

A master password can be set on the device to allow the administrator access to the device data and enable password changes for the user through the administrator.

1.5.2 Architecture

Figure 1 shows the principal architecture of the storage devices.

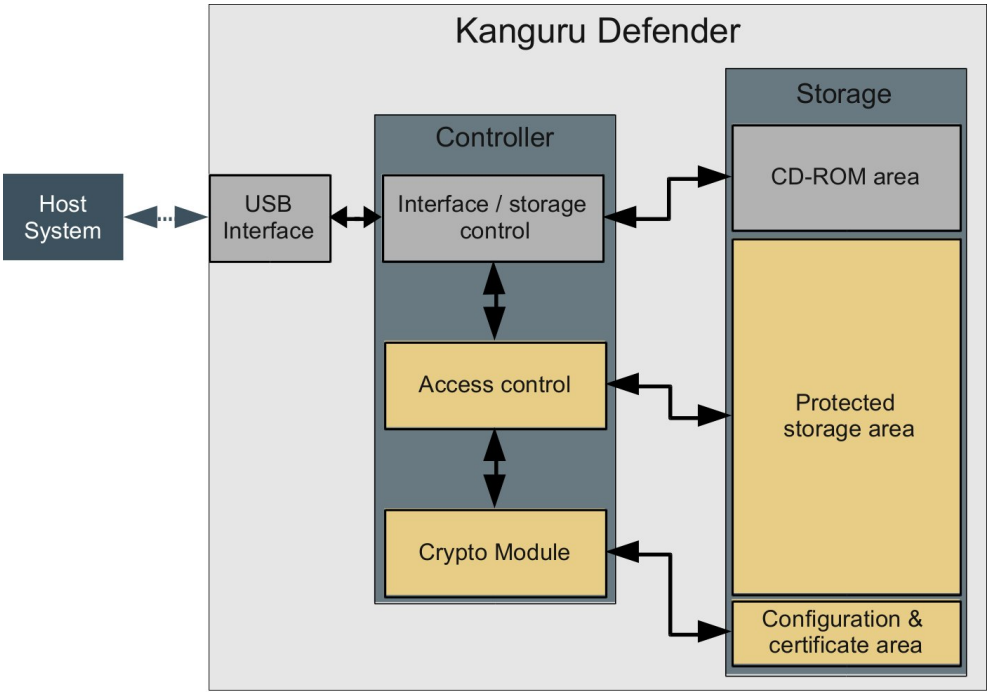

Figure 1: Device Overview

The CD-ROM area stores the local interface program that is used on the host to facilitate password entry. It is read-only and therefore provides a tamper protected version of the client software (in contrast to loading it from the host).

1.5.3 TOE boundaries

1.5.3.1 Physical

The TOE is comprised of:

- The USB device with protected and unprotected storage (CDROM area) and the crypto controller as depicted in figure 1. The following instances of the TOE are available:

  Kanguru Defender 2000
  a USB memory stick with a memory capacity of 4GB, 8GB, 16GB, 32GB, 64GB or 128GB.

  Kanguru Defender Elite 200
  a USB memory stick with a memory capacity of 4GB, 8GB, 16GB, 32GB, 64GB or 128GB.

- A Linux, MacOS or Windows based client software (KDM) to provide a GUI for the functions of the device.
- A Windows based local administration tool (KLA).
- A Windows Server 2003 or Windows Server 2008 based remote management console (KRMC) for enterprise wide management of the devices.

Please refer to chapter 1.2, TOE Identification for the version numbers of the different devices and software parts.

The TOE hardware and client software are shipped together. The documentation for KDM needs to be downloaded from the manufacturers website.

The KLA and KRMC are delivered on CD including the documentation, but the software and the documentation can also be downloaded.

Relevant guidance documents for the secure operation of the TOE are:

- Evaluated Product Guide ([ECG])
- Kanguru Defender User Manual ([USR200], [USR2000])
- KRMC Administrator User Manual and KLA Guide ([ADM], [KLA])

The following components can be found in the IT environment:

- A Linux, MacOS or Windows XP, Windows Vista or Windows 7 based host computer with a USB interface and a network connection on the host for the use of KRMC if needed for the planned usage scenarios of the TOE.
- The KRMC needs the following runtime environment:
  - Windows Server 2003 or Windows Server 2008
  - MS SQL Server
  - MS Internet Information Server

1.5.3.2 Logical

The TOE is a distributed system. The logical boundary consists of the crypto functionality of the device, the encrypted storage and the management software. A standalone device is either accessed via the client application (KDM) or the KLA.

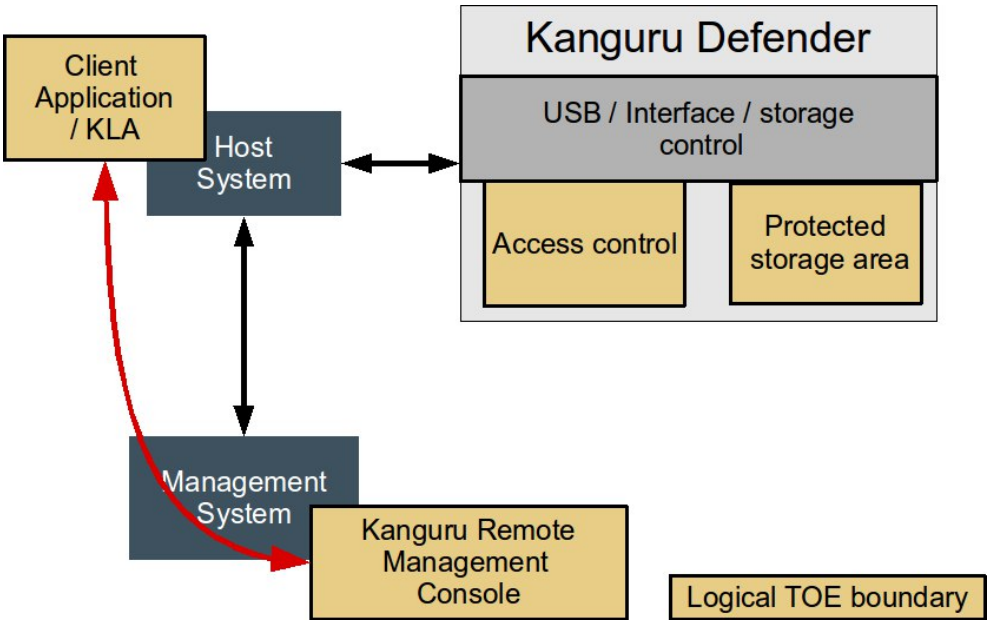

Figure 2: Logical Structure

While the device does not need the remote management console to operate (it only requires the local client software for password entry), the KRMC can be used to manage devices in an enterprise environment.

Likewise the KLA is not needed for operation but can be used to administer the device.

1.5.3.3 Evaluated configuration

The following configuration specifics apply to the evaluated configuration of the TOE:

- The devices indicated in 1.5.3.1, Physical.
- KDM client software:
  - KDME200: Kanguru Defender Manager Elite 200
  - KDM2000: Kanguru Defender Manager 2000
- KLA local management software
- KRMC
- Configuration of the TOE according to the [ECG].

Software and firmware releases as shown in 1.2, TOE Identification.

1.5.4 Security functionality

1.5.4.1 TOE security functions

The TOE provides the following security functionality:

User data protection

User data on the encrypted storage of the TOE is protected from access when the device is locked and it can only be unlocked with the user password.

TSF data protection

The keys used to encrypt the data on the device are protected against unauthorized access.

Local and Remote Management

The encryption key that is stored on the device can be generated by users or authorized administrators via a device reset which makes the protected storage unavailable, which is for all practical purposes identical to the deletion of the data.

An administrator can also trigger a reset of the user password, forcing the user to set a new one after authentication.

If a master password is set, then the administrator can change the user password via KLA or KRMC.

The administrative functions can be performed locally with KLA and remotely when KRMC is used.

The administrator needs to authenticate at KLA or KRMC before he can mange a device.

1.5.4.2 Usage scenarios

In the evaluated configuration, the Cloud and the No-Comms version are both treated as standalone, as the Cloud functionality is not part of the evaluation.

Key usage scenarios for the devices are as follows:

Standalone device

When attaching the device to a host, the client software (KDM) is available via the read-only CD partition. The user starts the client software from this partition to initiate the interaction with the device.

At first use, the device is initialized. The client software will set up the device with an initial password requested from the user. After this step, the protected storage of the device can only be accessed after entering this password or it can be completely reset (via KLA).

During the setup the device will generate the actual encryption key that is never accessible outside of the device. This key is unlocked via the user password.

The client software provides the means for the user to change the password.

Standalone device with augmented policy

Using KLA, an administrator can provision devices for users that allow additional management functions:

- A master password can be set which then allows an administrator to initialize new user passwords without the user losing access to the data.
- The device can be reset by the administrator.

Enterprise usage with remote management

The remote management via KRMC allows the same management functions as KLA but remotely.

To enable remote management, a device needs to be set up for it by using the appropriate functions of the KLA.

During the initialization of the device for the remote management, the address of the KRMC server is configured into the device. Only this server will be polled for management actions.

Once a device has been provisioned to be controlled by KRMC, the client software (KDM) will contact the KRMC any time it is run to check whether it needs to relay commands from the KRMC to the device. Commands are queued at the KRMC until polled form the device.

1.5.4.3 Out of scope

The TOE supports additional features that are not part of the scope:

- Antivirus solution
- Virtualization component
- Write protect switch
- KRMC Cloud

1.5.4.4 IT environment support

The TOE requires a version of Linux, MacOS or Windows (Windows XP, Windows Vista or Windows 7) to operate the client software. The host computer to which the TOE is attached to needs a USB (version 1 - 3) interface for the attachment.

If remote management is used, a network under the control of the TOE administrators is required.

When remote management functions with the master password are used, the management console is assumed to be located in the same protected network as the system that has the device attached and the administrator is required to check the validity of any user requests for password changes.

2 CC Conformance Claim

This Security Target is CC Part 2 extended and CC Part 3 conformant, with a claimed Evaluation Assurance Level of EAL2, augmented by ALC_FLR.1.

This Security Target claims conformance to the following Protection Profiles and PP packages, if any:

- [BSI-PP-0025]: Common Criteria Protection Profile for USB Storage Media. Version 1.4 as of 2006-03-27; demonstrable conformance.

Common Criteria [CC] version 3.1 revision 3 is the basis for this conformance claim.

2.1 Protection Profile tailoring and additions

2.1.1 PP Adjustments

[BSI-PP-0025] was written for CC 2.1. This ST is written for CC 3.1. Therefore some adjustments have to be made. The requirements for the environment (RE.Vertrau.WS and RE.Abwesend) have been dropped. The split of objectives for the IT environment and objectives for the non IT environment has been removed and all environment objectives have been stated as objectives for the operational environment.

The EAL2 SARs from CC 3.1 are used instead of the SARs from CC 2.3 to match the assurance level required by the PP.

The German labels for assumptions, objectives and threats have been translated for better readability according to the following table

PP Label	ST Label
A.Ausspähen	A.Spy
A.Vertrau.WS	A.TrustedWS
A.Abwesend	A.WSProtect
T.logZugriff	T.LogicalAccess
T.phyZugriff	T.PhysicalAccess
T.AuthÄndern	T.AuthChange
T.Störung	T.Disrupt
O.logZugriff	O.LogicalAccess
O.phyZugriff	O.PhysicalAccess
O.AuthÄndern	O.AuthChange
O.Störung	O.Disrupt
OE.Ausspähen	OE.Spy
OE.Vertrau.WS	OE.TrustedWS
OE.Abwesend	OE.WSProtect

Table 1: PP Label Translation

2.1.2 Conformance Claim Rationale

In addition to the threats, assumptions, objectives and SFRs taken from the PP, this ST contains additional threats, assumptions, objectives and SFRs to broaden the usage scenarios beyond those envisioned by the PP. They cover protection and management functions of the TOE that surpass the requirements of the PP by providing more security functionality:

- provisioning of the devices with administrator specified password quality restrictions to meed an organizations password quality policies
- administrator controlled device reset and erasure
- administrator password control to set a new user password to access the data in case a user password is lost

Apart from the added threats, assumptions and objectives, the following modifications were made to the PP objectives to keep them consistent instead of adding similar objectives:

O.AuthChange was changed to also include the administrator to include the scenario where an administrator changes the user password, which requires an administrator password.

The original A.Spy assumption was changed to exclude the biometric parts as the TOE does not support biometric authentication.

FMT_SMF.1 was extended with more management functions to include additional related functions (in contrast to adding another SFR for unrelated management functions) in FMT_SMF.1-POL. FMT_SMF.1-adm reflects the version from the PP.

FDP_ACF.1, FMT_MSA.1 and FMT_SMR.1 were extended to include the administrative functions.

The assurance requirements of [BSI-PP-0025] have been augmented with ADV_SPM.1 due to dependencies of FPT_FLS.1 and FPT_RCV.4.1. These dependencies no longer exist in CC 3.1, therefore the augmentation with ADV_SPM.1 has been dropped.

The following threats were added:

T.PhysicalTampering
This threat has been added to model tamper detection.

T.ConsoleAccess
This threat was added to include protection for the remote management.

The following assumptions were added:

A.Admin
As the TOE supports administrative functions assumption on the administrator have been added.

A.Net
The assumption is needed to reflect the usage of the remote management console in a network controlled by the users.

The following organizational security policies were added:

P.PWReset
This is used to drive the need for administrative password reset.

P.Delete
This is used to drive the need for the administrative device reset function.

The following objectives for the TOE were added to meet the threats:

O.TamperEvident
Used to specify physical tamper detection.

O.AuthAdmin
Used to authenticate administrators.

O.PWReset
Used to specify password reset by the administrator.

O.Delete
Used to specify administrative storage deletion.

The following objectives for the operational environment were added:

OE.Net
Used to require basic network environment protection so that the KRMC can be used for remote management activities.

OE.Admin
Used to require competent and trustworthy administrators.

The following SFRs were added:

FCS_RNG.1
Required by BSI for all TOEs containing a security relevant RNG.

FPT_PHP.1
Used to specify tamper detection.

FIA_UAU.2
Used for administrator authentication for KLA and KRMC.

FIA_UID.2
Used for administrator authentication for KRMC.

FMT_MTD.1
Used for administrative functions KLA and KRMC.

FMT_SMF.1-POL
Used to specify the administrative functions that cover the organizational security policies.

The FSUD defined in [BSI-PP-0025] and implemented via FDP_ACF.1 was modified to reflect that the TOE does not have a publicly writable area (it is a read only CD-ROM partition) which is more restrictive than the PP. A second augmentation to the FSUD added the reset capabilities of users and administrators for the device which does not impact the security functionality specified by the PP but is seen as a further confidentiality protection mechanism for the data on the device. A third augmentation includes administrative access which extends the usage scenarios for the device in larger environments without taking away any base security functionality required for PP conformance.

3 Security Problem Definition

3.1 TOE roles

To facilitate the description of security objectives, threats and requirements, the following role definitions are used:

Authorized user (S1)

- Holds the authentication attribute required to access the TOE's protected memory area, in which the confidential data is stored.
- Can modify the authentication attribute.

Non-authorized user (S2)

- Wishes to access S1's confidential data in the USB storage medium's memory
- Does not have the authentication attribute to access the protected data.
- Can obtain a USB storage medium of the same type. Can try out both logical and physical attacks on this USB storage medium.
- Can gain possession of the TOE relatively easily since the TOE has a compact form.

Authorized administrator (S3)

- Holds the authentication attribute for KLA or KRMC
- Holds the master authentication attribute for the device
- Can trigger various TOE actions remotely (via KRMC) or locally (via KLA):
  - Data deletion via key erasure
  - Modify or reset both master and user passwords
  - Reset of devices

3.2 Threat Environment

This section describes the threat model for the TOE and identifies the individual threats that are assumed to exist in the TOE environment.

The **assets** to be protected by the TOE are the user data in the protected storage and TSF data (encryption keys of the devices) stored on the TOE device and administrator password for the software (on KLA and KRMC).

The **threat agents** having an interest in manipulating the data model can be categorized as either:

- Unauthorized individuals ("attackers") which are unknown to the TOE and its runtime environment. (S2).

Threat agents originate from a well managed user community within an organizations internal network. Hence, only inadvertent or casual attempts to breach system security are expected from this community.

TOE administrators, including administrators of the TOE environment, are assumed to be trustworthy, trained and to follow the instructions provided to them with respect to the secure configuration and operation of the systems under their responsibility. Hence, only inadvertent attempts to manipulate the safe operation of the TOE are expected from this community.

The threats T.LogicalAccess, T.PhysicalAccess, T.AuthChange and T.Disrupt originate from [BSI-PP-0025] and are copied verbatim.

3.2.1 Threats countered by the TOE

T.LogicalAccess

Assuming that S2 gains possession of the TOE, he/she accesses the confidential data on the TOE. S2 gains logical access by, for example, connecting the TOE to the USB interface of a computer system.

Application note: *Threats arising from repeated theft and differential crypto analysis of the device are explicitly not considered.*

T.PhysicalAccess

Assuming that S2 gains possession of the TOE, he/she accesses the TOE's memory by means of a physical attack. Such an attack could take the following form, for example: S2 removes the TOE's memory and places it into another USB storage medium which he/she uses for the purpose of logical access to the memory.

T.AuthChange

Assuming that S2 gains possession of the TOE, he/she sets a new authentication attribute, with the result that the data becomes unusable for S1.

T.Disrupt

A failure (e.g. power failure or operating system error) stops the TOE operating correctly. As a result, confidential data remains unencrypted or the TOE's file system is damaged.

T.PhysicalTampering

A threat agent (S2) can access the TOE's storage by means of a physical attack, bypassing the exported TOE interfaces to obtain authentication data or cryptographic key material.

T.ConsoleAccess

Using the KRMC an attacker (S2) can attempt to modify the authentication attribute to gain access to the confidential data on the TOE.

3.3 Assumptions

3.3.1 Intended usage of the TOE

A.Spy

S1 ensures that his/her authentication attribute cannot be fraudulently obtained by, for example, someone else reading the password whilst it is being entered.

A.TrustedWS

Once S1 has unprotected the protected memory area, there are no unauthorized attempts to access the TOE from the host system or any connected networks.

A.WSProtect

If S1 leaves the host system to which the unprotected TOE is connected, he/she takes appropriate measures to protect the data whilst absent. Appropriate measures could be, for example, locking the computer with the aid of the operating system or taking the TOE with them when they leave.

A.Admin

The TOE is managed by one or more competent individuals (S3). The system administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the guidance documentation.

A.Net

The network over which the KRMC communicates with the TOE is under the same management domain as the TOE.

3.4 Organizational Security Policies

P.PWReset

The TOE must enable an administrator (S3) to trigger a password reset on the device.

P.Delete

The TOE must enable an administrator (S3) to trigger the deletion of the protected user data on the device.

4 Security Objectives

4.1 Objectives for the TOE

O.LogicalAccess

The TOE must provide a secure authentication mechanism via which only S1, following successful authentication, gains access to the protected data.

O.PhysicalAccess

The TOE encrypts all of the data in the protected area of the TOE. The encryption specifically protects confidentiality in the event of physical attacks on the TOE.

O.AuthChange

The TOE provides a function with which the authentication attribute can only be changed after S1 or S3 has successfully been authenticated.

O.Disrupt

The TOE recovers to a stable and consistent state following a failure (e.g. power failure). The failure does not result in damage to the file system, nor to data remaining unencrypted in the TOE's memory.

O.TamperEvident

The TOE provides functions to impede tampering with the physical components and to provide evidence of tampering with the physical device.

O.AuthAdmin

The TOE provides the means to securely authenticate administrators using KLA or KRMC.

O.PWReset

The user authentication attribute can be changed or reset by an authorized administrator.

O.Delete

The protected user data can be deleted by an authorized administrator.

4.2 Objectives for the Operational Environment

OE.TrustedWS

Once the protected memory area has been unprotected, the TOE cannot protect itself against unauthorized access attempts from the host system. So there are no unauthorized attempts, e.g. by malware, to access the TOE via the host system and any networks connected to it.

OE.WSProtect

Once the data has been unprotected, access to it is unrestricted while the TOE is still connected to the host system. This means that S1 has to take appropriate security measures for the duration of his/her absence from the host system in order to prevent S2 from accessing the unprotected data on the TOE.

OE.Spy

Since the TOE cannot recognise reproduced authentication attributes, S1 must ensure that it is not possible for others to see or reproduce his/her authentication attribute.

OE.Net

The TOE remote functions are designed to operate in a controlled network environment.

OE.Admin

Those responsible for the TOE are competent and trustworthy individuals, capable of managing the TOE and the security of the information it contains.

OE.TamperCheck

Users of the device check the device for evidence of physical tampering before use.

4.3 Security Objectives Rationale

4.3.1 Coverage

The following table provides a mapping of TOE objectives to threats and policies, showing that each objective counters or enforces at least one threat or policy, respectively.

Objective	Threats / OSPs
O.LogicalAccess	T.LogicalAccess T.ConsoleAccess
O.PhysicalAccess	T.PhysicalAccess
O.AuthChange	T.AuthChange
O.Disrupt	T.Disrupt
O.TamperEvident	T.PhysicalTampering
O.AuthAdmin	T.AuthChange
O.PWReset	P.PWReset
O.Delete	P.Delete

Table 2: Mapping of security objectives to threats and policies

The following table provides a mapping of the objectives for the Operational Environment to assumptions, threats and policies, showing that each objective holds, counters or enforces at least one assumption, threat or policy, respectively.

Objective	Assumptions / Threats / OSPs
OE.TrustedWS	A.TrustedWS T.LogicalAccess T.AuthChange
OE.WSProtect	A.WSProtect T.LogicalAccess T.AuthChange

Objective	Assumptions / Threats / OSPs
OE.Spy	A.Spy T.LogicalAccess T.AuthChange
OE.Net	A.Net
OE.Admin	A.Admin
OE.TamperCheck	T.PhysicalTampering

Table 3: Mapping of security objectives for the Operational Environment to assumptions, threats and policies

4.3.2 Sufficiency

The following rationale provides justification that the security objectives are suitable to counter each individual threat and that each security objective tracing back to a threat, when achieved, actually contributes to the removal, diminishing or mitigation of that threat:

Threat	Rationale for security objectives
T.LogicalAccess	**O.LogicalAccess** addresses the compensation of threat T.LogicalAccess directly by requiring an access-control method which only grants access to the protected memory area to the authorized user (S1). **OE.WSProtect** provides additional support for measures to counter threat T.LogicalAccess by requiring S1 to take appropriate security measures for the duration of his/her absence from the host system in order to protect the TOE which has been unprotected. **OE.TrustedWS** provides additional support for measures to counter threat T.LogicalAccess by ruling out unauthorized access attempts from the host system and any networks connected to it, e.g. by malware. **OE.Spy** provides additional support for measures to counter threat T.LogicalAccess by requiring S1 to ensure that it is not possible for others to see his/her authentication attribute while it is being entered or to reproduce it.
T.PhysicalAccess	**O.PhysicalAccess** addresses the compensation of threat T.PhysicalAccess directly by requiring that the data in the protected memory area be encrypted.
T.AuthChange	**O.AuthChange** addresses the compensation of threat T.AuthChange directly by requiring that modification of the authentication attribute only be possible once S1 or S3 has been authenticated. **O.AuthAdmin** O.AuthAdmin requires an administrator to be authenticated with the administrative Software (KLA or KRMC) before he can force any authentication attribute change on the device.

Threat	Rationale for security objectives
	OE.Spy provides additional support for measures to counter threat T.AuthChange by requiring S1 to ensure that it is not possible for others to see his/her authentication attribute while it is being entered or to reproduce it. **OE.WSProtect** provides additional support for measures to counter threat T.AuthChange by requiring S1 to take appropriate security measures for the duration of his/her absence from the host system in order to protect the TOE which has been unprotected. **OE.TrustedWS** addresses the compensation of threat T.AuthChange directly by requiring that modification of the authentication attribute only be possible once S1 has been authenticated. provides additional support for measures to counter threat T.AuthChange by ruling out unauthorized access attempts from the host system and any networks connected to it, e.g. by malware
T.Disrupt	**O.Disrupt** addresses the compensation of threat T.Disrupt directly by requiring that the TOE recover to a stable and consistent state following a failure (e.g. power failure) without data remaining unencrypted or the file system being damaged.
T.PhysicalTampering	**O.TamperEvident** addresses hiding of physical attacks on the TOE by sealing the TSF components. **OE.TamperCheck** addresses the responsibility of the user to check for tamper evidence.
T.ConsoleAccess	**O.LogicalAccess** restricts access to the protected memory to authenticated users without any exemption for administrators.

Table 4: Sufficiency of objectives countering threats

The following rationale provides justification that the security objectives for the environment are suitable to cover each individual assumption, that each security objective for the environment that traces back to an assumption about the environment of use of the TOE, when achieved, actually contributes to the environment achieving consistency with the assumption, and that if all security objectives for the environment that trace back to an assumption are achieved, the intended usage is supported:

Assumption	Rationale for security objectives
A.Spy	**OE.Spy** is the objective which implements the assumption. The assumption is necessary because the TOE cannot recognise reproduced authentication attributes.

Assumption	Rationale for security objectives
A.TrustedWS	**OE.TrustedWS** is the objective which implements the assumption. The assumption is necessary because the TOE cannot control access from the host system to the protected memory area once the protected area has been unprotected.
A.WSProtect	**OE.WSProtect** is the objective which implements the assumption. The assumption is necessary because, once the protected memory area has been unprotected, the TOE cannot control whether S1 or S2 accesses the data.
A.Admin	**OE.Admin** is the objective which implements the assumption. It requires trustworthy personnel managing the TOE.
A.Net	**OE.Net** is the objective which implements the assumption. The TOE is designed to be operated in a friendly networking environment when KRMC is used.

Table 5: Sufficiency of objectives holding assumptions

The following rationale provides justification that the security objectives are suitable to cover each individual organizational security policy, that each security objective that traces back to an OSP, when achieved, actually contributes to the implementation of the OSP, and that if all security objectives that trace back to an OSP are achieved, the OSP is implemented:

OSP	Rationale for security objectives
P.PWReset	**O.PWReset** The policy P.PWReset is implemented through the objective O.PWReset.
P.Delete	**O.Delete** The policy P.Delete is implemented through the objective O.Delete.

Table 6: Sufficiency of objectives enforcing Organizational Security Policies

5 Extended Components Definition

FCS_RNG.1 Generation of random numbers, requires that the random number generator implements defined security capabilities and that the random numbers meet a defined quality metric.

The definition of FCS_RNG has been supplied by BSI.

5.1 Class FCS: Cryptographic support

5.1.1 Random number generation (RNG)

Family behaviour

This family defines quality requirements for the generation of random numbers that are intended to be used for cryptographic purposes.

Component levelling

FCS_RNG.1 is not hierarchical to any other component within the FCS_RNG family.

Management: FCS_RNG.1

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

Audit: FCS_RNG.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: There are no actions defined to be auditable.

b) Basic: There are no actions defined to be auditable.

c) Detailed: There are no actions defined to be auditable.

5.1.1.1 FCS_RNG.1 - Random number generation (Class DRG.2)

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RNG.1.1 The TSF shall provide a deterministic random number generator that implements:

a) DRG.2.1: If initialized with a random seed [selection: **using a PTRNG of class PTG.2 as random source, using a PTRNG of class PTG.3 as random source , using an NPTRNG of class NTG.1 , [assignment: other requirements for seeding]**], the internal state of the RNG shall [selection: **have [assignment: amount of entropy], have [assignment: work factor], require [assignment: guess work]**].

b) DRG.2.2: The RNG provides forward secrecy.

c) DRG.2.3: The RNG provides backward secrecy.

FCS_RNG.1.2 The TSF shall provide random numbers that meet:

a) DRG.2.4: The RNG, initialized with a random seed [assignment: **requirements for seeding**], generates output for which [assignment: **number of strings**] strings of bit length 128 are mutually different with probability [assignment: **probability**].

b) DRG.2.5: Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A [assignment: **additional test suites**].

Rationale

The quality of the random number generator is defined using this SFR. The quality metric required in FCS_RNG.1.1 is detailed in the German Scheme AIS 20 and AIS 31

6 Security Requirements

6.1 Definition of the security functional policy for USB storage media

Before listing the security functional requirements for the TOE, this section first defines the security functional policy for USB storage media (FSUD), as follows:

- permitted actions by the authorized user (S1):
  - use of the authentication mechanism,
  - reading of the data in the public areas of the TOE (allow access),
  - reading/writing/modification of the data in the protected memory areas of the TOE (allow access),
  - re-authentication following a disruption in the connection between the host system and the TOE (e.g. due to physical disconnection, loss of power or an operating system error),
  - reset of a standalone device and
  - modification of the authentication attribute;
- permitted actions by the non-authorized user (S2):
  - reading of the data in the public memory area,
  - reset of a standalone device and
  - use of the authentication mechanism.
- permitted actions by the authorized administrator (S3):
  - initialization/Reset of the TOE,
  - change or reset the user password and
  - change the master password.

6.2 TOE Security Functional Requirements

The following table shows the security functional requirements for the TOE, and the operations performed on the components according to CC part 2: iteration (Iter.), refinement (Ref.), assignment (Ass.) and selection (Sel.).

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
FCS - Cryptographic support	FCS_CKM.1 Cryptographic key generation		BSI-PP-0025	No	No	Yes	No
	FCS_CKM.4 Cryptographic key destruction		BSI-PP-0025	No	No	Yes	No
	FCS_COP.1 Cryptographic operation		BSI-PP-0025	No	No	Yes	No
	FCS_RNG.1 Random number generation (Class DRG.2)		ECD	No	No	Yes	Yes
FIA - Identification and authentication	FIA_UAU.1 Timing of authentication		BSI-PP-0025	No	No	Yes	No

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FIA_UAU.6 Re-authenticating		BSI-PP-0025	No	No	Yes	No
	FIA_SOS.1 Verification of secrets		BSI-PP-0025	No	Yes	Yes	No
FDP - User data protection	FDP_ACC.1 Subset access control		BSI-PP-0025	No	No	Yes	No
	FDP_ACF.1 Security attribute based access control		BSI-PP-0025	No	No	Yes	No
FMT - Security management	FMT_MSA.1 Management of security attributes		BSI-PP-0025	No	No	Yes	Yes
FPT - Protection of the TSF	FPT_FLS.1 Failure with preservation of secure state		BSI-PP-0025	No	No	Yes	No
	FPT_RCV.4 Function recovery		BSI-PP-0025	No	Yes	Yes	No
	FPT_PHP.1 Passive detection of physical attack		CC Part 2	No	Yes	No	No
FIA - Identification and authentication	FIA_UAU.2 User authentication before any action		CC Part 2	No	No	No	No
	FIA_UID.2 User identification before any action		CC Part 2	No	No	No	No
FMT - Security management	FMT_MTD.1-dev Management of TSF data	FMT_MTD.1	CC Part 2	No	No	Yes	Yes
	FMT_MTD.1-adm Management of TSF data	FMT_MTD.1	CC Part 2	No	No	Yes	Yes
	FMT_SMF.1-POL Specification of Management Functions	FMT_SMF.1	BSI-PP-0025	Yes	No	Yes	No
	FMT_SMF.1-adm Specification of Management Functions	FMT_SMF.1	CC Part 2	No	No	Yes	No
	FMT_SMR.1 Security roles		BSI-PP-0025	No	No	Yes	No

Table 7: Security functional requirements for the TOE

6.2.1 Cryptographic support (FCS)

6.2.1.1 Cryptographic key generation (FCS_CKM.1)

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **AES key generator** and specified cryptographic key sizes **256bit** that meet the following: **no defined standard**.

Application note: *The key is basically the random number generated by the random number generator defined in FCS_RNG.1*

6.2.1.2 Cryptographic key destruction (FCS_CKM.4)

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method **overwriting of the old key with a new key** that meets the following: **no defined standard**.

6.2.1.3 Cryptographic operation (FCS_COP.1)

FCS_COP.1.1 The TSF shall perform **the encryption and decryption of the data in the TOE's protected memory area** in accordance with a specified cryptographic algorithm

a) Kanguru Defender Elite 200 and Kanguru Defender 2000: AES in CBC mode

and cryptographic key sizes **AES with 256-bit key** that meet the following: **FIPS-197, NIST SP800-38A, NIST SP800-38E**.

6.2.1.4 Random number generation (Class DRG.2) (FCS_RNG.1)

FCS_RNG.1.1 The TSF shall provide a deterministic random number generator that implements:

a) DRG.2.1: If initialized with a random seed **from the non-Approved hardware non-deterministic RNG**, the internal state of the RNG shall **have at least 40 bits of entropy**.

b) DRG.2.2: The RNG provides forward secrecy.

c) DRG.2.3: The RNG provides backward secrecy.

FCS_RNG.1.2 FCS_RNG.1.2 The TSF shall provide random numbers that meet:

a) DRG.2.4: The RNG, initialized with a random seed **of 40 bits**, generates output for which **2** strings of bit length 128 are mutually different with probability **1-2^{-8}**.

b) DRG.2.5: Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A **and no other test suite**.

Application note:

Kanguru Defender Elite 200 and Kanguru Defender 2000 use SP800-90 10.1.2 (HMAC_DRBG w/ SHA-256)

6.2.2 Identification and authentication (FIA)

6.2.2.1 Timing of authentication (FIA_UAU.1)

FIA_UAU.1.1 The TSF shall allow **access to the public memory area** on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Note: *This SFR applies to the device and the local GUI (KDM).*

6.2.2.2 Re-authenticating (FIA_UAU.6)

FIA_UAU.6.1 The TSF shall re-authenticate the user under the conditions **system crash, power failure, separation of the physical connection or another disruption to the connection**.

6.2.2.3 Verification of secrets (FIA_SOS.1)

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that ~~secrets~~*the authentication attribute* meets **the following quality metric: the probability that a secret can be obtained by an attacker is less than one in 268435456**.

Application Note: *For the calculation of the password guessing probability, NIST 800-63, table A.1 was used, assuming a user chosen password of 12 characters with composition rules as specified in the* [ECG]*, chapter 9. This results in* 2^{28} *attempts.*

Application Note: *In addition to password quality metrics for user and administrator password required in the guidance, the device implements a rate limit of 1/s for password guesses.*

6.2.2.4 User authentication before any action (FIA_UAU.2)

FIA_UAU.2.1 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Application Note: *This SFR applies to the use of KRMC and KLA.*

6.2.2.5 User identification before any action (FIA_UID.2)

FIA_UID.2.1 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

Application Note: *This SFR applies to the use of KRMC.*

6.2.3 User data protection (FDP)

6.2.3.1 Subset access control (FDP_ACC.1)

FDP_ACC.1.1 The TSF shall enforce the **FSUD** on **S1 and S3 accessing the data in the TOE's protected memory area**.

6.2.3.2 Security attribute based access control (FDP_ACF.1)

FDP_ACF.1.1 The TSF shall enforce the **FSUD** to objects based on the following:

a) Subject: S1; Object: protected memory area; Security attribute: authentication attribute

b) Subject: S3; Object: protected memory area; Security attribute: master authentication attribute

.

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

a) permitted actions by the authorized user (S1):

1. use of the authentication mechanism,

2. reading of the data in the public areas of the TOE (allow access),

3. **reading/writing/modification of the data in the protected memory areas of the TOE (allow access),**
4. **re-authentication following a disruption in the connection between the host system and the TOE (e.g. due to physical disconnection, loss of power or an operating system error),**
5. **reset of a standalone device and**
6. **modification of the authentication attribute;**

b) permitted actions by the non-authorized user (S2):

1. **reading of the data in the public memory area,**
2. **reset of a standalone device and**
3. **use of the authentication mechanism.**

c) permitted actions by the authorized administrator (S3):

1. **Initialization/Reset of the TOE**
2. **Change the master password**
3. **Change or reset the user password (when a master password is set)**

.

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **no rules**.

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **no rules**.

Application note: *Even the PP wording specifies writing and modification of public areas, the TOE does not allow any form of write access or modification of the public storage to protect the integrity of the client.*

6.2.4 Security management (FMT)

6.2.4.1 Management of security attributes (FMT_MSA.1)

FMT_MSA.1.1 The TSF shall enforce the **FSUD** to restrict the ability to **modify, no other operations** the security attribute **authentication attribute** to **S1 and to the authorized administrator S3**.

Application Note: *The administrator (S3) can only set the user (S1) password when the master password has been set.*

6.2.4.2 Management of TSF data (FMT_MTD.1-dev)

FMT_MTD.1.1 The TSF shall restrict the ability to

a) set

the **user authentication attribute** to **the**

a) authorized user

b) authorized administrator

.

Application Note:

Users can change their password via the local client application. Administrators can only change user password when a master password is set, without the master password they can only request the user to change the password (password reset).

6.2.4.3 Management of TSF data (FMT_MTD.1-adm)

FMT_MTD.1.1 The TSF shall restrict the ability to **modify** the **administrator password** to **the authorized administrator**.

Application note: *This SFR applies to the authentication attribute of the administrator using KLA or KRMC.*

6.2.4.4 Specification of Management Functions (FMT_SMF.1-POL)

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions:

a) modification of the user or administrator authentication attribute

b) trigger the reset of the user authentication attribute

c) deletion of the protected data

Application note: *Deletion of the protected data is not implemented through deletion of the data, but by deletion of the key used to encrypt the data, thereby rendering the data inaccessible.*

6.2.4.5 Specification of Management Functions (FMT_SMF.1-adm)

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: **change of the administrator password**.

Application note: *This SFR applies to the authentication attribute of the administrator using KLA or KRMC.*

6.2.4.6 Security roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles **authorized user S1 and authorized administrator S3**.

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

6.2.5 Protection of the TSF (FPT)

6.2.5.1 Failure with preservation of secure state (FPT_FLS.1)

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: **system crash, power failure, separation of the physical connection or another disruption to the connection**.

6.2.5.2 Function recovery (FPT_RCV.4)

FPT_RCV.4.1 The TSF shall ensure that **the user and TSF data protection** have the property that ~~the function either completes successfully, or for the indicated failure scenarios, recovers to a consistent and secure state.~~*in the event of a system crash, power failure, separation of the physical connection or another failure the SF either completes successfully, or for the indicated failure scenarios, recovers to a consistent and secure state.*

6.2.5.3 Passive detection of physical attack (FPT_PHP.1)

FPT_PHP.1.1 The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF *by a sealing encapsulation of the components*.

FPT_PHP.1.2 The TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.

6.3 Security Functional Requirements Rationale

6.3.1 Coverage

The following table provides a mapping of SFR to the security objectives, showing that each security functional requirement addresses at least one security objective.

Security functional requirements	Objectives
FCS_CKM.1	O.PhysicalAccess
FCS_CKM.4	O.PhysicalAccess
FCS_COP.1	O.PhysicalAccess
FCS_RNG.1	O.PhysicalAccess
FIA_UAU.1	O.AuthChange, O.LogicalAccess
FIA_UAU.6	O.AuthChange, O.LogicalAccess
FIA_SOS.1	O.AuthChange, O.LogicalAccess
FDP_ACC.1	O.AuthChange, O.LogicalAccess
FDP_ACF.1	O.AuthChange, O.LogicalAccess
FMT_MSA.1	O.AuthChange
FPT_FLS.1	O.Disrupt
FPT_RCV.4	O.Disrupt
FPT_PHP.1	O.TamperEvident
FIA_UAU.2	O.AuthAdmin, O.AuthChange
FIA_UID.2	O.AuthAdmin
FMT_MTD.1-dev	O.AuthChange
FMT_MTD.1-adm	O.AuthChange
FMT_SMF.1-POL	O.AuthChange, O.Delete, O.PWReset
FMT_SMF.1-adm	O.AuthChange
FMT_SMR.1	O.AuthChange, O.LogicalAccess

Table 8: Mapping of security functional requirements to security objectives

6.3.2 Sufficiency

The following rationale provides justification for each security objective for the TOE, showing that the security functional requirements are suitable to meet and achieve the security objectives:

Security objectives	Rationale
O.LogicalAccess	Component FIA_UAU.1, which requires user authentication before any TSF-mediated action, apart from access to the public memory area, addresses security objective O.LogicalAccess, which specifies that logical access to the TOE must be controlled. Component FIA_UAU.6, which requires re-authentication after any disruption to the connection, addresses security objectives O.LogicalAccess and O.AuthChange. It is not possible to avoid the access-control security mechanism by disrupting the connection. Component FIA_SOS.1 requires an authentication mechanism which ensures that the authentication attribute is hard to guess. This addresses the O.LogicalAccess and O.AuthChange security objectives. Component FDP_ACC.1 requires subset access control and thus addresses security objectives O.LogicalAccess and O.AuthChange by requiring controlled access by S1 to the TOE's resources apart from its public memory area. Component FDP_ACF.1 requires certain rules for user-defined access control and thus addresses security objectives O.LogicalAccess and O.AuthChange by specifying controlled user access, implicitly governed by certain rules, to the TOE's resources and functions. Component FMT_SMR.1 requires the role of authorized user (S1). This is necessary in order to be able to use authentication data for user-defined access control (see FDP_ACF.1 and FDP_ACC.1). The component therefore supports the O.LogicalAccess and O.AuthChange security objectives.
O.PhysicalAccess	Components FCS_CKM.1, FCS_CKM.4, FCS_COP.1 and FCS_RNG.1 are necessary in order to encrypt the data in the protected memory area. As such, they address security objective O.PhysicalAccess.
O.AuthChange	Component FIA_UAU.1, which requires user authentication before any TSF-mediated action, apart from access to the public memory area, addresses security objective O.AuthChange which is only possible after authentication. Component FIA_UAU.6, which requires re-authentication after any disruption to the connection, addresses security objectives O.LogicalAccess and O.AuthChange. It is not possible to avoid the access-control security mechanism by disrupting the connection. Component FIA_SOS.1 requires an authentication mechanism which ensures that the authentication attribute has a certain strength. This addresses the O.LogicalAccess and O.AuthChange security objectives. Component FDP_ACC.1 requires subset access control and thus addresses security objectives O.LogicalAccess and O.AuthChange by requiring controlled access by S1 to the TOE's resources apart from its public memory area. Component FDP_ACF.1 requires certain rules for user-defined access control and thus addresses security objectives O.LogicalAccess and O.AuthChange by specifying controlled user access, implicitly governed by certain rules, to the TOE's resources and functions.

Security objectives	Rationale
	Component FMT_SMF.1-POL requires a function for modifying the authentication attribute and thus addresses the O.AuthChange security objective. Component FMT_SMF.1-adm requires a function for modifying the administrator password used for KLA and KRMC and thus addresses the O.AuthChange security objective. Component FMT_SMR.1 requires the role of authorized user (S1). This is necessary in order to be able to use authentication data for user-defined access control (see FDP_ACF.1 and FDP_ACC.1). The component therefore supports the O.LogicalAccess and O.AuthChange security objectives. Component FMT_MSA.1 requires that management of the authentication attribute only be possible for the authorized user (S1). The component therefore addresses security objective O.AuthChange. Component FMT_MTD.1-dev restricts the ability to change the authentication attribute for the device to the authorized user (S1) and the authorized administrator (S3). The component therefore addresses security objective O.AuthChange. Component FMT_MTD.1-adm restricts the ability to change the administrator password for administrator software (KLM and KRMC) the authorized administrator (S3). The component therefore addresses security objective O.AuthChange.
O.Disrupt	Component FPT_FLS.1 requires the TSF to remain secure in the event of a failure (e.g. power failure). The component therefore addresses the O.Disrupt security objective. Component FPT_RCV.4 requires the TSF to ensure that, following a failure (e.g. power failure), the security function either completes successfully or recovers to a consistent and secure state. The component therefore addresses security objective O.Disrupt.
O.TamperEvident	The TOE is protected by sealing the TSF components in epoxy according to FPT_PHP.1.
O.AuthAdmin	The administrator using KLA or KRMC needs to be authenticated (FIA_UAU.2, FIA_UID.2) before being able to perform any actions on the TSF.
O.PWReset	The local and remote management consoles provides the means to reset the authentication attribute via FMT_SMF.1-POL.
O.Delete	The remote management console provides the means to delete the TOE protected data via FMT_SMF.1-POL.

Table 9: Security objectives for the TOE rationale

6.3.3 Security Requirements Dependency Analysis

Dependencies within the EAL2 package selected for the security assurance requirements have been considered by the authors of CC Part 3 and are not analyzed here again. The included component on flaw remediation, ALC_FLR.1, has no dependencies on other requirements.

The security functional requirements in this Security Target do not introduce dependencies on any security assurance requirement; neither do the security assurance requirements in this Security Target introduce dependencies on any security functional requirement.

The following table demonstrates the dependencies of SFRs modeled in CC Part 2 and how the SFRs for the TOE resolve those dependencies:

Security functional requirement	Dependencies	Resolution
FCS_CKM.1	[FCS_CKM.2 or FCS_COP.1]	FCS_COP.1
	FCS_CKM.4	FCS_CKM.4
FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1
FCS_COP.1	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1
	FCS_CKM.4	FCS_CKM.4
FCS_RNG.1	No dependencies.	
FIA_UAU.1	FIA_UID.1	Not resolved as the device does not need user IDs. Only authentication is required, since users need not be identified but the roles S1, S2 or S3.
FIA_UAU.6	No dependencies.	
FIA_SOS.1	No dependencies.	
FDP_ACC.1	FDP_ACF.1	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1	FDP_ACC.1
	FMT_MSA.3	Static attribute initialization is not needed for the TOE as the access control mechanism is not based on user or object attributes but on the state of the machine (locked or unlocked depending on the success of the authentication).
FMT_MSA.1	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1-POL
FPT_FLS.1	No dependencies.	
FPT_RCV.4	No dependencies.	
FPT_PHP.1	No dependencies.	
FIA_UAU.2	FIA_UID.1	FIA_UID.2
FIA_UID.2	No dependencies.	
FMT_MTD.1-dev	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1-POL

Security functional requirement	Dependencies	Resolution
FMT_MTD.1-adm	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1-adm
FMT_SMF.1-POL	No dependencies.	
FMT_SMF.1-adm	No dependencies.	
FMT_SMR.1	FIA_UID.1	Not resolved as the device does not need user IDs. Only authentication is required, since users need not be identified but only be assigned to the roles S1 or S3. The assignment is implicit in the way the roles access the TOE.

Table 10: TOE SFR dependency analysis

6.4 Security Assurance Requirements

The security assurance requirements for the TOE are the Evaluation Assurance Level 2 components as specified in [CC] part 3, augmented by ALC_FLR.1.

The following table shows the Security assurance requirements, and the operations performed on the components according to CC part 3: iteration (Iter.), refinement (Ref.), assignment (Ass.) and selection (Sel.).

Security assurance class	Security assurance requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
ADV Development	ADV_ARC.1 Security architecture description	CC Part 3	No	No	No	No
	ADV_FSP.2 Security-enforcing functional specification	CC Part 3	No	No	No	No
	ADV_TDS.1 Basic design	CC Part 3	No	No	No	No
AGD Guidance documents	AGD_OPE.1 Operational user guidance	CC Part 3	No	No	No	No
	AGD_PRE.1 Preparative procedures	CC Part 3	No	No	No	No
ALC Life-cycle support	ALC_CMC.2 Use of a CM system	CC Part 3	No	No	No	No
	ALC_CMS.2 Parts of the TOE CM coverage	CC Part 3	No	No	No	No
	ALC_DEL.1 Delivery procedures	CC Part 3	No	No	No	No
	ALC_FLR.1 Basic flaw remediation	CC Part 3	No	No	No	No
ASE Security Target evaluation	ASE_INT.1 ST introduction	CC Part 3	No	No	No	No
	ASE_CCL.1 Conformance claims	CC Part 3	No	No	No	No
	ASE_SPD.1 Security problem definition	CC Part 3	No	No	No	No
	ASE_OBJ.2 Security objectives	CC Part 3	No	No	No	No

Security assurance class	Security assurance requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
	ASE_ECD.1 Extended components definition	CC Part 3	No	No	No	No
	ASE_REQ.2 Derived security requirements	CC Part 3	No	No	No	No
	ASE_TSS.1 TOE summary specification	CC Part 3	No	No	No	No
ATE Tests	ATE_COV.1 Evidence of coverage	CC Part 3	No	No	No	No
	ATE_FUN.1 Functional testing	CC Part 3	No	No	No	No
	ATE_IND.2 Independent testing - sample	CC Part 3	No	No	No	No
AVA Vulnerability assessment	AVA_VAN.2 Vulnerability analysis	CC Part 3	No	No	No	No

Table 11: Security assurance requirements

6.5 Security Assurance Requirements Rationale

There are no security assurance components defined in this ST apart from the ones taken from CC Part 3. The SARs were chosen fit the requirements from the PP which this ST claims conformance to.

In addition, the evaluation assurance level has been augmented with ALC_FLR.1 commensurate with the flaw remediation capabilities offered by the developer beyond those required by the evaluation assurance level.

7 TOE Summary Specification

7.1 TOE Security Functionality

The TOE provides the following security functionality:

7.1.1 User data protection

User data on the encrypted storage of the TOE is protected from unauthorized access when the device is not in an unlocked state.

The TOE achieves this by storing all user data in encrypted form using a unique key (created via the random number generator on the device at device initialization) that is stored only on the crypto chip of the TOE. Only when unlocked via the user or master authentication attribute (the password) the data can be accessed and decrypted (the encrypted data is never accessible).

When a master password (which can be set and modified by the administrator) is set, it can be used to set a new user (or master) password and to access the protected data.

If the physical or logical connection of the unlocked device is broken, authentication is required again.

The password quality has to be set according to the [ECG] chapter 9.

Brute force attacks from the client are mitigated by rate limiting the password attempts to 1/s on the device.

This security functionality addresses FIA_UAU.1, FIA_UAU.6, FIA_SOS.1, FDP_ACC.1, FDP_ACF.1, FCS_CKM.1, FCS_CKM.4, FCS_COP.1, FCS_RNG.1, FPT_FLS.1, FPT_RCV.4.

7.1.2 TSF data protection

The encryption key for the protected data that is stored on the device is protected against unauthorized access within the crypto chip that has no interface to extract the key leaving only the potential of physical attacks.

Attacks are blocked by the resilient nature of the TOE that always returns to a locked state in case of disruptions and by coating the TSF enforcing chips in Epoxy to make tamper detection easy.

This security functionality addresses FPT_FLS.1, FPT_RCV.4, FPT_PHP.1.

7.1.3 Management

Users can change the authentication attribute via the client software on the public device CDROM partition.

Administrators can force the user to change the password (password reset) or set a new password for the user (if the master password is set).

If a master password is to be used, it needs to be set by the administrator before the user password is set by the user. If a master password is set after a user password has been set, the user password will be wiped which is equivalent to deleting the protected storage.

A managed device can be explicitly reset by the administrator resulting in the deletion of the encrypted data by overwriting the encryption key with a new one. Devices can also be explicitly reset by anyone with physical access.

Local administration can be performed by KLA, remote management via the KRMC.

An administrator can perform all operations remotely via KRMC. The regular client (KDM) is used to rely commands from the KRMC to the device.

To use the KLA, an administrator first needs to authenticate via password.

To use the KRMC, an administrator first needs to authenticate via User ID and password.

The administrator can change his password at KLA or KRMC if needed.

This security functionality addresses FDP_ACC.1, FDP_ACF.1, FMT_MSA.1, FMT_SMF.1-POL, FMT_SMF.1-adm, FMT_SMR.1, FIA_UAU.2, FIA_UID.2, FIA_UAU.2, FIA_UID.2, FMT_MTD.1-dev, FMT_MTD.1-adm, FMT_SMF.1-POL.

8 Abbreviations, Terminology and References

8.1 Abbreviations

FSUD
Security Functional Policy for USB Storage Media

KDM
Kanguru Defender Manager, used as a generic abbreviation for the different instances KDM200 and KDM2000

KDM200
Kanguru Defender Manager 200

KDM2000
Kanguru Defender Manager 2000

KLA
Universal Kanguru Local Administrator

KRMC
Kanguru Remote Management Console

8.2 Terminology

This section contains definitions of technical terms that are used with a meaning specific to this document. Terms defined in the [CC] are not reiterated here, unless stated otherwise.

User
Humans or machines interacting with the TOE via the provided user and programmatic interfaces. The term user in this document includes administrators of the TOE unless a specific distinction is made in the text.

8.3 References

ADM	**Administrator's User Manual**	
	Version	5.0.2
	Date	2013-01-11
BSI-PP-0025	**Common Criteria Protection Profile for USB Storage Media**	
	Version	1.4
	Date	27.03.2006
CC	**Common Criteria for Information Technology Security Evaluation**	
	Version	3.1R3
	Date	July 2009
	Location	http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R3.pdf
	Location	http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R3.pdf
	Location	http://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R3.pdf
ECG	**Evaluated Product User Guide**	
	Version	1.11
	Date received	2014-07-24

KLA	**Kanguru Local Administrator Version 3.2 User Manual**	
	Version	3.2.1
	Date	2014-03-04
USR200	**Kanguru Defender Elite 200 User Manual**	
	Date	2013-12-11
USR2000	**Kanguru Defender 2000 User Guide**	
	Version	1.1.3
	Date	2014-03-25