kaspersky

Kaspersky Endpoint Security for Windows

(version 11.6.0.394 AES256)

Security Target Lite

Document version: 2.04

26.11.2021

Document history

Date	Version	Author	Change
February 2021	2.00	Alexander Testov	First draft for KES 11.6.
July 2021	2.01	Alexander Testov	Corrections due to observation reports.
September 2021	2.02	Alexander Testov	Corrections due to observation reports.
October 2021	2.03	Alexander Testov	Corrections due to observation reports.
November 2021	2.04	Alexander Testov	Corrections due to observation reports.

Table of Contents

Terminology

This Security Target refers to the terms, definitions and abbreviations of Sections 4 and 5 of [CCp1].

Additionally, the following terms and abbreviations, most of which specific to Kaspersky products, shall be defined.

Term	Definition
AV	Anti-Virus Software
AES	Advanced Encryption Standard
BIOS	Basic Input/Output System
ECB	Electronic Code Book
EME	Encoding Method Encryption
FDE	Full Disk Encryption
HMAC	Keyed-Hash Message Authentication Code
KES	Kaspersky Endpoint Security
KSC	Kaspersky Security Center
LAN	Local Area Network
OAEP	Optimal Asymmetric Encryption Padding
PBKDF2	Password-Based Key Derivation Function 2
PP	Protection Profile
RSA	Rivest, Shamir and Adleman algorithm
SHA	Secure Hash Algorithm
TLS	Transport Layer Security
Token	Secure device (smart card/Integrated Circuit Card) able to perform RSA encryption with private key
UEFI	Unified Extensible Firmware Interface
XTS	Xor-encrypt-xor-based tweaked-codebook mode with ciphertext stealing
Malware	Malicious software that is specifically designed to disrupt, damage, or gain unauthorised access to a computer system and disrupt of information confidentiality, integrity and availability.
Virus	While technically computer virus is just a type of malware—specifically replicating file infector—this term, however, is often used to describe malware in general. In this document ‘virus’ means ‘malware’.
Known viruses	In this document ‘known viruses’ refer to the list of malware known to Kaspersky Lab, specifically those included in [UGDAA].

The referenced documents are listed in Chapter 8. The text does not refer to the document sources because it is always clear which reference is meant.

1 ST Introduction

This document is a sanitised version of the Security Target used for the evaluation. It is a real representation of the complete Security Target as referenced below, with removal of confidential proprietary technical information.

1.1 ST Reference

Title	Kaspersky Endpoint Security for Windows. Security Target Lite
Sponsor	AO Kaspersky Lab
Author(s)	Oleg Andrianov, AO Kaspersky Lab Alexander Testov, AO Kaspersky Lab
ST Version	2.04
ST Publication Date	26 November 2021
CC Version	Version 3.1, Revision 5
Assurance Level	EAL2+ (EAL2 augmented with ALC_FLR.1)
Keywords	Full Disk Encryption, Anti-Virus protection, Application Startup Control, Device Control, Web Control

1.2 TOE Reference

The target of evaluation (TOE) in this ST is **Kaspersky Endpoint Security for Windows (version 11.6.0.394 AES256)** developed by AO Kaspersky Lab.

1.3 TOE Overview

1.3.1 TOE Definition and Operational Usage

The TOE is **Kaspersky Endpoint Security for Windows** (also referred to as **KES**), a software product that provides wide range of cybersecurity functionality for the endpoint devices, such as encryption of device data (user data, operating system data), anti-virus, and access control. Together with the Kaspersky Security Center (KSC), a centralised management console, KES builds a cybersecurity suite for protection of personal computer systems (work stations, laptops and other devices) using Windows as operating system.

Kaspersky Endpoint Security (KES)

KES combines world-class anti-malware with application startup control, device access control, and web access control, plus data encryption in a single application.

The Full Disk Encryption functionality as part of KES helps to protect valuable business data from accidental loss due to lost or stolen devices. Kaspersky understands that data loss can result in devastating consequences. The Full Disk Encryption functionality provides a strong encryption algorithm integrated in the endpoint protection suite that can be easily managed with a centralised management console.

KES consists of several components, each of which is responsible for protection against a particular type of threat. They can be organised into four groups covering main product functionality:

1. Anti-Virus protection:
    a. File system protection
    b. Network protection and traffic scanning
    c. Proactive Defence
2. Controls:
    a. Application Startup Control
    b. Device Access Control
    c. Web Access Control
3. Encryption:
    a. Full Disk Encryption
    b. Removable Device Encryption (not a part of evaluation)
    c. File Level Encryption (not a part of evaluation)
4. Management of all above, including user identification and authentication.

The overview of the physical architecture is given on the following picture (Fig.1).

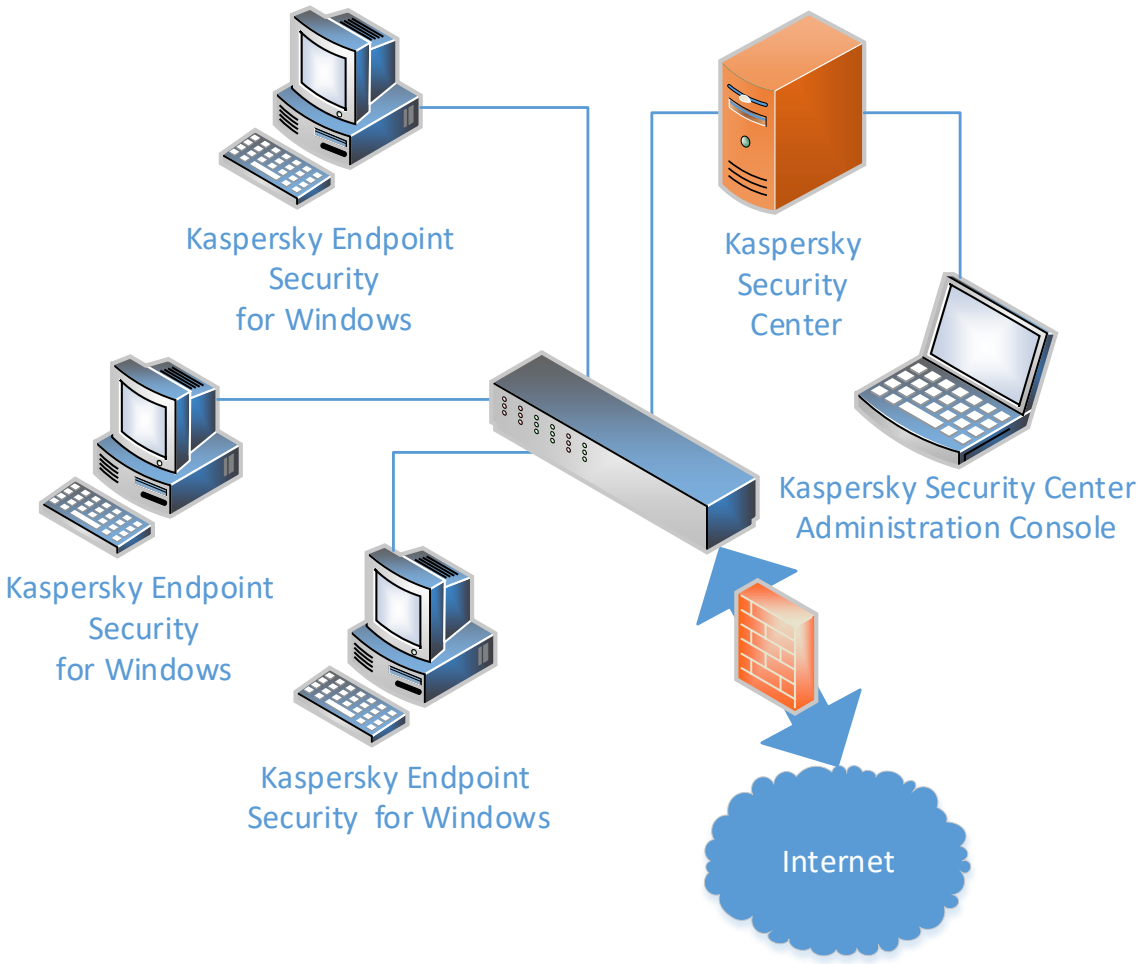

Figure 1. Physical architecture

1.3.2 Required Non-TOE hardware/software/firmware

Hardware and software requirements

To ensure proper operation of Kaspersky Endpoint Security, the computer must meet the following requirements:

Minimum general requirements:

- 2 GB free disk space on the hard drive
- CPU:
  - Workstation: 1 GHz
  - Server: 1.4 GHz
  - Support for the SSE2 instruction set
- RAM:
  - Workstation (x86): 1 GB
  - Workstation (x64): 2 GB
  - Server: 2 GB
- Microsoft .NET Framework 4.0 or later.

Supported operating systems for workstations:

- Windows 7 Home / Professional / Ultimate / Enterprise Service Pack 1 or later;
- Windows 8 Professional / Enterprise;
- Windows 8.1 Professional / Enterprise;
- Windows 10 Home / Pro / Education / Enterprise.
  NOTE. The SHA-1 module signature algorithm is deprecated by Microsoft. Update KB4474419 is required for successful installation of Kaspersky Endpoint Security on a computer running the Microsoft Windows 7 operating system. For more details about this update, visit the Microsoft technical support website.
  NOTE. For details about support for the Microsoft Windows 10 operating system, please refer to the Technical Support Knowledge Base.

Supported operating systems for servers:

- Windows Small Business Server 2011 Essentials / Standard (64-bit);
  NOTE. Microsoft Small Business Server 2011 Standard (64-bit) is supported only if Service Pack 1 for Microsoft Windows Server 2008 R2 is installed
- Windows MultiPoint Server 2011 (64-bit);
- Windows Server 2008 R2 Foundation / Standard / Enterprise / Datacenter Service Pack 1 or later;
- Windows Server 2012 Foundation / Essentials / Standard / Datacenter;
- Windows Server 2012 R2 Foundation / Essentials / Standard / Datacenter;
- Windows Server 2016 Essentials / Standard / Datacenter;
- Windows Server 2019 Essentials / Standard / Datacenter.
  NOTE. The SHA-1 module signature algorithm is deprecated by Microsoft. Update KB4474419 is required for successful installation of Kaspersky Endpoint Security on a computer running the Microsoft Windows Server 2008 R2 operating system. For more details about this update, visit the Microsoft technical support website.
  NOTE. For details about support for the Microsoft Windows Server 2016 and Microsoft Windows Server 2019 operating systems, please refer to the Technical Support Knowledge Base.

Supported virtual platforms:

- VMWare Workstation 15.5.2 Pro
- VMWare ESXi 7.0, Patch Release ESXi 7.0b
- Microsoft Hyper-V 2019 Server
- Citrix Virtual Apps and Desktops 7 2009
- Citrix Provisioning 2009
- Citrix Hypervisor 8.2 LTSR

Kaspersky Endpoint Security supports operation with the following versions of Kaspersky Security Center:

- Kaspersky Security Center 11

- Kaspersky Security Center 12
- Kaspersky Security Center 12 Patch A
- Kaspersky Security Center 12 Patch B
- Kaspersky Security Center 13

Kaspersky Security Center (KSC) – Not a part of the TOE

Every feature within KES can be set up and managed via Kaspersky Security Center – centralised management console.

Kaspersky Security Center has been developed with the sole purpose of making it easier and quicker to configure, run and manage your IT security and systems – across a complex IT environment. It provides a single, unified management console that can control all of the Kaspersky security and systems management technology.

Kaspersky Security Center enables the installation, configuration and management of Kaspersky's endpoint security technologies.

Kaspersky Security Center allows administrators to create various policies that will enable necessary protection functionality, set parameters, schedules and notifications for all protection components. Single-policy management allows creation of encryption, anti-malware, device control, application startup control and other endpoint security settings within a single policy:

- Scan for viruses – on demand or by chosen schedule
- Manage updates of anti-virus databases
- Manage cloud-assisted protection from the Kaspersky Security Network (KSN)
- Configure and manage Kaspersky's advanced firewall and Host-based Intrusion Prevention System
- Set your Application Startup Control policies
- Manage access privileges for devices that users attach to network – according to the type of device, the bus or the device's individual serial number
- Monitor and control web access and privileges – including segmentation of user groups
- Centrally manage all encryption settings and policies to encrypt data on:
    - Hard disks – file / folder encryption or full disk encryption
    - Removable devices – file / folder or full disk encryption

Kaspersky Security Center consists of server component that stores, process and issue data, and administration console, that can be connected.

For the indicated Windows operating systems, the TOE is compatible with all application software released for these Windows operating systems. If application software is not using the respective Application Programming Interface of the Windows OS for disk access, it will not be able to interpret the encrypted data reading directly from disk. Such software may also write plain text data directly onto a protected device. Then this data could not be protected by the TOE against unauthorised disclosure. However, such software is not used, except for special hard disk operations such as repair and copy functions. It is not recommended to use such software for hard disk operations together with the TOE.

The device, which is secured by the TOE as a part of Kaspersky Endpoint Security, must have a network connection to an administration server. This is usually the Kaspersky Security Center, which stores and manages the security policies and administration/configuration data for each KES. The connection between the device and the KSC server is possible using a web server interface. The server provides a web server socket, where the client connects to. The data exchanged in this connection must be secured. The TOE environment provides a Transport Layer Security (TLS) for this purpose.

1.3.3 Product physical / logical features and functionality not included in TOE scope

KES provides other features that are out of the scope of evaluation, therefore there is no assurance level associated with this features and functionality.

Those features are the following:

- Removable Device Encryption functionality
- File Level Encryption functionality
- Detection of malware that are not included in the list of known malware
- Firewall functionality
- Network Attack blocker functionality
- Web Control rules based on website categorisation (“By content category”)
- Application startup control rule based on application categorisation (“KL Category” rule condition)
- Protection of shared folders against remote encryption as part of the Behaviour Detection component
- Cloud mode for Threat Protection
- Anti-Bridging feature in Device Control
- Management through Simplified interface mode.

1.4 TOE Description

This section addresses the physical and logical components of the TOE that are included in the evaluation.

1.4.1 TOE Physical Scope

The Target of Evaluation (TOE) consists of

1) The program code of the KES. The program code is delivered as a binary installation package: keswin_11.6.0.394_en_aes256.exe with SHA256 checksum: 12DBDC9014EC71BC9EF1BE884343DD5C200A662026A7EB7FB9F82E766CC7156B
   The following parts of the installed programs implement the security functionality of the TOE:
   a) Pre-boot agent for establishing secure access to protected hard drive
   b) Main TOE service managing settings and operations
   c) Drivers for disk access and operations.
2) The Application Control Plugin delivered as a ZIP package: keswin_web_plugin_11.6.0.394.zip with SHA256 checksum: 43A8D7377CDB6130BF14E923590D3EE9291C13AE57D46F01E25DA71807CE8E3E
3) The User Manual for administering and maintaining the TOE ”Kaspersky Endpoint Security for Windows. User Manual. Version 2.01”, distributed as PDF file with SHA256 checksum 42D8BB9C86FF8062F7B459C4F87F1EB220691C48768DA460110C8231419FEF30.
4) The Addendum that references User Manual and TOE architectural evidences ”Kaspersky Endpoint Security for Windows. User Manual. Addendum A. Version 2.04”, distributed as PDF file with SHA256 checksum 12B67ADFD1B55554A375AA9170DAE3C3B76694BAC3AD4F872E593BF9EABA641D.
5) The Guide for preparing for installation and installing Kaspersky Endpoint ”Kaspersky Endpoint Security for Windows. Preparative Procedures. Version 2.03” distributed as PDF file with SHA256 checksum CAD0018F6279D26DD5B969C6429E85C2794D1C67EA34AFD82F75162A907DA8B0.

The delivery of the TOE is secured in a manner that any user is able to determine the authenticity of the software package received. The delivery package, including the TOE and associated documentation is downloaded from Kaspersky Lab website.

All executable files of the TOE, including installation package, are digitally signed with a Code Signing Certificate with a timestamp. This allows customers to verify the origin, integrity and authenticity of the TOE. The installation guide delivered together with the product explains how to securely install and configure product in order to bring it into the evaluated state. Also, the SHA256 checksums of the TOE binary files are provided to the customers to confirm that the received TOE files are the expected ones.

1.4.2 TOE Logical Scope

Cryptographic Functionality

The TOE generates cryptographic keys in accordance with a specified cryptographic key generation algorithm and specified cryptographic key sizes that meet the specified standards.

The TOE destroys cryptographic keys in accordance with a specified cryptographic key destruction method.

The TOE performs cryptographic operations in accordance with a specified cryptographic algorithm and cryptographic key sizes that meet the specified standards.

Access control policies

The TOE enforces the policies over FDE access control, application startup control, device access control, and web access control using securely configurable rules.

The TOE enforces the FDE access control policy restricting access to data on HDD for unauthorised users, using full disk encryption methods.

The TOE enforces the application startup control policy to ensure only authorised applications can be launched on protected device. Authorisation is granted based on defined access rules.

The TOE enforces the device access control policy to ensure only authorised removable devices can be used on protected device and device access is granted based on defined access rules.

The TOE enforces the web access control policy to ensure internet access to web sites is granted based on defined access rules.

Anti-Virus protection

The TOE is able to perform real-time, scheduled, and on-demand scans for viruses based upon known malware signatures.

The TOE performs scheduled scans at the time and frequency configured by the authorised administrator. Upon detection of a malware, the TOE takes the action specified by the authorised administrator and generates an audit event that might identify the affected object, the malware (name or type) that was detected, and the action taken by the TOE. The TOE also sends an alarm email to the administrator (if specified) when a malware is detected.

Security Management

The TOE maintains the roles of KLUser and KLAdmin and is able to associate particular users with them.

The TOE restricts the ability to modify implemented access policies' security attributes to the authorised administrator only.

The TOE enforces access control policies providing permissive default values for policies' security attributes, except for FDE access policy, which has restrictive ones.

The TOE allows the authorised administrator to specify alternative initial values to override the default values when an object or information is created.

The TOE is capable of performing the management functions on its anti-virus functionality and access controls.

Identification and authentication

The TOE requires users to be successfully identified and authenticated before allowing access to its functions.

1.4.3 TOE Development Environment

TOE is developed through established development processes and procedures, including secure SDL practices.

Configuration management, code access control is managed by MS Team Foundation System, set of automated tests, both quality and functional, are performed on each build during development process.

Build process is automated.

Physical security is maintained in development facility in Moscow, together with relevant personnel screening and education, preventing unauthorised influence on any stage of development.

Code quality is maintained via code standards, education, regular code reviews. Programming languages and tools are chosen for maximum development efficiency.

Flaw remediation and vulnerability handling are well documented and supported by relevant processes.

1.4.4 Evaluated configuration

The product was evaluated in the following configuration (see Fig. 2):

- Kaspersky Endpoint Security for Windows (version 11.6.0.394 AES256) is installed on a managed endpoint device (‘Host1’) running Windows 10 Education (20H2) 64-bit.
- Managed endpoint device (‘Host1’) also has Kaspersky Security Center 13 (Network Agent component) installed.
- Kaspersky Security Center 13 Administration Server and Network Agent components are installed on a device (‘Server1’) running Windows Server 2016 64-bit.
- Kaspersky Security Center 13 Web Console is installed on a device (‘Host2’) running Windows 10 Education (20H2) 64-bit.
- Kaspersky Endpoint Security for Windows management plug-in is also installed on the device with Web Console (‘Host2’).
- All devices connected to LAN.

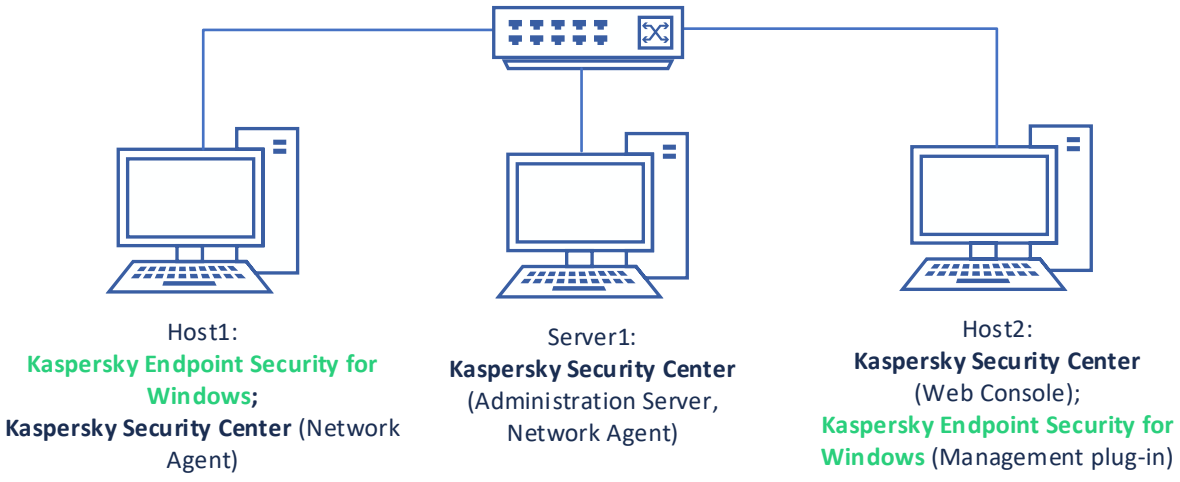

Figure 2. Evaluated configuration

2 Conformance Claims

2.1 CC Conformance Claim

This Security Target and the TOE claim conformance to part 2 and part 3 of CC Version 3.1, Revision 5:

CC part 2 extended (CCMB-2017-04-002),

CC part 3 conformant (CCMB-2017-04-003).

2.2 PP Claim

This Security Target does not claim conformance to a Protection Profile.

2.3 Package Claim

This Security Target claims conformance to EAL2 augmented with ALC_FLR.1.

2.4 Conformance Rationale

As this Security Target does not claim conformance to a Protection Profile a conformance claim rationale is not necessary.

3 Security Problem Definition

3.1 Assets

Assets protected by the TOE include two types:

1. User data on the computer.
2. TOE data.

Name	Short description	Description	Type
ASS.KEYS	Cryptographic Keys	Cryptographic keys used for FDE functionality.	TOE Data
ASS.TOE_DATA	TOE Settings, binary files, data in memory, and AV bases.	TOE settings, including secure configuration and system-specific settings that are stored in files and registry, TOE binaries, TOE process in memory and data file containing antimalware records and signatures.	TOE Data
ASS.USER_FILE	User files	User files containing user data that can be compromised.	User Data

3.2 Threats

The threats are defined by an adverse action performed by a threat agent on an asset.

Threat agents may be attackers that are unauthorised users, processes or external entities, for example:

- Persons, who obtain unauthorised physical access to protected device (i.e. stole laptop)
- Hackers (with substantial expertise, standard equipment, and being paid to do so) who is trying to compromise confidentiality and integrity of confidential data by disrupting TOE functionality.

The TOE addresses the following threats:

Name	Description
T.ACCESS_DD	Device data access An attacker with physical access to the switched off device attempts to access any data stored on the encrypted device. **Endangered asset: ASS.USER_FILE (confidentiality, integrity)**
T.ACCESS_CD	Configuration data access An attacker attempts to change the TOE security configuration while the device secured by the TOE is switched on (and both the device and the TOE are running). The aim of the manipulation is to get access to data or key in the next steps. The manipulation could be a change of TSF (authentication procedures, key derivation, etc.) or change of TSF data (local user password, policy, etc.) **Endangered asset: ASS.KEYS (confidentiality, integrity), ASS.TOE_DATA (confidentiality, integrity)**

T.ACCESS_KD	Cryptographic keys data access An attacker can try to obtain cryptographic keys in plain text or unencrypted key material, which allows keys derivation. **Endangered asset: ASS.KEYS (confidentiality, integrity)**
T.KEY_DISCLOSURE	Key disclosure An attacker discloses a cryptographic key as a result of a brute force attack or key guessing due to key generation weakness in the TOE. **Endangered asset: ASS.KEYS (confidentiality)**

3.3 Organisational Security Policies

The following table describes the organisational security policies relevant to the operation of the TOE.

Name	Description
P.ACCESS_DV	Use of removable devices should be controlled. Administrators should be able to restrict usage of removable devices on protected machines or certain types of operation.
P.LAUNCH	Launch of applications and scripts should be restricted. Administrators shall be able to maintain and enforce a list of permitted or forbidden applications.
P.WEBACC	Web access from protected machines have to be controlled or restricted based on web resource properties or data type.
P.VIRUS	Files and objects on protected machines have to be checked for known viruses (malware).

3.4 Assumptions

The following assumptions are made to guarantee the TOE's security:

Name	Description
A.PROTECT_ACCESS	Attacker access protection The device secured by the TOE should not fall under temporary and undetected physical control of an attacker when the device is booted. It is assumed that potential attackers do not have physical or logical access to the device secured by the TOE before and during the TOE installation.
A.AUTHORISED_USER	Correct behaviour of authorised users It is assumed that authorised users handle the device secured by the TOE and the TOE itself with the necessary care and diligence. Authorised users are not trying to actively compromise the security of the device secured by the TOE and the TOE itself, and are instructed not to leave a device secured by the TOE while it is switched on and running.

A.SECURE_SERVER	Secure environment of the KSC server It is assumed that KSC is installed and configured to enable management of the TOE. It is assumed that the KSC server is operating in a secure environment with strong physical and logical access restrictions. The secure environment provides the needed quality, integrity and confidentiality of the relevant cryptographic material and keys stored on the KSC server. Furthermore, the security environment provides a secure channel for connection between the device on which TOE is installed and running and the KSC.
A.SECURE_OPER	TOE secure operation It is assumed, that non-trusted software (especially with ability to perform direct access to the hard disk or kernel mode operations) is not placed on the device and cannot be executed.
A.PROTECT_PASSWD	Password protection It is assumed that all authorised individuals (users, administrators) protect their passwords and/or PINs for Token to avoid its disclosure. It is furthermore assumed that the corresponding security measures sufficiently protect against password eavesdropping and recording using software tools or additional hardware devices. For the configuration that allows the usage of Token as an authentication device, the assumption means a secure handling of Token too.
A.TRUST_ADMIN	Trusted administrat**or** It is assumed that administrators responsible for the device and KSC server administration can be trusted and perform all tasks regarding the TOE security correctly and with due diligence.

4 Security Objectives

The security objectives are a high-level statement of the intended response to the security problem. These objectives indicate how the security problem, as characterised in the “TOE Security Environment” section, is to be addressed.

4.1 Security Objectives for the TOE

The following security objectives are defined for the TOE:

Name	Description
O.ACCESS_DV	Device access control The TOE will provide mechanisms to authorise users to access devices specified by TOE. User authorisation is based on access rights configured by the authorised users of the TOE and the binding of external attributes to subjects recognised by the TSF.
O.LAUNCH	Application startup control The TOE will provide mechanisms to authorise launch of applications and PowerShell scripts or loading of DLLs and drivers. User authorisation is based on access rights configured by the authorised users of the TOE and the binding of external attributes to subjects recognised by the TSF.
O.WEBACC	Web access control The TOE will provide mechanisms to authorise users to access web sites specified by TOE once the user has been authenticated. User authorisation is based on access rights configured by the authorised users of the TOE and the binding of external attributes to subjects recognised by the TSF.
O.SECURE_DATA	Secure data The TOE must provide a functionality to prohibit the access to the device data (user data and operating system data) and cryptographic keys from an unauthorised individual who has physical access to the switched off device. Protection is ensured by a TSF that encrypts the data. Authorised individuals can access the device data after positive authentication (with password or Token usage) and data encryption. The cryptographic mechanisms of the TOE must ensure that an unauthorised individual cannot disclosure a cryptographic key by means of brute force attack or to guess a key due to key generation weakness in the TOE. The TOE has to use strong cryptographic algorithms based on keys with appropriate key length that are resistant against brute force attack. For key generation the TOE uses random numbers from random number generator providing unpredictable results.
O.SECURE_MANAGEMENT	Secure management The TOE must provide mechanisms to ensure that only authorised users are able to log in and configure the TOE, and provide protections for logged-in administrators. Authorised users can perform TOE management after successful authentication (password check).
O.VIRUS	Anti-Virus The TOE must provide mechanisms to detect and take action against known viruses (malware) introduced to the protected computer via network traffic or removable media.

4.2 Security Objectives for the Operational Environment

The following security objectives are defined for the TOE Operational Environment:

Name	Description
OE.PROTECT_ACCESS	Attacker access protection The device secured by the TOE should not fall under temporary and undetected physical control of an attacker when the device is booted. Potential attacker must not have physical or logical access to the device secured by the TOE before and during the TOE installation. Appropriate physical security measures and physical security policies have to be in place.
OE.AUTHORISED_USER	Correct behaviour of authorised users Authorised users shall not actively compromise the security of the device secured by the TOE and the TOE itself and should be instructed not to leave a device secured by the TOE while it is switched on and running.
OE.SECURE_SERVER	Secure environment of the KSC server. KSC has to be installed and configured to enable management of the TOE. The KSC server shall be located in a trusted environment that provides strong physical and logical access restrictions. The interaction of integrated security measures in the KSC server environment ensures the needed quality, integrity and confidentiality of the relevant cryptographic material and keys stored on the server. The TOE and the KSC server communicate using a secure TLS connection that is provided by the environment. The Network Agent of the KSC has to be used. The Network Agent TLS connection has to be configured to provide a strong server authentication together with strong encryption and integrity protection of all transmitted data.
OE.SECURE_OPER	TOE secure operation Non-trusted software (especially with ability to perform direct access to the hard disk) is not installed and will not be installed on the device secured by the TOE. The users are instructed not to install or use utility programs like partition managers or disk copy programs.
OE.PROTECT_PASSWD	Password protection All authorised individuals (users, administrators) protect their passwords and/or PINs for Token to avoid disclosure. They are instructed to keep their password secret and not to write down their password, neither manually nor electronically. Unauthorised individuals shall not get the password of an authorised individual. The corresponding security measures sufficiently protect against password/PIN eaves dropping and recording using software tools or additional hardware devices. In particular, the devices and the environment shall be protected against installing any software programs or hardware devices, which enable capturing user password inputs on the keyboard.
OE.TRUST_ADMIN	Trusted administrator The administrators responsible for the device and KSC server administration have to be trustworthy. They perform all tasks correctly regarding the TOE security.

4.3 Security Objective Rationale

The following table lists all objectives for the TOE and the Operational Environment to show which objectives are necessary to counter a threat or satisfy an assumption or organisational security policies relevant to the operation of the TOE. The table also shows that no objective exists which does not trace back to a threat, policy or assumption.

Objective / Threat, Policy, or Assumption	O.ACCESS_DV	O.LAUNCH	O.WEBACC	O.VIRUS	O.SECURE_DATA	O.SECURE_MANAGEMENT	OE.PROTECT_ACCESS	OE.AUTHORISED_USER	OE.SECURE_SERVER	OE.SECURE_OPER	OE.PROTECT_PASSWD	OE.TRUST_ADMIN
T.ACCESS_DD					X							
P.ACCESS_DV	X											
T.ACCESS_CD		X		X		X	X	X	X	X		X
T.ACCESS_KD					X		X	X	X			
T.KEY_DISCLOSURE					X							
P.LAUNCH		X										
P.WEBACC			X									
P.VIRUS		X		X								
A.PROTECT_ACCESS							X					
A.AUTHORISED_USER								X				
A.SECURE_SERVER									X			
A.SECURE_OPER										X		
A.PROTECT_PASSWD											X	
A.TRUST_ADMIN												X

The following table shows why the chosen objectives are sufficient to counter a threat or satisfy an assumption.

Threat, Policy, or Assumption	Objectives
T.ACCESS_DD	O.SECURE_DATA prevents that unauthorised individuals get access to device data and keys stored on the hard disk after the device has been switched off. This security objective exactly counters the threat T.ACCESS_DD.
P.ACCESS_DV	O.ACCESS_DV prevents that authorised individual use devices that they are not authorised to access.

	This security objective fulfils the security policy P.ACCESS_DV
T.ACCESS_CD	O.SECURE_MANAGEMENT prevents that unauthorised individuals are able to perform local management operations on the TOE and get access to configuration data and passwords. O.LAUNCH prevents that unauthorised individuals or processes execute applications or scripts or load DLLs or drivers that can be used to tamper TOE and TOE configuration. O.VIRUS prevents execution of malware that can be used to tamper TOE and TOE configuration. OE.PROTECT_ACCESS prevents that unauthorised individuals have undetected physical access to the device secured by the TOE that can be used to tamper device to coerce authorised user's credentials. OE.AUTHORISED_USER guarantees that authorised users are instructed not to install any software which could modify the TOE programs and configuration data (TSF and TSF data), not to use any software manipulating the hard disk directly and not to leave the device in the time in which the device and the TOE are running. OE.TRUST_ADMIN and OE.SECURE_SERVER guarantee that only authorised individuals provide configuration data for the TOE on the KSC server that is placed in the secure environment. OE.SECURE_OPER guarantees that non-trusted software (especially with ability to perform direct access to the hard disk) is not installed and will not be installed on the device secured by the TOE.
T.ACCESS_KD	O.SECURE_DATA prevents that unauthorised individuals get access to device data and keys stored on the hard disk after the device has been switched off. OE.PROTECT_ACCESS prevents that unauthorised individuals TOE should not fall under temporary and undetected physical control of an attacker when the device is booted, where attacker can perform attack on key data in memory. OE.AUTHORISED_USER guarantees that authorised users are instructed not to install any software which could modify the TOE programs and configuration data (TSF and TSF data), not to use any software manipulating the hard disk directly and not to leave the device secured by the TOE in the time in which the device and the TOE are running. OE.SECURE_SERVER guarantees that the TOE and the KSC server communicate using a secure TLS connection that ensures the confidentiality and integrity of all transmitted data.
T.KEY_DISCLOSURE	O.SECURE_DATA ensures that strong cryptographic algorithms based on keys with appropriate key length is in use to prevent the disclosure of cryptographic keys by means of brute force attack. Random number generator that provides unpredictable results is used for key generation. This security objective exactly counters threat T.KEY_DISCLOSURE.
P.LAUNCH	O.LAUNCH prevents that individuals or processes execute applications or scripts or load DLLs or drivers that are not authorised for use in certain operating environment. This security objective exactly fulfils the security policy P.LAUNCH.
P.WEBACC	O.WEBACC control web access from protected machine and prevents that an authorised individual connects to certain internet hosts that are restricted by an organisational policy. This security objective exactly fulfils the security policy P.WEBACC

P.VIRUS	O.VIRUS makes sure applications and files on a protected machines or removable devices are checked for known viruses (malware) signatures to implement this organisation policy. O.LAUNCH provides access control policies that can be used to prevent launching unknown malicious code, which has not been identified by O.VIRUS.
A.PROTECT_ACCESS	OE.PROTECT_ACCESS The device secured by the TOE should not fall under temporary and undetected physical control of an attacker when the device is booted. Potential attacker must not have physical or logical access to the device secured by the TOE before and during the TOE installation. Appropriate physical security measures and physical security policies have to be in place. This security objective exactly covers the assumption A.PROTECT_ACCESS.
A.AUTHORISED_USER	OE.AUTHORISED_USER guarantees that do not leave the device secured by the TOE in the time in which the device and the TOE are running, and are not trying to bypass actively compromise the device and TOE. This security objective exactly covers the assumption A.AUTHORISED_USER.
A.SECURE_SERVER	OE.SECURE_SERVER ensures that the KSC server is located in a trusted environment with required security measures that provides the needed quality, integrity and confidentiality of the relevant cryptographic material and keys stored on the server. OE.SECURE_SERVER guarantees that the TOE and the KSC server communicate using a secure TLS connection that ensures the confidentiality and integrity of all transmitted data. This security objective exactly covers the assumption A.SECURE_SERVER.
A.SECURE_OPER	OE.SECURE_OPER guarantees that non-trusted software (especially with ability to perform direct access to the hard disk) is not installed and will not be installed on the device secured by the TOE. This security objective exactly covers the assumption A.SECURE_OPER.
A.PROTECT_PASSWD	OE.PROTECT_PASSWD ensures that all authorised individuals (users, administrators) protect their passwords and PINs for Token to avoid disclosure. The corresponding security measures sufficiently protect against password eavesdropping and recording by use of software tools or additional hardware devices. This security objective exactly covers the assumption A.PROTECT_PASSWD.
A.TRUST_ADMIN	OE.TRUST_ADMIN guarantees that administrators responsible for the device and KSC server administration are trustworthy and perform all tasks correctly regarding the TOE security. This security objective exactly covers the assumption A.TRUST_ADMIN.

5 Extended Components Definition

This section defines extended security functionality that is not a part of CC.

5.1 Definition of the Class FAV - Anti-Virus

The FAV: Anti-Virus class is composed of three families: Anti-Virus Actions (FAV_ACT), Anti-Virus Alerts (FAV_ALR) and Anti-Virus Scanning (FAV_SCN).

Anti-Virus Actions (FAV_ACT) family addresses the aspects of actions taken on detected viruses, while the Anti-Virus Alerts (FAV_ALR) family is concerned with the correspondent informing about virus detection, and Anti-Virus Scanning (FAV_SCN) family addresses the scanning process.

5.1.1 FAV_ACT Anti-Virus Actions

Family Behaviour

This family defines requirements for actions to be taken on virus detection.

Component levelling

FAV_ACT: Anti-Virus Actions	1

FAV_ACT.1 requires that the TOE take actions against viruses once they detected and defines such actions.

Management: FAV_ACT.1

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

a) Configuration of parameters of actions.

Audit: There are no actions defined to be auditable.

FAV_ACT.1 Anti-Virus Actions

Hierarchical to: No other components.
Dependencies: No dependencies.

FAV_ACT.1.1 Upon detection of a virus, the TSF shall perform the action(s) specified by [assignment: *the authorised identified roles*]. Actions are configurable for each type of scan and consist of:

a) Disinfect,

b) Delete the file,

c) [selection: [assignment: *list of other actions*], *no other actions*].

5.1.2 FAV_ALR Anti-Virus Alerts

Family Behaviour

This family defines requirements for delivering security alerts to the users.

Component levelling

FAV_ALR.1 Anti-Virus Alerts defines alerting requirements to ensure the users are aware that a virus was detected.

Management: FAV_ALR.1

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

a) Configuration of parameters of alerts.

Audit: There are no actions defined to be auditable.

FAV_ALR.1	Anti-Virus Alerts Hierarchical to: Dependencies:	 No other components. No dependencies.
FAV_ALR.1.1	The TSF shall be able to generate an audit event indicating detection of a virus. The event shall identify the object, the virus that was detected, and the action taken by the TOE.	
FAV_ALR.1.2	The TSF shall send an alarm to [assignment: *alarm destination*] when a virus is detected.	

5.1.3 FAV_SCN Anti-Virus Scanning

Family Behaviour

This family defines requirements for scanning for viruses.

Component levelling

FAV_SCN.1 Anti-Virus Scanning requires that the TOE scans for viruses and defines parameters of scanning.

Management: FAV_SCN.1

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

a) Configuration of parameters of scanning.

Audit: There are no actions defined to be auditable.

FAV_SCN.1	Anti-Virus Scanning Hierarchical to: No other components. Dependencies: No dependencies.
FAV_SCN.1.1	The TSF shall perform real-time, scheduled, and on-demand scans for viruses based upon known signatures.
FAV_SCN.1.2	The TSF shall perform scheduled scans at the time and frequency configured by [assignment: *the authorised identified roles*].

6 Security Requirements

This ST claims to be CC Part 2 extended, see previous section. No extended Security Assurance Requirements are defined and used, as this ST claims to be CC Part 3 conformant.

The notation, formatting and conventions used in this section are consistent with those used in Version 3.1 of the Common Criteria (CC). The CC allows several operations to be performed on functional requirements; refinement, selection, assignment and iteration are defined in Section 8.1 of Part 1 of the CC:

- Refinements are indicated by **bold** text and ~~strikethrough~~
- Selections are enclosed in [square brackets]
- Assignments are enclosed in [square brackets and underlined]
- Iterations are numbered in sequence as appropriate

6.1 Security Functional Requirements

Following SFRs are defined for the TOE:

SFR	SFR name
FCS_CKM.1(1)	Cryptographic key generation (DEK/MK)
FCS_CKM.1(2)	Cryptographic key generation (User key)
FCS_CKM.4	Cryptographic key destruction
FCS_COP.1(1)	Cryptographic operation (Data Encryption/Decryption)
FCS_COP.1(2)	Cryptographic operation (Key Encryption/Decryption)
FCS_COP.1(3)	Cryptographic operation (HMAC calculation)
FCS_COP.1(4)	Cryptographic operation (RSA Key Encryption)
FDP_ACC.1(1)	Subset access control (FDE)
FDP_ACC.1(2)	Subset access control (ASC)
FDP_ACC.1(3)	Subset access control (DAC)
FDP_IFC.1	Subset information flow control (WAC)
FDP_ACF.1(1)	Security attribute based access control (FDE)
FDP_ACF.1(2)	Security attribute based access control (ASC)
FDP_ACF.1(3)	Security attribute based access control (DAC)
FDP_IFF.1	Simple security attributes (WAC)
FIA_UAU.2	User authentication before any action
FIA_UID.2	User identification before any action
FMT_MSA.1(1)	Management of security attributes (FDE)
FMT_MSA.1(2)	Management of security attributes (ASC)
FMT_MSA.1(3)	Management of security attributes (DAC)
FMT_MSA.1(4)	Management of security attributes (WAC)
FMT_MSA.3(1)	Static attribute initialisation (FDE)
FMT_MSA.3(2)	Static attribute initialisation (ASC)
FMT_MSA.3(3)	Static attribute initialisation (DAC)

FMT_MSA.3(4)	Static attribute initialisation (WAC)
FMT_MTD.1	Management of TSF data
FMT_SMF.1	Specification of management functions
FMT_SMR.1	Security roles
FAV_ACT.1	Anti-virus actions
FAV_ALR.1	Anti-virus alerts
FAV_SCN.1	Anti-virus scanning

6.1.1 Class FCS: Cryptographic Support

6.1.1.1 FCS_CKM.1(1) Cryptographic key generation (DEK/MK)

FCS_CKM.1(1).1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [key generation using a deterministic random number generator][1] and specified cryptographic key sizes [256 bit][2] that meet the following: [Hash_DRBG according to NIST SP 800-90A with SHA-256][3].

6.1.1.2 FCS_CKM.1(2) Cryptographic key generation (User key)

FCS_CKM.1(2).1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [Password-Based Key Derivation Function 2 (PBKDF2) with HMAC-SHA256, 10.000 iteration value, 256 bit salt and password as input][4] and specified cryptographic key sizes [256 bit][5] that meet the following: [NIST SP 800-132 Option 2a,][6].

6.1.1.3 FCS_CKM.4 Cryptographic key destruction

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [overwriting with zeroes][7] that meets the following: [none][8].

6.1.1.4 FCS_COP.1(1) Cryptographic operation (Data Encryption/Decryption)

FCS_COP.1(1).1 The TSF shall perform [symmetric encryption/decryption of device block data][9] in accordance with a specified cryptographic algorithm [XTS-AES-256][10] and cryptographic key sizes [256 bit][11] that meet the following: [NIST SP 800-38E, FIPS 197][12].

Application note 1: For the XTS-AES-256, the TSF uses two AES keys, each 256 bit long, which together make a 512-bit long key in accordance with the mentioned algorithm and standard.

[1] [assignment: *cryptographic key generation algorithm*]
[2] [assignment: *cryptographic key sizes]*
[3] [assignment: *list of standards*]
[4] [assignment: *cryptographic key generation algorithm*]
[5] [assignment: *cryptographic key sizes]*
[6] [assignment: *list of standards*]
[7] [assignment: *cryptographic key destruction method*]
[8] [assignment: *list of standards*]
[9] [assignment: *list of cryptographic operations*]
[10] [assignment: *cryptographic algorithm*]
[11] [assignment: *cryptographic key sizes]*
[12] [assignment: *list of standards*]

6.1.1.5 FCS_COP.1(2) Cryptographic operation (Key Encryption/Decryption)

FCS_COP.1(2).1 The TSF shall perform [symmetric encryption/decryption of data (key)][13] in accordance with a specified cryptographic algorithm [AES-256-ECB][14] and cryptographic key sizes [256 bit][15] that meet the following: [FIPS 197, NIST SP 800-38A][16].

6.1.1.6 FCS_COP.1(3) Cryptographic operation (HMAC calculation)

FCS_COP.1(3).1 The TSF shall perform [Keyed-Hash Message Authentication Code calculation][17] in accordance with a specified cryptographic algorithm [HMAC-SHA256][18] and cryptographic key sizes [256 bit][19] that meet the following: [FIPS 180-4, and FIPS 198-1][20].

Application note 2: The TSF uses the AES keys (Master Key) for HMAC-SHA256 calculation.

6.1.1.7 FCS_COP.1(4) Cryptographic operation (RSA Key Encryption)

FCS_COP.1(4).1 The TSF shall perform [asymmetric data encryption of data (key)][21] in accordance with a specified cryptographic algorithm [RSA-EME-OAEP][22] and cryptographic key sizes [2048 bit][23] that meet the following: [PKCS#1, v2.2][24].

6.1.2 Class FDP: User Data Protection

6.1.2.1 FDP_ACC.1(1) Subset access control (FDE)

FDP_ACC.1(1).1 The TSF shall enforce the [FDE Access Control SFP][25] on [Subjects (Authorised Users), Objects (Device data / DEK), Operations (transparently decrypt device data using DEK)][26].

6.1.2.2 FDP_ACC.1(2) Subset access control (ASC)

FDP_ACC.1(2).1 The TSF shall enforce the [Application Startup Control SFP][27] on [Subjects (Authorised Users), Objects (Executable files and Scripts), Operations (execute)][28].

Application note 3: Here and further in this class SFRs: Among Authorised Users can be: Users, Groups, or Built-in security principals (including the operating system process). Not to be mixed up with the security roles of KLUser and KLAdmin (FMT_SMR.1). While all Users are KLUsers, this SFP can also be applied to other system entities.

Application note 4: Here and further in this class' SFRs: 'Executable files' mean applications, dynamic link libraries (DLL modules), and system drivers. 'Scripts' mean JavaScripts, VBscripts, Windows Registry files (.reg), Windows Command scripts (.cmd), Batch Command Files (.bat), PowerShell Scripts.

[13] [assignment: *list of cryptographic operations*]
[14] [assignment: *cryptographic algorithm*]
[15] [assignment: *cryptographic key sizes]*
[16] [assignment: *list of standards*]
[17] [assignment: *list of cryptographic operations*]
[18] [assignment: *cryptographic algorithm*]
[19] [assignment: *cryptographic key sizes]*
[20] [assignment: *list of standards*]
[21] [assignment: *list of cryptographic operations*]
[22] [assignment: *cryptographic algorithm*]
[23] [assignment: *cryptographic key sizes]*
[24] [assignment: *list of standards*]
[25] [assignment: *access control SFP*]
[26] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]
[27] [assignment: *access control SFP*]
[28] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

Application note 5: Here and further in this class SFRs: 'Execution' mean application launch, loading of DLL modules or drivers, execution of Scripts through supported execution methods. Execution of scripts from running interpretation is not covered.

6.1.2.3 FDP_ACC.1(3) Subset access control (DAC)

FDP_ACC.1(3).1 The TSF shall enforce the [Device Access Control SFP][29] on [Subjects (Authorised Users), Objects (Device data), Operations (read, write)][30].

6.1.2.4 FDP_IFC.1 Subset information flow control (WAC)

FDP_IFC.1.1 The TSF shall enforce the [Web Access Control SFP][31] on [Subjects (Authorised Users), Information (web content through HTTP), Operations (get, post)][32].

6.1.2.5 FDP_ACF.1(1) Security attribute based access control (FDE)

FDP_ACF.1(1).1 The TSF shall enforce the [FDE Access Control SFP][33] to objects based on the following: [Subjects (Authorised Users), Objects (device data / DEK), Subject attributes (password); Object attributes: (IDs of disks covered by SFP)][34].

FDP_ACF.1(1).2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [If the checks (Password Key derivation with PBKDF2, User Key decryption with Password Key, Master Key decryption with User Key, HMAC-SHA256 calculation with Master Key, HMAC-SHA256 has to match the stored user verification value) transparent decryption is performed][35].

FDP_ACF.1(1).3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [if disk ID is not among IDs of disks covered by policy][36].

FDP_ACF.1(1).4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [no additional rules][37].

6.1.2.6 FDP_ACF.1(2) Security attribute based access control (ASC)

FDP_ACF.1(2).1 The TSF shall enforce the [Application Startup Control SFP][38] to objects based on the following: [Subjects (Authorised Users), Objects (Executable files and scripts), Subject attributes (Authorised users IDs) and Object's attributes as defined in a table below][39].

Object	Object attributes group	Object Attributes	Attribute description
	Properties	File hash code	SHA-256 hash value of object.
		Path to file	System path to object, including wildcards.

[29] [assignment: *access control SFP*]
[30] [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]
[31] [assignment: *information flow control SFP*]
[32] [assignment: *list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP*]
[33] [assignment: *access control SFP*]
[34] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]
[35] [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]
[36] [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]
[37] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]
[38] [assignment: *access control SFP*]
[39] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

Executable files and scripts		Condition by file drive	Object is being located on removable drive.
	Certificate *(Properties of Authenticode signature on the object if exists)*	Issuer	Issuer field in certificate, used to sign object.
		Subject	Subject field in certificate, used to sign object.
		Thumbprint	Thumbprint field in certificate, used to sign object.
	Metadata *(Fields in object properties section, if exist)*	File name	Field in file properties section.
		File version	Field in file properties section (with logical operations).
		Application name	Field in file properties section.
		Application version	Field in file properties section (with logical operations).
		Vendor	Field in file properties section.

FDP_ACF.1(2).2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [when an Authorised user is trying to execute application or script or to load DLLs or drivers the active user identity and object attributes are compared to applicable rules in active policy which applies to application. The outcome of these rule decisions is one of the following states: startup is allowed, startup is blocked][40].

FDP_ACF.1(2).3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [if Application Startup Control mode is set to “Denylist” all objects not matching attributes in policy are allowed to execute or object is part of the TOE][41].

FDP_ACF.1(2).4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [if Application Startup Control mode is set to “Allowlist” all objects not matching attributes in policy are blocked from execution][42].

Application note 6: The certificate-based rules depend on the certificate validity; hence they are not recommended to be used in Denylist mode. The certificate-based rule will not be enforced if the certificate is not valid.

6.1.2.7 FDP_ACF.1(3) Security attribute based access control (DAC)

FDP_ACF.1(3).1 The TSF shall enforce the [Device Access Control SFP][43] to objects based on the following: [Subjects (Authorised Users), Objects (removable media), Subjects attributes (Authorised Users IDs, assigned access schedule), Object attributes (target device type, device bus, device properties)][44].

FDP_ACF.1(3).2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [operations are checked against the set of rules containing device type, device bus, device properties, Authorised User ID and access schedule for this user. The outcome of this is two states: access allowed or access blocked][45].

FDP_ACF.1(3).3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [no additional rules][46].

[40] [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]
[41] [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]
[42] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]
[43] [assignment: *access control SFP*]
[44] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]
[45] [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]
[46] [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]

FDP_ACF.1(3).4 The TSF shall explicitly deny access to objects based on the following additional rules: [if there are conflicting rules both blocking and allowing to the object][47].

6.1.2.8 FDP_IFF.1 Simple security attributes (WAC)

FDP_IFF.1.1 The TSF shall enforce the [Web Access Control SFP][48] based on the following types of subject and information security attributes: [Subjects (Authorised Users), Information (web content through HTTP), Subject attributes (Authorised User ID) and Information security attributes (Web address, Content type, Time of operation)][49].

FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: [SFP rules do not contain any attributes matching transmitted information][50].

FDP_IFF.1.3 The TSF shall enforce the [warning alerts for user if action Warn is defined in rules, that matches transmitted information, subject and operation time and has higher priority than other matching rules][51].

FDP_IFF.1.4 The TSF shall explicitly authorise an information flow based on the following rules: [if action Allow is defined in rules, that matches transmitted information, subject and operation time and has higher priority than other matching rules][52].

FDP_IFF.1.5 The TSF shall explicitly deny an information flow based on the following rules: [if action Block is defined in rules, that matches transmitted information, subject and operation time and has higher priority than other matching rules][53].

6.1.3 Class FIA: Identification and Authentication

6.1.3.1 FIA_UAU.2 User authentication before any action

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.

6.1.3.2 FIA_UID.2 User identification before any action

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.

6.1.4 Class FMT: Security Management

6.1.4.1 FMT_MSA.1(1) Management of security attributes (FDE)

FMT_MSA.1(1).1 The TSF shall enforce the [FDE Access Control SFP][54] to restrict the ability to [modify][55] the security attributes [IDs of disks covered by SFP][56] to [KLAdmin][57].

[47] [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]
[48] [assignment: *information flow control SFP*]
[49] [assignment: *list of subjects and information controlled under the indicated SFP, and for each, the security attributes*]
[50] [assignment: *for each operation, the security attribute-based relationship that must hold between subject and information security attributes*]
[51] [assignment: *additional information flow control SFP rules*]
[52] [assignment: *rules, based on security attributes, that explicitly authorise information flows*]
[53] [assignment: *rules, based on security attributes, that explicitly deny information flows*]
[54] [assignment: *access control SFP(s), information flow control SFP(s)*]
[55] [selection: *change_default, query, modify, delete, [assignment: other operations]*]
[56] [assignment: *list of security attributes*]
[57] [assignment: *the authorised identified roles*]

6.1.4.2 FMT_MSA.1(2) Management of security attributes (ASC)

FMT_MSA.1(2).1 The TSF shall enforce the [Application Startup Control SFP][58] to restrict the ability to [modify, delete][59] the security attributes [Subject attributes and Object attributes as defined in 6.1.2.6][60] to [KLAdmin][61].

6.1.4.3 FMT_MSA.1(3) Management of security attributes (DAC)

FMT_MSA.1(3).1 The TSF shall enforce the [Device Access Control SFP][62] to restrict the ability to [modify][63] the security attributes [target device type, device bus, device properties][64] to [KLAdmin][65].

6.1.4.4 FMT_MSA.1(4) Management of security attributes (WAC)

FMT_MSA.1(4).1 The TSF shall enforce the [Web Access Control SFP][66] to restrict the ability to [modify, delete][67] the security attributes [Subject attributes and Object attributes as defined in 6.1.2.8][68] to [KLAdmin][69].

6.1.4.5 FMT_MSA.3(1) Static attribute initialisation (FDE)

FMT_MSA.3(1).1 The TSF shall enforce the [FDE Access Control SFP][70] to provide [restrictive][71] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3(1).2 The TSF shall allow the [KLAdmin][72] to specify alternative initial values to override the default values when an object or information is created.

Application note 7: For the first device data encryption, user accounts with pre-defined passwords have to be available. User accounts with pre-defined passwords are either generated by the TOE in accordance with the administration data (TSF data) received from the administration server or received by the TOE with these administration data (TSF data). The FDE Access Control SFP regulates the authorisation for device data decryption, which is possible only with the Data Encryption Key (DEK) and only by authenticated user with valid password. The password is not valid for device data access if the user did not change the pre-defined password or the administration data (TSF data) received from the administration server pointed a password changing necessity out. If the user has not changed the password, the access to the DEK and device data is forbidden. The successful authentication with such a password allows password changing only.

6.1.4.6 FMT_MSA.3(2) Static attribute initialisation (ASC)

FMT_MSA.3(2).1 The TSF shall enforce the [Application Startup control SFP][73] to provide [permissive][74] default values for security attributes that are used to enforce the SFP.

---

[58] [assignment: *access control SFP(s), information flow control SFP(s)*]
[59] [selection: *change_default, query, modify, delete, [assignment: other operations]*]
[60] [assignment: *list of security attributes*]
[61] [assignment: *the authorised identified roles*]
[62] [assignment: *access control SFP(s), information flow control SFP(s)*]
[63] [selection: *change_default, query, modify, delete, [assignment: other operations]*]
[64] [assignment: *list of security attributes*]
[65] [assignment: *the authorised identified roles*]
[66] [assignment: *access control SFP(s), information flow control SFP(s)*]
[67] [selection: *change_default, query, modify, delete, [assignment: other operations]*]
[68] [assignment: *list of security attributes*]
[69] [assignment: *the authorised identified roles*]
[70] [assignment: *access control SFP(s), information flow control SFP(s)*]
[71] [selection, choose one of: *restrictive, permissive, [assignment: other property]*]
[72] [assignment: *the authorised identified roles*]
[73] [assignment: *access control SFP(s), information flow control SFP(s)*]
[74] [selection, choose one of: *restrictive, permissive, [assignment: other property]*]

FMT_MSA.3(2).2 The TSF shall allow the [KLAdmin][75] to specify alternative initial values to override the default values when an object or information is created.

6.1.4.7 FMT_MSA.3(3) Static attribute initialisation (DAC)

FMT_MSA.3(3).1 The TSF shall enforce the [Device Access Control SFP][76] to provide [permissive][77] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3(3).2 The TSF shall allow the [KLAdmin][78] to specify alternative initial values to override the default values when an object or information is created.

6.1.4.8 FMT_MSA.3(4) Static attribute initialisation (WAC)

FMT_MSA.3(4).1 The TSF shall enforce the [Web Access Control SFP][79] to provide [permissive][80] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3(4).2 The TSF shall allow the [KLAdmin][81] to specify alternative initial values to override the default values when an object or information is created.

6.1.4.9 FMT_MTD.1 Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to [modify, change default][82] the [TSF data listed in a table below][83] to [roles listed in a table below][84].

SFR	TSF	TSF data	Authorised User
FMT_SMR.1	SF_MGMT	Password	KLUser, KLAdmin
FAV_ACT.1	SF_AVP	Default actions to be taken	KLAdmin
FAV_ALR.1	SF_AVP	Audit log size, notification settings	KLAdmin
FAV_SCN.1	SF_AVP	Types of scan, scan schedule, scan exclusions	KLAdmin

6.1.4.10 FMT_SMF.1 Specification of Management Functions

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

Managing the FDE Access Control SFP attributes used to make explicit access or denial based decisions,

Managing the Application Startup Control SFP attributes used to make explicit access or denial based decisions,

Managing the Device Access Control SFP attributes used to make explicit access or denial based decisions,

Managing the Web Access Control SFP attributes used to make explicit access based decisions,

Managing TSF data, as defined in FMT_MTD.1

Managing of the authentication data by an administrator or by the user associated with this data][85].

[75] [assignment: *the authorised identified roles*]
[76] [assignment: *access control SFP(s), information flow control SFP(s)*]
[77] [selection, choose one of: *restrictive, permissive, [assignment: other property]*]
[78] [assignment: *the authorised identified roles*]
[79] [assignment: *access control SFP(s), information flow control SFP(s)*]
[80] [selection, choose one of: *restrictive, permissive, [assignment: other property]*]
[81] [assignment: *the authorised identified roles*]
[82] [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*]
[83] [assignment: *list of TSF data*]
[84] [assignment: *the authorised identified roles*]
[85] [assignment: *list of management functions to be provided by the TSF*]

6.1.4.11 FMT_SMR.1 Security roles

FMT_SMR.1.1 The TSF shall maintain the roles [KLUser, KLAdmin][86,87].

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

6.1.5 Class FAV: Anti-Virus (Explicitly Stated)

6.1.5.1 FAV_ACT.1 Anti-Virus Actions

FAV_ACT.1.1 Upon detection of a virus, the TSF shall perform the action(s) specified by [KLAdmin][88]. Actions are administratively configurable for each type of scan and consist of:[

a) Disinfect,

b) Delete the file,

c) [[Block,

d) Ignore]][89]

Application note 8: Depending on malware and object type, some actions may not be applicable. The TOE should attempt alternative action or inform authorised users about the action failure.

6.1.5.2 FAV_ALR.1 Anti-Virus Alerts

FAV_ALR.1.1 The TSF shall be able to generate an audit event indicating detection of a malware. The event shall identify the object, the virus that was detected, and the action taken by the TOE.

FAV_ALR.1.2 The TSF shall send an alarm [to the specified email][90] when a virus is detected.

6.1.5.3 FAV_SCN.1 Anti-Virus Scanning

FAV_SCN.1.1 The TSF shall perform real-time, scheduled, and on-demand scans for viruses based upon known signatures.

FAV_SCN.1.2 The TSF shall perform scheduled scans at the time and frequency configured by [KLAdmin][91].

[86] KLUser and KLAdmin roles are referred to as User and Administrator respectively in [UGD].
[87] [assignment: *the authorised identified roles*]
[88] [assignment: *the authorised identified roles*]
[89] [selection: [*assignment: list of other actions*], *no other actions*]
[90] [assignment: *alarm destination*]
[91] [assignment: *the authorised identified roles*]

6.2 Security Assurance Requirements

The TOE conforms to all security assurance requirements in EAL2 as defined in CC part 3 augmented with ALC_FLR.1. The following table lists all SARs. A **bold** typeface is used to indicate that ALC_FLR.1 is an augmentation to EAL2.

Assurance Class	Assurance Components
ADV	ADV_ARC.1, ADV_FSP.2, ADV_TDS.1
AGD	AGD_OPE.1, AGD_PRE.1
ALC	ALC_CMC.2, ALC_CMS.2, ALC_DEL.1, **ALC_FLR.1**
ASE	ASE_CCL.1, ASE_ECD.1, ASE_INT.1, ASE_OBJ.2, ASE_REQ.2, ASE_SPD.1, ASE_TSS.1
ATE	ATE_COV.1, ATE_FUN.1, ATE_IND.2
AVA	AVA_VAN.2

6.3 Security Functional Requirements Rationale

The following table maps security objectives to security functional requirements, showing that each security objective is covered by at least one security functional requirement and that no security functional requirement exists that is not needed by any security objective.

Objective / SFR	O.SECURE_DATA	O.LAUNCH	O.ACCESS_DV	O.VIRUS	O.WEBACC	O.SECURE_MANAGEMENT
FCS_CKM.1(1)	X					
FCS_CKM.1(2)	X					X
FCS_CKM.4	X					
FCS_COP.1(1)	X					
FCS_COP.1(2)	X					
FCS_COP.1(3)	X					X
FCS_COP.1(4)	X					
FDP_ACC.1(1)	X					
FDP_ACC.1(2)		X				
FDP_ACC.1(3)			X			
FDP_IFC.1					X	
FDP_ACF.1(1)	X					
FDP_ACF.1(2)		X				

Objective / SFR	O.SECURE_DATA	O.LAUNCH	O.ACCESS_DV	O.VIRUS	O.WEBACC	O.SECURE_MANAGEMENT
FDP_ACF.1(3)			X			
FDP_IFF.1					X	
FIA_UAU.2	X					X
FIA_UID.2	X					X
FMT_MSA.1(1)						X
FMT_MSA.1(2)						X
FMT_MSA.1(3)						X
FMT_MSA.1(4)						X
FMT_MSA.3(1)						X
FMT_MSA.3(2)						X
FMT_MSA.3(3)						X
FMT_MSA.3(4)						X
FMT_MTD.1		X	X	X	X	X
FMT_SMF.1						X
FMT_SMR.1	X					X
FAV_ACT.1				X		
FAV_ALR.1				X		
FAV_SCN.1				X		

The following table shows what the individual security functional requirements contribute to the objective and that the requirements are sufficient to satisfy the objective.

Objective	Requirements
O.SECURE_DATA	User authentication and data encryption ensure the device data protection in case of a physical access to the device. FDP_ACC.1(1) and FDP_ACF.1 define the rules and conditions for the data and cryptographic key access. FCS_COP.1(1) describes the data encryption/decryption with the keys generated as defined in FCS_CKM.1(1). FCS_COP.1(2) defines the key encryption/decryption with the keys generated as defined in FCS_CKM.1(1). FCS_COP.1(3) defines the HMAC calculation needed for authentication verification.

Objective	Requirements
	FCS_COP.1(4) defines the key encryption using RSA with 2048-bit key. The SFR support the authentication with Token usage. FCS_CKM.1(1) requires key generation using a deterministic random number generator. FCS_CKM.1(2) describes key generation using defined Password-Based Key Derivation Function. FCS_CKM.4 ensures that keys are destroyed in a safe way. FIA_UAU.2 requires user authentication before allowing any other TSF-mediated actions. FIA_UID.2 requires user identification before allowing any other TSF-mediated actions. FMT_SMR.1 allows to associate users with the security role of authenticated user.
O.LAUNCH	FMT_MTD.1 allows the authenticated user to change settings. FDP_ACC.1(2) and FDP_ACF.1(2) define attributes and rules to be used when providing application startup control functionality.
O.ACCESS_DV	FMT_MTD.1 allows the authenticated user to change settings. FDP_ACC.1(3) and FDP_ACF.1(3) define attributes and rules to be used when providing device access control functionality.
O.WEBACC	FMT_MTD.1 allows the authenticated user to change settings. FDP_IFC.1 and FDP_IFF.1 define attributes and rules to be used when providing web access control functionality.
O.VIRUS	FAV_SCN.1 defines anti-virus scanning used to detect viruses. FAV_ACT.1 defines actions that the TOE is attempting on detected virus. FAV_ALR.1 defines alerts and audit events generated to inform the TOE users of detected viruses. FMT_MTD.1 allows the authenticated user to change settings.
O.SECURE_ MANAGEMENT	TOE management operations have to be forbidden for unauthorised individuals to protect the TSF and TSF data. FCS_COP.1(3) defines the HMAC calculation needed for authentication verification. FCS_CKM.1(2) describes key generation using defined Password-Based Key Derivation Function. FIA_UAU.2 requires user authentication before allowing any other TSF-mediated actions. FIA_UID.2 requires user identification before allowing any other TSF-mediated actions. FMT_SMR.1 allows to associate users with the security role of authenticated user. FMT_MSA.1(1) and FMT_MTD.1 restrict the password modification possibility to authenticated administrator. FMT_MSA.1(2) and FMT_MTD.1 restrict the possibility of application metadata and properties modification to authenticated administrator. FMT_MSA.1(3) and FMT_MTD.1 restrict the target device type, device bus, device properties modification possibility to authenticated administrator. FMT_MSA.1(4) and FMT_MTD.1 restrict the web site address and content modification possibility to authenticated administrator. FMT_MSA.3(1) specifies the initial encrypted disk(s) setting by administrator via administration server or authenticated administrator.

Objective	Requirements
	FMT_MSA.3(2) specifies the initial application metadata and properties setting by administrator via administration server or authenticated administrator. FMT_MSA.3(3) specifies the initial target device type, device bus, device properties setting by administrator via administration server or authenticated administrator. FMT_MSA.3(4) specifies the initial web site address and content setting by administrator via administration server or authenticated administrator. FMT_MTD.1 allows the authenticated user or administrator to change the password and TSF data. FMT_SMF.1 describes the security functions can be used to ensure the secure operation of the TOE.

6.3.1 Security Functional Requirements Dependencies

The table lists the dependencies for each Security Functional Requirement (SFR) and shows by which SFRs they are met.

SFR	Required Dependencies	Met/fulfilled by
FCS_CKM.1(1)	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4	FCS_COP.1(1) and FCS_COP.1(2) FCS_CKM.4
FCS_CKM.1(2)	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4	FCS_COP.1(3) FCS_CKM.4
FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1(1) and FCS_CKM.1(2)
FCS_COP.1(1)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	FCS_CKM.1(1) FCS_CKM.4
FCS_COP.1(2)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	FCS_CKM.1(1) FCS_CKM.4
FCS_COP.1(3)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	FCS_CKM.1(2) FCS_CKM.4
FCS_COP.1(4)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	The dependency refers to the question, how the TOE gets the cryptographic keys for the cryptographic operation. This is fulfilled during the TOE initialisation. Pre-generated Key is obtained from token FCS_CKM.4
FDP_ACC.1(1)	FDP_ACF.1	FDP_ACF.1(1)
FDP_ACC.1(2)	FDP_ACF.1	FDP_ACF.1(2)
FDP_ACC.1(3)	FDP_ACF.1	FDP_ACF.1(3)
FDP_IFC.1	FDP_IFF.1	FDP_IFF.1
FDP_ACF.1(1)	FDP_ACC.1	FDP_ACC.1(1)

SFR	Required Dependencies	Met/fulfilled by
	FMT_MSA.3	FMT_MSA.3(1)
FDP_ACF.1(2)	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1(2) FMT_MSA.3(2)
FDP_ACF.1(3)	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1(3) FMT_MSA.3(3)
FDP_IFF.1	FDP_IFC.1 FMT_MSA.3	FDP_IFC.1 FMT_MSA.3(4)
FIA_UAU.2	FIA_UID.1	FIA_UID.2 hierarchical to FIA_UID.1
FIA_UID.2	none	none
FMT_MSA.1(1)	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	FDP_ACC.1(1) FMT_SMR.1 FMT_SMF.1
FMT_MSA.1(2)	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	FDP_ACC.1(2) FMT_SMR.1 FMT_SMF.1
FMT_MSA.1(3)	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	FDP_ACC.1(3) FMT_SMR.1 FMT_SMF.1
FMT_MSA.1(4)	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	FDP_IFC.1 FMT_SMR.1 FMT_SMF.1
FMT_MSA.3(1)	FMT_MSA.1 FMT_SMR.1	FMT_MSA.1(1) FMT_SMR.1
FMT_MSA.3(2)	FMT_MSA.1 FMT_SMR.1	FMT_MSA.1(2) FMT_SMR.1
FMT_MSA.3(3)	FMT_MSA.1 FMT_SMR.1	FMT_MSA.1(3) FMT_SMR.1
FMT_MSA.3(4)	FMT_MSA.1 FMT_SMR.1	FMT_MSA.1(4) FMT_SMR.1
FMT_MTD.1	FMT_SMR.1 FMT_SMF.1	FMT_SMR.1 FMT_SMF.1
FMT_SMF.1	none	none
FMT_SMR.1	FIA_UID.1	FIA_UID.2 hierarchical to FIA_UID.1
FAV_ACT.1	none	none
FAV_ALR.1	none	none
FAV_SCN.1	none	none

6.4 Security Assurance Requirements Rationale

EAL2 has been chosen to establish a sufficient level of confidence in the security offered by the TOE. It has been augmented with ALC_FLR.1 to ensure that customers can report flaws and that those flaws can be corrected according to flaw remediation procedures.

Since ALC_FLR.1 does not have dependencies and EAL2 satisfies its own dependencies, EAL2 augmented with ALC_FLR.1 is consistent with regard to its dependencies.

7 TOE Summary Specification

This section contains description of how the TOE meets all the SFRs.

The following table helps associate SFR with relevant description.

SFR	SFR name	TOE Security Functionality
FCS_CKM.1(1)	Cryptographic key generation (DEK/MK)	SF_FDE
FCS_CKM.1(2)	Cryptographic key generation (User key)	SF_FDE
FCS_CKM.4	Cryptographic key destruction	SF_FDE
FCS_COP.1(1)	Cryptographic operation (Data Encryption/Decryption)	SF_FDE
FCS_COP.1(2)	Cryptographic operation (Key Encryption/Decryption)	SF_FDE
FCS_COP.1(3)	Cryptographic operation (HMAC calculation)	SF_FDE
FCS_COP.1(4)	Cryptographic operation (RSA Key Encryption)	SF_FDE
FDP_ACC.1(1)	Subset access control (FDE)	SF_FDE
FDP_ACC.1(2)	Subset access control (ASC)	SF_ASC
FDP_ACC.1(3)	Subset access control (DAC)	SF_DAC
FDP_IFC.1	Subset information flow control (WAC)	SF_WAC
FDP_ACF.1(1)	Security attribute based access control (FDE)	SF_FDE
FDP_ACF.1(2)	Security attribute based access control (ASC)	SF_ASC
FDP_ACF.1(3)	Security attribute based access control (DAC)	SF_DAC
FDP_IFF.1	Simple security attributes (WAC)	SF_WAC
FIA_UAU.2	User authentication before any action	SF_IA
FIA_UID.2	User identification before any action	SF_IA
FMT_MSA.1(1)	Management of security attributes (FDE)	SF_MGMT
FMT_MSA.1(2)	Management of security attributes (ASC)	SF_MGMT
FMT_MSA.1(3)	Management of security attributes (DAC)	SF_MGMT
FMT_MSA.1(4)	Management of security attributes (WAC)	SF_MGMT
FMT_MSA.3(1)	Static attribute initialisation (FDE)	SF_MGMT
FMT_MSA.3(2)	Static attribute initialisation (ASC)	SF_MGMT
FMT_MSA.3(3)	Static attribute initialisation (DAC)	SF_MGMT
FMT_MSA.3(4)	Static attribute initialisation (WAC)	SF_MGMT
FMT_MTD.1	Management of TSF data	SF_MGMT
FMT_SMF.1	Specification of management functions	SF_MGMT
FMT_SMR.1	Security roles	SF_MGMT
FAV_ACT.1	Anti-virus actions	SF_AVP
FAV_ALR.1	Anti-virus alerts	SF_AVP
FAV_SCN.1	Anti-virus scanning	SF_AVP

7.1 Full Disk Encryption Functionality (SF_FDE)

7.1.1 Cryptographic key generation (DEK/MK)

During installation of the TOE and initial encryption of the devices data (initialisation), a deterministic random number generator is used for the generation of the needed AES cryptographic keys. This applies to the following keys:

- Data Encryption Key (DEK),
- Master Key (MK).

DEK and MK are generated during the TOE initialisation and if the keys have to be changed.

Keys are generated by a TOE crypto library using Hash_DRBG algorithm according to NIST SP 800-90A with SHA-256.

For the XTS-AES-256 the TSF generates two AES keys, each 256 bit long in accordance with the mentioned algorithm.

SFRs that are met: FCS_CKM.1(1).

7.1.2 Cryptographic key generation (User key)

During installation of the TOE and initial encryption of the devices data (initialisation), a deterministic random number generator is used for the generation of the needed AES cryptographic keys (User Keys).

User Keys are generated during the TOE installation in accordance with the available accounts and if the keys have to be changed.

Keys are generated by TOE crypto library by Password-Based Key Derivation Function 2 (PBKDF2) with HMAC-SHA256, 10.000 iteration value, 256-bit salt and password as input as required by NIST SP 800-132, Option 2a. This key is later used during user authentication with username/password method.

SFRs that are met: FCS_CKM.1(2).

7.1.3 Cryptographic key destruction

Following recommended practice, the TOE overwrites cryptographic keys in memory with zeroes when it no longer needs them.

SFRs that are met: FCS_CKM.4.

7.1.4 Cryptographic operations

Cryptographic operations are done by TOE crypto library is way required be relevant standard, mentioned in each SFR iteration.

Cryptographic operation (Data Encryption/Decryption) is providing symmetric encryption/decryption of data using XTS-AES-256 algorithm for disk data during disk I/O operations.

Cryptographic operation (Key Encryption/Decryption) is used to encrypt/decrypt encryption keys stored on HDD during user authentication and creation of new users.

Cryptographic operation (HMAC calculation) is used during boot process to verify user credentials.

Cryptographic operation (RSA Key Encryption) is used by TOE for user authentication using Tokens instead of username/password combination on new user creation stage. RSA public key is obtained from Token and used to encrypt key for future authentication use.

SFRs that are met: FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4)

7.1.5 Full Disk Encryption security functionality

The TOE aims to protect user data on a disk drive from unauthorised access through “stolen laptop” scenario. This means that user data protection relies not on OS mechanisms, that can be bypassed if physical access to disk is obtained, but on strong encryption and user authentication data.

So, user (and system) data exists on a hard drive only in fully encrypted state, and successful decryption possible only with valid user authentication data (username/password or Token/Pin pairs).

To achieve this the TOE relies on cryptographic functionality described above.

This function performs the user authentication after power on before the operating system is booted. The function is implemented in a special Pre-Boot Authentication (PBA) software module that is running in UEFI/BIOS environment. After successful user authentication, the control is granted to the operating system loader.

The function supports two kinds of user authentications depending on the corresponding configurations: based on the username/password and with help of the Token (PIN).

[PARAGRAPH TEXT WAS REMOVED DURING SECURITY TARGET SANITISING FOR PUBLICATION]

If the authentication failed, the next attempt is possible as long as the defined attempt amount was not reached; otherwise the authentication function is blocked for the user.

After successful authentication, the function decrypts the user policy data with the Master Key and evaluates the policy. If the management tasks (password changing) have to be performed, the function starts the corresponding management function.

[PARAGRAPH TEXT WAS REMOVED DURING SECURITY TARGET SANITISING FOR PUBLICATION]

For read operation OS makes read request from file system driver to physical disk driver, the TOE relays this request. Physical disk driver reads relevant part of the disk and transmits the data to a TOE driver, the TOE then decrypts it using DEK and MK from memory and then passes decrypted data to file system driver for OS usage.

For write operation, OS passes plaintext data to file system driver for write operations, the TOE intercepts these requests, encrypts this data and passed encrypted data to physical disk driver to perform physical operation.

Thus, the TOE makes sure there is no unencrypted data on a disk drive at any given moment.

SFRs that are met: FDP_ACC.1(1), FDP_ACF.1(1).

7.2 Application Startup Control (SF_ASC)

Application Startup Control manages the startup of applications on users' computers. This allows implementing a corporate security policy when using applications. Application Startup Control also reduces the risk of computer infection by restricting access to applications.

Application startup control functionality of the TOE is based on filter driver interception mechanisms, where the TOE intercepts all processes being started in OS on a kernel level. When OS or application executes new application (process), the TOE scans the application being run, (or script being executed) to get process properties and metadata.

This can be application hash sum, path, application properties, application’s digital signature (certificate) parameters, application category (out of scope of evaluation), active user.

These metadata then are compared to active Application Startup control policy (set of rules).

Based on rules parameters (permissive or non-permissive), subject that executes the application, and conditions defined the TOE makes decision whether to allow execution, or prevent it. That decision is enforced via driver to system kernel, preventing unauthorised operations.

The Application Control operates in two modes: Denylist (denies the startup of certain applications listed in Application Control forbidding rules and allows the startup of all other applications) and Allowlist (allows the startup of certain applications listed in Application Control permissive rules and blocks the startup of all other applications).

The certificate-based rules depend on the certificate validity; hence they are not recommended to be used in Denylist mode. The certificate-based rule will not be enforced if the certificate is not valid.

SFRs that are met: FDP_ACC.1(2), FDP_ACF.1(2).

7.3 Device Access Control (SF_DAC)

Device control functionality of the TOE is based on filter driver interception mechanisms, where the TOE intercepts all file data operations in OS on a kernel level.

When OS initiates a data transmission to or from the attached device, the TOE collects operation properties and metadata.

This can be type of device, the bus or the device's individual serial number, type of operations (read or write), active user, operation time.

These metadata then are compared to active device control policy (set of rules).

Based on rules parameters (permissive or non-permissive), subject that executes the operation, and conditions defined in the rules, the TOE makes decision whether to allow operation, or prevent it.

That decision is enforced via driver to system kernel, preventing unauthorised execution.

SFRs that are met: FDP_ACC.1(3), FDP_ACF.1(3).

7.4 Web Access Control (SF_WAC)

Web control functionality of the TOE is based on filter driver interception mechanisms, where TOE intercepts all data operations in OS on a kernel level.

When OS initiates a data transmission to or from the network, the TOE collects operation properties and metadata.

This can be type of target address, operation time, active user.

These metadata then are compared to active web control policy (set of rules).

Based on rules parameters (permissive or non-permissive), subject that executes the operation, and conditions defined in the rules, the TOE makes decision whether to allow operation, or prevent it.

That decision is enforced via driver to system kernel, preventing unauthorised access to resources.

The TOE has the following 4 control options for each rule:

- Any content. All web traffic is blocked or permitted. (URL server name mask can be attached to this rule).
- By content categories. URLs accessed are checked against base of known categorised web sites. URLs are then blocked if URL category matches the one selected in the rule. Content categories are described in User Manual (this rule type is out of scope of evaluation).
- By type of data. When object is being downloaded from network its source URL is matched against known groups of file extensions.
- By content categories and types of data. Two above mentioned checks together (this rule type is out of scope of evaluation).

Multiple rules can be defined with relative priority, they will be applied based on their priority.

SFRs that are met: FDP_IFC.1, FDP_IFF.1.

7.5 Identification and authentication (SF_IA)

TOE performs user identification and authentication during pre-boot. User credentials are verified against stored values (as described in details section 7.1.5) and disk decryption operations are available to authenticated users.

When user tries to perform operations that require KLAdmin role authorisation additional authentication dialogue is shown via GUI or command line interface and additional authentication is performed when user issues command or tries to save changed settings TOE.

TOE checks provided credentials against stored values (calculates hash values for provided credentials and compares it with stored values) before accepting changes/commands, that require authentication.

Hash values are calculated using SHA256 algorithms with salt to provide protection against recovery using rainbow tables.

When KLAdmin uses KSC to manage TOE authentication is performed on program API level by checking for valid digital signature of program module that relays the management commands to TOE.

SFRs that are met: FIA_UAU.2, FIA_UID.2

7.6 Security management (SF_MGMT)

7.6.1 Security Roles

Specifics of this TOE is that it provides services to all users in the environment. Thus, the TOE has two distinct roles: KLUsers and KLAdmin.

Users are associated with KLUser role when they perform authentication during pre-boot (see section 7.5 above)

Users are associated with KLAdmin Role when they provide valid credentials (user name and password) when prompted by the TOE when action that is restricted to KLAdmin role is initiated, like modification of policies' security attributes, TFS data, authentication data. These commands will not be accepted until user provides valid credentials for KLAdmin role.

This is not related to OS user authorisation and rights. All actions done through KSC management interface are considered to be from KLAdmin.

KLUser and KLAdmin roles are referred to as User and Administrator respectively in [UGD].

SFRs that are met: FMT_SMR.1.

7.6.2 Management of policies security attributes, TSF data and authentication data

The TOE operates based on rules, access policies and other TOE data, such as KLAdmin password, encryption keys, task settings, default actions and values for Access Control Policies.

All TOE policies and rules are stored in Windows registry file and are read by TOE when necessary.

KLAdmin can manage the TOE by using relevant interfaces – via GUI, command prompt or using Kaspersky Security Center (not part of the TOE). When KLAdmin tries to change data or apply new policies, authorisation is required (see section 7.5).

Please note that FDE Access Control SFP can be managed only through Kaspersky Security Center (not part of TOE).

KLAdmin can set default values for TOE Access control SFPs policies and change them from permissive to restrictive mode (recommended after all necessary control policies are defined).

After TOE accepts new settings and policies, stored TOE data are modified accordingly by the TOE. The TOE data are being protected by the TOE from modification by all other subjects, except the TOE itself.

Encryption Keys are stored separately in a special section of drive in encrypted form, those are being modified by the TOE automatically, when User change password, or new users are being added to the system.

KLAdmin can modify certain authentication data (KLUsers passwords for pre-boot authentication, KLAdmin password for management functions).

Users with KLUser role can change their passwords for pre-boot authentication during pre-boot.

SFRs that are met: FMT_MSA.1(1), FMT_MSA.1(2), FMT_MSA.1(3), FMT_MSA.1(4), FMT_MSA.3(1), FMT_MSA.3(2), FMT_MSA.3(3), FMT_MSA.3(4), FMT_MTD.1, FMT_SMF.1.

7.7 Anti-Virus protection (SF_AVP)

7.7.1 Anti-Virus Scanning

Anti-virus functionality protects system from malicious software using wide range of technics, including real-time file access monitor, on-demand on on-schedule scans of system critical areas. Incoming network objects (including mail and web objects) are also scanned. It employs signature-based, heuristic and behaviouristic methods of detection of malicious object, and remediation functionality.

The TOE provides real-time protection and scheduled and on-demand scans of certain areas as configured by KLAdmin (see FMT_MTD.1) or triggered by KLUser.

Real-time protection functionality is based on filter driver mechanism, the TOE intercepts all data operations in OS on a kernel level.

When the TOE detects a file access operation, it intercepts that and passes file object to AV engine for scanning.

For network interaction – all data transmitted via network channels are also being passed to AV engine.

For executed process all actions initiated by process are passed to AV engine to be matched to existing patterns of malicious behaviour.

For scheduled or on-demand scan the TOE also uses its drivers to ensure reliable access to objects on disk or in memory. Areas and objects to be scanned are determined by scan task properties. The TOE reads relevant objects and passes them to AV engine for scanning.

For all these scan types AV engine conducts a set of analytic routines on the file, including unpacking, emulation, static detection and set of heuristic procedures providing a detection conclusion and proposed actions.

SFRs that are met: FAV_SCN.1.

7.7.2 Anti-Virus Actions

When the AV engine provides a detection conclusion, the TOE compares received conclusion with scan settings that define possible exclusions, and actions to be taken on detected objects.

The TOE can be configured to take action as described in table below.

Action	Description
Disinfect	This action can be applied only to the files that were infected by infector-type malware and malicious code can be removed, and this removal procedure is available to the TOE.

Action	Description
	It must be noted that the full recovery of an infected object to its original state (as before the infection) is not guaranteed.
Delete	The TOE deletes a detected object.
Block	The TOE does not take any action on a detected object but blocks system access to the detected object. E.g., file execution or file read operation would fail.
Ignore *(an implicit action)*	If a detected object matches the exclusion settings configured by KLAdmin no action is taken with the detected object.

The TOE tries a sequence of actions as configured:

- first disinfect, then delete it if disinfection fails; or
- first disinfect, then block if disinfection fails; or
- just block.

SFRs that are met: FAV_ACT.1.

7.7.3 Anti-Virus Alerts

When a malicious object is detected and processed, the TOE generates relevant audit records, also pop-up notifications or e-mail alerts can be configured.

Alerts can be obtained through all management interfaces, that are accessible to Administrators and Users – GUI, command-line interface or through Kaspersky Security Center administration console (KSC itself is not a part of the TOE).

Such alerts identify detected objects, detection conclusion, and actions taken by the TOE.

SFRs that are met: FAV_ALR.1.

8 References

Reference	Document
[CCp1]	Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; CCMB-2017-04-001, Version 3.1, Revision 5, April 2017
[CCp2]	Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components; CCMB-2017-04-002, Version 3.1, Revision 5, April 2017
[CCp3]	Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components; CCMB-2017-04-003, Version 3.1, Revision 5, April 2017
[CEM]	Common Methodology for Information Technology Security Evaluation, CCMB-2017-04-004, Version 3.1, Revision 5, April 2017
[IEEE P1619]	IEEE P1619™/D16, Standard for Cryptographic Protection of Data on Block-Oriented Storage Devices, IEEE Computer Society Committee, May 2007
[FIPS 180-4]	FIPS PUB 180-4, Secure Hash Standard (SHS) / National Institute of Standards and Technology (NIST), August 2015
[FIPS 197]	FIPS PUB 197, Advances Encryption Standard (AES) / National Institute of Standards and Technology (NIST), 26th November 2001
[FIPS 198-1]	FIPS PUB 198-1, The Keyed-Hash Message Authentication Code (HMAC) / National Institute of Standards and Technology (NIST), July 2008
[NIST SP 800-38A]	NIST Special Publication 800-38A, Recommendation for Block Cipher Modes of Operation: Methods and Techniques / National Institute of Standards and Technology (NIST), December 2001
[NIST SP 800-38E]	NIST Special Publication 800-38E, Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Storage Devices / National Institute of Standards and Technology (NIST), January 2010
[NIST SP 800-90A]	NIST Special Publication 800-90A Revision 1, Recommendation for Random Number Generation Using Deterministic Random Bit Generators / National Institute of Standards and Technology (NIST), June 2015
[NIST SP 800-132]	NIST Special Publication 800-132, Recommendation for Password-Based Key Derivation, Part 1: Storage Applications / National Institute of Standards and Technology (NIST), December 2010
[PKCS#1, v2.2]	PKCS #1: RSA Cryptography Specifications Version 2.2 / Internet Engineering Task Force (IETF), November 2016
[UGD]	Kaspersky Endpoint Security for Windows. User Manual. Version 2.01
[UGDAA]	Kaspersky Endpoint Security for Windows. User Manual. Addendum A. Version 2.04

kaspersky

www.kaspersky.com/
www.securelist.com