SanDisk Cruzer Enterprise FIPS edition

Security Target

EAL2 Augmented with ALC_FLR.1

Version 1.1

September 24, 2009

Document History

Version	Date	Author	Description
1.0	03-Aug-09	Jussipekka Leiwo	Final version.
1.1	24-Sep-09	Ken Hendrie	Minor grammatical changes

Table of Contents

List of Tables

1 Document Introduction

1 This section provides preliminary information and various documenting conventions which do not formally constitute elements of a Security Target but which are used to present the Security Target to the reader, as well as other information which aims at assisting the reader in understanding the ST and the TOE it describes.

1.1 Document conventions

2 Part 2 of the Common Criteria defines an approved set of operations that may be applied to the statement of security functional requirements. Following are the approved operations and the document conventions as used within this ST to depict their application:

- **Assignment.** The assignment operation provides the ability to specify an identified parameter within a requirement. Assignments are depicted using bolded text and are surrounded by square brackets as follows [**assignment**].
- **Selection.** The selection operation allows the specification of one or more items from a list. Selections are depicted using bold italics text and are surrounded by square brackets as follows [***selection***].
- **Refinement.** The refinement operation allows the addition of extra detail to a requirement. Refinements are indicated using bolded text, for **additions**, and strike-through, for ~~deletions~~.
- **Iteration.** The iteration operation allows a component to be used more than once with varying operations. Iterations are depicted by placing a letter at the end of the component identifier as follows FDP_IFF.1a and FDP_IFF.1b.

3 Additionally, the following conventions are used:

- **Application note** is an informal explanation by the author of the ST to highlight and explain an unusual or otherwise exceptional wording either in the requirements for an artefact of the ST or in the statement of a specific artefact in the ST.

1.2 Terminology

4 The essential terminology used in this ST is described in Table 1.

Table 1 – Terminology

Term	Description
AES	Advanced Encryption Standard, a symmetric cryptosystem
Authentication state	A flag maintained by the TOE to indicate whether the user has been successfully authenticated
CMC Software	Central Control and Management Software for administering the TOE and for recovering it from the Lockdown mode.
Cryptanalysis	The act of subjecting encrypted data to an attempt to discover the corresponding plaintext without knowledge of the relevant cryptographic key, or to an attempt to discover the cryptographic key used for producing the cipher text.
DRNG	Deterministic Random Number Generator

Term	Description
Host PC	The PC in which the TOE is inserted through the USB port and through which the human user accesses the files persistently stored on the TOE.
Human user	The human user with a legitimate right to access the TOE through the host PC or CMC Software
Lifecycle state	A flag maintained by the TOE to indicate whether it is in the operational mode.
Lockdown mode	The security mode the TOE enters in case of the number of consecutive failed authentication attempts exceeds a pre-defined threshold, indicating an attempted password guessing attack.
Lockdown state	A flag maintained by the TOE to indicate the Lockdown mode.
Login Device Window	The software executed on the host PC upon authentication of the end user to the TOE.
Man in the middle attack	An attack in which the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection when in fact the entire conversation is controlled by the attacker.
MAXNOA	The maximum number of allowed subsequent failed authentication attempts prior to the TOE entering the Lockdown mode.
NOA	The current number of subsequent failed authentication attempts since the last successful authentication.
Password guessing attack	An attempt of a user other than the human user or a software module acting on behalf of that user to guess a password that matches the RAD and which renders the user indistinguishable from the human user of the TOE and, therefore, grants the user access to the TOE.
RAD	Reference Authentication Data, the reference password against which the human user of the host PC is authentication. RAD is stored in the persistent memory of the TOE
Root key	The cryptographic key used for encrypting the keys used for encrypting user data and reference authentication data when stored persistently on the TOE.
SDK	Software Development Kit
Security state	The collective set of the values of parameters coordinating the security of the TOE.
Setup Wizard	The software executed on the host PC upon initialization of the TOE in case of the absence of the CMC Software.

Term	Description
SHA-1	Secure Hash Algorithm 1, a cryptographically secure hash function
USB Start-up window	The software executed on the host PC in the start-up of the TOE.
User data	The data persistently stored as files in a secure manner by the TOE.
VAD	Verification Authentication Data, the user entered password compared by the TOE to the RAD to authenticate the human user of the host PC.

1.3 References

[1] Common Criteria for Information Technology Security Evaluation Part 1: Introduction and General Model, version 3.1 Revisions 1, September 2006, CCMB-2006-09-001.

[2] Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, version 3.1 Revisions 2, September 2007, CCMB-2007-09-002.

[3] Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, version 3.1 Revision 2, September 2007, CCMB-2007-09-003.

[4] Cruzer Enterprise CMC, Setup and Deployment Guide Revision 2.1. Document No. IG-CMC-0807-22, January, 2008. SanDisk Corporation.

1.4 Document organization

5 This document is organized into the following sections:

- Section 1 provides introductory and preliminary explanations and document conventions to assist readers in understanding this ST.
- The assurance families required for fulfilling assurance class ASE (ST Evaluation) at EAL2, excluding the rationales, are covered as follows:
  - i) ASE_CCL.1 (Conformance claims) in Section 3.
  - ii) ASE_ECD.1 (Extended components definition) In Section 6.
  - iii) ASE_INT.1 (ST introduction) in Section 2.
  - iv) ASE_OBJ.2 (Security objectives) in Section 5.
  - v) ASE_REQ.2 (Derived security requirements) in Section 7.
  - vi) ASE_SPD.1 (Security problem definition) in Section 4.
  - vii) ASE_TSS.1 (TOE summary specification) in Section 8.
- The rationales as all presented centrally in Section 9.

2 ST Introduction

6 This section identifies the ST and describes the TOE in a narrative way.

2.1 ST Reference

7 The ST Reference that uniquely identifies the Security Target is a combination of ST Title and ST Version, the values of which are stated in Table 2.

Table 2 – ST identification information

ST Title	SanDisk Cruzer Enterprise FIPS edition EAL2 + [ALC_FLR.1] Security Target
ST Version	1.1, dated September 24, 2009

2.2 TOE Reference

8 The TOE Reference that identifies the TOE is a combination of TOE Version details, TOE Evaluation Assurance Level (EAL), and CC Version Identification, the values of which are stated in Table 3.

Table 3 – TOE identification information

TOE Version	SanDisk Cruzer Enterprise FIPS edition, consisting of the Firmware version v6.612 or v6.615.
EAL	EAL2 Augmented with ALC_FLR.1
CC Version Identification	Common Criteria for Information Technology (IT) Security Evaluation, Version 3.1 as stated in [1], [2], and [3].

2.3 TOE Overview

2.3.1 Usage and major security features of the TOE

9 Cruzer Enterprise FIPS Edition is a security product that caters to the security requirements of government agencies and financial institutions. It is principally an encrypted USB drive used for securely storing files on the FLASH memory of the device.

10 Instead of relying on end users to secure individual files on their own discretion, the SanDisk Cruzer Enterprise FIPS Edition stores all files in a secure partition implementing a 256-bit hardware-based AES encryption and decryption of files. The files are persistently stored on the device encrypted, and only when requested by an authorized user are they decrypted and released to the host PC.

11 Since the cryptographic key never leaves the device, they are safe from software attacks. The physical device housing the hardware and software is sealed with epoxy glue so that the cryptographic keys stored on the drive is also protected from physical tampering attempts and any such attempt is visually detected by the user.

12 In addition to the symmetric AES, the TOE implements digital signature verification using 1024-bit RSA. Digital signatures are used for verifying the authenticity of firmware upgrades intended for the TOE and for establishing a secure channel between the device and the CMC software.

13 End users are authenticated with a password mechanism. If the end user is not successfully authenticated (i.e. no correct password is entered), access to the decryption function for recovering the encrypted files is denied. Any occurrence of several consecutive failed authentication attempts is interpreted as a password guessing attack and causes the TOE to enter a Lockdown mode. In Lockdown mode, all access requests are denied.

14 The Cruzer® Enterprise FIPS Edition implements the security features and characteristics summarized in Table 4.

Table 4 – TOE Security features and characteristics

Feature	Characteristics
Cryptography	256-bit AES keys are generated on the device. The cryptographic key is never exported from the token. Data decryption is dynamic. Only the file required by the host PC is decrypted and sent to the PC. The device also implements secure hashing for password verification and AES key generation using SHA-1. SHA-1 is also used for storing the reference authentication data in an irreversible way. Verification of 1024-bit RSA digital signatures is implemented to establish a secure channel between the TOE and CMC software and for verifying the authenticity of firmware upgrades.
Protection of secrets	The secrets – cryptographic keys and reference authentication data – persistently stored on the device are protected physically and logically from any unauthorized modification and disclosure deemed as a violation of the integrity and confidentiality of those secrets.
Authentication and access control	The user of the host PC is authenticated prior to access being granted to the software running on the host PC to access the encrypted data persistently stored on the device. The device maintains an authentication state and a lifecycle state on which the granting and denying of access requests to protected data is based. The authentication state is set upon each successful authentication and cleared upon a logout command and when the device is powered up. The lifecycle state is determined by the existence of a root key: if the root key does not exist, the TOE is not initialized and is in the initialization state. If the root key exists, the TOE is in the operational state. If the user is not successfully authenticated or if the device is in a Lockdown state, all access requests to the data stored on the device are denied. TOE initialization, including the generation of the AES key, is only allowed if the lifecycle state is not set to indicate an operational stage.
Password security	Authentication is based on a password the owner of the device creates during the initialization of the device. Each time the device is inserted into a USB slot, the USB Start-up window appears and once the start-up is complete, a Login Device Window appears for the user to enter the password. The inserted password is sent to the USB Token in which the TOE software compares it to the stored reference password and depending on the comparison result, sets the authentication state or increases the counter keeping track of the number of consecutive authentication failures. The counter (NOA – Number of Attempts) is used for determining a possible password guessing attack. If the password is known to the user, it can at any time be changed through the Cruzer Enterprise Settings dialogue. A lost or forgotten password cannot be recovered and either external management software is needed to reset the password or the device has to be reformatted and re-initialized. In the latter case, all data previously stored on the TOE is lost.
Lock-down	The Cruzer Enterprise maintains a counter of consecutive, failed

Feature	Characteristics
mode	authentication attempts - NOA. If that number exceeds a pre-set threshold (MAXNOA – the maximum number of attempts), the TOE enters a Lockdown mode in which all access requests are blocked. If in the Lockdown mode, the only way to recover the TOE is to reformat and re-initialize the device in which case all data previously stored on the TOE is lost.
Secure access	The CMC Software can be used for securely administering the TOE. The legitimacy and authenticity of the CMC Software can be established by the TOE and access to the TOE restrict only to the legitimate CMC. Furthermore, the TOE is capable of establishing a trusted channel between itself and any dedicated application being executed on the host PC assuming that the application is specifically modified using a specific SDK. This prevents man in the middle attacks whereby malicious software residing in the host PC attempts to listen to the sensitive data being communicated between the TOE and the application being executed on the host PC.
Secure firmware upgrade	The Cruzer Enterprise implements a mechanism for firmware upgrading, and ensures that only authentic upgrades are implemented on the device. An authentic upgrade can be detected by verifying the digital signature attached to the upgrade. Only upgrades signed by SanDisk are deemed authentic.

15 The Cruzer® Enterprise FIPS Edition is available in 1, 2, 4 and 8GB configurations.

2.3.2 TOE Type

16 The TOE is not of any type defined in CC Part 1. Instead, TOE is a USB Token for secure storage of files. The TOE is also capable of authenticating the end user and encrypting and decrypting the files stored on the non-volatile memory of the token. Upon reading of the previously encrypted files, only the file that the end user chooses to access is decrypted, all other files remain encrypted on the non-volatile memory of the token.

2.3.3 Hardware, software and firmware required by the TOE

17 The TOE is a USB token that requires a host PC for power and connectivity. The TOE communicates with the software running on the host PC through a USB bus that is provided by the host PC. Both USB 1.1 and USB 2.0 interfaces are supported.

18 The host PC must run the operating system platform that controls the USB interface. The operating system platforms supported are

- Microsoft Windows 2000 SP4 or higher[1],
- Windows XP SP1 or higher,
- Microsoft Windows XP 64-bit SP1,
- Microsoft Windows 2000 Server,
- Microsoft® Windows® 2003 server (standard and enterprise editions), and
- Windows Vista (all editions).

19 Additionally, some administrative features of the TOE are only available in the presence of the SanDisk Central Management and Control (CMC) server software. The CMC

[1] **Application note**: USB 2.0 port is required for Windows 2000 users without administrative privileges

Software is not part of the TOE but the TOE implements features to authenticate the CMC Software and establish a trusted channel between itself and legitimate CMC Software.

2.4 TOE Description

2.4.1 Physical scope of the TOE

20 The TOE is a physical device housed in a regular casing for a USB token consisting of the integrated circuitry and the related software and firmware. The integrated circuitry consists of the S2 controller together with the EEPROM, ROM, RAM and FLASH memories. The software and firmware is stored on the memories of the integrated circuitry and used for controlling the device. The integrated circuitry and the PCB on which the components are soldered, including all the memories, are encapsulated in epoxy to prevent direct probing.

21 The S2 cryptographic coprocessor provides cryptographic primitives for the TOE firmware and software to use for protecting the user data persistently stored on the non-volatile FLASH memory of the TOE. The primitives include 256-bit AES for data encryption and decryption, 1024-bit RSA for digital signature verification, ANSI X9.31 Appendix 2.4 compliant deterministic random number generation (DRNG) with non-deterministic random number generator being implemented on HW and used for generating the seed for the DRNG for AES key generation, and SHA-1 for cryptographically secure hashing.

22 In the core of the TOE is the SanDisk S2 controller. It contains the critical elements of a USB controller and is implemented based on the ARM7-TDMI 32-bit RISC processor. External communication interfaces are to the NAND flash memory used for storing user data and to the USB host providing the necessary access services to the host PC. The overall architecture of the S2 controller is illustrated in Figure 1.

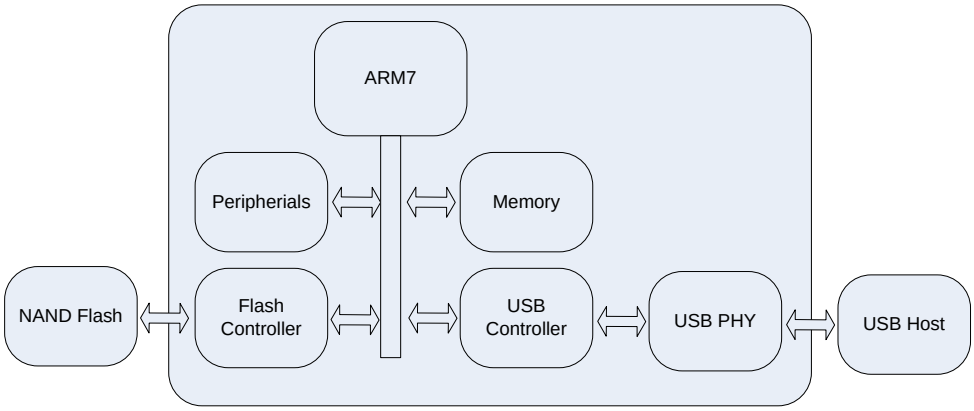

Figure 1 overall S2 controller architecture[2]

23 The software modules are either internal to the TOE or uploaded for execution on the host PC using the Auto run features of the USB tokens. The TOE software is that firmware that resides on the USB token and is not uploaded to the host PC.

24 The software modules uploaded for execution on the host PC are not part of the TOE. Neither is the host PC or the operating system or any application level software of the host PC part of the TOE.

[2] **Application note**: This figure is intentionally highly abstract but is sufficient at this stage to provide an overview of the S2 controller. A more detailed view and analysis from relevant parts shall be provided in the ADV documentation.

25 The TOE is capable of establishing the authenticity of the CMC Software and makes certain administrative functions only available to CMC Software. However, CMC Software is not part of the TOE. The TOE can also establish a trusted channel between itself and any specifically developed application being executed on the host PC. The application software is not included in the TOE, neither is the SDK provided for developers to engineer trustworthy application software to be used in conjunction with the TOE.

2.4.2 Logical scope of the TOE

26 The TOE provides the core security functionality of the SanDisk Cruzer Enterprise FIPS Edition. In the heart of the security is the cryptographic engine of the TOE. The cryptographic protection of the user data includes generation of secure 256-bit AES keys and encrypting the files stored on the memory of the TOE and decrypting those files requested by authorized users.

27 There are two types of keys relevant to the TOE: the root key and the user data encryption key. Both root key and user data encryption key are 256-bit AES keys but are used differently. The root key is used for encrypting the user data encryption key and reference authentication data (more precisely, a cryptographically secure hash of the reference authentication data) when persistently stored by the TOE.

28 The keys can be zeroized by the crypto officer of the TOE and are also erased if the memory of the TOE is formatted. In both cases, the keys are lost as is all the data encrypted with those keys. They keys are stored in the protected memory of the TOE and never leave the USB Token in any form. This allows storage of the keys only in the secure memory and helps avoid the need for storing them on the PC and performing cryptographic operations on the host PC. If the end user is successfully authenticated, the files requested by the host PC are decrypted and returned to the host PC. All other files persistently stored on the FLASH memory of the TOE remain encrypted.

29 The TOE stores reference authentication data in the FLASH memory hashed and encrypted. Reference authentication data is created by the user and is used for authenticating the user. In the initialization of the TOE, the user uses the software executed on the host PC but distributed on the USB token hosting the TOE for selecting the password. That software implements a number of quality controls for the password candidates and only accepts strong passwords. Once accepted, the password is forwarded to the TOE and transformed into the reference authentication data by hashing it using SHA-1. The reference authentication data (i.e. the hashed password) is persistently stored in the FLASH memory encrypted by AES using the root key.

30 The user authentication dialogue software distributed on the USB token and executed in the host PC using the Auto run feature upon insertion of the USB token into an available slot. The authentication dialogue requests the end user of the host PC to enter a password which becomes verification authentication data and is forwarded to the TOE for comparison. The verification authentication data may be hashed by the software running on the host PC to allow long passwords but that hashing is not in the scope of the TOE. The TOE further hashes the verification authentication data using SHA-1 and compares the result to the hash value of the decrypted reference authentication data stored on the FLASH memory. If the two match, the authentication is successful and the authentication state of the TOE is set. The authentication state remains set until the TOE receives a logout command or the USB token hosting the TOE looses power either by the host PC being shut down or the token being removed from the USB slot of the host PC.

31 Neither the password selection dialogue software nor the authentication dialogue software is part of the TOE. Rather, TOE constitutes of the S2 controller and the firmware executed thereon, and only receives input from the software running on the host PC.

32 If the verification authentication data and reference authentication data do not match, a NOA counter that keeps track of the number of consecutive authentication failures is

incremented. NOA value exceeding MAXNOA, the threshold set at the manufacturing stage of the TOE, is interpreted as a password guessing attempt which triggers the Lockdown state[3].

33 In the Lockdown state, all service requests are denied and the TOE can be recovered into an operational state only by reformatting and re-initialization of the TOE. In the latter case, all data persistently stored on the TOE is lost.

34 If the CMC Software is used for administering the TOE, the TOE is capable of authenticating a legitimate CMC Software and for establishing a SSL channel between itself and the CMC Software. The CMC software can prevent the Lockdown mode by regulating the authentication attempts so that the NOA never reaches MAXNOA but this feature is all done by the software running on the host PC without TOE's involvement.

35 The TOE can also prevent man in the middle attacks where illegitimate software attempts to listen to the data communication between the TOE and the application level software being executed on the host PC. Applications can be secured using the Software Development Kit (SDK) provided with the TOE so that they can establish a trusted channel between themselves and the TOE to prevent the man in the middle attacks. While the external applications remain outside the scope of the TOE, the TOE includes the token end functionality for establishing those channels.

36 If the authentication is successful, the authentication state of the TOE is set and the access control routines shall accept legitimate accesses to the data persistently stored on the TOE or the security attributes defined for managing the security of the TOE. Each file request is separately evaluated and if the TOE determines that the request is legitimate and that the authentication state of the TOE is set, the requested file is decrypted and returned to the software running on the host PC.

37 The internal control structures of the TOE and the security critical data are protected so that only authorized changes in the values are allowed. The TOE is housed in a tamper-evident epoxy glue so that the owner of the USB Token housing the TOE can disassemble the casing to visually inspect the circuitry and with a high likelihood detect whether the TOE has been subjected to attempted tampering.

38 The TOE can receive firmware upgrades from authorized distributors. The firmware upgrades are authenticated using 1024-bit RSA digital signatures, and only those distributions with a valid digital signature from SanDisk are accepted for installation.

3 Conformance Claims

39 The following conformance claims are made for the TOE and ST:

- **CCv3.1 Rev.2 conformant**. The ST and the TOE are Common Criteria conformant to Common Criteria version 3.1 Revision 2 defined in [1], [2] and [3].
- **Part 2 conformant.** The ST is Common Criteria Part 2 conformant.
- **Part 3 conformant.** The ST is Common Criteria Part 3 conformant.
- **Package conformant.** The ST is package conformant to the package Evaluation Assurance Level EAL2 augmented with ALC_FLR.1 as defined in [3].
- **Protection Profile conformance**. The ST does claims conformance to the following Protection Profiles: **None**.

4 Security Problem Definition

40 The TOE is concerned with the protection of the following assets enumerated in Table 5.

[3] **Application note**: the minimum allowed value is 1 and the maximum value allowed is 100 but any value in the valid range may be set during the manufacturing of the TOE.

Table 5 – Assets protected by the TOE

Identifier	Asset statement
AST.DATA	Confidentiality of the user data stored on the device.
AST.USER_AUTH	Authenticity of the human users and software accessing the data protected by the TOE.
AST.SEC _STATE	Integrity of the security state of the device, consisting of the authentication state, lifecycle state and lockdown state.
AST.LEG_ACC	Legitimacy of access to the device.

41 The subjects, some of which constitute threat agents as highlighted in the description of threats, are stated in Table 6

Table 6 – Subjects relevant to the TOE

Identifier	Subject definition
S.APPLICATION	Trusted software residing on the host PC used for administering or accessing the TOE. **Application note**: S.APPLICATION may be the CMC software or specifically modified application level software designed and engineered to establish a trusted channel between itself and the TOE.
S.UNKNOWN_SW	Any operating system or application software running on the host PC not known to have a legitimate right to initialize or administer the TOE or to gain access to the user data persistently stored on the TOE.
S.UNKNOWN_USER	The human user (owner) of the TOE, accessing the TOE through a host PC and engaging in initialization, administration and use of the TOE.

4.1 Threats

42 Threats enumerated in Table 7 are relevant to the TOE.

Table 7 – Threat statements

Identifier	Threat statement
T.AUTH_FAILURE	S.UNKNOWN_SW violates AST.USER_AUTH by succeeding in without detection falsifying a legitimate end user authentication procedure. The falsification may take place by the S.UNKNOWN_SW attempting to algorithmically guess the legitimate password or by selecting candidates from a dictionary and forwarding those guessed or selected passwords to the TOE as legitimate verification authentication data objects. If the resulting verification authentication data matches the reference authentication data stored persistently on the TOE, the TOE can not differentiate between legitimate and illegitimate users and a violation of authentication principles of the TOE results.

T.PW_GUESS	S.UNKNOWN_USER violates AST.USER_AUTH by succeeding in entering a weak (i.e. easy to guess) reference authentication data into the TOE. The malicious user in the possession of the TOE may attempt to guess the legitimate password and enter the guessed password candidates into the password dialogue to be forwarded to the TOE as verification authentication data. If the malicious user succeeds in correctly guessing the password the resulting verification authentication data matches the reference authentication data persistently stored on the TOE and the TOE cannot differentiate between legitimate and illegitimate users which results in a violation of authentication principles of the TOE. In order to prevent this from occurring, not only must the TOE be able to detect with a high likelihood that a password guessing attack is occurring (as articulated in T.AUTH_FAILURE) but also must the passwords accepted by the TOE be sufficiently strong to ensure that any passwords are not easy to guess.
T.AC_FAILURE	S.UNKNOWN_SW violates AST.LEG_ACC by succeeding in accessing protected objects. Access to the user data persistently stored on the TOE encrypted is controlled by access control measures to prevent unauthorized access. Discovery by unknown, potentially malicious software residing on the host PC of any additional access path to the protected user data would cause a loss of the legitimacy of access to the user data and result in an access control failure.
T.CRYPTO	S.UNKNOWN_SW obtains illegitimate access to AST.DATA by successfully cryptoanalysing the TOE to disclose either the cryptographic key used for encrypting the data persistently stored on the TOE or by exploiting a weakness on the cryptographic primitives or implementation thereof used for protecting the user data encrypted and persistently stored on the TOE.
T.SECSTATE_ALT	S.UNKNOWN_SW causes an unauthorized alteration in AST.SEC_STATE by succeeding in an unauthorized modification of the values of the authentication state or lockdown state of the TOE. In addition to causing an authentication or access control failure, unknown and potentially malicious software may attempt to find a way of modifying the values of the security parameters stored on the TOE and used for enforcing the security controls on the TOE. A successful, unauthorized modification of the authentication state would facilitate access to the user data without a need to bypass the access controls or masquerade as a legitimate end user as the TOE would lose the ability to differentiate between legitimate and illegitimate users. A successful, unauthorized modification of the Lockdown state of would facilitate a password guessing attack not restricted by the authentic number of consecutive failed authentication attempts resulting in a significantly increased likelihood of discovery of a correct password by the illegitimate end user or software acting on his behalf.
T.MAN_IN_THE_MIDDLE	S.UNKNOWN_SW violates AST.DATA or AST.LEG_ACC by successfully masquerading as S.APPLICATION or by successfully intercepting the communication between S.APPLICATION and the TOE. This threat concerns with the classical man in the middle attack. As the TOE is inserted to the USB port of a host PC running a commercial, general purpose operating system there are no guarantees that the host PC is free of malicious software attempting to intercept the communication between the TOE and the

	application level software used for accessing or administering the TOE. A successful man in the middle attack would allow the malicious software to learn the content of user data protected by the TOE or alter the content of communication between the application level software and the TOE. This would result in a violation of the confidentiality of user data protected by the TOE either directly by direct reading or indirectly through the malicious software triggering an unauthentic state of the TOE through illegitimate administrative actions. Application developers may use the SDK toolkit for securing the applications against man in the middle attacks. The SDK provides a library for routines to establish a trusted channel between the application and the TOE. While neither the specifically secure applications nor the SDK are part of the TOE, the TOE includes the corresponding functionality to support the trusted channels application level software requests.
T.FW_UPGRADE	S.UNKNOWN_SW violates AST.LEG_ACC by succeeding in forwarding to the TOE a firmware upgrade indistinguishable from a legitimate firmware upgrade. As the TOE provides a feature for the developer to issue firmware upgrades once the TOE is issued, there is an obvious need to ensure that only legitimate and authentic upgrades are accepted by the TOE. A successful installation of an unauthorized upgrade would enable a threat agent to execute any software on the TOE, rendering the security of the TOE undeterminable, hence causing a total inability of the TOE to preserve its security objectives.

4.2 Organizational security policies

43 The organizational security policies enumerated in Table 8 are relevant to the TOE.

Table 8 – Organizational Security Policies

Identifier	OSP statement
OSP.CMC	Administration and operation of the TOE is governed by a policy that states the conditions under which the CMC software can be used for administering the TOE and that no other software can be used for administering it. The CMC software **must** reside in the host PC that is operated by trusted administrators in a secure premises that prevents physical access by parties other than the administrators. The host PC of the CMC software **should** not be connected to public networks, but in case it is then the administrators must ensure that illegitimate access to the host PC and to the software residing in it through the network are prevented.
OSP.INIT	Initialization of the TOE is governed by a policy that states the criterion for trustworthiness to the host PC's on which the TOE may be initialized. The TOE initialization must take place in a trusted host PC not connected to any network. The trustworthiness of the host PC must be asserted by it residing in physically secure premises and only administered by trustworthy administrators. An administrator must be present and supervise each TOE initialization taking place using the host PC.
OSP.CERT	The issuance of firmware upgrades is governed by a policy stating the requirements for the trustworthiness of the signature keys, signature computation for the upgrades, and issuance of public key certificates for the

	upgrade issuing authority. The public key certificates should be valid X.509 (or equivalent) certificates and represent signing keys of RSA with at least 512bit keys, or other algorithms with at least similar theoretical strength than RSA with 512bit keys. The organization using the TOE must identify trustworthy certification authorities and ensure that each public key certificate is produced by a trusted certification authority. **Application note**: Specific advice on dealing with public key certificates and certification authorities is given in Sect. 5.2 of [4].

4.3 Assumptions

44 The following assumptions govern the operational environment of the TOE: **None.**

5 Security Objectives

45 This section states the exact security objectives for the TOE so that the security problem definition is adequately and completely addressed. The security objectives are stated for the TOE and for the operational environment of the TOE.

5.1 Security objectives for the TOE

46 Security objectives for the TOE are enumerated in Table 9.

Table 9 – Security objectives for the TOE

Identifier	Objective statement
O.USER_AUTH	The authenticity of each user is determined and any attempt of illegitimate users to masquerade as legitimate users is detected with a high likelihood and that the attempts to masquerade as legitimate users trigger the necessary actions required for protecting the TOE.
O.DATA_ACCESS	Access to the controlled data and functions is only granted to legitimate users. Controlled data is the user data persistently stored on the TOE to which access is only granted to legitimate end users. Controlled functions are the administrative functions of the TOE to which access is only granted to legitimate administrators.
O.TOE_INTEGRITY	The security state of the TOE, consisting of the secrets stored persistently on the TOE, the authentication state, and the lockdown state is protected against unauthorized modification and can only be altered by authorized and authenticated parties.
O.CMC	The Authentication state and Lockdown state of the TOE can only be administered through the interface provided by the CMC software.
O.CRYPTO	The cryptographic keys, the underlying cryptographic algorithms and other cryptographic primitives are sufficiently secure and cryptographic operations sufficiently protect the user data and security parameters stored on the TOE.
O.UPGRADE	Only authentic and legitimate firmware upgrades are accepted for installation by the TOE.

5.2 Security objectives for the environment

5.2.1 Security objectives for the IT environment

47 Security objectives for the IT environment of the TOE are stated in Table 10.

Table 10 – Security objectives for the IT environment

Identifier	Objective statement
OE.PASSWORDS	The passwords accepted by the host PC for use by the TOE are of high quality.
OE.CMC	The CMC used for administering the TOE is uncompromised.
OE.CERT	Public key certificates used for authenticating firmware upgrades are trustworthy and the corresponding private keys used for digitally signing the upgrades are uncompromised.

5.2.2 Security objectives for the non-IT environment

48 The following security objectives are stated for the non-IT environment of the TOE:
None.

6 Extended components definition

49 There are no extended components applicable to the TOE, hence none of the requirements for the Extended Components Definition (ASE_ECD) are applicable to this ST.

7 IT Security Requirements

7.1 Overview

50 This section defines the security requirements satisfied by the TOE. Each requirement has been extracted from version 3.1 of the Common Criteria, part 2 providing functional requirements and part 3 providing assurance requirements.

7.2 TOE Security Functional Requirements

51 This section contains the security functional components from part 2 of the Common Criteria with the operations completed.

52 Standard Common Criteria text is in regular black font and the text inserted to perform an operation on the requirement is in accordance with the conventions specified in section 1 of this ST.

Table 11 – Summary of TOE Security Functional Requirements

Identifier	Title
Cryptographic support	
FCS_CKM.1	Cryptographic key generation

Identifier	Title
FCS_CKM.4	Cryptographic key destruction
FCS_COP.1a	Cryptographic operation (Password transformation)
FCS_COP.1b	Cryptographic operation (Stored data protection)
FCS_COP.1c	Cryptographic operation (Firmware verification)
FCS_COP.1d	Cryptographic operation (key and password protection)
User data protection	
FDP_ACC.1a	Subset access control (Lockdown SFP)
FDP_ACF.1a	Security attribute based access control (Lockdown SFP)
FDP_ACC.1b	Subset access control (Data Access SFP)
FDP_ACF.1b	Security attribute based access control (Data Access SFP) **Application note**: It would appear beneficial to establish an SFP for device initialisation but given that there's no method available to the TOE for establishing the authenticity of the initialisation software prior to the TOE is initialised, device initialisation triggering the modification of the Lifecycle state can only take place in a trusted host PC.
FDP_ACC.1c	Subset access control (firmware upgrade SFP)
FDP_ACF.1c	Security attribute based access control (firmware upgrade SFP)
FDP_ITC.2	Import of user data with security attributes
FDP_SDI.2	Stored data integrity monitoring and action
FDP_RIP.1a	Subset residual information protection (Authentication Data)
FDP_RIP.1b	Subset residual information protection (Cryptographic Keys)
Identification and authentication	
FIA_AFL.1	Authentication failure handling
FIA_ATD.1	User attribute definition
FIA_UAU.1	Timing of authentication
FIA_UID.1	Timing of identification
Security management	
FMT_MSA.1a	Management of security attributes (Lockdown SFP)
FMT_MSA.1b	Management of security attributes(Data Access SFP)
FMT_MSA.1c	Management of security attributes (Firmware Upgrade SFP)
FMT_MSA.2	Secure security attributes
FMT_MSA.3a	Static attribute initialization (Lockdown SFP)

Identifier	Title
FMT_MSA.3b	Static attribute initialization (Data Access SFP)
FMT_MSA.3c	Static attribute initialization (Firmware upgrade SFP)
FMT_SMF.1	Specification of Management Functions
FMT_SMR.1	Security roles
Protection of the TSF	
FPT_FLS.1	Failure with preservation of secure state
FPT_PHP.1	Passive detection of physical attack
FPT_TDC.1	Inter-TSF basic TSF data consistency
FPT_TST.1a	TSF testing (key generation)
FPT_TST.1b	TSF testing (start-up)
Trusted Paths/Channels	
FTP_ITC.1a	Inter-TSF trusted channel (CMC)
FTP_ITC.1b	Inter-TSF trusted channel (PKI certificate)
FTP_ITC.1c	Inter-TSF trusted channel (Trusted application)

7.2.1 Cryptographic support

7.2.1.1 FCS_CKM.1 Cryptographic key generation

Hierarchical to:	No other components.
FCS_CKM.1.1	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**AES**] and specified cryptographic key sizes [**256 bits**] that meet the following: [**Federal Information Processing Standard (FIPS) Publication 197, "Advanced Encryption Standard (AES)", 26 November 2001**].
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction
Notes:	None.

7.2.1.2 FCS_CKM.4 Cryptographic key destruction

Hierarchical to:	No other components.
FCS_CKM.4.1	The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [**zeroization**] that meets the following: [**FIPS 140-2 Level 2**].
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]
Notes:	The crypto officer can zeroize the key. Alternatively, the key is destroyed upon device re-formatting prior to re-initialization.

7.2.1.3 FCS_COP.1a Cryptographic operation (password transformation)

Hierarchical to:	No other components.
FCS_COP.1a.1	The TSF shall perform [**password transformation**] in accordance with a specified cryptographic algorithm [**SHA-1**] and cryptographic key sizes [**N/A**] that meet the following: [**Federal Information Processing Standard (FIPS) Publication 180-1, "Secure Hash Algorithm", 17 April 1995**].
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
Notes:	None.

7.2.1.4 FCS_COP.1b Cryptographic operation (stored data protection)

Hierarchical to:	No other components.
FCS_COP.1b.1	The TSF shall perform [1. **decryption of user data persistently stored on the FLASH memory , and** 2. **encryption of user data for persistent storage on the FLASH memory**] in accordance with a specified cryptographic algorithm [**AES**] and cryptographic key sizes [**256 bits**] that meet the following: [**Federal Information Processing Standard (FIPS) Publication 197, "Advanced Encryption Standard (AES)", 26 November 2001**].
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
Notes:	None.

7.2.1.5 FCS_COP.1c Cryptographic operation (firmware verification)

Hierarchical to:	No other components.
FCS_COP.1c.1	The TSF shall perform [**digital signature verification**] in accordance with a specified cryptographic algorithm [**RSA**] and cryptographic key sizes [**1024 bits**] that meet the following: [**PKCS#1: RSA Encryption Standard v1.5, November 1993, RSA Laboratories**].
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
Notes:	The signature verification is used for authenticating the firmware upgrades.

7.2.1.6 FCS_COP.1d Cryptographic operation (key and password protection)

Hierarchical to:	No other components.
FCS_COP.1d.1	The TSF shall perform [1. **Encryption and decryption of the reference authentication data, and** 2. **Encryption and decryption of the user data encryption key**] in accordance with a specified cryptographic algorithm [**AES**] and cryptographic key sizes [**256 bits**] that meet the following: [**Federal Information Processing Standard (FIPS) Publication 197, "Advanced Encryption Standard (AES)", 26 November 2001**].

Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
Notes:	None.

7.2.2 User data protection

7.2.2.1 FDP_ACC.1a Subset access control (Lockdown SFP)

Hierarchical to:	No other components.
FDP_ACC.1a.1	The TSF shall enforce the [**Lockdown SFP**] on [**a) Subjects:** **i. Any user.** **b) Objects:** **i. Lockdown State, and** **ii. TOE Object.** **c) Operations:** **i. Any access.**].
Dependencies:	FDP_ACF.1 Security attribute based access control
Notes:	None.

7.2.2.2 FDP_ACF.1a Security attribute based access control (Lockdown SFP)

Hierarchical to:	No other components.
FDP_ACF.1a.1	The TSF shall enforce the [**Lockdown SFP**] to objects based on the following: [**a) Any user:** **i) None.** **b) TOE object:** **i) Lockdown state**.].
FDP_ACF.1a.2	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [**IF (Lockdown state Status is of value SET)** **THEN the following operations are allowed:** **a. Recovery from Lockdown mode.**].

FDP_ACF.1a.3	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [**None**].
FDP_ACF.1a.4	The TSF shall explicitly deny access of subjects to objects based on the following additional rules [**None**].
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialization
Notes:	None.

7.2.2.3 FDP_ACC.1b Subset access control (Data Access SFP)

Hierarchical to:	No other components.
FDP_ACC.1b.1	The TSF shall enforce the [**Data Access SFP**] on [**a) Subjects:** **i. Access Software.** **b) Objects:** **i. User Data.** **c) Operations:** **i. Decrypt and release, and** **ii. Encrypt and store.**].
Dependencies:	FDP_ACF.1 Security attribute based access control
Notes:	None.

7.2.2.4 FDP_ACF.1b Security attribute based access control (Data Access SFP)

Hierarchical to:	No other components.
FDP_ACF.1b.1	The TSF shall enforce the [**Data Access SFP**] to objects based on the following: [**a) TOE:** **i) Authentication State.** **b) User Data:** **i) Request Status.**].
FDP_ACF.1b.2	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [**a) IF** **(The value of TOE Authentication State is SET)** **THEN** **The TOE may** **a. Decrypt and release that User Data whose Request**

	Status is of value Requested, and **b. Encrypt and store that User Data whose Request Status is of value Requested.** **].**
FDP_ACF.1b.3	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [**None**].
FDP_ACF.1b.4	The TSF shall explicitly deny access of subjects to objects based on the following additional rules [**None**].
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialization
Notes:	None.

7.2.2.5 FDP_ACC.1c Subset access control (Firmware upgrade SFP)

Hierarchical to:	No other components.
FDP_ACC.1c.1	The TSF shall enforce the [**Firmware upgrade SFP**] on [**a) Subjects:** **i. Firmware upgrade officer.** **b) Objects:** **i. Firmware upgrade.** **c) Operations:** **i. Accept firmware for installation.**].
Dependencies:	FDP_ACF.1 Security attribute based access control
Notes:	None.

7.2.2.6 FDP_ACF.1c Security attribute based access control (Firmware upgrade SFP)

Hierarchical to:	No other components.
FDP_ACF.1c.1	The TSF shall enforce the [**Firmware upgrade SFP**] to objects based on the following: [**a) Firmware upgrade officer:** **i) Authenticity.** **b) Firmware update:** **i) Integrity.**].
FDP_ACF.1c.2	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [**IF ((Firmware upgrade officer Authenticity is of value**

	AUTHENTIC) **and** **(Firmware update Integrity is of value SUCCESFULLY VERIFIED)** **)** **THEN the following operations are allowed:** **a. The update agent is allowed to accept the Firmware update for installation.**].
FDP_ACF.1c.3	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [**None**].
FDP_ACF.1c.4	The TSF shall explicitly deny access of subjects to objects based on the following additional rules [**None**].
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialization
Notes:	The firmware upgrade is presented together with a digital signature. That digital signature is used for establishing both authenticity of the firmware upgrade officer and the integrity of the firmware upgrade. Firmware upgrade officer must only accept firmware upgrades when the upgrade is sufficiently evaluated to ensure that the upgrade does not invalidate the certificate of the TOE. Firmware upgrade officer is authentic if he presents a valid signature. The firmware upgrade is of good integrity if the verification of the attached signature determines that no changes have been made to the upgrade since the issuance of the signature. Both conditions will have to occur for successful acceptance of the firmware upgrade. If any of the conditions does not occur, the upgrade is rejected by the TOE.

7.2.2.7 FDP_ITC.2 Import of user data with security attributes

Hierarchical to:	No other components.
FDP_ITC.2.1	The TSF shall enforce the [**Firmware upgrade SFP**] when importing user data, controlled under the SFP, from outside of the TOE.
FDP_ITC.2.2	The TSF shall use the security attributes associated with the imported user data.
FDP_ITC.2.3	The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.
FDP_ITC.2.4	The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.
FDP_ITC.2.5	The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: [**None**]

Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] FPT_TDC.1 Inter-TSF basic TSF data consistency
Notes:	None.

7.2.2.8 FDP_RIP.1a Subset residual information protection (Authentication data)

Hierarchical to:	No other components.
FDP_RIP.1a.1	The TSF shall ensure that any previous information content of a resource is made unavailable upon the [***deallocation of the resource from***] the following objects: [**VAD representation, RAD representation**].
Dependencies:	None.
Notes:	The deallocation happens immediately after the authentication is completed.

7.2.2.9 FDP_RIP.1b Subset residual information protection (Cryptographic keys)

Hierarchical to:	No other components.
FDP_RIP.1b.1	The TSF shall ensure that any previous information content of a resource is made unavailable upon the [***allocation of the resource to***] the following objects: [**AES Key**].
Dependencies:	None.
Notes:	None.

7.2.2.10 FDP_SDI.2 Stored data integrity monitoring and action

Hierarchical to:	FDP_SDI.1 Stored data integrity monitoring
FDP_SDI.2.1	The TSF shall monitor user data stored in containers controlled by the TSF for [**inconsistencies between the object values and the object values from which checksums were calculated**] on all objects, based on the following attributes: [**Lockdown state** and **Authentication state**]
FDP_SDI.2.2	Upon detection of a data integrity error, the TSF shall [**indicate the error to the access software**].
Dependencies:	None.
Notes:	None.

7.2.3 Identification and authentication

7.2.3.1 FIA_AFL.1 Authentication failure handling

Hierarchical to:	No other components.
FIA_AFL.1.1	The TSF shall detect when [***an administrator configurable positive integer within [*****no less than 1 and no more than 100*****]***] unsuccessful authentication attempts occur related to [**User authentication**].
FIA_AFL.1.2	When the defined number of unsuccessful authentication attempts has been [***met***], the TSF shall [**Enter a Lockdown mode**].
Dependencies:	FIA_UAU.1 Timing of authentication
Notes:	The value of MAXNOA (the number indicating the maximum number of subsequent failed authentication attempts) is set on the manufacturing stage and has to be at least 25 and can be at most 100.

7.2.3.2 FIA_ATD.1 User attribute definition

Hierarchical to:	No other components.
FIA_ATD.1.1	**a)** The TSF shall maintain the following list of security attributes belonging to individual **TOE** users: [**Role**].
Dependencies:	None.
Notes:	None.

7.2.3.3 FIA_UAU.1 Timing of authentication

Hierarchical to:	No other components.
FIA_UAU.1.1	The TSF shall allow [**a) Request of self-tests,** **b) Device reset,** **c) CD emulation,** **d) Returning of error status,** **e) Returning version information,** **f) showing status,** **g) Device initiation, and** **h) Entering a Lockdown mode.**] on behalf of the user to be performed before the user is authenticated.
FIA_UAU.1.2	The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.
Dependencies:	FIA_UID.1 Timing of identification
Notes:	None.

7.2.3.4 FIA_UID.1 Timing of identification

Hierarchical to:	No other components.
FIA_UID.1.1	The TSF shall allow [**a) Establishment of a trusted channel between itself and an entity claiming to be the CMC software, and** **b) Receipt of a firmware upgrade candidate from an entity claiming to be firmware upgrade officer.**] on behalf of the user to be performed before the user is identified.
FIA_UID.1.2	The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.
Dependencies:	None.
Notes:	None.

7.2.4 Security Management

7.2.4.1 FMT_MSA.1a Management of security attributes (Lockdown SFP)

Hierarchical to:	No other components.
FMT_MSA.1a.1	The TSF shall enforce the [**Lockdown SFP**] to restrict the ability to [***modify***] the security attributes [**a) Lockdown Status.**] to [**the TOE**].
Dependencies:	[FDP_ACC.1 Subset access control, FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
Notes:	None.

7.2.4.2 FMT_MSA.1b Management of security attributes (Data Access SFP)

Hierarchical to:	No other components.
FMT_MSA.1b.1	The TSF shall enforce the [**Data Access SFP**] to restrict the ability to [***modify***] the security attributes [**a) Authentication Status.**] to [**the TOE**].
Dependencies:	[FDP_ACC.1 Subset access control, FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions

Notes:	None.

7.2.4.3 FMT_MSA.1c Management of security attributes (Firmware upgrade SFP)

Hierarchical to:	No other components.
FMT_MSA.1c.1	The TSF shall enforce the [**Firmware upgrade SFP**] to restrict the ability to [***modify***] the security attributes [**a) Authenticity of firmware upgrade officer, and** **b) Integrity of the firmware upgrade.**] to [**the TOE**].
Dependencies:	[FDP_ACC.1 Subset access control, FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
Notes:	None.

7.2.4.4 FMT_MSA.2 Secure security attributes

Hierarchical to:	No other components.
FMT_MSA.2.1	The TSF shall ensure that only secure values are accepted for [**a) Lockdown status,** **b) Authentication status,** **c) Authenticity of firmware upgrade officer, and** **d) Integrity of firmware upgrade.**].
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles
Notes:	None.

7.2.4.5 FMT_SMR.1 Security roles

Hierarchical to:	No other components.
FMT_SMR.1.1	The TSF shall maintain the roles [**a) Administrator,** **b) End user,** **c) Firmware update officer, and** **d) Crypto officer.**

	].
FMT_SMR.1.2	The TSF shall be able to associate users with roles.
Dependencies:	FIA_UID.1 Timing of identification
Notes:	The Administrator uses CMC software to access the TOE. The End user uses access software to access the TOE.

7.2.4.6 FMT_SMF.1 Specification of Management Functions

Hierarchical to:	No other components.
FMT_SMF.1.1	The TSF shall be capable of performing the following security management functions: [**a) Manage Authentication state,** **b) Manage Lockdown state, and** **c) Import public key certificate for Firmware upgrade verification.**].
Dependencies:	None.
Notes:	None.

7.2.4.7 FMT_MSA.3a Static attribute initialisation (Lockdown SFP)

Hierarchical to:	No other components.
FMT_MSA.3a.1	The TSF shall enforce the [**Lockdown SFP**] to provide [***permissive***] default values for security attributes that are used to enforce the SFP.
FMT_MSA.3a.2	The TSF shall allow the [**None**] to specify alternative initial values to override the default values when an object or information is created.
Dependencies:	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles
Notes:	The permissive initial value is when the Lockdown State is not set.

7.2.4.8 FMT_MSA.3b Static attribute initialisation (Data Access SFP)

Hierarchical to:	No other components.
FMT_MSA.3b.1	The TSF shall enforce the [**Data Access SFP**] to provide [***restrictive***] default values for security attributes that are used to enforce the SFP.
FMT_MSA.3b.2	The TSF shall allow the [**None**] to specify alternative initial values to override the default values when an object or information is created.
Dependencies:	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles

Notes:	The restrictive value is that the Authentication State is not set.

7.2.4.9 FMT_MSA.3c Static attribute initialisation (Firmware upgrade SFP)

Hierarchical to:	No other components.
FMT_MSA.3c.1	The TSF shall enforce the [**Firmware upgrade SFP**] to provide [***restrictive***] default values for security attributes that are used to enforce the SFP.
FMT_MSA.3c.2	The TSF shall allow the [**None**] to specify alternative initial values to override the default values when an object or information is created.
Dependencies:	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles
Notes:	The restrictive value is that the Authenticity of the firmware upgrade officer is NOT AUTHENTIC. The restrictive value is that the Integrity of the firmware upgrade is NOT SUCCESSFULLY VERIFIED.

7.2.5 Protection of the TSF

7.2.5.1 FPT_FLS.1 Failure with Preservation of Secure State

Hierarchical to:	No other components.
FPT_FLS.1.1	The TSF shall preserve a secure state when the following types of failures occur: [**power failure**].
Dependencies:	None.
Notes:	A start-up procedure clears the authentication data temporarily stored on the TOE as well as the authentication state each time the TOE is powered up to ensure a secure start-up when power is lost during authentication.

7.2.5.2 FPT_PHP.1 Passive Detection of Physical Attacks

Hierarchical to:	No other components.
FPT_PHP.1.1	The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF.
FPT_PHP.1.2	The TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.
Dependencies:	None.
Notes:	Detection of physical tampering is through tamper-evident sealing of the TOE in epoxy so that the casing of the TOE may be opened and the sealed circuitry inspected for visual traces of any attempted tampering.

7.2.5.3 FPT_TDC.1 Inter-TSF basic TSF data consistency

Hierarchical to:	No other components.
FPT_TDC.1.1	The TSF shall provide the capability to consistently interpret [**public key certificates**] when shared between the TSF and another trusted IT product.
FPT_TDC.1.2	The TSF shall use [**a) Only valid X.509 certificates are accepted, and** **b) Either** **a. X.509 certificates are only accepted if approved by the human user of the TOE, or** b. **The X.509 certificates are only accepted in the initialization phase of the TOE.**] when interpreting the TSF data from another trusted IT product.
Dependencies:	None.
Notes:	The human user must approve the trustworthiness of the party signing the public key certificates prior to the acceptance by the TOE.

7.2.5.4 FPT_TST.1a TSF Testing (key generation)

Hierarchical to:	No other components.
FPT_TST.1a.1	The TSF shall run a suite of self tests [***at the conditions* [prior to the generation of cryptographic keys]**] to demonstrate the correct operation of [[***key generation function***]].
FPT_TST.1a.2	The TSF shall provide authorised users with the capability to verify the integrity of [[**the Random Number Generator**]].
FPT_TST.1a.3	The TSF shall provide authorised users with the capability to verify the integrity of stored TSF executable code.
Dependencies:	None.
Notes:	None.

7.2.5.5 FPT_TST.1b TSF Testing (start-up)

Hierarchical to:	No other components.
FPT_TST.1b.1	The TSF shall run a suite of self tests [***during initial start-up***] to demonstrate the correct operation of [[***cryptographic algorithms***]].
FPT_TST.1b.2	The TSF shall provide authorised users with the capability to verify the integrity of [[**cryptographic algorithms**]].
FPT_TST.1b.3	The TSF shall provide authorised users with the capability to verify the integrity of stored TSF executable code.
Dependencies:	None.

Notes:	The tests are a) AES known answer test, b) SHA-1 known answer test, c) RSA pairwise consistency test, d) DRNG known answer test, and e) RSA 1024 signature verification test.

7.2.6 Trusted Paths/Channels

7.2.6.1 FTP_ITC.1a Inter-TSF Trusted Channel (CMC)

Hierarchical to:	No other components.
FTP_ITC.1a.1	The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.
FTP_ITC.1a.2	The TSF shall permit [***another trusted IT product***] to initiate communication via the trusted channel.
FTP_ITC.1a.3	The TSF shall initiate communication via the trusted channel for [**a) Executing management function Manage Authentication state, and** **b) Executing management function Manage Lockdown state.**].
Dependencies:	None.
Notes:	The another trusted IT product is the CMC Software.

7.2.6.2 FTP_ITC.1b Inter-TSF Trusted Channel (Firmware upgrade)

Hierarchical to:	No other components.
FTP_ITC.1b.1	The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.
FTP_ITC.1b.2	The TSF shall permit [***another trusted IT product***] to initiate communication via the trusted channel.
FTP_ITC.1b.3	The TSF shall initiate communication via the trusted channel for [**a) Executing management function Import public key certificate for Firmware upgrade verification.**].
Dependencies:	None.
Notes:	None.

7.2.6.3 FTP_ITC.1c Inter-TSF Trusted Channel (Trusted application)

Hierarchical to:	No other components.

FTP_ITC.1c.1	The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.
FTP_ITC.1c.2	The TSF shall permit [***another trusted IT product***] to initiate communication via the trusted channel.
FTP_ITC.1c.3	The TSF shall initiate communication via the trusted channel for [**a) Protecting the user data persistently stored on the TOE when returned to the requesting application level software residing in the host PC.**].
Dependencies:	None.
Notes:	This requires that the application level software residing in the host PC is sufficiently modified using the SDK.

7.3 TOE Security Assurance Requirements

53 The assurance package for the evaluation of the TOE is Evaluation Assurance Level 2 (EAL2) augmented with ALC_FLR.1 (Basic flaw remediation).

54 EAL2 assurance requirements provide confidence in the security functionality of the TOE by analysis using a functional and interface specification, guidance documentation and the high-level design of the TOE, to understand the security behaviour.

55 The analysis is supported by independent testing of the TOE security functions, evidence of developer testing based on the functional specification, selective independent confirmation of the developer test results, strength of function analysis, and evidence of a developer search for obvious vulnerabilities.

56 EAL2 also provides assurance through a configuration list for the TOE, and evidence of secure delivery procedures.

57 The TOE provides a means to upgrade the firmware once issued to the end user. Firmware upgrades are used for correcting known security vulnerabilities or other flaws that may result in security vulnerabilities. These procedures are subjected to further assurance by the inclusion of ALC_FLR.1.

58 Table 12 lists the TOE security assurance requirements for this evaluation. Complete details of all assurance components are located in part 3 of the Common Criteria.

Table 12 – Summary of TOE security assurance requirements

Assurance class	Assurance components
Development (ADV)	ADV_ARC.1
	ADV_FSP.2
	ADV_TDS.1
Guidance Documents (AGD)	AGD_OPE.1
	AGD_PRE.1

Assurance class	Assurance components
Life-Cycle Support (ALC)	ALC_CMC.2
	ALC_CMS.2
	ALC_DEL.1
	ALC_FLR.1
Security Target Evaluation (ASE)	ASE_CCL.1
	ASE_ECD.1
	ASE_INT.1
	ASE_OBJ.2
	ASE_REQ.2
	ASE_SPD.1
	ASE_TSS.1
Tests (ATE)	ATE_COV.1
	ATE_FUN.1
	AET_IND.2
Vulnerability Assessments (AVA)	AVA_VAN.2

8 TOE Summary Specification

8.1 Overview

59 This chapter provides the TOE summary specification, a high-level definition of the security functions of the TOE and a summary of how those Security Functions meet the SFR's.

8.2 Security Functions

60 The TOE security functions include the following:

- **User data protection.** The TOE provides the capability to protect user data in storage by encrypting when stored and decrypting when retrieved from storage and passed to the host PC, and by granting access to the data only upon successful authentication of the end user.
- **Protection of the TOE.** The TOE provides the capability to detect a potential password guessing attack by maintaining a counter for recording the number of consecutive authentication failures, and if the number exceeds a defined threshold the TOE enters a Lockdown state in which all user data access requests are denied until the device is restored into an operational mode by formatting and re-initialization of it. The TOE implements a means to protect its security state from unauthorized modification or disclosure.

- **Management.** The TOE provides the capability to differentiate between a legitimate CMC Software and untrusted software attempting to masquerade as legitimate CMC Software and only allow access to the management of the TOE to legitimate CMC Software.
- **Security Management.** The TOE provides the capability to restrict the management of critical security attributes to authorized parties only.
- **Firmware upgrading.** The TOE provides a capability to verify the authenticity of firmware upgrades and reject unauthentic upgrades.

8.2.1 User Data Protection

61 User data protection concerns with identification and authentication of users, controlling access to user data to ensure that user data is disclosed to the software running in the host PC only in the presence of a human user possessing the knowledge of a correct authentication data, and implementation of strong cryptography to guarantee resistance to software attacks.

62 The cryptographic protection is achieved by using AES for encrypting user data when persistently stored in the non-volatile memory of the device using 256-bit keys. The cryptographic keys are generated by the TOE, covering FCS_CKM.1, and never leave the TOE as all encryption and decryption operations are done on hardware constituting a part of the TOE, which covers FCS_COP.1b for AES. The TOE implements necessary checks for the Random Number Generator to ensure uniqueness of the resulting key prior to proceeding with the generation of the key, which covers FPT_TST.1a.

63 The TOE also performs a sequence of cryptographic checks during the start-up to ensure that the cryptographic engine operates correctly. At any time, an authorized user can restart the TOE and if the start-up sequence completes, be assured that the cryptographic engine is intact and functions correctly. This covers FPT_TST.1b.

64 The cryptographic keys are securely destroyed when so requested by the crypto officer or when the memory of the TOE is reformatted and re-initialized. This covers FCS_CKM.4.

65 Additionally, SHA-1 hashing is used for protecting the authentication data when stored in the volatile or non-volatile memory of the device. This covers FCS_COP.1a. The SHA-1 hashed reference authentication data is stored encrypted using the root key. Once the authentication of end user takes place, the reference authentication data is decrypted for allowing the comparison of the reference authentication data and verification authentication data. This covers partially FCS_COP.1d.

66 The other part of FCS_COP.1d is covered by the way the AES keys for encrypting and decrypting user data are protected. The TOE encrypts the keys using the root key and stores them only encrypted. Once the keys are used, they re fetched from the memory and decrypted so that they can be used for encrypting or decrypting the user data.

67 **Application note**: Usually the presence of FCS_COP.1 implies the need for key generation or key importing but in case of cryptographically secure hash functions, SHA-1 in this application, there are no cryptographic keys so there are also no corresponding SFRs for generating or importing the keys.

68 Prior to decrypting the requested file and passing it to the host PC or encrypting a file for storage on the FLASH memory, the TOE investigates whether the authentication state is set to indicate a successful authentication, representing a situation where the end user with access to the host PC of the TOE indeed knows the Password protecting the access to the encrypted user data. The access control covers FDP_ACC.1b and FDP_ACF.1b, and the verification of the password and alteration of the authentication state upon correct verification of the password presented to the TOE by the host PC covers FIA_ATD.1 and FIA_UAU.1.

69 Prior to the identification of the user, the user may need to establish a trusted channel between itself and the TOE. This is in particular when the user is the firmware upgrade

officer or CMC Software attempting to administer the TOE. Prior to the establishment of the trusted channel, the identification can not take place as any party may attempt to establish the channel, and the channel must be on place prior to the authentication of the CMC. This covers FIA_UID.1.

70 In case of a TOE being under a potential attack, it may enter the Lockdown mode. The TOE maintains the NOA-counter of the number of consecutive failed authentication attempts and if the value of NOA exceeds MAXNOA, the threshold defined in the manufacturing of the TOE, the Lockdown mode is entered. This covers FIA_AFL.1.

71 In the Lockdown mode, all access to the user data is denied. This covers FDP_ACC.1a and FDP_ACF.1a.

8.2.2 Protection of the TOE

72 The TOE implements a number of measures to ensure that the security state is maintained and only authorized alterations to the security state may take place and that any security critical data element processed by the TOE is sufficiently destroyed once no longer needed for immediate processing.

73 This protection includes ensuring that the security critical data and security attributes are stored persistently in a manner that prevents any undetected violations of integrity (FDP_SDI.2), and that any security critical data needed for processing by the TOE is immediately cleared when no longer needed, or upon powering up the TOE. Of particular interest is ensuring that the Authentication Data presented to the TOE for verification is cleared from memory immediately upon completion of the verification, covering FDP_RIP.1a, and that in case of a re-generation of the 256-bit AES key, the new key is totally random and independent of the previous key, covering FDP_RIP.1b.

74 As the TOE has no internal source of power, each session must be considered terminated upon loss of power, indicating that the token housing the TOE is removed from the host PC or that the power to the host PC is lost for other reasons. Once the TOE is powered again by the host PC, there are no guarantees that the end user of the host PC is still the authentic party using the host PC previously, and the TOE must clear the security state upon each powering up. This covers FPT_FLS.1. The circuitry of the TOE is sealed in tamper-evident epoxy, and a visual inspection of the sealing is likely to aid in the detection of any attempts to tamper with the TOE, addressing FPT_PHP.1.

8.2.3 Management and access by external devices

75 The TOE is capable of establishing a trusted channel between itself and a legitimate CMC Software, and of differentiating the legitimate CMC Software from any other software. This covers FTP_ITC.1a.

76 The TOE is also capable of establishing a secure session between itself and a specifically engineered application. The secure session may be initiated by the application. Specifically, the application level software being executed in the host PC may request a trusted channel between itself and the TOE using the commands available in the SDK toolkit associated to the TOE. The TOE can respond by accepting the request and establishing a trusted channel as explicated in FTP_ITC.1c.

8.2.4 Security management

77 The TOE supports a number of features for managing the security state. These measures include controlling the values of the security attributes to only allow approved modifications. This covers FMT_MSA.1a, FMT_MSA.1b and FMT_MSA.1c. Additionally, they ensure that appropriate controls are in place to restrict access to the functions available for modifying the values of security attributes to ensure that only secure values are accepted. This covers FMT_MSA.2.

78 Additionally, the TOE ensures that the initial values of security attributes are such that require authentication prior to accessing user data stored on the device (FMT_MSA.3b), which allow the TOE to be in an operational state until the conditions for a Lockdown

mode are met (FMT_MSA.3a), and which ensure that each party issuing a firmware upgrade to the TOE is considered unauthentic until successfully authenticated (FMT_MSA.3c).

79 The TOE has a well defined set of security management functions (FMT_SMF.1) and maintains the necessary roles for managing the security function behaviour and accessing the TOE functions (FMT_SMR.1).

8.2.5 Firmware upgrading

80 The TOE supports firmware upgrades issued by the developer once the TOE is issued to the user.

81 The firmware upgrades are digitally signed and the firmware upgrade officer is to present the RSA signature to the TOE together with the upgrade. The TOE then verifies the signature as expressed in FCS_COP.1c. The cryptographic key used in the signature verification, i.e. the public key of the issuer of the upgrade, is distributed to the TOE in form of a public key certificate. This covers FDP_ITC.2, FPT_TDC.1 and FTP_ITC.1b by requiring that the public certificates are properly distributed. (FDP_ITC.2), that proper investigation is conducted to ensure that only valid certificates are accepted (FPT_TDC.1) and that the sources of certificates and upgrades are properly determined so that only legitimate certificates and upgrades are accepted (FTP_ITC.1b).

82 Depending on the verification result, the TOE can either reject the upgrade or accept it for installation. This is coded in FDP_ACC.1c and FDP_ACF.1c.

9 Rationale

9.1 Conformance claim rationale

83 The Conformance Claim of this ST does not claim conformance to any Protection Profile. Hence, there are no elements to be covered in the conformance claim rationale.

9.2 Security objectives rationale

84 Security objectives rationale is provided for the TOE and for the environment of the TOE.

9.2.1 Security objectives for the TOE

85 Table 13 provides a mapping of the TOE Security objectives and threats and a justification for the mapping.

Table 13 – Mapping of TOE security objectives to threats

Threats	Objective	Justification
T.AUTH_FAILURE	O.USER_AUTH O.DATA_ACCESS	A successful violation of the authentication procedures by potentially malicious software running in the host PC would result in the loss of authenticity of the users of the TOE and in the facilitation of illegitimate access to the data. The former scenario would be a violation of O.USER_AUTH and the latter a violation of O.DATA_ACCESS. Consequently, security objectives O.USER_AUTH and O.DATA_ACCESS can only be preserved if threat T.AUTH_FAILURE does not occur.

Threats	Objective	Justification
		T.AUTH_FAILURE can be caused by two means: an illegitimate subject may succeed in guessing the password of a legitimate user, or discover an alternative means for accessing the TOE. In both cases, the authenticity of the users is violated as the scenarios result in the granting of access to illegitimate parties. The first concern is covered of the O.USER_AUTH is upheld so that any attempt of an illegitimate party to guess the password of a legitimate end user is detected with a high likelihood. The second concern is addressed if O.DATA_ACCESS is upheld so that no access paths exist that would allow illegitimate users access to the TOE. Jointly, preservation of these objectives fully prevents T.AUTH_FAILURE from occurring.
T.AC_FAILURE	O.DATA_ACCESS	A successful violation of the access control principles of the TOE would result in unauthorized parties succeeding in gaining access to the data protected by the TOE or in unauthorized modification of the TOE security state or the values of security parameters. All these accesses are considered illegitimate. Consequently, to only allow legitimate access to the resources protected by the TOE as coded in security objective O.DATA_ACCESS, threat T.AC_FAILURE must not occur. T.AC_FAILURE concerns with any means illegitimate access software may gain access to the user data or administrative functions of the TOE. Preventing T.AC_FAILURE from occurring, hence ensures that only legitimate access to the TOE are granted. Consequently, when the O.DATA_ACCESS is upheld, T.AC_FAILURE is fully prevented.
T.CRYPTO	O.CRYPTO	Cryptographic protection of the user data persistently stored on the TOE is in the core of the security of the TOE. The cryptographic primitives must be sufficiently secure and the TOE implementation must ensure that the underlying security of the cryptographic primitives is not reduced by the TOE design and implementation so that the software running on the host PC could successfully cryptoanalyse the TOE to deduce the user data without prior knowledge of the cryptographic key used for protecting that user data. Consequently, objective O.CRYPTO is preserved when T.CRYPTO does not occur. T.CRYPTO could occur if a malicious party succeeded in successfully cryptoanalyzing the

Threats	Objective	Justification
		TOE to exploit weaknesses in the cryptographic functions of the TOE. If O.CRYPTO is preserved all cryptographic primitives and measures are sufficiently secure to prevent cryptoanalysis from succeeding. Therefore, T.CRYPTO is fully addressed of O.CRYPTO is upheld. **Application note**: At EAL2 it is sufficient to not assume a high attack potential and the ability of a threat agent to physically possess the TOE and subject it to advanced cryptoanalysis. Instead, the threat scenarios of T.CRYPTO only cover software attacks by malicious software residing on the host PC while the TOE is being inserted into the USB slot.
T.SECSTATE_ALT	O.CMC O.TOE_INTEGRITY	Importantly, there are no practical methods to prevent malicious software running on the host PC or a malicious human user possessing the TOE from triggering the Lockdown state by consecutively feeding the authentication dialogue with random password candidates. Therefore, it is always possible for a malicious party to trigger the Lockdown mode. If the recovery from the Lockdown is possible by any software distributed with the TOE, there is no practical way to prevent the malicious human user of the TOE to execute that software on the host PC to generate a new password for the TOE and to subsequently gain access to the user data persistently stored on the TOE. Therefore, it is essential that only a legitimate CMC Software can be used for managing the Lockdown state or Authentication state of the TOE. Consequently, T.SECSTATE_ALT can only be prevented if O.CMC is enforced. Additionally, malicious software running on the host PC may attempt to directly violate the security state of the TOE by altering the values of the critical security parameters of the TOE. In order to preserve O.TOE_INTEGRITY, such attempts must be resisted by the TOE. T.SECSTATE_ALT may occur by two means: by unauthorized modification of the security state of the TOE (Authentication state or Lockdown state) or by altering the values of the security parameters of the TOE protecting either access to the TOE (reference authentication data) or protecting the user data persistently stored on the TOE (cryptographic keys). The first concern is countered if O.CMC is upheld and only legitimate accesses to the security management functions controlling the Authentication state and Lockdown state are granted, and the second concern is countered if

Threats	Objective	Justification
		O.TOE_INTEGRITY is upheld and the TOE remains protected against integrity violations. Consequently, T.SECSTATE_ALT is fully prevented from occurring if O.CMC and O.TOE_INTEGRITY are upheld.
T.MAN_IN_THE_MIDDLE	O.USER_AUTH	Authenticity of the users as formulated in O.USER_AUTH requires that no illegitimate party may obtain access to the user data protected by the TOE. While O.DATA_ACCESS covers threats where the implementation or configuration of the TOE might be flawed to allow T.AC_FAILURE to take place, T.MAN_IN_THE_MIDDLE concerns with the inability of the TOE to control the operating system of the host PC. As the TOE is intended to be used in any host PC running a commercial, general purpose operating system not in the full control of the human user who is the owner of the TOE, the TOE can never fully assert that the host PC is not compromised. There could be general or specifically engineered malicious software residing in the host PC attempting to either intercept the communication between the TOE and the host PC (or legitimate applications being executed on it) or modify the communication to trigger an insecure state in the TOE to allow further exploitation. In order to counter the Man in the Middle attacks formulated in T.MAN_IN_THE_MIDDLE, the TOE is capable of establishing a trusted channel between itself and any specifically modified application level software. This trusted channel ensures the authenticity of users as stated in O.USER_AUTH by preventing any illegitimate access to the user data by illegitimate software attempting to intercept the communication between the TOE and the specific application. When O.USER_AUTH is upheld, all attempts for illegitimate parties to hijack secure communication channels between the TOE and the host PC fail with an overwhelming probability. Consequently, man in the middle attacks may not practically occur. Hence, when O.USER_AUTH is upheld, T.MAN_IN_THE_MIDDLE is fully prevented from occurring.
T.FW_UPGRADE	O.UPGRADE	It is of essence that only legitimate firmware upgrades are accepted by the TOE. A malicious software succeeding in triggering an installation of an illegitimate upgrade in the firmware of the TOE would gain an ability to modify the TOE configuration and data by will and result in major deviation from the security objectives of the TOE.

Threats	Objective	Justification
		Consequently, the TOE must ensure that only authentic firmware upgrades from legitimate parties are accepted. Enforcement of O.UPGRADE fully prevents illegitimate parties from succeeding in triggering installation of unauthentic firmware upgrades by ensuring that each upgrade is authenticated prior to acceptance, and unauthentic upgrades are rejected. When O.UPGRADE is upheld, only legitimate firmware upgrades are accepted by the TOE and illegitimate upgrades are rejected with an overwhelming probability. This fully prevents occurrence of T.FW_UPGRADE.

9.2.2 Security objectives for the environment

86 Table 14 provides a mapping of the Security objectives for the environment of the TOE to relevant threats and organizational security policies, as well as a justification for the mapping. There are no assumptions governing the usage and operation of the TOE, hence no assumptions are relevant to the mapping and justification.

Table 14 – Mapping of security objectives for the environment to threats and OSPs

Threat/OSP	Objective	Justification
T.PW_GUESS	OE.PASSWORDS	The inability of a malicious human user to guess the correct password and subsequently gain illegitimate access to the user data protected by the TOE is enforced by ensuring that each password accepted by the TOE is of sufficient quality, i.e. can only be guessed with a sufficiently low likelihood. The TOE itself does not implement controls for the quality of passwords but relies on the controls implemented by the password change dialogue software which is distributed in the USB token hosting the TOE but which is not part of the TOE. If only passwords of sufficient quality are accepted for use by the password change dialogue software, i.e. OE.PASSWORDS is preserved, threat T.PW_GUESS is prevented from occurring. Furthermore, the consequence of ensuring that T.PW_GUESS can not occur is that all passwords accepted by the TOE are of high quality. This ensures that OE.PASSWORDS is upheld.
OSP.INIT	OE.PASSWORDS	The initial password used for user authentication is created by the host PC in which the TOE initialization takes place. As there is no central trusted facility for the initializations, the organization distributing TOEs to its members must set up a policy to govern the trustworthiness of the host PC's in which TOE initialization may take place. The ability of the environment of the TOE to ensure

Threat/OSP	Objective	Justification
		that passwords selected in the initialization of the TOE and forwarded to the TOE are not substituted with weaker password variants which would later facilitate a successful attack by anybody in the physical possession of the TOE. Consequently, OE.PASSWORDS depends on the enforcement of OSP.INIT. When OE.PASSWORDS is upheld, the passwords forwarded to the TOE and subsequently accepted by the TOE are of high quality. In the initialization of the TOE, the initial password is created and the secure operation of the TOE depends on that initial password. Hence, ensuring that the initial password is of appropriate strength, i.e. ensuring that OE.PASSWORDS is upheld, enforces OSP.INIT.
OSP.CMC	OE.CMC	Organizational security needs vary, and different organizations may have different regulations to govern the recovery of TOEs from the Lockdown mode. In order to ensure trustworthiness of the CMC to establish a trusted channel between itself and TOE, i.e. to enforce OE.CMC, the organization distributing the TOEs to its members must define and enforce a policy as stated OSP.CMC to ensure recovery of the TOE's to a state sufficiently secure to their needs. Furthermore, ensuring that the instances of the CMC software remain uncompromised is essential for the ability of the TOE administrators to differentiate the legitimate CMC software from other software and illegitimate instances of CMC software. Therefore, ensuring that OE.CMC is upheld enforces OSP.CMC.
OSP.CERT	OE.CERT	The authenticity of the firmware upgrades is established by verifying the digital signatures distributed together with the upgrades. The public keys used for signature verification are distributed to the TOE in form of public key certificates. While the certificates are not generated by the TOE but by the issuing authority using sufficient roots of trust or certification authorities, the security of the TOE depends on the trustworthiness of those requirements. Consequently, in order to preserve OE.CERT, the TOE must be governed by a security policy stating the security of the issuance of the public key certificates used for distributing the public keys corresponding to the signing keys used for digitally signing the firmware upgrades. OE.CERT covers all aspects of OSP.CERT and is properly enforced when OSP.CERT is established. The ability to differentiate between legitimate and illegitimate firmware upgrades depends on the quality and authenticity of the public keys used for

Threat/OSP	Objective	Justification
		authenticating the upgrades. Hence, the public key certificates must correspond to the private keys on which the legitimate firmware upgrades were digitally signed. If the quality of the public key certificates is reduced (i.e. OE.CMC is not upheld), the policies governing the public key certificates in relation to firmware upgrades fails. Consequently, once the OE.CERT is upheld also the policy OSP.CERT is enforced.

9.3 Security requirements rationale

9.3.1 SFR dependency rationale

87 Table 15 demonstrates the mutual supportiveness of the SFR's for the TOE by demonstrating how the SFR dependencies are fulfilled by the TOE, and by justifying those dependencies that are not fulfilled.

88 The SARs relevant to the TOE constitute an evaluation assurance level EAL2 as defined in Common Criteria and include no extensions or augmentations. Therefore, as a complete evaluation assurance level, they are a mutually supportive set and require no further justification.

Table 15 – TOE SFR dependency demonstration

SFR	Dependency	Justification
FCS_CKM.1	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1b by the TOE FCS_CKM.4 by the TOE
FCS_CKM.4	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]	FCS_CKM.1 by the TOE
FCS_COP.1a	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	Neither FDP_ITC.1, FDP_ITC.2, FCS_CKM.1 nor FCS_CKM.4 to fulfil dependencies of FCS_COP.1a is implemented by the TOE. As FCS_COP.1a concerns with a one-way, unkeyed hash function SHA-1, it uses no cryptographic keys and the dependencies are not applicable.

SFR	Dependency	Justification
FCS_COP.1b	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1b by the TOE FCS_CKM.4 by the TOE.
FCS_COP.1c	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FDP_ITC.2 by the TOE FCS_CKM.4 is not relevant as the key material of concern is a public key of the firmware upgrade issuing authority and as such is a public value. Hence, upon expiry of the public key and issuance of the new public key certificate, no special measures are required to ensure the inability of an attacker to construct the expired public key from the residual data.
FCS_COP.1d	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1 by the TOE FCS_CKM.4 by the TOE
FDP_ACC.1a	FDP_ACF.1 Security attribute based access control	FDP_ACF.1a by the TOE
FDP_ACC.1b	FDP_ACF.1 Security attribute based access control	FDP_ACF.1b by the TOE
FDP_ACC.1c	FDP_ACF.1 Security attribute based access control	FDP_ACF.1c by the TOE
FDP_ACF.1a	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialization	FDP_ACC.1a by the TOE FMT_MSA.3a by the TOE
FDP_ACF.1b	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialization	FDP_ACC.1b by the TOE FMT_MSA.3b by the TOE
FDP_ACF.1c	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialization	FDP_ACC.1c by the TOE FMT_MSA.3c by the TOE

SFR	Dependency	Justification
FDP_ITC.2	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] FPT_TDC.1 Inter-TSF basic TSF data consistency	FDP_ACC.1c by the TOE FTP_ITC.1b by the TOE FPT_TDC.1 by the TOE
FDP_RIP.1a	None.	None.
FDP_RIP.1b	None.	None.
FDP_SDI.2	None.	None.
FIA_AFL.1	FIA_UAU.1 Timing of authentication	FIA_UAU.1 by the TOE
FIA_ATD.1	None.	None.
FIA_UAU.1	FIA_UID.1 Timing of identification	FIA_UID.1 by the TOE
FIA_UID.1	None.	None.
FMT_MSA.1a	[FDP_ACC.1 Subset access control, FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FDP_ACC.1a by the TOE FMT_SMR.1 by the TOE FMT_SMR.1 by the TOE
FMT_MSA.1b	[FDP_ACC.1 Subset access control, FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FDP_ACC.1b by the TOE FMT_SMR.1 by the TOE FMT_SMF.1 by the TOE
FMT_MSA.1c	[FDP_ACC.1 Subset access control, FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FDP_ACC.1c by the TOE FMT_SMR.1 by the TOE FMT_SMF.1 by the TOE
FMT_MSA.2	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FDP_ACC.1a, FDP_ACC.1b and FDP_ACC.1c by the TOE FMT_MSA.1a, FMT_MSA.1b and FMT_MSA.1c by the TOE FMT_SMR.1 by the TOE

SFR	Dependency	Justification
FMT_MSA.3a	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1a by the TOE FMT_SMR.1 by the TOE
FMT_MSA.3b	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1b by the TOE FMT_SMR.1 by the TOE
FMT_MSA.3c	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1c by the TOE FMT_SMR.1 by the TOE
FMT_SMR.1	FIA_UID.1 Timing of identification	FIA_UID.1 by the TOE
FMT_SMF.1	None.	None.
FPT_FLS.1	None.	None.
FPT_PHP.1	None.	None.
FPT_TDC.1	None.	None.
FPT_TST.1a	None.	None.
FPT_TST.1b	None.	None.
FTP_ITC.1a	None.	None.
FTP_ITC.1b	None.	None.
FTP_ITC.1c	None.	None.

9.3.2 Tracing of SFR to security objectives

89 Table 16 provides the mapping of the TOE SFRs and the security objectives for the TOE.

Table 16 – Mapping TOE SFRs to objectives

Objective	SFRs	Demonstration
O.USER_AUTH	FIA_AFL.1 FIA_ATD.1 FIA_UAU.1 FIA_UID.1 FTP_ITC.1c	O.USER_AUTH is preserved when the authenticity of each user is determined and any attempt of illegitimate users to masquerade as legitimate users is detected with a high likelihood and that the attempts to masquerade as legitimate users trigger the necessary actions required for protecting the TOE. FIA_ATD.1 provides a definition of the exact attributes based on which authentication decisions are made. Authentication of users is mandatory for TOE users prior to any access except those explicitly defined in FIA_UAU.1. Furthermore, user identification is required for all but accesses defined in FIA_UID.1.

Objective	SFRs	Demonstration
		In case of a failed authentication attempt, the exact actions for the TOE to take are defined in FIA_AFL.1. Finally, FTP_ITC.1 allows a trusted application being executed on the host PC to establish a trusted channel for a secure session between itself and the TOE. This allows further assurance that unauthentic parties may not gain access to the user data or configuration of the TOE by eavesdropping or injecting or modifying the messages between the legitimate application level software and the TOE. Together these SFR's constitute a mutually supportive whole that fully addresses security objective O.USER_AUTH by ensuring that only authenticated users can access the protected functions of the TOE and that sufficient controls are implemented to take relevant action in case of failed authentication attempts.
O.DATA_ACCESS	FDP_ACC.1a FDP_ACC.1b FDP_ACC.1c FDP_ACF.1a FDP_ACF.1b FDP_ACF.1c FMT_MSA.1a FMT_MSA.1b FMT_MSA.1c FMT_MSA.2 FMT_MSA.3a FMT_MSA.3b FMT_MSA.3c FMT_SMF.1 FMT_SMR.1 FTP_ITC.1c	O.DATA_ACCESS is preserved when Access to the data stored on the TOE is only granted to legitimate users. FDP_ACC.1a, FDP_ACC.1b and FDP_ACC.1c establish the basis for defining the policies that control access to the data stored on the TOE. They form the basis for the explicitly stated access control policies the TOE enforces. These policies are as stated in FDP_ACF.1a, FDP_ACF.1b and FDP_ACF.1c. The security attributes by which access control policies are expressed in the ST and enforced by the TOE must be managed to ensure that only authorized changes are allowed to their values. These controls are stated in the SFRs for managing the security attributes, namely FMT_MSA.1a, FMT_MSA.1b, FMT_MSA.1c, FMT_MSA.2, FMT_MSA.3a, FMT_MSA.3b and FMT_MSA.3c. Furthermore, the management functions for modifying the behavior of the security functions of the TOE are defined in FMT_SMF.1. The roles in which the end users may use the TOE are defined in FMT_SMR.1. Finally, FTP_ITC.1c allows an application level software being executed on the host PC to establish a trusted channel between itself and the TOE allowing a secure session and preventing any eavesdropping or message injection or modification that could constitute an illegitimate access to the TOE. Together, these SFRs constitute a mutually supportive whole that ensures that access to the TOE functions available to the end user is only granted to legitimate end users, and that the security functions can only be managed by legitimate users. This ensures that access to the data stored on the TOE can only be accessed by legitimate users.
O.TOE_INTEGRITY	FPT_FLS.1 FPT_PHP.1 FPT_TST.1a	O.TOE_INTEGRITY is preserved when The security state of the TOE, consisting of the secrets stored persistently on the TOE, the authentication state, and the lockdown state is protected against unauthorized modification and can only be

Objective	SFRs	Demonstration
	FPT_TST.1b FDP_RIP.1a FDP_RIP.1b FDP_SDI.2	altered by authorized and authenticated parties. FPT_FLS.1 ensures that whenever a power loss occurs, the TOE recovers into a secure state with restrictive access rights and that each user has to be authenticated prior to further data accesses being granted. FPT_PHP.1 ensures that the TOE is sufficiently protected against physical tampering so that the human user visually inspecting the TOE can unambiguously detect that the physical integrity of the TOE is violated. FPT_TST.1a and FPT_TST.1b ensure that the necessary tests are being carried out to ensure integrity of the functions generating the secrets, namely the cryptographic keys, protecting the secrets persistently stored on the TOE. FDP_RIP.1a and FDP_RIP.1b ensure that the container objects for the sensitive data – password representations and cryptographic keys – are properly cleared when the data is no longer needed or prior to the new object is generated to ensure that there is no residual data a potentially hostile software running on the host PC could attempt to exploit and that the newly generated cryptographic keys are free of any correlation with the keys previously generated and persistently stored on the TOE. FDP_SDI.2 ensures that the TOE provides necessary features to ensure that only legitimate changes to the values of security parameters may occur, and that violations of the integrity of those parameters are detected. Together these SFRs ensure that the integrity of the TOE is ensured and only legitimate chances to the values of secrets and security parameters are allowed, hence fully enforce O.TOE_INTEGRITY.
O.CMC	FMT_SMF.1 FTP_ITC.1a	O.CMC is preserved when the Authentication state and Lockdown state of the TOE can only be managed through the interface provided by the CMC software. FMT_SMF.1 concerns with the specification of management functions for the TOE, including the management of the Authentication state and management of the Lockdown state. FTP_ITC.1a governs the establishment of the trusted channel between the TOE and the CMC software, and states that the specific administrative functions for managing the Authentication state and Lockdown state are only available through that trusted channel. As the channel is logically separate from other communication channels between the TOE and external IT devices, the TOE is capable of ensuring that only legitimate CMC Software can be used for administering the TOE, hence fully enforces O.CMC.
O.CRYPTO	FCS_CKM.1 FCS_CKM.4	O.CRYPTO requires that The cryptographic keys, the underlying cryptographic algorithms and other cryptographic primitives are sufficiently secure and cryptographic operations sufficiently protect the user data and security parameters

Objective	SFRs	Demonstration
	FCS_COP.1a FCS_COP.1b FCS_COP.1c FCS_COP.1d	stored on the TOE. The quality of cryptographic keys is addressed by FCS_CKM.1 and FCS_CKM.4 that govern the generation and destruction of the keys used for protecting the user data persistently stored on the TOE. Cryptographic functions to protect the user data and security parameters (namely the reference authentication data and verification authentication data when stored on the TOE) are covered by FCS_COP.1a, FCS_COP.1b and FCS_COP.1d. Additionally, the verification of digital signatures used for authenticating the firmware upgrades is covered by FCS_COP.1c. Together these SFRs constitute the mutually supportive whole of key management and cryptographic operations that fully covers O.CRYPTO.
O.UPGRADE	FMT_SMF.1 FTP_ITC.1b FDP_ITC.2 FPT_TDC.1	O.UPGRADE requires that only legitimate firmware upgrades are accepted. Firmware upgrade is a legitimate management function of the TOE (defined in FMT_SMF.1) and hence allowed for authorized users. This is implemented by ensuring that a trusted channel exists between the TOE and the issuing authority to make sure that only legitimate upgrades are accepted. This covers FTP_ITC.1b. Further assurance on the authenticity of firmware upgrades is obtained by implementing measures to ensure that only valid public key certificates are accepted as a method of distributing the verification keys for firmware upgrades to the TOE. This covers FDP_ITC.2. Finally, measures are implemented to ensure that the public key certificates are of proper format and originate from legitimate sources as stated in FPT_TDC.1.

9.3.3 SAR justification

90 The set of SARs selected for the TOE constitute the entire evaluation assurance level EAL2 augmented with ALC_FLR.1 for flaw remediation.

91 Excluding the augmentation, as a basic EAL2 package, the set of SARs is an internally consistent and mutually supportive set of SARs.

92 The TOE is used in a potentially untrusted host PCs but when not in use, in the physical possession of the end user. The relevant attack scenarios are logical attacks occurring through the external interfaces of the TOE by malicious software potentially residing in the host PC.

93 The potentially malicious software running in the host PC can only access the TOE through the USB interface. Attack scenarios concerning internal interfaces are not accessible as access to those interfaces would require physical probing of the TOE.

94 Attack scenarios concerned with physically probing the TOE with expert skill and resources are not relevant. Tamper evidence of the TOE sealing is designed to provide basic level of assurance against undetected attempts to physically tamper with the TOE and to draw end user's attention to the possible lack of integrity of the TOE.

95 Consequently, it is sufficient for the TOE to be engineered to demonstrate sufficient assurance against logical attacks by malicious software through externally visible interfaces as demonstrated by EAL2.

96 The TOE implements additional measures to allow TOE developer to issue firmware upgrades to the TOE once issued to the end user. While the TOE implements functional measures to ensure the authenticity of the upgrades, there is also a significant assurance component related to the trustworthiness of the upgrades. The developer must demonstrate a comprehensive set of measures followed to ensure that only legitimate and authentic firmware upgrades are issued for the TOE. The basic EAL2 package does not include such measures as the ability for upgrading the TOE is not implemented in all TOEs.

97 The TOE described in this ST, however, does implement such ability and for comprehensive assurance, the basic EAL2 package is augmented with ALC_FLR.1 to ensure that the upgrade procedures are sufficiently trustworthy. As the EAL2 selected for the TOE only provides a baseline of assurance, the developers determine that assurance component ALC_FLR.1 is sufficient for consistency.