USBK Cryptobridge v2.0
For Model A101 and Model A103

SECURITY TARGET

Version 0.9

August 22th 2011

Table of Contents

1. ST INTRODUCTION 5
   1.1. ST Reference and TOE Reference 5
   1.2. Document Conventions, Terminology & Acronyms 5
      1.2.1. Conventions 5
      1.2.2. Terminology 6
      1.2.3. Acronyms 7
   1.3. TOE Overview 8
      1.3.1. Usage and Major Security Features of the TOE 8
      1.3.2. TOE Type 10
      1.3.3. Required non-TOE hardware/software/firmware 11
   1.4. TOE Description 11
      1.4.1. Physical Scope 12
      1.4.2. Logical Scope 15
2. CONFORMANCE CLAIM 17
   2.1. CC Conformance Claim 17
   2.2. PP and Package Claim 17
      2.2.1. Protection Profile (PP) Claim 17
      2.2.2. Package Claim 17
   2.3. Conformance Rationale 17
3. SECURITY PROBLEM DEFINITION 18
   3.1. Threats 18
   3.2. Organizational Security Policies 19
   3.3. Assumptions 20
4. SECURITY OBJECTIVES 21
   4.1. Security Objectives for the TOE 21
   4.2. Security Objectives for the Operational Environment 22
   4.3. Security Objective Rationale 23
5. EXTENDED COMPONENT DEFINITION 26
6. SECURITY REQUIREMENTS 27
   6.1. Security Functional Requirements 27
      6.1.1. Cryptographic Support 29
      6.1.2. User Data Protection 30
      6.1.3. Identification and Authentication 37
      6.1.4. Security Management 38
      6.1.5. Protection of the TSF 39
      6.1.6. Resource Utilization 40
   6.2. Security Assurance Requirements 40
   6.3. Security Requirements Rationale 41
   6.4. Security Assurance Requirements Rationale 45
7. TOE SUMMARY SPECIFICATIONS 46
   7.1. User Request Management 46
   7.2. Main System Control 47

7.3. Bridge Control 47
7.4. Secure Bridge 48
7.5. Security Attributes 48
7.6. Program Memory Protection 48

List of Tables

Table 1 - ST and TOE References 5
Table 2 - Subjects relevant to the TOE 18
Table 3 - Threats relevant to the TOE 19
Table 4 - Organizational Security Policies 19
Table 5 - Assumptions 20
Table 6 - Security Objectives for the TOE 21
Table 7 - Security Objectives for the Operational Environment 22
Table 8 - Coverage of Assumptions, Threats and Organisational Security Policies By Security Objectives 23
Table 9 - TOE Security Functional Requirements 27
Table 10 – TOE Security Assurance Requirements 40
Table 11 - Mapping of SFR to Objectives 41
Table 12 - SFR Dependecies 43

List of Figures

Figure 1 - Generic Usage of the TOE 9
Figure 2 - Physical Scope of the TOE 12
Figure 3 - Firmware View of Physical Scope for TOE 13

Version History

Version No	**Reason for Change**	**Release Date**
0.1	First Release	17/01/2011
0.2	Developer Review&Update	19/01/2011
0.3	Update in TSS chapter 7	26/01/2011
0.4	Update according to the minutes of kick-off	10/03/2011
0.5	Update according to GR1 and change in standard of random key generation	25/04/2011
0.6	Update according to GR2	02/05/2011
0.7	Update according to GR3	11/05/2011
0.8	Update according to the meeting report dated 18.05.2011	23/05/2011
0.9	Update according to GR8	22/08/2011

Approvals

Name	**Role**	**Date**
Emre ÇAKIR	ST Author (Miron Yazılım)	
Mehmet ÇAKIR	ST Author (Miron Yazılım)	
Tülin YAMAN		
Zeki ÜNAL		

1. ST INTRODUCTION

This section presents the following information:

- Identifies the Security Target (ST) and Target of Evaluation (TOE);
- Specifies the ST conventions,
- Defines the terminology and acronyms used in the ST,
- Defines TOE overview and TOE description.

1.1. ST Reference and TOE Reference

Table 1 - ST and TOE References

ST Title:	USBK Cryptobridge v2.0 Model A101 and Model A103 Security Target v 0.9
ST Version:	v 0.9
TOE Identification:	USBK Cryptobridge v2.0 Model A101 and Model A103
CC Identification:	Common Criteria for Information Technology Security Evaluations, version 3.1R3
Keywords:	Disk Encryption, Data Protection, Secure USB Drives, AES, Cryptobridge.

1.2. Document Conventions, Terminology & Acronyms

This section specifies the formatting information used in the ST.

1.2.1. Conventions

In this Security Target some notations and conventions which are taken from the Common Criteria v3.1R3 have been used in order to guide the reader.

During the specification of the functional requirements under Section 6, the functional components are interpreted according to the “assignment” and “selection” operations.

The outcome of the assignment operations are shown with **bold** and identified between “[**brackets**]”.

The outcome of the selection operations are shown with **bold** and **underlined** and identified between "[**brackets**]".

Iterated functional requirement components are shown with a "/**IDENTIFIER**" for the components which used more than once with varying operations.

Under the term "**Application Note**", an informal explanation added under some of the functional requirements in order to highlight or to describe the component in detail.

1.2.2. Terminology

The following terminology is used in this Security Target:

AES: Advanced Encryption Standard (AES) is a symetric-key encryption defined in Federal Information Processing Standard (FIPS) Publication 197. The standard comprises three block ciphers, AES-128, AES-192 and AES-256.

Cryptanalysis: A methodology used by threat agents in order to break the cryptographic protection of the TOE.

Cryptographic Operation: Encryption and decryption operations inside the TOE with AES algorithm.

Encryption Key: The 128-bit or 256-bit AES key used by the TOE for encryption process.

Non-Volatile Memory: The memory portion developed with flash technology inside the integrated circuit.

Random Key Generation: 128-bit or 256-bit AES key generated by Pseudo-random number generation (PRNG) compliant with ANSI X9.31 which is seeding with USB channel frame number.

Transfer Key: 128-bit or 256-bit AES key used for encrypt/decrypt during the transmission of user data (disk content) to an external drive.

Storage Key: 256-bit AES key used to encrypt/decrypt user keys and user password stored in the non-volatile memory.

Host System: The system on which the TOE is plugged and used. (desktop pc, laptop pc, testing equipment .etc)

Back Disk: All kind of USB flash drives and USB external harddrives plugged into the TOE for secure data transfer.

Authorized User: Owner of the TOE.TOE recognizes the Authorized User by valid password.

Any User: Beside authorized user, other users on host system that can access the back disk when TOE is activated. Also, host system is considered as any user.

Retry Number: Number of consecutive failed authentication attempts which is incremented each time when a failure occurs and reset each time when an authentication is successful.

File System: A method for storing files in any kind of mass storage device. (etc. FAT32, NTFS, EXT3)

Small Computer System Interface: SCSI is a set of standards for physically connecting and transferring data between computers and peripheral devices.

Mass Storage Device Driver: A type of device driver which supports the interface with all kind of USB flash drives and USB external harddrives

Real Time Operating System: RTOS is an operating system which is intended to serve real-time application requests.

Logical Unit Number: Number of logical device interface in a single physical channel.

Model A101: A USBK model that can carry only one AES key for encrypting UserData.

Model A103: A USBK model that can carry three AES keys for encrypting UserData. User can select one of them during activation.

AVR: A family of microprocessor of vendor ATMEL Corporation. These processors are placed with other peripheral modules in a package and marketed as microcontroller under the brand of AVR.

1.2.3. Acronyms

AES	Advanced Encryption Standard
CC	Common Criteria
SCSI	Small Computer System Interface
TOE	Target of Evaluation
ST	Security Target
SFR	Security Functional Requirements

SAR	Security Assurance Requirements
EAL	Evaluation Assurance Level
MSD	Mass Storage Device
LUN	Logical Unit Number
FIPS	Federal Information Processing Standard
RTOS	Real-time Operating System
USB	Universal Serial Bus

1.3. TOE Overview

1.3.1. Usage and Major Security Features of the TOE

USBK Cryptobridge, the TOE, is a disk encryption product which the users have the ability to encrypt/decrypt all data transmitted between host system and a back disk. Since the main feature of the TOE is encrypting/decrypting the transmitted data from/to the TOE, the users of the TOE are not restricted with a limited disk space, on the contrary, they have the ability to use TOE with any USB flash drives and USB external harddrives which can be plugged to the TOE.

The TOE is also not dependent to any operating system on the host system which the encrypted data will be transmitted from. The TOE communicate with the host system with Small Computer System Interface (SCSI). TOE is supporting predefined vendor specific SCSI commands. An application on the host system can be used as an interface between user and SCSI. This type of communication between host system and TOE provides independence from the operating system.

For MS Windows Operating Systems the application is provided by TOE, for other type of Operating Systems vendor will provide the installation file through the vendor website.

On the other hand, the TOE will also support another interface for managing the TOE functionality through a simple text editor. This methodology can be used for the operating systems for which an application in the vendor website is not provided.

TOE is delivered to its customers with two different models called Model A101 and Model A103 provide the opportunity to use single cryptographic key where Model A103 support up to three keys. The customers of Model A103, select the key during activation and use TOE according to its operational guidance and on the other hand Model A101 use the only key supported by TOE. All the security functionalities defined in this ST are both valid for two TOE models as well as the assurance measures.

TOE supports cryptographic operation according to the supported AES key size. The users of the TOE can either generate a 128-bit or 256-bit AES key.

The following figure is showing the generic usage of the TOE.

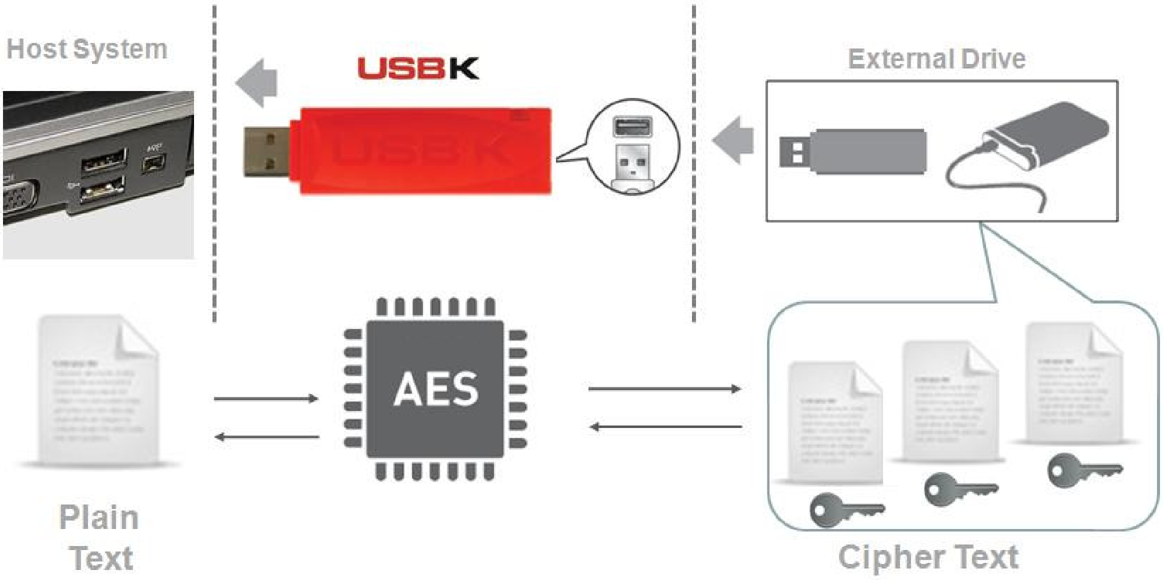

Figure 1 - Generic Usage of the TOE

Initial State of TOE;

The TOE can be assumed in initial state in three conditions. Either when the user purchase the TOE first time, after the retry number dropped to zero or after user data integrity is lost.

At initial state, transfer key(s) have been randomly generated by TOE.

At initial state, TOE enforce the user to provide user password. All other management functions are inaccessible before setting the user password.

Deactivate State of TOE - Configuring the TOE;

The user can change the following settings;

- User password,

- Transfer key(s),
- Auto-activation value,

The user can assign names for the following;

- Transfer key(s),
- Device,

TOE will request authorization for each operation defined above.

Activate State of TOE - Normal Usage;

Transfer functionality of the TOE will be activated by user after selecting the key with correct user password.

The user can plug a back disk to the TOE. Host system will recognize the back disk as decrypted. If the back disk is used for the first time with the active key of TOE, operating system will announce that back disk is unformatted. The user can transfer the data encrypted right after formatting the disk.

Transfer session with the back disk will be terminated upon deactivation.

File system information and user data stored in the back disk is always encrypted with 128-bit or 256-bit AES key which is user defined at setting and selected at activation.

Users of the TOE can configure the TOE as auto activated. With this settings, preselected transfer key is activated automatically. This feature is provided for integration with host systems without any interface for user authorization such as testing equipments.

Programming Mode of TOE

TOE is taken into programming mode when firmware upgrade is required.

TOE itself also goes into programming mode when it detects corruption in firmware.

1.3.2. TOE Type

TOE is an integrated system included firmware and hardware modules which can be categorized as a data protection product.

The TOE provides the opportunity to its users to encrypt/decrypt their data with an AES 128-bit or 256-bit key during the transmission to an external drive.

1.3.3. Required non-TOE hardware/software/firmware

TOE should be configured before the usage in a host system with the following minimum configuration;

- USB host interface,
- MSD class drivers with multiple LUN support,
- FAT16 file system,
- Text editor,
- A display and I/O unit.

TOE can be used in any host system with a USB interface and MSD driver and can encrypt the transmitted data to any external drive with USB interface.

1.4. TOE Description

TOE is an integrated system which provides users to protect their data after the transmission to a back disk. The components of TOE are integrated with a vendor specific firmware which enforce encrypt/decrypt operations during data transfer.

Upon the initialisation and activation of the TOE, the authorized user can transfer data by encrypting it with a 128-bit or 256- bit AES transfer key, according to his/her choice, to a formatted back disk. Also authorized user can perform the decryption operation for the encrypted files in a back disk.

The user can configure the security functions and user security attributes of the TOE only if the TOE is deactivated. Appropriate user authentication is performed during configuration.

Two different models of TOE can be used which the only difference is the number of supported transfer keys. One of the model is supporting only one transfer key and the other is supporting three different transfer keys.

The firmware is the same for both models of TOE. Only difference is the global setting-NumberOfKeys- that can be 1 or 3. Firmware acts according to this setting.

During the activation selection of the transfer key is supported to the user. But for TOE model A101, there is no chance other than 1. The physical and logical scope of the TOE is provided below for a better understanding of the evaluation scope and context.

1.4.1. Physical Scope

The following figures are showing the physical scope of the TOE and interface between the modules and TOE units.

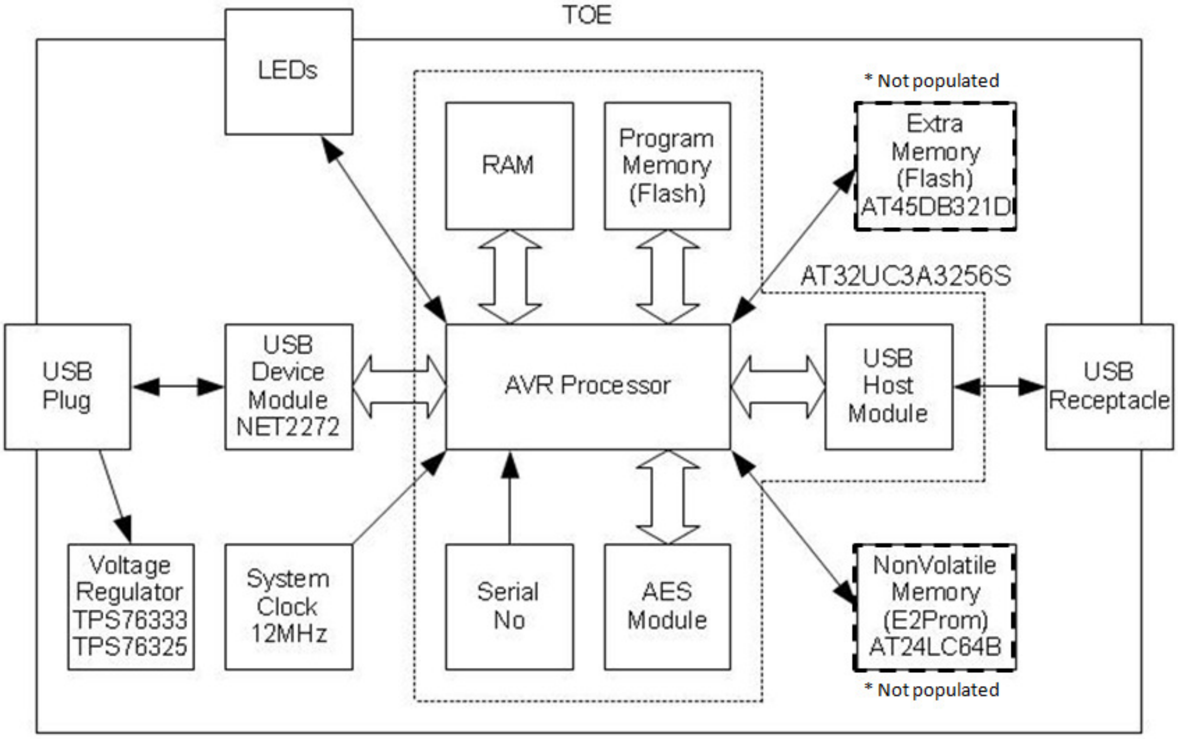

Figure 2 - Physical Scope of the TOE[1]

[1] Figures are intentionally highly abstract for a better understanding of the ST reader but sufficient at this stage to provide an overview.

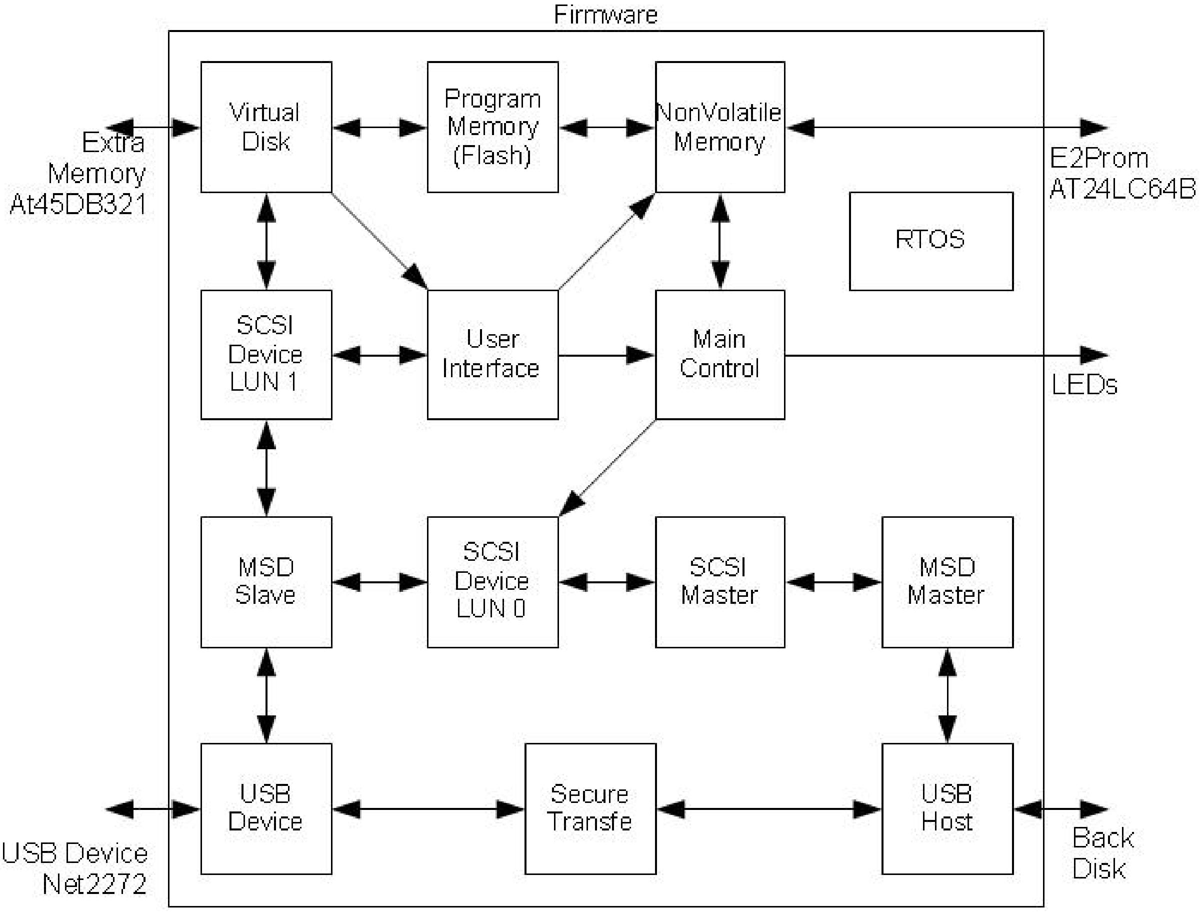

Figure 3 - Firmware View of Physical Scope for TOE

TOE consists of hardware module and firmware module. Hardware provides an execution environment for the firmware. Program is placed in program memory (Flash) and executed on RAM by AVR processor. All of them are in the microcontroller module.

- **Hardware Module:** AT32UC3A3256S is a 32 bit microcontroller module that includes
  - a 32 bit AVR processor,
  - 128KB of RAM,
  - 256KB of Program Memory (Flash),
  - an AES module,

- a USB On-The-Go module,
- 15 byte unique serial number,
- 110 General Purpose IOs (Only 3 of them is used in TOE)
- four USARTs(one of them used in UART mode in TOE)
- others (unused functionalities in TOE).

Manufacturer is ATMEL Corporation (www.atmel.com). Chip is 144 pin in ball grid form. The firmware is programmed into program memory of this integrated circuit. AES module is used to every encryption/decryption operation of TOE. USB module is used to perform Host functionality for Back disk.

An AT24LC64B E2Prom chip of Atmel Corporation is nonvolatile memory capacity for configuration data but not populated on printed circuit board. program memory (Flash) is used for this purpose.

An AT45DB321D Flash chip of Atmel Corporation is extra memory capacity for virtual disk content but not populated on printed circuit board. Instead, program memory (Flash) is used for this purpose.

USB Device functionality is supplied by NET2272chip. Manufacturer is PLX Technology (www.plxtech.com) TOE has two USB connectors both A type, device side is plug and host side is receptacle. Additionally there are 2 LEDs on the TOE.

Power taken from host system is isused in TOE and back disk. Regulated voltages 3.3V and 2.5V are supplied for hardware of the TOE by TPS76333 and TPS76325. Manufacturer is Texas Instruments (www.ti.com).

A special PCB is manufactured for the physical components above to perform proper functionality. TOE is using version 2.1 for hardware module.

- **Firmware Module:** Firmware is consist of modules running at the same time by the support of RTOS. It is a realtime multitasking environment. USB device provides USB device functionality. MSD Slave provides MSD slave functionality. It supports two SCSI devices. One represents the virtual disk and other represents the back disk. SCSI device LUN1 understands vendor specific SCSI commands. SCSI device LUN 0 is full functional only if transfer key is active. SCSI master, MSD master and USB host are protocol stack to handle the Back disk properly. Secure transfer module perform encryption/decryption on the transfer phase of communication protocol. It must be loaded with the transfer key for proper operation. Some permanent memory content is shown as a disk i.e., virtual disk. application software, UserMenu.Txt file and some other files are in this virtual disk. User menu operations are performed at this module. Main control makes all initial operations and securely store and recall operations of critical data by use of AES module with storage key.

Version of the developed firmware which performs the core functionality of the TOE with these components is v2.5.

1.4.2. Logical Scope

Cryptography: TOE provides the following two types of cryptographic operation with AES algorithm;

- Encryption/Decryption of user security attributes: Encryption/Decryption of user security attributes (user password and transfer key) into non-volatile memory by encrypting with 256-bit AES storage key. This storage key is generated randomly during first run of the firmware. The storage key is generated once and used during the life-cycle of the firmware.
- Encryption/Decryption of user data: Encryption/Decryption of transferred data between host system and back disk by using 128-bit or 256-bit AES transfer key according to user selection during definition of transfer key(s). Initially TOE fills transfer key(s) with randomly generated 256-bit one(s) and user is able to change them during setting up of TOE. User may make TOE generate random 128-bit or 256-bit AES key in order to get a stronger key. User data, encrypted/decrypted during this cryptographic operation includes both user files and file system information.

Data Protection: TOE provides data protection and confidentiality of user data by encrypting the data on the fly with AES algorithm. TOE also protect user security attributes by encrypting them with AES algorithm. TOE does not allow reading program memory which contains security attributes of user and TOE. This access is only valid after the erasure of the program memory. TOE also provides integrity of user security attributes and program memory by cycling redundancy check (CRC).

Authentication: TOE enforces users to provide password for each operation requests except deactivation.

Management: TOE allows users to change/set values for the parameters below;

- auto activation,

- user password,
- transfer key,
- device label,
- transfer key label

Testing: During the start-up of TOE, the following self tests are conducted;

- CRC check for program memory and user security attributes,
- Control of AES encryption/decryption operations,
- Control of communication bus within the TOE,

Resource Utilisation: User security attributes are encrypted and stored with a back up copy. According to the result of CRC checking, the back up copy of user security attributes will be overwritten to the corrupted one.

2. CONFORMANCE CLAIM

The conformance claims regarding to the TOE are stated in the following sub-sections.

2.1. CC Conformance Claim

This TOE and ST are consistent with the following specifications:

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; Version 3.1, Revision 3, July 2009.
- Common Criteria for Information Technology Security Evaluation Part 2: Security functional requirements, Version 3.1, Revision 3, July 2009.
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance requirements Version 3.1, Revision 3, July 2009.

This Security Target claims to be CC Part2 comformant and CC Part3 comformant.

2.2. PP and Package Claim

2.2.1.Protection Profile (PP) Claim

This ST makes no conformance claims to any certified Protection Profile.

2.2.2.Package Claim

The TOE is conformant to EAL2.

This Security Target elaborated in conformance with “Common Criteria for Information Technology Security Evaluation, Version 3.1 rev 3” contains the IT security requirements of the TOE and specifies the functional and assurance security measures to meet the stated requirements.

2.3. Conformance Rationale

The assurance level of EAL2 is considered to be most appropriate for this type of TOE since it is intended to defend against attacks that can be made given the assumptions, and the threats defined in chapter 3.

3. SECURITY PROBLEM DEFINITION

3.1. Threats

Assets:

- Security attributes of TOE such as
    - Storage Key (key used for storing user security attributes)
    - Retry Number[2]
    - Initial Vectors.
- Security attributes of the user. such as
    - Transfer Key(s)
    - User Password
- AutoActivation setting[3]
- User data, files to be stored in the back disk.

Subjects:

Table 2 - Subjects relevant to the TOE

Subjects	Description
U.OWNER	The Authorized User
U.BADMAN	A Threat Agent that has a chance of use USBK and Back disk of real owner (user). This agent may be any kind of person, malware, virus, trojan, worm, etc.
U.CRYPTANALYST	A Threat Agent that has plenty of cryptographic knowledge. This agent can get the Back disk and try to decrypt the content (ie User Data). This agent can get residueal of erasure Flash memory of

[2] The value is not secret as shown to the user as retry limit, but it is valuable to be protected from any unauthorized modification.

[3] The setting is not secret as shown to the user, but it is valuable to be protected from any unauthorized modification.

	TOE and try to decrypt the content (i.e. User Security Attributes).
U.HARDANALYST	A Threat Agent that has plenty of hardware knowledge. This agent probes the USBK hardware and tries to read the security attributes.
U.NATURALCAUSE	A Threat Agent that has a plenty of energy to change the bits of firmware.

Table 3 - Threats relevant to the TOE

Threats	Description
T.UNAUTHORISED	U.BADMAN can gain access to the user data on Back disk by activating TOE with correct password. TOE can not recognise the difference between U.OWNER and U.BADMAN since either provides correct password.
T.PROBING_NON-VOLATILE MEMORY	U.HARDANALYST can reveal the transfer key(s), user password by probing the non-volatile memory on the integrated circuit.
T.PROBING_PROGRAM MEMORY	U.HARDANALYST can reveal the storage key by probing the program memory on the integrated circuit.
T.CORRUPTION	The integrity of user security attributes and firmware might be corrupted by U.NATURALCAUSE .

3.2. Organizational Security Policies

Table 4 - Organizational Security Policies

Policy	Description
OSP.CRYPTANALYSIS	The cryptographic keys (transfer keys) , on which cryptographic algoritms depends, must be sufficiently strong to protect encrypted user data againts trial of U.CRYPTOANALYST. U.OWNER should take responsibility. TOE can generate random keys for U.OWNER. TOE implements AES as cryptographic algorithm which is mathematically strong against cryptanalysis.

3.3. Assumptions

Table 5 - Assumptions

Assumptions	Description
A.USER	U.OWNER should protect their security attributes (user passwords, transfer keys) from disclosure. He/she is aware of the value of his/her data and is strongly intented to protect it.
A.HOST	Operational environment should be protected against virus, trojan, malware or any type of network attacks which can compromise the security of data transfer between the host system and TOE. Operational environment should also be trusted.
A.OPERATIONAL ENVIRONMENT	Operational environment does not allow an attacker to access the back disk when sensitive data is accessible to rightful user on the host system.
A.AUTOACTIVATION	Users should physically protect the TOE if they set the auto activation state “on”.

4. SECURITY OBJECTIVES

4.1. Security Objectives for the TOE

Table 6 - Security Objectives for the TOE

Security Objective	Description
O.AUTHENTICATION	TOE shall enforce setting of password from user at first usage of it. TOE shall require password from user for authentication with every command, except "Deactivation", and "Generate Random Key" commands.
O.AUTHATTEMPT	TOE shall destroy security attributes of the user including user password, transfer key and return to fabric defaults after three failed authentication attempts.
O.CRYPTOGRAPHY	TOE shall encrypt/decrypt user data during transfer to/from back disk. The encryption/decryption is done by selected transfer key at Activation of TOE and AES algorithm that meets the following: U.S Department of Commerce / Natiaonal Institute of Standardsand Technology Advanced Encryption Standard (AES), FIPS PUB 197, 2001 November 26 The key size may 128 or 256 bit according to setting of that transfer key.
O.FIRMWARE PROTECTION	TOE shall prevent access to program memory unless TOE firmware including storage key is erased.
O.SELFTEST	TOE shall conduct self test before startup. The tests cover the functionality of modules and integrity of data storages (incudes firmware and security attributes of user and TOE.).
O.USERATTR	TOE shall store transfer key(s) and password in secure manner. The encryption/decryption is done by 256-bit storage key which is randomly generated at first usage of TOE.

4.2. Security Objectives for the Operational Environment

Table 7 - Security Objectives for the Operational Environment

Security Objective	Description
OE.USER	Users shall protect their user credentials in a secure manner.
OE.AUTOACTIVATION	Users shall physically protect TOE when they set the auto-activation function on.
OE.HOST	TOE Host shall be trusted and protected against vulnerabilities or threats. Host shouldn't be infected by any malware that can grab user security attributes during usage of TOE. Or it shouldn't copy user data stored in back disk over activated TOE.
OE.OPERATIONALENV	Operational Environment shall be secure when user data is accessible through TOE since activated TOE is not privacy of its user and all users of Host and its environment can reach back disk over activated TOE. Host may be open to usage of other users in a network environment. User is responsible on sharing disks including back disk over TOE. Any user or any malware running on environment shouldn't copy, modify or delete user data stored in back disk.
OE.PASSWORD	User password shall be strong enough as that can not be guessed easily. It is very important to create complex passwords that have a mixture of characters. An ideal password should be long enough, case-sensitive and consist of with letters, numbers and punctuations. It is also important to avoid using sequential numbers or characters such as 12345678 and abcdefg, repeating numbers such as 222222, your name, birthday date and important dates and years as password.

4.3. Security Objective Rationale

The following table is showing the mappings between security objectives and threats and assumptions. The table is also stating the rationales for the mappings.

Table 8 - Coverage of Assumptions, Threats and Organisational Security Policies By Security Objectives

Threat / Assumption/ Policy	Security Objective	Rationale
T.UNAUTHORISED	O.AUTHENTICATION O.AUTHATTEMP OE.PASSWORD	The objective O.AUTHANTICATION guarantees that TOE user shall be authenticated before conducting any actions. The objective of O.AUTHATTEMPT ensures that TOE will return to factory settings upon three failed authentication attempts. The objective of OE.PASSWORD is to ensure that users will select strong passwords.

Threat / Assumption/ Policy	**Security Objective**	**Rationale**
OSP.CRYPTANALYSIS	O.CRYPTOGRAPHY OE. PASSWORD	The objective of O.CRYPTOGRAPHY shall encrypt the user data which is transferred to the back disk through TOE and shall decrypt the user data which is transferred from back disk to the TOE using a mathematically strong cryptographic algorithm. The objective of OE.PASSWORD is to ensure that users will select strong passwords.
T.PROBING_NON-VOLATILE MEMORY	O.FIRMWARE PROTECTION O.USERATTR	The objective of O.FIRMWARE PROTECTION is to prevent access to transfer key(s) and user password. The objective of O.USERATTR shall provide a security mechanism against reading of transfer key(s) and user password
T.PROBING_PROGRAM MEMORY	O.FIRMWARE PROTECTION	The objective of O.FIRMWARE PROTECTION is to prevent access to firmware and storage key.

Threat / Assumption/ Policy	Security Objective	Rationale
T.CORRUPTION	O.SELFTEST	The objective of O.SELFTEST shall initiate a self test for integrity check of firmware and user security attributes during start-up.
A.USER	OE.USER OE.PASSWORD	The objective of OE.USER shall enforce users to store their user credentials securely. The objective of OE.PASSWORD shall enforce users to generate strong passwords.
A.HOST	OE.HOST	The objective of OE.HOST shall guarantees that host device which is connected with the TOE must be free from any vulnerabilities or threats.
A.OPERATIONAL ENVIRONMENT	OE.OPERATIONALENV	The objective of OE.OPERATIONALENV guarantees that anyone except the TOE user can access to the back disk when TOE user is accessing to the back disk.
A.AUTOACTIVATION	OE.AUTOACTIVATION	The objective of OE.AUTOACTIVATION is to enforce users to physically protect TOE while the auto-activation function is on.

5. EXTENDED COMPONENT DEFINITION

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 and shall be omitted.

6. SECURITY REQUIREMENTS

6.1. Security Functional Requirements

The following list of security functional requirements are selected to cover the security objectives for the TOE. Some of the following requirements are iterated in order to apply the requirement with different assignment and/or selection operations in different circumstances.

Table 9 - TOE Security Functional Requirements

Requirement Class	Requirement Component	Dependencies
FCS: Cryptographic Support	FCS_CKM.1	FCS_CKM.2 or FCS_COP.1, FCS_CKM.4
	FCS_CKM.3	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1, FCS_CKM.4
	FCS_CKM.4	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1
	FCS_COP.1	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1, FCS_CKM.4
FDP: User Data Protection	FDP_ACC.1/**USERSECURITYATTRIBUTES** FDP_ACC.1/ **CRYPTOKEYACCESS**	FDP_ACF.1
	FDP_ACF.1/**USERSECURITYATTRIBUTES** FDP_ACF.1//**CRYPTOKEYACCESS**	FDP_ACC.1, FMT_MSA.3
	FDP_IFC.1	FDP_IFF.1
	FDP_IFF.1	FDP_IFC.1, FMT_MSA.3
	FDP_RIP.1	No Dependencies
	FDP_SDI.2/**USERATTRIBUTES Stored data**	No Dependencies

Requirement Class	Requirement Component	Dependencies
	integrity monitoring and action FDP_SDI.2/ **PROGRAMMEMORY Stored data integrity monitoring and action**	
FIA: Identification & Authentication	FIA_AFL.1	FIA_UAU.1
	FIA_ATD.1	No Dependencies
	FIA_UAU.1	FIA_UID.1
FMT: Security Management	FMT_MOF.1	FMT_SMR.1, FMT_SMF.1
	FMT_MSA.1	FDP_ACC.1, or FDP_IFC.1, FMT_SMR.1, FMT_SMF.1
	FMT_MSA.3/**USERDATA Static attribute initialization** FMT_MSA.3//**USERSECURITYATTRIBUTES Static attribute initialization**	FMT_MSA.1, FMT_SMR.1
	FMT_MTD.1	FMT_SMR.1, FMT_SMF.1
	FMT_SMF.1	No Dependencies
	FMT_SMR.1	FIA_UID.1
FPT: Protection of the TSF	FPT_FLS.1	No Dependencies
	FPT_TST.1	No Dependencies
FRU: Resource Utilization	FRU_FLT.2	FPT_FLS.1

6.1.1. Cryptographic Support

FCS_COP.1

FCS_COP.1.1 The TSF shall perform [**encryption, decryption**][4] in accordance with a specified cryptographic algorithm [**AES in CBC mode**][5] and cryptographic key sizes [**128-bit, 256-bit**][6] that meet the following: [**AES standard (FIPS 197) and AES modes of operation (NIST Special Publication 800-38A, Recommendation for Block Cipher Modes of Operation-Methods and Techniques)**][7].

FCS_CKM.1

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**pseudo-random number generation (PRNG)**][8] and specified cryptographic key sizes [**128-bit, 256-bit**][9] that meet the following: [**ANSI X9.31**][10].

Application Note: The randomly generated number is directly used as cryptographic key.

FCS_CKM.3

FCS_CKM.3.1 The TSF shall perform [**key activation**][11] in accordance with a specified cryptographic key access method [**request with user authentication**][12] that meets the following: [**none**][13].

FCS_CKM.4

[4] Assignment: List of cryptographic operations

[5] Assignment: Cryptographic algorithm

[6] Assignment: Cryptographic key sizes

[7] Assignment: List of standards

[8] Assignment: Cryptographic key generation algorithm

[9] Assignment: Cryptographic key sizes

[10] Assignment: List of standards

[11] Assignment: Type of cryptographic key access

[12] Assignment: Cryptographic key access method

[13] Assignment: List of standards

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [**erase the previous transfer key, overwrite new transfer key**][14] that meets the following: [**none**][15].

6.1.2. User Data Protection

FDP_ACC.1/USERSECURITYATTRIBUTES

FDP_ACC.1.1 The TSF shall enforce the [**user security attributes access control policy**] [16] on

[

List of Subjects;

Authorized User

List of Objects;

User Password

Transfer Key(s)

Device Label

Transfer Key Label

Autoactivation Attribute

Activation State

List of Operations;

Change

List of Attribute Values;

Initial State Value

] [17].

[14] Assignment: Cryptographic key destruction method

[15] Assignment: List of standards

[16] Assignment: Access control SFP

[17] Assignment: List of subjects, objects, and operations among subjects and objects covered by the SFP

Application Note: If TOE is in the initial state, user can only set user password, all other attributes are in their fabric default values.

FDP_ACC.1/CRYPTOKEYACCESS

FDP_ACC.1.1 The TSF shall enforce the [**cryptographic key access policy**] [18] on

[

List of Subjects;

Any User

List of Objects;

Transfer Key(s)

User Password

List of Operations;

Read

] [19].

FDP_ACF.1/USERSECURITYATTRIBUTES

FDP_ACF.1.1 The TSF shall enforce the [**user security attributes access control policy**][20] to objects based on the following: [

List of Subjects;

Authorized User

List of Objects;

User Password

Transfer Key(s)

Device Label

[18] Assignment: Access control SFP

[19] Assignment: List of subjects, objects, and operations among subjects and objects covered by the SFP

[20] Assignment: Access control SFP

Transfer Key Label(s)

Autoactivation Attribute

Security Attribute of Subjects;

User Password

Security Attribute of Objects;

Initial State

][21].

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

If TOE is in its initial state only allow authorized user to set "user password", else allow authorized user to perform list of operations.

Deny list of operations if the user can not successfully authenticate to the TOE.

][22].

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [**none**][23].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [**none**][24].

FDP_ACF.1/CRYPTOKEYACCESS

FDP_ACF.1.1 The TSF shall enforce the [**cryptographic key access policy**][25] to objects based on the following: [

[21] Assignment: List of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes

[22] Assignment: Rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects

[23] Assignment: Rules, based on security attributes, that explicitly authorise access of subjects to objects

[24] Assignment: Rules, based on security attributes, that explicitly deny access of subjects to objects

[25] Assignment: Access control SFP

List of Subjects;

Any User

List of Objects;

Transfer Key(s)

User Password

Security Attribute of Subjects;

None

Security Attribute of Objects;

None

][26].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [**deny read**][27].

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [**none**][28].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [**none**][29].

FDP_IFC.1 Subset information flow control

FDP_IFC.1.1 The TSF shall enforce the **[user data flow control policy]**[30] on **[**

List of Subjects;

Any User

[26] Assignment: List of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes

[27] Assignment: Rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects

[28] Assignment: Rules, based on security attributes, that explicitly authorise access of subjects to objects

[29] Assignment: Rules, based on security attributes, that explicitly deny access of subjects to objects

[30] Assignment: Information flow control SFP

Host System

Back Disk

List of Objects;

User Data

List of Operations;

Data transfer

][31]**.**

Application Note: Although host system is a platform for users, it may acts also as user.

Application Note: Back disk is one side of data flow but it does not originate any transfer.

FDP_IFF.1 Simple Security Attributes

FDP_IFF.1.1 The TSF shall enforce the **[user data flow control policy]**[32] based on the following types of subject and information security attributes:

[

List of Subjects;

Authorized User

Any User

Host System

List of Information;

User Data

List of attributes of Subjects;

User Password

[31] Assignment: List of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP

[32] Assignment: Information flow control SFP

List of attributes of Information;

Flow Direction

][33].

Application Note : If authorized user (owner of the TOE) activates the TOE any user on the host system can acces user data on back disk through TOE. So, any user is different from authorized user.

Application Note : Although host system is a platform for users, it may acts also as a user.

FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via controlled operation if the following rules hold: **[**

Read request of subjects on host system side causes decryptive information flow through the TOE from back disk.

Write request of subjects on host system side causes encryptive information flow through the TOE to back disk.

][34].

FDP_IFF.1.3 The TSF shall enforce the

[If activation state of TOE is changed to "activate" when

a) user gives correct password to select a transfer key,

b) TOE is auto activated for a selected transfer key by autorized user before][35].

FDP_IFF.1.4 The TSF shall explicitly authorise an information flow based on the following rules: **[none][36].**

[33] Assignment: List of subjects and information controlled under the indicated SFP, and for each, the security attributes

[34] Assignment: For each operation, the security attribute-based relationship that must hold between subject and information security attributes

[35] Assignment: Additional information flow control SFP rules

[36] Assignment: Rules, based on security attributes, that explicitly authorise information flows

FDP_IFF.1.5 The TSF shall explicitly deny an information flow based on the following rules: **[If activation state of TOE is changed to "deactivate" by user][37].**

FDP_RIP.1 Subset residual information protection

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resoucre is made unavailable upon the **[deallocaiton of the resource from][38]** the following objects:

[

Transfer Key(s),

User Password

][39].

Application Note: "resource" stands for the Flash memory of AT32UC3A3256S chip and "deallocation" occures upon erasing the chip.

Unavailability of transfer key and user password is provided by encrypting them by storage key.

FDP_SDI.2/USERATTRIBUTES Stored data integrity monitoring and action

FDP_SDI.2.1 The TSF shall monitor user data stored in containers controlled by the TSF for [**CRC Error**][40] on all objects, based on the following attributes: [**user password, transfer key, device label, key label**][41].

FDP_SDI.2.2 Upon detection of a data integrity error, the TSF shall [**restore from back up copy**][42].

FDP_SDI.2/PROGRAMMEMORY Stored data integrity monitoring and action

[37] Assignment: Rules, based on security attributes, that explicitly deny information flows

[38] Selection: allocation of the resource to, deallocation of the resource from

[39] Assignment: List of objects

[40] Assignment: Integrity errors

[41] Assignment: User data attributes

[42] Assignment: Action to be taken

FDP_SDI.2.1 The TSF shall monitor user data stored in containers controlled by the TSF for [**CRC Error**][43] on all objects, based on the following attributes: [**storage key, firmware**][44].

FDP_SDI.2.2 Upon detection of a data integrity error, the TSF shall [**Lockdown TOE**][45].

6.1.3. Identification and Authentication

FIA_AFL.1 Authentication Failure Handling

FIA_AFL.1.1 The TSF shall detect when [[**3**]][46] unsuccessful authentication attempts occur related to [**user operation requests**][47].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been [**met**][48], the TSF shall [**erase transfer key and overwrite with random key, reset user security attributes, device and key labels and go to initial state**][49].

FIA_ATD.1 User Attribute Definition

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: [**user password, transfer key, auto activation attribute**][50].

FIA_UAU.1 Timing of Authentication

FIA_UAU.1.1 The TSF shall allow [**Self Test, Auto-activation,Deactivation, Initialize the TOE after the corruption of user security attributes, Initialize the TOE after retry count number is reached the limit**][51] on behalf of the user to be performed before the user is authenticated.

[43] Assignment: Integrity errors

[44] Assignment: User data attributes

[45] Assignment: Action to be taken

[46] Selection: [assignment: positive integer number], an administrator configurable positive integer within[assignment: range of acceptable values]

[47] Assignment: List of authentication events

[48] Selection: met, surpassed

[49] Assignment: list of actions

[50] Assignment: List of security attributes

[51] Assignment: List of TSF mediated actions

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.

6.1.4. Security Management

FMT_MOF.1 Management of Security Functions Behaviour

FMT_MOF.1.1 The TSF shall restrict the ability to [**disable, enable**][52] the functions [**auto-activation**][53] to [**authorized user**][54].

FMT_MSA.1 Management of security attributes

FMT_MSA.1.1 The TSF shall enforce the [**user security attributes access control policy**][55] to restrict the ability to [**change default**, **modify, delete**][56] the security attributes [**user password, transfer key, device label, transfer key label**][57] to [**authorized user**][58].

FMT_MSA.3/USERDATA Static attribute initialization

FMT_MSA.3.1 The TSF shall enforce the [**user data** flow **control policy**][59] to provide [[**auto activation setting**]][60] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the [**authorized user**][61] to specify alternative initial values to override the default values when an object or information is created.

FMT_MSA.3/USERSECURITYATTRIBUTES Static attribute initialization

[52] Selection: determine the behaviour of, disable, enable, modify the behaviour of

[53] Assignment: List of functions

[54] Assignment: The authorised identified roles

[55] Assignment: Access control SFP(s), information flow control SFP(s)

[56] Selection: change_default, query, modify, delete, [assignment: other operations]

[57] Assignment: List of security attributes

[58] Assignment: The authorised identified roles

[59] Assignment: Access control SFP, information flow control SFP

[60] Selection, choose one of: restrictive, permissive, [assignment: other property]

[61] Assignment: The authorised identified roles

FMT_MSA.3.1 The TSF shall enforce the [**user security attributes access control policy**][62] to provide [**permissive**][63] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the [**authorized user**][64] to specify alternative initial values to override the default values when an object or information is created.

FMT_MTD.1 Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to [**change_default, modify**][65] the [**user security attributes**][66] to [**authorized user**][67].

FMT_SMF.1 Specification of management functions

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: [**modify user security attributes**][68].

FMT_SMR.1 Security roles

FMT_SMR.1.1 The TSF shall maintain the roles [**authorized user**][69].

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

6.1.5. Protection of the TSF

FPT_FLS.1 Failure with preservation of secure state

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: [**Corruption of user security attributes in the Non-Volatile Memory, Corruption of Program Memory**][70].

[62] Assignment: Access control SFP, information flow control SFP

[63] Selection, choose one of: restrictive, permissive, [assignment: other property]

[64] Assignment: The authorised identified roles

[65] Selection: change_default, query, modify, delete, clear, [assignment: other operations]

[66] Assignment: List of TSF data

[67] Assignment: The authorised identified roles

[68] Assignment: List of management functions to be provided by the TSF

[69] Assignment: The authorised identified roles

[70] Assignment: List of types of failures in the TSF

FPT_TST.1 TSF testing

FPT_TST.1.1 The TSF shall run a suite of self tests [**at the conditions[start up]**][71] to demonstrate the correct operation of [**the TSF**][72].

FPT_TST.1.2 The TSF shall provide authorized users with the capability to verify the integrity of [**TSF data**][73].

FPT_TST.1.3 The TSF shall provide authorized users with the capability to verify the integrity of [**TSF**][74].

6.1.6. Resource Utilization

FRU_FLT.2 Limited Fault Tolerance

FRU_FLT.2.1 The TSF shall ensure the operation of **all the TOE's capabilities** when the following failures occur: [**Corruption of one copy of user security attributes in the Non-Volatile Memory**][75].

6.2. Security Assurance Requirements

The Security Assurance Requirements for the TOE are the Evaluation Assurance Level 2. The requirements for this level is listed below;

Table 10 – TOE Security Assurance Requirements

Assurance Class	Assurance Component
ADV: Development	ADV_ARC.1 – Security architecture description
	ADV_FSP.2 – Security enforcing functional specification
	ADV_TDS.1 – Basic design
AGD: Guidance Documents	AGD_OPE.1 – Operational user guidance
	AGD_PRE.1 – Preparative procedures

[71] Selection: during initial start-up, periodically during normal operation, at the request of the authorised user, at the conditions[assignment: conditions under which self test should occur]

[72] Selection: [assignment: parts of TSF], the TSF

[73] Selection: [assignment: parts of TSF data], TSF data

[74] Selection: [assignment: parts of TSF], TSF

[75] Assignment: List of type of failures

Assurance Class	Assurance Component
ALC: Life-cycle Support	ALC_CMC.2 – Use of a CM system
	ALC_CMS.2 – Parts of the TOE CM coverage
	ALC_DEL.1 – Delivery procedures
ASE: Security Target Evaluation	ASE_CCL.1 – Conformance claims
	ASE_ECD.1 - Extended components definition
	ASE_INT.1 – ST Introduction
	ASE_OBJ.2 – Security objectives
	ASE_REQ.2 – Derived security requirements
	ASE_SPD.1 – Security problem definition
	ASE_TSS.1 – TOE summary specification
ATE: Test	ATE_COV.1 – Evidence of coverage
	ATE_FUN.1 – Functional testing
	ATE_IND.2 – Independent testing - sample
AVA: Vulnerability Assessment	AVA_VAN.2 – Vulnerability analysis

6.3. Security Requirements Rationale

The following table shows that all SFRs contribute to at least one objective and all objectives are met at least by one SFR.

Table 11 - Mapping of SFR to Objectives

SFR	O.AUTHENTICATION	O.AUTHATTEMPT	O.CRYPTOGRAPHY	O.USERATTR	O.SELFTEST	O.FIRMWAREPROTECTION
FCS_COP.1			X	X		
FCS_CKM.1			X	X		
FCS_CKM.3	X		X			

SFR	O.AUTHENTICATION	O.AUTHATTEMPT	O.CRYPTOGRAPHY	O.USERATTR	O.SELFTEST	O.FIRMWAREPROTECTION
FCS_CKM.4		X			X	X
FDP_ACC.1/ USERSECURITY ATTRIBUTES	X					
FDP_ACC.1/ CRYPTOKEY ACCESS				X		X
FDP_ACF.1/ USERSECURITY ATTRIBUTES	X					
FDP_ACF.1/ CRYPTOKEY ACCESS				X		X
FDP_IFC.1	X		X			
FDP_IFF.1	X		X			
FDP_RIP.1				X		
FDP_SDI.2/ USERATTRIBUTES					X	
FDP_SDI.2/ PROGRAM MEMORY					X	
FIA_AFL.1		X				
FIA_ATD.1	X					
FIA_UAU.1	X				X	
FMT_MOF.1	X					
FMT_MSA.1	X					

SFR	O.AUTHENTICATION	O.AUTHATTEMPT	O.CRYPTOGRAPHY	O.USERATTR	O.SELFTEST	O.FIRMWAREPROTECTION
FMT_MSA.3/ USERDATA	X		X			
FMT_MSA.3/ USERSECURITY ATTRIBUTES	X					
FMT_MTD.1	X		X			
FMT_SMF.1		X	X	X		
FMT_SMR.1	X					
FPT_FLS.1					X	X
FPT_TST.1					X	
FRU_FLT.2					X	

The following table shows how the SFRs help to maintain the security objectives according to the mappings in the table above and the status of the dependencies.

Table 12 - SFR Dependecies

SFR	DESCRIPTION	DEPENDENCIES
FCS_COP.1	This component performs encryption and decryption operations	FCS_CKM.1 and FCS_CKM.4 are fulfilling the dependencies of the component.
FCS_CKM.1	This component generates the AES keys.	FCS_COP.1 is fulfilling the dependency of the component. However the dependency for FCS_CKM.4 is not met since the storage key is generated during the manufacturing and never been and yet not necessary to change during the lifecycle of the TOE.
FCS_CKM.3	This component activate the keys upon user request after authentication.	FCS_CKM.1 and FCS_CKM.4 are fulfilling the dependencies of the component.
FCS_CKM.4	This component performs destruction for cryptographic keys.	FCS_CKM.1 is fulfilling the dependencies of the component.

SFR	DESCRIPTION	DEPENDENCIES
FDP_ACC.1/USERSECURITY ATTRIBUTES	This component provides access control policy for user security attributes in the TOE.	FDP_ACF.1/USERSECURITY ATTRIBUTES is fulfilling the dependencies of the component.
FDP_ACC.1/CRYPTOKEY ACCESS	This component enforce an access control policy in order to deny each access attempts to read the cryptographic keys.	FDP_ACF.1/CRYPTOKEYACCESS is fulfilling the dependencies of the component.
FDP_ACF.1/USERSECURITY ATTRIBUTES	This component provides access to user security attributes according to the related policy.	FDP_ACC.1/USERSECURITY ATTRIBUTES and FMT_MSA.3/USERSECURITY ATTRIBUTES are fulfilling the dependencies of the component.
FDP_ACF.1/CRYPTOKEY ACCESS	This component prevents access to cryptographic keys according to the related policy.	FDP_ACC.1/CRYPTOKEYACCESS is fulfilling the dependency of the component. However the dependency for FMT_MSA.3 is not provided since the policy is not allowing any access to any users for any type of default security attribute values. That's why the dependency is not fulfilled.
FDP_IFC.1	This component enforce an flow control policy in order to manage flow of user data.	FDP_IFF.1 is fulfilling the dependencies of the component.
FDP_IFF.1	This component manages flow of user data according to the related policy.	FDP_IFC.1 and FMT_MSA.3/USERDATA are fulfilling the dependencies of the component
FDP_RIP.1	This component provides residual information protection	No Dependencies.
FDP_SDI.2/USERATTRIBUTES	This component provides the integrity of stored user security attributes. .	No Dependencies.
FDP_SDI.2/PROGRAMMEMORY	This component provides the integrity of storage keys in the program memory.	No Dependencies.
FIA_AFL.1	This component is handling the failed authentication attempts and take the defined action when number of failed attempts reach "3"	FIA_UAU.1 is fulfilling the dependencies of the component.
FIA_ATD.1	This component is defining the security attributes of authorized user.	No Dependencies.
FIA_UAU.1	This component is defining the actions before user authentication and other actions which enforce user authentication.	FIA_UID is not provided by the TOE since the each TOE is assign to a specific user and the owner of the TOE is assumed to be authenticated upon providing the user password. No further identification is required. That's why the dependency is not fulfilled.
FMT_MOF.1	This component is defining the security functions and the abilities of authorized user.	FMT_SMR.1 and FMT_SMF.1 are fulfilling the dependencies of the component.

SFR	DESCRIPTION	DEPENDENCIES
FMT_MSA.1	This component is defining the security attributes and the abilities of authorized user.	FDP_IFC.1, FDP_ACC.1/USERSECURITY ATTRIBUTES, FMT_SMR.1 and FMT_SMF.1 are fulfilling the dependencies of the component.
FMT_MSA.3/USERDATA	This component is defining the default values of security attributes on user data.	FMT_MSA.1 and FMT_SMR.1 are fulfilling the dependencies of the component.
FMT_MSA.3/USERSECURITY ATTRIBUTES	This component is defining the default values of user security attributes.	FMT_MSA.1 and FMT_SMR.1 are fulfilling the dependencies of the component.
FMT_MTD.1	This component is defining the TSF data managed by authorized user.	FMT_SMR.1 and FMT_SMF.1 are fulfilling the dependencies of the component.
FMT_SMF.1	This component is defining management functions performed by TSF.	No Dependencies.
FMT_SMR.1	This component is defining the security roles in TSF.	The dependency for FIA_UID.1 is not fulfilled since the TOE is only providing authentication.
FPT_FLS.1	This component is preserving TOE in the secure state when defined failure types occur.	No Dependencies.
FPT_TST.1	This component is conducting TSF testing in the given conditions.	No Dependencies.
FRU_FLT.2	This component defines the scope of fault tolerance when the specified failures occur.	FPT_FLS.1 is fulfilling the dependencies of the component.

6.4. Security Assurance Requirements Rationale

The overall security claim of this Security Target is aimed at EAL2.

EAL2 is accepted as the suitable assurance level where TOE can be conformant. While the TOE is assigned to each user and used by their responsibility according to the assumptions made above, it is suitable to claim basic attack potential.

If the opposite is not explicitly stated and justified all the dependencies and requirements of the selected assurance level are satisfied during the life cycle of the TOE.

7. TOE SUMMARY SPECIFICATIONS

7.1. User Request Management

TOE performs the following security requirements through the user request management function;

User request management function performs user authentication in order to perform access control and accounting. Since each TOE is assigned to a specific user instead of requesting a User ID, TOE only request user password for authentication.

This function also controls the number of failed authentication attempts by incrementing a retry number each time when a user provide an incorrect password. When the retry number reach to “3”, this function invokes main system control function(FIA_AFL.1, FIA_UAU.1).

This function controls and enforce two of the access control policies within the TOE;

- User Security Attributes Access Control Policy: Defines the rules, operations and subjects for accessing the user security attributes(FDP_ACC.1/USERSECURITYATTRIBUTES). These attributes are stored in the non-volatile memory as encrypted by storage key (FDP_ACF.1/USERSECURITYATTRIBUTES).
- Cryptographic Key Access Policy: TOE prevents access to the Cryptographic Keys according to this policy(FDP_ACC.1/CRYPTOKEYACCESS). Storage key is stored in the program memory with firmware. Program memory is protected by security bit function which is provided by IC manufacturer. When security bit is set, any access to program memory is permitted, unless erasing the whole program memory(FDP_ACF.1/CRYPTOKEYACCESS).

User request management function generate cryptographic keys in certain circumstances. When user request to generate random key, user this function generate the cryptographic key(FCS_CKM.1). On the other hand users can generate transfer key(s) through user request management function. All of the user configurable management functions defined in this ST within the TOE are performed by user request management function. TOE enforce authentication and access control in order to supply the resources to the user and perform the management operations.

Authenticated user can change password, transfer key(s) and auto activation value and also can give label to TOE and transfer key(s) (FMT_MOF.1,FMT_MSA.1,FMT_MTD.1, FMT_MSA.3/USERDATA). There is no login state for TOE. Every request of user must contain valid user password to confirm authenticity of user.

User must specify its password firstly, if TOE is in initial state. When TOE is in initial state, TOE does not allow any other command except changing password (i.e. setting) and does not require password match(FMT_MSA.3/USERSECURITYATTRIBUTES).

There is only one kind of user.That's why, there is just user authentication and no need for user identification. One who knows the correct password is interpreted as valid user by TOE (FMT_SMR.1).

7.2. Main System Control

Main system control function is one of the essential functions provided by TOE. As well as encrypting/decrypting the user security attributes(FCS_COP.1), main system control function is generating random key and transfer key(s) during regular operations and also is generating the storage key during the initialization(FCS_CKM.1). AES module is loaded with selected transfer key during activation of TOE and zerorized during deactivation. (FCS_CKM.3, FCS_CKM.4).

At the first run of firmware, TOE generates a storage key randomly and stores it into program memory (FCS_CKM.1). Then, calculates and stores CRC of program memory for integrity checking at next start-ups(FDP_SDI.2/PROGRAMMEMORY)

When user request management function invoke main system control function after three failed authentication attempts, this function will overwrite with a random key and reset the passwords and then return back to fabric defaults(FIA_AFL.1, FIA_UAU.1).

TOE preserve its secure state if a corruption occurs in non-volatile memory or program memory(FPT_FLS.1). User security attributes in the non-volatile memory is stored with a back up copy. If a corruption occur in one of the copies,TOE still performs all the TOE capabilities(FRU_FLT.2). Also TOE conducts self tests during start up in order to demonstrate the correct operation of the TOE. Integrity checks are performed at this stage and actions are taken according to the results of self tests(FPT_TST.1, FIA_UAU.1).

On the other hand TOE provides integrity of stored data. CRC checks are performed during the start up for security attributes of the user and for program memory(FDP_SDI.2). TOE either restore from the backup copy or lockdown TOE when integrity check is failed.

7.3. Bridge Control

Bridge control function is enforcing the 'user data flow control policy', which performs flow control during the data transfer to/from the back disk(FDP_IFC.1). With this policy, TOE defines the rules, operations and subjects for flow of the user data to/from the back disk(FDP_IFF.1). Secure communication bridge between host system and back disk will establish after the request of activation by authorized user with a selected transfer key(FCS_CKM.3). If TOE is configured as auto-activated, activation request is performed by TOE on behalf of user(FIA_UAU.1, FMT_MSA.3/USERDATA).

7.4. Secure Bridge

When the requirements for data transfer to the back disk established, secure bridge function provides the functionality to encrypt/decrypt of data to/from the back disk(FDP_AFC.1, FDP_AFF.1). Secure bridge function require bridge control function to establish the connection and then the transfer key(s) are imported to the AES module in order to encrypt/decrypt user data.(FCS_CKM.3)

7.5. Security Attributes

Security attributes function is managing the following attributes belong to the user within the TOE;

- user password,
- transfer key,
- auto activation attribute(FIA_ATD.1).

These attributes are protected with AES encryption in the program memory and decrypted while they are in use(FCS_COP.1).

Two copies of security attributes are stored. These copies also have CRC value for integrity checking of content when recalling (FDP_SDI.2/USERATTRIBUTES).

Storage key and retry number are the security attributes of TOE. Even to the authorized user , TOE does not provide any interface to change them. (FMT_SMF.1).

The firmware and the CPU on which the firmware runs are also the security attributes of TOE. TOE blocks any access to them.

7.6. Program Memory Protection

Program memory protection function is protecting the program memory with the support of a security bit. With the support of this function it is not possible to reach either to the storage key or the firmware. When anyone will try to access the program memory, TOE first erases all data including the firmware and storage key and then allows access. After that stage only the encrypted user security attributes can be accessible. Since the user security attributes are encrypted with the deleted storage key, it is assumed that user data is protecting its confidentiality even after probing or any defined physical attacks to the TOE(FDP_RIP.1).