W75F40WBYJEG

Secure Flash Memory

Security Target

Lite version

Contents

Table of Figures

Table of Tables

1 Security Target Introduction

This Security Target is based on the Security IC Platform Protection Profile with Augmentation Packages [5]. However, the Security Target does not include the Random Generation and the IC Identification security objectives. The corresponding assumptions of the Protection Profile are not used and replaced by other assumptions.

On the other hand, the Security Target includes additional elements which are not required by the Protection Profile [5]. Those security elements (threats, security objectives, SFR) are clearly identified in each Chapter of this document.

1.1 Security Target Reference

Title: W75F40WBYJEG Secure Flash Memory Security Target Lite Version

Version: E1

Authors: Winbond Technology Ltd.

Evaluator: Applus

Certified by: CCN Organismo de Certificacion

1.2 TOE Reference

The Target of Evaluation is identified below:

Commercial Name	SpiFlash® TrustME™ Secure Flash Memory
Product Name	W75F40WBYJEG
Version	A
Guidance	Refer to table 2

Table 1 TOE Identification

1.3 TOE Overview

1.3.1 TOE Type

The Target of Evaluation is a Memory Flash IC.

1.3.2 TOE Intended Usage

The TOE is dedicated to be embedded into highly critical hardware devices such as smart card, secure element, USB token, secure micro SD, etc. These devices will embed secure applications such as financial, telecommunication, identity (e-Government), etc and will be working in a hostile environment. In particular, the TOE is dedicated to the secure storage of the code and data of critical applications.

The security needs for the TOE consist in:

- Maintaining the integrity of the content of the memories and the confidentiality of the content of protected memory areas as required by the critical HW products (e.g. Security IC) the Memory Flash is built for;
- Providing a secure communication with the Host device that will embed the TOE in a secure HW product such as Security IC;

1.3.3 Non-TOE Hardware/Software/Firmware

For the present ST, the TOE is a pure storage hardware device.

The TOE does not comprise:

a) The Host device that will embed the TOE and will be needed to run the TOE in order to stimulate the TSF
b) SPI Bus for the communication between the Host device and the TOE
c) SFI Library enables communication with the Winbond TrustME™ Secure Flash Device over a regular SPI communication channel. The library allows a communication channel with the flash, and perform secure READ/WRITE/ERASE operations by using a typical SPI master interface. The Library must be used by the the Host developer

The ST assumes that all components (Hardware or Software) of the Host Device are appropriately protected in the TOE security environment.

1.4 TOE Description

1.4.1 Physical Scope

The TOE comprises:

- All security functionality necessary to ensure the secure execution of the Memory Flash:

No	Type	Identifier	Version	Delivery Method
Form of delivery : Packaged Device				
1	HW	IC Part number	W75F40WBYJEG	Via Courier
Form of delivery : Associated IC Dedicated Documentation				
1	PDF	W75F40WBYJEG Preparative User Guide	Version I	Encrypted mail
2	PDF	W75F40WBYJEG Operational User Guide	Version F	Encrypted mail
3	PDF	W75F40 Secure Flash Datasheet	Version A	Mail
4	PDF	SFI Library User Guide	Version E	Encrypted mail

Table 2 TOE Physical Scope

1.4.1.1 TOE Physical Characteristics

The TOE physical characteristics are described as follows.

Performance

50MHz Standard/ /Dual SPI clocks

20.5 MB/S continuous encrypted and authenticated data transfer rate

More than 100,000 erase/program cycles

More than 20-year data retention

Efficiency

16-byte burst read

Data Integrity Check

Allows secure execution in place (S-XIP) operation

Operating conditions

- Single 1.55 to 2.00V supply
- 20mA active current, <1µA Power-down (typ.)
- -40°C to +105°C operating range

1Mb-block Architecture

Uniform Block Erase (4K-bytes)

Program 1 to 16 byte in a single command

Erase/Program Suspend & Resume

1.4.1.2 TOE Architecture

The architecture of the Memory Flash is described in **Figure 1**. The TOE is delimited by the Red box.

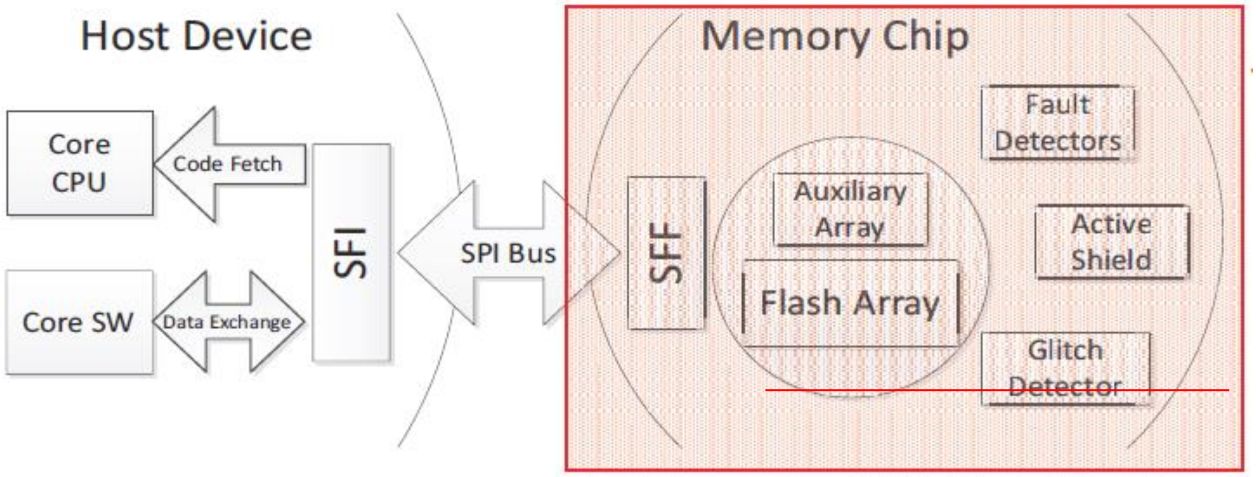

Figure 1 TOE Architecture

The TOE consists of the following Hardware components

- Auxiliary array contains the flash specific data: the binding key (and its digest value), the failure and session counters;
- Flash array stores the User data (i.e. the mass data including executable codes) and translates SPI commands into Flash operations;
- SFF (Secure Flash Front-end) which implements encrypted and authenticated interface for Flash operation and supports Flash memories up to 4GB;
- Detectors of abnormal operating conditions;

1.4.1.3 Interfaces of the TOE

- The physical interface of the TOE with the external environment is the entire surface of the Memory Flash module.
- The electrical interface of the TOE with the external environment is made of the chip's pads including the data pins for SPI bus:
  - Standard SPI: CLK, /CS, DI_IO0, DO_IO1
  - Dual SPI: CLK, /CS, DI_IO0, DO_IO1

1.4.2 Logical Scope

The main security features of the TOE are described as follows:

- Secure separation between Test mode and User mode. More precisely,
  - The switch from User mode to Test mode can only be done after completely erasing the flash content.
  - The confidentiality and the integrity of the flash content are protected in both Test mode and User mode.
- The confidentiality and the integrity of the transmitted data from/to the Host device are protected by a secure channel;
- Integrity protection of the flash content by error detection codes (CRC-32);
- Confidentiality protection of the flash content by memory scrambling with diversified key;
- Security sensors or detectors including power glitch detector and out-of-specified operating conditions (voltage, temperature, clock frequency);
- Active Shields against physical intrusive attacks (e.g. reverse-engineering, probing);
- State machine protection to counter fault injection;
- Dual Flip-Flops and bus encoding to counter fault injection and information leakage;
- Failure counter to detect and react to tamper attempts;

The logical interface of the TOE is made of Flash commands.

1.5 TOE Configurations

Part Number	Density	Binding Method	Note
W75F40WBYJEG	4 Mbit	Two-Phase	Support secure binding to be completed in non-secure environment

Table 3 TOE Configurations

For guidance on the usage of the TOE, see Table 2.

1.6 TOE Life Cycle

The development, manufacturing and integration processes of the TOE into a composite product can be separated into two distinct phases.

Phase	Title	Description
1	TOE Development	Memory flash designer is responsible for: - TOE (HW) development
2	TOE Manufacturing and Testing	Memory flash Manufacturer is responsible for: - Photomask manufacturing - wafer manufacturing and - testing
3	TOE Packaging and Final Testing	Memory flash packaging Memory flash final test

Table 5 TOE life-cycle

The TOE is delivered as in after the phase 3 in WLCSP packaging form.

The TOE user is responsible for developing the Host-based dedicated driver and for generating a random and unique binding key (Kb) for binding the TOE to a unique Host.

2 Conformance Claim

2.1 CC Conformance Claim

This Security target claims to be conformant to the Common Criteria version 3.1 Release 5.

Furthermore it claims to be CC Part 2 extended and CC Part 3 conformant.

2.2 PP Claim

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

2.3 Package Claim

The assurance level for this Security Target is EAL5 augmented with ALC_DVS.2 and AVA_VAN.5 because the TOE is dedicated to store highly critical applications and data which are submitted to advanced logical and physical attacks.

3 Security Problem Definition

3.1 Assets

Assets include all data stored in the TOE (including executable code of the applications). They include:

- User data, that is typically stored in the "flash array" part of the memory chip;
- TSF data, that is relied upon for the enforcement of the TOE security functionality.
  - TSF data contains only sensitive data stored in registers or in the auxiliary array of the memory chip. Legacy registers are not part of the TSF (i.e. non-TSF).
  - The TOE does not include any software, however the logic of the TOE security mechanisms is still part of the TSF data. This logic is hardcoded in SFF.

3.1.1 TSF data

TSF logic

The TSF logic is the functionality of the TSF, and is hardcoded in the SFF component.

The TSF logic is protected in terms of integrity and confidentiality.

Binding key (Kb)

A unique 256-bit key that is shared between the TOE and the Host.

This key is protected in terms of integrity and confidentiality.

Runtime data

The internal runtime data necessary for the execution of the SFF: session key, memory scrambling keys, Integrity Checking Engine register, stream-ciphering buffer, Bit mixing key, Failure counter, session counter, etc. All runtime data shall be protected in terms of integrity. All runtime data (except for the session counter) shall be protected in terms of confidentiality.

3.1.2 User data

User data corresponds to all data stored inside the memory flash (including executable code of the applications).

User Data

Mass data (including executable codes) stored in the "flash array" part of the memory chip. User data is protected in terms of integrity and confidentiality.

3.2 Users / Subjects

U.Host-Device

The host device communicates with the TOE through a SPI Bus.

3.3 Threats

T.Phys-Manipulation

Physical Manipulation

An attacker may physically modify the Memory Flash in order to

- modify *User Data* stored in the TOE;
- modify *TSF Data* stored in the TOE;
- modify or deactivate the security services of the TOE (provided by *TSF logic*);
- modify the security mechanisms of the TOE (provided by *TSF logic*) to enable attacks disclosing or manipulating *User Data*, for example the integrity protection mechanism.

T.Phys-Probing

Physical Probing

An attacker may perform physical probing of the TOE in order to disclose *User Data* and *TSF Data* while stored in Memory Flash.

T.Malfunction

Malfunction due to Environmental Stress

An attacker may cause a malfunction of *TSF logic* by applying environmental stress in order to deactivate or affect security mechanisms of the TOE. This enables attacks disclosing or manipulating *User Data*.

This may be achieved by operating the Memory Flash outside the normal operating conditions.

T.Abuse-Func

Abuse of Functionality

An attacker may use functions of the TOE which may not be used after TOE Delivery in order to

- disclose or manipulate *User Data* (user data or code stored in the TOE) or
- enable an attack disclosing or manipulating *User Data*.

T.Leak-Inherent

Inherent Information Leakage

An attacker may exploit information which is leaked from the TOE during usage of the Memory Flash in order to disclose confidential *User Data*.

T.Leak-Forced

Forced Information Leakage

An attacker may exploit information which is leaked from the TOE during usage of the Memory Flash in order to disclose confidential *User Data* even if the information leakage is not inherent but caused by the attacker.

T.Abuse-Communication

Communication Probing and Manipulation

An attacker may probe and modify the communication between the TOE and **U.Host-Device** in order to manipulate *User/TSF Data* or disclose *User/TSF Data* read from the TOE.

T.Host-Forging

Forge the functionality of an authorized Host device

An attacker may access to the User data currently stored in the TOE by:

- illegaly establishing a secure channel with the TOE (e.g. by tampering the Binding key or by forging the secure channel without knowing the Binding key) in order to execute the Flash commands;
- binding the TOE with another Host device in order to execute the Flash commands;

3.4 Organisational Security Policies

N/A, there is no OSP.

3.5 Assumptions

A.Secure-Channel

External protection during the secure channel

It is assumed that **U.Host-Device** supports the trusted communication channel with the TOE by protecting the confidentiality and the integrity of the transmitted data.

In particular, **U.Host-Device** is assumed to correctly protect the secure channel in order to prevent data modification, disclosure, insertion, deletion and replaying.

A.Binding-Process

Protection during Binding process

It is assumed that security procedures are used after delivery of the TOE by the TOE Manufacturer to maintain confidentiality and integrity of the TOE (to prevent any possible copy, modification, or unauthorised use).

This means that the binding process (i.e. generating a unique and random key Kb for **U.Host-Device** and the TOE) or the first stage of the two-stage binding, is assumed to be done in a secure environment where the communication between **U.Host-Device** and the TOE is protected.

Furthermore, **U.Host-Device** is assumed to provide a secure random source for generating a fresh Binding key (Kb) for the TOE.

The confidentiality and authenticity of the binding process is guarentied by uniqe pre-binding process during TOE manufacturing.

A.Host_SFI-Lib

SFI-Lib correct implementation

The TOE is operated by the Host device through the SFI-Lib. The SFI-Lib is developed by the TOE Manufacturer and delivered to the Host developer. The SFI_Lib is well-developed by properly parsing high-level API commands into the appropriate sequence of TOE commands of its functional specification.

4 Security Objectives

4.1 Security Objectives for the TOE

O.Phys-Probing

Protection against Physical Probing

The TOE must provide protection against disclosure/reconstruction of *User Data* and *TSF Data* while stored in the Flash.

This includes protection against

- measuring through galvanic contacts which is direct physical probing on the chips surface except on pads being bonded (using standard tools for measuring voltage and current) or
- measuring not using galvanic contacts but other types of physical interaction between charges (using tools used in solid-state physics research and IC failure analysis) with a prior reverse-engineering to understand the design and its properties and functions.

The TOE must be designed and fabricated so that it requires a high combination of complex equipment, knowledge, skill, and time to be able to derive detailed design information or other information which could be used to compromise security through such a physical attack.

O.Malfunction

Protection against Malfunctions

The TOE must ensure its correct operation. The TOE must indicate and prevent its operation outside the normal operating conditions where reliability and secure operation has not been proven or tested. This is to prevent malfunctions. Examples of environmental conditions are voltage, and clock frequency, temperature, or external energy fields.

O.Phys-Manipulation

Protection against Physical Manipulation

The TOE must provide protection against manipulation of *User Data* (the user data stored in the TOE) and *TSF data*. This includes protection against

- reverse-engineering (understanding the design and its properties and functions),
- manipulation of the hardware and TSF data, as well as
- undetected manipulation of User data (i.e. Flash array).

O.Abuse-Func

Protection against Abuse of Functionality

The TOE must prevent that functions of the TOE which may not be used after TOE Delivery can be abused in order to (i) disclose sensitive user data stored in the TOE, (ii) manipulate sensitive user data stored in the TOE.

O.Leak-Inherent

Protection against Inherent Information Leakage

The TOE must provide protection against disclosure of confidential data stored and processed in the TOE

- by measurement and analysis of the shape and amplitude of signals (for example on the power, clock, or I/O lines) and
- by measurement and analysis of the time between events found by measuring signals (for instance on the power, clock, or I/O lines).

O.Leak-Forced

Protection against Forced Information Leakage

The TOE must be protected against disclosure of confidential data processed in the TOE (using methods as described under O.Leak-Inherent) even if the information leakage is not inherent but caused by the attacker

- by forcing a malfunction (refer to "Protection against Malfunction due to Environmental Stress O.Malfunction") and/or
- by a physical manipulation (refer to "Protection against Physical Manipulation - O.Phys-Manipulation").

If this is not the case, signals which normally do not contain significant information about secrets could become an information channel for a leakage attack.

O.Sec-Binding

Protection of residual information at Re-binding

This objective protects against the disclosure of the User data when the TOE is re-bound to another Host device.

This includes protection against:

- integrity failure on Binding Key
- illegal modification on Binding Key
- illegal attempt to erase the Binding key

O.Trusted-Path

Trusted communication with authorized Host

The TSF provides a trusted path only with authorized **U.Host-Device** (based on the shared Binding key), and protects the confidentiality and the integrity of the User data /TSF data to be communicated with **U.Host-Device**.

4.2 Security Objectives for the Operational Environment

OE.Secure-Channel

Secure communication with the TOE

The authorized **U.Host-Device** shall support the trusted communication channel with the TOE by protecting the confidentiality and the integrity of the transmitted data.

In particular, **U.Host-Device** shall correctly protect the secure channel in order to prevent data modification, disclosure, insertion, deletion and replaying.

OE.Binding-Process

Protection during Binding process

Security procedures shall be used after the TOE delivery to maintain confidentiality and integrity of the TOE (to prevent any possible copy, modification, retention, theft or unauthorised use).

In addition, **U.Host-Device** shall provide a secure random source for generating a fresh Binding key (Kb) for the TOE.

OE.Host_SFI-Lib

SFI-Lib correct implementation

The TOE Manufacturer must develop according to the TOE functional specification for the secure operation of the TOE. The Host must operate the TOE through the SFI_Lib provided by the TOE Manufacturer.

4.3 Security Objectives Rationale

4.3.1 Threats

T.Phys-Manipulation This threat is countered by the security objectives O.Phys-Manipulation. This objective ensures that the protection against manipulation of the user data is provided by the TOE.

T.Phys-Probing This threat is countered by the security objectives O.Phys-Probing. This objective ensures that the protection against disclosure/reconstruction of User Data and TSF Data while stored in the Flash is provided by the TOE.

T.Malfunction This threat is countered by the security objectives O.Malfunction. This objective ensures the correct operation of the TOE outside the normal operating conditions.

T.Abuse-Func This threat is countered by the security objectives O.Abuse-Func. This objective prevents that functions of the TOE which may not be used after TOE Delivery can be abused in order to manipulate/disclose sensitive user data stored in the TOE.

T.Leak-Inherent This threat is countered by the security objectives O.Leak-Inherent. This objective ensures the protection against disclosure of confidential data stored and processed in the TOE.

T.Leak-Forced This threat is countered by the security objectives O.Leak-Forced. This objective ensures the protection against disclosure of confidential data stored and processed in the TOE even if the information leakage is not inherent but caused by the attacker.

T.Abuse-Communication This threat is countered by the security objective O.Trusted-Path. This objective protects the confidentiality and the integrity of the User/TSF data to be communicated with U.Host-Device.

T.Host-Forging This threat is countered by the security objectives:

- O.Trusted-Path to protect the confidentiality and the integrity of the User data to be communicated with U.Host-Device.
- O.Sec-Binding to protect against the disclosure of the User data when the TOE is re-bound to another Host device

4.3.2 Assumptions

A.Secure-Channel Since OE.Secure-Channel requires the Host device to implement the protection assumed in A.Secure-Channel, the assumption is covered by this objective.

A.Binding-Process Since OE.Binding-Process requires the Composite Product Manufacturer to implement those measures assumed in A.Binding-Process, the assumption is covered by this objective.

A.Host_SFI-Lib Since OE.Host_SFI-Lib is described in a way that directly meets the described assumption, the assumption A.Host_SFI-Lib is covered by this objective.

4.3.3 SPD and Security Objectives

Threats	Security Objectives	Rationale
T.Phys-Manipulation	O.Phys-Manipulation	Section 4.3.1
T.Phys-Probing	O.Phys-Probing	Section 4.3.1
T.Malfunction	O.Malfunction	Section 4.3.1
T.Abuse-Func	O.Abuse-Func	Section 4.3.1
T.Leak-Inherent	O.Leak-Inherent	Section 4.3.1
T.Leak-Forced	O.Leak-Forced	Section 4.3.1
T.Abuse-Communication	O.Trusted-Path	Section 4.3.1
T.Host-Forging	O.Trusted-Path, O.Sec-Binding	Section 4.3.1

Table 6 Threats and Security Objectives - Coverage

Security Objectives	Threats
O.Phys-Probing	T.Phys-Probing
O.Malfunction	T.Malfunction
O.Phys-Manipulation	T.Phys-Manipulation

Security Objectives	Threats
O.Abuse-Func	T.Abuse-Func
O.Leak-Inherent	T.Leak-Inherent
O.Leak-Forced	T.Leak-Forced
O.Sec-Binding	T.Host-Forging
O.Trusted-Path	T.Abuse-Communication, T.Host-Forging
OE.Secure-Channel	
OE.Binding-Process	

Table 7 Security Objectives and Threats - Coverage

Security Objectives
O.Phys-Probing
O.Malfunction
O.Phys-Manipulation
O.Abuse-Func
O.Leak-Inherent
O.Leak-Forced
O.Sec-Binding
O.Trusted-Path
OE.Secure-Channel
OE.Binding-Process

Table 8 Security Objectives and OSPs - Coverage

Assumptions	Security Objectives for the Operational Environment	Rationale
A.Secure-Channel	OE.Secure-Channel	Section 4.3.2
A.Binding-Process	OE.Binding-Process	Section 4.3.2

Table 9 Assumptions and Security Objectives for the Operational Environment - Coverage

Security Objectives for the Operational Environment	Assumptions
OE.Secure-Channel	A.Secure-Channel

Security Objectives for the Operational Environment	Assumptions
OE.Binding-Process	A.Binding-Process

Table 10 Security Objectives for the Operational Environment and Assumptions- Coverage

5 Extended Requirements

5.1 Extended Families

5.1.1 Extended Family FMT_LIM - Limited capabilities and availability

5.1.1.1 Description

To define the IT security functional requirements of the TOE an additional family (FMT_LIM) of the Class FMT (Security Management) is defined here. This family describes the functional requirements for the Test Features of the TOE. The new functional requirements were defined in the class FMT because this class addresses the management of functions of the TSF. The examples of the technical mechanism used in the TOE (refer to Section 6.2) appropriate to address the specific issues of preventing the abuse of functions by limiting the capabilities of the functions and by limiting their availability.

The family "Limited capabilities and availability (FMT_LIM)" is specified as follows.

FMT_LIM Limited capabilities and availability

Family behaviour

This family defines requirements that limit the capabilities and availability of functions in a combined manner. Note that FDP_ACF restricts the access to functions whereas the component Limited Capability of this family requires the functions themselves to be designed in a specific manner.

Component levelling:

FMT_LIM.1 Limited capabilities requires that the TSF is built to provide only the capabilities (perform action, gather information) necessary for its genuine purpose.

FMT_LIM.2 Limited availability requires that the TSF restrict the use of functions (refer to Limited capabilities (FMT_LIM.1)). This can be achieved, for instance, by removing or by disabling functions in a specific phase of the TOE's life-cycle.

Management: FMT_LIM.1, FMT_LIM.2

There are no management activities foreseen.

Audit: FMT_LIM.1, FMT_LIM.2

There are no actions defined to be auditable.

5.1.1.2 Extended Components

Extended Component FMT_LIM.1

Description

Limited capabilities requires that the TSF is built to provide only the capabilities (perform action, gather information) necessary for its genuine purpose.

Hierarchical to: No other components.

Definition

FMT_LIM.1 Limited capabilities

FMT_LIM.1.1 The TSF shall be designed and implemented in a manner that limits its capabilities so that in conjunction with "Limited availability (FMT_LIM.2)" the following policy is enforced [assignment: Limited capability policy].

Dependencies: (FMT_LIM.2)

Extended Component FMT_LIM.2

Description

Limited availability requires that the TSF restrict the use of functions (refer to Limited capabilities (FMT_LIM.1)). This can be achieved, for instance, by removing or by disabling functions in a specific phase of the TOE's life-cycle.

Hierarchical to: No other components.

Definition

FMT_LIM.2 Limited availability

FMT_LIM.2.1 The TSF shall be designed in a manner that limits its availability so that in conjunction with "Limited capabilities (FMT_LIM.1)" the following policy is enforced [assignment: Limited availability policy].

Dependencies: (FMT_LIM.1)

Application Note:

The functional requirements FMT_LIM.1 and FMT_LIM.2 assume that there are two types of mechanisms (limitation of capabilities and limitation of availability) which together shall provide protection in order to enforce the same policy or two mutual supportive policies related to the same functionality. This allows e.g. that

(i) the TSF is provided without restrictions in the product in its user environment but its capabilities are so limited that the policy is enforced or conversely

(ii) the TSF is designed with high functionality but is removed or disabled in the product in its user environment.

5.1.2 Extended Family FDP_SDC - Stored data confidentiality

5.1.2.1 Description

To define the security functional requirements of the TOE an additional family (FDP_SDC.1) of the Class FDP (User data protection) is defined here.

The family "Stored data confidentiality (FDP_SDC)" is specified as follows.

FDP_SDC Stored data confidentiality

Family behaviour

This family provides requirements that address protection of user data confidentiality while these data are stored within memory areas protected by the TSF. The TSF provides access to the data in the memory through the specified interfaces only and prevents compromise of their information bypassing these interfaces. It complements the family Stored data integrity (FDP_SDI) which protects the user data from integrity errors while being stored in the memory.

Component levelling:

FDP_SDC.1 Requires the TOE to protect the confidentiality of information of the user data in specified memory areas.

Management: FDP_SDC.1

There are no management activities foreseen.

Audit: FDP_SDC.1

There are no actions defined to be auditable.

5.1.2.2 Extended Components

Extended Component FDP_SDC.1

Description

Requires the TOE to protect the confidentiality of information of the user data in specified memory areas.

Hierarchical to: No other components.

Definition

FDP_SDC.1 Stored data confidentiality

FDP_SDC.1.1 The TSF shall ensure the confidentiality of the information of the user data while it is stored in the **[assignment: memory areas]**.

Dependencies: No dependencies.

6 Security Requirements

6.1 Security Functional Requirments Rational

In order to define the Security Functional Requirements Part 2 of the Common Criteria was used. However, some Security Functional Requirements have been refined.

The refinements are described below the associated SFR:

The refinement operation is used to add detail to a requirement, and, thus, further restricts a requirement. In such a case a extra paragraph starting with "Refinement" may be given.

The selection operation is used to select one or more options provided by the CC in stating a requirement. Selections having been made by the ST author are denoted as bold and italicized.

The assignment operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Assignments having been made by the ST author appear in bold text. The iteration operation is used when a component is repeated with varying operations. Iteration is denoted by showing a slash "/", and the iteration indicator after the component identifier.

6.2 Security Functional Requirements

6.2.1 Malfunctions

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: [assignment: *list of type of failures*].

The TSF shall ensure the operation of all the TOE's capabilities when the following failures occur: **exposure to operating conditions which are not detected according to the requirement Failure with preservation of secure state (FPT_FLS.1/Detectors)**.

Application Note:

The term "failure" above means "circumstances". The TOE prevents failures for the "circumstance" defined above.

FPT_FLS.1/Detectors Failure with preservation of secure state

FPT_FLS.1.1/Detectors The TSF shall preserve a secure state when the following types of failures occur: [assignment**:** *list of types of failures in the TSF*].

The TSF shall preserve a secure state when the following types of failures occur:

- o **Out-of-specified range voltage**
- o **Out-of-specified range temperature**
- o **Out-of specified range clock frequency**
- o **Power glitch.**

Application Note:

The term "failure" above means "circumstances". The TOE prevents failures for the "circumstance" defined above.

The secure state is maintained by TSF's detectors. The TSF's detectors monitor the failures. If a failure happens, the TSF disturbs the cryptographic computations, interrupts data interchange and inform **U.Host-Device**.

6.2.2 Abuse of Functionality

FMT_LIM.1 Limited capabilities

FMT_LIM.1.1 The TSF shall be designed and implemented in a manner that limits its capabilities so that in conjunction with "Limited availability (FMT_LIM.2)" the following policy is enforced [assignment**:** *Limited capability policy*]**.**

The TSF shall be designed and implemented in a manner that limits its capabilities so that in conjunction with "Limited availability (FMT_LIM.2)" the following policy is enforced **Deploying Test Features after TOE Delivery does not allow user data to be disclosed or manipulated, TSF data to be disclosed or manipulated, and no substantial information about construction of TSF to be gathered which may enable other attacks.**

Application Note:

In the Test mode, the following restrictions are enforced by the TSF:

- The Binding Key (Kb) cannot be read out by the Flash commands;
- The Binding key cannot be erased unless a complete erase has been done after the last reset;
- The read and write commands do not read and write effective values of the flash array;

FMT_LIM.2 Limited availability

FMT_LIM.2.1 The TSF shall be designed in a manner that limits its availability so that in conjunction with "Limited capabilities (FMT_LIM.1)" the following policy is enforced [assignment: *Limited availability policy*]**.**

The TSF shall be designed in a manner that limits its availability so that in conjunction with "Limited capabilities (FMT_LIM.1)" the following policy is enforced **Deploying Test Features after TOE Delivery does not allow user data to be disclosed or manipulated, TSF data to be disclosed or manipulated, and no substantial information about construction of TSF to be gathered which may enable other attacks.**

Application Note:

The switch from User mode to Test mode is allowed after TOE delivery but after the flash array is completely erased.

6.2.3 Physical Manipulation and Probing

FDP_SDC.1 Stored data confidentiality

FDP_SDC.1.1 The TSF shall ensure the confidentiality of the information of the user data while it is stored in the [assignment**:** *memory areas*]**.**

The TSF shall ensure the confidentiality of the information of the user data while it is stored in the **Flash array**.

FDP_SDI.2 Stored data integrity monitoring and action

FDP_SDI.2.1 The TSF shall monitor user data stored in containers controlled by the TSF for [assignment**:** *integrity errors*] on all objects, based on the following attributes: [assignment**:** *user data attributes*]**.**

The TSF shall monitor user data stored in containers controlled by the TSF for **CRC-32 error detecting code** on all objects, based on the following attributes: **stored in the Flash array with CRC-32 and read via authenticated read**.

FDP_SDI.2.2 Upon detection of a data integrity error, the TSF shall [assignment**:** *action to be taken*]**.**

Upon detection of a data integrity error, the TSF shall **inform U.Host-Device about the error. In addition, the TSF also sends a pseudo-randomly chosen part of the CRC-32 error detecting bits to U.Host-Device in a secure manner so that data integrity can be independently verified by U.Host-Device.**

FPT_PHP.3 Resistance to physical attack

FPT_PHP.3.1 The TSF shall resist [assignment**:** *physical tampering scenarios*] to the [assignment**:** *list of TSF devices/elements*] by responding automatically such that the SFRs are always enforced.

The TSF shall resist **physical manipulation and physical probing** to the **TSF** by responding automatically such that the SFRs are always enforced.

Application Note:

The TSF will implement appropriate mechanisms to continuously counter physical manipulation and physical probing. Due to the nature of these attacks (especially manipulation) the TSF can by no means detect attacks on all of its elements. Therefore, permanent protection against these attacks is required ensuring that security functional requirements are enforced. Hence, "automatic response" means here (i) assuming that there might be an attack at any time and (ii) countermeasures are provided at any time.

6.2.4 Leakage

FDP_ITT.1 Basic internal transfer protection

FDP_ITT.1.1 The TSF shall enforce the [assignment**:** *access control SFP(s) and/or information flow control SFP(s)*] to prevent the [selection: disclosure, modification, loss of use] of user data when it is transmitted between physically-separated parts of the TOE.

The TSF shall enforce the **Data Processing Policy** to prevent the ***disclosure*** of user data when it is transmitted between physically-separated parts of the TOE.

Application Note:

The Flash array and the SFF are seen as physically-separated parts of the TOE.

FPT_ITT.1 Basic internal TSF data transfer protection

FPT_ITT.1.1 The TSF shall protect TSF data from [selection: disclosure, modification] when it is transmitted between separate parts of the TOE.

The TSF shall protect TSF data from ***disclosure*** when it is transmitted between separate parts of the TOE.

Application Note:

The Flash array and the SFF are seen as physically-separated parts of the TOE.

FDP_IFC.1 Subset information flow control

FDP_IFC.1.1 The TSF shall enforce the [assignment**:** *information flow control SFP*] on [assignment*: list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP*].

The TSF shall enforce the **Data Processing Policy** on **User data that is processed or transferred by the TOE or by U.Host-Device**.

Application Note:

The following Security Function Policy (SFP) Data Processing Policy is defined for the requirement "Subset information flow control (FDP_IFC.1)"

"User data and TSF data shall not be accessible from the TOE except when the U.Host-Device decides to communicate the User data via an external interface".

6.2.5 Secure Data Exchange

FDP_UCT.1 Basic data exchange confidentiality

FDP_UCT.1.1 The TSF shall enforce the [assignment: *access control SFP(s) and/or information flow control SFP(s)*] to [selection: transmit, receive] user data in a manner protected from unauthorised disclosure.

The TSF shall enforce the **Data Processing Policy** to ***receive*** and ***transmit*** user data in a manner protected from unauthorised disclosure.

FDP_UIT.1 Data exchange integrity

FDP_UIT.1.1 The TSF shall enforce the [assignment: *access control SFP(s) and/or information flow control SFP(s)*] to [selection: transmit, receive] user data in a manner protected from [selection: modification, deletion, insertion, replay] errors.

The TSF shall enforce the **Data Processing Policy** to ***transmit*** and ***receive*** user data in a manner protected from ***replay, modification, deletion*** and ***insertion*** errors.

FDP_UIT.1.2 The TSF shall be able to determine on receipt of user data, whether [selection: modification, deletion, insertion, replay] has occurred.

The TSF shall be able to determine on receipt of user data, whether ***replay, modification, deletion*** and ***insertion*** has occurred.

FTP_TRP.1 Trusted path

FTP_TRP.1.1 The TSF shall provide a communication path between itself and [selection: remote, local] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from [selection: modification, disclosure, [assignment: *other types of integrity or confidentiality violation*]].

The TSF shall provide a communication path between itself and ***remote*** users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from ***modification*** and ***disclosure***.

FTP_TRP.1.2 The TSF shall permit [selection: the TSF, local users, remote users] to initiate communication via the trusted path.

The TSF shall permit ***remote users*** to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for [selection: initial user authentication, [assignment: *other services for which trusted path is required*]].

The TSF shall require the use of the trusted path for **any access to User data stored in the Flash array**.

6.2.6 Protection of Binding Key

FPT_FLS.1/Binding_Key Failure with preservation of secure state

FPT_FLS.1.1/Binding_Key The TSF shall preserve a secure state when the following types of failures occur: [assignment: *list of types of failures in the TSF*].

The TSF shall preserve a secure state when the following types of failures occur: **integrity failure on Binding Key**.

Application Note:

The secure state is defined as follows:

- if the Binding key is illegaly modified, then the TOE is locked;
- if the Binding key is erased, then the TOE User data (stored in the Flash array) is also erased;

FDP_RIP.1 Subset residual information protection

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the [selection: allocation of the resource to, deallocation of the resource from] the following objects: [assignment: *list of objects*]**.**

Refinement:

The TSF shall ensure that any previous information content of the Flash array is made unavailable upon the ***allocation of the resource*** to and ***deallocation of the resource*** from the following objects: the **Binding key (Kb)**.

Application Note:

- "Object Allocation" means that a new Binding key is set in order to replace the current Binding key.
- "Object Deallocation" means that the current Binding key is erased from the TSF (more precisely, from the auxiliary array).

6.3 Security Assurance Requirements

The Evaluation Assurance Level is EAL5 augmented with ALC_DVS.2 and AVA_VAN.5.

6.4 Security Requirements Rationale

6.4.1 Objectives

6.4.1.1 Security Objectives for the TOE

O.Phys-Probing The SFR FDP_SDC.1 requires the TSF to protect the confidentiality of the user data stored in specified memory areas and prevent its compromise by physical attacks bypassing the specified interfaces for memory access. The scenario of physical probing as described for this objective is explicitly included in the assignment chosen for the physical tampering scenarios in FPT_PHP.3. Therefore, it is clear that this security functional requirement supports the objective.

O.Malfunction The definition of this objective shows that it covers a situation, where malfunction of the TOE might be caused by the operating conditions of the TOE (while direct manipulation of the TOE is covered O.Phys-Manipulation). There are two possibilities in this situation: Either the operating conditions are inside the tolerated range or at least one of them is outside of this range. The second case is covered by FPT_FLS.1/Detectors, because it states that a secure state is preserved in this case. The first case is covered by FRU_FLT.2 because it states that the TOE operates correctly under normal (tolerated) conditions.

O.Phys-Manipulation The SFR FDP_SDI.2 requires the TSF to detect the integrity errors of the stored user data and react in case of detected errors. More precisely, FDP_SDI.2

prevents manipulation of memory contents by ensuring detection and response from the TSF (use of a filure counter and capability to lock the session or the TOE itself).

The scenario of physical manipulation as described for this objective is explicitly included in the assignment chosen for the physical tampering scenarios in FPT_PHP.3. Therefore, it is clear that this security functional requirement supports the objective.

O.Abuse-Func This objective states that abuse of functions (especially provided by the IC Dedicated Test Software, for instance in order to read secret data) must not be possible when TOE is used by the final user. There are two possibilities to achieve this: (i) They cannot be used by an attacker (i. e. its availability is limited) or (ii) using them would not be of relevant use for an attacker (i. e. its capabilities are limited) since the functions are designed in a specific way. The first possibility is specified by FMT_LIM.2 and the second one by FMT_LIM.1. Since these requirements are combined to support the policy, which is suitable to fulfil O.Abuse-Func, both security functional requirements together are suitable to meet the objective. Other security functional requirements (FPT_ITT.1, FDP_ITT.1, FPT_PHP.3, FRU_FLT.2, FPT_FLS.1/Detectors and FDP_IFC.1) which prevent attackers from circumventing the functions implementing these two security functional requirements (for instance by manipulating the hardware) also support the objective. The relevant objectives are O.Leak-Inherent, O.Phys-Probing, O.Malfunction, O.Phys-Manipulation, O.Leak-Forced.

O.Leak-Inherent The security functional requirements FPT_ITT.1 and FDP_ITT.1 together with the policy statement in FDP_IFC.1 explicitly require the prevention of disclosure of secret data (TSF data as well as user data) when while being processed. This includes that attackers cannot reveal such data by measurements of emanations, power consumption or other behaviour of the TOE while data is processed by TOE parts.

O.Leak-Forced This objective is directed against attacks, where an attacker wants to force an information leakage, which would not occur under normal conditions. In order to achieve this the attacker has to combine a first attack step, which modifies the behaviour of the TOE (either by exposing it to extreme operating conditions or by directly manipulating it) with a second attack step measuring and analysing some output produced by the TOE. The first step is prevented by the same mechanisms which support O.Malfunction (FPT_FLS.1/Detectors, FRU_FLT.2) and O.Phys-Manipulation (FPT_PHP.3), respectively. The requirements covering O.Leak-Inherent (FPT_ITT.1, FDP_ITT.1, FDP_IFC.1) also support O.Leak-Forced because they prevent the attacker from being successful if he tries the second step directly.

O.Sec-Binding The security functional requirement FDP_RIP.1 ensures that the User data is erased before the Host device is changed. The security functional requirement FPT_FLS.1/Binding_Key protects against integrity failure on Binding Key and illegal modification on Binding Key.

O.Trusted-Path The security functional requirement FTP_TRP.1 contribute in this protection because it only establishes a trusted path between the TSF and authorized **U.Host-Device** for the communication purpose.

The security functional requirement FPT_FLS.1/Binding_Key protects the Binding key against the tampering.

The security functional requirements FDP_UCT.1 and FDP_UIT.1 protect against the modification (integrity) and the disclosure (confidentiality) of the User data communication between the TSF and **U.Host-Device**.

6.4.2 Rationale tables of Security Objectives and SFRs

Security Objectives	Security Functional Requirements	Rationale
O.Phys-Probing	FPT_PHP.3, FDP_SDC.1	Section 6.4.1.1
O.Malfunction	FRU_FLT.2, FPT_FLS.1/Detectors	Section 6.4.1.1
O.Phys-Manipulation	FDP_SDI.2, FPT_PHP.3	Section 6.4.1.1
O.Abuse-Func	FDP_ITT.1, FPT_ITT.1, FPT_PHP.3, FRU_FLT.2, FPT_FLS.1/Detectors, FMT_LIM.1, FMT_LIM.2, FDP_IFC.1	Section 6.4.1.1
O.Leak-Inherent	FDP_ITT.1, FPT_ITT.1, FDP_IFC.1	Section 6.4.1.1
O.Leak-Forced	FDP_ITT.1, FPT_ITT.1, FRU_FLT.2, FPT_FLS.1/Detectors, FPT_PHP.3, FDP_IFC.1	Section 6.4.1.1
O.Sec-Binding	FDP_RIP.1, FPT_FLS.1/Binding_Key	Section 6.4.1.1
O.Trusted-Path	FDP_UCT.1, FDP_UIT.1, FPT_FLS.1/Binding_Key, FTP_TRP.1	Section 6.4.1.1

Table 11 Security Objectives and SFRs - Coverage

Security Functional Requirements	Security Objectives
FRU_FLT.2	O.Malfunction, O.Abuse-Func, O.Leak-Forced
FPT_FLS.1/Detectors	O.Malfunction, O.Abuse-Func, O.Leak-Forced
FMT_LIM.1	O.Abuse-Func
FMT_LIM.2	O.Abuse-Func
FDP_SDC.1	O.Phys-Probing
FDP_SDI.2	O.Phys-Manipulation
FPT_PHP.3	O.Phys-Probing, O.Phys-Manipulation, O.Abuse-Func, O.Leak-Forced
FDP_ITT.1	O.Abuse-Func, O.Leak-Inherent, O.Leak-Forced
FPT_ITT.1	O.Abuse-Func, O.Leak-Inherent, O.Leak-Forced
FDP_IFC.1	O.Abuse-Func, O.Leak-Inherent, O.Leak-Forced
FDP_UCT.1	O.Trusted-Path
FDP_UIT.1	O.Trusted-Path
FTP_TRP.1	O.Trusted-Path
FPT_FLS.1/Binding_Key	O.Trusted-Path, O.Sec-Binding
FDP_RIP.1	O.Sec-Binding

Table 12 SFRs and Security Objectives

6.4.3 Dependencies

6.4.3.1 SFRs Dependencies

Requirements	CC Dependencies	Satisfied Dependencies
FRU_FLT.2	(FPT_FLS.1)	FPT_FLS.1/Detectors
FPT_FLS.1/Detectors	No Dependencies	
FMT_LIM.1	(FMT_LIM.2)	FMT_LIM.2
FMT_LIM.2	(FMT_LIM.1)	FMT_LIM.1
FDP_SDC.1	No Dependencies	
FDP_SDI.2	No Dependencies	
FPT_PHP.3	No Dependencies	
FDP_ITT.1	(FDP_ACC.1 or FDP_IFC.1)	FDP_IFC.1
FPT_ITT.1	No Dependencies	
FDP_IFC.1	(FDP_IFF.1)	

Requirements	CC Dependencies	Satisfied Dependencies
FDP_UCT.1	(FDP_ACC.1 or FDP_IFC.1) and (FTP_ITC.1 or FTP_TRP.1)	FDP_IFC.1, FTP_TRP.1
FDP_UIT.1	(FDP_ACC.1 or FDP_IFC.1) and (FTP_ITC.1 or FTP_TRP.1)	FDP_IFC.1, FTP_TRP.1
FTP_TRP.1	No Dependencies	
FPT_FLS.1/Binding_Key	No Dependencies	
FDP_RIP.1	No Dependencies	

Table 13 SFRs Dependencies

Rationale for the exclusion of Dependencies

The dependency FDP_IFF.1 of FDP_IFC.1 is discarded. Part 2 of the Common Criteria defines the dependency of FDP_IFC.1 (information flow control policy statement) on FDP_IFF.1 (Simple security attributes). The specification of FDP_IFF.1 would not capture the nature of the security functional requirement nor add any detail.

As stated in the Data Processing Policy referred to in FDP_IFC.1, there are no attributes necessary. The security functional requirement for the TOE is sufficiently described using FDP_ITT.1 and its Data Processing Policy (FDP_IFC.1).

6.4.3.2 SARs Dependencies

Requirements	CC Dependencies	Satisfied Dependencies
ADV_ARC.1	(ADV_FSP.1) and (ADV_TDS.1)	ADV_FSP.5, ADV_TDS.4
ADV_FSP.5	(ADV_IMP.1) and (ADV_TDS.1)	ADV_IMP.1, ADV_TDS.4
ADV_IMP.1	(ADV_TDS.3) and (ALC_TAT.1)	ADV_TDS.4, ALC_TAT.2
ADV_INT.2	(ADV_IMP.1) and (ADV_TDS.3) and (ALC_TAT.1)	ADV_IMP.1, ADV_TDS.4, ALC_TAT.2
ADV_TDS.4	(ADV_FSP.5)	ADV_FSP.5
AGD_OPE.1	(ADV_FSP.1)	ADV_FSP.5
AGD_PRE.1	No Dependencies	
ALC_CMC.4	(ALC_CMS.1) and (ALC_DVS.1) and (ALC_LCD.1)	ALC_CMS.5, ALC_DVS.2, ALC_LCD.1
ALC_CMS.5	No Dependencies	
ALC_DEL.1	No Dependencies	
ALC_DVS.2	No Dependencies	
ALC_LCD.1	No Dependencies	
ALC_TAT.2	(ADV_IMP.1)	ADV_IMP.1
ASE_CCL.1	(ASE_ECD.1) and (ASE_INT.1) and (ASE_REQ.1)	ASE_ECD.1, ASE_INT.1, ASE_REQ.2

Requirements	CC Dependencies	Satisfied Dependencies
ASE_ECD.1	No Dependencies	
ASE_INT.1	No Dependencies	
ASE_OBJ.2	(ASE_SPD.1)	ASE_SPD.1
ASE_REQ.2	(ASE_ECD.1) and (ASE_OBJ.2)	ASE_ECD.1, ASE_OBJ.2
ASE_SPD.1	No Dependencies	
ASE_TSS.1	(ADV_FSP.1) and (ASE_INT.1) and (ASE_REQ.1)	ADV_FSP.5, ASE_INT.1, ASE_REQ.2
ATE_COV.2	(ADV_FSP.2) and (ATE_FUN.1)	ADV_FSP.5, ATE_FUN.1
ATE_DPT.3	(ADV_ARC.1) and (ADV_TDS.4) and (ATE_FUN.1)	ADV_ARC.1, ADV_TDS.4, ATE_FUN.1
ATE_FUN.1	(ATE_COV.1)	ATE_COV.2
ATE_IND.2	(ADV_FSP.2) and (AGD_OPE.1) and (AGD_PRE.1) and (ATE_COV.1) and (ATE_FUN.1)	ADV_FSP.5, AGD_OPE.1, AGD_PRE.1, ATE_COV.2, ATE_FUN.1
AVA_VAN.5	(ADV_ARC.1) and (ADV_FSP.4) and (ADV_IMP.1) and (ADV_TDS.3) and (AGD_OPE.1) and (AGD_PRE.1) and (ATE_DPT.1)	ADV_ARC.1, ADV_FSP.5, ADV_IMP.1, ADV_TDS.4, AGD_OPE.1, AGD_PRE.1, ATE_DPT.3

Table 14 SARs Dependencies

6.4.4 Rationale for the Security Assurance Requirements

The assurance level EAL5 and the augmentation with the requirements ALC_DVS.2, and AVA_VAN.5 were chosen in order to meet assurance expectations explained in the following paragraphs.

An assurance level of EAL5 with the augmentations AVA_VAN.5 and ALC_DVS.2 are required for this type of TOE since it is intended to defend against sophisticated attacks. This evaluation assurance package was selected to permit a developer to gain maximum assurance from positive security engineering based on good commercial practices. In order to provide a meaningful level of assurance that the TOE provides an adequate level of defence against such attacks, the evaluators should have access to the low level design and source code.

6.4.5 ALC_DVS.2 Sufficiency of security measures

Development security is concerned with physical, procedural, personnel and other technical measures that may be used in the development environment to protect the TOE.

In the particular case of a memory flash the TOE is developed and produced within a complex and distributed industrial process which must especially be protected. Details about the implementation, (e.g. from design, test and development tools as well as Initialisation Data) may make such attacks easier. Therefore, in the case of a memory flash, maintaining the confidentiality of the design is very important.

This assurance component is a higher hierarchical component to EAL5 (which only requires ALC_DVS.1). ALC_DVS.2 has no dependencies.

6.4.6 AVA_VAN.5 Advanced methodical vulnerability analysis

Due to the intended use of the TOE, it must be shown to be highly resistant to penetration attacks. This assurance requirement is achieved by the AVA_VAN.5 component.

Independent vulnerability analysis is based on highly detailed technical information. The main intent of the evaluator analysis is to determine that the TOE is resistant to penetration attacks performed by an attacker possessing high attack potential.

AVA_VAN.5 has dependencies to ADV_ARC.1 "Security architecture description", ADV_FSP.2 "Security enforcing functional specification", ADV_TDS.3 "Basic modular design", ADV_IMP.1 "Implementation representation of the TSF", AGD_OPE.1 "Operational user guidance", and AGD_PRE.1 "Preparative procedures". All these dependencies are satisfied by EAL5.

It has to be assumed that attackers with high attack potential try to attack memory flashes embedded in smart cards used for digital signature applications or payment systems. Therefore, specifically AVA_VAN.5 was chosen in order to assure that even these attackers cannot successfully attack the TOE.

7 TOE Summary Specification

This Chapter describes the TSF security functionality by a set of security features and justifies how the SFR defined in Chapter 6 are enforced by those features.

7.1 TOE Summary Specification

SF.SEC-COM

Secure communication

SF.SEC-COM protects the confidentiality and the integrity of the communication between the TOE and **U.Host-Device** against probing, Man-in-the-Middle, hammering and replay attacks. In particular,

- a fresh session key is used for each session;
- for update operations (write/erase): the payload (access address and data) is encrypted and a MAC digest is added to ensure integrity;
- for reading operation: 8 transport integrity check bits are added to each 32 bit long word, providing a progressive authentication of the transmitted data;
- session and transaction counters are also used to protect against replaying;

SF.SEC-COM is devised to enable in-place execution of the code stored in the TOE. For this purpose, each data-word sent by TOE is separately encrypted by applying a cascade of a stream ciphering operation and a mixing operation that cryptographically maps input bits to output bits.

Also, to maintain the throughput needed for the in-place execution, the data sent by TOE is authenticated by a sequence of authentication bytes interleaved with the data-words so that each given byte cumulatively authenticates the data words that were authenticated by a previous byte in the sequence and the data words transmitted between the previous byte and the given byte.

SF.PHY-PRO

Physical protection

SF.PHY-PRO protects the TOE against physical manipulation (including the TOE probing). SF.PHY-PRO includes the following security mechanisms:

- Failure counter: this counter is incremented after each tamper-detection and the TOE is locked if the counter reaches a pre-defined value.
- Active Shielding: The Active Shield detection is filtered using a counter, when that number reaches a preset threshold, the Active Shield raises a tamper alarm.
- Dual flip-flops: A difference in the state of two joint flip-flops indicates a fault and raises the Fault Injection Alarm output signal. This mechanism is designed to detect perturbation attacks like Laser or Electro-Magnetic fault injections.
- Clock-tree protection: The 0-1 pattern spreads in a dedicated shift register with every clock pulse provided all clock signals are active. This mechanism is designed to ensure that the clock-tree is intact.
- State machine monitoring: The TOE implements Tamper Detectors that detects abnormal conditions and reports a fault state.

- Bus Encoding: Command bus to the Flash array is encoded, such that more than 1-bit flip distinguishes between any two commands. Further more, some of the bits of the command are used as qualifiers for internal analog processes within the Flash array.

SF.PHY-PRO also protects the TOE against the inherent or intentional leak of the TOE operations by the following security mechanisms:

- advanced stream cipher using long linear feedback shift registers: the calculations are protected against timing and power consumption leak;
- anti-leakage measures for the hash functions that are used for stream-ciphering and MAC digest: masking input data and undisclosure of intermediate output values;
- session setup: the logic is protected against timing and power consumption leak;

SF.OPE-MODE

Control of Operating Modes

SF.OPE-MODE ensures that the TSF and User Data is not disclosed or manipulated via the features avalailible in the TEST mode.

In particular, the Flash array is completely erased before switching to TEST mode. Furthermore, the access to the TSF and User data is also restricted in the Test mode. More precisely:

- The Binding Key (Kb) cannot be read out by the Flash commands;
- The Binding key cannot be erased unless a complete erase has been done after the last reset;
- The read and write commands do not read and write effective values of the Fash array;

SF.OPE-COND

Control of Operating Conditions

SF.OPE-COND detects the abnormal operation conditions (voltage, temperature, clock frequency, power glitch) using the corresponding sensors.

If an abnormal operation condition happens, then SF.OPE-COND disturbs the cryptographic computations, interrupts data interchange and inform **U.Host-Device**.

SF.SEC-MEM-INT

Storage Integrity

SF.SEC-MEM-INT protects the integrity of the User Data (including executable codes) stored in the flash array using CRC-32 error detecting code. All User data can be protected by CRC-32 error detecting code but the integrity verification is performed only if the access is done via an authenticated read (i.e. AUTH_READ command).

If an inconsistency is detected between an User data and its error detecting code, then SF.SEC-MEM-INT informs U.Host-Device about the error.

In addition, SF.SEC-MEM-INT also sends pseudo-randomly chosen of the CRC-32 error detecting code to **U.Host-Device** in a secure way so that data integrity can be independently verified by **U.Host-Device**.

SF.SEC-MEM-CONF

Storage Confidentiality

SF.SEC-MEM-CONF protects the confidentiality of the User Data stored in the flash array by a memory scrambling mechanism that is based on diversified keys. Both the addresses and the memory content are scrambled but by a key that is unique for each instance of the TOE.

SF.KEY-PRO

Protection of Binding Key

SF.KEY-PRO protects the User data against disclosure by manipulating the binding key. In particular, the Flash array is completely erased before

- a new Binding key is set, or
- the current Binding key is erased.

Furthermore, the current Binding key is stored in the Auxiliary array and cannot be read out by the Flash commands. The integrity of the Binding key is protected by a digest value: if an illegal modifcation is detected on the Binding key, then the TOE is locked and can only be unlocked in Test mode (and the Flash array has been erased).

SF.SEC-AUTH

Secure Authentication

SF.SEC-AUTH ensures that only an authorized Host device (i.e. a Host device that knows the Binding key Kb) can open a secure channel to communicate with the TOE.

More precisely, SF.SEC-AUTH provides a mutual authentication between the Host device and the TOE by verifying that both of them share the same Binding key. A failed authentication increases the Failure counter: if this counter reaches a pre-defined value, then the TOE is locked.

7.2 SFRs and TSS

7.2.1 SFRs and TSS - Rationale

7.2.1.1 TOE Summary Specification

SF.SEC-COM enforces the FDP_UCT.1 and FDP_UIT.1 because the the User Data is protected while being transmitted to **U.Host-Device**. SF.SEC-COM enforces the FDP_IFC.1 in particular the user data is protected in terms of confidentility when being transferred by the TOE to **U.Host-Device**. Moreover, the user data is protected in terms of intergrity during the communication between the TOE and **U.Host-Device**.

SF.PHY-PRO enforces the TOE resistance against physical attacks (FPT_PHP.3). SF.PHY-PRO contributes to the integrity and confidentiality protection of the User data stored in the TOE (FDP_SDI.2 and FDP_SDC.1): the failure counter is increased when a data inconsistency is detected; the cryptographic services are also protected against the physical attacks.

SF.PHY-PRO protects against some attacks on the cryptographic services used for the transmission of the User data (FPT_ITT.1, FDP_ITT.1 and FDP_IFC.1).

SF.OPE-MODE enforces the restriction of the TSF capabilities and availabily during the deployment of the test features after the TOE delivery (respectively FMT_LIM.1 and FMT_LIM.2).

SF.OPE-COND enforces the TOE fault-tolerance and fail-secure (respectively FRU_FLT.2 and FPT_FLS.1/Detectors).

SF.SEC-MEM-INT By definition, SF.SEC-MEM-INT enforces FDP_SDI.2.

SF.SEC-MEM-CONF By definition, SF.SEC-MEM-CONF enforces FDP_SDC.1. SF.SEC-MEM-CONF also enforces the FDP_IFC.1 in particular the User data and TSF data are protected in terms of confidentility when being stored, processed or transferred between two TOE components (SFF and Flash array).

SF.KEY-PRO enforces FDP_RIP.1 because it erases the Flash content before a new Binding key is set or the current Binding key is erased. SF.KEY-PRO also detects the failure and put the TOE in a secure state (i.e. locked state) due to an illegal modification of the current Binding key. In other words, SF.KEY-PRO enforces FPT_FLS.1/Binding_Key.

SF.SEC-AUTH enforces the FTP_TRP.1 because only an authorized **U.Host-Device** can open a trusted channel with the TOE.

7.2.2 Association tables of SFRs and TSS

Security Functional Requirements	TOE Summary Specification
FRU_FLT.2	SF.OPE-COND
FPT_FLS.1/Detectors	SF.OPE-COND
FMT_LIM.1	SF.OPE-MODE
FMT_LIM.2	SF.OPE-MODE
FDP_SDC.1	SF.PHY-PRO, SF.SEC-MEM-CONF
FDP_SDI.2	SF.PHY-PRO, SF.SEC-MEM-INT
FPT_PHP.3	SF.PHY-PRO
FDP_ITT.1	SF.PHY-PRO
FPT_ITT.1	SF.PHY-PRO
FDP_IFC.1	SF.SEC-MEM-CONF, SF.SEC-COM, SF.PHY-PRO
FDP_UCT.1	SF.SEC-COM
FDP_UIT.1	SF.SEC-COM
FTP_TRP.1	SF.SEC-AUTH
FPT_FLS.1/Binding_Key	SF.KEY-PRO
FDP_RIP.1	SF.KEY-PRO

Table 15 SFRs and TSS - Coverage

TOE Summary Specification	Security Functional Requirements
SF.SEC-COM	FDP_IFC.1, FDP_UCT.1, FDP_UIT.1
SF.PHY-PRO	FDP_SDC.1, FDP_SDI.2, FPT_PHP.3, FDP_ITT.1, FPT_ITT.1, FDP_IFC.1
SF.OPE-MODE	FMT_LIM.1, FMT_LIM.2
SF.OPE-COND	FRU_FLT.2, FPT_FLS.1/Detectors
SF.SEC-MEM-INT	FDP_SDI.2
SF.SEC-MEM-CONF	FDP_SDC.1, FDP_IFC.1
SF.KEY-PRO	FPT_FLS.1/Binding_Key, FDP_RIP.1
SF.SEC-AUTH	FTP_TRP.1

Table 16 TSS and SFRs - Coverage

8 Revisions

Modification	Comment
A	New version
E	Certified Version
E1	Lite version

Table 17 History of Modifications

9 ANNEX

9.1 Glossary

SFI

Secure Flash Interface is the SPI interface on the Host device (i.e. SPI Master).

SFF

Secure Flash Front-end is the SPI interface on the memory chip (i.e. SPI Slave).

SPI

Serial Peripheral Interface is a synchronous serial data link, a *de facto* standard, that operates in full duplex mode.

9.2 Abbreviations

CC Common Criteria

EAL Evaluation Assurance Level

IT Information Technology

PP Protection Profile

ST Security Target

TOE Target of Evaluation

TSC TSF Scope of Control

TSF TOE Security Functionality

TSFI TSF Interface

TSP TOE Security Policy

9.3 References

[1] Common Criteria, *Part 1: Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model,* Version 3.1, Revision 5, April 2017, CCMB-2017-04-001

[2] Common Criteria, *Part 2: Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components*, Version 3.1, Revision 5, April 2017, CCMB-2017-04-002

[3] Common Criteria, *Part 3: Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components*, Version 3.1, Revision 5, April 2017, CCMB-2017-04-003

[4] *Common Methodology for Information Technology Security Evaluation, Evaluation Methodology*, Version 3.1, Revision 5, April 2017, CCMB-2017-04-004

[5] Eurosmart, *Security IC Platform with Augmentation Packages*, Version 1.0, February 2014, BSI-PP-0084.

[6] Winbond Technology Ltd., *SpiFlash 1.8V secure flash memory* Datasheet.

[7] Winbond Technology Ltd., *N/A*.

[8] Joint Interpretation Library: *Application of Attack Potential to Smartcards*, April 2019, Version 3.0

[9] Supporting Document, Mandatory Technical Document*: The Application of CC to Integrated Circuits*, March 2009, Version 3.0, Revision 1, CCDB-2009-03-002

[10] Supporting Document Guidance: *Smartcard Evaluation*, February 2010, Version 2.0, CCDB-2010-03-001

[11] *Supporting Document Guidance Security Architecture requirements (ADV_ARC) for smart cards and similar devices*, April 2012, Version 2.0, CCDB-2012-04-003

[12] *N/A*

[13] *Supporting Document Mandatory Technical Document: Application of Attack Potential to Smartcards* April 2013, Version 2.3, CCDB-2013-05-002

[14] *Supporting Document: Composite product evaluation for Smart Cards and similar devices*, May 2018, Version 2.1

[15] *Joint Interpretation Library: Minimum Site Security Requirements*, April 2019

[16] ISO/IEC 7816-3. *Identification cards — integrated circuit cards. Part 3: Cards with contacts Electrical interface and transmission protocols*.

Preliminary Designation

The "Preliminary" designation on a *Winbond* datasheet indicates that the product is not fully characterized. The specifications are subject to change and are not guaranteed. *Winbond* or an authorized sales representative should be consulted for current information before using this product.

Trademarks

Winbond, *SpiFlash* and *TrustME* are trademarks of *Winbond Electronics Corporation.*
All other marks are the property of their respective owner.

Important Notice

Winbond products are not designed, intended, authorized or warranted for use as components in systems or equipment intended for surgical implantation, atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for other applications intended to support or sustain life. Furthermore, *Winbond* products are not intended for applications wherein failure of *Winbond* products could result or lead to a situation wherein personal injury, death or severe property or environmental damage could occur.
Winbond customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify *Winbond* for any damages resulting from such improper use or sales.

Information in this document is provided solely in connection with Winbond products. Winbond reserves the right to make changes, corrections, modifications or improvements to this document and the products and services described herein at any time, without notice.