TrustME™

W77Q

[16/32]JW[W/R/SF/SS/ST/SN/SV/ZP/TB/TC/XF/XG/XH/UU/UX/UZ/BY/BJ/BK]

Secure Flash Memory

Security Target Lite

Contents

Table of Figures

Table of Tables

1 Security Target Introduction

1.1 Security Target Lite Reference

Title: W77Q[16/32]JW[W/R/SF/SS/ST/SN/SV/ZP/TB/TC/XF/XG/XH/UU/UX/UZ/BY/BJ/BK] Secure Flash Memory Security Target Lite

Version: F

Authors: Winbond Technology Ltd.

Evaluator: Brightsight B.V.

Certified by: TUV Rheinland

1.2 TOE Reference

The Target of Evaluation is identified as below:

Commercial Name	SpiFlash® TrustME™ Secure Flash Memory
Product Name	W77Q16/32
Version	C
Guidance	Datasheet [7] Operational User Guidance [8] Preparative Procedure [9] Security Manual [10]

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 systems that need memory content to be protected.The TOE is dedicated to the secure storage of the code and data of IoT applications.

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 to stimulate the TSF.
b) SPI Bus for the communication between the Host device and the TOE.

The ST assumes the components of the Host Device that provide the TOE security environment are properly protected, as described in Section 3.5 .

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.

The table below lists possible forms of the delivery. The difference between these forms is packaging. The silicon is the same in all cases.

NO	TYPE	IDENTIFIER	PART NUMBER	DELIVERY METHOD
FORM OF DELIVERY : KNOWN GOOD DIE FORM				
1	HW	IC Part number	W77Q16JWW	Via Courier
2	HW	IC Part number	W77Q32JWW	Via Courier
FORM OF DELIVERY : KNOWN GOOD DIE REDISTRIBUTION LAYER (RDL) FORM				
1	HW	IC Part number	W77Q16JWR	Via Courier
2	HW	IC Part number	W77Q32JWR	Via Courier
FORM OF DELIVERY : ASSEMBLED DEVICE IN SOP16 300MIL (THICKNESS 2.64 MM)				
1	HW	IC Part number	W77Q16JWSF	Via Courier
2	HW	IC Part number	W77Q32JWSF	Via Courier
FORM OF DELIVERY : ASSEMBLED DEVICE IN SOP8 208 MIL (THICKNESS 2.16MM)				
1	HW	IC Part number	W77Q16JWSS	Via Courier
2	HW	IC Part number	W77Q32JWSS	Via Courier
FORM OF DELIVERY : ASSEMBLED DEVICE IN VSOP8 208 MIL (THICKNESS 1.0MM)				
1	HW	IC Part number	W77Q16JWST	Via Courier
2	HW	IC Part number	W77Q32JWST	Via Courier
FORM OF DELIVERY : ASSEMBLED DEVICE IN SOP8 150 MIL (THICKNESS 1.75 MM)				
1	HW	IC Part number	W77Q16JWSN	Via Courier
2	HW	IC Part number	W77Q32JWSN	Via Courier
FORM OF DELIVERY : ASSEMBLED DEVICE IN VSOP8 150 MIL (THICKNESS 0.9 MM)				
1	HW	IC Part number	W77Q16JWSV	Via Courier
2	HW	IC Part number	W77Q32JWSV	Via Courier
FORM OF DELIVERY : ASSEMBLED DEVICE IN WSON8 6X5 (THICKNESS 0.8 MM)				

NO	TYPE	IDENTIFIER	PART NUMBER	DELIVERY METHOD
1	HW	IC Part number	W77Q16JWZP	Via Courier
2	HW	IC Part number	W77Q32JWZP	Via Courier
FORM OF DELIVERY : ASSEMBLED DEVICE IN TFBGA24 8X6 (5X5-1 BALL ARRAY)				
1	HW	IC Part number	W77Q16JWTB	Via Courier
2	HW	IC Part number	W77Q32JWTB	Via Courier
FORM OF DELIVERY : ASSEMBLED DEVICE IN TFBGA24 8X6 (6X4 BALL ARRAY)				
1	HW	IC Part number	W77Q16JWTC	Via Courier
2	HW	IC Part number	W77Q32JWTC	Via Courier
FORM OF DELIVERY : ASSEMBLED DEVICE IN XSON10 4X4 (THICKNESS 0.5 MM)				
1	HW	IC Part number	W77Q16JWXF	Via Courier
2	HW	IC Part number	W77Q32JWXF	Via Courier
FORM OF DELIVERY : ASSEMBLED DEVICE IN XSON8 4X4 (THICKNESS 0.5 MM)				
1	HW	IC Part number	W77Q16JWXG	Via Courier
2	HW	IC Part number	W77Q32JWXG	Via Courier
FORM OF DELIVERY : ASSEMBLED DEVICE IN XSON8 2X3 (THICKNESS 0.4 MM)				
1	HW	IC Part number	W77Q16JWXH	Via Courier
2	HW	IC Part number	W77Q32JWXH	Via Courier
FORM OF DELIVERY : ASSEMBLED DEVICE IN USON8 4X3 (THICKNESS 0.6 MM)				
1	HW	IC Part number	W77Q16JWUU	Via Courier
2	HW	IC Part number	W77Q32JWUU	Via Courier
FORM OF DELIVERY : ASSEMBLED DEVICE IN USON8 2X3 (THICKNESS 0.6 MM)				
1	HW	IC Part number	W77Q16JWUX	Via Courier
2	HW	IC Part number	W77Q32JWUX	Via Courier
FORM OF DELIVERY : ASSEMBLED DEVICE IN USON8 4X4 (THICKNESS 0.6 MM)				
1	HW	IC Part number	W77Q16JWUZ	Via Courier
2	HW	IC Part number	W77Q32JWUZ	Via Courier
FORM OF DELIVERY : ASSEMBLED DEVICE IN 12-BALL WLCSP (THICKNESS 0.54 MM)				
1	HW	IC Part number	W77Q16JWBY	Via Courier
2	HW	IC Part number	W77Q32JWBY	Via Courier
FORM OF DELIVERY : ASSEMBLED DEVICE IN 12-BALL WLCSP (THICKNESS 0.5 MM)				
1	HW	IC Part number	W77Q16JWBJ	Via Courier
2	HW	IC Part number	W77Q32JWBJ	Via Courier
FORM OF DELIVERY : ASSEMBLED DEVICE IN 12-BALL WLCSP (THICKNESS 0.5 MM)				
1	HW	IC Part number	W77Q16JWBK	Via Courier
2	HW	IC Part number	W77Q32JWBK	Via Courier
FORM OF DELIVERY : ASSOCIATED IC DEDICATED DOCUMENTATION				
1	PDF	Operational User Guidance [8]	Version C	Mail

NO	TYPE	IDENTIFIER	PART NUMBER	DELIVERY METHOD
2	PDF	Preparative Procedure [9]	Version C	Mail
3	PDF	Security Manual [10]	Version A7	Mail
4	PDF	Datasheet [7]	Version A6	Mail

Table 2 TOE Identification

1.4.1.1 TOE Physical Characteristics

The TOE physical characteristics are described as follows.

Performance:

Up to 133 MHz Standard/Quad SPI clocks (STR mode)

Up to 66 MHz Standard/Quad SPI clocks (DTR mode)

Up to 66 MB/s continuous data transfer rate (plain text)

Up to 6 MB/s encrypted and authenticated data transfer rate

More than 100,000 erase/program cycles

More than 20-year data retention

Operating conditions:

Single 1.65 to 1.95V supply

2mA active current, <1µA Power-down (typ.)

-40°C to +85°C or 105°C operating range

1.4.1.2 TOE Architecture

The architecture of the Memory Flash is described in Figure 1. The TOE includes only the W77Q device. The MCU and SPI bus appearing in this diagram are not part of the TOE.

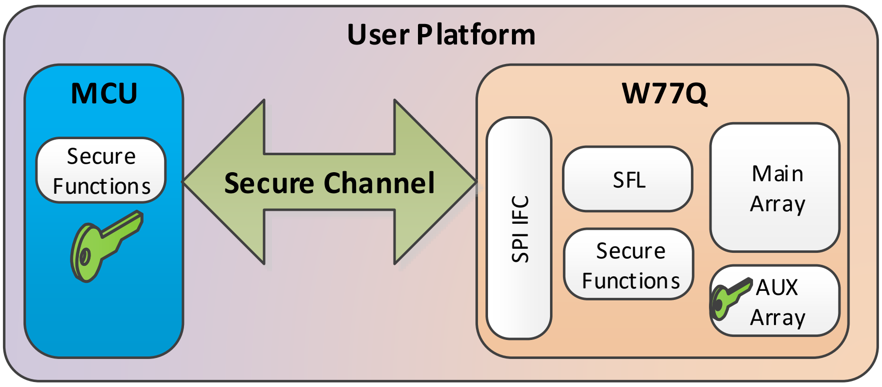


Figure 1 TOE Architecture

The TOE consists of the following Hardware components:

- **Main Flash Array:** stores the User data (i.e. the mass data including executable code);

- **Auxiliary array:** contains the TSF data, including: keys, secure configurations, Winbond unique ID (WID), and Monotonic Counter.
- **QSF Logic (Secure Functions):** implements the TSF, including: data encryption, command authentication, access privileges management, and Secure Channel support.
- **SFL (Standard Flash Logic):** implements datapath for plain access to the unprotected areas of the Main Array (as in standard Flash devices).
- **SPI Interface:** supports communication over the SPI bus.

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#, IO0, IO1
  - Quad SPI: CLK, CS#, IO0, IO1, IO2, IO3

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, Test mode entry is cryptographically protected – an unauthorized switch from User mode to Test mode erases all protected User data and all TSF data;
- Protection against **leakage and physical** attacks;
- ***Confidentiality, Authenticity*** and ***integrity*** of Secret User Data;
- ***Authenticity*** and ***integrity*** of Authenticated User Data;
- ***Integrity protection*** of the flash content by error detection codes (CRC-32);
- Memory **Rollback** protection, **Irreversibility-Anchor** and **Clone Replace Protection;**
- ***Secure Communication Channel*** with the host device and a remote operator;
- ***Memory Access Control*** of the flash content by implementing an access control policy with different levels of authorization, typically:
  - Integrity Protection
  - Write Protection
  - Rollback Protection
  - Plain Access Read
  - Plain Access Write
  - Plain Access Authenticated
- Protection of the **secure boot of the Host Device and secure update** process;

- Secure Key Provisioning Mechanism.

1.4.2.1 TOE Operating Modes

The TOE can operate in one of two operating modes: *User Mode* and *Test Mode*.

In **User Mode**:

- The Main array is logically divided into Sections; each can be configured with its own Memory access control Security Policy.
- Access Privileges are enforced by the TOE according to the Memory access control Security Policy.
- No external interface exists to access the Aux array.

Test Mode consists of two sub-states:

- *Test-Mode-Calibration* allows access to a limited set of non-TSF test commands, used for initial device calibration. In this state, any form of data access is prohibited.
- *Test-Mode-Full* allows access to all test commands, including access to the Main and Aux arrays. Although, the access to the Aux array is limited:
  - Read access to the Aux array is limited so it is not possible to read the keys managed by the TOE.
  - Write access to the Aux array is limited so only a limited set of test patterns could be written to the array.

2 Conformance Claim

This Conformance Claim chapter contains the following sections:

- CC Conformance Claim
- PP Claim
- Package 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 [1] extended and CC Part 3 [2] conformant.

The extended Security Functional Requirements are defined in chapter 5.

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 EAL2 + ALC_FLR.2.

3 Security Problem Definition

3.1 Users / Subjects

The security functionality of the TOE allows different usage scenarios. Most commonly, the following subjects interact with the TOE:

- *U.Host-Device* **–** the host device that embeds the TOE and communicates with it through a SPI Bus.
- *U.Remote-Operator* **–** communicates with the TOE through the *U.Host-Device*. However, the security of the communication between the TOE and the *U.Remote-User* is guaranteed even if the *U.Host-Device* is compromised.

3.2 Assets

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

- TSF, described in Section 3.2.1
- TSF data, described in Section 3.2.2
- Protected User data, described in Section 3.2.3

3.2.1 TSF

The TOE does not include any software – the logic of the TOE security mechanisms is implemented in Hardware.

- QSF Logic – The HW Logic implementing the TSF (see Figure 1 TOE Architecture) is part of the TSF and it is protected in terms of integrity and confidentiality.

3.2.2 TSF data

The following non-volatile TSF data is stored in the auxiliary array of the memory chip:

- *Device Master Key* – used for secure key provisioning and memory configuration, protected in terms of integrity and confidentiality
- *Per-Section Keys* –used to access the section's data and its security functions, protected in terms of integrity and confidentiality*:*
  - *restricted Section Master Keys (read-only Section access)*
  - *non-restricted Section Master Keys (full Section access)*
- *TOE Metadata* protected in terms of integrity:
  - Winbond Device ID
  - Secure Unique Device ID
  - Global Memory Configuration
  - Global Mapping Table

  - Section Configuration Registers

- *Monotonic Counter* **–** used for replay protection, protected in terms of integrity

At the runtime, the data necessary for the execution of the TSF is stored in the registers of the QSF Logic:

- *Runtime data* – session parameters and the non-volatile TSF data loaded to QSF Logic registers. The session parameters consist of the Session Key (protected in terms of confidentiality) and the Transaction Counter.

3.2.3 Protected User data

The memory is split into eight sections and the limits of each section as well as its security attributes are defined in the TOE Metadata (Global Memory Configuration, Global Mapping Table, and Section Configuration Registers):

- **Secret User Data** – Data (including executable codes) stored in the section of the Flash array that are defined as protected in terms of data confidentiality.
- **Authenticated User Data** – Data (including executable codes) stored in the section of the Flash array that are defined as protected in terms of data integrity and authentication.

3.3 Threats

The following threats agents and threats have been identified:

3.3.1 Threat Agents

The TOE may be attacked by the following threat agents:

- Local attackers carrying out (local) physical or logical attacks and
- Remote attackers carrying out (remote) logical attacks.

3.3.2 External NVM Augmentation Package Threats

As for any external memory device, the threats described in the Augmentation Package [5] are relevant for the TOE, suited to the threat model.

Application Note: The following threats have been taken from the Augmentation Package [5] and no modification has been carried out. Nonetheless, the following clarification is made:

- *The external NVM, external NVM of the TOE or NVM in [5] is the full TOE in this ST*
- *The host MCU in [5] is the user of the TOE (U.Host-Device or the U.Remote-Operator) in this ST*
- *The interface between host MCU and the external NVM in [5] is the TOE SPI interface.*

3.3.2.1 T.External-Content-Abuse - Unauthorized access of NVM contents

An attacker may attempt to access for disclosing or modifying the contents of the external NVM.

3.3.2.2 T.NVM-Command-Replay - Replay of commands between the host MCU and the external NVM

An attacker may attempt to replay the write and erase commands or responses to the read commands between the host MCU and the external NVM, to affect the freshness of the contents read from or written to the external NVM.

3.3.2.3 T.NVM-Unauthorized-Rollback - Unauthorized rollback of NVM contents to a previous version

An attacker may attempt to read the contents of the external NVM, record them, and later write them back to the external NVM after the original contents were updated by the host MCU.

3.3.2.4 T.NVM-Clone-Replace - Cloning or replacement of NVM

An attacker may attempt to clone the full contents of the external NVM of the TOE and write them to the external NVM of a different TOE unit; alternatively, an attacker may physically replace the NVM of a TOE with a different NVM that may come from a different TOE unit.

3.3.2.5 T.NVM-Abuse-Interface: Abuse of interface between host MCU and external NVM

An attacker may abuse the interface between the host MCU and the external NVM to disclose the data in transit, manipulate the data in transit, block a command or issue commands for modification of external NVM contents.

3.3.3 PP0084 Threats

As for any secure device, some of the threats described in [6] are relevant for the TOE. The threats have been adapted to the case of the secure memory.

3.3.3.1 T.Phys-Manipulation – Physical Manipulation

An attacker may physically modify the Memory Flash in order to achieve the following:

- Modify *Protected User Data* stored in the TOE;
- Modify *TSF Data* stored in the TOE;
- Modify the security mechanisms of the TOE (provided by *TSF logic*) to enable attacks disclosing or manipulating *Protected User Data and TSF Data*.

3.3.3.2 T.Abuse-Func - Abuse of Functionality

An attacker may try to use functions of the TOE, which may not be used after TOE Delivery, in order to achieve the following:

- Disclose or manipulate *TSF data* or *Protected User Data*,
- Enable an attack disclosing or manipulating *TSF data* or *Protected User Data*.

3.3.3.3 T.Leak-Inherent - Inherent Information Leakage

An attacker may exploit information leaked from the TOE during usage of the Memory Flash to disclose the confidential data stored and processed in the TOE

3.3.3.4 T.Leak-Forced - Forced Information Leakage

An attacker may exploit information leaked from the TOE during usage of the Memory Flash to disclose confidential *TSF data* stored and processed in the TOE, even if the information leakage is not inherent but caused by the attacker.

3.3.4 Other Threats

3.3.4.1 T.Insecure-Boot - Insecure state of the U.Host-Device after boot

An attacker may attempt to disturb the boot process of the *U.Host-Device* by corrupting the boot code stored on the TOE. An attacker may (i) force an invalid configuration of the U.Host-Device, (ii) masquerade the unique *U.Host-Device* identity, or (iii) archive an inconsistent initialization of the Root of Trust in order to compromise secrets or enable other threats.

3.3.4.2 T.Faulty-Update - Faulty Software update of the U.Host-Device code

An unauthorized user provides an unauthorized faulty software update, thus enabling attacks against the integrity of *U.Host-Device* TSF implementation, confidentiality and integrity of

U.Host-Device user data and *U.Host-Device* TSF data after installation of the faulty software update.

3.3.4.3 T.No Auth-Access – UnAuthorized memory access

An attacker may attempt to gain access to a memory section without having the right level of authorization.

Note: This threat is an extension of T.External-Content-Abuse for the case of memory access with different levels of authorization.

3.4 Organizational Security Policies

Either the TOE Manufacturer or the TOE Integrator shall apply the policy specified below.

3.4.1 P.Key-Provisioning: Secure Key provisioning

A secure process of in-the-field key provisioning must be carried out protecting the keys confidentiality and authenticity.

3.4.2 P.Gen-Unique-ID: Identification of each TOE instance

An accurate identification must be established for the TOE. The policy requires that each instantiation of the TOE stores its own unique identification.

3.4.3 P.Update-Host-Device: Authorized Software Update

The *U.Host-Device* Software Updates are delivered in (optionally encrypted) and genuine form by the *U.Remote-Operator*. The TOE verifies the authenticity of the received Software Update before storing it. The TOE restricts the storage of authentic Software Update to an authorized user.

3.5 Assumptions

3.5.1 A.Key-Protection: Protection of the TOE keys

It is assumed that security procedures are used for key generation, key provisioning ant TOE operation to maintain confidentiality and integrity of the TOE keys.

3.5.2 A.Secure-Channel: Communication Protection by Secure Channel

It is assumed that *U.Host-Device* and the *U.Remote-Operator* support the trusted communication channel with the TOE protecting the confidentiality, integrity, and freshness of the transmitted data.

3.5.3 A.Boot-Host-Device: Boot protected by TOE

It is assumed that *U.Host-Device* boot code is stored on the TOE.

4 Security Objectives

4.1 Security Objectives for the TOE

4.1.1 External NVM Augmentation Package Security Objectives for the TOE

As for any external memory device, the security objectives described in the Augmentation Package [5] are relevant for the TOE, suited to the threat model.

Application Note: the following security objectives have been taken from the Augmentation Package [5] and no modification has been carried out. Nonetheless, the following clarification is made:

- *The external NVM, external NVM of the TOE or NVM in [5] is the full TOE in this ST*
- *The host MCU and Security IC in [5] is the user of the TOE (U.Host-Device) in this ST*
- *The interface between host MCU and the external NVM in [5] is the TOE SPI interface.*

4.1.1.1 O.External-Content-Protection - Protection against disclosure and undetected modification of external NVM contents

Since an attacker can get direct access to the external NVM, the contents stored in the external NVM must be protected against disclosure and undetected modification. The TOE shall provide confidentiality and integrity protection of the contents stored in external NVM

Application Note:

This security objective is applicable to the protection of the *Protected User Data*.

4.1.1.2 O.NVM-Command-Replay-Protection - Protection against replay of commands between host MCU and external NVM

The TOE shall protect against replay of the read, write and erase commands issued from the Security IC to the external NVM through the interconnection bus

Application Note:

This security objective is applicable to the commands issued through the secure channel via the SPI interface.

4.1.1.3 O.NVM-Unauthorized-Rollback-Protection - Protection against an unauthorized rollback of NVM contents

The TOE shall protect replacing the external NVM contents with a previous version, even if it was valid in the past.

Application Note:

This security objective is applicable to the protection of the *Protected User Data*.

4.1.1.4 O.NVM-Irreversibility-Anchor - External NVM Contents Irreversibility Anchor

The TOE shall implement a non-volatile mutable mechanism that goes through a predefined sequence of states (that are associated with increasing 'values') that can never be returned to a previous state. This value given by a sequence of states shall be used to determine whether the external NVM contents meet the data freshness property and to prevent replay attacks.

Application Note:

This security objective is applicable to the access to the TOE through the secure channel.

4.1.1.5 O.NVM-Clone-Replace-Protection - Protection against NVM cloning or replacement

The TOE shall protect against cloning the memory contents of another unit into the TOE's external NVM and against replacement of the external NVM with the one from a different unit.

Application Note:

The TOE shall protect against cloning the *Protected User Data* of another unit into the TOE's NVM and against replacement of the TOE with a different NVM device. While the contents are valid for a TOE from where they were extracted, they shall be detected as non-belonging to the TOE unit where they were cloned to and, thus, non-valid.

4.1.1.6 O.NVM-Interface-Protection - Protection against abuse of the interface between host MCU and external NVM

The TOE shall protect the data in transit between the host MCU and the external NVM against disclosure. The TOE shall also detect manipulation of the data in transit through the interconnection bus and manipulation through issuing commands to the NVM.

Application Note:

The TOE protects the data in the following manner:

- By enforce secure communication (i.e., the secure channel) for the access to the Protected User Data and
- By protecting the secure communication against *Secret User Data* disclosure, and *Authenticated User Data* illegal modification

Since the SPI bus between the *U.Host-Device* and the TOE is an easy subject to attacks, it is required that the TOE prevents disclosure of data in transit through it, and it is required that the TOE detects manipulation of such data in transit. An attacker issuing new commands for modifying the data in the NVM must be detected by the TOE as well.

4.1.2 PP0084 Security Objectives for the TOE

As for any secure device, some of the security objectives for the TOE described in [6] are relevant for the TOE. The security objectives for the TOE have been adapted to the case of the secure memory.

4.1.2.1 O.Phys-Manipulation – Protection against Physical Manipulation

The TOE must provide protection against manipulation of Protected User and TSF data. This includes protection against the following threats:

- Reverse-engineering (understanding the design and its properties and functions);
- Manipulation of the hardware and TSF data, as well as;
- Undetected manipulation of Protected User Data (i.e. Flash array).

4.1.2.2 O.Abuse-Func - Protection against Abuse of Functionality

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

4.1.2.3 O.Leak-Inherent - Protection against Inherent Information Leakage

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

- For the *Secret User Data* – by information that can be perceived by a remote attacker – typically, the SPI signals.
- For the *TSF data*, in addition to this – by Side Channel information: measurement and analysis of the shape and amplitude of signals (for example on the power, clock, or I/O lines), timing, etc.

4.1.2.4 O.Leak-Forced - Protection against Forced Information Leakage

The TOE must be protected against disclosure of confidential *TSF 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 due to an environmental stress, i.e., by operating the TOE outside the normal operating conditions.
- By a physical manipulation of the TOE.

4.1.2.5 O.Identification - TOE Identification

The TOE must provide means to store Initialization Data in its non-volatile memory. The Initialization Data (or parts of them) are used for TOE identification.

Application Note:

The unique identifier is stored for each TOE instance. The unique identifier is protected against modification and it is used for TOE identification. In addition, any field in the Authenticated User Data can be used for the TOE identification.

In addition, the TOE identification can be used for Platform RoT initialization as explained in Section 4.1.4.3.

4.1.3 Other Security Objectives for the TOE

4.1.3.1 O.Secure-Boot - Protection against Boot code corruption

The TOE shall protect the boot process of the *U.Host-Device* against unauthorized change of the boot code stored on the TOE ensuring the code authenticity and integrity.

4.1.3.2 O.Secure-Update: Secure import of Host-Device Software Update

The TSF verifies the authenticity and freshness of received *U.Host-Device* Software Update and allows the TOE to perform the *U.Host-Device* Software Update.

4.1.3.3 O.Access-Policy: Memory Access Policy

The TOE shall implement an access control policy with different levels of authorization to the memory sections.

4.1.3.4 O.Key-Provisioning: Secure Key provisioning

The TOE shall provide a secure process of in-the-field key provisioning to protect the key confidentiality and authenticity.

4.1.4 TOE as a Root-of-Trust

The Security Objectives listed above and the security mechanisms implementing the corresponded SFR listed below allow the TOE to serve as the Root-of-Trust (RoT) for the U.Host-Device.

4.1.4.1 TOE is a Trusted Device.

Here is a place to explain the difference between “security” and “trust”:

- Security is a systemwide concept. By protecting a part of the system (e.g., the external memory interface), we cannot ensure that the system/platform/product is secure.
- A trusted product, on the other hand, takes specific tasks and carries them out securely. As long as the required resources are available, it can be trusted that the job is completed correctly, no secrets are compromised, etc.

To summarize, a trusted part cannot provide the overall system security, but it can ensure that the attacker cannot interfere with the protected functionality.

TOE, that controls the access to the code and data, can securely implement several critical security tasks:

- Cryptographic multi-level access control, including write-protect,
- Secure code update with rollback protection,
- Remote attestation with cryptographic challenge-response,

The trusted functionality described above can be used as the basis for systemwide protection. In particular, TOE can be used as a Root of Trust in two aspects:

- **Application RoT** – for protection of the Host-Device SW stack from unauthorized modification,
- **Platform RoT** – for assuring an external entity that it communicates with a genuine device running unmodified code

4.1.4.2 Application RoT

This functionality is based on O.Secure-Boot, which ensures secure initialization of the SW stack. Other SW layers can be protected either by TOE in the same way as the Boot code or by the SW mechanism implemented in, e.g., the Boot code

The Application RoT allows secure update of any SW layer, as covered by the O.Secure-Update Security Objective.

4.1.4.3 Platform RoT

This functionality is requires a cryptographic platform identification (e.g., via a challenge-response), which is allowed by O.Identification and O.Access-Policy. By identification of the TOE instance one achieves the Platform identification due to O.NVM-Clone-Replace-Protection

This Rot is available as soon as the unique identifier is programmed to the TOE. It may cover earlier stages of the life cycle such as shipping, installation, and operation

4.2 Security Objectives for the Operational Environment

4.2.1 OE.Gen-Unique-ID: Generation of device's individual identifier

Before a TOE instantiation is used, it shall be allotted with its own unique ID.

4.2.2 OE.Key-Protection: Protection of the TOE keys

Security procedures shall be used by the TOE operators to maintain the confidentiality and the integrity of the TOE keys. Namely:

- The keys shall be generated with the required amount of entropy.
- The provisioning of the *Device Master Key* shall be done in a secure environment where the communication between with the TOE is protected from eavesdropping.
- Note: Provisioning of all other keys is protected by the TOE based on the confidentiality of the *Device Master Key*
- Keys stored in the *U.Host-Device* and shared with the TOE shall be protected by the *U.Host-Device*
- Keys stored at the *U.Remote-Operator* and shared with the TOE shall be protected by the *U.Remote-Operator*

4.2.3 OE.Secure-Channel: Secure communication with the TOE

The *U.Host-Device* and the U.Remote Operator shall support the trusted communication channel with the TOE protecting the confidentiality, integrity, and freshness of the transmitted data.

Note: Data freshness means that the stored and transmitted data is always the one resulting in the last change carried out by the authorized user on the TOE.

4.2.4 OE.Boot-Host-Device: Boot protected by TOE

The *U.Host-Device* boots from code stored on the TOE and the *U.Host-Device* in a dedicated memory section.

4.2.5 OE.Update-Host-Device: Genuine Software Update

The *U.Host-Device* update of the code stored on the TOE is carried out by the *U.Remote-Operator* using the protective mechanisms of the TOE. The secure Software Update is delivered in encrypted and genuine protected form by the authorized issuer together with its security attributes.

4.3 Security Objectives Rationale

ASSUMPTION, THREAT OR ORGANIZATIONAL SECURITY POLICY	SECURITY OBJECTIVE
T.NVM-Abuse-Interface	O.NVM-Interface-Protection
T.External-Content-Abuse	O.External-Content-Protection
T.NVM-Command-Replay	O.NVM-Command-Replay-Protection O.NVM-Irreversibility-Anchor
T.NVM-Unauthorized-Rollback	O.NVM-Unauthorized-Rollback-Protection O.NVM-Irreversibility-Anchor
T.NVM-Clone-Replace	O.NVM-Clone-Replace-Protection
T.Phys-Manipulation	O.Phys-Manipulation
T.Abuse-Func	O.Abuse-Func
T.Leak-Inherent	O.Leak-Inherent
T.Leak-Forced	O.Leak-Forced
T.Insecure-Boot	O.Secure-Boot
T.Faulty-Update	O.Secure-Update
T.No Auth-Access	O.Access-Policy
P.Key-Provisioning	O.Key-Provisioning
P.Gen-Unique-ID	OE.Gen-Unique-ID O.Identification
P.Update-Host-Device	O.Secure-Update OE.Update-Host-Device
A.Key-Protection	OE.Key-Protection
A.Secure-Channel	OE.Secure-Channel
A.Boot-Host-Device	OE.Boot-Host-Device

Table 3 Security Objectives Rationale

5 Extended Requirements

5.1 Extended Families

5.1.1 FDP_URC: Protection against an unauthorized rollback of stored contents

Note: This extended family comes from [5].

5.1.1.1 Description

To define the security functional requirements of the TOE we use an additional family (FDP_URC) of the Class FDP (User data protection).

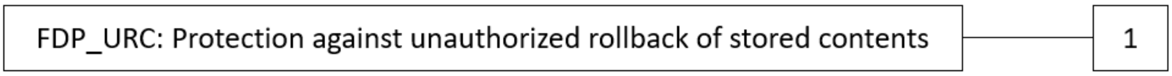
The family “Protection against an unauthorized rollback of stored contents (FDP_URC)” is specified as follows.

FDP_URC Protection against an unauthorized rollback of stored contents

Family Behaviour:

This family defines functional requirements for the detection of an unauthorized rollback of contents stored in the external NVM.

Component levelling:



FDP_URC: Protection against unauthorized rollback of stored contents

1

FDP_URC.1 Requires the TOE to protect against an unauthorized rollback of the contents stored in the external NVM.

Management: FDP_URC.1

There are no management activities foreseen.

Audit: FDP_URC.1

There are no actions defined to be auditable.

5.1.1.2 Extended Components

Description:

Requires the TOE to protect against an unauthorized rollback of the contents stored in the external NVM.

Hierarchical to: No other components.

Definition:

FDP_URC.1 Protection against an unauthorized rollback of stored contents

FDP_URC.1.1 The TOE shall detect an unauthorized replacement of the contents stored in the external NVM before the contents are used. The detection shall take place even if the contents were previously stored in the same NVM and were valid and consistent at a given past time.

FDP_URC.1.2 Upon detection of unauthorized rollback of external NVM contents, the TOE shall [selection: *stop TOE operation, [assignment: other actions]*].

Dependencies: No dependencies.

5.1.2 FDP_IRA: Irreversibility Anchor of NVM contents

Note: This extended family comes from [5].

5.1.2.1 Description

To define the security functional requirements of the TOE we use an additional family (FDP_IRA) of the Class FDP (User data protection).

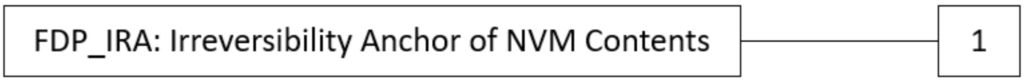
The family "Irreversibility Anchor of NVM contents (FDP_IRA)" is specified as follows.

FDP_IRA Irreversibility Anchor of NVM contents

Family Behaviour:

This family defines functional requirements for the implementation of a non-volatile mutable irreversibility anchor that goes through a series of predefined states in an irreversible way, i.e., without the possibility of going back to previous states. The irreversibility anchor value resulting from its state is linked to the data freshness of the external NVM contents. Violating data freshness property of the external NVM contents would result in a non-concordance of the value of the irreversibility anchor with the contents retrieved from the external NVM. Therefore, this mechanism serves to maintain the data freshness of the external NVM contents.

Component levelling:



FDP_IRA.1 Requires the TOE to implement a non-volatile mutable irreversibility anchor that goes through a series of predefined states in an irreversible way, i.e., without the possibility of going back to previous states.

Management: FDP_IRA.1

There are no management activities foreseen.

Audit: FDP_IRA.1

There are no actions defined to be auditable.

5.1.2.2 Extended Components

Description:

Requires the TOE to implement a non-volatile mutable irreversibility anchor that goes through a series of predefined states in an irreversible way, i.e., without the possibility of going back to previous states.

Hierarchical to: No other components.

Definition:

FDP_IRA.1 Irreversibility Anchor of NVM contents

FDP_IRA.1.1 The TOE shall implement a non-volatile irreversibility anchor mechanism that maintains a value that goes through a predefined sequence of states without the possibility of reversion to a previous state. The state of the Irreversibility Anchor is used to verify that NVM contents preserve data freshness as follows: [selection, choose one of:

- *Its value is (1) associated to the state of the external NVM contents, (2) advanced to the next state before contents are updated by the host MCU, and (3) checked to determine if the external NVM contents are fresh before they are used;*
- *Its value is (1) associated to sequences of commands or individual commands sent by the host MCU to the external NVM, (2) advanced to the next state before each command or sequence of commands are issued by the host MCU to the external NVM and (3) checked to determine if command contents are fresh when a command is issued;*
- *[assignment: other option]*].

[assignment: *indication of in which way the irreversibility anchor serves to determine that external NVM contents meet data freshness*].

Dependencies: No dependencies.

5.1.3 FPT_CRP: Protection against NVM cloning or replacement

Note: This extended family comes from [5].

5.1.3.1 Description

To define the security functional requirements of the TOE we use an additional family (FPT_CRP) of the Class FPT (Protection of the TSF).

The family "Protection against NVM cloning or replacement (FDP_CRP)" is specified as follows.

FPT_CRP Irreversibility Anchor of NVM contents

Family Behaviour:

This family describes the functional requirements for the detection of replacement or cloning of the NVM used for data and code storage. There are two main scenarios for this situation. First, the contents of the NVM from a TOE unit could be read and then written into the NVM of a second unit, constituting a clone operation, in an attempt to replace the user data and TSF data of a TOE unit with those from a different TOE unit. A second case consists of the physical replacement of the NVM of a TOE with the NVM of a different unit when it is physically feasible.

Component levelling:

FPT_CRP: Protection against NVM cloning or replacement
1

FPT_CRP.1 Requires the TOE to protect against cloning or replacement of the NVM contents.

Management: FPT_CRP.1

There are no management activities foreseen.

Audit: FPT_CRP.1

There are no actions defined to be auditable.

5.1.3.2 Extended Components

EXTENDED COMPONENT FPT_CRP.1

Description:

Requires the TOE to protect against cloning or replacement of the NVM contents.

Hierarchical to: No other components.

Definition:

FPT_CRP.1 Protection against NVM cloning or replacement

FPT_CRP.1.1 The TOE protection shall prevent a situation where the contents in the external NVM have been cloned from another external NVM or where the external NVM memory has been physically replaced with another external NVM.

Dependencies: No dependencies.

5.1.4 FMT_LIM – Limited capabilities and availability

Note: This extended family comes from [6].

5.1.4.1 Description

To define te 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 6.1) appropriate to address the specific issues of preventing the abuse of functions by limiting the capabilities of the functions and by limiting their availability.

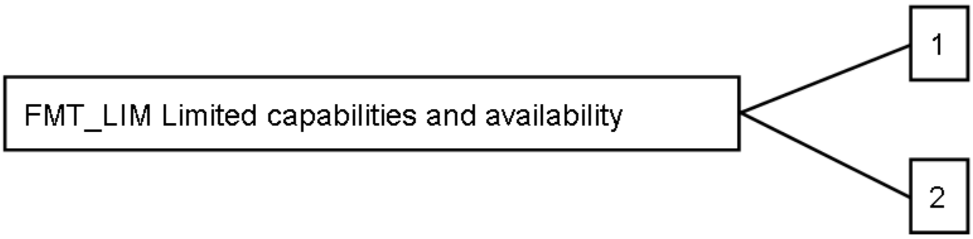
The family “Limited Capabilities and Availability (FMT_LIM)” is specified as follows:

FMT_LIM Limited Capabilities and Availability

Family Behavior:

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.

5.1.4.2 Extended Components

EXTENDED COMPONENT FMT_LIM.1

Description:

Limited capabilities require 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, for example, 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.

5.1.5 FDP_SDC – Stored data confidentiality

Note: This extended family comes from [6].

5.1.5.1 Description

To define the security functional requirements of the TOE an additional family (FDP_SDC) 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 Behavior:

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.

5.1.5.2 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.

5.1.6 FAU_SAS – Audit data storage

Note: This extended family comes from [6].

5.1.6.1 Description

To define the security functional requirements of the TOE an additional family (FAU_SAS) of the Class FAU (Security Audit) is defined here.

The family “Audit data storage (FAU_SAS)” is specified as follows.

FAU_SAS AUDIT DATA STORAGE

Family Behavior:

This family describes the functional requirements for the storage of audit data. It has a more general approach than FAU_GEN, because it does not necessarily require the data to be generated by the TOE itself and because it does not give specific details of the content of the audit records.

Component Levelling:

FAU_SAS Audit data storage
1

FAU_SAS.1 Requires the TOE to provide the possibility to store audit data.

Management: FAU_SAS.1

There are no management activities foreseen.

Audit: FAU_SAS.1

There are no actions defined to be auditable.

5.1.6.2 Extended Components

EXTENDED COMPONENT FAU_SAS.1

Description:

Requires the TOE to provide the possibility to store audit data.

Hierarchical to: No other components.

Definition:

FAU_SAS.1 Audit storage

FAU_SAS.1.1 The TSF shall provide [assignment: *list of subjects*] with the capability to store [assignment: *list of audit information*] in the [assignment: *type of persistent memory*].

Dependencies: No dependencies.

5.1.7 FDP_INI - Host-Device Boot Protection

5.1.7.1 Description

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

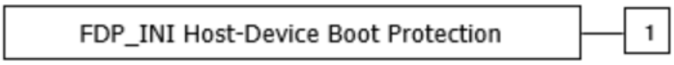
The family “Host-Device Boot Protection (FDP_INI)” is specified as follows.

FDP_INI Host-Device Boot Protection

Family Behavior:

This family describes the functional requirements for the TOE as it services the secure initialization of the U.Host-Device, namely the functional requirements for the protection of the Boot code stored in the TOE.

Component levelling:



FDP_INI.1 Requires the TOE to provide protection for the U.Host-Device boot code at U.Host-Device initialization at power-on.

Management: FDP_INI.1

There are no management activities foreseen.

Audit: FDP_INI.1

There are no actions defined to be auditable.

5.1.7.2 Extended Components

EXTENDED COMPONENT FPD_INI.1

Description:

Requires the TOE to provide protection for the U.Host-Device boot code at U.Host-Device initialization at power-on.

Hierarchical to: No other components

Definition:

FDP_INI.1 Host-Device Boot Protection

FDP_INI.1.1 The TOE initialization function shall verify U.Host-Device boot code integrity and authenticity prior to providing any access to it.

FDP_INI.1.2 The TOE shall detect and respond to a violation of the U.Host-Device boot code integrity and authenticity during initialization, such that the U.Host-Device either successfully completes initialization or is halted.

Dependencies: No dependencies.

5.1.8 FDP_TUD - Host-Device Trusted Updates

5.1.8.1 Description

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

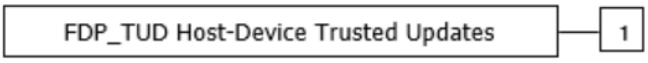
The family “Host-Device Trusted Updates (FDP_TUD)” is specified as follows.

FDP_TUD Host-Device Trusted Updates

Family Behaviour:

This family describes the functional requirements for the TOE as it services the secure update of the U.Host-Device code, namely the functional requirements for the protection of the U.Host-Device code update. Before the update is being installed, the updates must be verified to ensure the integrity and authenticity of the update and also that the updates are newer than the current running version. This is to prevent that manipulated or older updates, with known weaknesses, are being used.

Component levelling:



FDP_TUD.1 Requires the TOE to provide protection for the U.Host-Device code at the update process.

Management: FDP_TUD.1

There are no management activities foreseen.

Audit: FDP_TUD.1

There are no actions defined to be auditable.

5.1.8.2 Extended Components

EXTENDED COMPONENT FDP_TUD.1

Description:

Requires the TOE to provide protection for the U.Host-Device code at the update process.

Hierarchical to: No other components

Definition:

FDP_TUD.1 Host-Device Trusted Updates

FDP_TUD.1.1 The TSF shall provide the ability to update the U.Host-Device code.

FDP_TUD.1.2 The TSF shall verify the authentication and integrity of U.Host-Device code updates prior to storing those updates.

FDP_TUD.1.3 The TSF shall provide a means to verify software updates to the U.Host-Device to ensure that the software update version is newer than the current version of the U.Host-Device code prior to storing those updates.

Dependencies: No dependencies.

6 Security Requirements

6.1 Security Functional Requirements

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 an 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.

6.1.1 External NVM Augmentation Package Security Functional Requirements

6.1.1.1 Leakage

FPT_ITT.1/NVM Basic internal TSF data transfer protection

FPT_ITT.1.1/NVM 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, modification*** when it is transmitted between separate parts of the TOE.

Application Note:

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

Application Note:

The Protection Profile [6] already includes the SFR FPT_ITT.1. However, its second assignment only contemplates disclosure of the TSF data when it is transmitted between other parts of the TOE. This particular SFR requires that the TSF data exchanged between other parts of the TOE is protected both from disclosure and modification.

FDP_ITT.1/NVM Basic internal transfer protection

FDP_ITT.1.1/NVM 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 other parts of the TOE and the external NVM.

The TSF shall enforce the **Data Processing Policy** to prevent the ***disclosure, modification*** of user data when it is transmitted between other parts of the TOE and the external NVM.

Application Note:

The Flash array and the QSF Logic are seen as physically-separated parts of the TOE. The external NVM in [5] is the TOE in this ST.

Application Note:

The Protection Profile [6] already includes the SFR FDT_ITT.1. However, its second assignment only contemplates disclosure of the user data when it is transmitted between other parts of the TOE. This particular SFR requires that the user data exchanged between other parts of the TOE is protected both from disclosure and modification.

Application Note:

The Data Processing Policy defined in FDP_IFC.1 is applicable for FDP_ITT.1/NVM, and no other information flow control SFP needs to be defined for the case of the user data stored in the external NVM.

6.1.1.2 Protection of NVM content freshness

FDP_URC.1 Protection against an unauthorized rollback of stored contents

FDP_URC.1.1 The TOE shall detect an unauthorized replacement of the contents stored in the external NVM before the contents are used. The detection shall take place even if the contents were previously stored in the same NVM and were valid and consistent at a given past time.

FDP_URC.1.2 Upon detection of unauthorized rollback of external NVM contents, the TOE shall [selection: *stop TOE operation, [assignment: other actions]*]

Upon detection of unauthorized rollback of external NVM contents, the TOE shall prevent the unauthorized rollback and signals the violation in the status register.

Application Note:

The external NVM in [5] is the full TOE in this ST.

6.1.1.3 Data transfer between host MCU and external NVM

FPT_RPL.1 Replay detection

FPT_RPL.1.1 The TSF shall detect replay for the following entities: [assignment: list of identified entities].

The TSF shall detect replay for the following entities: commands issued by the host MCU to the external NVM for the read, write and erase operations.

FPT_RPL.1.2 The TSF shall perform [assignment: *list of specific actions*] when a replay is detected.

The TSF shall perform preventing the command execution and signaling the violation in the status register when a replay is detected.

Application Note:

This SFR applies in this TOE to the protected commands, i.e., commands that are signed and therefore are replay-protected.

Application Note:

The host MCU in [5] is the user of the TOE (U.Host-Device or U.Remote-Operator) in this ST. The external NVM in [5] is the full TOE in this ST.

6.1.1.4 Host-Device Boot Protection

FDP_IRA.1 Irreversibility Anchor of NVM contents

FDP_IRA.1.1 The TOE shall implement a non-volatile irreversibility anchor mechanism that maintains a value that goes increasingly through a predefined sequence of states without the possibility of reversion to a previous state. The state of the Irreversibility Anchor is used to verify that NVM contents preserve data freshness as follows: [selection, choose one of:

- *Its value is (1) associated to the state of the external NVM contents, (2) advanced to the next state before contents are updated by the host MCU, and (3) checked to determine if the external NVM contents are fresh before they are used;*
- *Its value is (1) associated to sequences of commands or individual commands sent by the host MCU to the external NVM, (2) advanced to the next state before each command or sequence of commands are issued by the host MCU to the external NVM and (3) checked to determine if command contents are fresh when a command is issued;*
- *[assignment: other option]*].

[assignment: *indication of in which way the irreversibility anchor serves to determine that external NVM contents meet data freshness*].

The TOE shall implement a non-volatile irreversibility anchor mechanism that maintains a value that goes increasingly through a predefined sequence of states without the possibility of reversion to a previous state. The state of the Irreversibility Anchor is used to verify that NVM contents preserve data freshness as follows:

Its value is (1) associated to sequences of commands or individual commands sent by the host MCU to the external NVM, (2) advanced to the next state before each command or sequence of commands are issued by the host MCU to the external NVM and (3) checked to determine if command contents are fresh when a command is issued

Application Note:

The signature of the commands depend on the anchor value so that using an outdated value creates an invalid signature.

Application Note:

The NVM in [5] is the full TOE in this ST.

FPT_CRP.1 Protection against NVM cloning or replacement

FPT_CRP.1.1 The TOE protection shall prevent a situation where the contents in the external NVM have been cloned from another external NVM or where the external NVM memory has been physically replaced with another external NVM.

Application Note:

The external NVM in [5] is the full TOE in this ST.

6.1.2 PP0084 Security Functional Requirements

6.1.2.1 Abuse of Functionality

Application Note:

The following Security Function Policy (SFP) **Test Mode Security Policy** is defined for the requirements "Limited capabilities (FMT_LIM.1)" and “Limited availability (FMT_LIM.2)”:

Deploying Test Features after TOE Delivery does not allow

- Any Protected User data to be disclosed or manipulated,
- Any TSF data to be disclosed or manipulated, and
- No substantial information about construction of TSF to be gathered which may enable other attacks.

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: **the Test Mode Security Policy**.

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: **the Test Mode Security Policy.**

6.1.2.2 Physical Manipulation

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 **Memory Sections defined as confidentiality protected**.

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: **Memory Sections defined as integrity protected**.

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.**

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 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. 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.

6.1.2.3 Leakage

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 **TSF data and Protected User data that is processed or transferred by the TOE**.

Application Note:

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

- Confidentiality protected TSF data (see Section 3.2.2) shall not be accessible from the TOE in any case, including the Test Mode.
- Secret User data (see Section 3.2.3) shall not be accessible from the TOE except when the U.Host-Device or the U.Remote-Operator decide to communicate the secret User data via the external interface through a secure channel.

- Authenticated User data (see Section 3.2.3) shall not be received or modified by the TOE except when the U.Host-Device or the U.Remote-Operator provide a properly authenticated write command.

6.1.2.4 Identification

FAU_SAS.1 Audit storage

FAU_SAS.1.1 The TSF shall provide [assignment: list of subjects] with the capability to store [assignment: list of audit information] in the [assignment: type of persistent memory].

The TSF shall provide **the manufacturer** with the capability to store **Winbond Device ID and Secure Unique Device ID** in the **dedicated portion of the Flash Array**.

6.1.3 Other Security Functional Requirements

6.1.3.1 Host-Device Boot Protection

FDP_INI.1 Host-Device Boot Protection

FDP_INI.1.1 The TOE initialization function shall verify [assignment: list of verifications] prior to providing any access to it.

The TOE initialization function shall verify **U.Host-Device boot code integrity and authenticity** prior to providing any access to it.

FDP_INI.1.2 The TOE shall detect and respond to a violation of the U.Host-Device boot code integrity and authenticity during initialization such that the U.Host-Device either successfully completes initialization or is halted.

6.1.3.2 Trusted Update

FDP_TUD.1 Host-Device Trusted Updates

FDP_TUD.1.1 The TSF shall provide the ability to update the U.Host-Device code.

FDP_TUD.1.2 The TSF shall verify the authentication and integrity of U.Host-Device code updates prior to storing those updates.

FDP_TUD.1.3 The TSF shall provide a means to verify software updates to the U.Host-Device to ensure that the software update version is newer than the current version of the U.Host-Device code prior to storing those updates.

6.1.3.3 Memory Access Control

FDP_ACC.2 Complete access control

FDP_ACC.2.1 The TSF shall enforce the [assignment: *access control SFP*] on [assignment: *list of subjects and objects*] and all operations among subjects and objects covered by the SFP.

The TSF shall enforce the **Memory access control SFP** on **U.Host-Device and U.Remote-Operator accessing memory sections** and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2 The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

FDP_ACF.1 Security attribute based access control

FDP_ACF.1.1 The TSF shall enforce the [assignment: access control SFP] to objects based on the following: [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].

The TSF shall enforce the **Memory access control SFP** to objects based on the following:

- **Subjects: U.Host-Device and U.Remote-Operator**
- **Objects: memory sections**
- **Attributes of the memory sections: Integrity Protection, Write Protection, Rollback Protection, Plain Access Read, Plain Access Write and Plain Access Authenticated.**

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects].

The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **U.Host-Device and U.Remote-Operator are allowed to access a memory region according to the access rights determined by the memory section attributes**.

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [assignment: rules, based on security attributes that explicitly authorize access of subjects to objects].

The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **U.Host-Device and U.Remote-Operator are allowed to access a memory region if the memory section access right attributes allow it**.

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [assignment: rules, based on security attributes, that explicitly deny access of subjects to objects].

The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **U.Host-Device and U.Remote-Operator are denied to access a memory region if the memory section access right attributes deny it**.

FMT_MSA.3 Static attribute initialization

FMT_MSA.3.1 The TSF shall enforce the [assignment: *access control SFP, information flow control SFP*] to provide [selection, choose one of: *restrictive, permissive, [assignment: other property]*] default values for security attributes that are used to enforce the SFP.

The TSF shall enforce the **Memory access control SFP** to provide ***restrictive*** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the [assignment: *the authorised identified roles*] to specify alternative initial values to override the default values when an object or information is created.

The TSF shall allow the **U.Remote-Operator** to specify alternative initial values to override the default values when an object or information is created.

FMT_MSA.1 Management of security attributes

FMT_MSA.1.1 The TSF shall enforce the [assignment: access control SFP(s), information flow control SFP(s)] to restrict the ability to [selection: change_default, query, modify, delete, [assignment: other operations]] the security attributes [assignment: list of security attributes] to [assignment: the authorised identified roles].

The TSF shall enforce the **Memory access control SFP** to restrict the ability to **modify** the security attributes **of the memory sections** to **U.Remote-Operator.**

6.1.3.4 Key Provisioning & Secure Communication

FTP_ITC.1 Inter-TSF trusted channel

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

Application Note:

The trusted channel is used for key provisioning and transmitting/receiving secure commands.

FTP_ITC.1.2 The TSF shall permit [selection: *the TSF, another trusted IT product*] to initiate communication via the trusted channel.

The TSF shall permit ***another trusted IT product*** to initiate communication via the trusted channel.

Application Note:

The other trusted IT product hereby is either U.Host-Device or U.Remote-Operator.

6.2 Security Assurance Requirements

The Evaluation Assurance Level is EAL2+ALC_FLR.2.

6.3 Security Requirements Rationale

6.3.1 Security Objectives for the TOE

O.Abuse-Func: Protection against Abuse of Functionality.

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/NVM, FDP_ITT.1/NVM, FDP_IFC.1 and FPT_PHP.3) 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-Manipulation, O.Leak-Forced.

O.Leak-Inherent: Protection against Inherent Information Leakage.

The refinements of the security functional requirements FPT_ITT.1/NVM and FDP_ITT.1/NVM 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: Protection against Forced Information Leakage.

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 behavior of the TOE (either by exposing it to extreme operating conditions or by directly manipulating it) with a second attack step measuring and analyzing some output produced by the TOE. The first step is prevented by the O.Phys-Manipulation (FPT_PHP.3), respectively. The requirements covering O.Leak-Inherent (FPT_ITT.1/NVM, FDP_ITT.1/NVM and FDP_IFC.1) also support O.Leak-Forced because they prevent the attacker from being successful if he tries the second step directly.

O.NVM-Interface-Protection: Protection against abuse of the interface between host MCU and external NVM.

The TOE (FPT_ITT.1/NVM, FDP_ITT.1/NVM, FTP_ITC.1) shall protect against abusing the interface:

By enforce secure communication for the access to the Protected User Data and

By protecting the secure communication against *Secret User Data* disclosure, and *Authenticated User Data* illegal modification

Since the SPI bus between the *U.Host-Device* and the TOE is an easy subject to attacks, it is required that the TOE prevents disclosure of data in transit through it, and it is required that the TOE detects manipulation of such data in transit. An attacker issuing new commands for modifying the data in the NVM must be detected by the TOE as well.

O.NVM-Irreversibility-Anchor: External NVM Contents Irreversibility Anchor.

The TOE (FDP_IRA.1)shall implement a non-volatile mutable mechanism that goes through a predefined sequence of states (that are associated with increasing 'values') that can never be returned to a previous state. This value given by a sequence of states shall be used to determine whether the Authenticated User Data meets the data freshness property and to prevent replay attacks.

O.NVM-Command-Replay-Protection: Protection against replay of commands between host MCU and external NVM.

The TOE (FPT_RPL.1) shall protect against replay of the read, write, erase, and security-related configuration commands issued within the secure communication through the interconnection bus.

O.NVM-Unauthorized-Rollback-Protection: Protection against an unauthorized rollback of NVM contents.

The TOE (FDP_URC.1) shall protect against replacing Authenticated User Data with a previous version, even if it was valid in the past.

O.NVM-Clone-Replace-Protection: Protection against NVM cloning or replacement.

The TOE (FPT_CRP.1) shall protect against cloning the Protected User Data of another unit into the TOE's NVM and against replacement of the TOE with a different NVM device. While the contents are valid for a TOE from where they were extracted, they shall be detected as non-belonging to the TOE unit where they were cloned to and, thus, non-valid.

O.Identification: TOE Identification

The TOE (FAU_SAS.1) must provide means to store a unique identification for the TOE. The unique identifier is protected against modification and it is used for TOE identification. In addition, the TOE identifier can be used by the U.Host-Device for RoT initialization during the Secure Boot.

O.Secure-Boot: Protection against Boot code corruption.

The TOE (FDP_INI.1) shall protect the boot process of the U.Host-Device against corruption of the boot code stored on the TOE.

O.External-Content-Protection: Protection against disclosure and undetected modification of external NVM contents.

The SFR FDP_SDC.1 ensures protection of confidentiality of the contents stored in the external NVM, while the SFR FDP_SDI.1 ensures protection of the integrity of the contents stored in the NVM. Therefore, it is clear that these security functional requirements support the objective.

O.Secure-Update: Protection of the Host-Device code update.

The TOE (FDP_TUD.1) shall support the trusted update protecting the confidentiality, integrity, and freshness of the transmitted data.

O.Access-Policy: Memory Access Policy.

This objective states that there is an access control policy which implements different levels of authorization in order to provide access to each memory section. The TOE (FDP_ACC.2, FDP_ACF.1, FMT_MSA.3, FMT_MSA.1) implements this access control policy.

O.Key-Provisioning: Secure Key provisioning.

The SFR FTP_ITC.1 ensures the establishment of a trusted path for key provisioning.

O.Phys-Manipulation: Protection against Physical 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 failure 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.

6.3.2 Rationale tables of Security Objectives and SFRs

SECURITY OBJECTIVES	SECURITY FUNCTIONAL REQUIREMENTS
O.Abuse-Func	FMT_LIM.1, FMT_LIM.2, FPT_ITT.1/NVM,FDP_ITT.1/NVM, FDP_IFC.1, FPT_PHP.3
O.NVM-Command-Replay-Protection	FPT_RPL.1
O.NVM-Unauthorized-Rollback-Protection	FDP_URC.1
O.NVM-Irreversibility-Anchor	FDP_IRA.1
O.NVM-Clone-Replace-Protection	FPT_CRP.1
O.NVM-Interface-Protection	FPT_ITT.1/NVM, FDP_ITT.1/NVM, FTP_ITC.1 Note: In addition to the SFRs mapped in [5] to meet the security objective, it is necessary to add FTP_ITC.1 because the U.Host-Device is not TOE as in the [5] case.
O.External-Content-Protection	FDP_SDC.1 and FDP_SDI.2
O.Phys-Manipulation	FPT_PHP.3 and FDP_SDI.2
O.Leak-Inherent	FPT_ITT.1/NVM, FDP_ITT.1/NVM, FDP_IFC.1
O.Leak-Forced	FPT_PHP.3, FDP_IFC.1 FPT_ITT.1/NVM, FDP_ITT.1/NVM
O.Identification	FAU_SAS.1, FTP_ITC.1
O.Secure-Boot	FDP_INI.1
O.Secure-Update	FDP_TUD.1
O.Access-Policy	FDP_ACC.2, FDP_ACF.1, FMT_MSA.3, FMT_MSA.1
O.Key-Provisioning	FTP_ITC.1

Table 4 Security Objectives and SFRs - Coverage

SECURITY FUNCTIONAL REQUIREMENTS	SECURITY OBJECTIVES
FPT_ITT.1/NVM	O.NVM-Interface-Protection, O.Abuse-Func, O.Leak-Inherent, O.Leak-Forced
FDP_ITT.1/NVM	O.NVM-Interface-Protection, O.Abuse-Func, O.Leak-Inherent, O.Leak-Forced
FDP_URC.1	O.NVM-Unauthorized-Rollback-Protection
FPT_RPL.1	O.NVM-Command-Replay-Protection
FDP_IRA.1	O.NVM-Irreversibility-Anchor
FPT_CRP.1	O.NVM-Clone-Replace-Protection
FMT_LIM.1	O.Abuse-Func
FMT_LIM.2	O.Abuse-Func
FDP_SDC.1	O.External-Content-Protection
FDP_SDI.2	O.External-Content-Protection, O.Phys-Manipulation

SECURITY FUNCTIONAL REQUIREMENTS	SECURITY OBJECTIVES
FPT_PHP.3	O.Phys-Manipulation, O.Abuse-Func, O.Leak-Forced
FDP_IFC.1	O.Abuse-Func, O.Leak-Inherent, O.Leak-Forced
FAU_SAS.1	O.Identification
FDP_INI.1	O.Secure-Boot
FDP_TUD.1	O.Secure-Update
FDP_ACC.2	O.Access-Policy
FDP_ACF.1	O.Access-Policy
FMT_MSA.3	O.Access-Policy
FMT_MSA.1	O.Access-Policy
FTP_ITC.1	O.Key-Provisioning, O.NVM-Interface-Protection, O.Identification

Table 5 SFRs and Security Objectives

6.3.3 Dependencies

6.3.3.1 SFRs Dependencies

REQUIREMENTS	CC DEPENDENCIES	SATISFIED DEPENDENCIES
FPT_ITT.1/NVM	No Dependencies	
FDP_ITT.1/NVM	(FDP_ACC.1 or FDP_IFC.1)	FDP_IFC.1
FDP_URC.1	No Dependencies	
FPT_RPL.1	No Dependencies	
FDP_IRA.1	No Dependencies	
FPT_CRP.1	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	FDP_SDI.2	
FPT_PHP.3	No Dependencies	
FDP_IFC.1	(FDP_IFF.1)	Rationale
FAU_SAS.1	No Dependencies	
FDP_INI.1	No Dependencies	
FDP_TUD.1	No Dependencies	
FDP_ACC.2	FDP_ACF.1	FDP_ACF.1

REQUIREMENTS	CC DEPENDENCIES	SATISFIED DEPENDENCIES
FDP_ACF.1	FDP_ACC.1 FMT_MSA.3	FDP_ACC.2 FMT_MSA.3
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	FMT_MSA.1 Rationale
FMT_MSA.1	(FDP_ACC.1 or FDP_IFC.1) FMT_SMR.1 FMT_SMF.1	FDP_ACC.2 Rationale Rationale
FTP_ITC.1	No Dependencies	

Table 6 SFRs Dependencies

Rationale for the exclusion of Dependencies

The dependency FMT_SMR.1 of FMT_MSA.3 and FMT_SMR.1 and FMT_SMF.1 of FMT_MSA.1 is discarded. Part 2 of the Common Criteria defines the dependency. However, there is only one role (U.Remote-Operator) and the management functions associated to the access control policy are already included in FMT_MSA.3 and FMT_MSA.1. Therefore, it has been decided not to include these SFRs to avoid more complexity in the security target.

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).

6.3.3.2 SARs Dependencies

REQUIREMENTS	CC DEPENDENCIES	SATISFIED DEPENDENCIES
ADV_ARC.1	(ADV_FSP.1) and (ADV_TDS.1)	ADV_FSP.2, ADV_TDS.1
ADV_FSP.2	ADV_TDS.1	ADV_TDS.1
ADV_TDS.1	(ADV_FSP.2)	ADV_FSP.2
AGD_OPE.1	(ADV_FSP.1)	ADV_FSP.2
AGD_PRE.1	No Dependencies	
ALC_CMC.2	ALC_CMS.1	ALC_CMS.2
ALC_CMS.2	No Dependencies	
ALC_DEL.1	No Dependencies	
ALC_FLR.2	No Dependencies	
ASE_CCL.1	(ASE_ECD.1) and (ASE_INT.1) and (ASE_REQ.1)	ASE_ECD.1, ASE_INT.1, ASE_REQ.2
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.2, ASE_INT.1, ASE_REQ.2
ATE_COV.1	(ADV_FSP.2) and (ATE_FUN.1)	ADV_FSP.2, ATE_FUN.1
ATE_FUN.1	(ATE_COV.1)	ATE_COV.1
ATE_IND.2	(ADV_FSP.2) and (AGD_OPE.1) and (AGD_PRE.1) and (ATE_COV.1) and (ATE_FUN.1)	ADV_FSP.2, AGD_OPE.1, AGD_PRE.1, ATE_COV.1, ATE_FUN.1
AVA_VAN.2	(ADV_ARC.1) and (ADV_FSP.2) and (ADV_TDS.1) and (AGD_OPE.1) and (AGD_PRE.1)	ADV_ARC.1, ADV_FSP.2, ADV_TDS.1, AGD_OPE.1, AGD_PRE.1,

Table 7 SARs Dependencies

6.3.4 Rationale for the Security Assurance Requirements

These SARs have been chosen to meet the market needs of a Secure Flash with resistance to attacks performed by an attacker possessing Basic attack potential.

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

7.1.1 SF.SEC-MEM: Secure External Memory

SF.SEC-MEM protects the integrity, the confidentiality, the authenticity and the freshness of the data and code stored on the Memory Flash.

- The confidentiality of the **Secret User Data** is achieved by data encryption of the secure Read and Write commands – the only type of commands allowed to access the protected User Data
- The integrity and authenticity of the **Authenticated User Data** is achieved by signature verification of the secure Read and Write commands – the only type of commands allowed to access the protected User Data
- The data freshness applies in particular to the sealed memory section (i.e., a section locked from further modifications after verifying the section digest and the version flag). The TOE has an update mechanism that allows the U.Remote-Operator to seal a half of the section, to fill the other half with new content and then to perform an atomic swap of the two halves with a cryptographic verification of the signature of the new contents and the new version tag. The rollback-protection mechanism ensures that it is not possible to revert the memory to its previous state.

SF.SEC-MEM enforces FDP_SDC.1, FDP_SDI.2, FDP_IFC.1 and FDP_URC.1.

7.1.2 SF.MEM-ACC: Memory Access Control

SF.MEM-ACC protects the memory from illegal or unauthorized access to its contents. The access privileges allow two levels of user authorization, allowing different usage scenarios. Most commonly, the lower authorization level is dedicated to the host device that embeds the TOE and communicates with it through a SPI Bus. The higher level is usually dedicated to the remote operator that communicates with the TOE by means of the host device. The security attributes of each memory section control the allowed read mode (plain, encrypted, requiring user authentication) and the allowed write mode (plain, forbidden, or authenticated)

SF.MEM-ACC enforces FDP_ACC.2 and FDP_ACF.1 for the access control and FMT_MSA.1 and FMT_MSA.3 for the management of the security attributes of the memory sections.

7.1.3 SF.SEC-COM: Secure Communication

SF.SEC-COM protects the communication between the TOE and the user (U.Host-Device or U.Remote-Operator) against bus probing, modification, man-in-the-middle and replay attacks. In particular,

- A secure channel with fresh Session Key is established for each session. The secure channel setup provides mutual authorization of both the Memory Flash IC and the user.

- In order to avoid key repetition, the TOE implements counters like a non-volatile Session Counter and a Transaction Counter;
- The transmitted data (in both directions) and the command address are encrypted and signed.
  - The encryption key is generated for each transaction from the Session Key and the Transaction Counter to prevent replay attacks.
  - The signature is a calculated as a MAC tag with a combination of the Session Key and the Transaction Counter to prevent replay attacks.

SF.SEC-COM enforces the following requirements related to the secure communications: FDP_IFC.1, FTP_ITC.1, FPT_ITT.1/NVM and FDP_ITT.1/NVM. Moreover, the replay protection enforces FPT_RPL.1 and FDP_IRA.1.

7.1.4 SF.PHY-PRO: Physical protection

SF.PHY-PRO protects the TOE against physical manipulation. For this purpose, SF.PHY-PRO includes the following security mechanisms:

- The bus connecting the Flash array and the QSF Logic is hidden by the HW logic
- TOE has CRC32 protection of keys and registers. When violation is detected, access is blocked, key usage is prevented, status indication.

SF.PHY-PRO also protects the TOE against the inherent or intentional leak of the keys used in TOE operations by corresponding security mechanisms.

SF.PHY-PRO enforces the TOE resistance against physical attacks (FPT_PHP.3).

7.1.5 SF.OPE-MODE: Control of Operating Modes

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

Test Mode (TM) entry is protected in the following manner:

- When first entering TM, the entire Flash is erased (Main array and Aux array), including user data, configurations and keys, and device management data (Monotonic Counter, Winbond Unique ID (WID), etc).
- Test mode entry is disabled before the device is shipped. When re-entering TM (after it was previously disabled), the device is Formatted before switching to TM.
- This formatting is skipped if user sets the Fault Analysis Mode entry flag in a cryptographically protected user register. This flag should be set only after user has removed any sensitive information stored on the device.

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

7.1.6 SF.IDENTITY: Secure Identification

SF. IDENTITY ensures that the user can identify the TOE and, with proper authentication, the process can be carried out by a cryptographic challenge-response.

For the identification purposes, the TOE provides two ID fields that can be set at different stages of the TOE life cycle:

- Winbond Device ID – a factory-programmed 64-bit device identifier
- Secure Unique Device ID – a 128-bit device identifier that can be programmed after Manufacturing

SF.IDENTITY enforces FAU_SAS.1 for the ID storage, FTP_ITC.1 for the challenge-response protocol and provides FPT_CRP.1 against NVM cloning or replacement

7.1.7 SF.KEY-PROV: Key Provisioning

SF.KEY-PROV provides a way to provision the TOE keys in a non-secure environment, e.g., in the field.

After the Device Master Key (KD) is programmed to the device in a secure environment, one uses the dedicated Key Provisioning Keys, derived from KD to open a secure session to deliver the Per-Section Keys to the device.In this way, the key provisioning process protects the keys integrity, authenticity, and confidentiality.

SF.KEY-PROV enforces FTP_ITC.1 for the key provisioning process

7.1.8 SF.BOOT&UPDATE_PRO: U.Host-Device Boot &Update Protection

SF. BOOT&UPDATE_PRO is a service provided by the TOE to its host-device. It gives maximum protection to the boot code integrity, along with a mechanism for the secure code update by a remote operator to the Host itself.

To enable this service, the Boot code shall be stored in a sealed memory section with rollback protection. Namely, a half of the section is locked from further modifications after verifying the section digest and the version flag, and the other half can be filled with new content. Then the remote operator can issue an authenticated command that in atomic way

- First verifies the signature of the new contents and checks the new version tag
- Then performs a swap of the two halves so that the new content is now mapped to the address range of the old one.

The rollback-protection mechanism ensures that the swap is possible only if the next version tag is not less than the current one.

SF.BOOT&UPDATE_PRO enforces FDP_INI.1 and FDP_TUD.1

7.2 SFRs and TSS

7.2.1 SFRs and TSS – Tables

SECURITY FUNCTIONAL REQUIREMENTS	TOE SUMMARY SPECIFICATION
FPT_ITT.1/NVM	SF.SEC-COM
FDP_ITT.1/NVM	SF.SEC-COM
FDP_URC.1	SF.SEC-MEM
FPT_RPL.1	SF.SEC-COM

SECURITY FUNCTIONAL REQUIREMENTS	TOE SUMMARY SPECIFICATION
FDP_IRA.1	SF.SEC-COM
FPT_CRP.1	SF.IDENTITY
FMT_LIM.1	SF.OPE-MODE
FMT_LIM.2	SF.OPE-MODE
FDP_SDC.1	SF.SEC-MEM
FDP_SDI.2	SF.SEC-MEM
FPT_PHP.3	SF.PHY-PRO
FDP_IFC.1	SF.SEC-MEM, SF.SEC-COM
FAU_SAS.1	SF.IDENTITY
FDP_INI.1	SF.BOOT&UPDATE_PRO
FDP_TUD.1	SF.BOOT&UPDATE_PRO
FDP_ACC.2	SF.MEM-ACC
FDP_ACF.1	SF.MEM-ACC
FMT_MSA.3	SF.MEM-ACC
FMT_MSA.1	SF.MEM-ACC
FTP_ITC.1	SF.SEC-COM, SF.IDENTITY, SF.KEY-PROV

Table 8 SFRs and TSS - Coverage

TOE SUMMARY SPECIFICATION	SECURITY FUNCTIONAL REQUIREMENTS
SF.SEC-MEM	FDP_SDC.1, FDP_SDI.2, FDP_IFC.1, FDP_URC.1
SF.MEM-ACC	FDP_ACC.2, FDP_ACF.1, FMT_MSA.1, FMT_MSA.3
SF.SEC-COM	FDP_IFC.1, FTP_ITC.1, FPT_RPL.1, FDP_IRA.1, FPT_ITT.1/NVM, FDP_ITT.1/NVM
SF.PHY-PRO	FPT_PHP.3
SF.OPE-MODE	FMT_LIM.1, FMT_LIM.2
SF.IDENTITY	FAU_SAS.1, FTP_ITC.1, FPT_CRP.1
SF.KEY-PROV	FTP_ITC.1
SF.BOOT&UPDATE_PRO	FDP_INI.1, FDP_TUD.1

Table 9 TSS and SFRs - Coverage

8 Revisions

MODIFICATION	COMMENT
E	Derived from Security Target revision E
F	Update Project name to W77Q16/32

Table 10 History of Modifications

9 Glossary and Abbreviations

CC	Common Criteria
EAL	Evaluation Assurance Level
IT	Information Technology
PP	Protection Profile
SPI	Serial Peripheral Interface is a synchronous serial data link that operates in full duplex mode
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFI	TSF Interface
TSP	TOE Security Policy

10 References

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

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

[3]

[4] Security Evaluation Standard for IoT Platforms (SESIP), GlobalPlatform Technology, Version 1.0. GP_FST_070

[5] Security IC Platform Augmentation Package: External NVM Storage, version 1.3 Available Online: https://www.eurosmart.com/security-ic-platform-augmentation-package-external-nvm-storage/.

[6] Security IC Platform Protection Profile with Augmentation Packages, Developed by Inside Secure, Infineon Technologies AG, NXP Semiconductors Germany GmbH, STMicroelectronics; Registered and Certified by BSI, Version 1.0. BSI-CC-PP-0084-2014

[7] W77Q - Secure Serial NOR Flash Memory Data Sheet, ver A6, Winbond Technology Ltd

[8] W77Q16JW/W77Q32JW Operational User Guidance, ver C, Winbond Technology Ltd

[9] W77Q16JW/W77Q32JW Preparative Procedure, ver C, Winbond Technology Ltd

[10] W77Q - Secure Serial NOR Flash Memory Security Manual, Ver A7, Winbond Technology Ltd