EUROSMART
The Voice of the Digital Security Industry

Security IC Platform Augmentation Package: External NVM Storage

EUROSMART
The Voice of the Digital Security Industry

www.eurosmart.com @Eurosmart_EU @Eurosmart

Rue de la Science 14b | B-1040 Brussels | Belgium
Tel +32 2 880 36 00 | mail eurosmart@eurosmart.com

Table of Contents

1. Document Introduction

This chapter Document Introduction contains the following sections:

- Document Introduction (1.1)
- Document Overview (1.2)

1.1. Document Identification

Title:	Security IC Platform Augmentation Package: External NVM Storage
Version Number:	1.3
Provided by:	Eurosmart ITSC
Technical Editors:	Eurosmart ITSC

Based upon the:

[1] Security IC Platform Protection Profile with Augmentation Packages, Version 1.0. Developed by Inside Secure, Infineon Technologies AG, NXP Semiconductors Germany GmbH, STMicroelectronics, Registered and Certified by Bundesamt für Sicherheit in der Informationstechnik (BSI) under the reference BSI-CC-PP-0084-2014.

The Security IC Platform Augmentation Package: External NVM Storage has been built with the:

[2] Common Methodology for Information Technology Security Evaluation. Evaluation methodology. April 2017. Version 3.1. Revision 5.

[3] Common Criteria for Information Technology Security Evaluation. Part 1: Introduction and general model. April 2017. Version 3.1. Revision 5

[4] Common Criteria for Information Technology Security Evaluation. Part 2: Security functional components. April 2017. Version 3.1. Revision 5

[5] Common Criteria for Information Technology Security Evaluation. Part 3: Security assurance components. April 2017. Version 3.1. Revision 5

though it is not a Protection Profile but an augmentation for the Security IC Platform Protection Profile with Augmentation Packages (BSI-CC-PP-0084, Version 1.0, 13.01.2014).

The following documents have also been used for the elaboration of Appendix I:

[6] Protection of flash integrated circuits at delivery point, Application note ANSSI-CC-NOTE-14/1.0, v1.0, Date 26/07/2016.

[7] CCRA Requirements, "Assurance Continuity," version 2.1 June 2012.

[8] JIL, "Security requirements for post-delivery code loading," version 1.0 February 2016.

Disclaimer: The term “Package” has been used to define the External NVM Augmentation Package to maintain consistency with [1]. There this term has been chosen to describe security functionality as required by Common Criteria that is not mandatory to be included in a security target in conformance to the IC protection profile [1].

1.2. Document Overview

1.2.1. Contents of this document

This document defines a series of additional elements of the security problem definition, security objectives, and security functional requirements (SFRs), to support those architectures of Security ICs according to [1] that use an external NVM for storage. This augmentation package provides guidelines for the option where the external NVM is considered part of the TOE (and therefore part of the evaluation of the Security IC) and for the option where the external NVM is not part of the TOE.

Other IT products different from Security ICs conformant to BSI-CC-PP-0084-2014 [1] that could benefit from the use of an external NVM (e.g., other System-On-Chip products) are out of the scope of application of this document.

The Security IC Platform Protection Profile with Augmentation Packages (BSI-CC-PP-0084-2014; Version 1.0, 13.01.2014) [1] defines standard requirements for the TOE. In particular, according to [1] the TOE must meet the following security functions requirements:

Security Functional Requirement	Dependencies
FRU_FLT.2	FPT_FLS.1
FPT_FLS.1	None
FMT_LIM.1	FMT_LIM.2
FMT_LIM.2	FMT_LIM.1
FAU_SAS.1	None
FPT_PHP.3	None
FDP_ITT.1	FDP_ACC.1 or FDP_IFC.1
FDP_IFC.1	FDP_IFF.1
FPT_IFC.1	FDP_IFF.1
FDP_ITT.1	None
FDP_SDC.1	None
FDP_SDI.2	None
FCS_RNG.1	None

Table 1 Security Functional Requirements of BSI-CC-PP-0084-2014

The security functional requirements in Table 1 are common to all TOEs declaring conformance to BSI-CC-PP-0084-2014 in their security targets. Among those, specifically, FDP_SDC.1 and FDP_SDI.2 involve the security of user data stored in the TOE non-volatile memories, as per confidentiality and integrity, respectively. However, modern approaches and market needs present frequent cases where the Security IC Platform uses an external non-volatile memory storage media for the persistence of data. An external NVM refers to a storage component that is not a physical part of the same physical chip as the Security IC platform. This storage component is usually operated by the security IC through the interconnection bus between the host MCU (the Security IC) and the external NVM.

This kind of architecture brings up new security concerns in terms of protection of the code and data, which is transferred between the external NVM and the host MCU, including potentially through the interconnection bus. On the one hand, confidentiality (FDP_SDC.1) and integrity (FDP_SDI.1) of the stored user data can be potentially compromised by an attacker having physical access to the external NVM. On the other hand, an attacker can be able to replace the contents of the external NVM with a previous copy. The scenario of a Security IC Platform using an external NVM device must address such security challenges.

Therefore, it is required to review the security paradigm provided in [1] and refine it to cover the security needs derived from using an external non-volatile storage device. Such refinement of the security problem is done considering that the contents stored in the external NVM can be either user data, code, or both. This document defines an augmentation package to the BSI-CC-PP-0084-2014 protection profile. This augmentation package has been elaborated following the methodology described in [2].

1.2.2. Usage of this document

A specific Security Target shall comprise:

- All the requirements stated in the Security IC Platform Protection Profile (BSI-CC-PP-0084-2014; Version 1.0, 13.01.2014) [1] (compliance),
- A definition of the TOE scope and its perimeter depending on the external NVM architecture used,
- The additional security functional requirements from this document Security IC Platform Augmentation Package: External NVM Storage, and
- Additional requirements from other sources, if relevant.

For all additional requirements (especially those from other sources), it must be shown in the Security Target that these requirements do not conflict with those described in the Protection Profile [1]. Therefore, this document contains an augmentation, which is intended to be copied into a specific Security Target as appropriate. This augmentation has been written according to Common Criteria rules and, therefore, can be easily included in a Security Target to address additional security aspects.

Note that the complete Security Target must be evaluated according to Common Criteria.

The additional security functionality defined in this document (Security IC Platform Augmentation Package: External NVM Storage) is organized in chapters, each according to one of the sections of the Protection Profile [1], so that each chapter contains the additions and considerations to the corresponding section.

Additional guidance is given in these chapters below in paragraphs, which are marked with “Application Note” following a number. All other text in the chapters below (numbered paragraphs as this one) can directly be copied into the Security Target. The names of the target chapters or sections are indicated in the headings in this document.

Many of the sections in this document are intentionally left empty. The headings are still included for the sake of better orientation. An empty section means that no addition must be made to the text given in the Protection Profile due to the additional functionality being considered.

Note that the “application notes” provide guidance and are not intended to be copied but shall be considered while producing the Security Target to ensure that it is consistent and complete.

It also must be noted that the contents of Appendix I: are informative and **shall not** be included in the Security Target.

1.2.3. ST Specific information

The contents of the section PP Introduction (Section 1 in the Security IC Platform Protection Profile with Augmentation Packages Version 1.0, 13.01.2014 [1]) is used to create the Introduction in the Security Target. The author of the Security Target may add this information, primarily to section 1.2.1 of [1].

However, the TOE specific information should mostly be added to chapter 2 of the Security Target (TOE description) and especially to the text provided in section 1.2.2 (TOE Definition) of [1].

In addition, the Security Target with conformance to [1] should provide the following information:

- The Security Target must explicitly state whether (i) TOE Delivery is after Phase 3 only or (ii) after Phase 4 as well.
- If the TOE provides functionality to be used after TOE Delivery, this is part of the IC Dedicated Support Software. Then such functions must be specified in the Security Target of the actual TOE to make clear if the TOE comprises IC Dedicated Support Software (e.g., a loader for the Flash Memory).
- The Security Target must identify the configurations or versions of the TOE. If, for instance, a specific function (derived from an organizational security policy) is provided by some IC Dedicated Support Software, which is not necessarily part of the final product, these options or different configurations or versions must be identified.

Therefore, a Security Target can cover multiple versions or configurations of the TOE. The Security Target must identify and clearly describe the differences between the versions or configurations in the TOE description and take into account these differences throughout the document, especially where the security functional requirements and the security functions are defined. The rationale must address the dependencies and the aspect of mutual support.

The Chapter TOE Summary Specification allows organizing and specifying the IT security functions in a way that makes the TOE security functionality easier for a reader of the Security Target to understand, as compared with reading the security functional requirements.

2. TOE Overview (PP Ch. 1.2)

2.1. Introduction (PP Ch. 1.2.1)

2.2. TOE Definition (PP Ch. 1.2.2)

The TOE uses an external non-volatile memory (NVM) for the storage of data and code. This storage is considered as external because the NVM is not embedded in the same physical chip as the Security IC. Due to this disposition, an interconnection channel exists between the external NVM and the rest of the TOE elements. This channel is typically a bus used for signals related to write, erase and read operations that are sent to the external NVM for its operation.

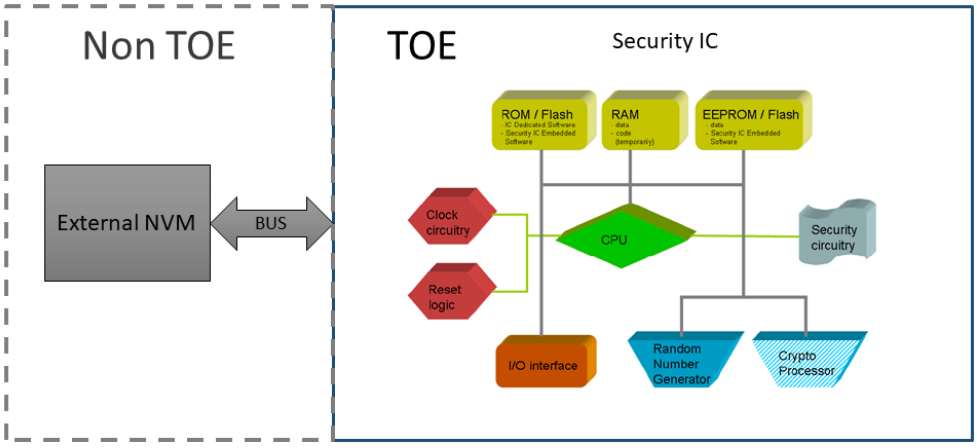
The external NVM can be used to store data, code, or both. A particular implementation could differ on the treatment given to data and code stored in the external NVM. The differences are mostly on the system level, and the definition of the security problem, security objectives and security requirements is the same for both.

Unlike the embedded NVMs, the external NVM is considered more accessible for attackers in terms of providing the possibility of reading, writing, or erasing the stored information. Besides, the interconnection channel between the external NVM and the Security IC embedded chip is subject to other threats such as eavesdropping, content blocking or replay. This situation brings up more challenges for the protection of the code and data stored in the external NVM relative to the protection of code and data stored in internal NVMs.

The TOE can use different types of non-volatile memories, involving various technologies. The main factor to consider is whether the external NVM is a “passive” NVM that passively receives commands from the Security IC for data operations, or whether the external NVM also includes TOE Security Functionality. In architectures where an external passive NVM is used, no TSF is implemented by the external NVM. Other types of external memories implement some TSF, such as protection of data stored in the external NVM or data in transit between the external NVM and the Security IC.

According to the above criteria, two possible variations of the TOE scope are defined:

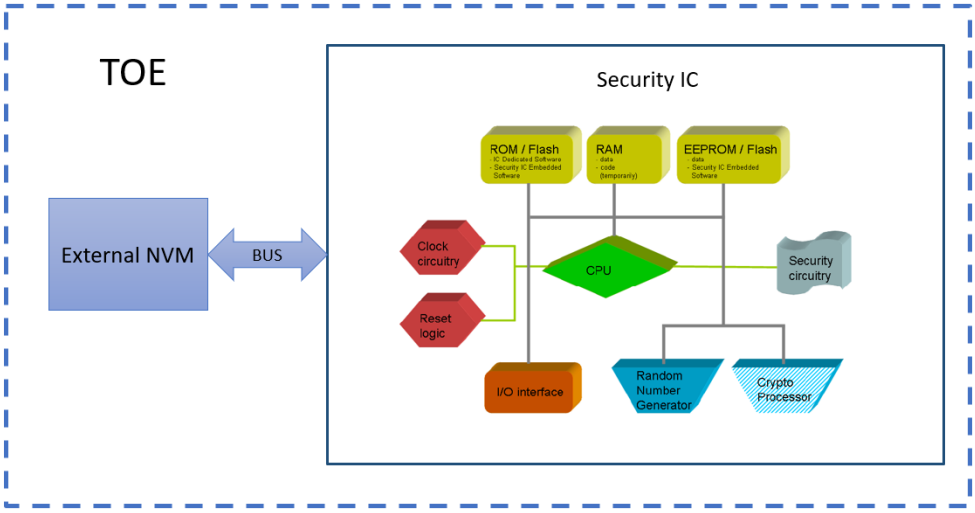
- **Option PM:** If the external NVM does not implement any TSF (e.g., passive NVM), it can be left out of the scope of the TOE – then it is considered a part of the operational environment. In this case, all the TOE security functionality is entirely implemented by the host MCU. This option is depicted in the figure below:



- **Option SM:** If the external NVM implements any security functionality for protection of the code and data stored in it, (while at rest in the NVM or while in transit between the host MCU and the NVM), then the external NVM and the communication bus are part of the TOE and are in the scope of the evaluation.

  Secure memory that is certified at AVA_VAN.5, can to some extent be interpreted as second Secure Element (with limited set of features).

  This case is depicted in the figure below:



Application Note 1: *The ST author must explicitly state whether the TOE architecture and scope follow the variant defined by Option PM (Non-TSF external NVM) or that defined by Option SM (NVM implementing TSF and in-scope of the TOE), as described above.*

As an informative note, "PM" stands for "Passive Memory", while "SM" stands for "Secure Memory".

Application Note 2: *This augmentation package supports the scenario of a host MCU using more than one external NVM for the storage of data and code. The definition of the security problem, security objectives, and security functional requirements provided here applies to the usage of each of the external NVMs that the host MCU may include.*

It is also possible to have different external NVMs, combining those implementing TSF (SM option) and those not implementing TSF (PM option). In such a case, the ST author shall refine the architecture diagrams provided here to reflect the architecture used. In the diagram and the associated explanation, it shall be evident that the external NVMs implementing TSF (according to option SM) are part of the TOE.

When multiple external NVMs are used for storage of data and code, there is an impact on the security functional requirements described in section 6.1. The ST author shall follow the guidelines provided in Application Note 21 for addressing that impact.

Concept of Data Freshness

The physical protection of the Security IC does not cover the external NVM or the interconnection bus between Security IC and the external NVM. If the only physical protection available is the packaging, it is not enough to mitigate threats requiring physical access to the external NVM. This situation enables attackers to have direct physical access to the external NVM, granting them the ability to read or write the contents of the memory. Also, the interconnection bus between the external NVM and the host MCU can be subject to eavesdropping, command blocking and replay threats.

Even when confidentiality and integrity of the contents stored in the external NVM are ensured, a new scenario of threat exists for this kind of architecture. The contents stored in the external NVM could be read, stored, and later written back to the external NVM. This situation opens the possibility of an unauthorized rollback of the contents in the external NVM to a previous version. The same effect could be achieved by intercepting communications passing across the interconnection bus between the external NVM and the host MCU and replaying the replies to previous read commands. Although the contents replayed or written to the external NVM were valid at a given moment in the past, this attack prevents the TOE from obtaining or updating the latest or "fresh" version of the contents in the external NVM.

This document defines a new dimension in which the security of the data must be preserved: *data freshness*, meaning that the stored data is always the one resulting in the last change carried out by the Security IC on the external NVM.

An attack consisting on replacing the contents in the external NVM by a previous version (e.g., cloning at a given time), which would result in writing into the external NVM data or code that preserves its confidentiality, integrity, and authenticity, would however violate the *"freshness"* of the data.

The adverse effect of data freshness violation can be seen as data or code unauthorized rollback, considering that the external NVM can be used for either storing data or code.

Hence, the data stored in the external NVM must also be protected in terms of data freshness.

Application Note 3: *This augmentation package contemplates the protection of the data freshness for the data and code stored in the external NVM. It must be noted that the augmentation package does not include the protection of data freshness for data stored in other memories, such as the Security IC internal memories.*

Examples of supported architectures

Some of the typical architectures that a TOE equipped with external NVM may present are listed below, according to Option PM or Option SM defined in Section 2.2. However, possible architectures are not limited to these:

- Option PM: Typically, this architecture implies a passive external NVM, such as flash memory, that does not implement any security functionality. The Security IC is entirely in charge of protecting the confidentiality and integrity of the contents stored in the external NVM or in transit through the interconnection bus, as well as protecting against replay attacks.
- Option SM: Typically, this architecture relies on a secure external NVM, implementing part or all of the security functions required for the protection of the code and data stored and in transit through the interconnection bus. E.g. Secure Flash memory, with a secure trusted channel between the host MCU and the memory. This kind of memory typically features replay protection and integrated confidentiality and integrity protection of stored code and data (implemented in the external NVM).

Application Note 4: *The Security Target must provide details on the architecture used, explaining at a high level how the Security IC is interconnected to the external NVM and whether the external NVM implements any TOE Security Functionality. The TOE Description section of the Security Target must provide the necessary information for clarifying the above topic.*

Confidentiality and integrity protection of contents stored in the external NVM

The Protection Profile includes SFRs aimed to protect the confidentiality and integrity of the data stored in the TOE non-volatile memories: FDP_SDC.1 and FDP_SDI.2. Depending on the TOE architecture and the technology of the external NVM, there are various possibilities for the protection of the stored code and data:

- Confidentiality and integrity protection of the contents stored in the NVM is entirely implemented by the Security IC.
- The NVM is responsible for the implementation of confidentiality or integrity protection of the contents stored in it.

Protection against cloning, replacement, replay and rollback

The Protection Profile includes SFRs aimed to protect the contents stored in the external NVM against cloning, replacement, rollback and command replay. Depending on the TOE architecture and the technology of the external NVM, there are various possibilities to implement such protection, either in the Security IC or in the secure NVM, or through a combination of both.

2.3. TOE life cycle (PP Ch. 1.2.3)

Application Note 5: *The external NVM is not a part of the same embedded chip as the Security IC. Thus the most common case would be that it is not be integrated at phase 3.*

TSF-relevant actions involving the NVM can occur at different stages of the life-cycle. E.g., non-personalized Firmware could be written to the NVM, or the NVM may be programmed with pre-personalization or personalization data, etc.

The stage at which the integration of the NVM with the IC takes place and the TSF-relevant processes that comprise such integration ***shall be defined*** *in the Security Target and will be part of the scope of the ALC activities.*

In the case of an architecture with an NVM and bus that are not in-scope of the TOE (PM option), life-cycle actions may be relevant even when no TSF is provided by the external NVM. For example, TSF data or code can be inserted into the external NVM in some phase previous to operational usage. The ALC assurance activities shall also cover that example scenario to assess that the process is carried out securely, even though the passive external NVM provides only storage of that data and code, without any TSF.

Application Note 6: *If external NVM implements TOE Security Functionality, all the sites where the external NVM is developed, produced, and integrated must be listed in the Security Target's TOE life-cycle section. Moreover, in such cases, those sites must be covered by a valid STAR report.*

If the external NVM does not implement any TSF, the above statement is not applicable.

2.4. Life-Cycle versus Scope and Organization of this Protection Profile (PP Ch. 1.2.4)

2.5. Specific Issues of Security IC Hardware and the Common Criteria (PP Ch. 1.2.5)

3. Security Problem Definition (PP Ch. 3)

3.1. Description of Assets (PP Ch. 3.1)

Application Note 7: *The following explanatory paragraphs shall be added at the end of section 3.1 of [1]*

When the TOE uses an external NVM for code and data storage, there is the following security concern related to data freshness:

SC.NVM-Content-Freshness Freshness of contents stored in the NVM

In order to ensure that the TOE is always able to retrieve the external NVM content according to the last change issued by the Security IC, the user data and TSF data stored in the external NVM must be protected in terms of data freshness.

When the TOE uses an external NVM for code and data storage, there is the following security concern related to binding the contents stored externally to the original TOE.

SC.NVM-Content-Binding : Binding of contents stored in the external NVM

In order to ensure that the contents stored in external NVM is bound to the original TOE, the user data and TSF data stored in the external NVM must be protected against cloning, replacement and rollback.

3.2. Threats (PP Ch. 3.2)

3.2.1. Threats common to architectures with PM and SM options

Application Note 8: *This Section lists the security threats common to architectures with either PM or SM option as defined in Section 2.2.*

In case of PM option, attacks against the external NVM are not attacks against the TOE, but rather against externalized TOE contents. However, the content of the NVM must be protected by the TOE.

The TOE shall avert the threat “Unauthorized abuse of NVM content (T.External-Content-Abuse)”, as specified below.

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.

An attacker may obtain direct access to the external NVM and attempt to access the contents stored to disclose, modify or replace it by its own contents.

The TOE shall avert the threat “Replay of commands between the host MCU and the external NVM (T.NVM-Command-Replay)” as specified below.

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.

The read, write and erase commands are issued by the host MCU, who performs such operations to exercise the storage functionality of the external NVM. The commands from the host MCU to the external NVM are issued through the interconnection bus. Those commands and their payloads can be intercepted by an attacker. Such attacker may replay the commands sent to the external NVM in different forms:

- Read commands: the host MCU issues a read command, and the attacker replies with a previously recorded answer to a previous read request. The host MCU gets old versions of such contents.
- Write commands: the attacker impersonates the host MCU and issues previous write commands, trying to overwrite the external NVM with the previous content, leading to the host MCU obtaining old versions of such contents in a later read operation.
- Erase command: similar to write command cases.

The TOE shall avert the threat "Unauthorized rollback of NVM contents (T.NVM-Unauthorized-Rollback)", as specified below.

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.

This threat takes advantage of the fact that the external NVM is not integrated into the host MCU chip. Hence, the physical protections for preventing the replacement of stored contents may not cover the external NVM. This situation enables an attacker to read and write the contents of the external NVM. Even when the NVM contents are protected in confidentiality and integrity, the replacement may be valid as well, since it is retrieved from the external NVM.

If the TOE code is stored in the external NVM, this threat may lead to an unauthorized rollback of the TOE code to an older version. Even when the TOE stores data and not code in the external NVM, this data might affect the behavior of the TSF.

The replacement of contents stored in the external NVM with previous versions of them may refer to the full contents of the NVM or partial contents of it, depending on the implementation of the mechanisms used by the TOE to organize and protect the information stored in the external NVM.

Application Note 9:	*An illustrative example of an attack related to this threat is described as follows. An attacker, who obtains physical access to the TOE, reads all the contents of its external NVM and saves them in auxiliary storage. Then, after the host MCU carries out write operations on external NVM, thus updating its contents with a new version, the attacker writes the contents from the auxiliary storage back to the external NVM of the TOE.*

As a result, the contents of the external NVM have been reverted to a version different from the one resulting from the last write/erase operation issued by the host MCU on the external NVM and, therefore, data freshness property has been violated.

The TOE shall avert the threat “Cloning or replacement of NVM (T.NVM-Clone-Replace)”, as specified below.

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.

This threat refers to the case where the full contents of the external NVM are cloned to a different device. It can also cover the replacement of the NVM of the TOE with the NVM of a different unit. The second case might not be viable on some architectures when the physical design or assembling procedures impede it.

The effect of this threat is in replacing the data or code of a TOE with a different one.

Unlike T.NVM-Unauthorized-Rollback, the threat of T.NVM-Clone-Replace involves using two different TOE units or instances. One TOE unit is used as a source for the external NVM contents. Those contents are used to replace the genuine contents of the external NVM of the second TOE unit. The notion of data freshness does not apply to this situation

Another possible scenario for this threat can be contemplated: the TOE NVM is replaced with an empty or virgin NVM, removing the user and TSF data from the TOE and possibly forcing the TSF to generate new user and TSF data, potentially affecting the TSF behavior.

3.2.2. Threats specific to architecture with SM option

Application Note 10: *The contents in this section must be added in the ST when the TOE uses an architecture with an NVM and bus that are in-scope of the TOE (option SM as defined in section 2.2).*

In the PM option, the host MCU TOE must ensure that confidentiality of contents is guaranteed before contents is serialized into the external NVM, and that integrity of contents is verified when contents is deserialized from the external NVM, as defined by the previous T.External-Content-Abuse threat. In addition, in the PM option, the TOE must ensure that commands issued to external NVM cannot be blocked as defined by the T.Malfunction threat.

The TOE shall avert the threat “Abuse of interface between host MCU and external NVM (T.NVM-Abuse-Interface)” as specified below.

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.

Read, write and erase operations between the Security IC and the external NVM can be blocked or intercepted by an attacker eavesdropping to the interconnection bus to disclose the user data and TSF

data being written or read from the external NVM. This threat only applies to SM option since in PM option the bus and external NVM are outside the TOE.

3.3. Organizational Security Policies (PP Ch. 3.3)

3.4. Assumptions (PP Ch 3.4)

4. Security Objectives (PP Ch 4)

4.1. Security Objectives for the TOE (PP Ch. 4.1)

4.1.1. Security Goals common to architectures with PM and SM options

When the Security IC Platform uses an external NVM for storage, the following high-level Security Goals are defined.

SG.External-Content-Protection	Protect against disclosure and undetected modification of external NVM contents.

The contents stored in external NVM must be protected from disclosure and undetected modification. The TOE shall provide confidentiality and integrity protection of the contents stored in external NVM. The contents stored in external NVM must be protected in confidentiality so that it cannot be disclosed if successfully accessed. The contents stored in external NVM must be protected in integrity so that the change is either protected by the external NVM if it is capable to do it, or detected by the host MCU and rejected.

The contents stored in external NVM must be protected to ensure that it is not replaced by different contents not bound to the TOE.

SG.NVM-Updated-Contents	Ensure that the contents stored in the external NVM have not been replaced by a previous version of them.

External NVM is not part of the same physical chip as the host MCU, and it may be possible to write or read them directly. Since the data and code of the TOE must be protected in terms of confidentiality and integrity, the contents in the external NVM cannot be arbitrarily manipulated (e.g., by raw block writing), due to the mentioned protections. Instead, the contents of the external NVM can be read and recorded at a given point in time, then such recorded contents can be written back to the external NVM at a future time.

The contents stored in the external NVM are transferred between the host MCU and the external NVM through an interconnection bus. Data freshness of the external NVM contents could be violated by replaying recorded sequences of such commands.

SG.NVM-Updated-Contents aims to prevent the replacement of the external NVM contents with older (non-fresh) versions of those contents (coming from the same external NVM. The related security concern is preserving data freshness. Threats countered by this security goal require only one TOE unit to attempt to mount a related attack.

SG.NVM-Genuine-Content	Ensure that the contents of the NVM are genuine.

The physical access available to the NVM could enable attackers to replace the contents of the external NVM of the TOE with contents obtained from the external NVM of a different TOE unit. The TOE must be protected against such a replacement.

SG.NVM-Genuine-Content aims to avoid the replacement of external NVM contents of the TOE with those coming from the external NVM of a different unit. Hence, attempting to mount a related attack requires two different units, and the contents of the replacement are not genuine for the TOE in which they are written.

4.1.2. Security Goals specific to architecture with SM option

Application Note 11: *The contents in this section must be added in the ST when the TOE uses an architecture with an NVM and bus that are in-scope of the TOE (option SM as defined in section 2.2).*

SG.NVM-Interface-Protection	Protect against disclosure and undetected modification of data transferred between the host MCU and the external NVM.

The data transferred between the Security IC and the external NVM through the interconnection bus needs to be protected from disclosure and undetected modification. The TOE shall provide confidentiality protection of the data in transit as well as detecting modification of data by abusing the interface between host MCU and external NVM.

4.1.3. Security Objectives common to architectures with PM and SM options

Security Objectives Related to SG.External-Content-Protection

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.

This security objective requires that the TOE protects the contents stored in external NVM so that unauthorized users cannot read it. In addition, the TOE protects the contents stored in external NVM so that it cannot be modified by unauthorized users without such modifications being detected. The TOE shall then identify the contents as invalid.

Security Objectives Related to SG.NVM-Updated-Contents

The TOE shall provide “Protection against replay of commands between host MCU and external NVM (O.NVM-Command-Replay-Protection)”, as specified below.

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.

This security objective requires that the TOE protects against replaying payloads of the read commands that have been recorded by an attacker previously. The TOE shall be able to detect this scenario and identify the contents as old, outdated, or deprecated.

The TOE shall provide “Protection against an unauthorized rollback of NVM contents (O.NVM-Unauthorized-Rollback-Protection)”, as specified below.

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.

By means of this objective, the TOE shall be able to detect replacing the external NVM contents with a previous version from the same external NVM, read at some point in the past. Such contents were

valid in confidentiality and integrity protection at the moment they were read. When this situation occurs, the TOE must detect that data freshness has been compromised.

The TOE shall provide "(O.NVM-Irreversibility-Anchor)", as specified below:

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.

In order to manage data freshness, the TOE shall provide an irreversibility anchor, as described in the above security objective. Advancing the value of the irreversibility anchor through its defined sequence of states mechanism occurs when write/erase operations are issued by the Security IC. This mechanism shall allow detecting and protecting against violation of data freshness of the external NVM contents.

Application Note 12: *This mechanism can be vulnerable to fault injection attacks, e.g. voltage glitches. The attacker may try to prevent the Irreversibility Anchor from incrementing, or alter its perceived value, which would enable the attacker to replay old sessions of commands. These kinds of attack are covered by the T.Malfunction threat defined in [1].*

Security Objectives Related to SG.NVM-Genuine-Content

The TOE shall provide "Protection against external NVM cloning or replacement (O.NVM-Clone-Replace-Protection)", as specified below.

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.

By means of this objective, the TOE shall protect against the replacement of its external NVM contents with ones from a different unit. While those contents are valid for the 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. The TOE shall also protect against a similar scenario where, instead of cloning the contents of one NVM into another, the external NVM is physically replaced by the external NVM of a different TOE unit.

4.1.4. Security Objectives specific to architecture with SM option

Application Note 13: *The contents in this section must be added in the ST when the TOE uses an architecture with an NVM and bus that are in-scope of the TOE (option SM as defined in section 2.2).*

Security Objectives Related to SG.NVM-Interface-Protection

The TOE shall provide "Protection against abuse of the interface between host MCU and external NVM (O.NVM-Interface-Protection)", as specified below.

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.

Since the interconnection bus between the host MCU and the external NVM can be 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. Manipulation through issuing new commands for modifying the data in the NVM must be detected by the TOE as well.

Application Note 14: *This security objective does not mandate any particular implementation mechanism for achieving protection against disclosure or undetected manipulation of data in transit between the host MCU and the external NVM. The related mechanisms are dependent on the implementation of the TOE. In implementations where the NVM implements TSF, this kind of protection can be provided by the external NVM in conjunction with the host MCU. The TOE shall meet this objective considering the particular characteristics and limitations of the chosen implementation.*

4.2. Security Objectives for the Security IC Embedded Software (PP Ch. 4.2)

4.3. Security Objectives for the operational Environment (PP Ch. 4.3)

4.4. Security Objectives Rationale (PP Ch. 4.4)

4.4.1. Rationale common to architectures with PM and SM options

Application Note 15: *Add the following entry to Table 1 in [1].*

Assumption, Threat or Organizational Security Policy	Security Objective	Notes
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	

Application Note 16: *The security objectives rationale in [1] already includes the following text in paragraph 127:*

"For all threats, the corresponding objectives (refer to Table 1) are stated in a way, which directly corresponds to the description of the threat (refer to Section 3.2). It is clear from the description of each objective (refer to Section 4.1) that the corresponding threat is removed if the objective is valid. More specifically, in every case the ability to use the attack method successfully is countered if the objective holds."

However, for those threats and security objectives introduced in this augmentation package, a more detailed rationale is required and provided.

The rationale below shall be added after such a paragraph of [1] as justification to the threats introduced in this Augmentation Package.

T.External-Content-Abuse is countered by O. External-Content-Protection, which requires the TOE to prevent disclosure and undetected modification of the contents stored in external NVM.

T.NVM-Command-Replay is countered by O.NVM-Command-Replay-Protection and O.NVM-Irreversibility-Anchor as follows:

- O.NVM-Command-Replay-Protection requires that the TOE implements protection against the replay of commands between the Security IC and the external NVM through the interconnection bus, mitigating T.NVM-Command-Replay.
- O.NVM-Irreversibility-Anchor requires that the TOE implements a mechanism that goes through a sequence of states associated with the changes issued by the Security IC on the NVM. This mechanism helps for the detection of older NVM commands, as contents of such commands would not meet the requirement of having a consistent value of the irreversibility anchor.

T.NVM-Unauthorized-Rollback is countered by O.NVM-Unauthorized-Rollback-Protection and O.NVM-Irreversibility-Anchor as follows:

- O.NVM-Unauthorized-Rollback-Protection requires that the TOE protects against replacement of external NVM contents with older contents of the same NVM, where the data freshness property is not met, thus, mitigating this threat.
- O.NVM-Irreversibility-Anchor requires that the TOE implements a mechanism that goes through a sequence of states associated with the changes issued by the Security IC on the external NVM. Unauthorized rollback of contents of the external NVM would result in a state that would fail the validation against the current value of the irreversibility anchor mechanism.

T.NVM-Clone-Replace is countered by O.NVM-Clone-Replace-Protection, which requires the TOE to detect the replacement of the external NVM contents with those of a different TOE's NVM, or physical replacement of the external NVM with the external NVM of a different TOE unit.

4.4.2. Rationale specific to architecture with SM option

Application Note 17: *Add the following entry to Table 1 in [1] when the TOE uses an architecture with an NVM and bus that are in-scope of the TOE (option SM as defined in section 2.2).*

Assumption, Threat or Organizational Security Policy	Security Objective	Notes
T.NVM-Abuse-Interface	O.NVM-Interface-Protection	

Application Note 18: *The rationale below shall be added after such a paragraph of [1] as justification to the threats introduced in this Augmentation Package in the case of the SM option.*

T.NVM-Abuse-Interface is countered by O.NVM-Interface-Protection, which requires the TOE to prevent disclosure and detect modification of the data in transit between the host MCU and the external NVM.

5. Extended Components Definition (PP Ch. 5)

Application Note 19: *Add the following definition at the end of section 5 of [1] when writing the ST.*

In order to address the security aspects derived from the usage of an external NVM for storage of Security IC code and data, the following extended families have been added:

- **FDP_URC**, for protection against unauthorized rollback (e.g., revert to a previous non-fresh version) of the contents of the external NVM.
- **FDP_IRA,** for providing a non-volatile irreversibility anchor that serves to ensure freshness of the external NVM stored contents.
- **FPT_CRP,** for protection against replacement of the external NVM contents with those from the NVM of a different TOE unit.

Definition of the Family FDP_URC

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

This family describes the functional requirements for the detection of unauthorized rollback of the contents stored in the external NVM. An unauthorized rollback situation occurs when the contents of the external NVM are replaced with other previous contents of the same NVM that were valid at a certain moment in the past. The unauthorized rollback is understood as a full replacement of external NVM contents with previous valid contents, or partial content replacement.

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 behavior

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.

FDP_URC.1	**Protection against an unauthorized rollback of stored contents**
Hierarchical to:	No other components
Dependencies:	No dependencies
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]].

Definition of the Family FDP_IRA

To define the security functional requirements of the TOE, an additional family (FDP_IRA) of the Class FDP (User Data Protection) is defined here.

This family describes the functional requirements for the implementation of an irreversibility anchor mechanism linked to the data freshness property for the contents of the external NVM. The external NVM irreversibility anchor mechanism consists of a non-volatile mutable mechanism that goes through a predefined sequence of states (that are associated with increasing values). These values are increasing and cannot revert to previous states. They serve to determine whether the contents of the NVM meet the property of data freshness, by verifying that they are those resulting from the latest write or erase operation issued by the Security IC on the external NVM. Otherwise, the data freshness property is not met, and this mechanism serves to prevent or detect it.

The family FDP_IRA aims to cover the requirements associated with the above problem, providing the ability to prevent or detecting the scenario of external NVM contents not being fresh

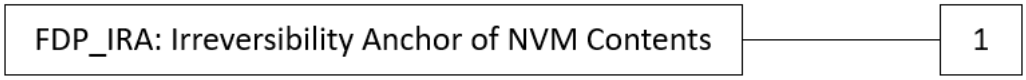
The family “Irreversibility Anchor of NVM contents (FDP_IRA)” is specified as follows:

FDP_IRA: Irreversibility Anchor of NVM contents

Family behavior

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



Management	FDP_IRA.1

There are no management activities foreseen.

Audit	FDP_IRA.1

There are no actions defined to be auditable.

FDP_IRA.1	**Irreversibility Anchor of NVM contents**
Hierarchical to:	No other components
Dependencies:	No dependencies
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 contentsmeet data freshness].

Application Note 20: *The TSF related to this mechanism can be implemented in multiple ways. Typical architectures include:*

- *A protected field in an internal memory of the Security IC, e.g., an effuse array, an EEPROM variable, etc.*
- *A mechanism in the external NVM (in case it implements any TSF), e.g., a dedicated Flash array programmed bit-by-bit, etc.*

The above implementation mechanisms are illustrative, and the possible implementation options are not limited to them.

Definition of the Family FPT_CRP

To define the security functional requirements of the TOE, an additional family (FPT_CRP) of the Class FPT (Protection of the TSF) is defined here.

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.

The family FPT_CRP aims to cover those requirements that are intended to detect the above-described situations, providing the ability to the TSF for protecting against such kind of NVM contents cloning or replacement.

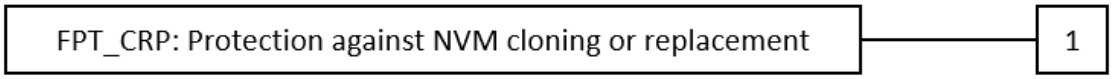
The family “Protection against NVM cloning or replacement (FPT_CRP)” is specified as follows:

FPT_CRP.1: Protection against NVM cloning or replacement

Family behavior

This family defines functional requirements for the detection of cloning or replacement of the NVM.

Component Levelling



Management	FPT_CRP.1

There are no management activities foreseen.

Audit	FPT_CRP.1

There are no actions defined to be auditable.

FPT_CRP.1	**Protection against NVM cloning or replacement**
Hierarchical to:	No other components
Dependencies:	No dependencies
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.

6. IT Requirements (PP Ch. 6)

6.1. Security Functional Requirements for the TOE (PP Ch. 6.1)

Application Note 21: *When several external NVMs are used by the TOE for storage of data and code, the SFRs described in this section shall be iterated or modified for each used external NVM, as described in the guidelines below. The guidelines also contemplate how to include the additional iterations in the security requirements rationale, dependency analysis, and internal consistency analysis.*

When multiple external NVMs are used, the ST author shall follow the guidelines below:

- *SFRs in section 6.1.1, common for both PM and SM architectures, shall be iterated for each external NVM used. Alternatively, if it is possible to contemplate multiple external NVMs by modifying an assignment given in one of those SFRs (providing that such assignment is open in [1]), the modification of the assignments to cover multiple external NVMs shall be done as an alternative to the SFR iteration (e.g., FDP_SDC.1).*
- *When more than one TSF external NVM (according to option SM) are used, each SFR in section 6.1.2 shall be iterated for each of those memories.*
- *For each SFR iteration added due to additional external NVMs, the ST author shall:*
    - *Include that iteration in the* Rationale for the security functional requirements (PP Ch. 6.3.1) *provided in section 6.5, where each iteration shall be mapped to the TOE same security objectives as the original SFR.*
    - *Include that iteration in the* Dependencies of security functional requirements (PP Ch. 6.3.2) *dependency analysis given in section 6.6, for justification of the dependencies of those iterated SFRs, in a way similar to the dependency analysis provided for the original SFR.*

Include that iteration in the analysis done in section Security Requirements are Internally Consistent (PP Ch. 6.3.4) *given in section 6.8, providing an analysis similar to that of the original SFR.*

6.1.1. Security Functional Requirements for external NVM

Application Note 22: *This Section lists the Security Functional Requirements common to both PM and SM architectural options defined in Section 2.2*

Application Note 23 *Assignment in SFR FDP_SDC.1.1 of [1].*

*The author of the ST **shall** complete the assignment of FDP_SDC.1.1 ([assignment: memory area]), indicating the NVM external memory as the area where the data protected by FDP_SDC.1.1.is stored*

Application Note 24: *The author of the ST shall add an application note after the SFR FDP_SDI.2 defined in [1], with the following content:*

"The monitoring of the integrity of stored data includes the data stored in the external NVM."

Protection of NVM content freshness

The TOE shall meet the requirement "Protection against an unauthorized rollback of stored contents (FDP_URC.1)", as specified below.

FDP_URC.1 Protection against an unauthorized rollback of stored contents

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

The TOE shall meet the requirement "Irreversibility Anchor of NVM contents (FDP_IRA.1)", as specified below.

FDP_IRA.1 Irreversibility Anchor of NVM contents

Hierarchical to:	No other components
Dependencies:	No dependencies
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 meet the requirement "Protection against NVM cloning or replacement (FPT_CRP.1)", as specified below.

FPT_CRP.1 Protection against NVM cloning or replacement

Hierarchical to:	No other components
Dependencies:	No dependencies
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 25: *If the external NVM Is not in the scope of the TOE (option PM), the part of this SFR that requires detection of physical replacement of the external NVM is relevant only if the contents used for the replacement are different from those in the original NVM. If the replaced contents are equal, the physical replacement is not relevant for that specific subcase because the external NVM is a passive memory without security functionality.*

Data transfer between host MCU and external NVM

The TOE shall meet the requirement "Replay detection (FPT_RPL.1)", as specified below.

FPT_RPL.1 Replay detection

Hierarchical to:	No other components
Dependencies:	No dependencies
FPT_RPL.1	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[1].
FPT_RPL.2	The TSF shall perform [assignment: list of specific actions] when a replay is detected.

Application Note 26: *The replay can occur as a response to read command, but also sequences of write and erase commands can be recorded and sent to the external NVM by an attacker. In this case, the replay may not be detected until a later read operation happens.*

6.1.2. Security Functional Requirements for external NVM with SM architecture option

Application Note 27: *The contents in this section must be added when the TOE uses an architecture with a TSF external NVM **(option SM)**, as defined in Section 2.2.*

Application Note 28: *If the NVM implements TSF logic for protecting the confidentiality of the information stored in it, the TOE Summary Specification **shall detail** how FDP_SDC.1 is implemented for each memory area where stored data is protected with a distinct logic, including the external NVM.*

[1] [assignment: list of identified entities].

Example: "The confidentiality of the data stored in the TOE internal memories is implemented by TDES encryption carried out by the Security IC, while the confidentiality of the data stored in the external NVM is provided by a memory scrambling mechanism implemented in the external NVM. This way, the FDP_SDC.1 requirement is met."

Application Note 29: *If the NVM implements TSF logic for protecting the integrity of the information stored in it, the TOE Summary Specification **shall detail** how FDP_SDI.2 is implemented for each memory area where stored data is protected with a distinct logic, including the external NVM.*

Example: "The TOE maintains parity bits for each memory block in the internal TOE NVMs, and the external NVM computes a CRC digest for each memory block in it, thus meeting FDP_SDI.2."

Application Note 30: *This application note refers to FPT_PHP.3.*

If any of the following SFRs is implemented in the NVM:

- *FDP_SDC.1*
- *FDP_SDI.1*

*then, the TOE Summary Specification **shall outline** how FPT_PHP.3 prevents the violation of the above SFRs by physical manipulation of the external NVM.*

Application Note 31: *The following SFRs cover the protection of data transfer between host MCU and external NVM when the external NVM is part of the TOE:*

FPT_ITT.1 is iterated (FPT_ITT.1/NVM) for ensuring the confidentiality and integrity of the TSF data in transit between the host MCU and the external NVM.

FDP_ITT.1 is iterated (FDP_ITT.1/NVM) for ensuring the confidentiality and integrity of the user data in transit between the host MCU and the external NVM.

*These contents belong under the **Data transfer between host MCU and external NVM** title.*

The TOE shall meet the requirement "Basic internal TSF data transfer protection" (FPT_ITT.1/NVM)" as specified below.

FPT_ITT.1/NVM Basic internal TSF data transfer protection

Hierarchical to:	No other components
Dependencies:	No dependencies
FPT_ITT.1.1/NVM	The TSF shall protect TSF data from disclosure, modification[2] when it is transmitted between separate parts of the TOE.
Refinement:	**The external NVM is considered a physically separated part of the TOE, even though it is packaged together with the Security IC.**

[2] [selection: disclosure, modification]

Application Note 32: *The Protection Profile [1] 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 and the NVM. This particular SFR requires that the TSF data exchanged between other parts of the TOE and the external NVM is protected both from disclosure and modification.*

The TOE shall meet the requirement "Basic internal transfer protection (FDP_ITT.1/NVM)", as specified below.

FDP_ITT.1/NVM Basic internal transfer protection

Hierarchical to:	No other components
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]
FDP_ITT.1.1/NVM	The TSF shall enforce the Data Processing Policy[3] to prevent the disclosure, modification[4] of user data when it is transmitted between other parts of the TOE and the external NVM.
Refinement:	**The NVM is seen as a physically separated part of the TOE, as well as other TOE elements mentioned in the refinement of FDP_ITT.1 SFR.**

Application Note 33: *The Protection Profile [1] 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 and the NVM. This particular SFR requires that the user data exchanged between other parts of the TOE and the external NVM is protected both from disclosure and modification.*

Application Note 34: *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.*

This requirement is equivalent to FDP_ITT.1/NVM above but refers to TSF data instead of user data. Therefore, it should be understood as to refer to the same Data Processing Policy defined under FDP_IFC.1.

6.2. Security Assurance Requirements for the TOE (PP Ch. 6.2)

6.3. Refinements of the TOE Assurance Requirements (PP Ch. 6.2.1)

6.4. Security Requirements Rationale (PP Ch. 6.3)

6.5. Rationale for the security functional requirements (PP Ch. 6.3.1)

[3] [assignment: access control SFP(s) and/or information flow control SFP(s)]
[4] [selection: disclosure, modification, loss of use]

6.5.1. Common Security Functional Rationale for external NVM with PM or SM architecture

Application Note 35: Add the following entry to Table 2 in [1].

Objective	TOE Security Functional and Assurance Requirements
O.External-**Content-Protection**	- FDP_SDC.1 for confidentiality protection
O.External-Content-**Protection**	- FDP_SDI.2 for integrity protection
O.NVM-Command-Replay-Protection	- FPT_RPL.1 "Replay detection"
O.**NVM-Unauthorized-Rollback-Protection**	- FDP_URC.1 "Protection against an unauthorized rollback of stored contents"
O.NVM-Irreversibility-Anchor	- FDP_IRA.1 "Irreversibility Anchor of NVM contents"
O.NVM-Clone-**Replace-Protection**	- FPT_CRP.1 "Protection against NVM Cloning or Replacement"

Application Note 36: Add the following text to section 6.3.1 of [1].

The SFR FDP_SDC.1 and FDP_SDI.2 documented in [1] support the objective O.External-Content-Protection.

The justification related to the security objective "Protection against unauthorized disclosure and undetected modification of external NVM contents (O.External-Content-Protection)" is as follows:

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.

The justification related to the security objective "Protection against replay of commands between host MCU and external NVM (O.NVM-Command-Replay-Protection)" is as follows:

The SFR FPT_RPL.1 requires the TSF to detect replays in responses in the read commands to the NVM or replays of sequences of write/erase commands to the external NVM. This requirement is considered in the assignment of FPT_RPL.1.1. Therefore, it is clear that this security functional requirement supports the objective.

The justification related to the security objective "Protection against contents (O.NVM-Unauthorized-Rollback-Protection)" is as follows:

The SFR FDP_URC.1 requires that the TSF detects the case when the contents of the external NVM have been replaced by previous versions of them. This way, this security functional requirement supports the objective.

The justification related to the security objective "External NVM Contents Irreversibility Anchor (O.NVM-Irreversibility-Anchor)" is as follows:

The SFR FDP_IRA.1 requires the TOE to implement a non-volatile mechanism that is irreversible and serves to mark the NVM contents as meeting or not meeting data freshness property. By providing the mechanism required by this SFR, the security objective O.NVM-Irreversibility-Anchor is directly supported.

The justification related to the security objective "Protection against NVM cloning or replacement (O.NVM-Clone-Replace-Protection)" is as follows:

The SFR FDP_URC.1 requires the TOE to detect the situation where the contents in the external NVM have been cloned from the external NVM of a different TOE unit. It also requires that the physical replacement of the external NVM with another NVM. This way, this SFR supports the objective.

6.5.2. Security Functional Rationale for external NVM with SM architecture option

Application Note 37: *Add the following entry to Table 2 in [1] and the justification text after the table to section 6.3.1 of [1] only if the architecture **Option SM** is chosen according to section 2.2.*

Objective	TOE Security Functional and Assurance Requirements
O.NVM-Interface-Protection	- FPT_ITT.1/NVM "Basic internal TSF data transfer protection" - FDP_ITT.1/NVM "Basic internal transfer protection"

The justification related to the security objective "Protection against abuse of the interface between host MCU and external NVM (O.NVM-Interface-Protection)" is as follows:

FPT_ITT.1/NVM requires the TOE to protect TSF data when transferred between physically separate parts of the TOE, and FDP_ITT.1/NVM requires the TOE to protect user data when transferred between separate parts of the TOE. The external NVM is seen as a separate part of the TOE. Therefore, those two requirements support the security objective.

6.6. Dependencies of security functional requirements (PP Ch. 6.3.2)

6.6.1. Common dependencies of security functional requirements for external NVM with PM or SM architecture

Application Note 38: *Add the following entry to Table 3 in [1].*

Security Functional Requirement	Dependencies	Fulfilled by security requirements in this PP
FPT_RPL.1	None	No dependency
FDP_URC**.1**	None	No dependency
FDP_IRA.1	None	No dependency
FPT_CRP.1	None	No dependency

6.6.2. Dependencies of security functional requirements for external NVM with SM architecture option

Application Note 39: *Add the following entry to Table 3 in [1] only if the architecture **Option SM** is chosen according to section 2.2.*

Security Functional Requirement	Dependencies	Fulfilled by security requirements in this PP
FPT_ITT.1/NVM	None	No dependency
FDP_ITT.1/NVM	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	FDP_IFC.1

6.7. Rationale for the Assurance Requirements (PP Ch. 6.3.3)

6.8. Security Requirements are Internally Consistent (PP Ch. 6.3.4)

6.8.1. Common Security Requirements are Internally Consistent Analysis for external NVMs with PM or SM architecture options

The contents of the external NVM (user data, TSF data and code) need to be protected, considering that the external NVM may be more accessible to attackers than the internal (embedded in the same chip) one.

Confidentiality and integrity of the contents stored in the external NVM are given by FDP_SDC.1 and FDP_SDI.2. Also, the contents stored in the NVM require a guarantee of “data freshness”, namely that the contents are the last ones stored under control of the Security IC). The TOE needs to be capable of detecting the replay of old write, erase, and read operations to the external NVM. This requirement is provided by FPT_RPL.1.

The non-volatile mutable irreversibility anchor provides a mechanism for protection against reverting external NVM contents to old non-fresh versions of them is given by FDP_IRA.1. In addition, the extended component FDP_URC.1 provides protection against an unauthorized rollback of the NVM contents to old versions of them, and the extended component FPT_CRP.1 ensures protection against cloning or replacement of the external NVM.

6.8.2. Requirements are Internally Consistent Analysis for external NVMs with SM architecture options

Application Note 40: *Add the following text to section 6.3.4 of [1] only if the architecture **Option SM** is chosen according to section 2.2.*

The data in transit between the external NVM and the host MCU between the interconnection bus needs the protection of confidentiality and detection of modification. For this reason, FDP_ITT.1/NVM and FDP_ITT.1/NVM are included in this PP.

Appendix I: Additional external NVM-Security related issues

This section lists several security issues correlated with the further usage of an external NVM for storage of the Security IC data and code that remain out of the scope of the TOE Protection Profile. It is important to have these issues in mind to include them in the scope of the Common Criteria evaluation methodology. References to relevant documentation have been included.

Software updates

The traditional security paradigm established in the Security IC Platform Protection Profile PP0084 [1] does not cover the security of software updates to products in the field. The related issues include the following topics:

a) The Assurance Continuity, i.e., keeping the TOE code updates secure and certified. The process is partially defined by [7].

b) Protecting the TOE from abusing its Code Update Mechanism (Loader) by a malicious attacker. This aspect is partially described in [4] and [8], but the exposition there lacks treating the data freshness protection, which comes up as a more important issue with the use of external NVMs.

c) Keeping the TOE properly updated, by giving a Security Provider a set of tools that would prevent an attacker from blocking software updates to the TOE. Currently, no coverage of this task is provided in the Common Criteria evaluation standards or its supporting documents.

Code loading after TOE delivery

If the Security IC embedded software must be loaded in the NVM of the IC, the TOE can be delivered after Phase 3 or Phase 4 without the embedded software being loaded (e.g., it is loaded at Composite Product Manufacturer sites after delivery). In this case, the ANSSI-CC-NOTE-14/1.0 [6] is applicable, and it must be considered if it is mandatory for the Certification Body involved in the evaluation.