EUROPEAN UNION AGENCY
FOR CYBERSECURITY

EUCC SCHEME

STATE-OF-THE-ART DOCUMENT

Security Architecture requirements (ADV_ARC) for smart cards and similar devices extended to Secure Sub Systems in SoCs

Version 1.1, October 2023

DOCUMENT HISTORY

Date	Version	Modification	Author's comments
October 2022	1.0	Creation	Version submitted to the ECCG Based on JIL-Security-Architecture-requirements-(ADV_ARC)-v2-1.pdf
October 2023	1.1	Addition of a document history section with a link to the SOG-IS document the state-of-the-art document is based on, where applicable	Based on ECCG comments received Approved for publication on 20/10/2023 by the ECCG and the European Commission

CONTENTS

1 INTRODUCTION

1.1 Objective of the document

This state-of-the-art document supporting the EUCC scheme provides requirements for the developer and guidance for the evaluator on how to apply the assurance requirements of the family ADV_ARC to the Technical Domain of smart cards and similar devices extended to Secure Sub-Systems (3S) in SoC[1]. The developer documentation provided to fulfil the ADV_ARC family is denoted as "ARC document" in the text. The current document keeps same information applicable to smart cards & similar devices, and it includes additional sections dedicated to Secure Sub-Systems in SoC.

The technology associated to this Technical Domain requires special interpretation because it combines security integrated circuits or 3S in SoC, operating systems and applications to high secure devices. Therefore, this document is intended to provide mandatory interpretation for the application of the ADV_ARC family. It is addressed to both developers of security integrated circuits and developers of composite products, consisting of a hardware platform and embedded software. The embedded software can be organised in different ways (native software, closed operating systems with one or more applications, open software platforms and more).

The expected assurance level is AVA_VAN.5.

The mandatory interpretation defines what kind of information the ARC document SHALL contain and in which level of detail this information SHALL be provided. It does not define mandatory tasks for the evaluator. However, the document also serves as a guideline for the evaluator: in order to have a clear agreement between evaluator and developer, it states which kind of developer information is mandatory and may also define which is not.

An informative part is provided in two appendix (one for smart cards & similar devices and one dedicated to 3S in SoC) which contain examples for the type of information and level of detail to be provided in the ARC document.

A number of supporting documents have been issued for this Technical Domain with which the current document is in coherency as listed in § References.

1.2 References

ID	Title of document
[AAPS]	Application of attack potential to smart cards
[AMS]	Attack methods for smart cards
[GC3S]	Guidance for the evaluation of a Secure Sub-System within a SoC
[CPE]	"Composite product evaluation for Smart Cards and similar devices"
[EFI]	ETR For Integration
[SCE]	Guidance for Smartcard evaluation
[PP0084]	Security IC Platform Protection Profile with Augmentation Packages, BSI-CC-PP-0084-2014
[PP3SinSoC]	Secure Sub-System in System-on-Chip Protection Profile

1.3 Scope of Security Architecture for smart cards & similar devices

The Common Criteria (CC) introduces a security assurance requirements (SAR) family Security Architecture (ADV_ARC). Its objective is described as follows:

"The objective of this family is for the developer to provide a description of the security architecture of the TSF. This will allow analysis of the information that, when coupled with the other evidence presented for the TSF, will confirm the TSF achieves the desired properties. The security architecture descriptions support the implicit claim that security analysis of the TOE can be achieved by examining the TSF; without a sound architecture, the entire TOE functionality would have to be examined."

A security architecture is a set of properties that the TSF exhibits; these properties include self-protection, domain separation, and non-bypassability. These properties are distinct from security functionality expressed in SFRs because they largely have no directly observable interface at the TSF. Rather, they are properties of the TSF that are achieved through the design of the TOE and TSF, and enforced by the correct implementation of that design.

The Security Architecture shall also describe the TOE security functionality (TSF) initialisation, i.e. the processing that occurs in transitioning from the "down" state to the initial secure state, when power-on or a reset is applied.

[1] SoC : System on Chip

Some features of the security architecture such as non-bypassability and domain separation shall as well be described through the ADV_ARC family.

The Technical Domain of Smart card & Similar Devices presents specificities that have to be taken into account in the drawing up of the ADV_ARC documentation. The main characteristics for Smart card & Similar Devices are:

- A device belonging to this domain is a combination of a one-chip Integrated Circuit with embedded software implementing cryptographic services using secrets. The TOE could cover the full product or only a layer that includes the IC (an underlying platform).
- The TOE may start up in a low-function mode and then transition to the evaluated secure configuration. A transition from power off also happens each time the device is used by the final holder.
- In its operational environment an attacker might have physical access to the TOE through the physical port and the IC surface.
- The TOE may contain a flash loader, used to load data (e.g. software, keys and other assets) to the non-volatile memory of the TOE or TOE environment. The lifecycle is similar to that described in the "Guidance for smartcard evaluation" document.

Depending on the respective TOE, a so called flash loader can be present. The flash loader or short loader, is used to write data (e.g. firmware, software, keys and other assets) to non-volatile memory of the TOE. The loader is considered as a part of the TOE and shall be described accordingly. For example, the protection profile [PP0084, 7.3] already provides pre-defined functional packages covering different usage scenarios.

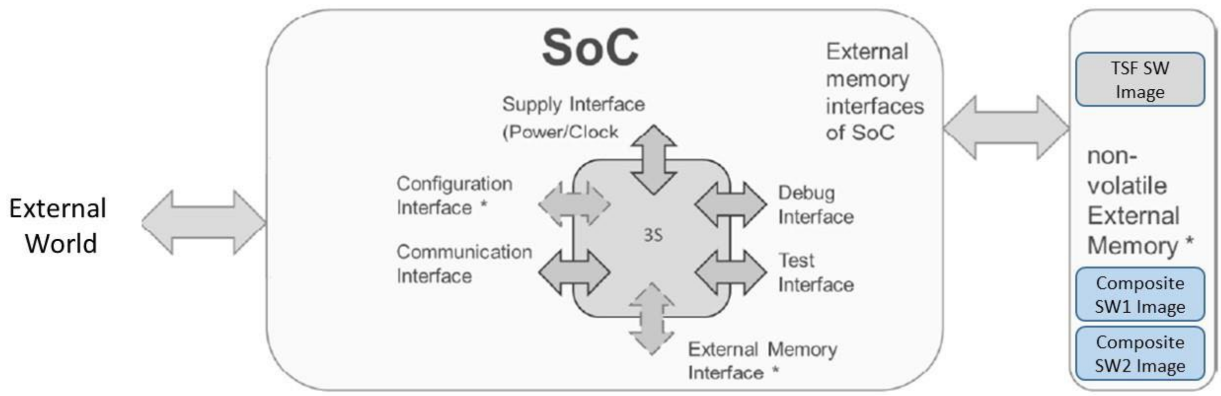
The description of the related TSF shall enable the evaluator to access the implemented TOE security functionality. It depends on the implemented security mechanisms of the loader and the related security policies, if the loader requires a trusted environment.

Moreover, the document focuses on devices that must be resistant to attackers with a high attack potential.

1.4 Scope for Secure Sub-Systems (3S) on a System-on-a-Chip (SoC)

A security architecture for a Secure Sub-Systems (3S) includes the same set of properties as defined for smart cards in §1.2. Only some key characteristics are different as illustrated on next figure.

Figure 1[2]: 3S in SoC environment



The main characteristics for 3S in SoC are to protect SoC and Composite Software[3] (as defined in [PP3SinSoC], equivalent to Secure IC Embedded Software in [PP0084]) high value assets and more specifically:

- A 3S in SoC is an IP block of an Integrated Circuit (SoC) with software implementing services (e.g. cryptography) optionally extended with volatile memory (e.g DDR[4] RAM) and non-volatile memory (e.g. Flash).
- The TOE starts up using only firmware (FW) in TOE and may then load an encrypted software (SW) image from external memory (if any). If loading from external memory fails, TOE is considered in

[2] *: means optional component or interface.
[3] See definition in Glossary [PP3SinSoC].
[4] DDR: Double Data Rate.

degraded mode and may implement a recovery mode where optional services contribute to recover some or all TOE features.

- In operational environment, an attacker might have physical access to the TOE through the physical interface of the SoC and the SoC surfaces or direct access to 3S surfaces or interface(s) of the 3S in SoC according to SoC architecture.
- The lifecycle is similar to that described in the [GC3S] document.

2 GENERAL ASPECTS OF CONTENT AND PRESENTATION

ARC documentation supports the vulnerability analysis of the evaluator but it does not provide a developer vulnerability analysis. The developer designs, implements and describes the security architecture of the TOE. The ARC documentation describes security domains and the secure initialisation process and demonstrates self-protection and non-bypassability. The description focuses on the use of security mechanisms, which are put into place and their collaboration in order to achieve overall security. To this end, the developer may analyse and conclude how the security features and countermeasures of the TOE are intended to resist the general attacks listed in the state-of-the –art document “Application of Attack Potential to smart cards” viewed in the light of tampering and bypass. In contrast to the ARC document, the evaluator performs an independent vulnerability analysis to determine the actual resistance of TOE to attacks. The evaluator shall consider all potential vulnerabilities encountered while performing evaluator activities or found by independent methodical search. The evaluator will determine whether vulnerabilities are exploitable by an attacker possessing the attack potential addressed in the ST. Thus ARC documentation and vulnerability analysis are different in responsibility, methods and result.

The security architecture description shall describe all properties of the TOE and the TSF and all security mechanisms of the TSF that contribute to enforce the security architecture. The security mechanisms specific for enforcement of security architecture properties may be fully described:

- in the ARC documents or,
- in the TDS documentation and the ARC documentation refers to these descriptions.

Note that some security mechanisms are spread across the whole implementation and cannot be expressed or are not easily expressible within TDS documents and mapping to modules. The description of the security architecture should avoid redundancy with other parts of ADV.

The CC requires the security architecture description being at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document. But this does not imply the same rigor of the presentation in the ARC documents; the use of semi-formal or formal methods is not required for ARC documents. Even though the CC requires the developer to provide a mapping between the TOE design description and the sample of the implementation representation, such mapping is not required for the security architecture description.

The evaluator is reminded that it is the synergy and not the distinction of self-protection, non-bypassability, domain separation and secure initialisation that are in the focus of the ARC documentation.

2.1 Specific points for smartcards and similar devices

For smartcards and similar devices, the TOE physical boundaries are TSFIs. The device surface is the TSFI for physical protection against manipulation. The surface of the IC itself can output physical signals such as electromagnetic emanations that could be used for side channel analysis or input energy used for perturbation like laser attacks. The ports are physical entry or exit points of power supply and physical signals for the TOE that provides access to the TSF. The physical signal contains more information (e.g. timing, signal level) than the data intended to be exchange through the logically defined TSFI. The power supply port is not part of the logical interface but may affect the TSF (e.g. by glitches).

2.2 Specific points for Secure Sub-System in SoC

For 3S in SoC, the TOE boundaries and interfaces with SoC and external memories (if any) are TSFIs. The 3S surface is the TSFI for physical protection against manipulation and in addition interaction with the SoC has also to be considered.

The surface of the 3S, but also the SoC itself, can unintentionally leak physical signals such as electromagnetic emissions that could be used for side channel analysis or alternatively be faulted by input energy used for external perturbations like laser attacks.

The 3S interfaces with the SoC are physical entry or exit points (e.g. from/to power supply, timer, clocks, configurations registers, communication link(s) and physical signals for the TOE) that provide access to the TSF. The physical signal contain more information (e.g. timing, signal level) than the data intended to be exchanged through the logically defined TSFI. The power supply interface is not part of the logical interface but may affect the TSF (e.g. by glitches).

The interfaces with external memories also have to be considered for protection against emanation and perturbation.

Globally,Evaluation results covering the mandatory description for Security Domains, Secure Start-up, Self-protection and Non-bypassability should be included in ETR for integration according to its template [EFI] to allow reuse in case of 3S is included in several SoC.

3 LEVEL OF DESCRIPTION IN ADV_ARC

ADV_ARC.1.1C requires the architecture description to be "*at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document*".

As the expected assurance level is AVA_VAN.5, the level of description corresponds to parameters, actions and error message for TSFI, the module interface level and in some case to implementation specific details. But semi-formal or formal description is not required because it does not bring more comprehensive details.

The security architecture description is based upon security mechanisms (SFR-enforcing entities, mechanisms enforcing the properties, design countermeasures, coding conventions). Each security mechanism must be explained in terms of purpose and behaviour with the exception of SFR-enforcing entities that are described in decomposition documentation.

For Security Mechanisms spread across the whole implementation, it shall be ensured that there is little ambiguity between the description in ADV_ARC and ADV_IMP by providing the principles that have led to their implementation in the code. The security mechanisms description may be illustrated with code sample or example.

4 SECURITY DOMAINS

The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs (cf. ADV_ARC.1.2C).

Domain separation is a property whereby the TSF creates separate security domains on its own and for each untrusted active entity to operate on its resources, and then keeps those domains separated from one another so that no entity can run in the domain of any other.

The security architecture description explains the different kinds of domains that are created by the TSF, how they are defined in terms of resources allocated to each domain, and how the domains are kept separated so that active entities in one domain cannot tamper with resources in another domain.

If the TSF is the only active entity and there are only data structures maintained by the TSF to manage the interactions with the users, the security architecture will describe that there is no security domain available for active entities.

If the TSF provides security domains for other active entities, the TSF shall protect these domains against adverse actions of these potentially-harmful entities on TSF resources. Moreover, the TSF will keep this domain separated from the security domain of other active entities.

If the ARC documentation describes security domains, the allocation and deallocation of the resources for the active entities should be under SFR control (e.g. FDP_ACC: access control). The use of the resources by the active entity in the security domain is outside TSF control. The active entities may use these resources according to their own security policies but they are not allowed usage of other resources outside their security domain. Therefore the domain description provided in the ARC documentation shall meet TSF access control to the security domain resources as expressed by the SFR and the other SFR must not contradict the security domain definition. If the ARC documentation describes security domains in term of resources not controlled by a SFR, that would mean that a SFR is missing.

In case of composite evaluation the applicative layer could rely upon the underlying platform to correctly instantiate the domains that the TOE defines. The developer should list the used security services offered by the platform to support security domain separation and make reference to these services in the description.

4.1 Additions for 3S – Isolation between 3S FW/SW and Composite SW

Secure Sub-System (3S) is implemented as functional block of a SoC and TOE provides its services to Composite SW[5], to other SoC components and potentially external world but isolated from the other SoC components and based on physical and/or logical mechanisms".

Therefore the ARC documentation describes security domains to handle TSF data and User data and specifies the allocation and deallocation of the resources inside the 3S or external memories (if used).

It should be stated which security domains are used by the 3S to provide services to the different (if any) composite SW through interfaces and to isolate any composite SW from another one.

4.2 Additions for 3S – Usage of dedicated external NVM

In order to extend the capability of the 3S in SoC, dedicated external NVM can be used to store TSF data and/or code and User data and/or code.

A description of the content of the dedicated external memory shall be provided including its type (passive or secure) as described in the PP configurations; where passive means that 3S is responsible of the protection of the content of the NVM and secure means that the NVM has defined its own way to assume protection[6] of its content, potentially in collaboration with the 3S. Important is to address the separation between the different subjects using memory areas.

5 SECURE START-UP

The security architecture description shall describe how the TSF initialization process is secure (in accordance with ADV_ARC.1.3C). The information provided in the security architecture description relating to TSF initialisation is directed at the process bringing the TSF from the "down" state (e.g. power-off or after reset) into an initial secure state (i.e. when all parts of the TSF are operational). For smart cards and similar devices:

- parts of the TSF may be active even in power off e.g. physical protection against undetected manipulation;
- parts of the TSF may be temporally deactivated e.g. in power save modes.

The goal of the secure initialisation process of smart cards and similar devices is to enforce the security objectives even while some TSF parts are not active (i.e. during power off or power save modes) or in activation process (e.g. start-up) or in deactivation process (e.g. transition into power save mode). The secure initialisation process requires that self-protection and non-bypassability are ensured during these transitions. This implies that in any point of time the TOE function is not available if the TSF parts protecting this function are not activated.

The secure initialisation process will be implemented by specific security features or security functionality not directly following SFRs. This specific security functionality and their security mechanisms may be not described in other ADV assurance families. The objective of the ARC documentation for secure initialisation is to provide all the information required to treat these components as part of the TSF.

The secure initialisation process may implement mechanisms protecting the confidentiality or checking the integrity of the implementation of other TSF. Some mechanisms may be not needed after secure initialisation and shall be protected against misuse.

If external interfaces of the initialisation process are fully described as TSFI in terms of actions in ADV_FSP.4 and beyond or the mechanisms as part of the TSF are described in terms of purpose and interactions of modules in ADV_TDS.3 and beyond they do not have to be described again.

5.1 Additions for 3S – Recovery when TSF code is stored in external NVM during start-up sequence

When external NVM is used to store TSF code using a TSF SW image representation, it is necessary during start up sequence to download part or full TSF SW image to decipher it to be able to execute TSF code stored externally.

In addition to confidentiality of TSF code, the integrity, authenticity and freshness shall be checked.

In case of loss of integrity, authenticity or freshness discovered during start up sequence, at least a part of TSF code cannot be loaded and executed, meaning that TSF is executed in "degraded mode". Such situation shall be described as a failed secured state of the start-up sequence.

[5] Composite SW is the software of composite product (as defined in [PP 3S in SoC]).
[6] It is assumed that secure NVM is also evaluated at the same time or independently.

In such situation, the TOE may provide a recovery mode accessible on given conditions. During start-up sequence; in such mode, the TOE may perform automatic operations to recover the damaged area of the external NVM or even better restore a valid version of the TSF SW image. Such situation may also be solved by combination of TOE and TOE environment services. Such recovery mode, available operations and conditions allowing their execution have to be explained in ADV_ARC.

6 SELF-PROTECTION

The component ADV_ARC.1.4C requires that the security architecture description demonstrates that the TSF protects itself from tampering.

Self-protection refers to the ability of the TSF to protect itself from manipulation from external entities that may result in changes to the TSF, so that it no longer fulfils the security objectives or SFRs.

Tampering with the TSF may be realized by untrusted active entity running on behalf of an external entity. Mechanisms that provide domain separation to define a TSF domain that is protected from other (user) domains would be identified and described.

Within the Technical Domain of smart cards and similar devices, the TOE physical boundaries from which an external entity may intervene are the ports and the surface of the IC. The ports are physical entries of the TOE supporting logical interface that provide access to the TSF for physical parasitic signals. The surface of the chip may be also an entry point for physical parasitic signals. These signals may induce a modification of the stored code & data or of the correct execution of the code.

The functional requirement class FPT (Protection of TSF) contains families of functional requirements that relate to the integrity and management of the mechanisms that constitute the TSF and to the integrity of TSF data. Components from the class FPT are necessary to provide requirements that the SFPs in the TOE cannot be tampered.

Self-protection can therefore not generally be achieved by a mere implementation of an SFR but other security mechanisms may be added and collaborate with the security mechanisms implementing the SFR.

Self-protection of the TSF will be achieved by:

- security mechanisms: the ability of each security mechanism to contribute to the protection against direct attacks;
- binding of security mechanisms: the ability of the security mechanisms to work together in a way that is mutually supportive and provides an integrated and effective whole;
- combination of hardware and software security mechanisms.

The initialisation process shall guarantee that the TSF is in an initial secure state and has not been spoofed by any means. The developer shall explain how the initialization process checks the TSF code integrity. The integrity of the initialisation process code shall also be checked during this process.

In some cases the TOE starts up in a low-function mode, a mode whereby untrusted users are able to login and use the services and resources of the TOE. In this mode the code does not run in the evaluated configuration and these services are no more accessible.

In this case the security architecture description shall include an explanation of how the TSF is protected against this code in the evaluated configuration:

- what prevents this code from running;
- what prevents those services from being accessible.

In case of a composite evaluation the platform could provide security services that contribute to the self-protection in cooperation with the application layer security mechanisms. The developer shall list the used security services offered by the platform and make reference to them in the following analysis.

The developer shall describe the security mechanisms and their collaboration to protect the TSF from tampering. The developer shall provide a description on how the TOE reacts in presence of the relevant attacks listed in the state-of-the-art document Application of Attack Potential to Smart Cards and provide a conclusion.

6.1 Additions for 3S – Isolation between 3S and hosting SoC

Interaction between the 3S and the SoC through interfaces may contradict self-protection property. Therefore, features implemented in the 3S to protect itself against perturbation from the host SoC shall be described.

In particular, if some resources (e.g. clock) are shared with other components of the SoC, it should be explicitly described as such. This description should help to determine the level of isolation and sharing between 3S and SoC.

It should be stated how the 3S to provide/get services to/from the other components of the SoC through interface and how the remaining part of the SoC interacts with the 3S.

6.2 Additions for 3S – Power Management

Power management of a typical SoC that has more options than a standalone smart card. For example:

1) TOE has its own power domain with direct power supply from the outside, without contribution of host SoC to the TOE power line.

2) TOE has its own power domain but the TOE and the host SoC may influence each other.

3) TOE and host SoC share a power domain.

Therefore, security mechanisms to protect against glitch attacks are depending on the configuration of the 3S and the SoC for power management and shall be described..

6.3 Additions for 3S – Recovery when TSF code is stored in external NVM during execution

When external NVM is used to store TSF code using a TSF SW image representation, it may be necessary during execution time to download part or full TSF SW image to decipher it to be able to execute TSF code stored externally.

In addition to confidentiality of TSF code, the integrity, authenticity and freshness shall be checked.

In case of loss of integrity, authenticity or freshness, at least a part of TSF code cannot be executed, meaning that TSF is executed in “degraded mode”. Such situation shall be described as a failed secured state of execution time.

In such situation, the TOE may provide a recovery mode accessible on given conditions. In such mode, the TOE may perform self-protection operations to recover the damaged area of the external NVM or even better restore a valid version of the TSF SW image. Such situation may also be solved by combination of TOE and TOE environment services. Such recovery mode, available operations and conditions allowing their execution have to be explained in ADV_ARC.

6.4 Additions for 3S – Usage of (passive or secure) external NVM shared with SoC

In order to extend the capability of the 3S in SoC, external NVM can be used to store TSF data and/or code and User data and/or code, when other SOC components may access to this external NVM.

A description of the content of the external memory shall be provided including its type (passive or secure) as described in the PP configurations; where passive means that 3S is responsible of the protection of the content of the NVM and secure means that the NVM has defined its own way to assume protection[7] of its content, potentially in collaboration with the 3S.

It shall be stated which external memory (or part of NVM) is dedicated to the 3S or shared with other components of the SoC.

Any binding mechanism between the 3S and the external NVM shall be described preventing replacement of NVM or content by another.

In addition, any mechanism to assume freshness[8] of external memory content requires a description including existence of secure channel between the 3S and the external NVM explaining contribution to self-protection property.

7 NON-BYPASSABILITY

The component ADV_ARC.1.5C requires that the security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality

Non-bypassability is a property that the security functionality as specified by the SFRs is always invoked and cannot be circumvented when appropriate for that specific mechanism (cf. CC).

[7] It is assumed that secure NVM is also evaluated at the same time or independently.
[8] See definition in Glossary [PP3SinSoC].

7.1 TSF always invoked

Non-bypassability means firstly that there is no possibility to bypass the SFR-enforcing entity by using unexpected and undocumented paths in the design. Any possibility to bypass the TSF is therefore attributed to a flaw in the design or implementation.

The functional specification shall describe all actions associated with each TSFI (ADV_FSP.4.4C) and the design shall describe each SFR-supporting or SFR-non-interfering module in terms of its purpose and interaction with other modules (ADV_TDS.3.9C). In this case all modes or operations of TSFI are documented at a sufficient level to provide evidence of non-bypassability by exploiting a flaw in the design.

Secondly, non-bypassability requires that no Functional Interface can be used to violate the TOE security objectives, to circumvent SFR or to conflict with SFR. When Functional Interfaces exist the developer shall list them and explain either why they have no interaction with the TSF or why they are not providing a path for circumventing the TSF. In this case domain separation description (see the corresponding section) may bring evidence of non-bypassability.

Thirdly, non-bypassability deals with those cases where the attacker has only logical access to the TOE as opposed to the case of “tampering” which is to be countered by self-protection (see the corresponding section).

The developer shall describe the security mechanisms and their collaboration to protect the TSF from software attacks exploiting an insufficient design or implementation to meet the TOE security objectives. The developer shall provide a description on how the TOE reacts in presence of the relevant attacks listed in the state-of-the-art document Application of Attack Potential to Smart Cards and provide a conclusion.

7.2 Addition of Non-bypassability features in 3S within SoC

3S – Interaction between 3S and hosting SoC

Presence of unexpected and undocumented interaction between the 3S and its hosting SoC may contribute to alteration of non-bypassability property. It has to be considered during ADV_ARC review.

Binding or Trusted channel between 3S and External Non-volatile memory

Presence of binding features between the 3S and external NVM may contribute to alteration of non-bypassability property. It has to be considered during ADV_ARC review.

The ability to circumvent any communication channel between the 3S and external NVM or the SoC has also to be considered during ADV_ARC review.

7.3 Side channel

Side channels are unenforced signalling channels carrying information about internal secrets, states or processes provided by monitoring of the processing of any object containing or related to this information (cf. CEM). The information may be contained in any observable physical value as power consumption of the device, voltage and timing on ports of the output interfaces, electromagnetic emanation on IC surface. The signals of output ports may contain more information than the data intended to be exchange through the logical interface defined in the TSF documentation. The power supply interface and the electromagnetic emanation through the IC surface are not intended for information output at all but may carrying information.

The side channels bypass the TSF because they leak any information intended to be kept secret. The secret information includes but is not limited to authentication reference data (e.g. for PIN verification), symmetric secret or asymmetric private cryptographic keys, timing of data processing enabling other attacks.

The developer shall describe the countermeasures implemented in order to prevent potential side channels of the TOE in the intended operational environment. The side channel analysis as part of the evaluator’s vulnerability analysis shall determine whether side channel exist and are exploitable i.e. these countermeasures are effective.

The developer and the evaluator should consult the SFRs and the security objectives they enforce in order to determine whether an unintended information flow bypass the TSF or not. The implementation of a symmetric message authentication code calculation will keep the confidentiality of the key but it may or may be not required to protect the confidentiality of the processed user data. Therefore the decision about bypass of the TSF by leaking information about the processed user data depends on the security objective enforced by the SFRs.

7.4 Addition of Side channel protection in 3S and hosting SoC

The TOE description shall provide evidence of the security mechanisms used during execution of software in 3S that have been implemented to mitigate or thwart side channel attacks performed using measurement of power consumption at 3S interface or on whole SoC, but also measurement of emissions around the 3S and the complete SoC. Such leakage (obtained using side channel attacks) shall be minimized to avoid any retrieval of values of sensitive assets stored in or manipulated by the 3S.

As defined in § Additions for 3S – Power Management, there may be several configurations of the 3S and SoC for power management. Therefore mitigation of attacks on power consumption shall be described addressing 3S and / or SoC according to power configuration.

Concerning mitigation of attacks on emanation concerning 3S and SoC, it is mandatory to explain the independence between the 3S self-protection mechanisms from the influences of the host SoC, or minimized dependences to address existing relations.

ABOUT ENISA

The mission of the European Union Agency for Cybersecurity (ENISA) is to achieve a high common level of cybersecurity across the Union, by actively supporting Member States, Union institutions, bodies, offices and agencies in improving cybersecurity. We contribute to policy development and implementation, support capacity building and preparedness, facilitate operational cooperation at Union level, enhance the trustworthiness of ICT products, services and processes by rolling out cybersecurity certification schemes, enable knowledge sharing, research, innovation and awareness building, whilst developing cross-border communities. Our goal is to strengthen trust in the connected economy, boost resilience of the Union's infrastructure and services and keep our society cyber secure. More information about ENISA and its work can be found at www.enisa.europa.eu.

ENISA
European Union Agency for Cybersecurity

Athens Office
1 Vasilissis Sofias Str
151 24 Marousi, Attiki, Greece

Heraklion office
95 Nikolaou Plastira
700 13 Vassilika Vouton, Heraklion, Greece

enisa.europa.eu