PSA Certified™ Attack Methods Version 1.2

Document number:	JSADEN004
Version:	1.2
Release Number:	01
Author	PSA JSA Members:
	Applus+ Laboratories
	Arm Limited
	CAICT
	ECSEC Laboratory Inc.
	Prove & Run S.A.S.
	Riscure B.V.
	SGS Brightsight B.V.
	TrustCB B.V.
	UL TS B.V.
Authorized by:	PSA JSA Members
Date of Issue:	25/04/2022

Abstract

PSA Certified is the independent security evaluation scheme for Platform Security Architecture (PSA) based IoT systems. It establishes trust through a multi-level assurance program for chips containing a security component called a Root of Trust (PSA-RoT) that provides trusted functionality to the platform. The multi-level scheme has been designed to help device makers and businesses get the level of security they need for their use case.
PSA Certified is a fixed time, test laboratory based, evaluation of the PSA-RoT. It is aimed at IoT devices that need to protect against scalable software attacks. The documents include: a Protection Profile (PP) that describes the Target of Evaluation, its assets, the security objectives and security functions that will be evaluated; an Evaluation Methodology (EM) that details how the evaluation will be carried out, and an Attack Methods (AM) document describing the attacks in scope.

Developers submit their PSA-RoT to an approved test laboratory, listed on **www.psacertified.org**, for Level 2, Level 2+SE or Level 3 evaluation and receive an Evaluation Technical Report. If the PSA-RoT is assessed as passing and approved by the independent Certification Body, a digital certificate will be issued on the PSA Certified website.

Keywords

PSA Certified Level 2, PSA Certified Level 2+SE, PSA Certified Level 3, Certification, IoT, Platform Security Architecture, Questionnaire, PSA, Security

Non-Confidential Proprietary Notice

Arm Limited. Company 02557590 registered in England.
110 Fulbourn Road, Cambridge, England CB1 9NJ.

Contents

1 About this document

1.1 Current Status and Anticipated Changes

Current Status: Final

1.2 Release Information

The change history table lists the changes that have been made to this document.

Date	Version	Confidentiality	Change
15/03/2022	1.2	JSA Confidential	Additional examples and reorder Expanded to cover Level 2+SE and Level 3.
18/02/2020	1.1	Non-confidential	Clarifications for PSA L2 Ready and new template
25/09/2019	1.0	Non-confidential	Initial version, approved by JSA members

1.3 References

This document refers to the following informative documents.

1.3.1 Normative references

Ref	Doc No	Author(s)	Title
[PSA-EM]	JSADEN003	ARM JSA	PSA Certified: Evaluation Methodology
[PSA-L1]	JSADEN001	ARM JSA	PSA Certified: Level 1 Questionnaire
[PSA-PP]	JSADEN002	ARM JSA	PSA Certified Level 2 Lightweight Protection Profile

1.3.2 Informative references

Ref	Doc No	Author(s)	Title
[GP-ROT]	GP_REQ_025	GlobalPlatform	Root of Trust Definitions and Requirements, Version 1.1, Public Release, June 2018
[JIL-APSC]		Version 2.9 January 2013	Joint Interpretation Library – Application of Attack Potential to Smartcards
[PSA-FF]	ARM DEN 0063A	ARM	ARM® Platform Security Architecture Firmware Framework and RoT Services – M-profile
[PSA-SM]	ARM DEN 0079	ARM	PSA: Device Security Model
[GP-ROT]	GP_REQ_025	GlobalPlatform	Root of Trust Definitions and Requirements, Version 1.1, Public Release, June 2018

1.4 Terms and Abbreviations

This document uses the following terms and abbreviations

Term	Meaning
Application firmware	The main application firmware for the platform, typically comprising a System software and application tasks. PSA provides no isolation services for this firmware, although the System software may make use of available hardware support to provide internal isolation of operation
Application Specific Software	Software that provides the functionality required of the specific device. This software runs in the Non-Secure Processing Environment, making use of the System software, Application RoT Services and PSA-RoT Services.
Application Root of Trust	This is the security domain in which additional security services are implemented. See *PSA Security Model* [PSA-SM] for details
Application Root of Trust Service	Application specific security service(s) that are not defined by PSA. Such services execute in the Secure Processing Environment and are required to be in Secure Partitions.
Critical Security Parameter	Secret information, with integrity and confidentiality requirements, used to maintain device security, such as authentication data (passwords, PIN, certificates), secret cryptographic keys, etc. In some contexts, these are classed as assets.
Evaluation laboratory	Laboratory or facility that performs the technical review of questionnaires submitted for PSA Certified. The list of evaluation laboratories participating to PSA Certified can be found on www.psacertified.org
JTAG	Joint Test Action Group
Hardware Unique Key (HUK)	Secret and unique to the device – this symmetric key must not be accessible outside the PSA Root of Trust
Non-secure Processing Environment (NSPE)	The processing environment that executes the non-secure System software and Application Specific Software. PSA requires the NSPE to be isolated from the SPE. Isolation between partitions within the NSPE is not required by PSA though is encouraged where supported.
PSA	Platform Security Architecture
PSA Certification Body	The entity that receives applications for PSA security certification, issues certificates, maintains the security certification scheme, and ensures consistency across all the evaluation laboratories.
PSA Functional APIs	Foundations from which security services are built, allowing devices to be secure by design. Three sets of APIs have been defined, so far, and include Crypto, Secure Storage and Attestation

PSA Functional API Certification	PSA defined Application Programming Interfaces on which security services can be built. APIs defined so far include Crypto, Secure Storage and Attestation.
Immutable Platform Root of Trust	The minimal set of hardware, firmware and data of the PSA-RoT, which is inherently trusted because it cannot be modified following manufacture. There is no software at a deeper level that can verify that it as authentic and unmodified.
PSA Root of Trust	The PSA defined combination of the Immutable Platform Root of Trust and the Updateable Platform Root of Trust and considered to be the most trusted security component on the device. See [PSA-SM].
PSA Root of Trust Service	This is a Root of Trust Service within the PSA Root of Trust domain
Root of Trust (RoT)	This is the minimal set of software, hardware and data that is implicitly trusted in the platform – there is no software or hardware at a deeper level that can verify that the Root of Trust is authentic and unmodified. See *Root of Trust Definitions and Requirements* [GP-ROT]
Updateable Platform Root of Trust	The firmware, software and data of the PSA-RoT that can be securely updated following manufacture.
Platform Root of Trust Service(s)	PSA defined security services for use by PSA-RoT, Application RoT Service(s) and by the NSPE. Executes in the Secure Processing Environment and may use Trusted Subsystems. This includes the services offered by the PSA Functional APIs.
Secure Partition	A Partition in the Secure Processing Environment.
Secure Processing Environment Partition Management	Management of the execution of software in Secure Partitions. Typical implementations will provide scheduling and inter-partition communication mechanisms. Implementations may also enforce isolation between the managed Secure Partitions.
Secure Processing Environment (SPE)	The processing environment that executes the PSA-RoT, the PSA-RoT Services, and any Application RoT Service(s).
SiP	System in Package
SoC	System on Chip
Secure boot	Secure boot is technology to provide a chain of trust for all the components during boot
System Software	NSPE software that may comprise an operating system or some run-time executive, together with any middleware, standard stacks and libraries, chip specific device drivers, etc., but not the application specific software.

Trusted Subsystem	A security subsystem that the PSA-RoT relies on for protection of its critical security parameters, or that implement some of its services.

1.5 Feedback

The PSA JSA Members welcome feedback on its documentation.

If you have comments on the content of this documentation, send an e-mail to psacertified@arm.com. Give:

- The title (PSA Certified Attack Method).
- The number (JSADEN-004) and version.
- The page numbers to which your comments apply.
- A concise explanation of your comments.

PSA JSA Members also welcome general suggestions for additions and improvements.

Note

PDFs are tested only in Adobe Acrobat and Acrobat Reader and cannot guarantee the appearance or behavior of any document when viewed with any other PDF reader.

2 Introduction

2.1 Document Context

This document provides guidance as to which attack methods must be considered in PSA Root of Trust evaluation according to PSA Certified, including Level 2, Level 2 Component, Level 2+SE, and Level 3.

By describing the key factors of these methods, a harmonization of vulnerability assessment and penetration testing in evaluations can be achieved.

2.2 Targeted Audience

This document is directly aimed at Evaluation Laboratories, who perform PSA Certified evaluations according to the security requirements set in [PSA-PP].

It can also be used by Chip Vendors, who develop the chip and the PSA components for the Secure Processing Environment, to design security measures able to withstand attacks described in this document.

2.3 PSA Certified Level 2 Ready Evaluation

This document considers a pre-certification evaluation of FPGA or development-based systems, which provide reference designs for ASIC or custom chip, but which may not be able to meet all nine security functions of the protection profile [PSA-PP]. In this case, only the claimed security functions are tested by the Evaluation Laboratory who issues the Evaluation Technical Report. No Level 2 certificate is generated for a Level 2 Ready evaluation, but the Developer can obtain the rights to use a specific “PSA Certified Level 2 Ready” logo and showcase its solution on **www.psacertified.org** .

Such a logo could be used to demonstrate, for example, the benefit of software security assurance offered from an evaluated FPGA based system for development of secure AROTs, RTOS or device while maximizing chances of passing PSA Certified Level 2 certification for future ASIC or custom chips based on the FPGA reference design.

2.4 How to Use this Document

This document first provides the definition of the rating factors that will be used by Evaluation Laboratory to quote identified attacks.

Then, this document describes the classes of attacks that shall be considered during the evaluation. For each evaluation it must be decided which of the attack methods are applicable for the product under evaluation and how the attacks should be best implemented. It might be possible to exclude whole classes of attacks just by considering specific properties of the TOE, such as FPGA systems considered for Section 2.3.

Exclusion of classes of attacks applies for PSA Certified Level 2 Ready certification, see Section 2.3.

3 Scope

3.1 Components

The scope for a PSA Certified Level 2 evaluation, or Target of Evaluation (TOE), is the combination of the hardware and firmware components supporting a device compliant with PSA specification. The considered hardware may be a System-in-Package (SiP), a System-on-Chip (SoC) integrated on a board, or similar set-up.

The hardware is in the scope of the security evaluation as it provides security features, such as immutable storage or protection of JTAG, which are essential for ensuring the security of the PSA implementation. The case of hardware limitations of FPGA systems is considered in PSA Certified Level 2 Ready evaluations.

The PSA platform components that are in the scope of the security evaluation, as described in [PSA-FF], are:

- PSA updateable Root of Trust, such as Software isolation framework, protecting more trusted software from less trusted software, Generic services such as binding, initial attestation, generic crypto services, FW update validation.
- PSA immutable root of trust, for example Boot ROM, Root secrets and IDs, Isolation hardware, Security lifecycle management and enforcement. This component cannot be updated.
- Trusted Subsystems used by the PSA root of trust, such as security subsystems, trusted peripherals, SIM, or SE, which include both hardware and software components are also in the scope of evaluation.

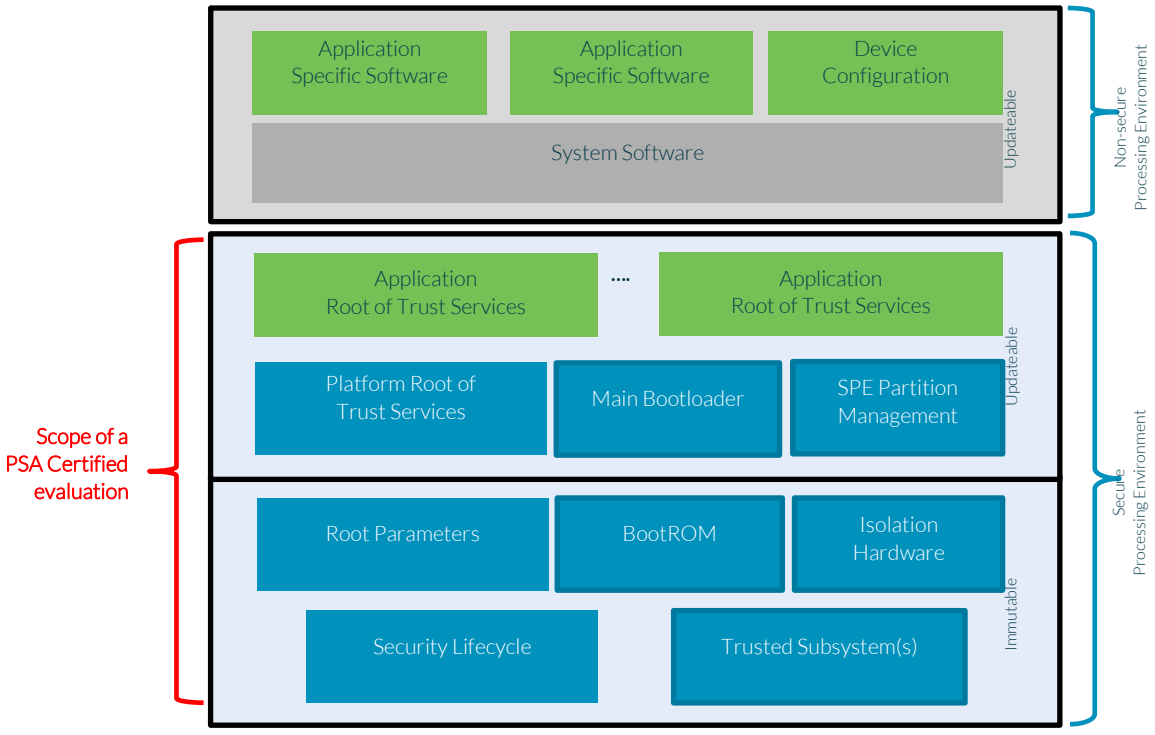

Figure 1: Scope of PSA Certified Level 2

3.2 Interfaces

The following interfaces constitute a boundary between the TOE and its environment and can be used to interact with the TOE and perform attack(s):

- API between Application RoT and PSA RoT within the SPE
- API between NSPE and SPE
- Interface between PSA RoT and external devices

4 Identification of factors

4.1 How to compute an attack

Attack potential calculation distinguishes between the cost of “identification” (demonstration of the attack) and the cost of “exploitation” (repetition of the attack on another instance of the TOE, for example once it has become public).

For each attack type, there can be multiple ways of achieving the result. For example, the attacker can use standard equipment and spend extra time on an attack or use custom equipment and take less time. Each potential method is known as an attack path. When evaluating attacks, the lab will look at the path with the lowest attack potential. However, when testing, they may use an attack path with a higher potential but lower elapsed time as a proxy, to complete testing in the allocated time.

Attack path identification as well as exploitation analysis and tests are mapped to relevant factors, based on [JIL-APSC. Currently, PSA uses the JIL table directly. However, the JSA retain the right to amend the table if it proves not to be suitable for the use cases PSA covers.

To complete an attack potential calculation the points for identification and exploitation must be added as both phases together constitute the complete attack. When presenting the attack potential calculation in the Evaluation Test Report, the evaluators will make an argument for the appropriateness of the parameter values used and will therefore give the developer a chance to challenge the calculation before certification.

4.1.1 Elapsed time

Elapsed time for identification is the time required to create the attack, and to demonstrate that it can be successfully applied to the TOE (including setting up or building any necessary test equipment).

Elapsed time for exploitation is the time required to achieve the attack on another instance of the TOE using the analysis and techniques defined in the identification part of an attack.

For this factor, all time categories defined in [JIL-APSC] are used.

It may not be possible for the Evaluator to perform a full attack in the workload allocated for the evaluation. The Evaluator may extrapolate a quotation for the full attack, with the proper rationale, based on the performed partial tests.

Not Practical is used when the attack path is not exploitable within a timescale that would be useful to an attacker. This is normally because the exploitation phase would take too long – for example, an attack that exposes only a TLS session key is not practical other than for eavesdropping unless it can be used in the context of that session. It would be unusual for an attack to be labelled not practical in the identification phase, as obviously the evaluator has identified the attack, unless the device contains sufficient countermeasures to defeat it.

Timing assumes the attacker is working eight hours a day or running scripts for twenty-four hours a day.

Timing in the identification phase must include all the time to write and test the script and to run it until it achieves a first break.

If this is probabilistic, for example brute forcing an unknown password, take the time required for a 50/50 chance, so for example, half the time required to search the entire range.

When in doubt, the laboratory should discuss this with the CB

4.1.2 Expertise

For this factor, three types of experts are defined:

- **Laymen** are unknowledgeable when compared to experts or proficient persons, with no particular expertise.
- **Proficient** persons are knowledgeable in that they are familiar with the security behavior of the product.
- **Experts** are familiar with the underlying algorithms, protocols, hardware, structures, etc., implemented in the product or system type and the principles and concepts of the security employed.

4.1.3 Knowledge of the TOE

The following classification is to be used:

- **Public information about the TOE** (or no information): Information is considered public if it can be easily obtained by anyone (for example, from the Internet) or if it is provided by the vendor to any customer.
- **Restricted information concerning the TOE**: Information is considered restricted if it is distributed on request and the distribution is registered or subject to contractual terms. An example might be the functional specification provided by the vendor on payment of a license fee.
- **Sensitive information about the TOE**; Information is considered sensitive if it obtained by inappropriate means, for example, knowledge of internal design, which may have to be obtained by “social engineering” or exhaustive reverse engineering. An example might be High-Level Design, Low-Level Design information or the Source Code.
- **Critical information about the TOE**. Information is considered critical if it is known by only a few individuals, access to which is very tightly controlled on a strict need to know basis and individual undertaking.

4.1.4 Access to TOE

This factor refers to the number of devices with the TOE that are necessary during the identification or exploitation phase. Numbers defined in [JIL-APSC] apply.

Availability of samples (in terms of time and cost) needs to be considered as well as the number of samples needed to carry out an attack path.

The attack scenario might require access to more than one device with the TOE because:

- the attack succeeds only with some probability on a given device such that several devices need to be tried out,
- the attack succeeds only after having destroyed several devices (on average),
- the attacker needs to collect information from several copies of the TOE.

PSA certified assumes that the attacker can obtain samples in the pre-provisioned state. If the vendor only ships devices in a locked state, they might be able to argue for additional points for access to open samples.

4.1.5 Equipment

Equipment refers to the equipment that is required to identify or exploit a vulnerability.

To clarify equipment category, price and availability must be considered.

- **Standard equipment** is equipment that is readily available to the attacker, either for the identification of vulnerability or for exploitation of an attack. This equipment can be readily obtained—for example, at a nearby store or downloaded from the Internet. The equipment might consist of simple attack scripts, personal computers, SW debuggers, JTAG probes, pattern generators, simple optical microscopes, power supplies, oscilloscopes, or simple mechanical tools.
- **Specialized equipment** is equipment that is not readily available to the attacker but could be acquired without undue effort. This could include purchase of moderate amounts of equipment (for example, dedicated electronic cards, specialized test bench, protocol analyzers, specialized JTAG probes, etc.) or development of more extensive attack scripts or programs.
- **Bespoke equipment** is equipment that is not readily available to the public as it might need to be specially produced (for example, very sophisticated software) or because the equipment is so specialized that its distribution is controlled, possibly even restricted. Bespoke equipment, which can be rented, might have to be treated as specialized equipment. Software that has been developed during the identification phase is considered as bespoke equipment; it must not additionally be considered for in the exploitation phase.

Over time it is inevitable that equipment becomes cheaper and more commonplace. Therefore, equipment previously considered bespoke, could become specialist and specialist could become standard.

4.2 Attack quotation table

Table 1 provides the number of points for each factor level. Values for each factor are identical to the ones found in the JHAS attack quotation table for smart cards [JIL-APSC], although the Open Sample factor is missing from the table below as it is not applicable in the context of PSA Certified.

Factors	Identification	Exploitation
Elapsed time		
≤ one hour	0	0
≤ one day	1	3
≤ one week	2	4
≤ one month	3	6
> one month	5	8
Not practical	*	*
Expertise		
Layman	0	0
Proficient	2	2
Expert	5	4
Knowledge of the TOE		
Public	0	0
Restricted	2	2
Sensitive	4	3
Critical	6	5
Access to TOE		
< 10 samples	0	0
< 30 samples	1	2
< 100 samples	2	4
> 100 samples	3	6
Not practical	*	*
Equipment		
None	0	0
Standard	1	2
Specialized	3	4
Bespoke	5	6
Multiple Bespoke	7	8

Table 1: Table for the rating factors

The final attack potential of an attack is equal to the sum of the points for identification plus the sum of the points for exploitation.

PSA level	Attacks that must be considered	Attacks that are out of scope
PSA Level 2 PSA Level 2+SE	Attacks with a rating of 0 – 15 inclusive	Attacks with a rating higher than 15
PSA Level 3	Attacks with a rating of 0 – 20 inclusive	Attacks with a rating higher than 20

For a device to pass, it must be shown to resist all attacks in the relevant range. If the evaluator can show that it is vulnerable to one attack in the relevant range then the evaluation fails, even if it resists some attacks of a higher potential.

When an attack is analyzed as being just above the threshold the laboratory must conduct the test, to ensure that this device meets those typical numbers.

4.3 Combining Attacks

In many cases, a complete attack might require the attacker to combine two or more techniques to successfully complete the attack. In this case, for each factor, the examiner must look at each category in this way and calculate the score for the combined attack.

While it will usually be higher than any of the component attacks, it may not be. For example, an attack that takes two days is rated at 3 points. If this needs to be combined with another method that also takes two days, the total time is four days, so still less than one week and therefore still 3 points. Therefore, adding the attack potentials together, will overstate the difficulty by 3 points.

The correct total for the combined attacks is always a lower value than that reached by simply add the totals for the two attacks.

5 Attack Methods

This chapter looks at the attack methods arranged by security function from the CSPN Protection Profile. For each security function there are three example attack paths. One which should be blocked by any system capable of completing a Level 1 questionnaire. One which should be blocked by a system aiming at Level 2 certification and one which need only be blocked by a Level 3 system.

For each attack path there is a description, a calculation of the attack potential and an example of the tests the lab might perform to verify the system does indeed protect against this attack.

The exact tests to be performed will be determined by the laboratory after it has performed its risk analysis. There is no requirement to perform all the tests, the test plan should concentrate on the attack paths that are most likely to succeed.

The same attacks will need to be addressed by a device submitted under the SESIP scheme – though the names of the security functions do not align.

In all cases, due to the size and complexity of the NSPE code, it is assumed that it is possible to find an exploit in the NSPE that permits the attack to run rogue code in this execution environment. The calculated attack potentials do not include any points for the effort to do this – they only include the costs of the actual attack on the Secure Processing Environment – which is the target of the evaluation.

5.1 INITIALIZATION

5.1.1 No initialization check performed

No chip should be able to pass the level 1 questionnaire if they do not have a secure boot process. Although there is no requirement to supply a Level 1 questionnaire, the attack potential is also so low that it would be an automatic failure for a Level 2 or Level 3 evaluation.

Example Tests

The evaluator reads the documentation to determine how much of the image is covered by the test. They then verify this by modifying either some part of the image covered by the test or the check value and resetting the device and observing that the device rejects the modified image.

5.1.2 Error in initialization process

On reading the developer documentation, the attacker finds a flaw in the signature verification process and constructs an image with a signature that does not match the image, or a valid signature with an incorrect certificate chain that is due to a flaw in the design accepted by the boot sequence.

If such a flaw exists, an expert could take a few days to discover it, but may require knowledge of the exact process used – which is normally not included in standard documentation, so is classed as restricted. Only a single sample is needed and no equipment.

Factor	Comment	Identification	Exploitation
Elapsed time	Assume that the attack leads to the ability to load malicious image on any device.	≤ one week (2)	≤ one hour (0)
Expertise	Expert knowledge is required identify the flaw and develop an attack script. A layman attacker can execute the script.	Expert (5)	Layman (0)
Knowledge of the TOE	Identification may require restricted knowledge. Exploitation only assumes public knowledge of the TOE.	Restricted (2)	Public (0)
Access to TOE	Only few samples of the TOE are needed.	< 10 (0)	< 10 (0)
Equipment	No equipment is needed.	Standard (1)	Standard (2)
Open samples	No open sample is required	Public (0)	NA
Sub-total		10	2
Total		12	

Example Tests

The evaluator will read the technical documentation.

The evaluator will verify the authentication methods.

The evaluator will verify the code and ensure it does not include any known flaws (for example use of strcmp to compare hashes).

Unless supplied by the vendor, the evaluator will create test cases for each part of the evaluation chain, incorrect signature, hash in signature does not match the code, incorrect certificate, invalid dates etc. Where the product uses interactive approval process, such as Online certificate Status protocol (OCSP), they shall determine how the product respond if the service cannot respond.

5.1.3 Glitch against initialization

The attacker manipulates the power or the external clock, or overwhelms the device with a large electromagnetic field, to cause the chip to skip steps in the signature verification so that it incorrectly

accepts an unsigned image. That image then reads critical values from the chip and exposes them to the attacker.

In general, the lab will not create a new malicious image, but will demonstrate that they can load an image with an incorrect signature. For example, by modifying a bit in either the image or the signature.

In the past, when designs had no protection, then a glitch could be triggered by a wide band source, such as a piezo sparker used to light gas fires. However, chip designers now routinely include countermeasures that monitor power and frequency, so more specialized equipment is required – though nothing not found in a well-equipped home workshop – pulse generators, coils etc.

At PSA Level 2 physical attacks on the TOE are out of scope during the exploitation phase. Therefore, at Level 2, this attack is only considered if it leads to the exposure of a class secret, which can be used to mount a simple attack, for example if it exposes a symmetric key that can be used to authorize loading new images.

Assuming this leads to developing an image that can attack any device, a layman can load it in under an hour and read out critical data. However, this requires there to be a universal authorization key that can be exposed – so a symmetric MAC rather than an asymmetric signature.

At PSA Level 3, physical attacks on the device are permitted in the exploitation phase. Therefore, the evaluator must consider both attacks that expose a secret, and those that develop an attack method that needs to be repeated on each target.

Glitch revealing a class secret

This attack scenario assumes that the second stage code is stored in external flash and is signed by not encrypted. Therefore, the attacker only needs to bypass the signature verification. If the code is stored in internal storage or is encrypted – the attacker would need to attack the update mechanism as well as the secure boot.

Determining when and how to perturb the power supply, manipulate the clock or fire the EM pulse to skip the signature checks requires careful preparation and planning. The JSA believes it will take an expert person more than a week with specialized equipment. Only public knowledge of the TOE is needed and only one sample. They would then need to discover the correct key and build an attack image that can be loaded onto the target device.

In practice, a well-designed chip has defense in depth, and a successful exploit may take multiple glitches at separate times to carry out.

Example rating – glitch exposing class secret

Factor	Comment	Identification	Exploitation
Elapsed time	Assume that the attack leads to the ability to load malicious image on any device.	≤ one month (3)	≤ one hour (0)
Expertise	Expert knowledge is required to find the correct timing for the glitch. A layman attacker can execute the application.	Expert (5)	Layman (0)
Knowledge of the TOE	Identification or exploitation only assume public knowledge of the TOE.	Public (0)	Public (0)
Access to TOE	Only few samples of the TOE are needed.	< 10 (0)	< 10 (0)
Equipment	The equipment to perform the glitch attack is specialized. Assume no extra equipment is needed to load the rogue application, other than that required to communicate with the device.	Specialized (3)	Standard (2)
Open samples	No open sample is required	Public (0)	NA
Sub-total		11	2
Total		13	

Even if the glitch required bespoke equipment, if it exposes a class secret that leads to the ability to craft a universal image and thus leads to a simple attack on all instances, it would still be in scope.

Attack requiring a glitch on the target device

In this attack path, the attacker finds a glitching attack that enables them to bypass the signature check and therefore to run modified code, but there is no class secret. Therefore, the recipe must be repeated on each instance that is to be attacked.

The attack potential for identification is the same as in the previous attack. However now the attacker needs to repeat the glitch in the exploitation phase.

This analysis assumes that as the timing is fixed, and it can be recreated after a few attempts using a script prepared by the expert who discovered it. This probably takes more than an hour but less than a day. The attacker needs to be proficient and needs specialist equipment.

Example rating – glitch required to attack target device

Factor	Comment	Identification	Exploitation
Elapsed time		≤ one month (3)	≤ one day (3)
Expertise	Proficient expertise is required to find the correct timing for the glitch. A layman attacker can execute the application.	Expert (5)	Proficient (2)
Knowledge of the TOE	Identification or exploitation only assume public knowledge of the TOE.	Public (0)	Public (0)
Access to TOE	Only few samples of the TOE are needed.	< 10 (0)	< 10 (0)
Equipment	The equipment needed is specialized, both to identify and exploit the glitch.	Specialized (3)	Specialized (4)
Open samples	No open sample is required	Public (0)	NA
Sub-total		11	9
Total		20	

The total attack is 20 so in scope for Level 3, however, if any of the factors was slightly larger, it would become out of scope.

Notes

If the glitch requires a laser, especially if the chip needs to be decapsulated to enable the laser to be focused on the silicon, therefore the attacker may need more samples. The equipment would be classed as bespoke for both identification and exploitation. Also, in general characterization time is longer for laser attacks. This would raise the attack potential beyond 21. If would therefore be out of scope even for Level 3.

Example tests

The evaluator will examine the documentation to determine whether there are class secrets.

The evaluator will write a test that calls the signature verification process repeatedly and runs it while varying power and clock to determine if there are settings that cause invalid results.

The evaluator will then use those settings during the boot process to determine if the glitch can force the module to accept an incorrect signature.

5.2 SOFTWARE_ISOLATION

5.2.1 Misuse of DMA

Attack path

In this example, in a naïve implementation with no protection, the attacker uses a rogue application on the NSPE that has access to a peripheral with an ill-configured DMA also used by the SPE. Through this application, the attacker can break memory isolation between NSPE and SPE and to read or modify sensitive information managed by the TOE.

Once the application is developed, it can be used on other TOEs.

Example rating

Factor	**Comment**	**Identification**	**Exploitation**
Elapsed time	Developing the rogue application and identifying memory that can be accessed with DMA can take up to one week. Exploitation just consists in executing the rogue application.	≤ one week (2)	≤ one hour (0)
Expertise	Proficient expertise is required to develop the application. A layman attacker can execute the application.	Proficient (2)	Layman (0)
Knowledge of the TOE	Identification or exploitation only assume public knowledge of the TOE.	Public (0)	Public (0)
Access to TOE	Only few samples of the TOE are needed.	< 10 (0)	< 10 (0)
Equipment	The attack requires standard equipment, such as a compiler tool chain to build the rogue code and a remote connection, to load and execute the code.	Standard (1)	Standard (2)
Open samples	No open sample is required	Public (0)	NA
Sub-total		5	2
Total		7	

5.2.2 Remote Code Execution on SPE

Attack path

In this example, the attacker is dumping the flash memory that holds the executable image for the TOE to get access to TOE assets. This memory is found not to be visible from NSPE, contrarily to the encrypted secure storage area used by the TOE.

Step 1: Analysis of the PCB

The attacker first needs to open the device and get access to the PCB. He analyses the surface of the PCB and identifies Flash chip. With the datasheet of this chip found on Internet, the attacker determines this chip supports Serial Peripheral Interface (SPI) and finds the corresponding pins.

Step 2: Probing of the Flash

The attacker uses microprobes to connect to the Flash memory SPI, a USB-SPI bridge and SPI capable software for a PC. The attacker manages to use the SPI interface to dump content of the Flash memory.

Step 3: Analysis of Flash memory

The attacker analyses flash memory to find strings that look like cryptographic assets. After several attempts, the attacker can relate a cryptographic key for the TOE secure storage.

Step 4: Exploitation

The attacker uses a public vulnerability on the NSPE to remotely connect to the TOE and extract the encrypted secure storage area used by the TOE. The attacker can now decrypt the TOE secure storage with the cryptographic key extracted from Step 3.

Example rating

Factor	Comment	Identification	Exploitation
Elapsed time	Analyzing the TOE PCB, performing probing and analyzing Flash memory can take up to one month. Exploitation only requires remote access to the TOE.	≤ one month (3)	≤ one hour (0)
Expertise	In order perform probing and related analyses, an expert is needed. The exploitation needs less expertise and can be performed by Proficient attacker.	Expert (5)	Proficient (2)
Knowledge of the TOE	Identification or exploitation only assume public knowledge of the TOE.	Public (0)	Public (0)
Access to TOE	Only few samples of the TOE are needed.	< 10 (0)	< 10 (0)
Equipment	The attacker needs standard equipment for SPI probing (microprobes, SPI adapter). For UART probing, standard equipment (solder station, UART adapter) is sufficient.	Standard (1)	Standard (2)
Open samples	No open sample is required	Public (0)	NA
Sub-total		9	4
Total		13	

5.2.3 Glitch to break isolation.

Isolation could potentially also be broken using a glitch attack, assuming that the appropriate decision point can be determined. For example, a NSPE program could make a request that should be denied, and then the chip glitched to cause the code to be granted access. The calculation will look similar to the glitch against initialization.

5.3 SECURE_STORAGE

5.3.1 Dumping Flash

Attack path

In this example, the attacker is dumping the flash memory that holds the executable image for the TOE to get access to TOE assets. This memory is found not to be visible from NSPE, contrarily to the encrypted secure storage area used by the TOE.

Step 1: Analysis of the PCB

The attacker first needs to open the device and get access to the PCB. He analyses the surface of the PCB and identifies Flash chip. With the datasheet of this chip found on Internet, the attacker determines this chip supports Serial Peripheral Interface (SPI) and finds the corresponding pins.

Step 2: Probing of the Flash

The attacker uses microprobes to connect to the Flash memory SPI, a USB-SPI bridge and SPI capable software for a PC. The attacker manages to use the SPI interface to dump content of the Flash memory.

Step 3: Analysis of Flash memory

The attacker analyses flash memory to find strings that look like cryptographic assets. After several attempts, the attacker can relate a cryptographic key for the TOE secure storage.

Step 4: Exploitation

The attacker uses a public vulnerability on the NSPE to remotely connect to the TOE and extract the encrypted secure storage area used by the TOE. The attacker can now decrypt the TOE secure storage with the cryptographic key extracted from Step 3.

Example rating

Factor	Comment	Identification	Exploitation
Elapsed time	Analyzing the TOE PCB, performing probing and analyzing Flash memory can take up to one month. Exploitation only requires remote access to the TOE.	≤ one month (3)	≤ one hour (0)
Expertise	In order perform probing and related analyses, an expert is needed. The exploitation needs less expertise and can be performed by Proficient attacker.	Expert (5)	Proficient (2)
Knowledge of the TOE	Identification or exploitation only assume public knowledge of the TOE.	Public (0)	Public (0)
Access to TOE	Only few samples of the TOE are needed.	< 10 (0)	< 10 (0)
Equipment	The attacker needs standard equipment for SPI probing (microprobes, SPI adapter). For UART probing, standard equipment (solder station, UART adapter) is sufficient.	Standard (1)	Standard (2)
Open samples	No open sample is required	Public (0)	NA
Sub-total		9	4
Total		13	

5.3.2 Rowhammer

Attack path

In this example, the attacker analyses the TOE and identifies that the DRAM which is used may be vulnerable to a Rowhammer attack (see for instance Drammer: Deterministic Rowhammer Attacks on Mobile Platforms by Victor van der Veen and all in Proceedings of CCS'16). The attacker exploits this vulnerability to retrieve the value of a cryptographic key.

In real life examples, Rowhammer normally only forms one part of an attack. However, for the purpose of this example, consider a case where it can be used for the entire attack.

Assume that the file system stores access control information alongside a file in the form of attribute bits – like those used in Unix. Therefore, if the attacker discovers where this metadata is stored, they only need to change the access from *owner only* to *all users* and then can read data that was meant to be restricted. Once the attacker has the value, they can export it.

The attacker develops a rogue application in the NSPE for this attack. By applying physical memory messaging, the application arranges the physical memory in such a way to cause the cryptographic key to land in a vulnerable physical memory page. Then the rogue application induces bit- by repeatedly accessing the same adjacent row of the DRAM bank.

Obviously, this attack only applies to devices that use Dynamic RAM. Many Systems on Chip systems will use Static RAM, which is not vulnerable to this specific attack – but the examiner should check the literature to see if similar techniques have been discovered.

Example rating

Factor	**Comment**	**Identification**	**Exploitation**
Elapsed time	Most of the time required for identification consists in developing and testing the rogue application and performing the fault analysis to retrieve the key. Exploitation consists of repeating the attack. As the attack is not entirely deterministic, it is considered that it can be done in one week or less.	≤ one month (3)	≤ one week (4)
Expertise	An expert expertise is required to develop the attack. For the exploitation, a proficient attacker is required to run the attack.	Expert (5)	Proficient (2)
Knowledge of the TOE	Identification or exploitation only assume public knowledge of the TOE.	Public (0)	Public (0)
Access to TOE	Only few samples of the TOE are needed.	< 10 (0)	< 10 (0)
Equipment	The attack requires standard equipment for both identification and exploitation.	Standard (1)	Standard (2)
Open samples	No open sample is required	Public (0)	NA
Sub-total		9	8
Total		17	

5.3.3 Decapsulating and probing internal flash

Attacks that require the attacker to physically grind away encapsulation and pacification layers can expose interfaces that are not normally accessible, leading to the exposure of secrets.

Example rating

Factor	**Comment**	**Identification**	**Exploitation**
Elapsed time	The attack requires use of a CNC milling machine to remove the passivation to reveal, but not damage, the silicon, such that a probe can be connected to internal bus. Exploitation consists of repeating the attack, having gathered the knowledge of where and how deep to mill.	> one month (5)	≤ one week (4)
Expertise	An expert expertise is required to develop the attack. For the exploitation, a proficient attacker is required to run the attack.	Expert (5)	Expert (4)
Knowledge of the TOE	Identification or exploitation only assume public knowledge of the TOE.	Public (0)	Public (0)
Access to TOE	Multiple samples of the TOE are needed during identification as some will be destroyed.	> 10 (1)	< 10 (0)
Equipment	The attack requires bespoke equipment for both identification and exploitation.	Bespoke (5)	Bespoke (6)
Open samples	No open sample is required	Public (0)	NA
Sub-total		16	14
Total		30	

The total points are 30 so out of scope for Level 3.

Even if this attack path were to reveal a class secret, it is out of scope for Level 2, due to the cost of identification.

5.4 FIRMWARE_UPDATE

5.4.1 Network Replay

Attack path

In this example, the attacker uses a network probe to capture traffic between the TOE and external entities. He can record corresponding network traffic, even if it is encrypted, and to replay it on another TOE. The attacker must be able to communicate with the TOE.

The exchange between the TOE and the remote entity can for instance consists of:

- a personalization stage, then the attacker will be able to clone the TOE
- of a remote update, then the attacker will be able to replay this update on another TOE
- of activation of a TOE features, then the attacker will be able to also activate this feature on another TOE.

It is assumed that the TOE has no countermeasures against replay attacks, such as counters or sessions.

Example rating

Factor	**Comment**	**Identification**	**Exploitation**
Elapsed time	Analyzing the traffic between the TOE and remote entities can take up to one week before identifying an interesting exchange. Exploitation is fast, as in consist of replaying the same traffic.	≤ one week (2)	≤ one hour (0)
Expertise	To identify an interesting exchange, a proficient attacker is needed. If well documented during the identification phase, the exploitation only needs layman expertise.	Proficient (2)	Layman (0)
Knowledge of the TOE	No specific knowledge of the TOE is required for identification or exploitation	Public (0)	Public (0)
Access to TOE	Only few samples of the TOE are needed.	< 10 (0)	< 10 (0)
Equipment	The attacker needs network equipment to communicate with the TOE and for identification a network probe.	Standard (1)	Standard (2)
Sub-total		5	2
Total		7	

5.4.2 Firmware Update Flaw

Attack path

In this example, the attacker first needs to obtain a firmware update file, for instance by downloading it from the update website of the manufacturer, and to capture through a network probe a valid remote update sequence.

The attacker analyses the structure of a firmware update and tries to find flaws on how the integrity of the update if preserved. For instance, he can generate collisions for the hash algorithm used in signature verification. Or some simple flaw in checking the signature, for example using strcmp() instead of memcmp(), or preferably a constant time binary comparison

The attacker then develops a rogue application and adds this application on the firmware update file.

Before being able to push this update to the TOE, the attacker needs to spoof the identity of the update web server (assume that connection is not protected by TLS or similar) and replay a valid remote update sequence but with the rogue firmware update file. Even if TLS is used, it might be poorly configured, permitting a Man-in-the-Middle attack.

Example rating

Factor	**Comment**	**Identification**	**Exploitation**
Elapsed time	Analyzing the firmware update file to find a flaw and developing a rogue application can take up to one month. Exploitation is fast, as in consist of pushing a rogue firmware update.	≤ one week (2)	≤ one hour (0)
Expertise	To find a flaw in the protection of firmware update, an expert is needed. If well documented during the identification phase, the exploitation only needs layman expertise.	Expert (5)	Layman (0)
Knowledge of the TOE	No specific knowledge of the TOE is required for identification or exploitation	Public (0)	Public (0)
Access to TOE	Only few samples of the TOE are needed.	< 10 (0)	< 10 (0)
Equipment	The attacker needs network equipment to communicate with the TOE and for identification a network probe.	Standard (1)	Standard (2)
Sub-total		8	2
Total		10	

5.4.3 Bypass firmware rollback protection

Description of the attack

Perturbation attacks change the normal behavior of the TOE to create an exploitable error during operation. The behavior is typically changed by:

- operating the TOE outside its intended operating environment (usually characterized in terms of temperature, voltage and the externally supplied clock frequency).
- or injecting an optical glitch (for example a. laser pulse) to alter the intended behavior of the TOE.
- Or injecting an electromagnetic pulse to alter the intended behavior of the TOE.

Effect of attack

Perturbation attacks can:

- produce faults in memory, which can be exploitable for instance in cryptanalysis,
- alter the semantics of a program, for instance by performing different instructions,
- change the expected control flow, for instance during access control or lifecycle state checks.

The success of perturbation attacks may be repeatable with some probability.

This attack method potentially violates any security function from [PSA-PP].

Impact on the TOE

The impacts of the voltage/clock glitches or temperature stress on the TOE are:

- a modification of data: one or several bits in memory are changed temporarily or permanently during read or write operation.
- or a change in the program flow of the TOE, with instructions being skipped, replaced by another instruction, or have an altered effect, such as an inverted test in a conditional jump instruction.

Characteristics of the attack

For the identification phase of a perturbation attack, the attacker needs to:

- Analyze the code of the TOE to identify parts potentially vulnerable to a perturbation and that can give access to unauthorized operations or data, such as branch instruction in a sensitive operation or cryptographic operation.
- Identify the type of fault to apply.
- Potentially develop a support application on the NSPE to synchronize the perturbation with the execution of the targeted code.
- Set-up the equipment and glitch bench, laser, or electromagnetic pulse.
- Determine fault parameters (for example, the duration and intensity of the glitch, the time at which to apply the glitch), usually using a trial-and-error method.

Once the attack has been identified, it may require adjustments to be repeatable on another TOE.

Example: Bypass firmware rollback protection

Attack path

In this example, the attacker uses a voltage glitch to bypass firmware rollback protection and install an old firmware with known vulnerabilities on the TOE. The attacker may first disassemble TOE code obtained from a firmware update file to identify part of code responsible for rollback protection and determine whether it can be vulnerable to a glitch.

The attacker develops code on the NSPE that triggers firmware update and that will help him synchronizing the attack. A first approximation on the time to apply the voltage glitch is obtained by comparing time to perform update on an old firmware and a new firmware. Power or electromagnetic traces can be used to have a more accurate time range to trigger the glitch.

The attacker sets-up the glitch bench by analyzing the datasheet of the chip and the operating voltage range. With trial and error on glitch parameters of the voltage generator, the attacker successfully bypasses firmware rollback protection.

Using the same equipment and similar parameters, the attacker can repeat it on other TOEs.

Example rating

Factor	Comment	Identification	Exploitation
Elapsed time	One month should be sufficient to identify vulnerable code and set-up the attack, one week to adapt and repeat it.	≤ one month (3)	≤ one week (4)
Expertise	An expert attacker is required to identify vulnerable code and set-up the attack. Once the attack has been identified, a proficient attacker should be able to adapt and repeat it.	Expert (5)	Proficient (2)
Knowledge of the TOE	Identification or exploitation only assume public knowledge of the TOE.	Public (0)	Public (0)
Access to TOE	Only few samples of the TOE are needed. Some may be destroyed when applying the perturbation.	< 10 (0)	< 10 (0)
Equipment	A voltage generator is required for this attack. The same equipment is required for identification and exploitation.	Specialized (3)	Specialized (4)
Open samples	No open sample is required	Public (0)	NA
Sub-total		11	10
Total		21	

5.5 SECURE_STATE

5.5.1 No state protection - reprovision secrets.

In most designs, it is possible to read and write configuration data and keys while the device is in the pre-production state. However, once the device moves into the production state, these secrets must be protected in integrity and confidentiality.

If there is an attack that permits the attacker to reverse the state transition and return the device to a configuration state, they would be able to read or alter these secrets. By changing the public key, they could make the device accept firmware updates signed by a key they control. By obtaining root storage keys, they could gain access to the stored data.

If there is no protection – or a simple flaw can be found this is likely to be around 7 points.

5.5.2 Glitch to change state and reprovision keys.

Just as it is possible to use a glitch to subvert the secure boot, or the isolation, it might be possible to glitch the process that sets the state. If the attacker can discover the point in the code where the decision to permit or deny write access to secrets is made, and then glitch the processor, it could leave access to secrets open. Then it would be possible to read or write these values.

In some architectures this might not be possible. For example, if secrets are written to e-fuse, then an attack path that relies on rewriting the value might not be applicable.

However, if they are in scope, then the attack potentials are likely to be similar to glitching the boot sequence.

5.6 CRYPTO

5.6.1 Cache Timing on AES

Description of the attack

This class of attacks assumes that the TOE uses cache memory for cryptographic operations. It consists in deducing keys or parts of the keys managed by the TOE by exploiting the cache memory shared by the TOE during cryptographic operations and a rogue application in the NSPE. The rogue application has partial control on the cache memory, not its content, and is able produce statistics on the execution time of the TOE and to relate it to the use of the cache, to cache misses and finally to values of the cryptographic keys used by the TOE.

These attacks assume that the TOE performs the cryptography using software primitives only, without cryptographic accelerators, and that it relies in frequent memory accesses in cache memory, such as for lookup tables. The rogue application must also be able to access the same cache memory.

Effect of attack

By performing many measures of the execution time of a cryptographic algorithm while manipulating the cache memory, the attacker can in some conditions retrieve the cryptographic key used in the algorithm.

This attack method violates the CRYPTO security function from [PSA-PP].

Impact on the TOE

The cryptographic key managed by the TOE during cryptographic operations becomes compromised.

Characteristics of the attack

To perform this class of attacks, the attacker has first to be able to load a rogue application on the device. If this application is on the SPE, the rating of this step is combined with the rest of the attack path.

The application can trigger execution by the TOE of the targeted cryptographic operation and that can control the memory cache by performing for instance cache line eviction or cache flushing on the lines used by the TOE for cryptographic operations. By repeating the same operation multiple times while manipulating the cache, by measuring the execution time for each execution of the cryptographic

operation and by analyzing the differences, the attacker can infer the value of the key used in the cryptographic operation.

These attacks rely on frequent cache accesses by the cryptographic algorithm, such as an AES with T-table. The attacker can detect by measuring the execution time whether memory cache was accessed or not and relate this access to the value of key.

Example: Retrieve AES key using Cache attack

Attack path

An attacker can retrieve an AES key of a software cryptographic implementation used by the TOE.

This attack requires to control the cache memory from a rogue NSPE application while the TOE executes the AES algorithm, to gather information leaked by the time differences when data is loaded from the cache.

We assume that the attack is made using standard equipment available to an academic or basic attacker. There is more specialized equipment that is available to test labs. Using such specialized equipment would increase the rating but might also reduce the time – therefore this analysis has not costed it separately.

Given the attacker has up to a month to identify the attack, they should be able to gather sufficient traces even on the slowest of microcontrollers – unless there is some architectural restriction on using the key.

Example rating

Factor	Comment	Identification	Exploitation
Elapsed time	Identification can be done in less than one month. Exploitation assumes that the attack is public and ready to be executed. Having extracted the secret, the identifier prepares a script that can be run by a layman.	≤ one month (3)	≤ one day (3)
Expertise	The identification requires an expert on hardware and an expert on software. Exploitation requires a layman.	Expert (5)	Layman (0)
Knowledge of the TOE	Identification or exploitation only assume public knowledge of the TOE.	Public (0)	Public (0)
Access to TOE	Only few samples of the TOE are needed.	< 10 (0)	< 10 (0)
Equipment	Standard equipment is required.	Standard (1)	Standard (2)
Open samples	No open sample is required	Public (0)	NA
Sub-total		9	5
Total		14	

Or course this attack can be mitigated by using separate memory or specialized hardware for the AES engine, so it does not touch the main cache.

5.6.2 Bad RNG

This attack consists in predicting the output of the Random Number Generator or in reducing the possible ranges of values for the output of the RNG.

Effect of attack

The attack may allow the attacker to either:

- compromise the past values of the output of the RNG, based on the analysis of next output values.
- or predict the next values of the output of the RNG, based on the analysis of past output values.
- or force the output of the RNG to specific values or patterns.

All these attacks have the effect to reduce the entropy to the RNG.

Impact on the TOE

RNG attacks weaken the strength of cryptography primitives or protocols based on RNG, for instance for generation of a cryptographic key or generation of a nonce using in a cryptographic protocol. The attacker may be able to directly retrieve the value which is based on the output of the RNG or to reduce the spectrum values of the output to the point a brute force attack is possible. With the knowledge of this value, the attacker may compromise the integrity or confidentiality of assets, as well as the authenticity of the TOE by impersonating the TOE.

Characteristics of the attack

The characteristics of the attack will depend on the type of RNG implemented by the device: True RNG (TRNG), Pseudo RNG (PRNG) or Hybrid RNG (HRNG).

TRNGs are most likely to be vulnerable to physical attacks, such as perturbation or probing attacks (see Section **Error! Reference source not found.**) to force or modify output of the TOE.

Attacks on PRNGs can make use of statistical analysis on past values of the RNG to predict probable future values, target the seed used by the RNG algorithm, use side-channel analysis, or also flood RNG of requests to repeat previous values.

Attacks on HRNGs will usually combine attacks on TRNGs and PRNGs.

RNGs compliant with FIPS 140-2, ISO/IEC 19790:2012 or NIST SP 800-90A/B should be immune to RNG attacks considered in this document.

Example: Low seed entropy of PRNG

Attack path

In this example, the attacker is targeting the seed used by a PRNG. It is assumed that this seed is initialized at start-up with low entropy and not mixed with another signal or mixed with hard-coded key in source code.

The attacker first performs numerous samplings of the output of the RNG, as close as possible of TOE start-up. Then he performs statistical analysis on the obtained data. Due to the low entropy, output of the RNG is not equally distributed. The attacker can then build look-up tables to retrieve past and future values of the RNG.

Example rating

Factor	Comment	Identification	Exploitation
Elapsed time	Performing sampling, on multiple TOEs and statistical analysis, and can take up to one month. Exploitation using look-up tables can take up to one day depending on the frequency in which patterns can be found.	≤ one month (3)	≤ one day (3)
Expertise	To find a flaw in the RNG seed, an expert is required. To correctly use the look-up tables and exploit the result, the exploitation needs Proficient attacker.	Expert (5)	Proficient (2)
Knowledge of the TOE	No specific knowledge of the TOE is required for identification or exploitation	Public (0)	Public (0)
Access to TOE	Only few samples of the TOE are needed.	< 10 (0)	< 10 (0)
Equipment	The attacker needs standard equipment to query the TOE.	Standard (1)	Standard (2)
Sub-total		9	7
Total		16	

5.6.3 Timing attack on a Diffie-Hellman

Attack path

There are many references on how to perform timing attacks on naïve implementations of Diffie-Hellman. Consider for instance *Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems* by Paul C. Kocher in Proceedings of CRYPTO'96.

Example rating

Factor	Comment	Identification	Exploitation
Elapsed time	Designing or customizing timing attack tool dedicated for the TOE and performing measurement to test the tools and can take up to one month. Exploitation using dedicated tools for timing attack can take up to one day depending on the noise on measurements.	≤ one week (2)	≤ one day (3)
Expertise	To set-up tools to perform timing attacks, an expert is needed. To correctly use these tools, provided they are sufficiently documented, the exploitation needs Proficient attacker.	Expert (5)	Proficient (2)
Knowledge of the TOE	No specific knowledge of the TOE is required for identification or exploitation	Public (0)	Public (0)
Access to TOE	Only few samples of the TOE are needed.	< 10 (0)	< 10 (0)
Equipment	The attacker needs standard equipment to query the TOE.	Standard (1)	Standard (2)
Sub-total		8	7
Total		15	

5.6.4 Side-channel attack on AES using Chipwhisperer

Attack path

In this example, the attacker analyses the electromagnetic leakage of TOE to retrieve the AES key used for secure storage.

To set-up the attack, the attackers need an electromagnetic antenna, a device to capture the signal and software to analyses the signal. The open-source Chipwhisperer hardware and software can be used for that purpose.

The attacker also needs to install an application on the device that makes use of secure storage and that will trigger generation by the TOE of the signal to analyze. An application on the NSPE is sufficient.

Chipwhisperer open hardware solutions can be used to capture and store the EM traces of the AES operations. Once the trace acquisition phase is finished, Chipwhisperer open-source software can be used to align and correlate the EM traces to retrieve the AES key.

Example rating

Factor	Comment	Identification	Exploitation
Elapsed time	Setting up the bench for Chipwhisperer can take up to one month for identification and for exploitation, considering the time needed to acquire enough traces and to analyze them.	≤ one month (3)	≤ one week (4)
Expertise	To correctly set up Chipwhisperer, an expert is needed. After a correct configuration of the tool, the exploitation needs a Proficient attacker.	Expert (5)	Proficient (2)
Knowledge of the TOE	Identification or exploitation only assumes public knowledge of the TOE.	Public (0)	Public (0)
Access to TOE	Only few samples of the TOE are needed.	< 10 (0)	< 10 (0)
Equipment	Chipwhisperer is open-source and considered as standard equipment	Standard (1)	Standard (2)
Open samples	No open sample is required	Public (0)	NA
Sub-total		9	8
Total		17	

Even if equipment like Chipwhisperer improves and therefore the level of expertise required in the identification phase is reduced bringing the total attack potential to below 16.

However, any attack path that needs use of equipment – even standard equipment like Chipwhisperer, in the exploitation phase is out of scope for Level 2 – which only includes exploits that can be performed remotely. This is therefore only in scope for Level 3.

5.6.5 Bad crypto API

Attack Path

Most cryptographic primitives rely on parameters being chosen uniformly. In the recent literature, there have been CVEs issue for attacks where the attacker has been able to specify parameters which should be selected randomly, for example attacks where the same caller can specify the IV used with an AES-GCM key, thereby creating two messages using the same key and counter. Or in the case of ECDSA, signing different messages with the same nonce. In both cases, simple mathematical manipulations lead to exposing the key material.

This is particularly problematic if the cryptographic functions can be called from the Non-Secure Processing Environment. Obviously, if the device uses the PSA CryptoAPI this attack should not be feasible.

Example rating

Factor	Comment	Identification	Exploitation
Elapsed time	Once the expert has spotted the flaw, it will take a few days to write an attack script. In the exploitation phase, the script only needs to run a couple of cryptographic operations and do a few calculations to expose the key.	≤ one week (2)	≤ one day (3)
Expertise	We assume it takes an expert to spot the flaw. They can develop a script a layman can use remotely.	Expert (5)	Layman (0)
Knowledge of the TOE	Identification or exploitation only assumes public knowledge of the TOE.	Public (0)	Public (0)
Access to TOE	Only few samples of the TOE are needed.	< 10 (0)	< 10 (0)
Equipment	Aside from the normal equipment required to communicate with the device, no extra equipment is required.	Standard (1)	Standard (2)
Open samples	No open sample is required	Public (0)	NA
Sub-total		6	5
Total		11	

5.7 ATTESTATION

The attestation function is used to prove that a specific part is genuine and running the correct firmware. Attacks on this function are used to pass a fake part as genuine and thereby attack the larger system.

5.7.1 Badly defined format allowing substitution

If a system uses an in-house attestation report format, there is a risk that there will be a flaw which permits messages to be altered after they have been signed.

If a system is using a standard attestation format, such as EAT or DICE it is unlikely that messages will be malleable in this way.

5.7.2 Poor isolation permitting signing of arbitrary messages

A correct attestation system will only sign measurements made by the Root of Trust or clearly indicate the origin of the message that it signs.

If there is a flaw in the isolation, it might be possible for an attacker to use the attestation key to sign an arbitrary message – thereby replacing the genuine measurements with some chosen values. The mechanisms to break isolation are discussed in section 5.2 SOFTWARE_ISOLATION.

5.8 AUDIT

Currently Audit is optional. Therefore, attacks on audit will be considered in a later version of this document.

5.9 DEBUG

5.9.1 Debug left open

If the debug port is accidentally left open, an attacker can make use of this to extract data. The attack is likely to be easy to discover and the attack potential would be in single figures – assuming that the debug data can be accessed from the NSPE.

It is a requirement of PSA Certified Level 1 that all debug ports are properly protected, either by being completely disabled or by requiring strong authorization to move into a debug state. Therefore, even though PSA Certified Level 2 normally only considers remote attacks, any device with an open debug port must be considered a failure.

5.9.2 Flaw in debug authentication

In this example, assume that the developer has made an error in designing the authorization method, such that it is possible to forge an authorization token, but that this takes a significant amount of work – perhaps the device uses a counter mode AEAD with a short tag and after manipulating the authorization string of a genuine message to extract the information needed, they need to submit 2^24 requests to find the correct tag. While the device can be used as an oracle, it may only accept requests at a certain rate. As the key is different on each device, this must be performed on every target device.

Example rating

Factor	Comment	Identification	Exploitation
Elapsed time	We assuming that it would take time to identify the flaw and to prove that it is exploitable. The script brute forces the tag, so needs to work though all possible versions.	> one month (5)	≤ one month (6)
Expertise	It would take a cryptographic expert to devise the attack and convert it into a script. However, once packaged any competent computer professional could use it.	Expert (5)	Proficient (2)
Knowledge of the TOE	Identification or exploitation only assume public knowledge of the TOE.	Public (0)	Public (0)
Access to TOE	Only few samples of the TOE are needed.	< 10 (0)	< 10 (0)
Equipment	Although	Standard (1)	Standard (2)
Open samples	No open sample is required	Public (0)	NA
Sub-total		11	10
Total		21	

5.9.3 Glitch to bypass debug

In any system with a debug system that can be switched off, there is some point where the decision to enable or disable the system is made. Either, at a fixed point in the boot sequence, based on the current lifecycle state, or when some authorization is presented. As with all other similar decision points this can be attacked using glitching techniques. The rating for a glitching attack will be the same as for a glitch against initialization.

5.10 Trusted Subsystem interface

In systems where some services are provided by a physically separated Trusted Subsystem there is an additional attack surface offered by the communication channel between the two parts of the SPE. By physically separated, we mean either a separate package, or a separate die within a package. Where the Trusted Subsystem resides on the main die – and is not identifiable as a distinct system, the normal rules for Levels 2 or 3 apply.

This is more important if the Trusted Subsystem is a separate physical component, as the channel will be exposed. A recent blog post by Dolos Group showed how it is possible to extract the disk encryption key from a TPM using a logic analyzer on the unencrypted communication interface.

https://dolosgroup.io/blog/2021/7/9/from-stolen-laptop-to-inside-the-company-network

However, even if the channel is in package or even on-die, if there are any other devices connected to the same bus, the data is could still be exposed if an attack can subvert one of the other components.

5.10.1 Software based attacks

Software only – the NSPE listens for exchanges between the SPE and Trusted Subsystem. This would require that the NSPE has access to the bus used by the Trusted Subsystem. This may permit either a timing attack on the cryptography, or simply reading the key-material, as in the Dolos Group attack.

Example rating

Factor	**Comment**	**Identification**	**Exploitation**
Elapsed time	An Expert creates an attack script that runs in the NSPE, and which monitors the communication channel	≤ one month (3)	≤ one hour (0)
Expertise	Writing the attack script requires an Expert. Implementing it However, it may be possible that a layman can operate the script once developed.	Expert (5)	Layman (0)
Knowledge of the TOE	The attack script will require knowledge of the TOE, including the logical layout of the communication bus, which should be at least restricted.	Restricted (2)	Public (0)
Access to TOE	The attack is nondestructive, as it is pure firmware. Therefore, only a single sample is needed to develop the attack.	< 10 (0)	< 10 (0)
Equipment	The attack uses standard equipment to communicate with the TOE and to develop an attack script.	Standard (1)	Standard (2)
Open samples	No open sample is required	Public (0)	NA
Sub-total		11	2
Total		13	

5.10.2 Physical Probing of the trusted sub-system interface

Physical probing is only in scope for Level 2 if the attacker can find a class secret that can be used in the exploitation phase. In general, if the attack requires decapsulation of the device, this would also be out of scope – see 5.3.3.

Therefore, we limit the discussion to probing of a channel exposed on external pins. This includes interfaces for in-package or on-die components where the bus terminates in exposed pins – even if those pins are not identified in the public documents. It is assumed that an attack could attach an oscilloscope to pins marks as not connected and determine if they are carrying any signals.

The attacker identifying the attack must extract the secret and then develop an attack script that can be run remotely. This has a similar attack potential to the other similar attacks.

Example rating

Elapsed time	Analyzing the TOE PCB, performing probing and analyzing traffic can take up to one month. Exploitation only requires remote access to the TOE.	≤ one month (3)	≤ one hour (0)
Expertise	In order perform probing and related analyses, an expert is needed. The exploitation needs less expertise and can be performed by Proficient attacker.	Expert (5)	Proficient (2)
Knowledge of the TOE	Identification or exploitation only assume public knowledge of the TOE.	Public (0)	Public (0)
Access to TOE	Only few samples of the TOE are needed.	< 10 (0)	< 10 (0)
Equipment	The attacker needs standard equipment for SPI probing (microprobes, SPI adapter). For UART probing, standard equipment (solder station, UART adapter) is sufficient.	Standard (1)	Standard (2)
Open samples	No open sample is required	Public (0)	NA
Sub-total		9	4
Total		13	

Physical probing is in scope for Level 3, even if there are individual keys. The attacker identifying the attack will discover how to find the key and will publish an attack method. Then in the exploitation phase, a proficient attack will attack test probes as described and run an attack script to discover the local key, which they can then use in an attack script: potentially by dumping and decrypting flash.

5.10.3 Physical probing required for exploitation.

For Level 2+SE attacks that require physical probing of an external bus between the main processor and a Trusted sub-system for exploitation are in scope. However, if the bus is fully internal, physical probing is not considered, as the attack potential is going to be above 16.

For Level 3, physical probing is always in scope. However, an attack that requires decapsulation in the exploitation phase is not going to be practical within the attack potential.

Example rating

Factor	**Comment**	**Identification**	**Exploitation**
Elapsed time	Analyzing the TOE PCB, performing probing and analyzing Flash memory can take up to one month. Probing an individual device may take more than a day, but less than a week.	≤ one month (3)	≤ one day (3)
Expertise	In order perform probing and related analyses, an expert is needed. The exploitation needs less expertise and can be performed by Proficient attacker.	Expert (5)	Proficient (2)
Knowledge of the TOE	Identification or exploitation only assume public knowledge of the TOE.	Public (0)	Public (0)
Access to TOE	Only few samples of the TOE are needed.	< 10 (0)	< 10 (0)
Equipment	The attacker needs standard equipment for SPI probing (microprobes, SPI adapter). For UART probing, standard equipment (solder station, UART adapter) is sufficient.	Standard (1)	Standard (2)
Open samples	No open sample is required	Public (0)	NA
Sub-total		9	7
Total		16	

5.10.4 Man-in-the-Middle attack between the Trusted Subsystem and the rest of the SPE

The communication between the Trusted Subsystem and the rest of the SPE shall maintain its integrity and its confidentiality. This can be done both physically to prevent probing and logically using cryptographic means. An attacker might be able to add the third hardware between the SPE and the Trusted Subsystem to compromise the integrity of the communication. This could lead to assets being exposed or lowering the security posture of the entire system. One example is

https://www.cl.cam.ac.uk/research/security/banking/nopin/oakland10chipbroken.pdf where MitM attack allows for bypassing the PIN verification of EMV card.

This attack is likely to be outside the attacker potential for L2 but can be considered for L3. The estimate rating of this attack is as follows.

Example rating

Factor	Comment	Identification	Exploitation
Elapsed time	It takes more than a week and less than one month to reverse engineer the commination line as well as the logical protocol used. More than one day is needed to apply the attack since hardware modification needs to be performed.	≤ one month (3)	≤ one week (4)
Expertise	An expert is needed to identify the attack and create the exploit because it requires deeper understanding on how to reverse engineer communication protocols. Once the hardware exploit has been created, a proficient person can apply the attack.	Expert (5)	Proficient (2)
Knowledge of the TOE	No sensitive information is required	Public (0)	Public (0)
Access to TOE	Less than 10 samples are sufficient to identify and perform the attack	< 10 (0)	< 10 (0)
Equipment	Standard equipment such as soldering iron and basic logic analyzer are sufficient to perform this attack	Standard (1)	Standard (2)
Open samples	No open sample is required	Public (0)	NA
Sub-total		9	8
Total		17	

Note, this makes certain assumptions about the physical format, in some cases the attack scores might be lower, and the evaluator will need to make this assessment.

This attack relies on a plain text link. A well-designed encrypted link should make this attack Not practical and therefore out of scope

5.10.5 Replacing the Trusted Subsystem

An attacker could probe the communication between the SPE and the Trusted Subsystem. Reverse engineering the communication the internal operation of the Trusted Subsystem could allow for an attacker to replace the Trusted Subsystem with a malicious subsystem prepared by an attacker. This could lead to assets being exposed or lowering the security posture of the entire system. One example is **https://tches.iacr.org/index.php/TCHES/article/view/9063/8650** where replacing a secure element could allow for an attacker to steal a vehicle.

Assuming that the that the Trusted Subsystem is a soldered component, this attack is likely to be outside the attacker potential for L2 but can be considered for L3.

If it is socketed, for example a SIM card, then the attack is much quick and would be in scope for Level 2.

The estimate rating of this attack is as follows.

Note,

As with the man in the middle attack, a well-designed binding would make the attack Not Practical, even if the Trusted Subsystem is replaceable.

Example rating

Factor	Comment	Identification	Exploitation
Elapsed time	It takes more than a week and less than one month to reverse engineer the communication line as well as the logical protocol used. For a soldered component more than one day would be required. If the trusted component is socketed, then replacing the hardware module can be done quickly and no points would be scored for exploitation.	≤ one month (3)	≤ one week (4)
Expertise	An expert is needed to identify the attack and create the exploit because it requires deeper understanding on how to reverse engineer communication protocols. Once the hardware exploit has been created, a proficient person can apply the attack.	Expert (5)	Proficient (2)
Knowledge of the TOE	No sensitive information is required	Public (0)	Public (0)
Access to TOE	Less than 10 samples are sufficient to identify and perform the attack	< 10 (0)	< 10 (0)
Equipment	Standard equipment such as soldering iron and basic logic analyzer are sufficient to perform this attack	Standard (1)	Standard (2)
Open samples	No open sample is required	Public (0)	NA
Sub-total		9	8
Total		17	