psacertified™

PSA Certified Level 2
Attack Method
Version 1.1

psacertified™
level two

Document number:	JSADEN004
Version:	1.1
Release Number:	01
Author	PSA JSA Members: Arm Limited Brightsight B.V. CAICT Prove & Run S.A.S. Riscure B.V. Trust CB B.V. UL TS B.V.
Authorized by:	PSA JSA Members

Date of Issue: 18/02/2020

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 Level 2 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 PSA Certified Level 2 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 PSA Certified Level 2 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, Certification, IoT, Platform Security Architecture, Questionnaire, PSA, Security

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
18/02/2020	1.1	Non-confidential	Clarifications for PSA Level 2 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 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	This is a Root of Trust Service within the Application Root of Trust domain
Evaluation laboratory	Laboratory or facility that performs the technical review of questionnaires submitted for PSA Certified Level 1. 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)	This is the security domain outside of the SPE, the Application domain, typically containing the application firmware and hardware
PSA	Platform Security Architecture
PSA Certification Body	Entity that receives applications for PSA security certification, issues certificates, updates security certification scheme
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	Functional certification for a device that ensures that the device has implemented PSA Functional APIs and passed the PSA Functional certification Test Suites
PSA Immutable Root of Trust	The hardware and code and data that cannot be modified following manufacturing. See *PSA Security Model* [PSA-SM] for details
PSA Root of Trust	This defines the most trusted security domain within a PSA system. See *PSA Security Model* [PSA-SM] for details
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]
Root of Trust Service (RoT Service)	A set of related security operations that are implemented in a Secure Partition. The server endpoint of a PSA IPC channel. Multiple RoT Services can co-exist in a single Secure Partition
Secure Partition	A thread of execution with protected runtime state within the Secure Processing Environment. Container for the implementation of one or more RoT Services. Multiple Secure Partitions are allowed in a platform
Secure Partition Manager (SPM)	The part of a PSA implementation that is responsible for isolating software in partitions, managing the execution of software within partitions, and providing IPC between partitions
Secure Processing Environment (SPE)	A platform's processing environment for software that provides confidentiality and integrity for its runtime state from software and hardware outside of the SPE
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 a Real-Time Operating System (RTOS) or some other run-time executive, middleware, standard stacks, chip specific device drivers, etc., but not the application specific code
Trusted subsystem	Any trusted component outside of the functional scope of the PSA Root of Trust but within the trust boundary of the PSA Root of Trust. For example, DDR protection system, trusted peripherals, SIM or TPM

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 Level 2 Attack Method).
- The number (JSADEN-004) and version.
- The page numbers to which your comments apply.
- The rule identifiers to which your comments apply, if applicable.
- 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 behaviour of any document when viewed with any other PDF reader.

2 Introduction

2.1 Document Context

This document is the result of the cooperation of PSA JSA group. It provides guidance as to which attack methods have to be considered in PSA Root of Trust evaluation according to PSA Certified Level 2 and PSA Certified Level 2 Ready (see Section 2.3).

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 Level 2 or PSA Certified Level 2 Ready (see below) 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, in order 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 certificate is generated for a PSA Certified 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 attack that shall be considered during the evaluation. For each evaluation it has to 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 in particular for PSA Certified Level 2 Ready evaluation, 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.

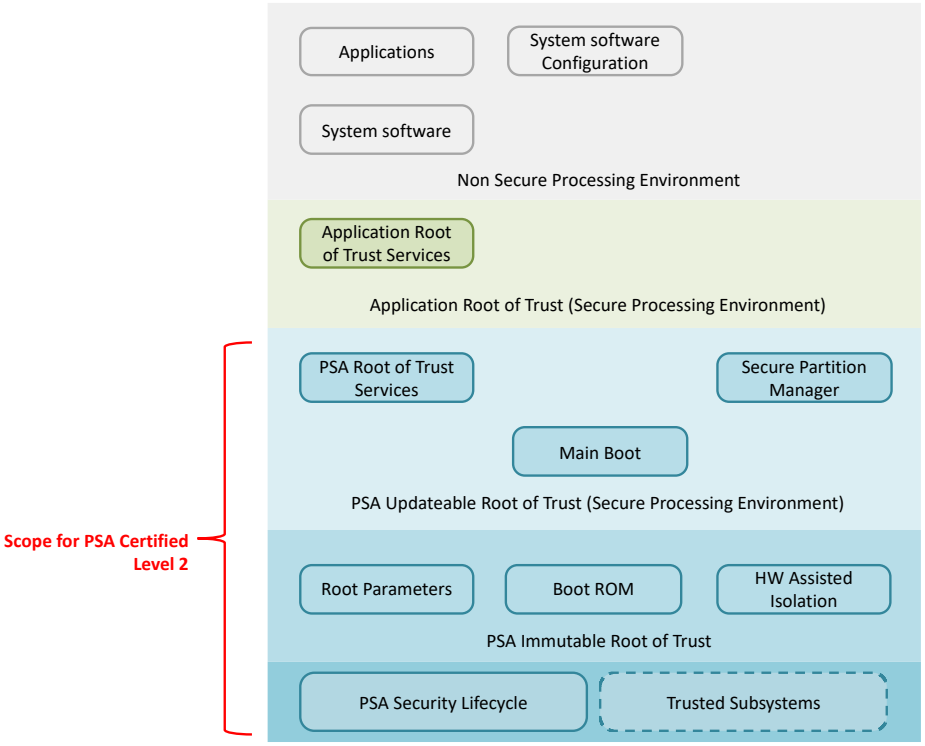

Figure 1: Scope of PSA Certified Level 2

3.2 Interfaces

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

- API between Application RoT and PSA-RoT within the SPE
- API between NPSE 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, e.g. once it has become public).

Attack path identification as well as exploitation analysis and tests are mapped to relevant factors, based from [JIL-APSC]: elapsed time, expertise, knowledge of the TOE, access to the TOE, equipment needed to carry out an attack. Even if the attack consists of several steps, identification and exploitation need only be computed for the entire attack path.

To complete an attack potential calculation the points for identification and exploitation have to be added as both phases together constitute the complete attack. When presenting the attack potential calculation in the ETR, 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

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

Not practical is used as the attack path is not exploitable within a timescale that would be useful to an attacker.

4.1.2 Expertise

For this factor, three types of experts are defined:

- Laymen are unknowledgeable compared to experts or proficient persons, with no particular expertise.
- Proficient persons are knowledgeable in that they are familiar with the security behaviour 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 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 (e.g., from the Internet) or if it is provided by the vendor to any customer.
- Restricted information concerning the TOE (e.g., as gained from vendor technical specifications): Information is considered restricted if it is distributed on request and the distribution is registered. Suitable example might be the functional specification.
- Sensitive information about the TOE (e.g., knowledge of internal design, which may have to be obtained by "social engineering" or exhaustive reverse engineering). Suitable example might be High-Level Design, Low-Level Design information or the Source Code.

4.1.4 Access to TOE

This factor refers to the number of devices with the TOE necessary during the identification or exploitation phase.

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 a number of devices need to be tried out,
- The attack succeeds only after having destroyed a number of devices (on average),
- The attacker needs to collect information from several copies of the TOE.

4.1.5 Equipment

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

In order to clarify equipment category, price and availability has to be considered.

- Standard equipment is equipment that is readily available to the attacker, either for the identification of vulnerability or for an attack. This equipment can be readily obtained—e.g., 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 isn't readily available to the attacker, but could be acquired without undue effort. This could include purchase of moderate amounts of equipment (e.g., dedicated electronic cards, specialized test bench, protocol analysers, specialized JTAG probes, etc.) or development of more extensive attack scripts or programs.
- Bespoke equipment is not readily available to the public as it might need to be specially produced (e.g., 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.

4.2 Attack quotation table

Table 1 provides the number of points for each factors 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 sum of points for identification plus the sum of points of exploitation.

For PSA Certified Level 2, TOE must be resistant to attackers with attack potential of at **most 16**.

5 Attacks Methods

The same template is used for each class of attacks retained in this section.

Description of Attack

Gives a short description of the purpose and method of the attack.

Effect of Attack

Contains a more detailed attack description and how to recognise when this attack has succeeded. This may include variations of a basic attack.

Impact on TOE

Examples of how the attack may result in an exploitable vulnerability in the TOE.

This will also explain the motivation for carrying out this attack.

Characteristics of the Attack

- Factors that make the attack difficult or easy to carry out or to apply to a real TOE.
- Skills and tools required to carry out the attack.
- References to books, papers or standard methods, where appropriate. This list of references will probably not be complete – more techniques are used in labs than are published – but they may give an understanding of the basics of the attack or attack techniques.

Examples of Attack Potential Ratings

These examples illustrate in more details what is behind the different attack methods and aim at providing the most average rating expected for the illustrated attack(s). The presentation of these ratings also helps to come to consistent and commonly agreed ways to interpret the attack potential table.

5.1 Remote attacks

For these classes of attacks, the attacker is assumed to only have remote access to the TOE for exploitation of the attack. The identification phase of the attack may involve local attacks.

5.1.1 Data injection

5.1.1.1 Description of the attack

The exploitation phase for this attack consists in remotely sending malicious commands to the TOE, for instance commands related to remote firmware upgrade.

5.1.1.2 Effect of attack

Data injection is a way of exercising access to TOE interfaces outside of expected conditions. These interfaces are those exposed remotely. The purpose of the attack is to find a directly exploitable flaw in the TOE or to gain knowledge on the internal design of the TOE.

5.1.1.3 Impact on the TOE

Using this attack, the attacker may compromise assets managed by the TOE, for instance by performing unauthorized operation on TOE assets. The attacker may have access to PSA Root of Trust security features outside of expected conditions.

5.1.1.4 Characteristics of the attack

To perform this attack, the attacker must have the capability to remotely communicate with the device hosting the TOE. Then the attacker can proceed to various technique to overcome TOE protection:

- Protocol attacks, by analyzing exchanges with the TOE and send ill-formed commands that will be misinterpreted by the TOE (and lead for instance to a buffer overflow or provide access to unauthorized functions). Identifying protocol vulnerabilities can be performed by remote fuzzing interfaces exposed by the TOE.
- Man-in-the-middle attacks, by spoofing the identity of a remote entity trusted by the TOE and then performing commands on its behalf.
- Replay attacks, by resending commands already accepted by the TOE, or another TOE, in a different context.

5.1.1.5 Example: Network replay attack

5.1.1.5.1 Attack path

In this example, the attacker uses a network probe to capture traffic between the TOE and external entities. He is able to 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.

5.1.1.5.2 Example rating

Factor	Comment	Identification	Exploitation
Elapsed time	Analysing 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	In order to identify an interesting exchange, a proficient attacker is needed. If well documented during the identification	Proficient (2)	Layman (0)

	phase, the exploitation only needs layman expertise.		
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 also for identification a network probe.	Standard (1)	Standard (2)
Sub-total		5	2
Total		7	

5.1.2 Rogue code execution

5.1.2.1 Description of the attack

This attack consists in executing rogue code on Secure Processing Environment as Application Root of Trust. This rogue code targets TOE interfaces.

5.1.2.2 Effect of attack

Rogue code is a way of exercising access to TOE logical interfaces outside of expected conditions. Such interfaces can be direct interfaces such as APIs exposed by the TOE or indirect interfaces such as peripherals also used by the TOE. The purpose of the attack is to find a directly exploitable flaw in the TOE or to gain knowledge on the internal design of the TOE.

5.1.2.3 Impact on the TOE

Using this attack, the attacker may compromise assets managed by the TOE, for instance by bypassing access control performed by the TOE or executing code with the same privileges as the TOE, thus invalidating the TOE software isolation property. The attacker may perform dump of memory containing assets, unauthorized modification of asset or control of the internal information flow of the PSA Root of Trust.

5.1.2.4 Characteristics of the attack

To perform this attack, the attacker must have the capability to execute rogue code on the device. It can be obtained remotely by exploiting a flaw in the firmware update feature to include rogue code, for instance by forging firmware update signature.

For identification of the attack, the attacker may have to load a first set of applications to identify vulnerabilities and once it is confirmed that vulnerabilities can be exploited then load another application that reuses there results and provides the expected effect of the attack.

Vulnerabilities can be found for instance by performing buffer overflow or stack overflow, or by fuzzing of the available APIs, including potential undocumented commands, and then by checking potential inconsistent responses from the TOE. With these techniques, the rogue code may try to perform privilege escalation by modifying the current execution context to obtain kernel privileges.

5.1.2.5 Example: Exploiting a flaw in the firmware update feature

5.1.2.5.1 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 also 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 is able to generate collisions for the hash algorithm used in signature verification.

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 (we assume that connection is not protected by TLS or similar) and replay a valid remote update sequence but with the rogue firmware update file.

5.1.2.5.2 Example rating

Factor	Comment	Identification	Exploitation
Elapsed time	Analysing 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	In order 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 also for identification a network probe.	Standard (1)	Standard (2)

Sub-total		8	2
Total		10	

5.2 Cryptographic attacks

5.2.1 RNG

5.2.1.1 Description of the attack

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.

5.2.1.2 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.

5.2.1.3 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.

5.2.1.4 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 5.3) 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 possible 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.

5.2.1.5 Example: Low seed entropy of PRNG

5.2.1.5.1 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 equi-distributed. The attacker is able to build look-up tables to retrieve past and future values of the RNG.

5.2.1.5.2 Example rating

Factor	Comment	Identification	Exploitation
Elapsed time	Performing sampling, possibly 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	In order to find a flaw in the RNG seed, an expert is needed. In order 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	

5.2.2 Brute force

5.2.2.1 Description of the attack

This attack consists in enumerating all possible values of a cryptographic asset (key, nonce, cipher text or clear text) until guessing the correct value of a cryptographic key.

5.2.2.2 Effect of attack

This attack compromises a cryptographic key normally protected by the TOE.

5.2.2.3 Impact on the TOE

The assets protected by the brute-force compromised cryptographic key become in turn compromised.

5.2.2.4 Characteristics of the attack

There is no specific strategy for brute-forcing a cryptographic key. For a key size of *n* bit, should take a maximum a 2_n tries before guessing the correct value, considering that the RNG used to generate the key is equi-distributed.

It is usually considered that checking all possible values of a 56 bit key, such as DES key is achievable with today's computing power of an average PC.

The attacker may exploit weakness of the RNG (see Section 5.2.1) or side-channel attacks (see Section 5.2.3) to narrow the possible values of the key before using brute-force.

5.2.2.5 Example: Brute-force 128 bit key

5.2.2.5.1 Attack path

In this example, the attacker targets a 128 bit key. Even with very specialized equipment or cloud computing for testing key values, the key space is too big to be explored in less than a matter of years. This attack is considered as Not practical.

5.2.3 Side-channel

5.2.3.1 Description of the attack

This class of attacks consisting in exploiting unintentional measurable physical characteristics during the execution of an algorithm, in order to compromise a secret value used by this algorithm. Such unintentional information channels can be timing characteristics of the algorithm, power consumption of the IC, or electromagnetic radiation of the IC or buses.

Also, side-channel attacks are not applicable for PSA Certified Level 2 Ready pre-certification, as they depends on the physical characteristics of the FPGA system. These characteristics will differ for the future ASIC or custom chip based on the FPGA system design and targeting PSA Certified Level 2 certification.

5.2.3.2 Effect of attack

By measuring the execution time of a cryptographic algorithm on several input, the attacker can in some conditions retrieve the cryptographic key used in the algorithm.

5.2.3.3 Impact on the TOE

The assets protected by the timing-attack compromised cryptographic key become in turn compromised.

5.2.3.4 Characteristics of the attack

Execution time of trivial implementations of some cryptographic algorithm depends linearly on the number of '1' in the binary representation of the cryptographic key. This applies for instance to

implementation relying on exponentiation by squaring to perform modular exponentiation. By precisely measuring time to execute the algorithm on different input, the attacker can perform statistical correlation analysis and retrieve value of the key.

The attack can also be performed remotely, but it requires more measures to remove network noise.

5.2.3.5 Example: Timing attack on a Diffie-Hellman

5.2.3.5.1 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. Koche in Proceedings of CRYPTO'96.

5.2.3.5.2 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 month (3)	≤ one day (3)
Expertise	In order to set-up tools to perform timing attacks, an expert is needed. In order 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		9	7
Total		16	

5.3 Physical attacks

For these classes of attacks, the attacker has physical access to the TOE.

It is assumed for PSA Certified Level 2 that physical attacks can only be applied during the identification stage of the attack, in order for instance to extract an information that can later be used during the exploitation stage. The objective here is to prevent scalable attacks, where an identified attack can be replicated remotely on a wide range of devices, and for which impacts on IoT devices can be severe.

5.3.1 Probing

5.3.1.1 Description of the attack

Probing attacks consists in having direct access to the internal signals the TOE and circumventing existing security measures to protect TOE assets. This attack is performed using connectors, probes or microprobes on the available pins, buses or unprotected or ill-protected debug (JTAG, Single Wire Debug) or test interfaces of the Printed Circuit Board (PCB) of the TOE. Probing may require reverse engineering to understand which is the information carried by the probed signal.

5.3.1.2 Effect of attack

Probing attacks allow direct access to memory (such as flash, DRAM, CPU registers) and data exchanged on buses. The attacker can also force the value of internal signals.

5.3.1.3 Impact on the TOE

The impacts of the probing attacks on the TOE are:

- Disclosure or compromise of TOE assets, such as cryptographic keys or executable code.
- Deactivation of TOE security features.
- Change the expected control flow of programs.

5.3.1.4 Example: Dumping Flash memory

5.3.1.4.1 Attack path

In this example, the attacker is dumping the flash memory that holds the executable image for the TOE in order to get access to TOE assets. This memory is assumed 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 analyze 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. This key is not diversified among TOEs.

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.

5.3.1.4.2 Example rating

Factor	Comment	Identification	Exploitation
Elapsed time	Analysing the TOE PCB, performing probing and analysing 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)	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 specialized equipment for SPI probing (microprobes, SPI adapter). For UART probing, standard equipment (solder station, UART adapter) is sufficient.	Specialized (3)	Standard (2)
Sub-total		11	2
Total		13	

5.3.2 Perturbation

5.3.2.1 Description of the attack

Perturbation attacks change the normal behaviour of the TOE in order to create an exploitable error during operation. The behaviour is typically changed by operating the TOE outside its intended

operating environment (usually characterised in terms of temperature, voltage and the externally supplied clock frequency).

Also, perturbation attacks are not applicable for PSA Certified Level 2 Ready pre-certification, as they depend on the physical characteristics of the FPGA system. These characteristics will differ for the future ASIC or custom chip based on the FPGA system design and targeting PSA Certified Level 2 certification.

5.3.2.2 Effect of attack

Perturbation attacks can produce faults in memory, that can be exploitable for instance in cryptanalysis, alter the semantics of a program, for instance by performing different instructions, or 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.

5.3.2.3 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 instructions or have an altered effect, such as an inverted test in a conditional jump instruction.
- or also altering the output of the RNG, with predictable value.

5.3.2.4 Example

5.3.2.4.1 Attack path

5.3.2.4.2 Example rating

Factor	Comment	Identification	Exploitation
Elapsed time		≤ one week (2)	≤ one day (3)
Expertise		Proficient (2)	Proficient (2)
Knowledge of the TOE		Public (0)	Public (0)
Access to TOE		< 10 (0)	< 10 (0)
Equipment		Specialized (4)	Standard (2)
Sub-total		8	7
Total		15	

Non-Confidential Proprietary Notice

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