# Smart Lock SESIP Profile

psacertified™

| | |
|---|---|
| Document number: | JSADEN020 |
| Version: | 0.5 |
| Release Number: | 01 |
| Author: | PSA JSA Members:<br>Applus+, Laboratories<br>Arm Limited<br>CAICT<br>DEKRA Testing and Certification<br>ECSEC Laboratory Inc<br>ProvenRun S.A.S.<br>Riscure B.V.<br>Serma Safety & Security S.A.S.<br>SGS Brightsight B.V.<br>TrustCB B.V.<br>UL TS B.V. |
| Authorized by: | PSA JSA Members |
| Date of Issue: | 24/10/2023 |

## Abstract

PSA Certified encourages device makers to develop a threat model to understand the security problem for their specific product.  A SESIP Profile is a good way of doing this as the document can optionally be used by the developer to create a Security Target (ST) that may be used to get the device evaluated in the test laboratory.

This SESIP Profile takes the example of a Smart Lock, starting with a platform functional overview, an analysis of security objectives and requirements which leads on to a set of Security Functional Requirements (SFRs) that could be tested by an evaluation lab.  Many of the device level security requirements build on the chip's Root of Trust (PSA-RoT).  The reader is encouraged to look at Appendix A (Security Problem) and Appendix B (Mapping with PSA Certified PSA-RoT) for a fuller understanding of how a PSA Certified chip can help the device maker meet their security requirements.

## Keywords

Smart Lock, Platform Security Architecture, SESIP, Protection Profile, PSA Certified, Matter

# Contents

# 1 About this document

## 1.1 Current Status and Anticipated Changes

Current Status: Release 01

## 1.2 Release Information

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

| Date | Version | Confidentiality | Change |
|---|---|---|---|
| 29/09/2022 | 0.1 | Non-confidential | First draft version |
| 19/12/2022 | 0.2 | Non-confidential | Second draft version |
| 11/01/2023 | 0.3 | Non-confidential | Added Matter mapping and other fixes |
| 16/01/2023 | 0.4 | Non-confidential | Fixed PSA Certified Level 3 mapping |
| 24/10/2023 | 0.5 | Non-confidential | Added abstract and edits for publication |
| | | | |
| | | | |
| | | | |
| | | | |

## 1.3 References

This document refers to the following informative documents.

| Ref | Doc No | Author(s) | Title |
|---|---|---|---|
| [SESIP] | GP_FST_070 | GlobalPlatform | Security Evaluation Standard for IoT Platforms (SESIP), Version 1.1, June 2021 |
| [PSAL3PP] | JSADEN011 | PSA JSA | SESIP Profile for PSA Certified Level 2 |
| [Matter] | | Connectivity Standard Alliance | Matter Specification, Version 1.0 |

## 1.4 Terms and Abbreviations

This document uses the following terms and abbreviations

| Term | Meaning |
|---|---|
| **API** | Application Programming Interface |
| **ARoT** | Application specific Root of Trust |

| | |
|---|---|
| **CPU** | Central Processing Unit |
| **HTTPS** | HyperText Transfer Protocol Secure |
| **IPSec** | Internet Protocol Security |
| **NTP** | Network Time Protocol |
| **NSPE** | Non-Secure Processing Environment |
| **OEM** | Original Equipment Manufacturer |
| **OS** | Operating System |
| **OTP** | One-Time-Programmable |
| **RAM** | Random Access Memory |
| **REE** | Rich Execution Environment |
| **ROM** | Read Only Memory |
| **RoT** | Root of Trust |
| **SFR** | Security Functional Requirement |
| **SESIP** | Security Evaluation Standard for IoT Platforms |
| **SPE** | Secure Processing Environment |
| **SoC** | System-on-Chip |
| **ST** | Security Target |
| **TEE** | Trusted Execution Environment |
| **TLS** | Transport Layer Security |

## 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 (Smart Lock SESIP Profile).
- The number (JSADEN-020) 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 behavior of any document when viewed with any other PDF reader.

# 2 Introduction

This SESIP Profile targets smart locks for doors, to provide smart, connected access to home, office or more generally building or room.

The considered platform is composed of a hardware device and firmware implementing the smart lock functionalities. The firmware itself may include a generic purpose operating system.

Profile Reference

See title page.

## 2.1 Platform Reference

| | |
|---|---|
| Platform name | *<Platform name>* |
| Platform version | *<Platform version>* |
| Platform identification | *<Platform id details>* |
| Platform Type | Hardware device and a firmware implementing the smart lock functionalities |

## 2.2 Platform Functional Overview and Description

### 2.2.1 Usage and Major Security Features

Smart lock devices are used to control locking and unlocking of doors through an electromechanical system (e.g. a solenoid or motor) after user authentication.

User authentication is usually performed through a keypad, biometrics, or virtual keys, stored for instance on a wallet of the user smart phone or on a proximity card.

The smart locks considered in this Profile require a local or remote connection to configure the device and retrieve access logs.

Smart locks include at least the following security features:

- Secure operational life-cycle of the smart lock:
  - Secure start up (see "Secure Initialization of Platform").
  - Device commissioning. The device uniquely identifies itself (see "Verification of Platform Identity" and "Verification of Platform Instance Identity") and shows it is genuine (see "Attestation of Platform Genuineness") to the administrator.
  - Software updates. The software running on the lock can be updated to fix vulnerabilities identified after the device's deployment. See "Secure Update of Platform" and Flaw Reporting Procedure (ALC_FLR.2).
  - Protection of private data through security measures for data at rest (see "Secure Storage") and data in transit (see "Secure Communication Support") and erasure of private date after end-of-life (see "Factory Reset of Platform").

    - Secure access to debug features, if any (See “Secure Debugging”).
    - Protection against physical attacks (See “Physical Attacker Resistance”).
- User and admin authentication. Authentication before access to the smart lock, and before modification of its configuration or performing any maintenance operations is required. Local and network authentication may rely on different methods and credentials. See “Authenticated Access Control”.
- Secure communication. More generally, any network communication is performed using a protocol that includes integrity and confidentiality protection. See “Secure Communication Support”.
- Log of security events. Security events are logged locally on the smart lock, to support forensic analysis of an attack or other suspicious event. See “Audit Log Generation and Storage”.

### 2.2.2 Platform Architecture

<A short introduction and description of the Platform, the combination of hardware and software to be evaluated, must be provided. Typically, this would be taken from the datasheet.>

The Platform is the combination of hardware and software that provide a runtime environment and related applications for user authentication and smart lock administration. It is to be embedded in a hardware device that provides the power source, includes the lock mechanism itself, the network interfaces or other hardware used by the smart lock, but which are not part of the scope of evaluation.

Figure 1 illustrates the main components for a smart lock Platform in this Profile <Replace this generic figure according to the specific Platform architecture and scope>. It distinguishes between a Secure Processing Environment (SPE), in charge of the platform root of trust functions, such as secure boot, secure update, secure storage, and the Non-Secure Processing Environment, in charge of supporting connectivity (either local, mesh or global), service discovery (using Matter for instance), peripherals (such as keypad) and all locate or remote users interactions.

The Secure Processing Environment can also support applications, illustrated as Applications Root of Trust (ARoT) in Figure 1. For instance, the smart lock can use the SPE to host an ARoT for biometric authentication, using biometric identifiers protected by the SPE.

<Add all the necessary details for the software scope: libraries, drivers, versions, …>

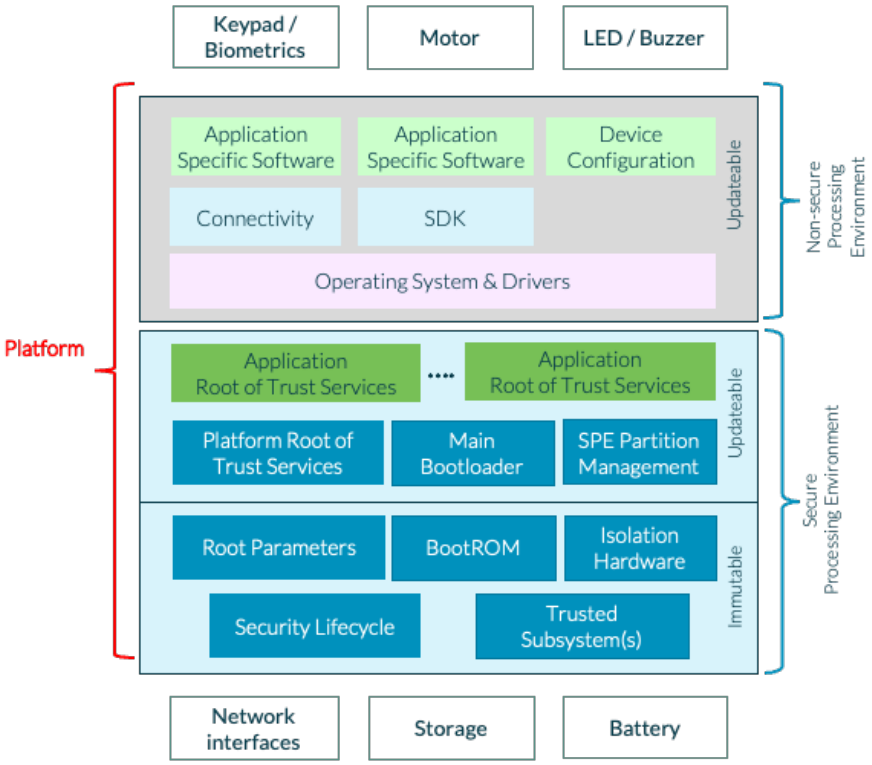


*Figure 1: Smart lock Platform*

The Physical scope for the Platform is typically composed of a microcontroller which supports, secure boot and isolation between SPE and NSPE. The MCU may also support OTPs to store sensitive data, such as smart lock ID or secrets. *<write specific scope details, which may be a silicon chip, a PCB, … >*

The out-of-scope part comprises *<to be completed by developer>*.

# 3 Security Objectives for the Operational Environment

For the platform to fulfil its security requirements, the operational environment (technical or procedural) must fulfil the following objectives.

### 3.1.1 Credential Management

The cryptographic keys, credentials and certificates used in the Platform shall be securely generated and provisioned to the Platform.

Additionally, they should be securely managed during the life cycle of Platform when used outside of the Platform (such as in gateways, back-end servers or maintenance devices).

Trusted Administrator

The Admin of the Platform must not be careless, wilfully negligent or hostile.

### 3.1.2 Environment

The environment of the platform shall include all hardware components required for platform operation, such as network interface or storage or remote servers.

*<ST writer: describe the platform environment including remote services such as secure update server, NTP server.>*

### 3.1.3 Others

<ST writer: list all other mandatory objectives for the environment with reference to where in the guidance documents this objective is described.>

# 4 Security Requirements and Implementation

## 4.1 Security Assurance Requirements

The claimed assurance requirements package is SESIP3, as described in Section 5.1.

### 4.1.1 Flaw Reporting Procedure (ALC_FLR.2)

In accordance with the requirement for a flaw reporting procedure (ALC_FLR.2), including a process to generate any needed update and distribute it, the developer has defined the following procedure:

*<ST writer: Describe the procedure, including where flaws can be reported (website and/or email address), how the reported flaws are handled in a timely manner, and how an application developer/end-user can get informed of the update. The process to receive flaw reports and handle them in a timely manner needs to be described.>*

## 4.2 Security Functional Requirements

Platforms conformant to this Profile must satisfy the following security functional requirements.

### 4.2.1 Verification of Platform Identity

The platform provides a unique identification of the platform, including all its parts and their versions.

*<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this.>*

### 4.2.2 Verification of Platform Instance Identity

The platform provides a unique identification of that specific instantiation of the platform, including all its parts and their versions.

*<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this.>*

### 4.2.3 Attestation of Platform Genuineness

The platform provides an attestation of the “Verification of Platform Identity” and “Verification of Platform Instance Identity”, in a way that ensures that the platform cannot be cloned or changed without detection.

*<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this.>*

### 4.2.4 Secure Storage

The platform ensures that all data stored by the application, except for *<list of data stored in plaintext>*, is protected to ensure its authenticity and integrity as specified in *<specification>* with a platform instance unique key of key length *<key length>.*

**Note** 1**:** This requirement is used to protect at least Smart lock ID, Configuration and Credentials, Smart lock Logs. Therefore, those must not be listed in the “list of data stored in plaintext”.

### 4.2.5 Secure Initialization of Platform

The platform ensures its authenticity and integrity during platform initialization. If the platform authenticity or integrity cannot be ensured, the platform will go to *<list of controlled states>*.

**Note** 2: Secure initialization must cover all software parts of the evaluation.

*<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this. This should include all stages of the boot chain and describe for each stage how the verification of the loaded software is performed, and the cryptographic material used for that purpose.>*

### 4.2.6 Secure Update of Platform

The platform can be updated to a newer version in the field such that the integrity, authenticity, and confidentiality of the platform is maintained.

*<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this. This should include the verifications performed by the secure update mechanism, the order of these verifications, the behaviour of the platform is case of a failed verification and the cryptographic material used for that purpose.>*

### 4.2.7 Secure Communication Support

The platform provides the application with one or more secure communication channel(s).

The secure communication channel authenticates *<list of endpoints>* and protects against *<list of attacks including disclosure, modification, replay, erasure>* of messages between the endpoints, using *<list of protocols and measures>*.

*<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this. This should include the types of supported secure communication channels, the method they can be invoked, the used cryptographic material, the contexts they shall always be used.>*

**Note** 3: Secure communication channels may include any of IPsec, TLS or HTTPS performed by the platform. Validity of the peer certificate shall at least be determined by the certificate path, the expiration date, and the revocation status.

**Note** 4: If TLS is used then TLS 1.2 or greater must be used. The device shall implement certificate validation for all such TLS connections and validate that connections to the device are signed using the correct certificate. Initial setup shall not include the transmission of credentials over a non-TLS session.

### 4.2.8 Secure Communication Enforcement

The platform ensures that the application can only communicate with *<list of endpoints>* over the secure communication channel(s) supported by the platform using *<list of protocols and measures>*.

*<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this. This should include the types of supported secure communication channels, the method they can be invoked, the used cryptographic material, the contexts they shall always be used.>*

### 4.2.9 Audit Log Generation and Storage

The platform generates and maintains an audit log of <failed and successful authentication attempts, firmware upgrade requests and progress, integrity errors, list of other significant security events> and allows access and analysis of these logs following a specific <access control policy>.

<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this.>

**Note** 5: Audit log record should mention the nature of the event, date and time of the event and the user, if any, responsible for the event. The Platform should rely on a secure NTP server to provide reliable source for time stamps for the audit trail.

**Note** 6: Significant security events include at least failed and successful authentication attempts, firmware upgrade requests and progress, integrity errors.

### 4.2.10 Software Attacker Resistance: Isolation of Platform (between SPE and NSPE)

The platform provides isolation between the application and itself, such that an attacker able to run code as an application on the platform cannot compromise the other functional requirements.

*<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this. This should include how the isolation hardware is used to enforce isolation between SPE and NSPE.>*

### 4.2.11 Cryptographic Operation

The platform provides the application with *<list of cryptographic operations>* functionality with *<list of algorithms>* as specified in *<specification>* for key lengths *<list of key lengths>* and modes *<list of modes>*.

*<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this.>*

### 4.2.12 Cryptographic Random Number Generation

The platform provides the application with a way based on *<list of entropy sources>* to generate random numbers to as specified in *<specification>*.

*<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this.>*

### 4.2.13 Cryptographic Key Generation

The platform provides the application with a way to generate cryptographic keys for use in *<list of cryptographic algorithms>* as specified in *<specification>* for key lengths *<list of key lengths>*.

*<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this. This should include the storage locate of these assets and the cryptographic means and materials used to protect these assets.>*

### 4.2.14 Cryptographic KeyStore

The platform provides the application with a way to store *<list of assets, such as cryptographic keys and passwords>* such that not even the application can compromise the *<authenticity, integrity, confidentiality>* of this data. This data can be used for the cryptographic operations *<list of operations>*.

*<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this. This should include the storage locate of these assets and the cryptographic means and materials used to protect these assets.>*

**Note** 7: Cryptographic keystore is used for cryptographic assets and operations related to attestation, secure update, secure storage, secure communication support and roles authentication (see Authenticated Access Control requirement).

### 4.2.15 Factory Reset of Platform

The platform can be reset to the state in which it exists when the composite product embedding the platform is delivered to the user, before any personal user data, user credentials, or user configuration is present on the platform.

*<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this.>*

### 4.2.16 Authenticated Access Control

The platform allows only *<list of role(s)>*, identified, authenticated and authorized as specified by *<specification>* to allow performing of *<control lock motor, device commissioning and administration operations>*.

*<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this.>*

**Note** 8: This SFR is not yet of part of the SFRs catalogue [SESIP] but will be integrated in a future version.

### 4.2.17 Physical Attacker Resistance

The platform detects or prevents attacks by an attacker with physical access before the attacker compromises any of the other functional requirements.

*<ST writer: add a short conformance rationale describing how this is done and which types of physical attacks the platform is able to detect or prevent.>*

**Note** 9: For a smart lock, physical attacker resistance typically encompasses physical access to the device internals, such as the control of lock motor, or cryptographic keys or bus.

### 4.2.18 Secure Debugging

The platform only provides *<list of endpoints>* authenticated as specified in *<specification>* with debug functionality.

The platform ensures that all data stored by the application, with the exception of *<list of exceptions>*, is made unavailable.

*<ST writer: add a short conformance rationale describing how this is done with a reference to the specific guidance/functional specification section describing this.>*

**Note** 10: This security functional requirements shall be included if secure debugging is supported.

# 5 Mapping and Sufficiency Rationales

## 5.1 SESIP3 Sufficiency

SESIP3 deliverables add basic documentation required to perform a white-box evaluation, as well as basic evidence of the use of configuration management.

| Assurance Class | Assurance Family | Covered by | Rationale |
|---|---|---|---|
| ASE: Security Target evaluation | ASE_INT.1 ST Introduction | <Section "Introduction" and title page of the Security Target> | <TBD> |
| | ASE_OBJ.1 Security requirements for the operational environment | <Section "Security Objectives for the Operational Environment" of the Security Target> | <TBD> |
| | ASE_REQ.3 Listed Security requirements | <Section "Security Requirements and Implementation" of the Security Target> | <TBD> |
| | ASE_TSS.1 TOE Summary Specification | <Section "Security Requirements and Implementation" of the Security Target> | <TBD> |
| ADV: Development | ADV_FSP.4 Complete functional specification | <Description of which developer evidence is used to meet this requirement> | <TBD> |
| | ADV_IMP.3 Complete mapping of the implementation representation of the TSF to the SFRs | <Description of which developer evidence is used to meet this requirement> | <TBD> |
| AGD: Guidance documents | AGD_OPE.1 Operational user guidance | <Description of which developer evidence is used to meet this requirement> | <TBD> |
| | AGD_PRE.1 Preparative procedures | <Description of which developer evidence is used to meet this requirement> | <TBD> |
| ALC: Life-cycle support | ALC_CMC.1 Labelling of the TOE | <Description of which developer evidence is used to meet this requirement> | <TBD> |
| | ALC_CMS.1 TOE CM Coverage | <Description of which developer evidence is used to meet this requirement> | <TBD> |
| | ALC_FLR.2 Flaw reporting procedures | <ALC_FLR section in the Security Target and description of which | <TBD> |

| | | | |
|---|---|---|---|
| | | developer evidence is used to meet this requirement> | |
| ATE: Tests | ATE_IND.1 Independent testing: conformance | <Description of which developer evidence is used to meet this requirement> | <TBD> |
| AVA: Vulnerability Assessment | AVA_VAN.3 Focused Vulnerability analysis | N.A. A vulnerability analysis is performed by the platform evaluator to ascertain the presence of potential vulnerabilities. | <TBD> |

# Appendix A Security Problem Definition

This informational appendix provides the risks analysis elements that justify the choice of the security requirements in Section 4.

## A.1 Users and External Entities

The external entities that are considered in this Profile are the User and the Device Owner, who also plays the role of the Admin.

The Device Owner can pair the smart lock with mobile APP for setting and configuration, after authentication.

The Admin can modify the smart lock configuration, perform firmware update and access audit logs after authentication.

## A.2 Assets

### *A.2.1 Platform Data*

#### *A.2.1.1 Smart lock ID*

A unique ID to identify the platform on a network, such as a MAC address or a unique device ID managed by OEM.

Properties: Integrity

#### *A.2.1.2 Firmware*

The smart lock's firmware.

Properties: Integrity, Authenticity, Confidentiality

#### *A.2.1.3 Firmware Certificate*

The cryptographic certificate used to authenticate firmware and firmware updates.

Properties: Integrity, Authenticity

#### *A.2.1.4 Logs*

The event logs, that can be used to detect suspicious activities.

Properties: Integrity

### *A.2.2 User Data*

#### *A.2.2.1 Configuration*

The smart lock's configuration, split into two components:

- Smart lock's dynamic configuration, including network configuration such as its Thread configuration, or the name of a WLAN network, or IP and DNS addresses

- Smart lock settings such as users access control policy (list of authorised users and time ranges). Depending on the implementation, the configurations are locally and/or remotely stored.

Properties: Integrity

#### *A.2.2.2 Credentials*

The authentication credentials, used for local and remote authentication, such as:

- Network credentials, to authenticate if needed on the network, for instance a Wi-Fi pre-shared key or an 802.1x certificate.
- Device authentication credentials to authenticate on remote servers.
- Server authentication data, such as public key certificates, to be protected in integrity only.
- Session keys, used after establishment of a trusted communication channel with servers.
- Administration and user credentials, to authenticate to the services provided by the smart lock, either for administration or for regular use.
- User biometric patterns to be used in face recognition or similar algorithms.

Properties: Integrity, Confidentiality

#### *A.2.2.3 Application Root of Trust Data*

Data used by Applications Root of Trust if such applications are present in the Secure Processing Environment.

This data is isolated from the Non-Secure Processing Environment.

Properties: Integrity, Confidentiality

## A.3 Threats

An attacker is a threat agent (a person or a process acting on his/her behalf) trying to undermine the Platform security policy defined by the current Profile. The attacker especially tries to change properties of the assets defined in Section A.2.

### *A.3.1 Impersonation*

An attacker impersonates a legitimate user on the smart lock, either a regular user that can use the device (to lock or unlock a door) or an admin user.

The user credentials may be obtained through default admin passwords, interception, for instance in insecure communication links, or exposed through data disclosure.

The attacker may then use the device, modify configuration or try to modify firmware.

Assets threatened directly: Credentials
Assets threatened indirectly: Firmware, Configuration, Logs.

### *A.3.2 MITM*

An attacker performs a Man-In-The-Middle attack or impersonates a server the smart lock connects to, for instance to download configuration or to upload the event logs.

The attacker may rely on insecure communication links or prior modification of the server credentials on the smart lock through insecure configuration.

The attacker may then access and modify Logs, Credentials, Configuration data.

Assets threatened directly: Credentials (Server), Logs, Configuration

### *A.3.3 Firmware Abuse*

An attacker exploits a flawed version of the firmware and obtains partial or total control of the smart lock. The firmware may have been modified prior to the attack to include a malware or consist of an outdated version of the original firmware.

The attacker may for instance use data injection or modify on the device the value of the firmware certificate used to authenticate the installed firmware or firmware updates.

Such an attack can allow for elevation of privileges, where a regular user gains access to admin privileges.

Assets threatened directly: Firmware, Firmware Certificate.

Assets threatened indirectly: All.

# Appendix B Mapping with PSA Certified

This appendix provides a mapping between the Security Requirements of PSA Certified Level 3 SESIP Profiles for PSA-RoT [PSAL3PP] and this Profile.

The smart lock Security Requirements for which the following table provides a “Same” mapping for PSA Certified Level 3 SESIP Profile SFR are already part of the certified PSA-RoT platform. The Security Requirements with an “Optional” mapping are part of the certified PSA-RoT platform only if they have been included in the Security Target for the considered PSA-RoT.

| Smart lock Profile SFR | PSA Certified Level 3 SESIP Profile SFR |
|---|---|
| Verification of Platform Identity | Same |
| Verification of Platform Instance Identity | Same |
| Attestation of Platform Genuineness | Same |
| Secure Storage | Same: Either Secure Encrypted Storage (internal storage) or Secure Storage (internal storage) or Secure External Storage |
| Secure Initialization of Platform | Same |
| Secure Update of Platform | Same |
| Secure Communication Support | Same |
| Audit Log Generation and Storage | Optional |
| Software Attacker Resistance: Isolation of Platform (Between SPE and NSPE) | Same |
| Cryptographic Operation | Same |
| Cryptographic Random Number Generation | Same |
| Cryptographic Key Generation | Same |
| Cryptographic KeyStore | Same |
| Secure Debugging | Optional |
| Factory Reset of Platform | Not in PSA Certified Level 3 |
| Authenticated Access Control | Not in PSA Certified Level 3 |
| Physical Attacker Resistance | Same |
| Secure Debugging | Optional |

# Appendix C Mapping with Matter Countermeasures

This appendix provides with supported countermeasures from Matter specification [Matter]. As Matter has a wider technical scope than this Profile, it also defines a wider set of countermeasures that need to be considered for Matter products and that are not covered by this Profile.

| ID | Matter Countermeasure | PSA Certified Level 3 SESIP Profile SFR |
|---|---|---|
| CM23 | All Devices include a Device Attestation Certificate and private key, unique to that Device. | Attestation of Platform Genuineness |
| CM77 | All Devices protect the confidentiality of attestation (DAC) private keys. The level and nature of protection for these keys may vary depending on the nature of the Device. | Cryptographic KeyStore |
| CM107 | Devices include protection (if it exists) against known remote attacks that can be used to extract or infer cryptographic key material. | Cryptographic Operation |
| CM35 | Factory reset removes all local data and key material created during or after commissioning except data explicitly required to persist across resets. | Factory Reset of Platform |
| CM62 | Protection against physical attacks (especially those that impact cybersecurity) is needed for some Devices, as determined by the manufacturer. | Physical attacker resistance |
| CM45 | Configuration for secure channel protocol is carefully negotiated and validated by both parties. | Secure communication |
| CM21 | Devices have cryptographically signed firmware, including all firmware and software on the Device. | Secure Initialization |
| CM22 | Devices have a verified boot based in an immutable root of trust to verify the authenticity of firmware. | Secure Initialization |

## Acknowledgements

GlobalPlatform develops the SESIP Evaluation Methodology
**https://globalplatform.org/sesip/**

This document was written for Arm by ProvenRun
**http://www.provenrun.com**