ETSI TS 103 732 V1.1.1 (2021-11)

CYBER;
Consumer Mobile Device Protection Profile

Reference
DTS/CYBER-0052

Keywords
cybersecurity, mobile, privacy, terminal

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM®** and the GSM logo are trademarks registered and owned by the GSM Association.

BLUETOOTH® is a trademark registered and owned by Bluetooth SIG, Inc.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Cyber Security (CYBER).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Introduction

Consumer mobile devices like smartphones are becoming the entrance to digital services, such as mobile banking, electronic identity verification, digital key management, etc. Meanwhile more and more security attack vectors are being explored, such as malicious applications, network eavesdropping. Defining security and assurance requirements for mobile devices can mitigate potential risks and drive the mobile device security to an appropriate level in order to protect users of such mobile devices.

The present document identifies key assets to be protected in typical consumer usage scenarios and identifies security threats associated to these key assets. The identified threats are mitigated by security objectives, which are in their turn fulfilled by implementing appropriate security functional requirements.

The present document is defined as a Protection Profile (hereafter called PP) following PP structure from the CC standards [1], [2], [3] and therefore can be used for third party CC security assessments and certification. Notice that the present document has not been evaluated or certified as a formal PP.

The requirements in the present document take published standards, recommendations and guidance in clause 2 into consideration.

1 Scope

The present document defines a PP for Consumer Mobile Device (CMD), which is typically a user-customisable device utilising an operating system, supporting installation and maintenance of applications, with wireless internet connectivity, high computation power and rich user interface, such as smartphones or tablets, used for various purposes by the individual owner.

The present document identifies key assets of the CMD to be protected and identifies the threats associated to them and the functional capabilities (objectives and security functional requirements) that are required to mitigate those threats. Finally, the present document specifies the security assurance requirements against which the CMD security can be assessed in a CC security evaluation.

The present document is intended for CMD manufacturers implementing those security requirements for device certification and for third parties looking to assess the security functions on CMD such as evaluators.

The Target Of Evaluation (TOE) described by the present document is a consumer mobile device. The following items are excluded from the scope:

- all applications (apps) downloaded by a human user and pre-installed non-system permission apps which can be uninstalled by the human user;
- all peripheral devices, including any data residing on these devices and any services associated with these devices, for example memory card;
- CMD features related to cellular mobile communication, including secure element which stores user credentials for cellular mobile communication, for example UICC [i.6];
- features related to multiple authenticated human users using the same CMD.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference/.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] Common Criteria for Information Technology Security Evaluation: "Part 1: Introduction and General Model", version 3.1 revision 5, CCMB-2017-04-01, April 2017.

[2] Common Criteria for Information Technology Security Evaluation: "Part 2: Security Functional Components", version 3.1 revision 5, CCMB-2017-04-02, April 2017.

[3] Common Criteria for Information Technology Security Evaluation: "Part 3: Security Assurance Components", version 3.1 revision 5, CCMB-2017-04-03, April 2017.

[4] SOG-IS Crypto Evaluation Scheme Agreed Cryptographic Mechanisms, SOG-IS Crypto Working Group, version 1.2, January 2020.

[5] Common Methodology for Information Technology Security Evaluation: "Evaluation methodology", version 3.1 revision 5, CCMB-2017-04-04, April 2017.

[6] IETF RFC 2818: "HTTP over TLS".

[7] IETF RFC 5246: "The Transport Layer Security (TLS) Protocol Version 1.2".

[8] IETF RFC 5280: "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile".

[9] IETF RFC 5288: "AES Galois Counter Mode (GCM) Cipher Suites for TLS".

[10] IETF RFC 5289: "TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM)".

[11] IETF RFC 8446: "The Transport Layer Security (TLS) Protocol Version 1.3".

[12] Bluetooth® SIG: "Bluetooth Core Specification, v4.1".

[13] Bluetooth® SIG: "Bluetooth Core Specification, v4.2".

[14] Bluetooth® SIG: "Bluetooth Core Specification, v5.0".

[15] Bluetooth® SIG: "Bluetooth Core Specification, v5.1".

[16] Bluetooth® SIG: "Bluetooth Core Specification, v5.2".

[17] IEEE 802.11™-2016: "IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications".

[18] IEEE 802.1X™-2020: "IEEE Standard for Local and Metropolitan Area Networks--Port-Based Network Access Control".

[19] IETF RFC 5216: "The EAP-TLS Authentication Protocol".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI EN 303 645 (V2.1.1): "CYBER; Cyber Security for Consumer Internet of Things: Baseline Requirements".

[i.2] Secure Communications Alliance IoT PP Working Group: "IoT Secure Element Protection Profile", version 1.0.0, December 19, 2019.

[i.3] ISO/IEC TS 30104:2015: "Information Technology - Security Techniques - Physical Security Attacks, Mitigation Techniques and Security Requirements".

[i.4] ISO/IEC 30107-4:2020: "Information Technology - Biometric Presentation Attack Detection - Part 4: Profile for testing of mobile devices".

[i.5] Global Platform Security Evaluation Standard for IoT Platforms, version 1.0, March 2020.

[i.6] ETSI TS 102 221: "Smart Cards; UICC-Terminal interface; Physical and logical characteristics (Release 16)".

[i.7] GSMA SGP.22: "RSP Technical Specification".

[i.8] ETSI TS 133 102: "Universal Mobile Telecommunications System (UMTS); 3G security; Security architecture".

[i.9] ETSI TS 133 401: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; 3GPP System Architecture Evolution (SAE); Security architecture".

[i.10] ETSI TS 133 501: "5G; Security architecture and procedures for 5G System".

[i.11] Security IC Platform Protection Profile with Augmentation Packages version 1.0, February 2014, BSI-CC-PP-0084-2014.

[i.12] GSMA SGP.25: "Embedded UICC for Consumer Devices Protection Profile".

[i.13] GSMA SGP.08: "Security Evaluation of Integrated eUICC".

[i.14] GSMA SGP.24: "GSMA SGP.24: "RSP Compliance Process".

[i.15] NIST SP 800-90A Rev. 1: "Recommendation for Random Number Generation Using Deterministic Random Bit Generators".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

best practice cryptography: cryptography that is suitable for the corresponding use case and has no indications of a feasible attack with current readily available techniques

consumer mobile device: user customizable device utilising an operating system, supporting installation and maintenance of applications, with wireless internet connectivity, high computation power and rich user interface, used for various purposes by the individual owner

EXAMPLE: Smartphones and tablets are typical consumer mobile devices.

device ID: unique identity of a consumer mobile device, which is not resettable

EXAMPLE: International Mobile Equipment Identity (IMEI) and Serial Number (SN).

device unique key: unique key stored in the device hardware during the initial manufacturing of the device, which is used to derive or encrypt other keys

human user: physical person using the CMD including actions taken by an object on behalf of the physical person, such as a stylus pen

NOTE: Both physical person and external IT entity such as trusted peer device are users of the TOE as defined in [3], to differentiate the two types of users, the term "human user" is used to refer to a physical person. In the present document, user data refers to human user data.

security problem: statement, which in a formal manner defines the nature and scope of the security that the TOE is intended to address [1]

security objective: statement of an intent to counter identified threats and/or satisfy identified organisation security policies and/or assumptions [1]

security functional requirement: requirement, stated in a standardised language, which is meant to contribute to achieving the security objectives for a TOE [1]

security assurance requirements: description of how assurance is to be gained that the TOE meets the SFRs.

system permission: permission granted by the operating system to manage the operating system (such as power off), provide core functions (such as SMS and Telephone), or access to underlying software and hardware interfaces

target of evaluation: set of software, firmware and/or hardware possibly accompanied by guidance [1]

TOE security functionality: combined functionality of all hardware, software, and firmware of a TOE that are relied upon for the correct enforcement of the security functional requirements

TOE software: operating system and pre-installed system permission apps which are updated together from a Trusted Update Source

trusted peer device: device with a trusted relationship with the TOE for purposes of interaction with the TOE

EXAMPLE: Screen sharing, file sharing, moving the entire content from an old device to a new device.

trusted update source: central repository from which updates to the TOE software can be downloaded

NOTE: This repository is typically managed by the TOE developer/OEM and authenticity and integrity of updates are typically guaranteed by digitally signing the updates.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

API	Application Program Interface
CC	Common Criteria
CMD	Consumer Mobile Device
DEK	Data Encryption Key
DUK	Device Unique Key
EAL	Evaluation Assurance Level
ECDH	Elliptic Curve Diffie-Hellman
EEPROM	Electrically Erasable Programmable Read Only Memory
FAR	False Acceptance Rate
FCS	Functional class Cryptographic Support
FDP	Functional class user Data Protection
FIA	Functional class Identification and Authentication
FMT	Functional class security ManagemenT
FPR	Functional class PRivacy
FPT	Functional class Protection of the TSF
FRR	False Rejection Rate
FTP	Functional class Trusted Path/Channels
GCF	Global Certification Forum
GPS	Global Positioning System
GSM	Global System for Mobile
IT	Information Technology
JTAG	Joint Test Action Group
KEK	Key Encryption Key
NFC	Near Field Communication
OEM	Original Equipment Manufacturer
OS	Operating System
PCS	Personal Communication Service
PIN	Personal Identification Number
PP	Protection Profile
PRF	Pseudo Random Function
QR	Quick Response
RNG	Random Number Generator
SAR	Security Assurance Requirement
SESIP	Security Evaluation Standard for IoT Platforms
SFR	Security Functional Requirement
SMS	Short Message Service
SoC	System-on-Chip
SOG-IS	Senior Officials Group Information Systems Security

ST	Security Target
TLS	Transport Layer Security
TOE	Target Of Evaluation
TSF	TOE Security Functionality
UI	User Interface
USB	Universal Serial Bus
WLAN	Wireless Local Access Network

4 TOE Definition

4.1 TOE Overview

The TOE described by the present document is a subset of a CMD as shown in Figure 1. The CMD includes hardware, an operating system and apps. Apps are categorised as pre-installed system permission apps, pre-installed non-system permission apps, and downloaded apps. Examples of a CMD include smart phone, tablet and other device with similar capabilities. Human users can customise their CMDs (modify their UI appearance, download apps, etc.) and use these devices for a wide range of purposes.

The TOE includes hardware, the operating system and pre-installed system permission apps that are delivered with the CMD out of the box. Pre-installed non-system permission apps and apps that are installed later by the human user (downloaded apps) are not considered part of the TOE. However, if a pre-installed non-system permission app cannot be uninstalled by the human user, it is included in the TOE.

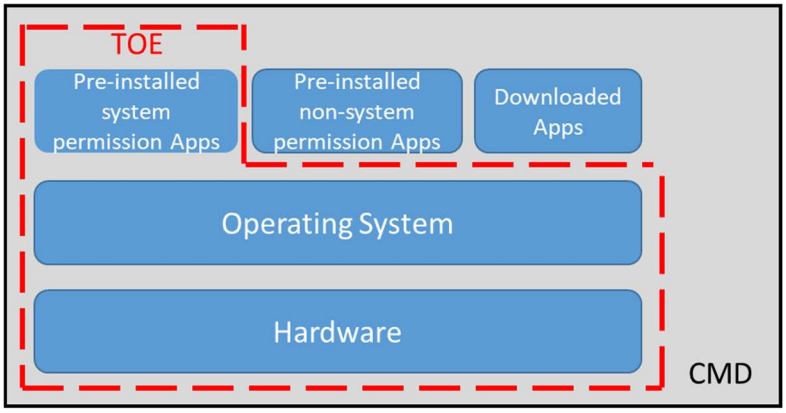


Figure 1: TOE boundary

The hardware of the TOE includes the hardware platform, physical enclosure and peripheral components such as sensors and the display. The hardware does not include any devices removable by a human user, including any data residing on these devices and any services associated with these devices, for example a memory card.

Any data on these devices or services associated with these devices is out of scope of the TOE.

The operating system of the TOE controls and manages the hardware and the apps (both pre-installed and downloaded) and provides the user operation interface and application programming interface(s).

The pre-installed apps are apps that are already present on the CMD when it is delivered to the consumer. Pre-installed apps are divided into two kinds:

1) Pre-installed system permission apps: apps which have permissions to manage the operating system (such as power off), provide core functions (such as SMS and Telephone) or access to underlying software and hardware interfaces. These apps require permissions granted by the operating system and cannot be revoked by the human user. These permissions are denoted as system permissions for the purpose of the present document. System permission apps include apps that provide core operating system functionality or security enforcement functionality, apps signed by a platform key from the operating system provider, and other apps allowed by the TOE developer to get such permissions. System permission apps can also require permissions granted by the human user in addition to system permissions.

2) Pre-installed non-system permission apps: apps which do not require system permissions. Non-system permission apps can have permissions granted by human user to access to user data and/or permissions granted by the operating system which are necessary to the operation of apps. Non-privileged apps can usually be uninstalled.

Downloaded Apps are apps that are downloaded and installed by the human user and can subsequently be uninstalled by the human user. Downloaded Apps do not have system permissions.

Security functionalities of the TOE related to cellular mobile communication are defined in [i.8], [i.9], [i.10] and will be certified by Global Certification Forum (GCF) and PCS Type Certification Review Board (PTCRB). Security of a secure element which stores user credentials for cellular mobile communication, e.g. UICC, eUICC [i.7], and integrated eUICC is specified in [i.11], [i.12], [i.13] and [i.14], and will be certified by a CC Certification Body. Therefore, these functions are out of scope for the present document. It is assumed that the TOE meets applicable security requirements defined in these specifications and TOE developer should provide evidence for such assumption if the evaluation of the TOE depends on such evidence, such as certificate of eUICC.

Functionality related to multiple authenticated human users using the same CMD is also out of scope. Whether one human user can see or alter the user data of another human user, or whether one human user can delete the account of another human user is not covered in the present document.

The present document is intended to be used for TOE which does not carry an altered operating system which enables the user to manage system permissions, such as rooted device.

The TOE developer shall define the TOE clearly as part of submission for CC evaluation.

4.2 Usage and Major Security Features

The TOE is a subset of a CMD with wireless connectivity, high computation power and rich user interface. A human user can customise the device by downloading apps and changing settings. A human user can perform a wide range of actions with the TOE, such as make phone and video calls, perform various productivity tasks, play games, music and videos, and access the Internet.

The TOE interacts with its environment as shown in Figure 2.

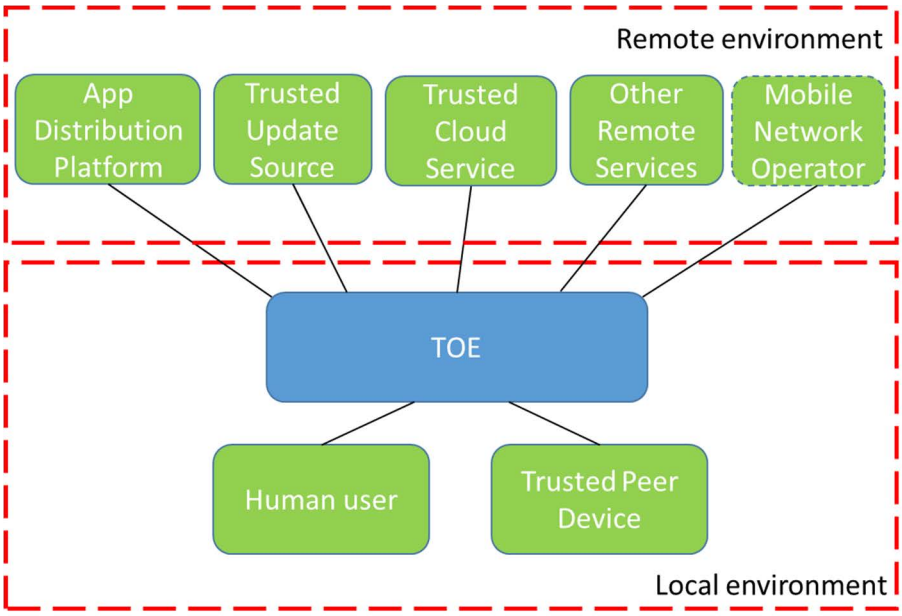


Figure 2: TOE environment

The TOE has a local environment with:

- A human user, who physically interacts with the TOE across its user interface(s).
- (Optionally) one or more Trusted Peer Devices, which can interact with the TOE in actions such as screen sharing or collaborative editing.

The TOE has a remote environment with:

- A Trusted Update Source, from which the TOE can download updates for TOE software. These updates are digitally signed, and the TOE checks whether this signature is correct.
- App Distribution Platform of the TOE developer and/or OS developer, from which the TOE can download and install apps. The App Distribution Platform will detect malicious in-app behaviour, conducts privacy disclosure inspections for apps that call, collect or upload sensitive data from human users without permission, as well as scans apps for the presence of loopholes, vulnerabilities or backdoors. How the App Distribution Platform performs security checks of applications is out of scope of present document. The App Distribution Platform application on the TOE is in scope. It is assumed that only App Distribution Platform applications from TOE developer and/or OS developer will be pre-installed. A human user may install additional App Distribution Platform applications, in which case the human user takes the responsibility for the security of apps downloaded and installed from any additional App Distribution Platforms.
- Trusted Cloud Service provided by the TOE developer. A human user can use cloud service to access user data in the TOE. It is assumed that the cloud service will be secure, and vulnerabilities of cloud service are not in scope of present document. The security of the connection from the TOE to any trusted cloud service is assured by trusted channel requirements defined in clause 8.1.7.
- Other Remote Services offered by third parties, such as enterprise services, additional App Distribution Platforms used by a human user, websites, gaming servers, etc. The present document provides no assurance in these remote services, so it is up to the human user to trust a particular remote service.
- (Optionally) one or more Mobile Network Operators when the TOE supports cellular radio connection. This will be assured by GCF and therefore it is assumed that the Mobile Network Operator will provide secure cellular communication with the TOE.

The major security features are:

- Authentication of human user: to ensure that the human user is authenticated by the TOE before he/she can fully use the TOE (it may be possible to make very limited use of the TOE before authentication, such as making emergence call).
- Authentication of Trusted Peer Devices: the TOE allows other devices to act as a trusted peer device for purposes such as screen sharing and collaborative editing. To be able to do this, these devices first authenticate themselves.
- Secure communication: the TOE offers one or more secure communication channels, protected against unauthorised modification and unauthorised disclosure. These channels can subsequently be used by apps and by the TOE itself for various communication purposes.
- Secure updating of TOE software: the TOE can update the TOE software by downloading an update from a Trusted Update Source to address known vulnerabilities in a timely manner.
- Secure updating of apps: the TOE can update pre-installed non-system permission apps and Downloaded Apps by downloading an update from the App Distribution Platform.
- Self-protection and integrity verification of the TOE: the TOE protects both itself and other apps against malicious apps who can try to hack into the TOE. The TOE also checks its own integrity every time it starts up to check whether it has been altered.
- Protecting user data at different levels of security: the TOE supports classification of user data according to risk levels and usage scenarios and protects user data to ensure it is accessed by the right person on the right device in the right condition.
- Permission management of apps: to ensure that apps can only access to user data on the TOE and services provided by the TOE which are essential to their operation and where permission has been granted by the human user and/or by the operating system.
- Protection against tracking by app developers and advertisers: The TOE can provide an alias to app developers and advertisers, so that they have limited tracking of the human user. The human user can replace that alias with another alias to limit this tracking.

- Protection against physical attacks: the TOE can protect keys, data used to derived keys and other sensitive data from being read or modified by physical attacks.

4.3 Additional Hardware/Software/Firmware required by the TOE

The TOE is standalone and does not require any additional hardware/software/firmware.

4.4 Conformance Claim

The present document:

- claims conformance to CC v3.1 Release 5 [1], [2], [3];
- is CC Part 2 [2] conformant (extended with FCS_RNG and FCS_CKH) and CC Part 3 [3] conformant (with refinement of ALC_FLR.3);
- does not claim conformance to any other PP;
- conforms to the package EAL2 augmented with ALC_FLR.3;
- requires demonstrable conformance.

Demonstrable conformance is chosen for the present document, rather than strict or exact. This was done because it is expected that some components of the TOE, such as the SoC, (parts of) the operating system, biometric solutions, and/or communication ICs, can have already been certified, and that the developer of the CMD wishes to re-use the certified status of these components to simplify the CMD evaluation.

Demonstrable conformance allows the re-use of these components when they have the required functionality, but their Security Target describes this functionality in a different way than the present document.

5 Security Problem Definition

5.1 Assets and interfaces of the TOE

Assets to be protected:

- User data assets stored in the TOE and in transit to the Trusted Cloud Service, such as:
  - user files: photos, videos, etc.;
  - user location: GPS information, location record, etc.;
  - non-TOE account information;
  - user communication: communication records, address books, emails, SMS, chat session, audio/video calls, etc.;
  - credentials for other devices and/or services, e.g. login password for web service;
  - data collected by sensors: acceleration sensor, blood sugar, body fat ratio, heart rate, blood pressure, etc.;
  - App data: list of installed apps, user data in apps, etc.
- User data assets are grouped in three classes, Low, Medium and High (see P.DATA_CLASSIFICATION), each requiring more stringent protection.
- TSF data: data which is used for the enforcement of operations of security functions, such as configuration data, user authentication data.

- The operating system and apps included in the TOE.

Interfaces of the TOE:

- The local wireless interface(s): these are interface(s) to local wireless networks.
- The wide-area network interface(s): these are interface(s) to wide area networks such as the Internet. This interface generally lies on top of either the local wireless interface or interfaces such as GSM, 3G, 4G, 5G (which are out of scope, see clause 4.2).
- The user interface: this interface includes the screen, buttons, speaker, etc.
- The physical interface: this is the physical enclosure of the TOE, includes JTAG ports, USB (and similar) ports, charging ports, etc.
- The application interface: this is the API that applications can use to interact with the underlying operating system.

5.2 Threat agents and threats

Threat Agents:

- TA.LOCAL: a threat agent in the general vicinity of a TOE when it is used, and therefore has access to the local wireless interface.
- TA.REMOTE: a threat agent with access to the wide-area interface.
- TA.PHYSICAL: a threat agent who has physical access to the TOE, and therefore to both the user interface and the physical interface.
- TA.FLAWAPP: a malicious or poorly programmed app that the human user has installed on the TOE and that therefore has access to the application interface, and possibly to the local wireless interface and/or the wide-area network interface.

NOTE 1: The same person or entity can be multiple threat agents simultaneously.

Threat Agents are limited to Basic attack potential. For further detail and examples on attack potential, see annexes B3, B4 and B5 of [5].

The present document identifies threats based on each interface defined in clause 5.1 on what nefarious actions each of the threat agents could perform on that interface. Similar threats are subsequently combined into one threat where possible. The threats are identified as below:

T.EAVESDROP - TA.LOCAL, TA.REMOTE or TA.FLAWAPP read communication between the TOE and other entities and thereby access confidential user data assets in transit.

T.SPOOF - TA.LOCAL or TA.REMOTE create a spoofed device or service and wait for the TOE to connect to that device or service. Once the TOE connects to the spoofed device or service the threat agents actively or passively extract user data assets from the TOE.

T.MODIFY-COMMS - TA.LOCAL or TA.REMOTE initiate or intercept communication between the TOE and other entities and thereby modify user data assets in transit.

T.COUNTERFEIT_DEVICE: TA.PHYSICAL or TA.LOCAL attempts to connect to the TOE and thereby gain access to user data assets with a device masquerading as a trusted peer device.

NOTE 2: The exact capabilities of a trusted peer device are specified in FIA_UAU.2.2.

T.IMPERSONATE - TA.PHYSICAL impersonates the legitimate user of the TOE thereby gaining access to the User Data Assets.

EXAMPLE 1: Impersonation includes the guessing of passwords, PINs or patterns and/or the spoofing of biometrics.

NOTE 3: Attacks based on the human user revealing their credentials, such as voluntarily disclosing the password to an attacker, writing down a PIN where an attacker can see it, or an attacker replacing the TOE with a similar device where the human user enters their credentials on, after which the attacker uses these credentials to login on the real TOE, are out-of-scope of the present document.

T.PHYSICAL - TA.PHYSICAL attempts to gain access to assets by accessing physical interfaces of the TOE.

EXAMPLE 2: Physical interfaces include JTAG ports, USB (and similar) ports, charging ports, probing the PCB, direct access to the TOE storage media.

T.RECOVER_DATA - A human user sells their TOE and attempts to remove all user data assets beforehand, but the new user (TA.PHYSICAL) is still able to retrieve some or all of these user data assets.

T.MODIFY_DEVICE - TA.PHYSICAL obtains a TOE, modifies that TOE, and reinserts that TOE into the supply chain. Later on a legitimate human user buys this CMD and the modified CMD allows compromise of the assets of that user.

T.FLAWAPP-ACCESS - TA.FLAWAPP attempts to access to user data assets that it should not be able to access and subsequently modify them or export them to third parties. This includes additional data gathered from the TOE sensors (GPS, camera, microphone, etc.).

T.PERSISTENT - Successful realisation of one of the other threats can lead to a persistent presence on the TOE constituting an ongoing threat in itself. The threat agent associated with the other threat can possibly control the device.

T.NEW_ATTACKS - Any of the threat agents can make use of newly discovered vulnerabilities in the TOE and thereby becomes able to execute one or more of the other threats.

T.FLAWAPP_HACKS_TOE - TA.FLAWAPP attempts to modify the security behaviour of the TOE.

T.FLAWAPP_HACKS_OTHER_APPS - TA.FLAWAPP attempts to modify the behaviour of other apps, without access permission to the peer app being granted through the operating system or the peer app.

The mapping of threats with interfaces of the TOE is shown in annex C.

5.3 Organisational Security Policies

P.DATA_CLASSIFICATION - The TOE developer classifies groups of user data assets into at least three different classes, according to policies such as the potential harm that may result to the human user if the user data asset were inappropriately accessed, used, or disclosed:

- Low: user data assets to which unauthorised access is expected to have limited adverse effect on the human user. Low user data assets can only be decrypted on the TOE.
- Medium: user data assets to which unauthorised access can have serious adverse effect on the human user. Medium user data assets can only be decrypted on the TOE following the first successfully user authentication.
- High: user data assets to which unauthorized access can have severe adverse effect on the human user. High user data assets can only be decrypted on the unlocked TOE following a successful user authentication.

5.4 Assumptions

A.APP_DISTRIBUTION_PLATFORM - It is assumed that the human user will only install apps which have been downloaded from the App Distribution Platform of the TOE developer or OS provider, which has performed best practice security checks on these apps. Security checks include but are not limited to:

- detect malicious in-app behaviour;
- conduct privacy disclosure inspections for apps that call, collect or upload sensitive data from human user without permission;
- scan apps for the presence of loopholes, vulnerabilities or backdoors;
- verify an App developer has a valid developer certificate.

NOTE: The APP_DISTRIBUTION_PLATFORM does best practice security checking taking into account of state of art, but it is not assumed APP_DISTRIBUTION_PLATFORM is free of malicious apps.

A.Trusted_Update_Source - It is assumed that the TOE will trust the Trusted Update Source to install secure updates delivered by the update source.

A.Trusted_Cloud_Service - It is assumed that the cloud service provided by the TOE developer is secure and unauthorised parties cannot access to the user data assets on the TOE via remote cloud services.

A.PASSWORD_PIN_PATTERN - It is assumed that the human user will not downgrade the authentication of the TOE or choose easily guessable passwords/PINs/Patterns.

EXAMPLE: Easily guessable passwords and PINS include aaaa, 1234, birthdate of the human user.

6 Security Objectives

6.1 Security Objectives for the TOE

O.PROTECT_COMMS - The TOE can setup communication channels with other devices/services that are protected against disclosure and allow detection (either by the TOE or by the other device/service) of any modification of data exchanged over these channels. The TOE authenticates such devices/services before allowing communication. The TOE also allows these services/devices to authenticate the TOE.

NOTE 1: This security objective includes both "standard" hardware channels such as Bluetooth® [16], but also covers secure communication channels, either on top of these "standard" channels or on otherwise unprotected channels.

O.UPDATES - The TOE supports updating of its operating system and apps.

O.AUTHENTICATED_UPDATES - The TOE only allows updates of its operating system and apps when these updates are authenticated as being from a trusted source.

O.PROTECT_ASSETS_AT_REST - The TOE ensures that assets are unreadable when not in use, e.g. by encryption.

O.SECURE_WIPE - The TOE is able to make user data assets permanently unreadable.

O.CRITICAL_STORAGE - The TOE provides storage for critical security parameters such that physical attackers are unable to access these parameters.

EXAMPLE: Critical security parameters can be stored user credentials (or keys used to encrypt them), keys used to encrypt assets (see O.PROTECT_ASSETS_AT_REST), secure boot (see O.SECURE_BOOT), or authenticated updates (see O.AUTHENTICATED_UPDATES)

O.ACCESS_CONTROL - The TOE ensures that apps only gain access to user data assets that they are specifically allowed by the human user or the operating system to have access to.

O.SECURE_BOOT - The TOE, at the start of its boot process, checks the TSF to ensure it has not been tampered with.

O.AUTHENTICATE_USER - The TOE will identify and authenticate the human user of the TOE before allowing that human user has full access to the TOE functionality.

O.CRYPTOGRAPHY - The TOE provides best practice cryptographic functionality (encryption, decryption, signing, signature checking), to implement other security objectives.

O.RANDOMS - The TOE provides best practice random number generation functionality to support O.CRYPTOGRAPHY.

O.DATA_CLASSIFICATION - The TOE supports and encrypts at least three different classes of user data assets:

- Low: user data assets that can only be decrypted on the TOE.
- Medium: user data assets that can only be decrypted on the TOE following the first successfully user authentication.

- High: user data assets that can only be decrypted on the unlocked TOE following a successfully user authentication.

O.AUTHENTICATE_PEER_DEVICE - The TOE allows other devices to authenticate themselves to the TOE and become a trusted peer device.

O.PERSISTENT - The TOE is able to detect and remove malevolent persistent presences from itself.

NOTE 2: A possible form of detection is through the secure boot process (O.SECURE_BOOT). A possible form of removal is the reset to factory settings.

O.SELF_PROTECTION - The TOE protects itself against apps attempting to modify TSF data and its security behaviour.

O.SEPARATION - The TOE provides a separate security domain for each app, protecting them against unauthorised modification by other apps.

6.2 Security Objectives for the Operational Environment

OE.APP_DISTRIBUTION_PLATFORM - The operational environment ensures that the human user of the TOE is instructed to use the App Distribution Platform of the TOE developer or OS provider which performs security checks on these apps.

OE.Trusted_Update_Source - The operational environment ensures the security update delivered by the Trusted Update Source is sufficiently protected from tampering and is secure to be installed by the TOE.

OE.Trusted_Cloud_Service - The operational environment ensures the cloud services are sufficiently protected from unauthorised access.

OE.PASSWORD_PIN_PATTERN - The operational environment ensures that human user of the TOE is instructed not to disable or downgrade the authentication of the TOE or choose easily guessable passwords/PINs/patterns such as aaaa, 1234, birthdates and the like.

6.3 Security Objectives Rationale

Threat	Rationale
T.EAVESDROP	This threat is countered by O.PROTECT_COMMS that enables protection of the communication channel(s) against disclosure. This objective is supported by O.CRYPTOGRAPHY to encrypt these channels and O.RANDOM to provide random numbers for key generation.
T.SPOOF	This threat is countered by O.PROTECT_COMMS that ensures that the TOE authenticates devices it connects to.
T.MODIFY-COMMS	This threat is countered by: • O.PROTECT_COMMS that enables protection of the communication channel(s) against disclosure. This objective is supported by O.CRYPTOGRAPHY to encrypt these channels and O.RANDOM to provide random numbers for key generation. • O.AUTHENTICATED_UPDATES preventing unauthorised updates of any part of the TOE.
T.IMPERSONATE	This threat is countered by O.AUTHENTICATE_USER ensuring that only authenticated user can access the device functionality.
T.PHYSICAL	This threat is countered by: • O.PROTECT_ASSETS_AT_REST ensuring that assets are encrypted (or erased) when not in use thus preventing a physical attacker who directly accesses TOE storage media from reading that data. This objective is supported by O.CRYPTOGRAPHY to encrypt/decrypt this data and O.CRITICAL_STORAGE to securely store the encryption/decryption keys. • O.SECURE_BOOT to ensure that the integrity of the TOE is checked every time it boots thus preventing undetected loss of integrity from physical attack. • O.ACCESS_CONTROL to ensure that access to physical interface such as charging interface is controlled by the human user.
T.FLAWAPP-ACCESS	This threat is countered by O.ACCESS_CONTROL preventing the app from gaining access to user data assets which they do not have permission to access, so they cannot be modified or exported.
T.FLAWAPP_HACKS_TOE	This threat is countered by O.SELF_PROTECTION, stating that the TOE protects itself against apps attempting to alter its security behaviour.
T.FLAWAPP_HACKS_OTHER_APPS	This threat is countered by O.SEPARATION, stating that the TOE provides a security domain for each app, thereby separating them from other apps.
T.PERSISTENT	This threat is countered by O.PERSISTENT and possibly by O.SECURE_BOOT allowing detection and removal of persistent presences of malevolent entities from the TOE.
T.MODIFY_DEVICE	This threat is countered by O.SECURE_BOOT that can detect integrity issues with the TOE and O.CRITICAL_STORAGE that prevents attackers from modifying the keys from O.SECURE_BOOT, which would make it ineffective.
T.COUNTERFEIT_DEVICE	This threat is countered by O.AUTHENTICATE_PEER_DEVICE which forces devices to authenticate in order to become a trusted peer device.
T.RECOVER_DATA	This threat is countered by O.SECURE_WIPE, ensuring that the user data assets are permanently unreadable.
T.NEW_ATTACKS	This threat is countered by O.UPDATES, allowing the TOE to be updated to stay ahead of new attacks and/or discovered vulnerabilities.
P.DATA_CLASSIFICATION	This policy is directly implemented by O.DATA_CLASSIFICATION.
A.APP_DISTRIBUTION_PLATFORM	This assumption is directly implemented by OE.APP_DISTRIBUTION_PLATFORM.
A.PASSWORD_PIN_PATTERN	This assumption is directly implemented by OE.PASSWORD_PIN_PATTERN.
A.Trusted_Update_Source	This assumption is directly implemented by OE.Trusted_Update_Source.
A.Trusted_Cloud_Service	This assumption is directly implemented by OE.Trusted_Cloud_Service.

7 Extended Components Definition

7.1 Definition of the family Random Number Generation (FCS_RNG)

NOTE: This definition is based on [i.2].

Family behaviour

This clause describes the functional requirements for the generation of random numbers, which can be used as secrets for cryptographic purposes or authentication. The security functional components are defined in an additional family (FCS_RNG) of the Class FCS (Cryptographic support). The components address the type of the random number generator and the quality of the random numbers.

Component levelling

FCS_RNG.1, Generation of random numbers, requires that the random numbers meet a defined quality metric.

Management: FCS_RNG.1

There are no management activities foreseen.

Audit: FCS_RNG.1

There are no actions defined to be auditable.

FCS_RNG.1 Random number generation

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RNG.1.1 The TSF shall provide a [selection: physical, non-physical true, deterministic, hybrid physical, hybrid deterministic] random number generator.

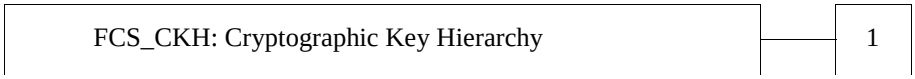
FCS_RNG.1.2 The TSF shall provide random numbers that meet [assignment: a defined quality metric].

7.2 Definition of the family Cryptographic Key Hierarchy (FCS_CKH)

Family behaviour

This clause describes the functional requirements for a cryptographic key hierarchy, a related set of keys that together protect data. Some keys in the key hierarchy will encrypt the data, while the other keys are used to encrypt and/or derive other keys in the key hierarchy. The sole security functional component is defined in an additional family (FCS_CKH) of the Class FCS (Cryptographic support). The component addresses the data that is protected, and how the keys in the key hierarchy are derived and protected.

Component levelling



FCS_CKH.1, Cryptographic key Hierarchy, requires definition of the key hierarchy and the data protected by the key hierarchy, and how the keys in the key hierarchy are derived and protected.

Management: FCS_CKH.1

There are no management activities foreseen.

Audit: FCS_CKH.1

There are no actions defined to be auditable.

FCS_CKH.1 Cryptographic Key Hierarchy

Hierarchical to:	No other components.
Dependencies:	FCS_COP.1 Cryptographic Operation.
	FCS_CKM.4 Cryptographic Key Destruction.

FCS_CKH.1.1 The TSF shall support a key hierarchy for [assignment: list of data to be protected by the key hierarchy].

FCS_CKH.1.2 The TSF shall ensure that all keys in key hierarchy are derived and/or generated according to key lengths and according to which standards].

FCS_CKH.1.3 The TSF shall ensure that all keys in the key hierarchy and all data used in deriving the keys in the hierarchy are protected according to [assignment: rules].

NOTE: The protection assignment in FCS_CKH.1.3 can use protection methods like encryption by other keys in the key hierarchy, keys that are deleted after use and re-generated/re-derived when needed, or secure storage of the key.

8 Security requirements

8.1 Security functional requirements

8.1.0 Introduction

The security functional requirements in this clause contain open operations that are to be completed by the TOE developer in the Security Target. This is necessary to allow different vendors to have different implementations instead of mandating a specific implementation. During the security evaluation it will be checked whether the security functional requirements with all completed operations meet the security objectives, as required by ASE_REQ.2.7C, which is included in the chosen assurance level EAL2.

8.1.1 Cryptographic Support (FCS)

FCS_RNG.1 Random number generation

FCS_RNG.1.1 The TSF shall provide a [selection: physical, non-physical true, deterministic, hybrid physical, hybrid deterministic] random number generator.

FCS_RNG.1.2 The TSF shall provide random numbers that meet [assignment: SOG-IS Agreed Cryptographic Mechanisms [4], or, where appropriate, national or regional cryptographic standards].

FCS_COP.1_Update Cryptographic operation

FCS_COP.1.1/Update The TSF shall perform signature verification of an Update_Package in accordance with a specified cryptographic algorithm [assignment: cryptographic algorithm] and cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: SOG-IS Agreed Cryptographic Mechanisms [4], or, where appropriate, national or regional cryptographic standards].

FCS_COP.1_User_Data_Assets Cryptographic operation

FCS_COP.1.1/UserDataAssets The TSF shall perform encryption and decryption of user data assets in accordance with a specified cryptographic algorithm [assignment: cryptographic algorithm] and cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: SOG-IS Agreed Cryptographic Mechanisms [4], or, where appropriate, national or regional cryptographic standards].

NOTE 1: User data assets are defined in clause 8.1.2.3.

FCS_CKH.1_Low Cryptographic key hierarchy

FCS_CKH.1.1/Low The TSF shall support a key hierarchy for the data encryption key(s) for Low user data assets.

FCS_CKH.1.2/Low The TSF shall ensure that all keys in the key hierarchy are derived and/or generated according to [assignment: description of how each key in the hierarchy is derived and/or generated, with which key lengths and according to which standards] ensuring that the key hierarchy uses the DUK directly or indirectly in the derivation of the data encryption key(s) for Low user data assets.

NOTE 2: The key derivation process requires that the data encryption key(s) for Low user data assets is derived directly from DUK or from another key which further derived from DUK. This ensures that Low user data assets can only be decrypted in the device which has the DUK. Example of Key derivation algorithm can be DRBG AES-256-CTR which meets NIST SP 800-90A [i.15].

FCS_CKH.1.3/Low The TSF shall ensure that all keys in the key hierarchy and all data used in deriving the keys in the hierarchy are protected according to [assignment: rules].

EXAMPLE 1: Examples of rules can be: keys and all data used in deriving the keys are stored in hardware based secure environment as defined in FPT_PHP.3.1, keys and all data used in deriving the keys are encrypted, or other rules defined by the TOE developer.

FCS_CKH.1_Medium/High Cryptographic key hierarchy

FCS_CKH.1.1/MediumHigh The TSF shall support a key hierarchy for the data encryption keys for Medium and High user data assets.

FCS_CKH.1.2/MediumHigh The TSF shall ensure that all keys in the key hierarchy are derived and/or generated according to [assignment: description of how each key in the hierarchy is derived and/or generated, with which key lengths and according to which standards] ensuring that the key hierarchy:

- uses the DUK directly or indirectly in the derivation of the data encryption keys for Medium and High user data assets; and
- uses the PIN, password, pattern or biometric template directly or indirectly in the derivation of the data encryption keys for Medium and High user data assets.

NOTE 3: The key derivation process requires that the data encryption key(s) for Medium/High user data assets is derived from a combination of the DUK and user authentication information. This ensures that Medium/High user data assets can only be decrypted in the device which has the DUK and after user is successfully authenticated. Example of Key derivation algorithm can be DRBG AES-256-CTR which meets NIST SP 800-90A [i.15].

FCS_CKH.1.3/MediumHigh The TSF shall ensure that all keys in the key hierarchy and all data used in deriving the keys in the hierarchy are protected according to [assignment: rules].

EXAMPLE 2: Examples of rules can be: keys and all data used in deriving the keys are stored in hardware based secure environment as defined in FPT_PHP.3.1, keys and all data used in deriving the keys are encrypted, or other rules defined by the TOE developer.

FCS_CKM.4 Cryptographic key destruction

FCS_CKM.4.1 The TSF shall destroy [assignment: keys from the key hierarchy for Low, Medium, and High keys] in accordance with a specified cryptographic key destruction method [selection:

- for volatile memory, by a single direct overwrite consisting of [selection: a random pattern, using the TSF's RNG, zeroes];
- for non-volatile EEPROM, by a single direct overwrite consisting of a random pattern, using the TSF's RNG, followed by a read-verify;
- for non-volatile flash memory, that is not wear-levelled, by [selection: a single direct overwrite consisting of zeros followed by a read-verify, a block erase that erases the reference to memory that stores data as well as the data itself];
- for non-volatile flash memory, that is wear-levelled, by [selection: a single direct overwrite consisting of zeros, a block erase];
- for non-volatile memory other than EEPROM and flash, by a single direct overwrite with a random pattern that is changed before each write];

that meets the following: [assignment: list of standards].

NOTE 4: When no standard is applicable in the assignment, it can be left none.

8.1.2 User Data Protection (FDP)

8.1.2.1 The Update Policy

The Update Policy defines:

- How often the TSF will check whether new Update Packages are available. Update Packages can replace (parts of) the TOE software and apps and can also replace the cryptographic data used to determine the validity (see FCS_COP.1_Update) of any future Update Packages.
- The conditions under which the TSF uses these Update Packages to update the TOE software, and optionally other apps not included in the TOE software, to a different version.

FDP_ACC.1_Update Subset access control

FDP_ACC.1.1/Update The TSF shall enforce the Update Policy on [Subjects: the TSF, Objects: the TOE_software, App, Update_Package, Operations: Check_For, Receive, Update_the_TOE_software, Update_App].

FDP_ACF.1_Update Security attribute based access control

FDP_ACF.1.1/Update The TSF shall enforce the Update Policy to objects based on the following: [[selection: the TOE_software [version number], App[version number]], Update_Package[version number, signature]].

FDP_ACF.1.2/Update The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

- the TSF shall be able to Check_For an Update_Package every [assignment: interval] and inform the human user when a new Update_package is available; and
- the TSF shall allow the TSF to Update_the_TOE_software with an Update_Package if and only if:
  - the TSF successfully verifies the signature of that Update_Package; and
  - the version number of the Update Package is not lower than the version number of the TOE_software; or
  - the update is pushed from the Trusted Update Source.
- the TSF shall allow the TSF to Update_App with an Update_Package if and only if:
  - the TSF successfully verifies the signature of that Update_Package; and

  - the version number of the Update_Package is not lower than the version number of the App; or
  - the update is pushed from the App Distribution Platform of the TOE developer or OS developer;
- the TSF shall Update_the_TOE_software and Update_App in an atomic way.

FDP_ACF.1.3/Update The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [assignment: other additional rules ensuring authenticity and integrity of the update package].

FDP_ACF.1.4/Update The TSF shall explicitly deny access of subjects to objects based on the following additional rules:

- the TSF shall not allow any TSF-mediated actions during its updating.

NOTE: Failure of correct installation of the update is handled in FPT_FLS.1.

8.1.2.2 The Permissions Policy

The Permissions Policy defines the access of apps to objects such as the camera, microphone, and address book and that either human user permits this access, or that this access has been granted by the operating system (in the case of some pre-installed system permission apps). The permission/revocation of this access is defined in FMT_SMF.1_Permissions.

FDP_ACC.1_Permissions Subset access control

FDP_ACC.1.1/Permissions The TSF shall enforce the Permissions Policy on [

- Subjects: Downloaded Apps, pre-installed system permission Apps, pre-installed non-system permission Apps;
- Objects: camera, microphone, GPS, contacts, calendar, call log, stored pictures, text messages, Device ID, the list of installed apps, [assignment: list of other sensitive user data assets and/or system services that can be accessed by an App];
- Operations: read, write].

FDP_ACF.1_Permissions Security attribute based access control

FDP_ACF.1.1/Permissions The TSF shall enforce the Permissions Policy to objects based on the following [Downloaded Apps, pre-installed system permission Apps, pre-installed non-system permission Apps], [camera, microphone, GPS, contacts, calendar, call log, stored pictures, text messages, Device ID, list of installed apps, [assignment: list of other sensitive user data assets and/or system services that can be assessed by an App]].

FDP_ACF.1.2/Permissions The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

- a Downloaded App is allowed to read from an object if this has been specifically allowed by the human user; and
- a Downloaded App is allowed to write to an object if this has been specifically allowed by the human user; and
- a pre-installed system permission App or pre-installed non-system permission App is allowed to read from an object if:
  - this permission was granted by the operating system; or
  - this has been specifically allowed by the human user; and
- a pre-installed system permission App or pre-installed non-system permission App is allowed to write to an object if:
  - this permission was granted by the operating system; or
  - this has been specifically allowed by the human user.

8.1.2.3 The Data Classification Policy

The TOE supports the following classification of user data assets:

- Low: This data is to be encrypted in such a way that the data can only be decrypted on the TOE itself, when the TOE is successfully powered on, low data is decrypted and can be accessed without user authentication, e.g. alarm.
- Medium: This data is to be encrypted in such a way that the data can only be decrypted on the TOE and when the user is logged in after first successfully authentication, e.g. calendar.
- High: This data is to be encrypted the same as Medium, but additionally it can only be accessed when the screen of the TOE is unlocked, e.g. sensitive health data.

Each of these classes of user data assets is encrypted (see FCS_COP.1_User_Data_Assets) by a Data Encrypting Key (DEK). These DEKs, in turn, can be encrypted themselves by Key Encryption Keys (KEKs), which can be encrypted with or derived from further KEKs, etc.

Each DEK or KEK can be randomly generated, or it can be derived from other keys (such as the DUK) or data (such as the user password), or a combination of random generation and derivation. The whole structure of data and keys used to derive/encrypt a DEK is called a key hierarchy, which typically starts from the DUK and/or user credentials and have derivation/encryption of KEKs in the middle of the hierarchy (see the FCS_CKH requirements).

DUK is a unique key stored in the device hardware during the initial manufacturing of the device, which is accessible only by a hardware cryptography module and not directly exposed to any device software.

To prevent attackers from decrypting the user data assets all keys and data used for derivation in the hierarchy are to be protected, by being encrypted with a KEK, or by being discarded after use and re-generated/re-derived when needed, or by being securely stored (see FPT_PHP.3).

When a user wishes to dispose of the TOE and permanently delete all user data assets, this can be done by deleting one or more of the appropriate keys from the key hierarchy, thereby ensuring that the data can no longer be decrypted as either the key needed for decryption has been deleted or it is not possible to decrypt that key (see FCS_CKM.4), or all user data assets can be erased.

FDP_ACC.1_User_Data_Asset_Decryption Subset access control

FDP_ACC.1.1/UserDataAsset The TSF shall enforce the User Data Asset Decryption Policy on [

- Subjects: the TSF;
- Objects: user data assets;
- Operations: decrypt].

FDP_ACF.1_User_Data_Asset_Decryption Security attribute based access control

FDP_ACF.1.1/UserDataAsset The TSF shall enforce the User Data Asset Decryption Policy to objects based on the following: [TSF, user data assets, [Low, Medium, High]].

FDP_ACF.1.2/UserDataAsset The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

- the TSF is allowed to decrypt Low user data assets if and only if the TOE is successfully powered on; and
- the TSF is allowed to decrypt Medium user data assets if and only if the TOE is successfully powered on and the user is successfully authenticated during the first authentication after power on; and
- the TSF is allowed to decrypt High user data assets if and only if the TOE is successfully powered on, the user is successfully authenticated and the screen of the TOE is not locked.

NOTE: The phrase "successfully powered on" means that the TOE has successfully booted up and completed all tests required by FPT_TST.1.

8.1.3 Identification and Authentication (FIA)

FIA_UAU.1 Timing of authentication

FIA_UAU.1.1 The TSF shall allow [assignment: list of TSF mediated actions not accessing user data asset stored before the action is performed unless permitted by the human user] on behalf of the human user to be performed before the human user is authenticated.

NOTE 1: Actions not listed in the assignment are not allowed before human user authentication.

EXAMPLE 1: The actions in FIA_UAU.1.1 can include taking a picture and making an emergency call.

FIA_UAU.1.2 The TSF shall require the human user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that human user.

FIA_UAU.2 User authentication before any action

FIA_UAU.2.1 The TSF shall require each trusted peer device to be successfully authenticated by: [selection:

- both TOE and trusted peer device being logged in at the same human user account connecting to a trusted server;
- the TOE scanning a QR code generated by the trusted peer device to establish a shared secret;
- the trusted peer device scanning a QR code generated by the TOE to establish a shared secret;
- the TOE reading the NFC label embedded in the trusted peer device to establish a shared secret;
- the trusted peer device reading the NFC label embedded in the TOE to establish a shared secret;
- the human user input a PIN generated by or stored in trusted peer device in the TOE;
- the human user input a PIN generated by or stored in the TOE into trusted peer device;
- [assignment: other authentication method(s) which establish a shared secret between the TOE and trusted peer device];

before allowing any TSF-mediated actions on behalf of that trusted peer device.

NOTE 2: The shared secret in FIA_UAU.2.1 can be pre-installed or generated in various ways, such as an earlier successful authentication attempt by one of the other authentication methods.

NOTE 3: The authentication can be required each time a peer trusted device is connecting to the TOE, or required once to build a trusted relationship after first successfully authentication.

FIA_UAU.2.2 The TSF shall allow each trusted peer device to [assignment: list of actions] after successful authentication.

EXAMPLE 2: Actions in FIA_UAU.2 can include screen sharing, restoring a backup, answering a (video) call to the other device, copying files by dragging, collaborative editing.

FIA_UAU.5 Multiple authentication mechanisms

FIA_UAU.5.1 The TSF shall provide [selection: password, PIN, pattern] and [selection: fingerprint, iris, face, voice, vein, [assignment: other authentication mechanisms]] to support human user authentication.

FIA_UAU.5.2 The TSF shall authenticate any human user's claimed identity according to the [assignment: rules describing how the multiple authentication mechanisms provide authentication].

EXAMPLE 3: Rules can be: "Always try biometric authentication first", "Always require both password and biometric authentication", "User chooses which mechanism to use", etc.

FIA_UAU.6 Re-authenticating

FIA_UAU.6.1 The TSF shall re-authenticate the human user under the conditions:

- attempted enrolment, change or deletion of any biometric authentication factor; or

- attempted change of password, PIN or pattern; or
- attempted unlocking of a locked TOE; or
- [assignment: other conditions].

FIA_UAU.7 Protected authentication feedback

FIA_UAU.7.1 The TSF shall provide only feedback that provides at most:

- information about the length of the password or PIN; and
- each password/PIN character for a brief moment as it is entered by the human user; and
- the pattern for a brief time as it is entered by the human user;

to the human user while the authentication is in progress.

NOTE 4: FIA_UAU.7 does not apply to biometric authentication.

FIA_SOS.1 Verification of secrets

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet:

- for passwords and PINs: length 4 or more.
- for patterns: consists of at least 4 and at most 9 connected points, where each point shall only be used once

NOTE 5: The human user takes care of the complexity of the PIN/Password: how easy it is to guess for others. See OE.PASSWORD_PIN.

NOTE 6: The TOE can provide an indication of the complexity of the password/PIN to assist human user in selecting passwords/PINS that are not easy to guess.

FIA_SOS.2 TSF Generation of secrets

If the assignment in FIA_UAU.5.1 is not none, FIA_SOS.2.1 and FIA_SOS.2.2 apply:

FIA_SOS.2.1 The TSF shall provide a mechanism to generate biometric templates that meet:

- A one-attempt False Acceptance Rate (FAR) that shall not exceed:
  - 1:50.000 for 2D Facial
  - 1:100.000 for 3D Facial
  - 1:50.000 for Fingerprint
  - [assignment: other value smaller or equal than 1:1000] for other method(s), and
- A one-attempt False Rejection Rate (FRR) that shall not exceed:
  - 1:15 for 2D Facial
  - 1:33 for 3D Facial
  - 1:33 for Fingerprint
  - [assignment: other value smaller or equal than 1:20] for other method(s)

FIA_SOS.2.2 The TSF shall be able to enforce the use of TSF generated biometric templates for biometric authentication.

NOTE 7: Guidance on biometric testing can be found in [i.4].

FIA_AFL.1_Password/PIN/Pattern Authentication failure handling

FIA_AFL.1.1/Password/PIN/Pattern The TSF shall detect when [assignment: configurable integer between 3 and 10] unsuccessful authentication attempts occur related to [selection: Password, PIN, Pattern].

FIA_AFL.1.2/Password/PIN/Pattern When the defined number of unsuccessful authentication attempts has been met, the TSF shall [selection: reboot, progressively lengthen the time to attempt an authentication, make all user data assets unreadable, [assignment: list of other action(s) to reduce the risk of attacks such as brute-force attack]].

FIA_AFL.1_Other Authentication failure handling

If the assignment in FIA_UAU.5.1 is not none:

FIA_AFL.1.1/Other The TSF shall detect when [assignment: configurable integer between 3 and 10] unsuccessful authentication attempts occur related to [selection: fingerprint, iris, face, voice, vein, [assignment: other authentication mechanism]].

FIA_AFL.1.2/Other When the defined number of unsuccessful authentication attempts has been met, the TSF shall [selection: reboot, progressively lengthen the time to attempt an authentication, make all user data assets unreadable, disable this form of authentication until successful authentication by [assignment: other method(s) supported by the TOE] has been performed, [assignment: list of other action(s) to reduce the risk of biometric authentication attack]].

For each authentication mechanisms specified by the ST author under FIA_UAU.5, one iteration of FIA_AFL.1 shall be included in the ST. For Password/PIN FIA_AFL.1_Password/PIN/Pattern shall be used, for all others FIA_AFL.1_Other shall be used.

8.1.4 Security Management (FMT)

FMT_SMF.1_Authentication Specification of Management Functions

FMT_SMF.1.1/Authentication The TSF shall be capable of performing the following management functions by the human user:

- register the initial password/PIN/pattern; and
- change the password/PIN/pattern; and
- register the initial [selection: fingerprint, iris, face, voice, vein, [assignment: other authentication mechanisms]; and
- change the [selection: fingerprint, iris, face, voice, vein, [assignment: other authentication mechanisms]

FMT_SMF.1_Permissions Specification of Management Functions

FMT_SMF.1.1/Permissions The TSF shall be capable of performing the following management functions by the human user:

- view permissions granted to an App; and
- revoke permission from a Downloaded App or Pre-installed non-system permission App to have read and/or write access to an object; and
- revoke permission from a Pre-installed system permission App to have read and/or write access to an object if the human user has granted this permission themselves; and
- grant permission to an App to have read and/or write access to an object; and
- select [selection: charge only mode by default, file transfer mode, [assignment: other wired charging mode]] when the TOE is connected via charging interface such as USB interface to a computer/laptop; and
- grant/revoke permission to/from a Downloaded App or Pre-installed non-system permission App to have access to accessibility service; and
- grant/revoke permission to/from a Downloaded App or Pre-installed non-system permission App to have access to device notification; and

- [assignment: list of management functions provided by the TSF].

NOTE 1: Object is defined in FDP_ACC.1.1/Permissions.

NOTE 2: The selection for charging mode is per connection basis, and the "charge only mode" is the default mode when the human user does not select any mode.

FMT_SMF.1_Privacy Specification of Management Functions

FMT_SMF.1.1/Privacy The TSF shall be capable of performing the following management functions by human user:

- change the alias provided to a particular App developer to a new random alias; and
- change the alias provided to Advertisers to a new random alias.

NOTE 3: These aliases are defined in clause 8.1.5.

NOTE 4: The change of alias can be explicit (for instance by providing a specific option in the app or in the TOE) or implicit (for instance rebooting the TOE automatically generates a new alias).

FMT_SMF.1_Update Specification of Management Functions

FMT_SMF.1.1/Update The TSF shall be capable of performing the following management functions by human user:

- initiate an update of the CMD software (if available); and
- display the version number of the CMD software; and
- uninstall Downloaded Apps; and
- uninstall Pre-installed non-system permission Apps which are not included in the TOE.

NOTE 5: CMD software can be the operating system of the TOE or an application.

8.1.5 Privacy (FPR)

To avoid tracking of the unique Device ID of the TOE, the TOE provides aliases of the Device ID for App developers. The aliases for App developer can be used for push services. One App developer gets the same alias for all the Apps on the TOE developed by this developer. The human user can reset this alias to prevent tracking to continue indefinitely (see FMT_SMF.1/Privacy).

The TOE also provides an alias of the Device ID for Advertisers, this alias can be used for advertising services. All Advertisers get the same alias. The human user can reset this alias to prevent this tracking to continue indefinitely (see FMT_SMF.1/Privacy).

FPR_PSE.1_Developers Pseudonymity

FPR_PSE.1.1/Developers The TSF shall ensure that App developers are unable to determine the Device ID bound to the TSF, unless the App has been permitted access to the Device ID by the human user or this permission was granted by the operating system.

FPR_PSE.1.2/Developers The TSF shall be able to provide at least one unique alias of the Device ID to each App developer.

NOTE: Each developer gets a different alias.

FPR_PSE.1_Advertisers Pseudonymity

FPR_PSE.1.1/Advertisers The TSF shall ensure that Advertisers in Apps are unable to determine the Device ID bound to the TSF, unless the App has been permitted access to the Device ID by the human user or this permission was granted by the operating system.

FPR_PSE.1.2/Advertisers The TSF shall be able to provide at least one alias of the Device ID to Advertisers.

8.1.6 Protection of the TSF (FPT)

FPT_PHP.3 Resistance to physical attack

FPT_PHP.3.1 The TSF shall resist to:

- read or modify the DUK; and
- read or modify [assignment: list of data and keys in the key hierarchies of the FCS_CKH.1 SFRs which are not encrypted themselves and whose leakage would affect the security of the key hierarchy]; and
- modify [assignment: any key(s), hashes of key(s), certificate(s) and/or other data] used to verify the integrity of the TSF in FPT_TST.1]; and
- modify [assignment: any key(s), hashes of key(s), certificate(s) and/or other data] used to verify the integrity and authenticity of updates to the TSF in FCS_COP.1_Update; and
- read or modify [assignment: list of other data and/or keys];

to the hardware based secure environment of the TSF by responding automatically such that it is impossible to read or modify this data and/or key(s).

NOTE 1: This requirement is on physical attacks only, logical attacks are addressed by the ADV_ARC and AVA_VAN assurance requirements.

NOTE 2: The phrase "responding automatically" means both:

- detection of an attack and taking appropriate protective action when detected. This is similar to an alarm system detecting an attack (tamper responsive);
- passively resisting an attacker by being very difficult to access, similar to a safe, which is very hard to open, and can even be hidden so hard to be located.

NOTE 3: Guidance on types of physical attacks and mitigations thereof can be found in [i.3].

NOTE 4: This requirement does not prescribe a particular implementation, such as a secure element, but does imply that the data is to be stored in hardware based secure environment, and that it will be difficult to read or modify the data inside this secure environment. The strongest constraint on the implementation is the chosen AVA_VAN level in the ST. At the minimum level (AVA_VAN.2), solutions like an isolated secure software environment running on the main SoC of the TOE can well be sufficient, but at AVA_VAN.3 and AVA_VAN.4, dedicated security hardware is more likely to be needed, and at AVA_VAN.5 a highly resistant secure element, either separate or integrated in the SoC is very likely to be required.

NOTE 5: The choice and severity (how much time, expertise, TOE knowledge, opportunity and equipment will be needed) of these physical tampering scenarios is again constrained by the AVA_VAN level chosen in the ST. For instance, at AVA_VAN.2 only very basic versions of these physical tampering scenarios are to be considered, and many attacks will be out of scope, while at AVA_VAN.5 the vulnerability analysis is based on the complete design information with consideration of state-of-the-art attack knowledge and methods and therefore all but the most extremely sophisticated attacks are in scope. An example of this can be found in annex B.

FPT_FLS.1 Failure with preservation of secure state

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: Failure of the Update_the_TSF_Software operation in FDP_ACF.1_Update.

NOTE 6: A secure state can be the state before the update is executed or a state for recovery as defined in FPT_RCV.2 Automated recovery.

FPT_TST.1 TSF testing

FPT_TST.1.1 The TSF shall run a suite of self-tests during initial start-up to demonstrate the correct operation of [selection: [assignment: parts of TSF], the TSF].

NOTE 7: FPT_TST.1.1 allows the ST author to specify tests for the correct functioning of security mechanisms (such as the random generator) when starting up. The ST author can choose which mechanisms are to be tested and complete FPT_TST.1.1 accordingly.

FPT_TST.1.2 The TSF shall verify the integrity during initial start-up of [selection: [assignment: parts of TSF data], TSF data].

NOTE 8: The ST author needs to specify the list of TSF data that will be verified for integrity during initial start-up.

FPT_TST.1.3 The TSF shall verify the integrity of [selection: [assignment: parts of TSF], the TSF] during initial start-up by [selection: a digital signature using an asymmetric key, a hash of an asymmetric key, [assignment: other integrity verification method]].

FPT_RCV.2 Automated recovery

FPT_RCV.2.1 When automated recovery from detection of a malevolent persistent presence by FPT_TST.1 is not possible, the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

FPT_RCV.2.2 For detection of a malevolent persistent presence by FPT_TST.1, the TSF shall ensure the return of the TOE to a secure state using automated procedures.

NOTE 9: FPT_RCV.2.2 mandates that the TOE returns to a secure state using automated procedures. This state can be one where the malevolent persistent presence is completely removed, or it can be a state where the malevolent persistent presence is not loaded, or otherwise not activated. In the case where it is not possible to automatically remove the malevolent persistent presence, FPT_RCV.2.1 allows human user to remove it manually, for example by a factory reset or download of an update.

8.1.7 Trusted Path/Channels (FTP)

A TOE usually supports many different communication channels, conforming to different standards, and within these standards, using different settings, resulting in different levels of security.

The ST author shall provide FTP_ITC SFRs for each channel that the ST author claims to be secure. At least one secure channel shall be claimed in the Security Target.

This does not preclude the TOE providing communication channels based on these standards with less secure settings to communicate with devices that are legacy or have limited secure communication capabilities. As part of the vulnerability analysis the impact these lower settings can have on the overall security of the TOE will be assessed.

The TOE should provide the ability for downloaded apps to use some of the secure channel mechanisms for their communications. In this case, the "trusted IT product" that the app communicated with will be determined by the app itself.

FTP_ITC.1 Inter-TSF trusted channel

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2 The TSF shall permit [selection: the TSF, another trusted IT product] to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for communication with the trusted IT product.

If Bluetooth is supported by the TOE, the Bluetooth channel shall as a minimum:

- conform to Bluetooth® Core Specification selection v4.1 [12], v4.2 [13], v5.0 [14], v5.1 [15], v5.2 [16] or higher version of Bluetooth Core Specification; and
- require explicit user authorisation before pairing; and
- use Secure Simple Pairing and Secure Connections for pairing; and

- not allow more than one Bluetooth connection to the same Bluetooth device address; and
- generate new ECDH public/private key pairs every [selection: 24 hours, three failed authentication attempts from any Bluetooth device address, ten successful pairings from any Bluetooth device address, [assignment: other frequency and/or criteria for new key pair generation]].

If HTTPs is supported by the TOE, the HTTPs channel shall as a minimum:

- conform to IETF RFC 2818 [6]; and
- use TLS v1.2 [7], TLS v1.3 [11], or higher version of TLS to implement HTTPs.

If TLS is supported by the TOE, the TLS channel shall as a minimum:

- implement TLS v1.2 [7], TLS v1.3 [11] or higher version of TLS; and
- support X.509v3 certificates for mutual authentication; and
- determine validity of the peer certificate by certificate path, expiration date and revocation status according to IETF RFC 5280 [8]; and
- notify the TSF and [selection: not establish the connection, request application authorization to establish the connection, no other action] if the peer certificate is deemed invalid; and
- support one of the following ciphersuites:
  - TLS_RSA_WITH_AES_256_GCM_SHA384 (IETF RFC 5288 [9])
  - TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (IETF RFC 5288 [9])
  - TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (IETF RFC 5289 [10])
  - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (in IETF RFC 5289 [10])
  - TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (in IETF RFC 5289 [10])
  - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (in IETF RFC 5289 [10])
  - TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (IETF RFC 5289 [10])
  - TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (IETF RFC 5289 [10])
  - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (IETF RFC 5289 [10])
  - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (IETF RFC 5289 [10])

If WLAN is supported by the TOE, the WLAN channel shall as a minimum:

- implement 802.11-2012 [17], 802.1X [18] and EAP-TLS [19]; and
- generate symmetric keys according to PRF 384 or PRF-704 with key length 128 bit or 256 bit; and
- use TLS v1.2 [7] or higher; and
- support X.509v3 certificates for mutual authentication; and
- determine validity of the peer certificate by certificate path, expiration date and revocation status according to IETF RFC 5280 [8]; and
- notify the TSF, and not establish the connection or request application authorization to establish the connection if the peer certificate is deemed invalid; and
- support one of the following ciphersuites:
  - TLS_RSA_WITH_AES_256_GCM_SHA384 (IETF RFC 5288 [9])
  - TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (IETF RFC 5288 [9])

- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (IETF RFC 5289 [10])
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (IETF RFC 5289 [10])
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (IETF RFC 5289 [10])
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (IETF RFC 5289 [10])
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (IETF RFC 5289 [10])
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (IETF RFC 5289 [10])
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (IETF RFC 5289 [10])
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (IETF RFC 5289 [10])

- Randomly generate a new MAC-address each time it connects to a different access point.

8.2 Security assurance requirements

The security assurance requirements consist of EAL2 augmented with ALC_FLR.3 [3], where ALC_FLR.3 is refined as below.

ALC_FLR.3 Systematic flaw remediation

ALC_FLR.3.1D The developer shall document and provide flaw remediation procedures addressed to TOE developers.

ALC_FLR.3.2D The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.

ALC_FLR.3.3D The developer shall provide flaw remediation guidance addressed to TOE users.

ALC_FLR.3.1C The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.3.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.3.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.3.4C The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users. The flaw remediation procedures documentation shall also define the planned minimum length of time after release of the TOE that these methods will be used to maintain the TOE.

ALC_FLR.3.5C The flaw remediation procedures shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.

ALC_FLR.3.6C The flaw remediation procedures shall include a procedure requiring timely response and the automatic distribution of security flaw reports and the associated corrections to registered users who might be affected by the security flaw.

ALC_FLR.3.7C The procedures for processing reported security flaws shall ensure that any reported flaws are remediated and the remediation procedures issued to TOE users.

ALC_FLR.3.8C The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.

ALC_FLR.3.9C The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.

ALC_FLR.3.10C The flaw remediation guidance shall describe a means by which TOE users can register with the developer, to be eligible to receive security flaw reports and corrections.

ALC_FLR.3.11C The flaw remediation guidance shall identify the specific points of contact for all reports and enquiries about security issues involving the TOE.

ALC_FLR.3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

NOTE: ISO/IEC JTC 1 SC27 WG3 is currently in the process of creating a TR for patch management, but this is not stable yet. Once this is ready, it is suggested to revise the present document to take this TR into account.

8.3 Security requirements rationale

8.3.1 Rationale for choosing the SARs

EAL2 is chosen as it provides a good balance between effort expected from the developer and assurance gained.

EAL2 provides assurance by a full security target and an analysis of the SFRs in that ST, using a functional and interface specification, guidance documentation and a basic description of the architecture of the TOE, to understand the security behaviour.

NOTE: Requirements for functional and interface specification, guidance documentation and description of the architecture of the TOE are defined in [3].

The analysis is supported by independent testing of the TSF, evidence of developer testing based on the functional specification, selective independent confirmation of the developer test results, and a vulnerability analysis (based upon the functional specification, TOE design, security architecture description and guidance evidence provided) demonstrating resistance to penetration attackers with a basic attack potential.

EAL2 also provides assurance through use of a configuration management system and evidence of secure delivery procedures.

EAL2 is augmented by ALC_FLR.3 because CMDs are complex devices that are often subject to many hacking attempts and security investigations, that can result in the discovery of security flaws. ALC_FLR.3 provides assurance that the TOE will be maintained and supported in the future, and require the TOE developer to have procedures to:

- accept suspected security flaws in the TOE from third parties and from their own internal processes; and
- process these suspected flaws to determine whether they are actual security flaws; and
- correct the actual security flaws; and
- distribute these corrections to TOEs in the field automatically in a timely fashion.

An example is that the present document requires resistance against physical attacks through FPT_PHP.3 in such a way that it allows different implementations and is therefore fairly abstract. If a developer uses a certified secure element (where the implementation is known), it is likely that the requirements in the ST of that secure element are much more detailed and therefore different, but still be more than sufficient to meet the FPT_PHP.3 requirement of the present document. If strict or exact conformance was specified, the secure element would have to be recertified against a ST containing a suitably completed copy of the FPT_PHP.3 in the present document, which would cause a lot of unnecessary work.

8.3.2 The SFRs meet all the security objectives for the TOE

Security Objective	Rationale
O.PROTECT_COMMS	This objective is achieved by FTP_ITC.1 which sets up a secure channel with authentication and protection from modification and disclosure.
O.UPDATES **O.AUTHENTICATED_UPDATES**	These objectives are achieved by: • FDP_ACC.1_Update and FDP_ACF.1_Update specifying the policy for updating. • FCS_COP.1_Update specifying the cryptographic mechanism for checking the validity if an update. • FMT_SMF.1_Update, specifying that human user can initiate an update. • FPT_FLS.1, specifying that failure to correctly update will not lead to an insecure state.
O.PROTECT_ASSETS_AT_REST **O.DATA_CLASSIFICATION**	These objectives are achieved by: • FDP_ACC.1_User_Asset_Data_Encryption and FDP_ACF.1_User_Data_Encryption showing the three classes of user data assets, and when each class can be decrypted. • FCS_COP.1_User_Data_Assets, specifying how they are encrypted and decrypted. • FSC_CKH.1_Low, FCS_CKH.1_Medium/High describing how the cryptographic keys are derived and protected.
O.SECURE_WIPE	This objective is achieved by FCS_CKM.4 specifying that keys from the key hierarchy for each class of data can be deleted on request of the human user, making the data unreadable.
O.CRITICAL_STORAGE	This objective is achieved by FPT_PHP.3 which directly implements the objective.
O.ACCESS_CONTROL	This objective is achieved by FDP_ACC.1_Permissions, FDP_ACF.1_Permissions, specifying that user permission is needed and FMT_SMF.1_Permissions allowing human users to provide and revoke such permission. This objective is supported by FPR_PSE.1_Developers and FPT_PSE.1_Advertisers allowing App developers and Advertisers the ability to track TOEs, but denying their Apps access to a permanent ID of the TOE, and providing an alias instead. FMT_SMF_Privacy allows human users to reset the alias.
O.SECURE_BOOT **O.PERSISTENT**	These objectives are achieved by FPT_TST.1, testing the integrity of the TSF, and FPT_RCV.2 specifying the actions to be undertaken (either automatic or by the human user) when a malevolent persistent presence is found.
O.AUTHENTICATE_USER	This objective is achieved by: • FMT_SMF.1_Authentication specifying that human users can register their authentication data and change this later. • FIA_UAU.1 specifying that each human user can only perform limited actions before being authenticated and is authenticated to gain full access • FIA_UAU.6 listing the conditions under which a human user is to be re-authenticated • FIA_UAU.5 listing the multiple authentication mechanism a TOE has, and the rules for using these. • FIA_SOS.1 and FIA_SOS.2 listing the minimum quality requirements for authentication (password/PIN length, biometric quality) • FIA_AFL.1_Password/PIN/Pattern and FIA_AFL.1_Other specifying what happens when authentication fails repeatedly for each mechanism • FIA_UAU.7 specifying that passwords and PINs are not displayed on the screen when entering them, preventing shoulder surfing.
O.CRYPTOGRAPHY	This objective is achieved by the requirements in the FCS chapter. These requirements are also required by various other security objectives (see the other entries in this table).
O.RANDOMS	This objective is achieved by FCS_RNG.1 defining a random generator, whose output meets stringent international standards.
O.AUTHENTICATE_PEER_DEVICE	This objective is achieved by FIA_UAU.2 specifying how trusted peer devices are authenticated and what they can do after authentication.

Security Objective	Rationale
O.SELF_PROTECTION	This objective is achieved by: • FPT_TST.1 specifying that the TSF runs self-testing and integrity verification of the TSF or part of the TSF to protect the TSF to be tampered, and • FPT_PHP.3.1 to prevent modification of key(s), hashes of key(s), certificate(s) and/or other data used to verify the integrity of the TSF.
O.SEPARATION	This objective is achieved by: • FDP_ACC.1_Permissions and FDP_ACF.1_Permissions defining security attribute based access control for apps, and • FMT_SMF.1_Permissions specifying management functions for apps.

NOTE 1: O.SELF_PROTECTION is further achieved by ADV_ARC.1.4C, which requires the security architecture description to demonstrate that the TSF protects itself from tampering.

NOTE 2: O.SEPARATION is further achieved by ADV_ARC.1.2C, which requires the security architecture description to describe the security domains maintained by the TSF consistently with the SFRs.

8.3.3 Dependency analysis

SFR	Dependency	Rationale
FCS_RNG.1	-	
FCS_COP.1_Update	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 FCS_CKM.4	 Not fulfilled, as the key for checking update packages is inserted in the TOE during production Not fulfilled, as the key for checking update packages is never destroyed
FCS_CKH.1_Low	FCS_COP.1 FCS_CKM.4	Fulfilled by FCS_COP.1_User_Data_Assets Fulfilled by FCS_CKM.4
FCS_CKH.1_Medium/High	FCS_COP.1 FCS_CKM.4	Fulfilled by FCS_COP.1_User_Data_Assets Fulfilled by FCS_CKM.4
FCS_CKM.4	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	Fulfilled by FCS_CKH.1. This replaces FCS_CKM.1 as it generates a key hierarchy rather than an individual key like FCS_CKM.1
FDP_ACC.1_Update	FDP_ACF.1	Fulfilled by FDP_ACF.1_Update
FDP_ACF.1_Update	FDP_ACC.1 FMT_MSA.3	Fulfilled by FDP_ACC.1_Update Not fulfilled, as the only security attribute is the version of the TSF and this is inserted during production and the rules for modifying it (copying the version number of the Update Package is already described in the SFR)
FDP_ACC.1_Permissions	FDP_ACF.1	Fulfilled by FDP_ACF.1_Permissions
FDP_ACF.1_Permissions	FDP_ACC.1 FMT_MSA.3	Fulfilled by FDP_ACC.1_Permissions Not necessary, as there are no security attributes in the Permissions Policy
FDP_ACC.1_User_Data_ Asset_Encryption	FDP_ACF.1	Fulfilled by FDP_ACF.1_User_Data_Asset_Encryption
FDP_ACF.1_User_Data_Asset Encryption	FDP_ACC.1 FMT_MSA.3	Fulfilled by FDP_ACC.1_User_Data_Asset_Encryption Not necessary, as the security attributes Low, Medium and High do not change
FIA_UAU.1	FIA_UID.1	Not fulfilled, as the person authenticating is automatically identified as the owner of the TOE
FIA_UAU.2	FIA_UID.1	Not fulfilled, as the human user will actively select the trusted peer device to be authenticated
FIA_UAU.5	-	
FIA_UAU.6	-	
FIA_UAU.7	FIA_UAU.1	Fulfilled by FIA_UAU.1
FIA_SOS.1	-	
FIA_SOS.2	-	
FIA_AFL.1_Password/PIN/ Pattern	FIA_UAU.1	Fulfilled by FIA_UAU.1
FIA_AFL.1_Other	FIA_UAU.1	Fulfilled by FIA_UAU.1
FMT_SMF.1_Authentication	-	
FMT_SMF.1_Permission	-	
FMT_SMF.1_Privacy	-	
FMT_SMF.1_Update		
FPR_PSE.1_Developers	-	
FPR_PSE.1_Advertisers	-	
FPT_FLS.1	-	
FPT_PHP.3	-	
FPT_TST.1	-	
FPT_RCV.2	AGD_OPE.1	Fulfilled by EAL2
FTP_ITC.1	-	
EAL2	Many	All dependencies in an EAL are met
ALC_FLR3	-	-

Annex A (informative): Other related specifications

A.1 ETSI EN 303 645

The table below shows the correspondence between clause 5 (Cyber Security Requirements) and clause 6 (Data Protection Provisions) of ETSI EN 303 645 [i.1] and the present document. If a device meets SFRs of the present document, it can also meet corresponding provisions in ETSI EN 303 645 [i.1]. Note that the present document is defined as a PP for consumer mobile devices, and products claiming conformance to the PP will be evaluated by an accredited security test lab for CC evaluation, while ETSI EN 303 645 [i.1] is defined as baseline security requirements for consumer IoT devices, which can be used for conformance testing. These two documents are defined for different purpose of use, therefore the table below indicates the approximate relation, and details can differ.

Table A.1-1

ETSI EN 303 645	The present document	Comments
Provision 5.1-1 *Where passwords are used and in any state other than the factory default, all consumer IoT device passwords shall be unique per device or defined by the user*	FIA_SOS.1 and OE.PASSWORD_PIN	Passwords are human user generated
Provision 5.1-2 *Where pre-installed unique per device passwords are used, these shall be generated with a mechanism that reduces the risk of automated attacks against a class or type of device.*	-	Not relevant, as passwords are not pre-installed
Provision 5.1-3 *Authentication mechanisms used to authenticate users against a device shall use best practice cryptography, appropriate to the properties of the technology, risk and usage.*	The SOG-IS spec [4] or local equivalent is mandated in the present document.	
Provision 5.1-4 *Where a user can authenticate against a device, the device shall provide to the user or an administrator a simple mechanism to change the authentication value used.*	FMT_SMF.1_Authentication	
Provision 5.1-5 *When the device is not a constrained device, it shall have a mechanism available which makes bruteforce attacks on authentication mechanisms via network interfaces impracticable.*	-	User authenticates locally to the CMD and not via network interfaces
Provision 5.2-1 *The manufacturer shall make a vulnerability disclosure policy publicly available. This policy shall include, at a minimum:* *- contact information for the reporting of issues; and* *information on timelines for:* *1) initial acknowledgement of receipt; and* *2) status updates until the resolution of the reported issues.*	ALC_FLR.3	
Provision 5.2-2 *Disclosed vulnerabilities should be acted on in a timely manner.*	ALC_FLR.3	

ETSI EN 303 645	The present document	Comments
Provision 5.2-3 *Manufacturers should continually monitor for, identify and rectify security vulnerabilities within products and services they sell, produce, have produced and services they operate during the defined support period.*	ALC_LCD.1, ALC_FLR.3	
Provision 5.3-1 *All software components in consumer IoT devices should be securely updateable.*	FDP_ACF.1_Update	This only applies to the OS and the OEM apps. All other apps are outside the scope of this requirement
Provision 5.3-2 *When the device is not a constrained device, it shall have an update mechanism for the secure installation of updates.*	FDP_ACF.1_Update	
Provision 5.3-3 *An update shall be simple for the user to apply.*	OE.USER	
Provision 5.3-4 *Automatic mechanisms should be used for software updates.*	FDP_ACF.1_Update	CMDs usually only update after explicit user permission
Provision 5.3-5 *The device should check after initialization, and then periodically, whether security updates are available.*	FDP_ACF.1_Update	
Provision 5.3-6 *If the device supports automatic updates and/or update notifications, these should be enabled in the initialized state and configurable so that the user can enable, disable, or postpone installation of security updates and/or update notifications.*	FMT_SMF.1_Update	This only allows the human user to determine to initiate an update, not to disable it
Provision 5.3-7 *The device shall use best practice cryptography to facilitate secure update mechanisms.*	FCS_COP.1_Update	
Provision 5.3-8 *Security updates shall be timely.*	ALC_FLR.3	
Provision 5.3-9 *The device should verify the authenticity and integrity of software updates.*	FDP_ACF.1_Update	
Provision 5.3-10 *Where updates are delivered over a network interface, the device shall verify the authenticity and integrity of each update via a trust relationship.*	FDP_ACF.1_Update	
Provision 5.3-11 *The manufacturer should inform the user in a recognizable and apparent manner that a security update is required together with information on the risks mitigated by that update.*	FDP_ACF.1_Update	This does not include informing the human user of the risks mitigated by the update
Provision 5.3-12 *The device should notify the user when the application of a software update will disrupt the basic functioning of the device.*	-	Not a security requirement
Provision 5.3-13 *The manufacturer shall publish, in an accessible way that is clear and transparent to the user, the defined support period.*	-	Not in scope of present document

ETSI EN 303 645	The present document	Comments
Provision 5.3-14 *For constrained devices that cannot have their software updated, the rationale for the absence of software updates, the period and method of hardware replacement support and a defined support period should be published by the manufacturer in an accessible way that is clear and transparent to the user.*	-	CMDs are not constrained devices
Provision 5.3-15 *For constrained devices that cannot have their software updated, the product should be isolable and the hardware replaceable.*	-	CMDs are not constrained devices
Provision 5.3-16 *The model designation of the consumer IoT device shall be clearly recognizable, either by labelling on the device or via a physical interface.*	ALC_CMC.1	
Provision 5.4-1 *Sensitive security parameters in persistent storage shall be stored securely by the device.*	FCS_CKH.1_Low, FCS_CKH.1 _Medium, FPT_PHP.3	
Provision 5.4-2 *Where a hard-coded unique per device identity is used in a device for security purposes, it shall be implemented in such a way that it resists tampering by means such as physical, electrical or software.*	FPT_PHP.3	
Provision 5.4-3 *Hard-coded critical security parameters in device software source code shall not be used.*	-	Source code inspection is outside EAL2
Provision 5.4-4 *Any critical security parameters used for integrity and authenticity checks of software updates and for protection of communication with associated services in device software shall be unique per device and shall be produced with a mechanism that reduces the risk of automated attacks against classes of devices.*	FCS_RNG.1, FCS_COP.1_Update, FTP_ITC.1,	
Provision 5.5-1 *The consumer IoT device shall use best practice cryptography to communicate securely.*	The first note in the Cryptographic Support (FCS) Section	
Provision 5.5-2 *The consumer IoT device should use reviewed or evaluated implementations to deliver network and security functionalities, particularly in the field of cryptography.*	The SOG-IS spec [4] or local equivalent is mandated in the present document.	
Provision 5.5-3 *Cryptographic algorithms and primitives should be updateable.*	FDP_ACF.1_Update	The TSF is updateable and this encompasses cryptographic algorithms.
Provision 5.5-4 *Access to device functionality via a network interface in the initialized state should only be possible after authentication on that interface.*	FTP_ITC.1	

ETSI EN 303 645	The present document	Comments
Provision 5.5-5 *Device functionality that allows security-relevant changes in configuration via a network interface shall only be accessible after authentication. The exception is for network service protocols that are relied upon by the device and where the manufacturer cannot guarantee what configuration will be required for the device to operate.*	FTP_ITC.1	
Provision 5.5-6 *Critical security parameters should be encrypted in transit, with such encryption appropriate to the properties of the technology, risk and usage.*	FTP_ITC.1	
Provision 5.5-7 *The consumer IoT device shall protect the confidentiality of critical security parameters that are communicated via remotely accessible network interfaces.*	FTP_ITC.1	
Provision 5.5-8 *The manufacturer shall follow secure management processes for critical security parameters that relate to the device.*		Physical and logical security of the manufacturer (ALC_DVS.1) is outside EAL2
Provision 5.6-1 *All unused network and logical interfaces shall be disabled.*	AVA_VAN.2 (see detailed explanation after the table)	
Provision 5.6-2 *In the initialized state, the network interfaces of the device shall minimize the unauthenticated disclosure of security-relevant information.*	AVA_VAN.2 (see detailed explanation after the table)	
Provision 5.6-3 *Device hardware should not unnecessarily expose physical interfaces to attack.*	AVA_VAN.2 (see detailed explanation after the table)	
Provision 5.6-4 *Where a debug interface is physically accessible, it shall be disabled in software.*	AVA_VAN.2 (see detailed explanation after the table)	
Provision 5.6-5 *The manufacturer should only enable software services that are used or required for the intended use or operation of the device.*	AVA_VAN.2 (see detailed explanation after the table)	
Provision 5.6-6 *Code should be minimized to the functionality necessary for the service/device to operate.*	-	Source code inspection is outside EAL 2
Provision 5.6-7 *Software should run with least necessary privileges, taking account of both security and functionality.*	FDP_ACF.1_Permissions	
Provision 5.6-8 *The device should include a hardware-level access control mechanism for memory.*	ADV_ARC.1	
Provision 5.6-9 *The manufacturer should follow secure development processes for software deployed on the device.*	EAL2	
Provision 5.7-1 *The consumer IoT device should verify its software using secure boot mechanisms.*	FPT_TST.1	

ETSI EN 303 645	The present document	Comments
Provision 5.7-2 *If an unauthorized change is detected to the software, the device should alert the user and/or administrator to the issue and should not connect to wider networks than those necessary to perform the alerting function.*	FPT_TST.1, FPT_RCV.2	
Provision 5.8-1 *The confidentiality of personal data transiting between a device and a service, especially associated services, should be protected, with best practice cryptography.*	FTP_ITC.1	
Provision 5.8-2 *The confidentiality of sensitive personal data communicated between the device and associated services shall be protected, with cryptography appropriate to the properties of the technology and usage.*	FTP_ITC.1	
Provision 5.8-3 *All external sensing capabilities of the device shall be documented in an accessible way that is clear and transparent for the user.*	AGD_OPE.1, OE.USER	
Provision 5.9-1 *Resilience should be built in to consumer IoT devices and services, taking into account the possibility of outages of data networks and power.*	AVA_VAN.2 (see detailed explanation after the table)	
Provision 5.9-2 *Consumer IoT devices should remain operating and locally functional in the case of a loss of network access and should recover cleanly in the case of restoration of a loss of power.*	AVA_VAN.2 (see detailed explanation after the table)	
Provision 5.9-3 *The consumer IoT device should connect to networks in an expected, operational and stable state and in an orderly fashion, taking the capability of the infrastructure into consideration.*		Not applicable to CMD
Provision 5.10-1 *If telemetry data is collected from consumer IoT devices and services, such as usage and measurement data, it should be examined for security anomalies.*	-	A requirement for the provider, App developer, Advertiser etc. Not for the CMD itself
Provision 5.11-1 *The user shall be provided with functionality such that user data can be erased from the device in a simple manner.*	FCS_CKM.4	
Provision 5.11-2 *The consumer should be provided with functionality on the device such that personal data can be removed from associated services in a simple manner.*		A requirement for the service provider, not for the CMD itself
Provision 5.11-3 *Users should be given clear instructions on how to delete their personal data.*	OE.USER	
Provision 5.11-4 *Users should be provided with clear confirmation that personal data has been deleted from services, devices and applications.*	-	Usability requirement, not a security requirement
Provision 5.12-2 *The manufacturer should provide users with guidance on how to securely set up their device.*	AGD_PRE.1	

ETSI EN 303 645	The present document	Comments
Provision 5.12-3 *The manufacturer should provide users with guidance on how to check whether their device is securely set up.*	AGD_PRE.1	
Provision 5.13-1 *The consumer IoT device software shall validate data input via user interfaces or transferred via Application Programming Interfaces (APIs) or between networks in services and devices.*	AVA_VAN.2 (see detailed explanation after the table)	
Provision 6-1 *The manufacturer shall provide consumers with clear and transparent information about what personal data is processed, how it is being used, by whom, and for what purposes, for each device and service. This also applies to third parties that can be involved, including advertisers.*	-	A requirement for the provider, App developer, advertiser etc. Not for the CMD itself
Provision 6-2 *Where personal data is processed on the basis of consumers' consent, this consent shall be obtained in a valid way.*	FDP_ACF.1_Permissions	
Provision 6-3 *Consumers who gave consent for the processing of their personal data shall have the capability to withdraw it at any time.*	FMT_SMF.1_Permissions	
Provision 6-4 *If telemetry data is collected from consumer IoT devices and services, the processing of personal data should be kept to the minimum necessary for the intended functionality.*		A requirement for the provider, App developer, Advertiser etc. Not for the CMD itself
Provision 6-5 *If telemetry data is collected from consumer IoT devices and services, consumers shall be provided with information on what telemetry data is collected, how it is being used, by whom, and for what purposes.*		A requirement for the provider, App developer, Advertiser etc. Not for the CMD itself

This PP requires conformance to the CC package EAL2, and EAL2 requires AVA_VAN.2 vulnerability analysis as minimum. In order to perform an independent vulnerability analysis, the TOE developer is required to provide guidance documentation (AGD_OPE.1), functional specification (ADV_FSP.2), TOE design (ADV_TDS.1), security architecture description (ADV_ARC.1) and preparative procedures (AGD_PRE.1) as defined in [3], and the evaluator will review these documents, identify potential vulnerabilities, conduct penetration testing to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

Regarding Provision 5.6-1 to Provision 5.6-5 and Provision 5.13-1 in ETSI EN 303 645 [i.1] which are related to interfaces of the device, ADV_FSP.2 requires the TOE developer to provide the purpose, method of use, parameters, and parameter descriptions for all interfaces of the TSF, the evaluator will evaluate whether those interfaces expose risks for Basic attack potential.

Regarding Provision 5.9-1 and Provision 5.9-2, these are requirements to avoid outages of data networks and power causing impact on the human user and to design products and services that provide a level of resilience to such case. The evaluator may review the documents provided by the TOE developer to check whether appropriate implementation is in place. However, these are not typical security requirements from CC perspective.

A.2 SESIP

The table below shows the correspondence between SESIP [i.5] and the present document. The purpose of this mapping is to provide references for how the SFRs of the present document, which are currently expressed in the language defined in [3], can be expressed in a more readable form in the language defined in [i.5].

Table A.2-1

The present Document	SESIP	Comments
FCS_RNG.1	Cryptographic Random Number Generation	
FCS_COP.1_Update FCS_CKH.1_Low FCS_CKH.1_Medium/High	Cryptographic Operation Cryptographic Key Generation Cryptographic Key Store	
FCS_CKM.4	Factory Reset of Platform Decommission of Platform Field Return of Platform Secure Uninstall of Application	
FDP_ACC.1_Update FDP_ACF.1_Update FPT_FLS.1	Secure Update of Platform Secure Update of Application	
FDP_ACC.1_Permissions FDP_ACF.1_Permissions FMT_SMF.1_Permission	Software Attacker Resistance: Isolation of Platform Software Attacker Resistance: Isolation of Platform Parts	
FDP_ACC.1_User_Data_Asset Encryption FDP_ACF.1_User_Data_Asset Encryption	Secure Cryptographic Storage	
FIA_*		No equivalent, as authentication is outside the scope of SESIP (IoT platforms).
FMT_SMF.1_Privacy FPR_*		No equivalent, as privacy is outside the scope of SESIP (IoT platforms).
FPT_PHP.3	Physical Attacker Resistance	
FPT_TST.1 FPT_RCV.2	Secure Initialization of Platform	
FTP_ITC.1	Secure Communication Support Secure Communication Enforcement	
EAL2	SESIP Level 2	SESIP level 2 is a stripped version of EAL2, providing (in the opinion of the SESIP authors) a similar level of assurance, but at significant less cost.
ALC_FLR.3	ALC_FLR.2	SESIP level 2 requires ALC_FLR.2 instead of 3.

Annex B (informative): Rating of a physical attack

An example of an attack scenario is an electromagnetic side-channel attack on the TSF part that performs memory encryption to discover the key that is used for memory encryption.

Table B-1

Factor and value	Rationale	Points
Expertise = Expert	A side-channel specialist is needed to mount a simple power/electro-magnetic analysis attack or differential power/electro-magnetic analysis attack. A second, proficient level person, with knowledge on the protocols used inside the TOE is not counted additionally, in accordance with the footnote in [5]. The same applies for a third, proficient level person preparing the hardware part and mounting a coil on the part for measuring electromagnetic emanation.	6
Knowledge of the TOE = Restricted	Information on the exact key management methods and key derivation techniques is needed. Detailed sensitive design information is probably not needed.	3
Window of opportunity = Moderate	First, one needs to get hold of a TOE with a known key. Secondly, one needs to obtain (steal) the target TOE of a particular target user.	4
Equipment = Specialized	For advanced side-channel attacks usually high-end oscilloscopes with sampling rates in the Gigahertz range are used. In the particular case of an attack on a CMD, it can be sufficient to use a cheaper oscilloscope with sampling rates in the Megahertz range. For this reason, Specialized is chosen and not bespoke.	4
Elapsed time	The attacker starts out by using a TOE with a known key, so that the attacker knows when its attack is successful (when the known key is found) and find the optimal parameters. This is the most time-consuming part of the attack. This assumes that a TOE can be configured to use a known key, which may well not be the case. The time needed for the second stage, where one applies these parameters on the target TOE to discover the key takes negligible time compared to the first stage.	T
Overall rating	Sum of all points.	17+T

In the very unlikely event that both stages of the attack can be done in less than a day (so T = 0), the rating of 17 is sufficient to pass AVA_VAN.1, AVA_VAN.2, and AVA_VAN.3, but would fail for AVA_VAN.4 and AVA.VAN.5.

To pass AVA_VAN.5, the TOE would have to withstand the above attack for more than 2 months (T = 10).

Annex C (informative): Mapping of threats with interfaces of the TOE

This annex shows each interface defined in clause 5.1 on what nefarious actions each of the threat agents defined in clause 5.2 could perform on that interface.

Table C-1

Threats	Local wireless interface(s)	Wide-area network interface(s)	User interface	Physical interface	Application interface
T.EAVESDROP	X	X			
T.SPOOF	X	X			
T.MODIFY-COMMS	X	X			
T.COUNTERFEIT_DEVICE	X				
T.IMPERSONATE			X		
T.PHYSICAL				X	
T.RECOVER_DATA			X	X	
T.MODIFY_DEVICE			X	X	X
T.FLAWAPP-ACCESS					X
T.PERSISTENT	X	X	X	X	X
T.NEW_ATTACKS	X	X	X	X	X
T.FLAWAPP_HACKS_TOE					X
T.FLAWAPP_HACKS_OTHER_APPS					X

History

Document history		
V1.1.1	November 2021	Publication