ETSI TS 103 732-1 V2.1.2 (2023-11)

CYBER;
Consumer Mobile Device;
Part 1: Base Protection Profile

Reference
RTS/CYBER-00123-1

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).

The present document is part 1 of a multi-part deliverable covering the Consumer Mobile Device, as identified below:

Part 1: "Base Protection Profile";

Part 2: "Biometric Authentication Protection Profile Module";

Part 3: "Multi-user Protection Profile Module";

Part 4: "Consumer Module Devices - Preloaded Apps PP-Module".

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. Smartphones and tablets are typical consumer 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.

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-customizable device utilizing 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 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.5];
- features related to multiple authenticated 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] CCMB-2017-04-001 Version 3.1 revision 5, April 2017: "Common Criteria for Information Technology Security Evaluation - Part 1: Introduction and general model".

[2] CCMB-2017-04-002 Version 3.1 revision 5, April 2017: "Common Criteria for Information Technology Security Evaluation - Part 2: Security functional components".

[3] CCMB-2017-04-003 Version 3.1 revision 5, April 2017: "Common Criteria for Information Technology Security Evaluation - Part 3: Security assurance components".

[4] CCMB-2017-04-004 Version 3.1 revision 5, April 2017: "Common Methodology for Information Technology Security Evaluation - Evaluation methodology".

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

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

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

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

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

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

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

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

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

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

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

[16] 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".

[17] ISO/IEC 18033-1:2021: "Information security - Encryption algorithms - Part 1: General".

[18] CCDB-2017-05-xxx Version 0.5, May 2017: "CC and CEM addenda, Exact Conformance, Selection-Based SFRs, Optional SFRs".

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] Global Platform: "Security Evaluation Standard for IoT Platforms", version 1.0, March 2020.

[i.5] ETSI TS 102 221: "Smart Cards; UICC-Terminal interface; Physical and logical characteristics (Release 16)".

[i.6] GSMA SGP.22: "RSP Technical Specification".

[i.7] ETSI TS 133 102: "Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; 5G; 3G security; Security architecture (3GPP TS 33.102)".

[i.8] ETSI TS 133 401: "Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; 3GPP System Architecture Evolution (SAE); Security architecture (3GPP TS 33.401)".

[i.9] ETSI TS 133 501: "5G; Security architecture and procedures for 5G System (3GPP TS 33.501)".

[i.10] BSI-CC-PP-0084-2014: "Security IC Platform Protection Profile with Augmentation Packages" version 1.0, February 2014.

[i.11] GSMA SGP.25: "Embedded UICC for Consumer Devices Protection Profile".

[i.12] GSMA SGP.08: "Security Evaluation of Integrated eUICC".

[i.13] GSMA SGP.24: "GSMA SGP.24: "RSP Compliance Process".

[i.14] "SOG-IS Crypto Evaluation Scheme Agreed Cryptographic Mechanisms", SOG-IS Crypto Working Group, version 1.2, January 2020.

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

ad network: technology platform used to match the sale of ads to be displayed to users between publishers and advertisers

atomic update: update process in which the result will either be a completely successful update or the update process will be completely abandoned or reversed, returning to the state before the update process was started

authentication factor: general term that can stand for any input used to verify a user as an authentication credential. There are three types of possible factors:

- **biometric authentication factor:** use of a biometric sample matched to template generated by an enrolment process (and possibly updated during successful authentication attempts);
- **external authentication factor:** use of a separate item in possession of the user to provide authentication such as a security key;
- **known authentication factor:** use of something the user knows, a password, PIN or pattern for authentication.

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 utilizing 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

critical security parameters: parameters, such as stored user credentials (or keys used to encrypt them), keys used to encrypt assets, securely boot the device, or authenticate updates which need to be stored securely

derivation: cryptographic method for deriving keys from input values such as passwords that increase the computational cost of guessing the input values (such as in a brute force attack)

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

EXAMPLE: Typical examples of a device ID are the IMEI and SN of a device.

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

embedded UICC: UICC which is not easily accessible or replaceable, is not intended to be removed or replaced in the terminal, and enables the secure changing of subscriptions

lock screen: screen that is displayed when the device is locked and requires credentials to be entered to access the primary functionality of the TOE

lock screen(boot): screen that is displayed when the device is locked after the device has been (re)started, prior to any user successfully entering any credentials

main OS: primary operating system of the device (as opposed to subsystems that may provide specialized, usually security-related, functions)

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

security functional requirement: requirement, stated in a standardized language, which is meant to contribute to achieving the security objectives for a TOE

NOTE: As defined in [1].

security objective: statement of an intent to counter identified threats and/or satisfy identified organization security policies and/or assumptions

NOTE: As defined in [1].

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

NOTE: As defined in [1].

separate execution environment: operating environment separate from the main OS with highly restricted access used to provide secure isolation for sensitive operations

(internal) storage: storage which is not intended to be removed from the device (generally this would be integrated into the main board of the device)

(removable) storage: storage which is intended by design to be added to and removed from the device over an available connector (such as a microSD slot or an external port that can be used to mount media)

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

system software: main OS, bootloaders, any preloaded apps and other components necessary for the device to function as expected

system software update: update for the system software, including firmware updates for any updateable component on the device itself (but does not update user data or downloaded apps)

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

NOTE: As defined in [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

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.

trustworthy update source: central repository from which updates to the system software can be downloaded, typically managed by the TOE manufacturer or the network carrier

UICC: *smart card* that conforms to the specifications written and maintained by the ETSI Secure Element Technologies Technical Body

NOTE: UICC is neither an abbreviation nor an acronym.

user: "human user" of the TOE, someone in physical presence of the TOE, as opposed to another system or service for which actions may be taken

version ID: identifier used to determine the difference between two releases of software

3.2 Symbols

Void.

3.3 Abbreviations

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

3GPP	3G (mobile) Partnership Project
ADP	Application Distribution Platform
AES	Advanced Encryption Standard
API	Application Program Interface
App	Application
CC	Common Criteria
CEM	Common Evaluation Methodology
CMD	Consumer Mobile Device
DEK	Data Encryption Key
DUK	Device Unique Key
EAF	External Authentication Factor
EAL	Evaluation Assurance Level
ECD	Extended Component Definition
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EEPROM	Electrically Erasable Programmable Read Only Memory
eUICC	embedded UICC
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
FTP	Functional class Trusted Path/Channels
GCF	Global Certification Forum
GSM	Global System for Mobile
HMAC	Hash-based Message Authentication Code
ID	Identifier
IMEI	International Mobile Equipment Identity
IoT	Internet of Things
IPC	Inter-Process Communication
ISO/IEC	International Organization for Standardization/International Electrotechnical Commission
IT	Information Technology
JTAG	Joint Test Action Group
KAF	Known Authentication Factor
KDF	Key Derivation Function
KEK	Key Encryption Key
MAC	Message Authentication Code
NFC	Near Field Communication
NIST	National Institute of Standards and Technology
OEM	Original Equipment Manufacturer
OS	Operating System
PBKDF	Password-Based Key Derivation Function
PCB	Printed Circuit Board
PCS	Personal Communication Service
PIN	Personal Identification Number
PP	Protection Profile
PRF	Pseudo Random Function
RSA	Rivest-Shamir-Adleman (algorithm)
QR	Quick Response
RNG	Random Number Generator
SAR	Security Assurance Requirement
SEE	Separate Execution Environment

SESIP	Security Evaluation Standard for IoT Platforms
SFP	Security Function Policy
SFR	Security Functional Requirement
SHA2	Secure Hash Algorithm 2
SHA3	Secure Hash Algorithm 3
SMS	Short Message Service
SN	Serial Number
SOG-IS	Senior Officials Group Information Systems Security
SPD	Security Problem Definition
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 categorized as preloaded and downloaded apps. Examples of a CMD include smart phone, tablet and other device with similar capabilities. Users can customize 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 (main OS) and preloaded apps that are delivered with the CMD out of the box. Downloaded apps that can be completely uninstalled by the user are not considered part of the TOE.

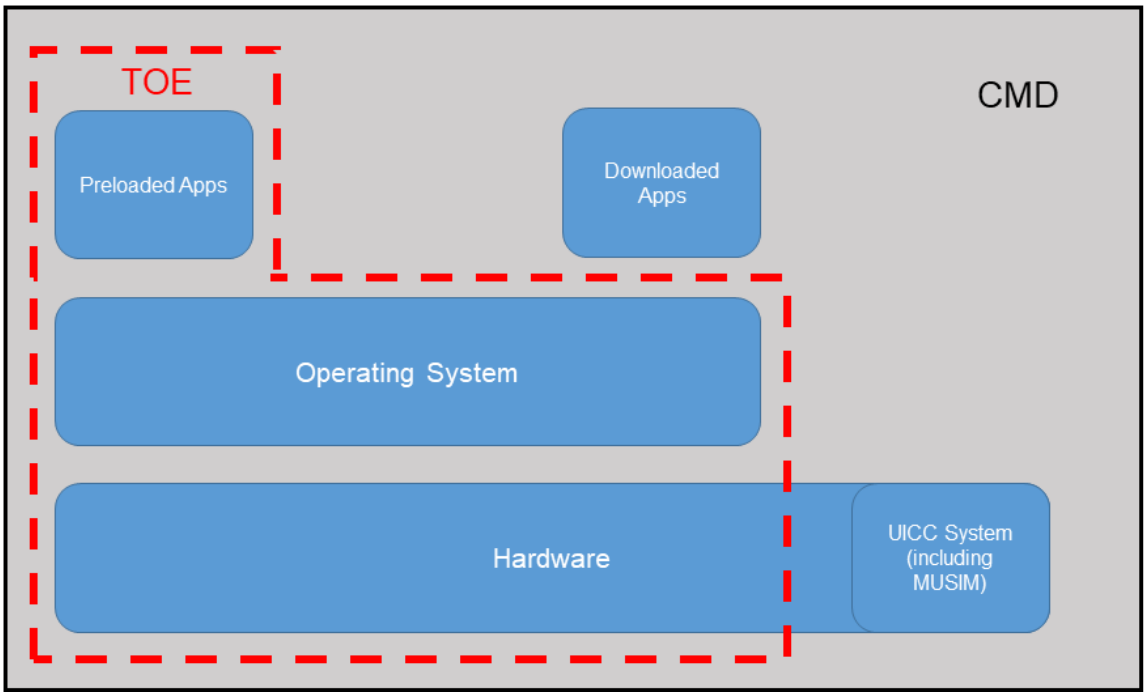


Figure 1: TOE boundary

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 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 security functionality and radio interface of the TOE related to cellular mobile communication are defined in [i.7], [i.8], [i.9] 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.6], and integrated eUICC is specified in [i.10], [i.11], [i.12] and [i.13], and will be certified by a CC Certification Body. Therefore, these cellular mobile communication 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 manufacturer should provide evidence for such assumption if the evaluation of the TOE depends on such evidence, such as certificate of eUICC.

The main OS of the TOE controls and manages the hardware and the apps (both downloaded and preloaded) and provides the user interface and application programming interface(s).

Apps on the CMD can be categorized in many ways, but generally are divided into downloaded and preloaded. The difference between them is whether they are present on the device when it is delivered to the consumer (preloaded apps).

Downloaded apps are those installed by the user by the choice of the user. Generally this is done through an App Distribution Platform (ADP) (this is out of scope of the TOE), though the exact mechanism for the installation of an app is up to the user. TOE manufacturers normally provide an ADP on the device for the user to take advantage of. The ADP generally also provides the ability to update the apps on the device that have been installed from the ADP. In some cases, to preserve user bandwidth, the TOE manufacturer may provide downloaded apps to the user as part of the initial system configuration. Any apps, regardless of whether they are pre-installed or downloaded by the user, that can be removed completely by the user, are treated as downloaded apps.

Preloaded apps are provided by the TOE manufacturer as part of the system software. Depending on the preloaded app, updates may be applied as part of the system software update or in the same manner as a downloaded app. As part of system software updates the TOE manufacturer may change the preloaded apps included on the device (adding or removing the apps from the system software).

Permissions are used to control access to specific services or capabilities on the device. A TOE manufacturer may provide specialized access to preloaded apps (such as an app for phone calls or for downloading system software updates). These permissions are collectively called system permissions, and can only be assigned by the TOE manufacturer to preloaded apps, they cannot be assigned to downloaded apps. A user can control access to other capabilities provided by the device, such as to storage or location services. Preloaded apps may also request permissions for additional capabilities (ones controlled specifically by the user), either on first use, or have them assigned by the TOE manufacturer by default. Use of permissions ensures that apps do not have access to more capabilities than are necessary.

The present document is intended to be used with a TOE which has system software approved by the TOE manufacturer, not an altered system which may enable the user to manage restricted system permissions, such as a rooted device.

The TOE manufacturer 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 user can customize the device by downloading apps and changing settings. A 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. The local environment consists of the device itself and peer devices such as those connected over short-range connections (such a wired connection, Bluetooth® or NFC). The remote environment consists of systems connected over longer-range wireless protocols such as WLAN (Wi-Fi®) and Mobile Networks (cellular). As specified in the TOE boundary, the connectivity provided by the Mobile Network is out of scope for this evaluation, though the connection offered by the network may be used by the TOE to access remote services. These have been highlighted in a different colour in the figure to show this difference.

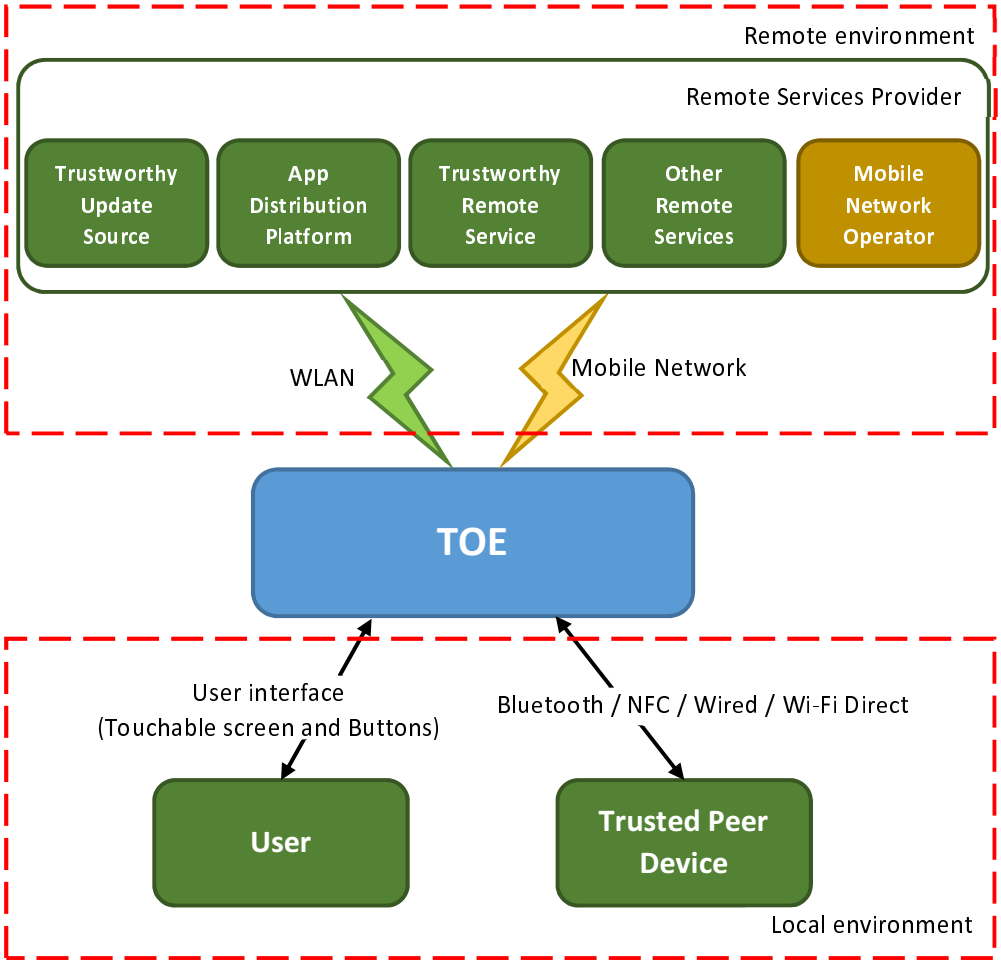


Figure 2: TOE environment

The TOE has a local environment with:

- A user, who physically interacts with the TOE across the provided interface(s).
- 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 Trustworthy Update Source, from which the TOE can download system software updates for the TOE software. The update source assured to be secure and the authenticity and integrity of the updates are ensured by the TOE manufacturer. These updates are digitally signed, and the TOE checks whether this signature is correct.
- The App Distribution Platform (ADP) of the TOE manufacturer and/or OS developer, from which the TOE can download and install apps (the TOE manufacturer may provide multiple ADPs in the TOE boundary). The App Distribution Platform performs operations for the detection of malicious in-app behaviour, conduct privacy disclosure inspections for apps that call, collect or upload sensitive data from users without permission, as well as scan apps for the presence of loopholes, vulnerabilities or backdoors. How the ADP performs security checks of applications is out of scope of this evaluation. The ADP application included on the TOE is in scope.

  To be considered a trustworthy source, 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.

  - (Out of scope) A user may install additional ADP applications, in which case the user takes the responsibility for the security of apps downloaded and installed from any additional ADPs.

- (Optionally) Trustworthy Remote Services provided by the TOE manufacturer. A user can use remote services to access user data in the TOE. It is assumed that the remote service will be secure, and vulnerabilities of remote service are not in scope of present document. The TOE manufacturer should ensure the security of the connection from the TOE to any remote service by utilizing capabilities provided by the TOE for secure remote connections.

- Other Remote Services offered by third parties, such as enterprise services, additional ADPs used by the user, websites, gaming servers, etc. The present document provides no assurance in these remote services, so it is up to the user to trust a particular remote service. The security of the connection from the TOE to any remote service can be secured utilizing capabilities provided by the TOE for secure remote connections but it is up to the app developer to utilize those capabilities.

- (Out of scope) 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 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.

It is important to be clear that the TOE shall provide three classification levels, but only Medium is explicitly implemented by the TOE itself for user data. Low and High classifications are generally implemented as an additional set of APIs as part of the main OS that an app can use to enable specific data storage functionality. As such, while the TOE supports the three classifications, only Medium is provided by default on the TOE for user data assets while Low/High requires apps to be provided (either by preload or download) to store user data assets according to one of those classifications. The key hierarchies described by these classification levels do not need to be tied together. Normally Medium and High will share a key hierarchy while Low will have a separate key hierarchy, but it is possible to have multiple independent hierarchies for different purposes.

Each of these classes of user data assets is encrypted (see FCS_CKH_EXT.1) 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_EXT.1 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.

The major security features are:

- Authentication of user: to ensure that the 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 can allow 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 unauthorized modification and unauthorized disclosure. These channels can subsequently be used by apps and by the TOE itself for various communication purposes.
- Secure updating of system software: the TOE can update the system software by downloading a system software update from a Trustworthy Update Source to address known vulnerabilities in a timely manner. System software updates do not update user data or any downloaded apps (and data associated with downloaded apps).
- Secure updating of apps: the TOE can update apps (as determined by the type of app) by downloading an update from the ADP.
- 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.
- 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 user and/or by the TOE manufacturer.
- 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 user. The 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 software/firmware, but trusted peer device(s) may be required.

4.4 Base-PP reference

Base-PP Title	ETSI TS 103 732-1: "Consumer Mobile Device; Part 1: Base Protection Profile"
Base-PP Version	2.1.2
Base-PP Date	November 16, 2023

4.5 Conformance Claim

The present document:

- claims conformance to CC V3.1 Release 5 [1], [2], [3] and the CC and CEM addenda [18];
- is CC Part 2 [2] extended and CC Part 3 [3] extended;
- does not claim conformance to any other PP;
- conforms to the package EAL2 augmented with ALC_DVS_EXT.1 & ALC_FLR.3.

In order to be conformant to this PP, a TOE shall demonstrate Exact Conformance. Exact Conformance, as a subset of Strict Conformance as defined by the CC, is defined as the Security Target (ST) containing all of the Security Functional Requirements in clause 8.3 (these are the mandatory SFRs) of this PP. While iteration is allowed, no additional requirements (from the CC parts 2 or 3, or definitions of extended components not already included in this PP) are allowed to be included in the ST. Further, no SFRs in clause 8.3 of this PP are allowed to be omitted.

While for SFRs the use of mandatory, optional and selection-based SFRs allows some customization when modelling the TOE, this does not work for the SPD in clause 5 and the security objectives in clause 6. Some parts in these chapters are marked as "(applies to ... only)" (e.g. "(applies to distributed TOEs only)"). These parts only need to be included in the ST for TOEs that comply with the corresponding conditions (i.e. parts marked as "(applies to distributed TOEs only)" only need to be included in STs for distributed TOEs and shall be omitted otherwise).

The packages to which exact conformance can be claimed in conjunction with this PP are specified in the "Allowed Packages" list as below:

- Consumer Mobile Device; Part 2: Biometric Authentication Protection Profile Module.

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 Trustworthy Remote Service, such as:
  - user files: photos, videos, etc.;
  - user location: location 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.
- TSF data: data which is used for the enforcement of operations of security functions, such as configuration data, user authentication data.
- The main OS and apps included in the TOE.

The interfaces defined here are used in the description of the threat agents and threats to the assets protected by the TOE. These interfaces implement security protocols to protect some of the TOE assets when transit to and from the TSF (with the exception of 3GPP-defined radio interfaces which are out of scope for this PP).

Interfaces of the TOE:

- The radio interface(s): these are wireless interfaces to wireless networks such as those defined by 3GPP (which are out of scope, see clause 4.2), Wi-Fi® or Bluetooth®.
- The network interface(s): these are interfaces to networks such as the internet. These interfaces generally (though not exclusively) lie on top of radio interfaces such as those provided by 3GPP, Wi-Fi® or Bluetooth.
- The physical interface: this is the physical enclosure of the TOE and may include the following types of ports: JTAG, USB, Lightning®, charging, media, etc.
- The user interface: this interface includes the screen, buttons, speaker, 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

The following threat Agents are defined for the TOE (note that one entity may be multiple threat agents simultaneously):

- 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 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.

Threat Agents are limited to Basic attack potential as defined by EAL2. For further detail and examples on attack potential, see annexes B3, B4 and B5 of [4].

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.

Attacks based on the 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 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.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.

T.IMPERSONATE - TA.PHYSICAL impersonates the legitimate user of the TOE thereby gaining access to the user data assets.

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

EXAMPLE: 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 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 user buys this CMD and the modified CMD allows compromise of the user data assets.

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 (location, camera, microphone, etc.).

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

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

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 Organizational Security Policies

P.DATA_CLASSIFICATION - The TOE manufacturer 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 user if the user data asset were inappropriately accessed, used, or disclosed:

- Low: user data assets to which unauthorized access is expected to have limited adverse effect on the user. Low user data assets can only be decrypted on the TOE.
- Medium: user data assets to which unauthorized access can have serious adverse effect on the 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 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 user will only install apps which have been downloaded from the App Distribution Platform of the TOE manufacturer and/or OS developer, 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 user without permission;
- scan apps for the presence of loopholes, vulnerabilities or backdoors;
- verify an App developer has a valid developer certificate.

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

A.TRUSTWORTHY_REMOTE_SERVICE - It is assumed that the remote service provided by the TOE manufacturer is secure and unauthorized parties cannot access to the user data assets on the TOE via remote services.

A.PASSWORD_PIN_PATTERN - It is assumed that the user will not downgrade/disable authentication to the TOE or choose easily guessable known authentication factors.

6 Security Objectives

6.1 Security Objectives for the TOE

O.PROTECT_COMMS - The TOE ensures confidentiality and integrity of assets are protected during transmission over secure protocol.

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

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

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.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.

O.ACCESS_CONTROL - The TOE ensures that apps only gain access to user data assets that they are specifically allowed by the user or the main OS 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 user of the TOE before allowing that user has full access to the TOE functionality.

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

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

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

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 unauthorized modification by other apps.

6.2 Security Objectives for the Operational Environment

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

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

OE.TRUSTWORTHY_REMOTE_SERVICE - The operational environment ensures the remote services are sufficiently protected from unauthorized access.

OE.PASSWORD_PIN_PATTERN - The operational environment ensures that user of the TOE is instructed not to disable or downgrade the authentication of the TOE or choose easily guessable known authentication factors 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 threat is further countered by O.CRYPTOGRAPHY to encrypt these channels and O.RANDOMS to provide random numbers for key generation used in the encryption.
T.SPOOF	This threat is countered by O.AUTHENTICATE_USER and O.ACCESS_CONTROL 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 threat is further countered by O.CRYPTOGRAPHY to encrypt these channels and O.RANDOMS to provide random numbers for key generation used in the encryption.
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. • O.CRYPTOGRAPHY and O.RANDOMS ensures that stored data is encrypted with strong keys. • 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 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 and O.SECURE_BOOT, stating that the TOE protects itself against apps attempting to alter its security behaviour, and O.AUTHENTICATED_UPDATES to prevent unauthorized updates of any part of the TOE.
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, and O.AUTHENTICATED_UPDATES to prevent unauthorized updates of any part of 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, O.CRYPTOGRAPHY and O.RANDOMS, ensuring that the user data assets, which are all encrypted with strong keys, are permanently unreadable.
T.NEW_ATTACKS	This threat is countered by O.AUTHENTICATED_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.TRUSTWORTHY_UPDATE_SOURCE	This assumption is directly implemented by OE.TRUSTWORTHY_UPDATE_SOURCE.
A.TRUSTWORTHY_REMOTE_SERVICE	This assumption is directly implemented by OE.TRUSTWORTHY_REMOTE_SERVICE.

7 Extended Components Definition

7.1 Definition of the family Random Number Generation (FCS_RNG_EXT)

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

Family behaviour

This family 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_EXT) 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_EXT: Generation of Random Numbers	1

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

Management: FCS_RNG_EXT.1

There are no management activities foreseen.

Audit: FCS_RNG_EXT.1

There are no actions defined to be auditable.

FCS_RNG_EXT.1 Random number generation

Hierarchical to: No other components.

Dependencies: No dependencies.

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

FCS_RNG_EXT.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_EXT)

Family behaviour

This family 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_EXT) 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_EXT: Cryptographic Key Hierarchy	1

FCS_CKH_EXT.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_EXT.1

There are no management activities foreseen.

Audit: FCS_CKH_EXT.1

There are no actions defined to be auditable.

FCS_CKH_EXT.1 Cryptographic Key Hierarchy

Hierarchical to:	No other components.
Dependencies:	[FCS_CKM.1 Cryptographic key generation, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic Key Destruction.

FCS_CKH_EXT.1.1 The TSF shall support a key hierarchy for data encryption keys to protect [assignment: *list of data to be protected by the key hierarchy*].

FCS_CKH_EXT.1.2 The TSF shall ensure that all keys in key hierarchy are derived and/or generated according to [assignment: *description of how each key in the hierarchy is derived and/or generated, according to FCS_CKM.1/Symmetric, FCS_CKM.1/Asymmetric and/or FCS_COP.1/Derivation*] ensuring that the key hierarchy uses the DUK and [selection: *the user credentials, no user credentials*] directly or indirectly in the derivation of the data encryption key(s) for [assignment: *class of data assets*].

FCS_CKH_EXT.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*].

User application notes

The selection in FCS_CKH_EXT.1.2 specifies how user credentials should be used to protect the user data assets. While the DUK is always used to tie the key hierarchy to the unique device, to tie data specifically to the user requires the user of the user credentials as part of the derivation process. The specification of the class of data assets will tie the type of protection offered to the specific data, such that any data tied to user credentials can only be accessed once the user is successfully authenticated.

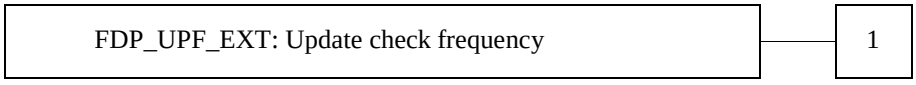
The protection assignment in FCS_CKH_EXT.1.3 can use protection methods like encryption by other keys in the key hierarchy, keys that are deleted after use and re-generated/re-derived when needed, or secure storage of the key (such as in a hardware-based secure environment).

7.3 Definition of the family Update Check Frequency (FDP_UPF_EXT)

Family behaviour

This family defines requirements for the frequency the TOE shall check for updates to apps or system software and actions if an update is available. The sole security functional component is defined in an additional family (FDP_UPF_EXT) of the Class FDP (User Data Protection).

Component levelling



FDP_UPF_EXT.1, Update check frequency, requires the TSF to provide an automatic check for updates of apps or system software. Rules defined by for the update process will determine the actions taken when an update is found.

Management: FDP_UPF_EXT.1

There are no management activities foreseen.

Audit: FDP_UPF_EXT.1

There are no actions defined to be auditable.

FDP_UPF_EXT.1 Update check frequency

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_UPF_EXT.1.1 The TSF shall be able to check for a [selection: *app, system software*] update package every [assignment: *interval period*].

User application notes

The assignment in FDP_UPF_EXT.1 specifies the frequency of the update check from the TOE. This does not mandate a successful connection which can be impacted by the network connectivity of the TOE during the period of the check. A check of the online status of the TOE before an attempt to connect to the update location is part of the update process.

7.4 Definition of the Trusted Channel Protocol (FTP_ITC_EXT.1)

This clause describes a component of the family Inter-TSF trusted channel (FTP_ITC) for the definition of protocols to be available for use to on the TOE. The use of standard protocols provides a means for disparate systems to communicate securely.

Management: FTP_ITC_EXT.1

There are no management activities foreseen.

Audit: FTP_ITC_EXT.1

There are no actions defined to be auditable.

FTP_ITC_EXT.1 Trusted channel protocol

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_ITC_EXT.1.1 The TSF shall use [assignment: *protocol type and standards defining the implementation of this protocol*] to 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_EXT.1.2 The TSF shall permit [selection: *the TSF, another trusted IT product*] to initiate communication via the trusted channel.

FTP_ITC_EXT.1.3 The TSF shall initiate communication via the trusted channel for [assignment: *list of functions for which a trusted channel is required*].

FTP_ITC_EXT.1.4 The protocol used by the communications channel shall support the following requirements: [assignment: *list of requirements to be met by the protocol*].

7.5 Definition of the Identification of security measures for device identifiers (ALC_DVS_EXT)

The security of a mobile device depends in part on several steps taken during the manufacturing process. Devices which are manufactured without proper controls on key security information may open the device to later unauthorized access.

Identification of security measures for device identifiers (ALC_DVS_EXT) defines requirements for evaluator review of the manufacturing process related to how device identifiers and encryption keys are handled while they are installed on the TOE. Formally, ALC_DVS_EXT.1 narrows the scope of the review to focus on the manufacturing process surrounding these keys and identifiers while removing the requirement for an onsite review of the manufacturing facilities.

This clause describes a component of the family Development security (ALC_DVS) for reviewing manufacturing steps in the creation of a mobile device around signing keys and the deployment of unique identifiers on the device.

Application notes

The development security requirement ALC_DVS_EXT.1 focuses on the core security requirements surrounding the deployment of unique identifiers and keys to the device as well as the signing keys used on the software during manufacturing.

The evaluator performs a review of the documentation related to the manufacturing process and the organizational security policies that control access to the deployment of the keys and identifiers to the devices.

ALC_DVS_EXT.1 Identification of security measures for device identifiers

Dependencies: No dependencies.

Developer action elements:

ALC_DVS_EXT.1.1D The developer shall produce and provide development security documentation on the generation and protection of signing keys and device-unique identifiers.

Content and presentation elements:

ALC_DVS_EXT.1.1C The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the following manufacturing components:

- keys used to sign the publicly released system software and its updates and to ensure the use of the proper keys in the build process;
- unique, non-modifiable identifiers (such as IMEI, attestation keys or Device Unique Keys) are properly created and provisioned for each device.

Evaluator action elements:

ALC_DVS_EXT.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

ALC_DVS_EXT.1.2E The evaluator *shall review* the provided evidence that the security measures are being applied.

Evaluation of sub-activity (ALC_DVS_EXT.1)

Objectives:

The objective of this sub-activity is to determine whether the developer's security controls on the development environment are adequate to provide the confidentiality and integrity of the TOE design and implementation that is necessary to ensure that secure operation of the TOE is not compromised.

Input:

The evaluation evidence for this sub-activity is:

a) the ST;

b) the development security documentation.

In addition, the evaluator may need to examine other deliverables to determine that the security controls are well-defined and followed. Evidence that the procedures are being applied is also required.

Action ALC_DVS_EXT.1.1E

ALC_DVS_EXT.1.1C ***The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the following manufacturing components:***

- ***keys used to sign the publicly released system software and its updates and to ensure the use of the proper keys in the build process;***
- ***unique, non-modifiable identifiers (such as IMEI, attestation keys or Device Unique Keys) are properly created and provisioned for each device.***

ALC_DVS_EXT.1-1 The evaluator ***shall examine*** the development security documentation to determine that it details all security measures used in the manufacturing environment that are necessary to protect the confidentiality and integrity of the TOE signing keys and unique, non-modifiable identifiers.

The evaluator determines what is necessary by first referring to the ST for any information that may assist in the determination of necessary protection.

If no explicit information is available from the ST the evaluator will need to make a determination of the necessary measures. In cases where the developer's measures are considered less than what is necessary, a clear justification should be provided for the assessment, based on a potential exploitable vulnerability.

The following types of security measures are considered by the evaluator when examining the documentation:

a) physical, for example physical access controls used to prevent unauthorized access to the TOE manufacturing environment (during normal working hours and at other times);

b) procedural, for example covering:

- granting of access to the manufacturing environment or to specific parts of the environment;
- revocation of access rights when a person leaves the manufacturing team;
- transfer of protected material within and out of the manufacturing environment and between different sites in accordance with defined acceptance procedures;
- admitting and escorting visitors to the manufacturing environment;
- roles and responsibilities in ensuring the continued application of security measures, and the detection of security breaches.

c) personnel, for example any controls or checks made to establish the trustworthiness of new manufacturing staff;

d) other security measures, for example the logical protections on any machines related to manufacturing.

The development security documentation should identify the locations at which manufacturing occurs, and describe the security measures applied at each location and for transports between different locations. For example, manufacturing may could occur at one facility while the keys and unique identifiers are acquired/generated at a different facility. Transport of information between facilities is covered by ALC_DVS_EXT.

ALC_DVS_EXT.1-2 The evaluator ***shall examine*** the manufacturing confidentiality policies in order to determine the sufficiency of the security measures employed.

The evaluator should examine whether the following is included in the policies:

a) which members of the manufacturing staff are allowed to access the various key and identifier materials;

b) how signing keys are generated, and which members are allowed to generate and assign keys to device models to be manufactured.

The evaluator should determine that these policies are described in the development security documentation, that the security measures employed are consistent with the policies, and that they are complete.

Action ALC_DVS_EXT.1.2E

ALC_DVS_EXT.1-3 The evaluator ***shall examine*** the development security documentation and associated evidence to determine that the security measures are being applied.

This work unit requires the evaluator to determine that the security measures described in the development security documentation are being followed, such that the integrity of the TOE and the confidentiality of associated documentation is being adequately protected. For example, this could be determined by examination of the documentary evidence provided.

8 Security requirements

8.1 Overview

This clause describes the security requirements which have to be fulfilled by the product under evaluation. Those requirements comprise functional components from Part 2 [2] and assurance components from Part 3 [3] of the Common Criteria.

8.2 Conventions

The following conventions are used for the completion of operations defined in the SFRs:

- Unaltered SFRs are stated in the form used in Part 2 [2] or their Extended Component Definition (ECD)
- Refinement made in the PP: the refinement text is indicated with **bold text** and ~~strikethroughs~~
- Selection wholly or partially completed in the PP: the selection values (i.e. the selection values adopted in the PP or the remaining selection values available for the ST) are indicated with UNDERLINED UPPERCASE TEXT:
  - e.g. '[selection: *disclosure, modification, loss of use*]' in Part 2 [2] or an ECD might become 'DISCLOSURE' (completion) or '[selection: DISCLOSURE, MODIFICATION]' (partial completion) in the PP
- Assignment wholly or partially completed in the PP: *INDICATED WITH UPPERCASE ITALICIZED TEXT*
- Assignment completed within a selection in the PP: the completed assignment text is indicated with *ITALICIZED AND UNDERLINED UPPERCASE TEXT:*
  - e.g. '[selection: change_default, query, modify, delete, [assignment: other operations]]' in Part 2 [2] or an ECD might become 'CHANGE_DEFAULT, *SELECT_TAG'* (completion of both selection and assignment) or '[selection: CHANGE_DEFAULT, *SELECT_TAG*, *SELECT_VALUE*]' (partial completion of selection, and completion of assignment) in the PP
- Iteration: indicated by adding a string starting with '/' (e.g. 'FCS_COP.1/Hash')
- Extended SFRs are identified by having a label 'EXT' at the end of the SFR name

8.3 Security functional requirements

8.3.1 Cryptographic Support (FCS)

A CMD under evaluation may support many cryptographic capabilities to provide the functionality expected by customers of the device. The cryptographic support of the TOE, as a subset of the overall functionality of the device, does not need to include capabilities which do not support the claims being made in the evaluation. The capabilities and algorithms specified in the requirements here only shall list those algorithms which are used to support requirements dependent on those algorithms.

A CMD may support a wide range of algorithms that can be used by calling apps and services. For the purposes of the evaluation, only those algorithms that are used by the CMD to support claimed security functions shall be enumerated and evaluated here. While the cryptographic algorithms used for the evaluation are not specified, they are expected to meet a minimum level of strength, generally defined as being equivalent to 128-bit block cipher encryption.

The requirements do not specify a catalogue of algorithms to choose from, but instead focus on the traits of algorithms that would be considered acceptable. An example catalog of algorithms that would be considered acceptable is the SOG-IS Agreed Cryptographic Mechanisms [i.14] document.

A CMD may support multiple key hierarchies depending on the root key(s) used for security of the hierarchy. The design of the key hierarchies shall be specified as to how the requirements specified here are met. The key hierarchy requirements focus on how the keys are tied to the root key(s) and how they are used for protection of data, not how they are designed or the number of hierarchies used to meet the individual requirements.

FCS_RNG_EXT.1 Random number generation

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

FCS_RNG_EXT.1.2 The TSF shall provide random numbers that meet [assignment: *FOR A DETERMINISTIC RNG A PUBLIC CRYPTOGRAPHIC STANDARD THAT MEETS THE QUALIFICATION CRITERIA FOR NEW CIPHERS AS SPECIFIED IN ISO/IEC 18033-1 ANNEX A [17], FOR OTHER RNGS EVIDENCE THAT THE OUTPUT HAS BEEN STATISTICALLY REVIEWED TO PROVIDE SUFFICIENT ENTROPY FOR KEY GENERATION]*.

FCS_CKM.1/Asymmetric Cryptographic key generation

FCS_CKM.1.1/Asymmetric The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [assignment: *ASYMMETRIC cryptographic key generation algorithm*] and specified cryptographic key sizes [assignment: *cryptographic key sizes EQUAL TO OR GREATER THAN 128-BIT SYMMETRIC KEYS*] that meet the following: [assignment: *A PUBLIC CRYPTOGRAPHIC STANDARD THAT MEETS THE QUALIFICATION CRITERIA FOR NEW CIPHERS AS SPECIFIED IN ISO/IEC 18033-1 ANNEX A [17]*.

FCS_CKM.1/Symmetric Cryptographic key generation

FCS_CKM.1.1/Symmetric The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [~~assignment: *cryptographic key generation algorithm*~~ selection: *FOR SYMMETRIC ENCRYPTION USING A RANDOM NUMBER GENERATOR AS SPECIFIED IN FCS_RNG_EXT.1, FOR SYMMETRIC ENCRYPTION A COMBINATION OF PRECURSOR KEY MATERIAL USING A DERIVATION FUNCTION AS SPECIFIED IN FCS_COP.1/DERIVATION*] and specified cryptographic key sizes [assignment: *cryptographic key sizes EQUAL TO OR GREATER THAN 128-BIT SYMMETRIC KEYS*] ~~that meet the following: [**assignment**: *list of standards*]~~.

FCS_COP.1/SigGen Cryptographic operation

FCS_COP.1.1/SigGen The TSF shall perform [*CRYPTOGRAPHIC SIGNATURE SERVICES (GENERATION AND VERIFICATION)*] in accordance with a specified cryptographic algorithm [assignment: *cryptographic algorithm*] and cryptographic key sizes [assignment: *cryptographic key sizes EQUAL TO OR GREATER THAN 128-BIT SYMMETRIC KEYS*] that meet the following: [assignment: *A PUBLIC CRYPTOGRAPHIC STANDARD THAT MEETS THE QUALIFICATION CRITERIA FOR NEW CIPHERS AS SPECIFIED IN ISO/IEC 18033-1 ANNEX A [17]*.

Application Note 1: Examples of signature generation/verification algorithms are: RSA or ECDSA.

FCS_COP.1/KeyEst Cryptographic operation

FCS_COP.1.1/KeyEst The TSF shall perform [*CRYPTOGRAPHIC KEY ESTABLISHMENT*] in accordance with a specified cryptographic algorithm [assignment: *cryptographic KEY ESTABLISHMENT SCHEME algorithm*] and cryptographic key sizes [assignment: *cryptographic key sizes EQUAL TO THE KEY SIZES BEING USED FOR ENCRYPTION*] that meet the following: [assignment: *A PUBLIC CRYPTOGRAPHIC STANDARD THAT MEETS THE QUALIFICATION CRITERIA FOR NEW CIPHERS AS SPECIFIED IN ISO/IEC 18033-1 ANNEX A [17]*.

Application Note 2: Examples of key establishment schemes are: NIST Special Publication 800-56Ar3 for Pair-Wise Key Establishment.

FCS_COP.1/Symmetric Cryptographic operation

FCS_COP.1.1/Symmetric The TSF shall perform [*SYMMETRIC ENCRYPTION AND DECRYPTION*] in accordance with a specified cryptographic algorithm [assignment: *cryptographic algorithm*] and cryptographic key sizes [assignment: *cryptographic key sizes EQUAL TO OR GREATER THAN 128-BIT SYMMETRIC KEYS*] that meet the following: [assignment: *A PUBLIC CRYPTOGRAPHIC STANDARD THAT MEETS THE QUALIFICATION CRITERIA FOR NEW CIPHERS AS SPECIFIED IN ISO/IEC 18033-1 ANNEX A [17]*.

Application Note 3: Examples of symmetric algorithms are: AES or Camellia.

FCS_COP.1/Derivation Cryptographic operation

FCS_COP.1.1/Derivation The TSF shall perform [*DERIVATION FUNCTION*] in accordance with a specified cryptographic algorithm [assignment: *cryptographic algorithm*] and cryptographic key sizes [assignment: *cryptographic key sizes EQUAL TO THE KEY SIZES BEING USED FOR ENCRYPTION*] that meet the following: [assignment: *A PUBLIC CRYPTOGRAPHIC STANDARD THAT MEETS THE QUALIFICATION CRITERIA FOR NEW CIPHERS AS SPECIFIED IN ISO/IEC 18033-1 ANNEX A [17]*.

Application Note 4: Examples of derivation functions algorithms are: PBKDF, NIST Special Publication 800-108 KDF.

FCS_COP.1/Hash Cryptographic operation

FCS_COP.1.1/Hash The TSF shall perform [*CRYPTOGRAPHIC HASHING*] in accordance with a specified cryptographic algorithm [assignment: *cryptographic algorithm*] and cryptographic key sizes [*NONE*] that meet the following: [assignment: *A PUBLIC CRYPTOGRAPHIC STANDARD THAT MEETS THE QUALIFICATION CRITERIA FOR NEW CIPHERS AS SPECIFIED IN ISO/IEC 18033-1 ANNEX A [17]*.

Application Note 5: Examples of hash algorithms are: SHA2 or SHA3.

FCS_COP.1/KeyedHash Cryptographic operation

FCS_COP.1.1/KeyedHash The TSF shall perform [*KEYED-HASH MESSAGE AUTHENTICATION*] in accordance with a specified cryptographic algorithm [assignment: *cryptographic algorithm*] and cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: *A PUBLIC CRYPTOGRAPHIC STANDARD THAT MEETS THE QUALIFICATION CRITERIA FOR NEW CIPHERS AS SPECIFIED IN ISO/IEC 18033-1 ANNEX A [17]*.

Application Note 6: Examples of hashed message authentication algorithms are: HMAC-SHA2 or KMAC256.

FCS_CKH_EXT.1/Low Cryptographic key hierarchy

FCS_CKH_EXT.1.1/Low The TSF shall support a key hierarchy for data encryption keys to protect [*LOW USER DATA ASSETS*].

FCS_CKH_EXT.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, according to FCS_CKM.1/Symmetric, ~~FCS_CKM.1/Asymmetric~~ and/or FCS_COP.1/Derivation*] ensuring that the key hierarchy uses the DUK and [NO USER CREDENTIALS] directly or indirectly in the derivation of the data encryption key(s) for [*LOW USER DATA ASSETS*].

FCS_CKH_EXT.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*].

FCS_CKH_EXT.1/MediumHigh Cryptographic key hierarchy

FCS_CKH_EXT.1.1/MediumHigh The TSF shall support a key hierarchy for data encryption keys to protect [*MEDIUM AND HIGH USER DATA ASSETS*].

FCS_CKH_EXT.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, according to FCS_CKM.1/Symmetric, FCS_CKM.1/Asymmetric and/or FCS_COP.1/Derivation*] ensuring that the key hierarchy uses the DUK and [THE USER CREDENTIALS] directly or indirectly in the derivation of the data encryption key(s) for [*MEDIUM AND HIGH USER DATA ASSETS*].

FCS_CKH_EXT.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*].

FCS_CKM.4 Cryptographic key destruction

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [~~assignment:~~

- *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*].

Application Note 7: When no standard is applicable in the assignment, it can be left none.

8.3.2 User Data Protection (FDP)

8.3.2.1 The Update Policy

The Update Policy defines:

- How often the TSF will check whether new app or system software updates are available.
- Depending on the update type, the package can update either a single app or portions of the system software. System software updates can also replace the cryptographic data used to determine the validity (see FCS_COP.1/SigGen) of any future system software updates.
- The conditions under which the TSF uses update packages to install newer versions of the app or system software.

The Update Policies define the following objects, attributes and operations:

Objects:

- App: any installed application.
- App_Update_Package: the file(s) that contain a new version of an installed application.
- SSW: the system software of the TOE.

- SSW_Update_Package: the file(s) that contain a new version of the system software.

Attributes:

- Version_ID: the version identifier of the app, system software or update package.
- Signature: the cryptographic signature associated with, as applicable, the app (or app developer), system software or update package.
- Package_Source: the location where the Update_Package is acquired from (from the standpoint of the TSF).

Operations:

- Update_App: The update process for installing the new version of the application.
- Update_SSW: The update process for installing a new version of the system software.

FDP_ACC.1/APP_Update Subset access control

FDP_ACC.1.1/APP_Update The TSF shall enforce the [*APP_UPDATE POLICY*] on [*SUBJECTS: THE TSF, OBJECTS: APP, APP_UPDATE_PACKAGE, OPERATIONS: UPDATE_APP*].

Application Note 8: The Object "App" is used here generically to mean any app that can be updated through the ADP (downloaded or preloaded).

FDP_ACF.1/APP_Update Security attribute based access control

FDP_ACF.1.1/APP_Update The TSF shall enforce the [*APP_UPDATE POLICY*] to objects based on the following: [*SUBJECTS: THE TSF, OBJECTS[ATTRIBUTES]: APP[VERSION_ID, SIGNATURE], APP_UPDATE_PACKAGE[VERSION_ID, SIGNATURE, PACKAGE_SOURCE]*].

Application Note 9: The Object "App" is used here generically to mean any app that can be updated through the ADP (downloaded or preloaded).

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

- *THE TSF SHALL ALLOW THE TSF TO UPDATE_APP WITH AN APP_UPDATE_PACKAGE ONLY IF:*
  - *THE TSF SUCCESSFULLY VERIFIES THE SIGNATURE OF THE APP_UPDATE_PACKAGE AND THE SIGNATURE IS FROM THE SAME APP OR APP DEVELOPER; AND*
  - *[selection: <u>THE VERSION_ID OF THE APP_UPDATE_PACKAGE IS NOT LOWER THAN THE VERSION_ID OF THE INSTALLED APP, THE UPDATE IS DOWNLOADED FROM THE APP DISTRIBUTION PLATFORM OF THE TOE MANUFACTURER OR OS DEVELOPER</u>];*
- *THE TSF UPDATE_APP IS AN ATOMIC UPDATE FUNCTION*].

Application Note 10: Version_ID tracking is only available for currently installed apps, so if a user uninstalls an app it is possible for the user to install any version of the app, including older ones than what was uninstalled, as the process of uninstalling the app removes any reference for the version_ID comparison.

FDP_ACF.1.3/APP_Update The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [assignment: *OTHER RULES ENSURING AUTHENTICITY AND INTEGRITY OF THE UPDATE PACKAGE*].

FDP_ACF.1.4/APP_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 RELATED TO THE UPDATE_APP OPERATION OR ACCESS TO THE APP DURING ITS UPDATING*].

FDP_UPF_EXT.1/APP_Update Update check frequency

FDP_UPF_EXT.1.1/APP_Update The TSF shall be able to check for a [<u>APP</u>] update package every [assignment: ~~*interval period*~~ ***an interval of no more than 1 day***].

FDP_ACC.2/SSW_Update Complete access control

FDP_ACC.2.1/SSW_Update The TSF shall enforce the [*SSW_UPDATE POLICY*] on [*SUBJECTS: THE TSF, OBJECTS: THE SSW, SSW_UPDATE_PACKAGE, OPERATIONS: UPDATE_SSW*].

FDP_ACC.2.2/SSW_Update The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

FDP_ACF.1/SSW_Update Security attribute based access control

FDP_ACF.1.1/SSW_Update The TSF shall enforce the [*SSW_UPDATE POLICY*] to objects based on the following: [*SUBJECTS: THE TSF, OBJECTS[ATTRIBUTES]: SSW[VERSION_ID, SIGNATURE], SSW_UPDATE_PACKAGE[VERSION_ID, SIGNATURE, PACKAGE_SOURCE]*].

FDP_ACF.1.2/SSW_Update 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 PERFORM THE UPDATE_SSW OPERATION IF THE FOLLOWING CONDITIONS HOLD:*
  - *[selection: THE SSW_UPDATE_PACKAGE[VERSION_ID] IS NOT LOWER THAN THE SSW[VERSION_ID], THE SSW_UPDATE[PACKAGE_SOURCE] IS THE TRUSTWORTHY UPDATE SOURCE DIRECTLY];*
  - *THE SSW_UPDATE_PACKAGE[SIGNATURE] IS VERIFIED BY A DIGITAL SIGNATURE FROM THE TOE MANUFACTURER STORED ON THE DEVICE;*
  - *THE SIGNATURE CHECK AND THE UPDATE_SSW ARE PERFORMED AS AN ATOMIC UPDATE FUNCTION]*.

FDP_ACF.1.3/SSW_Update The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [assignment: *OTHER RULES ENSURING AUTHENTICITY AND INTEGRITY OF THE UPDATE PACKAGE*].

FDP_ACF.1.4/SSW_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 RELATED TO THE UPDATE_SSW FUNCTION DURING ITS UPDATING*].

Application Note 11: Failure of correct installation of the update is handled in FPT_FLS.1.

FDP_UPF_EXT.1/SSW_Update Update check frequency

FDP_UPF_EXT.1.1/SSW_Update The TSF shall be able to check for a [SYSTEM SOFTWARE] update package every [assignment: ~~*interval period*~~ ***an interval of no more than 1 month***].

8.3.2.2 The Permissions Policy

The Permissions Policy defines the access of apps and processes to objects outside its own security domain. The permissions can range from user-oriented controls, such as access to the camera, microphone, and address book, to more explicit access to underlying functionality in the main OS, such as to files, secure IPCs and secure sockets.

User Objects and Manufacturer Objects are defined by whether the user actually has the ability to change the access to the object. Access control lists related to User Objects can be managed by the user, while access control lists related to Manufacturer Objects can only be managed by the TOE manufacturer.

User Objects focus on access to system components and services which the user normally sees as permissions, such as to the camera or contacts. Even when assigned from the TOE manufacturer by default these permissions can be changed by the user (though changing TOE manufacturer-assigned permissions can have unintended consequences to the capabilities of the device). The permission/revocation of this access to User Objects is defined in FMT_SMF.1/Permissions.

Manufacturer Objects focus on access to the underlying system components and services which the user does not have direct access to and cannot change. These objects provide controls that ensure apps or processes only have access to their own defined security domain using and usually controlled by the main OS kernel. The Manufacturer Objects may provide the underlying capabilities which the user sees as permissions, but also enforces the boundary of the app or process around the overall security domain.

Of the Manufacturer Objects defined in the requirement, system permissions is a general list of access rights that control functions of the main OS that can only be assigned by the TOE manufacturer. As a Manufacturer Object, the user does not have the ability control the permissions set by the TOE manufacturer.

FDP_ACC.2/Permissions Complete access control

FDP_ACC.2.1/Permissions The TSF shall enforce the [*PERMISSIONS POLICY*] on [

- *SUBJECTS: APPS, PROCESSES;*
- *USER OBJECTS: [selection: CAMERA, MICROPHONE, LOCATION, CONTACTS, CALENDAR, CALL LOG, STORED PICTURES, TEXT MESSAGES, THE LIST OF INSTALLED APPS, [assignment: LIST OF OTHER SENSITIVE USER DATA ASSETS AND/OR SYSTEM SERVICES THAT CAN BE ACCESSED BY AN APP OR PROCESS]];*
- *MANUFACTURER OBJECTS: DEVICE ID, SYSTEM PERMISSIONS, FILES (INCLUDING INDIVIDUAL APP DATA), [SELECTION: SECURE SOCKETS, SECURE IPC, [assignment: LIST OF OTHER SENSITIVE USER DATA ASSETS AND/OR SYSTEM SERVICES THAT CAN BE ACCESSED BY AN APP OR PROCESS]];*
- *OPERATIONS: READ, WRITE, EXECUTE*].

Application Note 12: If a device supports a specific User Object, it shall be selected.

Terminology used for Manufacturer Objects can be defined in the assignment if secure sockets and secure IPC are not relevant terms.

The TOE manufacturer may assign a User Object permission to a privileged app or process that cannot be changed by the user. If this is done, the app or process and the User Object shall be specified in the Manufacturer Objects assignment and for the purposes of that app or process, the permission is considered a Manufacturer Object.

The TOE manufacturer establishes the permissions for app data to ensure that apps are restricted from accessing the data from another app except under specific circumstances (such as apps from a common app developer).

FDP_ACC.2.2/Permissions The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

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: [

- *[APPS AND PROCESSES] AND THE OPERATIONS ASSOCIATED WITH THE SUBJECT;*
- *THE ACCESS CONTROL LIST ASSOCIATED WITH THE OBJECT BEING REQUESTED*].

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: [

- *THE SUBJECT, OR A GROUPING THE SUBJECT IS MAPPED TO, IS EXPLICITLY GRANTED PERMISSION BY THE USER OR TOE MANUFACTURER TO THE USER OBJECT IN THE ACCESS CONTROL LIST*].

FDP_ACF.1.3/Permissions The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [

- *THE SUBJECT IS GRANTED PERMISSION BY THE TOE MANUFACTURER TO THE MANUFACTURER OBJECT IN THE ACCESS CONTROL LIST*].

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

- *THE SUBJECT IS EXPLICITLY BLOCKED BY THE USER FROM ACCESSING THE USER OBJECT;*
- *THERE IS NO RULE GRANTING THE SUBJECT ACCESS TO THE USER OR MANUFACTURER OBJECT IN THE ACCESS CONTROL LIST*].

8.3.2.3 The Data Classification Policy

FDP_ACC.1/UserDataAsset Subset access control

FDP_ACC.1.1/UserDataAsset The TSF shall enforce the [*USER DATA ASSET DECRYPTION POLICY*] on [

- *SUBJECTS: THE TSF;*
- *OBJECTS: INTERNAL STORAGE (ALL SAVED USER DATA ASSETS), [selection: REMOVABLE STORAGE (ALL SAVED USER DATA ASSETS WHEN ENCRYPTION IS ENABLED FOR REMOVABLE STORAGE), NO OTHER STORAGE];*
- *OPERATIONS: DECRYPT*].

FDP_ACF.1/UserDataAsset 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: [*SUBJECTS: THE TSF, OBJECTS: USER DATA ASSETS, ATTRIBUTES: SENSITIVE LEVEL OF OBJECTS, 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*].

Application Note 13: The phrase "successfully powered on" means that the TOE has successfully booted up and completed all tests required by FPT_TST.1.

FDP_ACF.1.3/UserDataAsset The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [*NO ADDITIONAL RULES*].

FDP_ACF.1.4/UserDataAsset The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [*NO ADDITIONAL RULES*].

8.3.3 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 USER*] on behalf of the user to be performed before the user is authenticated.

Application Note 14: Actions not listed in the assignment are not allowed before user authentication.

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 user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

FIA_UID.1 Timing of identification

FIA_UID.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 USER*] on behalf of the user to be performed before the user is identified.

FIA_UID.1.2 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

FIA_UAU.5/Local Multiple authentication mechanisms

FIA_UAU.5.1/Local The TSF shall provide [*PASSWORD, PIN* and selection: *PATTERN, [assignment: EAF], NO OTHER MECHANISM*] to support user authentication.

Application Note 15: Use of an EAF assumes that the EAF is trusted by the user and so its security is out of scope of the evaluation. The use of the EAF shall be described and the connection to the TOE shall be stated (for example, over a wired connection or the particular wireless protocol used).

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

Application Note 16: Rules can be: "Always try password authentication first", "Always require both password and EAF authentication", "User chooses which mechanism to use", etc.

FIA_UAU.5/Peer Multiple authentication mechanisms

FIA_UAU.5.1/Peer The TSF shall provide **support to use** [selection: *EXPLICIT ACCEPTANCE OF CONNECTION, QR CODES, NFC LABELS, PINS, USING A COMMON USER ACCOUNT TO A REMOTE SERVICE ON BOTH DEVICES, [assignment: OTHER MECHANISM TO VERIFY THE PEER IDENTITY]*] **for** ~~to support user~~ authentication **to peer devices before allowing any actions on behalf of the user**.

Application Note 17: Explicit acceptance of connection means the user of the device has entered an approval into the interface of the TOE. Use of QR codes, NFC labels and PINs can be in either direction, either generating the invitation to connect or responding. For example the TSF can generate a QR code that can be shared with the peer device, or it could scan the QR code from the peer.

Bluetooth® authentication is handled as part of FTP_ITC_EXT.1/BT, WLAN authentication is handled as part of FTP_ITC_EXT.1/WLAN and their requirements do not need to be duplicated here.

FIA_UAU.5.2/Peer The TSF shall authenticate any ~~user's~~ **peer device's** claimed identity according to the [assignment: *rules describing how the multiple authentication mechanisms provide **peer device** authentication*].

FIA_UAU.6 Re-authenticating

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

- *ATTEMPTED CHANGE OF ANY USER AUTHENTICATION FACTOR;*
- *ATTEMPTED UNLOCKING OF A LOCKED TOE;*
- *[selection: WHEN CHANGING USER ACCOUNTS USING THE CREDENTIALS FOR THE NEW USER, USING NO CREDENTIALS FOR A GUEST USER, [assignment: OTHER CONDITIONS], NO OTHER CONDITIONS]*].

Application Note 18: The selections of "when changing user accounts using the credentials for the new user" or "using no credentials for a guest user" require including the requirements from the PP-Module for Multi-User Support.

FIA_UAU.7 Protected authentication feedback

FIA_UAU.7.1 The TSF shall provide only [*BRIEF FEEDBACK ABOUT THE ENTERED CREDENTIALS*] to the user while the authentication is in progress.

FIA_SOS.1 Verification of secrets

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

- *FOR PASSWORD AND PIN: LENGTH 4 OR MORE FROM A DEFINED CHARACTER SET;*
- *FOR PATTERNS: CONSISTS OF AT LEAST 4 FROM A SET OF AT LEAST 9 AVAILABLE POINTS, WHERE EACH POINT SHALL ONLY BE USED ONCE*].

Application Note 19: The user takes care of the complexity of the PIN/Password: how easy it is to guess for others. See OE.PASSWORD_PIN_PATTERN.

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

FIA_AFL.1 Authentication failure handling

FIA_AFL.1.1 The TSF shall detect when [*AN INTEGER BETWEEN 3 AND 10*] unsuccessful authentication attempts occur related to [~~assignment: list of authentication events~~ selection: *PASSWORD*, *PIN*, *PATTERN*, *EAF*].

FIA_AFL.1.2 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 actions TO REDUCE THE RISK OF ATTACKS SUCH AS BRUTE-FORCE ATTACK]*].

8.3.4 Security Management (FMT)

As the CMD is not a managed device in the enterprise sense, all management is handled locally by the user as opposed to a specific management service as would be the case for an enterprise device.

FMT_MSA.1/Permissions Management of security attributes

FMT_MSA.1.1/Permissions The TSF shall enforce the [*PERMISSIONS POLICY*] to restrict the ability to [selection: ~~*change_default, query*~~, APPROVE PERSISTENTLY, APPROVE TEMPORARILY, REJECT PERSISTENTLY, REJECT TEMPORARILY, MODIFY, ~~*delete*~~ *[assignment: other operations]*] the security attributes [*LIST OF USER OBJECTS*] to [*THE CURRENT USER FOR THEIR OWN PERMISSIONS*].

FMT_MSA.3/Permissions Static attribute initialization

FMT_MSA.3.1/Permissions The TSF shall enforce the [*PERMISSIONS POLICY*] to provide [RESTRICTIVE] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/Permissions The TSF shall allow the [*CURRENT USER FOR THEIR OWN PERMISSIONS*] to specify alternative initial values to override the default values when an object or information is created.

Application Note 20: "restrictive" is chosen here as the default condition is that no permissions should be allowed unless authorized by the user or the TOE manufacturer.

FMT_SMF.1/Authentication Specification of Management Functions

FMT_SMF.1.1/Authentication The TSF shall be capable of performing the following management functions: [~~assignment~~ selection:

- *ENTERING AN INITIAL OR CHANGING (INCLUDING REMOVAL OF) THE KAF;*
- *REGISTRATION OR CHANGING (INCLUDING REMOVAL OF) THE EAF;*
- *SHOW THE LENGTH OF THE CREDENTIAL THE USER IS REQUESTED TO ENTER DURING AUTHENTICATION*].

FMT_SMF.1/Permissions Specification of Management Functions

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

- *VIEW PERMISSIONS GRANTED TO AN APP; AND*
- *GRANT/REVOKE PERMISSION TO/FROM AN APP OR PROCESS TO HAVE READ AND/OR WRITE ACCESS TO A USER OBJECT*].

Application Note 21: User Object is defined in FDP_ACC.2.1/Permissions.

FMT_SMF.1/UserControls Specification of Management Functions

FMT_SMF.1.1/UserControls The TSF shall be capable of performing the following management functions: [

- *GRANT/REVOKE PERMISSION TO/FROM AN APP OR PROCESS TO HAVE ACCESS TO ACCESSIBILITY SERVICE; AND*
- *GRANT/REVOKE PERMISSION TO/FROM AN APP OR PROCESS TO HAVE ACCESS TO DEVICE NOTIFICATION; AND*
- *[selection: CHARGE ONLY MODE BY DEFAULT, FILE TRANSFER MODE, [ASSIGNMENT: OTHER WIRED CHARGING MODE]] WHEN THE TOE IS CONNECTED VIA THE WIRED CHARGING INTERFACE TO ANOTHER DEVICE; AND*
- *[selection: ENABLE/DISABLE REMOVABLE MEDIA ENCRYPTION, NO SUPPORT FOR REMOVABLE MEDIA]; AND*
- [assignment: *list of management functions to be provided by the TSF*]].

Application Note 22: The selection for charging mode is per connection basis, and the "charge only mode" is the default mode when the 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: [~~assignment:~~ **selection:**

- *CHANGE OR RESET THE PRIVACY ALIASES;*
- *BLOCK THE CREATION/USE OF A UNIQUE ID FOR ADVERTISING (NO PERSONALIZED TRACKING)*].

Application Note 23: These aliases are defined in clause 8.3.5.

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 or uninstalling/reinstalling an app automatically generates a new alias).

FMT_SMF.1/APP_Update Specification of Management Functions

FMT_SMF.1.1/APP_Update The TSF shall be capable of performing the following management functions: [~~assignment:~~

- *SPECIFY TO [selection: NOTIFY THE USER WITHOUT DOWNLOADING, NOTIFY THE USER AFTER DOWNLOADING, AUTOMATICALLY INSTALL] WHEN AN AUTOMATIC CHECK TO THE ADP HAS FOUND AN UPDATE; AND*
- *INITIATE AN IMMEDIATE CHECK FOR UPDATE; AND*
- *INITIATE AN UPDATE OF THE APP (IF AVAILABLE); AND*
- *DISPLAY THE VERSION NUMBER OF THE APP; AND*
- *UNINSTALL DOWNLOADED APPS (INCLUDING APPS DOWNLOADED AS PART OF THE SETUP PROCESS)*].

Application Note 24: The selection for setting the automatic update check may provide only a single option such that the user does not have the ability to make a specific choice based on the options available on the TOE.

FMT_SMF.1/SSW_Update Specification of Management Functions

FMT_SMF.1.1/SSW_Update The TSF shall be capable of performing the following management functions: [~~assignment:~~

- *SPECIFY TO [selection: NOTIFY THE USER WITHOUT DOWNLOADING, NOTIFY THE USER AFTER DOWNLOADING, AUTOMATICALLY INSTALL] WHEN AN AUTOMATIC CHECK TO THE TRUSTWORTHY UPDATE SOURCE HAS FOUND AN UPDATE; AND*
- *INITIATE AN IMMEDIATE CHECK FOR UPDATE; AND*
- *INITIATE AN UPDATE OF THE SYSTEM SOFTWARE (IF AVAILABLE); AND*
- *PROVIDE THE STATUS OF THE UPDATE PROCESS AND THE RESULTS OF THE UPDATE; AND*
- *[selection: DELAY TEMPORARILY, BLOCK PERSISTENTLY] AUTOMATIC INSTALLATION OF SYSTEM SOFTWARE UPDATES; AND*
- *DISPLAY THE VERSION NUMBER OF THE SYSTEM SOFTWARE*].

Application Note 25: The selection for setting the automatic update check may provide only a single option such that the user does not have the ability to make a specific choice based on the options available on the TOE.

8.3.5 Privacy (FPR)

To avoid tracking of the unique Device ID of the TOE, the TOE provides aliases of the Device ID for App developers and ad networks. App developers may be able to generate their own aliases on top of ones provided by the TOE itself (these are out of scope) or request the TSF to provide an unique alias for its use. The TOE may provide unique aliases for different purposes or a single global aliases as needed. The user can reset these aliases to prevent tracking to continue indefinitely (see FMT_SMF.1/Privacy).

FPR_PSE.1/Advertisers Pseudonymity

FPR_PSE.1.1/Advertisers The TSF shall ensure that [*AD NETWORKS*] are unable to ~~determine~~ **access** the ~~real user name~~ **Device ID** bound to [*THE TOE (HARDWARE PLATFORM)*] **according to the Permissions Policy**.

FPR_PSE.1.2/Advertisers The TSF shall be able to provide [*AT LEAST ONE UNIQUE*] alias**(**es**)** of the ~~real user name~~ **Device ID** to [*AD NETWORKS*].

FPR_PSE.1.3/Advertisers The TSF shall [DETERMINE AN ALIAS FOR A DEVICE ID]~~ and verify that it conforms to the [assignment: *alias metric*]~~.

FPR_PSE.1/APP_Dev Pseudonymity

FPR_PSE.1.1/APP_Dev The TSF shall ensure that [*APP DEVELOPERS*] are unable to ~~determine~~ **access** the ~~real user name~~ **Device ID** bound to [*THE TOE (HARDWARE PLATFORM)*] **according to the Permissions Policy**.

FPR_PSE.1.2/APP_Dev The TSF shall be able to provide [*AT LEAST ONE UNIQUE*] alias**(**es**)** of the ~~real user name~~ **Device ID** to [*EACH APP DEVELOPER*].

FPR_PSE.1.3/APP_Dev The TSF shall [PROVIDE AN ALIAS FOR A DEVICE ID]~~ and verify that it conforms to the [assignment: *alias metric*]~~ **upon request by the App developer**.

8.3.6 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_EXT.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/ASYMMETRIC; 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 ~~the SFRs are always enforced~~ **it is impossible to read or modify this data and/or key(s)**.

Application Note 26: This requirement is on physical attacks only, logical attacks are addressed by the ADV_ARC and AVA_VAN assurance requirements.

The phrase "responding automatically" means both active detection of attacks and passive resistance to attacks. Active detection of an attack and taking appropriate protective action when detected is similar to an alarm system detecting an attack (tamper responsive). Passive resistance to an attack can be 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.

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

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_TOE_SOFTWARE OPERATION IN FDP_ACF.1/SSW_UPDATE*].

Application Note 27: 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 **and integrity verification** [DURING INITIAL START-UP] to demonstrate the correct operation of [[selection: *[assignment: parts of TSF], the TSF*] BY SELF TESTS AND THE BOOTLOADER, MAIN OS KERNEL [SELECTION: SEE, *[ASSIGNMENT: PARTS OF TSF]*] BY INTEGRITY, WHERE INTEGRITY IS VERIFIED BY [SELECTION: *A DIGITAL SIGNATURE USING AN IMMUTABLE HARDWARE ASYMMETRIC KEY, AN IMMUTABLE HARDWARE HASH OF AN ASYMMETRIC KEY, AN IMMUTABLE HARDWARE HASH, A DIGITAL SIGNATURE USING A HARDWARE-PROTECTED ASYMMETRIC KEY]*].

Application Note 28: 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, while mandating specific methods for verifying integrity. The ST author can choose which mechanisms are to be tested and complete FPT_TST.1.1 accordingly. The bootloader and main OS kernel shall have some type of test (normally integrity verification).

FPT_TST.1.2 The TSF shall provide authorized users with the capability to verify the integrity [*NO DATA*].

FPT_TST.1.3 The TSF shall provide authorized users with the capability to verify the integrity of [*NONE*].

FPT_RCV.2 Automated recovery

FPT_RCV.2.1 When automated recovery from [*DETECTION OF A MALEVOLENT PERSISTENT PRESENCE BY FPT_TST.1 OR AN UPDATE FAILURE BY FDP_ACF.1/SSW_UPDATE*] 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 OR AN UPDATE FAILURE BY FDP_ACF.1/SSW_UPDATE*], the TSF shall ensure the return of the TOE to a secure state using automated procedures.

Application Note 29: 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 user to remove it manually, for example by a factory reset or download of an update.

8.3.7 Trusted Path/Channels (FTP)

A TOE supports many different communication channels, conforming to different standards, and within these standards, using different settings, resulting in different levels of security. While there may be additional channels, the four below are considered to be the most fundamental to a mobile device (outside cellular mobile communications which are out of scope). These cover Bluetooth and WLAN wireless communications as well as software functions to ensure secure communications between applications/services on the device and servers hosted elsewhere.

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_EXT.1/BT Inter-TSF trusted channel

FTP_ITC_EXT.1.1/BT The TSF shall use [*BLUETOOTH® CORE SPECIFICATION THAT CONFORMS TO [selection: V4.1 [11], V4.2 [12], V5.0 [13], V5.1 [14], V5.2 [15], [assignment: A VERSION HIGHER THAN V5.2]]*] to 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_EXT.1.2/BT The TSF shall permit [selection: *the TSF, another trusted IT product*] to initiate communication via the trusted channel.

FTP_ITC_EXT.1.3/BT The TSF shall initiate communication via the trusted channel for [*CONNECTIONS TO BLUETOOTH DEVICES*].

FTP_ITC_EXT.1.4/BT The protocol used by the communications channel shall support the following requirements: [

- *REQUIRE EXPLICIT USER AUTHORIZATION 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]]*].

FTP_ITC_EXT.1/HTTPS Inter-TSF trusted channel

FTP_ITC_EXT.1.1/HTTPS The TSF shall use [*HTTPS THAT CONFORMS TO IETF RFC 2818 [5]*] to 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_EXT.1.2/HTTPS The TSF shall permit [THE TSF] to initiate communication via the trusted channel.

FTP_ITC_EXT.1.3/HTTPS The TSF shall initiate communication via the trusted channel for [*COMMUNICATION WITH A TRUSTED IT PRODUCT*].

FTP_ITC_EXT.1.4/HTTPS The protocol used by the communications channel shall support the following requirements: [*USE TLS AS SPECIFIED IN FTP_ITC_EXT.1/TLS TO IMPLEMENT HTTPS*].

FTP_ITC_EXT.1/TLS Inter-TSF trusted channel

FTP_ITC_EXT.1.1/TLS The TSF shall use [*TLS THAT CONFORMS TO [selection: TLS V1.2 [6], TLS V1.3 [*10*], [ASSIGNMENT: A VERSION OF TLS HIGHER THAN V1.2 [6]]]*] to 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_EXT.1.2/TLS The TSF shall permit [selection: *the TSF, another trusted IT product*] to initiate communication via the trusted channel.

FTP_ITC_EXT.1.3/TLS The TSF shall initiate communication via the trusted channel for [*COMMUNICATION WITH A TRUSTED IT PRODUCT*].

FTP_ITC_EXT.1.4/TLS The protocol used by the communications channel shall support the following requirements: [

- *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 [7]; 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*
- *SUPPORTS THE FOLLOWING CIPHER SUITES [selection:*
  - *TLS_RSA_WITH_AES_256_GCM_SHA384 (IETF RFC 5288 [8]);*
  - *TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (IETF RFC 5288 [8]);*
  - *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (IETF RFC 5289 [9]);*
  - *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (IETF RFC 5289 [9]);*
  - *TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (IETF RFC 5289 [9]);*
  - *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (IETF RFC 5289 [9]);*
  - *[assignment: OTHER PUBLISHED TLS CIPHER SUITE]]*

].

Application Note 30: This list of cipher suites is not meant to be exhaustive. The assignment provides flexibility to support alternative cipher suites as needed in specific markets.

FTP_ITC_EXT.1/WLAN Inter-TSF trusted channel

FTP_ITC_EXT.1.1/WLAN The TSF shall use [*WLAN THAT CONFORMS TO 802.11-2016 [16]*] to 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_EXT.1.2/WLAN The TSF shall permit [selection: *the TSF, another trusted IT product*] to initiate communication via the trusted channel.

FTP_ITC_EXT.1.3/WLAN The TSF shall initiate communication via the trusted channel for [*COMMUNICATION WITH THE TRUSTED IT PRODUCT THROUGH WLAN CHANNEL*].

FTP_ITC_EXT.1.4/WLAN The protocol used by the communications channel shall support the following requirements: [

- *GENERATE SYMMETRIC KEYS ACCORDING TO [selection: PRF-384 WITH KEY LENGTH 128 BIT, PRF-704 WITH KEY LENGTH 256 BIT]; AND*
- *USES [selection: TLS V1.2 [6], TLS V1.3 [10], [assignment: A VERSION OF TLS HIGHER THAN V1.2 [6]]]; AND*
- *SUPPORTS THE FOLLOWING CIPHER SUITES [selection:*
  - *TLS_RSA_WITH_AES_256_GCM_SHA384 (IETF RFC 5288 [8]);*
  - *TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (IETF RFC 5288 [8]);*
  - *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (IETF RFC 5289 [9]);*
  - *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (IETF RFC 5289 [9]);*
  - *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (IETF RFC 5289 [9]);*
  - *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (IETF RFC 5289 [9]);*
  - *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (IETF RFC 5289 [9]);*
  - *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (IETF RFC 5289 [9]);*
  - *TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (IETF RFC 5289 [9]);*
  - *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (IETF RFC 5289 [9])];*
- *RANDOMLY GENERATE A NEW MAC ADDRESS EACH TIME IT CONNECTS TO A DIFFERENT ACCESS POINT*].

8.4 Security assurance requirements

The security assurance requirements consist of EAL2 augmented with ALC_DVS_EXT.1 and ALC_FLR.3 [3] as defined here:

ALC_DVS_EXT.1 Identification of security measures for device identifiers

ALC_DVS_EXT.1.1D The developer shall produce and provide development security documentation on the generation and protection of signing keys and device-unique identifiers.

ALC_DVS_EXT.1.1C The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the following manufacturing components:

- keys used to sign the publicly released system software and its updates and to ensure the use of the proper keys in the build process;
- unique, non-modifiable identifiers (such as IMEI, attestation keys or Device Unique Keys) are properly created and provisioned for each device.

ALC_DVS_EXT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_DVS_EXT.1.2E The evaluator shall review the provided evidence that the security measures are being applied.

ALC_FLR.3 Systematic flaw remediation (refined)

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

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 may 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.

8.5 Security requirements rationale

8.5.1 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 with ALC_DVS_EXT.1 (extended) because a major part of securing CMDs is to ensure that the TOE software that will be deployed to the devices is authorized by the developer. Proper key management includes ensuring keys are generated and stored properly as well as ensuring that the proper keys are used when production devices are produced. ALC_DVS_EXT.1 (extended) requires the TOE manufacturer to:

- ensure keys are properly generated and stored;
- ensure proper access (physical and logical) to signing keys and how they are used in the build process;
- ensure proper access (physical and logical) to the generation of device unique identifiers.

EAL2 is augmented by ALC_FLR.3 (refined) 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 (refined) provides assurance that the TOE will be maintained and supported in the future, and require the TOE manufacturer 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.

8.5.2 The SFRs meet all the security objectives for the TOE

Security Objective	Rationale
O.PROTECT_COMMS	This objective is achieved by FTP_ITC_EXT.1/BT, FTP_ITC_EXT.1/HTTPS, FTP_ITC_EXT.1/TLS and FTP_ITC_EXT.1/WLAN which set up secure channels with authentication and protection from modification and disclosure.
O.AUTHENTICATED_UPDATES	These objectives are achieved by: • FDP_ACC.1/APP_Update, FDP_ACF.1/APP_Update and FDP_ACC.2/SSW_Update and FDP_ACF.1/SSW_Update specify the policies for updating. • FDP_UPF_EXT.1/APP_Update and FDP_UPF_EXT.1/SSW_Update specify the notification frequency for updating. • FCS_COP.1/SigGen specifying the cryptographic mechanism for checking the validity of an update. • FMT_SMF.1/APP_Update and FMT_SMF.1/SSW_Update, specifying that user can initiate an update. • FPT_FLS.1, specifying that failure to correctly update will not lead to an insecure state.

Security Objective	Rationale
O.PROTECT_ASSETS_AT_REST **O.DATA_CLASSIFICATION**	These objectives are achieved by: • FDP_ACC.1/UserDataAsset and FDP_ACF.1/UserDataAsset showing the three classes of user data assets, and when each class can be decrypted. • FCS_COP.1/Symmetric, specifying how they are encrypted and decrypted. • FSC_CKH_EXT.1/Low, FCS_CKH_EXT.1/MediumHigh 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 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.2/Permissions, FDP_ACF.1/Permissions, specifying that user permission is needed and FMT_SMF.1/Permissions, FMT_MSA.1/Permissions allowing users to provide and revoke such permission. FMT_MSA.3/Permissions that the default permission is no access until granted by the user. • FMT_SMF.1/UserControls that the user can control additional privileges around functions which can allow access to user data. • FPR_PSE.1 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.1/Privacy allows users to reset the alias.
O.SECURE_BOOT	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 user) when a malevolent presence is found.
O.AUTHENTICATE_USER	This objective is achieved by: • FMT_SMF.1/Authentication specifying that users can register their authentication data and change this later. • FIA_UAU.1 specifying that each user can only perform limited actions before being authenticated and is authenticated to gain full access. • FIA_UID.1 specifying that each user can only perform limited actions before being identified (via authentication) to gain full access. • FIA_UAU.6 listing the conditions under which a user is to be re-authenticated. • FIA_UAU.5/Local listing the multiple authentication mechanism a TOE has, and the rules for using these. • FIA_SOS.1 listing the minimum quality requirements for authentication (password/PIN length). • FIA_AFL.1 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 FCS_CKM.1/Asymmetric, FCS_CKM.1/Symmetric, FCS_COP.1/SigGen, FCS_COP.1/KeyEst, FCS_COP.1/Symmetric, FCS_COP.1/Derivation, FCS_COP.1/Hash and FCS_COP.1/KeyedHash by ensuring properly vetted cryptographic algorithms are used.
O.RANDOMS	This objective is achieved by FCS_RNG_EXT.1 defining a random generator, whose output meets stringent international standards.
O.AUTHENTICATE_PEER_DEVICE	This objective is achieved by FIA_UAU.5/Peer 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 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.2/Permissions and FDP_ACF.1/Permissions defining security attribute based access control for apps; and • FMT_SMF.1/Permissions specifying management functions for apps.

8.5.3 Dependency analysis

SFR	Dependency	Rationale
FCS_RNG_EXT.1	-	
FCS_CKM.1/Asymmetric	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4	Fulfilled by FCS_CKM.2, to establish key hierarchy. Fullfilled by FCS_CKM.4.
FCS_CKM.1/Symmetric	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4	Fulfilled by FCS_COP.1/Symmetric, to encrypt and decrypt data. Fullfilled by FCS_CKM.4.
FCS_COP.1/SigGen	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	Fulfilled by FCS_CKM.1/Asymmetric. Fulfilled by FCS_CKM.4 for most cases, though some keys (such as the system software update key) are never destroyed as they are stored in write-once memory.
FCS_COP.1/AKeyGen	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	Fulfilled by FCS_CKM.1/Asymmetric. Fulfilled by FCS_CKM.4.
FCS_COP.1/KeyEst	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	Fulfilled by FCS_CKM.1/Asymmetric. Fulfilled by FCS_CKM.4.
FCS_COP.1/Symmetric	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	Fulfilled by FCS_CKM.1/Symmetric. Fulfilled by FCS_CKM.4.
FCS_COP.1/Derivation	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	Fulfilled by FCS_CKM.1/Symmetric to generate key seed, and then derive keys. Fulfilled by FCS_CKM.4.
FCS_COP.1/Hash	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	Not fulfilled. There is no key involved for this operation.
FCS_COP.1/KeyedHash	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	Fulfilled by FCS_CKM.1/Symmetric. Fulfilled by FCS_CKM.4.
FCS_CKH_EXT.1/Low	[FCS_CKM.1 or FCS_COP.1] FCS_CKM.4	Fulfilled by FCS_CKM.1/Symmetric, FCS_COP.1/Symmetric. Fulfilled by FCS_CKM.4.
FCS_CKH_EXT.1/MediumHigh	[FCS_CKM.1 or FCS_COP.1] FCS_CKM.4	Fulfilled by FCS_CKM.1/Symmetric, FCS_CKM.1/Asymmetric, FCS_COP.1/Symmetric, FCS_COP.1/Asymmetric. Fulfilled by FCS_CKM.4.

SFR	Dependency	Rationale
FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	Fulfilled by FCS_CKM.1/Asymmetric and FCS_CKM.1/Symmetric.
FDP_ACC.1/APP_Update	FDP_ACF.1	Fulfilled by FDP_ACF.1/APP_Update.
FDP_ACF.1/APP_Update	FDP_ACC.1 FMT_MSA.3	Fulfilled by FDP_ACC.1/APP_Update. Not necessary, there are no security attributes that can be managed by the user.
FDP_UPF_EXT.1/APP_Update	-	
FDP_ACC.2/SSW_Update	FDP_ACF.1	Fulfilled by FDP_ACF.1/SSW_Update.
FDP_ACF.1/SSW_Update	FDP_ACC.1 FMT_MSA.3	Fulfilled by FDP_ACC.2/SSW_Update. Not necessary, there are no security attributes that can be managed by the user.
FDP_UPF_EXT.1/SSW_Update	-	
FDP_ACC.2/Permissions	FDP_ACF.1	Fulfilled by FDP_ACF.1/Permissions.
FDP_ACF.1/Permissions	FDP_ACC.1 FMT_MSA.3	Fulfilled by FDP_ACC.2/Permissions. Fulfilled by FMT_MSA.3/Permissions.
FDP_ACC.1/UserDataAsset	FDP_ACF.1	Fulfilled by FDP_ACF.1/UserDataAsset.
FDP_ACF.1/UserDataAsset	FDP_ACC.1 FMT_MSA.3	Fulfilled by FDP_ACC.1/UserDataAsset. Not necessary, as the security attributes Low, Medium and High do not change.
FIA_UAU.1	FIA_UID.1	Fulfilled by FIA_UID.1.
FIA_UID.1	-	
FIA_UAU.5/Local	-	
FIA_UAU.5/Peer	-	
FIA_UAU.6	-	
FIA_UAU.7	FIA_UAU.1	Fulfilled by FIA_UAU.1.
FIA_SOS.1	-	
FIA_AFL.1/	FIA_UAU.1	Fulfilled by FIA_UAU.1.
FMT_MSA.1/Permissions	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	Fulfilled by FDP_ACC.2/Permissions (hierarchical to FDP_ACC.1). Only one user is supported. Fulfilled by FMT_SMF.1/Permissions.
FMT_MSA.3/Permissions	FMT_MSA.1 FMT_SMR.1	Fulfilled by FMT_MSA.1/Permissions. Only one user is supported.
FMT_SMF.1/Authentication	-	
FMT_SMF.1/Permissions	-	
FMT_SMF.1/UserControls	-	
FMT_SMF.1/Privacy	-	
FMT_SMF.1/APP_Update	-	
FMT_SMF.1/SSW_Update	-	
FPR_PSE.1	-	
FPT_PHP.3	-	
FPT_FLS.1	-	
FPT_TST.1	-	
FPT_RCV.2	AGD_OPE.1	Fulfilled by EAL2.
FTP_ITC_EXT.1/BT	-	
FTP_ITC_EXT.1/HTTPS	-	
FTP_ITC_EXT.1/TLS	-	
FTP_ITC_EXT.1/WLAN	-	

SFR	Dependency	Rationale
EAL2	Many	All dependencies in an EAL are met.
ALC_DVS_EXT.1	-	-
ALC_FLR.3	-	-

Annex A (informative): Other related specifications

A.1 ETSI EN 303 645

Table A.1 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 table A.1 indicates the approximate relation, and details can differ.

Table A.1

ETSI EN 303 645 [i.1]	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_PATTERN	Passwords are 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.*	Cryptography is required to meet the requirements of ISO/IEC 18033-1 [17] for new algorithms	
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.*	FIA_AFL.1 provides protection for local brute force attacks	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 [i.1]	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/APP_Update FDP_ACF.1/SSW_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/APP_Update FDP_ACF.1/SSW_Update	
Provision 5.3-3 *An update shall be simple for the user to apply.*	FMT_SMF.1/APP_Update FMT_SMF.1/SSW_Update	
Provision 5.3-4 *Automatic mechanisms should be used for software updates.*	FDP_ACF.1/APP_Update FDP_UPF_EXT.1/APP_Update FDP_ACF.1/SSW_Update FDP_UPF_EXT.1/SSW_Update	CMD system software usually only updates after explicit user permission, though app updates may be automatic
Provision 5.3-5 *The device should check after initialization, and then periodically, whether security updates are available.*	FDP_UPF_EXT.1/APP_Update FDP_UPF_EXT.1/SSW_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/APP_Update FDP_UPF_EXT.1/APP_Update FMT_SMF.1/SSW_Update FDP_UPF_EXT.1/SSW_Update	This only allows the 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_CKM.1/Asymmetric FCS_COP.1/SigGen ALC_DVS_EXT.1	
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/SSW_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/SSW_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_UPF_EXT.1/APP_Update FDP_UPF_EXT.1/SSW_Update	This does not include informing the 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.*	ALC_FLR.3	Not in scope of present document

ETSI EN 303 645 [i.1]	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_EXT.1/Low FCS_CKH_EXT.1/MediumHigh 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_CKM.1/Asymmetric FCS_COP.1/SigGen ALC_DVS_EXT.1	
Provision 5.5-1 *The consumer IoT device shall use best practice cryptography to communicate securely.*	Cryptography is required to meet the requirements of ISO/IEC 18033-1 [17] for new algorithms	
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.*	Cryptography is required to meet the requirements of ISO/IEC 18033-1 [17] for new algorithms	
Provision 5.5-3 *Cryptographic algorithms and primitives should be updateable.*	FDP_ACF.1/SSW_Update	Some low-level capabilities cannot be changed (for example the signature checks for the system software update), but main OS functionality can be updated
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.*	FIA_UAU.5/Peer FTP_ITC_EXT.1/WLAN FTP_ITC_EXT.1/TLS FTP_ITC_EXT.1/BT FTP_ITC_EXT.1/HTTPS	

ETSI EN 303 645 [i.1]	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.*	FIA_UAU.5/Peer FTP_ITC_EXT.1/WLAN FTP_ITC_EXT.1/TLS FTP_ITC_EXT.1/BT FTP_ITC_EXT.1/HTTPS	
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_EXT.1/WLAN FTP_ITC_EXT.1/TLS FTP_ITC_EXT.1/BT FTP_ITC_EXT.1/HTTPS	
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_EXT.1/WLAN FTP_ITC_EXT.1/TLS FTP_ITC_EXT.1/BT FTP_ITC_EXT.1/HTTPS	
Provision 5.5-8 *The manufacturer shall follow secure management processes for critical security parameters that relate to the device.*	ALC_DVS_EXT.1	
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_ACC.2/Permissions 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 [i.1]	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_EXT.1/WLAN FTP_ITC_EXT.1/TLS FTP_ITC_EXT.1/BT FTP_ITC_EXT.1/HTTPS	
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_EXT.1/WLAN FTP_ITC_EXT.1/TLS FTP_ITC_EXT.1/BT FTP_ITC_EXT.1/HTTPS	
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	
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.*	-	A requirement for the service provider, not for the CMD itself
Provision 5.11-4 *Users should be provided with clear confirmation that personal data has been deleted from services, devices and applications.*	-	A requirement for the service provider, not for the CMD itself
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 [i.1]	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.*	-	A requirement for the provider, App developer, advertiser etc. Not for the CMD itself
Provision 6-3 *Consumers who gave consent for the processing of their personal data shall have the capability to withdraw it at any time.*	-	A requirement for the provider, App developer, advertiser etc. Not for the CMD itself
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 manufacturer 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 manufacturer 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 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 manufacturer to check whether appropriate implementation is in place. However, these are not typical security requirements from CC perspective.

A.2 SESIP

Table A.2 shows the correspondence between SESIP [i.4] 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.4].

Table A.2

The present Document	SESIP	Comments
FCS_RNG_EXT.1	Cryptographic Random Number Generation	
FCS_CKM.1/* FCS_CKM.2 FCS_COP.1/* FCS_CKH_EXT.1/*	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/APP_Update FDP_ACF.1/APP_Update FDP_UPF_EXT.1/App_Update FDP_ACC.2/SSW_Update FDP_ACF.1/SSW_Update FDP_UPF_EXT.1/SSW_Update FPT_FLS.1	Secure Update of Platform Secure Update of Application	
FDP_ACC.2/Permissions FDP_ACF.1/Permissions FMT_SMF.1/Permissions	Software Attacker Resistance: Isolation of Platform Software Attacker Resistance: Isolation of Platform Parts	
FDP_ACC.1/UserDataAsset FDP_ACF.1/UserDataAsset	Secure Cryptographic Storage	
FIA_*		No equivalent, as authentication is outside the scope of SESIP (IoT platforms).
FMT_SMF.1/Privacy FPR_PSE.1		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_EXT.1/WLAN FTP_ITC_EXT.1/TLS FTP_ITC_EXT.1/BT FTP_ITC_EXT.1/HTTPS	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 [4]. 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	Radio interface(s)	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	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 as ETSI TS 102 732
V2.1.1	October 2023	Publication
V2.1.2	November 2023	Publication