Requirements from the

Mobile Device Fundamentals

Version: 3.2

2020-01-07

National Information Assurance Partnership

Revision History

Version	Date	Comment
1.0	2013-10-21	Initial Release
1.1	2014-01-12	Typographical changes and additional clarifications in application notes. Removed assignment from FCS_TLS_EXT.1 and limited testing to those ciphersuites in both FCS_TLS_EXT.1 and FCS_TLS_EXT.2.
2.0	2015-09-14	Included changes based on Technical Rapid Response Team Decisions. Clarified many requirements and assurance activities. Mandated objective requirements: • Application Access Control (FDP_ACF_EXT.1.2) • VPN Information Flow Control (FDP_IFC_EXT.1) Added new objective requirements: • Suite B cryptography for IEEE 802.11 • Certificate enrollment • Protection of additional key material types • Heap overflow protection • Bluetooth requirements • Cryptographic operation services for applications • Remote Attestation (FPT_NOT_EXT.1) Added transition dates for some objective requirements. Included hardware-isolated REK and key storage selections. Allowed key derivation by REK. Clarified FTP_ITC_EXT.1 and added FDP_UPC_EXT.1. Mandated HTTPS and TLS for application use. (FDP_UPC_EXT.1) Removed Dual_EC_DRBG as an approved DRBG. Adopted new TLS requirements. Mandated TSF Wipe upon authentication failure limit and required number of authentication failures be maintained across reboot. Clarified Management Class. Included more domain isolation discussion and tests. Updated Audit requirements and added Auditable Events table. Added SFR Category Mapping Table. Updated Use Case Templates. Moved Glossary to Introduction.
3.0	2015-09-17	Included changes based on Technical Rapid Response Team Decisions. Clarified many requirements and assurance activities. Mandated objective requirements: • Generation of Audit Records (FAU_GEN.1) • Audit Storage Protection (FAU_STG.1) • Audit Storage Overwrite (FAU_STG.4) • Lock Screen DAR (FDP_DAR_EXT.2) • Discard Bluetooth Connection Attempts from Bluetooth Addresses with Existing Connection (FIA_BLT_EXT.3) • JTAG Disablement (FPT_JTA)

		Added new objective requirements: • Application Backup • Biometric Authentication Factor • Access Control • User Authentication • Bluetooth Encryption WLAN client requirements moved to Extended Package for WLAN Client. Added SFRs to support BYOD Use Case BYOD Use Case Updated key destruction SFR
3.1	2017-04-05	Included changes based on Technical Rapid Response Team Decisions and incorporated Technical Decisions. Modified biometric requirements: • FIA_UAU.5 - Added iris, face, voice and vein as supported modalities, in addition to fingerprint (allowed in version 3) • FIA_BMG_EXT.1.1 - Clarified AA to specify that vendor evidence is acceptable and expectations of evidence provided. • FIA_BMG_EXT.1.2 - SAFAR was changed to an assignment of a SAFAR no greater than 1:500. • FIA_AFL_EXT.1 - Updated to allow each biometric modality to utilize an individual or shared counter. FCS_TLSC_EXT.1.1 - Removed TLS ciphersuites that utilized SHA1 and updated optional ciphersuites to be uniformed across PPs. FCS_STG_EXT.2.2 - Modified to require long term trusted channel key material be encrypted by an approved method. FIA_UAU_EXT.1.1 - Modified to allow the long term trusted channel key material to be available prior to password being entered at start-up.
3.2	2019-03-01	Removed TLS SFRs and utilized TLS Functional Package Removed Bluetooth SFRs and utilized Bluetooth Module. Bluetooth SFR moved to Implementation Dependent. FPT_TUD_EXT.4.2 renumbered to FPT_TUD_EXT.5.1

Introduction

Purpose. This document presents the functional and assurance requirements found in the *Mobile Device Fundamentals*. Common Criteria evaluation, facilitated in the U.S. by the National Information Assurance Partnership (NIAP), is required for IA and IA-enabled products in National Security Systems according to CNSS Policy #11.

Using this document. This representation of the Protection Profile includes:

- *Security Functional Requirements* for use in evaluation. These are featured without the formal Assurance Activities specified in the Protection Profile, to assist the reader who is interested only in the requirements.

  It also includes, in tables shown later, particular types of security functional requirements that are not strictly required in all cases. These are:

  - *Selection-based Security Functional Requirements* which become required when certain selections are made inside the regular Security Functionality Requirements (as indicated by the **[selection:]** construct).
  - *Objective Security Functional Requirements* which are highly desired but not yet widely-available in commercial technology.
  - *Optional Security Functional Requirements* which are available for evaluation and which some customers may insist upon.

- *Security Assurance Requirements* which relate to developer support for the product under evaluation, development processes, and other non-functionality security relevant requirements.

Security Functional Requirements

Audit Data Generation

The TSF shall be able to generate an audit record of the following auditable events:

1. Start-up and shutdown of the audit functions

2. All auditable events for the [not selected] level of audit
3. All administrative actions
4. Start-up and shutdown of the Rich OS
5. Insertion or removal of removable media
6. Specifically defined auditable events in
7. [**selection**: *Audit records reaching [**assignment**: integer value less than 100] percentage of audit capacity, [**assignment**: other auditable events derived from this profile]*]
8. [**selection**: *Specifically defined auditable event in , no additional auditable events*]

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	
FAU_STG.1	None.	
FAU_STG.4	None.	
FCS_CKM_EXT.1	[**selection**: *generation of a REK*, *None*].	No additional information.
FCS_CKM_EXT.2	None.	
FCS_CKM_EXT.3	None.	
FCS_CKM_EXT.4	None.	
FCS_CKM_EXT.5	[**selection**: *Failure of the wipe*, *None*].	No additional information.
FCS_CKM_EXT.6	None.	
FCS_CKM.1	[**selection**: *Failure of key generation activity for authentication keys*, *None*].	No additional information.
FCS_CKM.2/UNLOCKED	None.	
FCS_CKM.2/LOCKED	None.	
FCS_COP.1/ENCRYPT	None.	
FCS_COP.1/HASH	None.	
FCS_COP.1/SIGN	None.	
FCS_COP.1/KEYHMAC	None.	
FCS_COP.1/CONDITION	None.	
FCS_IV_EXT.1	None.	
FCS_SRV_EXT.1	None.	
FCS_STG_EXT.1	Import or destruction of key.	Identity of key. Role and identity of requestor.
	[**selection**: *Exceptions to use and destruction rules*, *No other events*]	
FCS_STG_EXT.2	None.	
FCS_STG_EXT.3	Failure to verify integrity of stored key.	Identity of key being verified.
FDP_DAR_EXT.1	[**selection**: *Failure to encrypt/decrypt data*, *None*].	No additional information.
FDP_DAR_EXT.2	Failure to encrypt/decrypt data.	No additional information.
FDP_IFC_EXT.1	None.	
FDP_STG_EXT.1	Addition or removal of certificate from Trust Anchor Database.	Subject name of certificate.
FIA_PMG_EXT.1	None.	
FIA_TRT_EXT.1	None.	
FIA_UAU_EXT.1	None.	
FIA_UAU.5	None.	

FIA_UAU.7	None.	
FIA_X509_EXT.1	Failure to validate X.509v3 certificate.	Reason for failure of validation.
FMT_MOF_EXT.1	None.	
FPT_AEX_EXT.1	None.	
FPT_AEX_EXT.2	None.	
FPT_AEX_EXT.3	None.	
FPT_JTA_EXT.1	None.	
FPT_KST_EXT.1	None.	
FPT_KST_EXT.2	None.	
FPT_KST_EXT.3	None.	
FPT_NOT_EXT.1	[**selection**: *Measurement of TSF software*, *None*].	[**selection**: *Integrity verification value*, *No additional information*].
FPT_STM.1	None.	
FPT_TST_EXT.1	Initiation of self-test.	[**selection**: *Algorithm that caused the failure*, *none*]
	Failure of self-test.	
	Start-up of TOE.	No additional information.
FPT_TST_EXT.2/PREKERNEL	[**selection**: *Detected integrity violation*, *none*]	[**selection**: *The TSF code file that caused the integrity violation*, *No additional information*]
FPT_TUD_EXT.1	None.	
FTA_SSL_EXT.1	None.	

: Mandatory Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_SAR.1	None.	
FAU_SEL.1	All modifications to the audit configuration that occur while the audit collection functions are operating.	No additional Information.
FCS_CKM_EXT.7	None.	
FCS_DTLS_EXT.1 (TLS Package)	Failure of the certificate validity check.	Issuer Name and Subject Name of certificate.
FCS_HTTPS_EXT.1	Failure of the certificate validity check.	Issuer Name and Subject Name of certificate. [**selection**: *User's authorization decision*, *No additional information*].
FCS_RBG_EXT.1	Failure of the randomization process.	No additional information.
FCS_RBG_EXT.2	None.	
FCS_RBG_EXT.3	None.	
FCS_SRV_EXT.2	None.	
FCS_TLSC_EXT.1 (TLS Package)	Establishment/termination of a TLS session.	Non-TOE endpoint of connection.
	Failure to establish a TLS session.	Reason for failure.
	Failure to verify presented identifier.	Presented identifier and reference identifier.
FCS_TLSC_EXT.2 (TLS Package)	None.	

FCS_TLSC_EXT.3 (TLS Package)	None.	
FDP_ACF_EXT.1	None.	
FDP_ACF_EXT.2	None.	
FDP_ACF_EXT.3	None.	
FDP_BCK_EXT.1	None.	
FDP_PBA_EXT.1	None.	
FDP_UPC_EXT.1/NORMAL	Application initiation of trusted channel.	Name of application. Trusted channel protocol. Non-TOE endpoint of connection.
FDP_UPC_EXT.1/BLUETOOTH	Application initiation of trusted channel.	Name of application. Trusted channel protocol. Non-TOE endpoint of connection.
FIA_AFL_EXT.1	Excess of authentication failure limit.	Authentication factor used.
FIA_BMG_EXT.1	None.	
FIA_BMG_EXT.2	None.	
FIA_BMG_EXT.3	None.	
FIA_BMG_EXT.4	None.	
FIA_BMG_EXT.5	None.	
FIA_BMG_EXT.6	None.	
FIA_UAU_EXT.2	Action performed before authentication.	No additional information.
FIA_UAU.6	User changes Password Authentication Factor.	No additional information.
FIA_UAU_EXT.4	None.	
FIA_X509_EXT.2	Failure to establish connection to determine revocation status.	No additional information.
FIA_X509_EXT.3	None.	
FIA_X509_EXT.4	Generation of Certificate Enrollment Request.	Issuer and Subject name of EST Server. Method of authentication. Issuer and Subject name of certificate used to authenticate. Content of Certificate Request Message.
	Success or failure of enrollment.	Issuer and Subject name of added certificate or reason for failure.
	Update of EST Trust Anchor Database	Subject name of added Root CA.
FIA_X509_EXT.5	None.	
FMT_SMF_EXT.1	[**selection**: *Initiation of policy update*, *none*].	[**selection**: *Policy name*, *none*].
	[**selection**: *Change of settings*, *none*]	[**selection**: *Role of user that changed setting*, *Value of new setting*, *none*].
	[**selection**: *Success of failure of function*, *none*]	[**selection**: *Role of user that performed function*, *Function performed*, *Reason for failure, none*].
	Initiation of software update.	Version of update.
	Initiation of application installation or update.	Name and version of application.
FMT_SMF_EXT.2	[**selection**: *Unenrollment*, *Initiation of unenrollment*, *none*]	[**selection**: *Identity of administrator Remediation action performed*, *failure of accepting command to unenroll*, *none*]
FMT_SMF_EXT.3	None.	

FPT_AEX_EXT.4	None.	
FPT_AEX_EXT.5	None.	
FPT_AEX_EXT.6	None.	
FPT_AEX_EXT.7	None.	
FPT_BBD_EXT.1	None.	
FPT_BLT_EXT.1	None.	
FPT_NOT_EXT.2	None.	
FPT_TST_EXT.2/POSTKERNEL	[**selection**: *Detected integrity violation*, *none*]	[**selection**: *The TSF code file that caused the integrity violation*, *No additional information*]
FPT_TST_EXT.3	None.	
FPT_TUD_EXT.2	Success or failure of signature verification for software updates.	No additional information.
	Success or failure of signature verification for applications.	No additional information.
FPT_TUD_EXT.3	None.	
FPT_TUD_EXT.4	None.	
FPT_TUD_EXT.5	None.	
FTA_TAB.1	None.	
FTP_ITC_EXT.1	Initiation and termination of trusted channel.	Trusted channel protocol. Non-TOE endpoint of connection.

: Additional Auditable Events

Application Note: Administrator actions are defined as functions labeled as mandatory for FMT_MOF_EXT.1.2 (i.e. ‘M-MM’ in). If the TSF does not support removable media, number 4 is implicitly met.

The TSF shall generate an audit record for all events contained in . Generating audit records for events in is currently objective. It is acceptable to include individual SFRs from in the ST, without including the entirety of .

Application Note:
FPT_TST_EXT.1 – Audit of self-tests is required only at initial start-up. Since the TOE "transitions to non-operational mode" upon failure of a self-test, per FPT_NOT_EXT.1, this is considered equivalent evidence to an audit record for the failure of a self-test.

FDP_DAR_EXT.1 - "None" shall be selected, if the TOE utilizes whole volume encryption for protected memory, since it is not feasible to audit when the encryption/decryption fails. If the TOE utilizes file-based encryption for protected data and audits when this encryption/decryption fails, then that auditable event shall be selected.

Application Note:
If the audit event for FMT_SMF_EXT.1 is included in the ST, it is acceptable for the initiation of the software update to be audited without indicating the outcome (success or failure) of the update.

The TSF shall record within each audit record at least the following information:

1. Date and time of the event
2. Type of event
3. Subject identity
4. The outcome (success or failure) of the event
5. Additional information in
6. [**selection**: *Additional information in* , *no additional information*]

Application Note: The subject identity is usually the process name/ID. The event type is often indicated by a severity level, for example, ‘info’, ‘warning’, or ‘error’.

If "no additional auditable events" is selected in the second selection of FAU_GEN.1.1, then "no additional information" shall be selected.

For each audit event selected from in FAU_GEN.1.1 if additional information is required to be recorded within the audit record, it should be included in this selection.

Audit Storage Protection

The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.

The TSF shall be able to prevent unauthorized modifications to the stored audit records in the audit trail.

Prevention of Audit Data Loss

The TSF shall overwrite the oldest stored audit records if the audit trail is full.

Cryptographic key generation

The TSF shall generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm [**selection**:

- *RSA schemes using cryptographic key sizes of 2048-bit or greater that meet FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3,*
- *ECC schemes using [**selection**:*
  - *"NIST curves" P-384 and [**selection**: P-256, P-521, no other curves] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4,*
  - *Curve25519 schemes that meet the following: RFC 7748*

  ] ,
- *FFC schemes using [**selection**:*
  - *cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1,*
  - *Diffie-Hellman group 14 that meet the following: RFC3526, Section 3,*
  - *"safe-prime" groups that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"', and [**selection**: RFC 3526, RFC 7919]*

  ]

].

Application Note: The ST author shall select all key generation schemes used for key establishment and entity authentication. When key generation is used for key establishment, the schemes in FCS_CKM.2/UNLOCKED and selected cryptographic protocols must match the selection. When key generation is used for entity authentication, the public key may be associated with an X.509v3 certificate.

If the TOE acts as a receiver in the RSA key establishment scheme, the TOE does not need to implement RSA key generation.

Curve25519 can only be used for ECDH and in conjunction with FDP_DAR_EXT.2.2.

Cryptographic key establishment

The TSF shall perform cryptographic key establishment in accordance with a specified cryptographic key establishment method [**selection**:

- *RSA-based key establishment schemes that meet the following [**selection**:*
  - *NIST Special Publication 800-56B, “Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography”,*
  - *RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1:RSA Cryptography Specifications Version 2.1"*

  ],
- *Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography",*
- *Finite field-based key establishment schemes that meets the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography",*
- *Key establishment schemes using Diffie-Hellman group 14 that meets the following: RFC 3526, Section 3,*
- *FFC schemes using "safe-prime" groups that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"', and [**selection**: RFC 3526, RFC 7919],*
- *No other schemes*

].

Application Note: The ST author shall select all key establishment schemes used for the selected cryptographic protocols.

The RSA-based key establishment schemes are described in Section 9 of NIST SP 800-56B; however, Section 9 relies on implementation of other sections in SP 800-56B. If the TOE acts as a receiver in the RSA key establishment scheme, the TOE does not need to implement RSA key generation.

The elliptic curves used for the key establishment scheme shall correlate with the curves specified in FCS_CKM.1.1.

The domain parameters used for the finite field-based key establishment scheme are specified by the key generation according to FCS_CKM.1.1.

Cryptographic key establishment

The TSF shall perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [**selection**:

- *RSA-based key establishment schemes that meet the following: NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography"*,
- *Elliptic curve-based key establishment schemes that meets the following: [**selection**:*
  - *NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography",*
  - *RFC 7748, "Elliptic Curves for Security"*

  ] ,
- *Finite field-based key establishment schemes that meets the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"*

] for the purposes of encrypting sensitive data received while the device is locked.

Application Note: The RSA-based key establishment schemes are described in Section 9 of NIST SP 800-56B; however, Section 9 relies on implementation of other sections in SP 800-56B. If the TOE acts as a receiver in the RSA key establishment scheme, the TOE does not need to implement RSA key generation.

The elliptic curves used for the key establishment scheme shall correlate with the curves specified in FCS_CKM.1.1.

The domain parameters used for the finite field-based key establishment scheme are specified by the key generation according to FCS_CKM.1.1.

Extended: Cryptographic Key Support

FEL-MUT-HARD The TSF shall support [**selection**: *immutable hardware*, *mutable hardware*] REK(s) with a [**selection**: *symmetric*, *asymmetric*] key of strength [**selection**: *112 bits*, *128 bits*, *192 bits*, *256 bits*].

Each REK shall be hardware-isolated from Rich OS on the TSF in runtime.

Each REK shall be generated by a RBG in accordance with FCS_RBG_EXT.1.

Application Note: Either asymmetric or symmetric keys are allowed; the ST author makes the selection appropriate for the device. Symmetric keys must be of size 128 or 256 bits in order to correspond with FCS_COP.1/ENCRYPT. Asymmetric keys may be of any strength corresponding to FCS_CKM.1.

The raw key material of "immutable hardware" REK(s) is computationally processed by hardware and software cannot access the raw key material. Thus if "immutable-hardware" is selected in FCS_CKM_EXT.1.1 it implicitly meets FCS_CKM_EXT.7. If "mutable-hardware" is selected in FCS_CKM_EXT.1.1, FCS_CKM_EXT.7 must be included in the ST.

The lack of a public/documented API for importing or exporting the REK, when a private/undocumented API exists, is not sufficient to meet this requirement.

The RBG used to generate a REK may be a RBG native to the hardware key container or may be an off-device RBG. If performed by an off-device RBG, the device manufacturer shall not be able to access a REK after the manufacturing process has been completed. The assurance activities for these two cases differ.

Extended: Cryptographic Key Random Generation

All DEKs shall be [**selection**:

- *randomly generated*,
- *from the combination of a randomly generated DEK with another DEK or salt in a way that preserves the effective entropy of each factor by [**selection**: using an XOR operation, concatenating the keys and using a KDF (as described in SP 800-108), concatenating the keys and using a KDF (as described in SP 800-56C)]*

] with entropy corresponding to the security strength of AES key sizes of [**selection**: *128*, *256*] bits.

Application Note: The intent of this requirement is to ensure that the DEK cannot be recovered with less work than a full exhaust of the key space for AES. The key generation capability of the TOE uses a RBG implemented on the TOE device (FCS_RBG_EXT.1). Either 128-bit or 256-bit (or both) are allowed; the ST author makes the selection appropriate for the device. A DEK is used in addition to the KEK so that authentication factors can be changed without having to re-encrypt all of the user data on the device.

The ST author selects all applicable DEK generation types implemented by the TOE.

If combined, the ST author shall describe which method of combination is used in order to justify that the effective entropy of each factor is preserved, and the ST author shall describe that each combined value was originally generated from an Approved DRBG described in FCS_RBG_EXT.1

The documentation of the product's encryption key management should be detailed enough that, after reading, the evaluator will thoroughly understand the product's key management and how it meets the requirements to ensure the keys are adequately protected. This documentation should include an essay and diagram(s). This documentation is not required to be part of the TSS - it can be submitted as a separate document and marked as developer proprietary.

SP 800-56C specifies a two-step key derivation procedure that employs an extraction-then-expansion technique for deriving keying material from a shared secret generated during a key establishment scheme. The Randomness Extraction step as described in Section 5 of SP 800-56C is followed by Key Expansion using the key derivation functions defined in SP 800-108 (as described in Section 6 of SP 800-56C).

Extended: Cryptographic Key Generation

The TSF shall use [**selection**:

- *asymmetric KEKs of [**assignment**: security strength greater than or equal to 112 bits] security strength*,
- *symmetric KEKs of [**selection**: 128-bit, 256-bit] security strength corresponding to at least the security strength of the keys encrypted by the KEK*

].

Application Note: The ST author selects all applicable KEK types implemented by the TOE.

The TSF shall generate all KEKs using one of the following methods:

- Derive the KEK from a Password Authentication Factor according to FCS_COP.1.1/CONDITION and

[**selection**:

- *Generate the KEK using an RBG that meets this profile (as specified in FCS_RBG_EXT.1)*,
- *Generate the KEK using a key generation scheme that meets this profile (as specified in FCS_CKM.1)*,
- *Combine the KEK from other KEKs in a way that preserves the effective entropy of each factor by [**selection**: using an XOR operation, concatenating the keys and using a KDF (as described in SP 800-108), concatenating the keys and using a KDF (as described in SP 800-56C), encrypting one key with another]*

].

Application Note: The conditioning of passwords is performed in accordance with FCS_COP.1/CONDITION.

It is expected that key generation derived from conditioning, using an RBG or generation scheme, and through combination, will each be necessary to meet the requirements set out in this document. In particular, has KEKs of each type: KEK_3 is generated, KEK_1 is derived from a Password Authentication Factor, and KEK_2 is combined from two KEKs. In , KEK_3 may either be a symmetric key generated from an RBG or an asymmetric key generated using a key generation scheme according to FCS_CKM.1.

If combined, the ST author shall describe which method of combination is used in order to justify that the effective entropy of each factor is preserved.

The documentation of the product's encryption key management should be detailed enough that, after reading, the evaluator will thoroughly understand the product's key management and how it meets the requirements to ensure the keys are adequately protected. This documentation should include an essay and diagram(s). This documentation is not required to be part of the TSS - it can be submitted as a separate document and marked as developer proprietary.

SP 800-56C specifies a two-step key derivation procedure that employs an extraction-then-expansion technique for deriving keying material from a shared secret generated during a key establishment scheme. The Randomness Extraction step as described in Section 5 of SP 800-56C is followed by Key Expansion using the key derivation functions defined in SP 800-108 (as described in Section 6 of SP 800-56C).

Extended: Key Destruction

The TSF shall destroy cryptographic keys in accordance with the specified cryptographic key destruction methods:

- by clearing the KEK encrypting the target key
- in accordance with the following rules

- For volatile memory, the destruction shall be executed by a single direct overwrite [**selection**: *consisting of a pseudo-random pattern using the TSF's RBG*, *consisting of zeroes*].
- For non-volatile EEPROM, the destruction shall be executed by a single direct overwrite consisting of a pseudo random pattern using the TSF's RBG (as specified in FCS_RBG_EXT.1), followed by a read-verify.
- For non-volatile flash memory, that is not wear-leveled, the destruction shall be executed [**selection**: *by a single direct overwrite consisting of zeros followed by a read-verify*, *by 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-leveled, the destruction shall be executed [**selection**: *by a single direct overwrite consisting of zeros*, *by a block erase*].
- For non-volatile memory other than EEPROM and flash, the destruction shall be executed by a single direct overwrite with a random pattern that is changed before each write.

Application Note: The clearing indicated above applies to each intermediate storage area for plaintext key/cryptographic critical security parameter (i.e. any storage, such as memory buffers, that is included in the path of such data) upon the transfer of the key/cryptographic critical security parameter to another location.

Because plaintext key material is not allowed to be written to non-volatile memory (FPT_KST_EXT.1), the second selection only applies to key material written to volatile memory.

The TSF shall destroy all plaintext keying material and critical security parameters when no longer needed.

Application Note: For the purposes of this requirement, plaintext keying material refers to authentication data, passwords, secret/private symmetric keys, private asymmetric keys, data used to derive keys, values derived from passwords, etc. If a BAF is selected in FIA_UAU.5.1 the enrollment or authentication templates are not subject to this requirement, since templates are not suitable for deriving keying material. However, source biometric data (i.e. fingerprint image or friction ridge pattern), the features an algorithm uses to perform biometric authentication for enrollment or verification (e.g. location of minutia), threshold values used in making the match adjudication, intermediate values calculated while building an enrollment or authentication template (i.e. direction maps, minutia counts, binarized and skeletonized representations of friction ridge patterns, etc.), and final match scores are examples of critical security parameters that must be destroyed when no longer needed.

Key destruction procedures are performed in accordance with FCS_CKM_EXT.4.1.

There are multiple situations in which plaintext keying material is no longer necessary, including when the TOE is powered off, when the wipe function is performed, when trusted channels are disconnected, when keying material is no longer needed by the trusted channel per the protocol, and when transitioning to the locked state (for those values derived from the Password Authentication Factor or that key material which is protected by the password-derived or biometric-unlocked KEK according to FCS_STG_EXT.2 – see). For keys (or key material used to derive those keys) protecting sensitive data received in the locked state, "no longer needed" includes "while in the locked state."

Trusted channels may include TLS, HTTPS, DTLS, IPsec VPNs, Bluetooth BR/EDR, and Bluetooth LE. The plaintext keying material for these channels includes (but is not limited to) master secrets, and Security Associations (SAs).

If REK(s) are processed in a separate execution environment on the same Application Processor as the Rich OS, REK key material must be cleared from RAM immediately after use, and at least, must be wiped when the device is locked, as the REK is part of the key hierarchy protecting sensitive data.

Extended: TSF Wipe

The TSF shall wipe all protected data by [**selection**:

- *Cryptographically erasing the encrypted DEKs and/or the KEKs in non-volatile memory by following the requirements in FCS_CKM_EXT.4.1*,
- *Overwriting all according to the following rules:*
  - *For EEPROM, the destruction shall be executed by a single direct overwrite consisting of a pseudo random pattern using the TSF's RBG (as specified in FCS_RBG_EXT.1, followed by a read-verify.*
  - *For flash memory, that is not wear-leveled, the destruction shall be executed [**selection**: by a single direct overwrite consisting of zeros followed by a read-verify, by a block erase that erases the reference to memory that stores data as well as the data itself].*
  - *For flash memory, that is wear-leveled, the destruction shall be executed [**selection**: by a single direct overwrite consisting of zeros, by a block erase].*
  - *For non-volatile memory other than EEPROM and flash, the destruction shall be executed by a single direct overwrite with a random pattern that is changed before each write.*

].

The TSF shall perform a power cycle on conclusion of the wipe procedure.

Application Note: The ST author shall select which method of wipe the TSF performs.

Extended: Salt Generation

The TSF shall generate all salts using a RBG that meets FCS_RBG_EXT.1.

Application Note: This requirement refers only to salt generation. In the examples given, a salt may be used as part of the scheme/algorithm. Requirements on nonces and/or ephemeral keys are provided elsewhere, if needed. The list below is provided for clarity, in order to give examples of where the TSF may be generating cryptographic salts; it is not exhaustive nor is it intended to mandate implementation of all of these schemes/algorithms. Cryptographic salts are generated for various uses including:

- RSASSA-PSS signature generation
- DSA signature generation
- ECDSA signature generation
- DH static key agreement scheme
- PBKDF
- Key Agreement Scheme in NIST SP 800-56B
- AES GCM

Cryptographic operation

The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm:

- AES-CBC (as defined in FIPS PUB 197, and NIST SP 800-38A) mode
- AES-CCMP (as defined in FIPS PUB 197, NIST SP 800-38C and IEEE 802.11-2012), and
- [**selection**:
  - *AES Key Wrap (KW) (as defined in NIST SP 800-38F)*,
  - *AES Key Wrap with Padding (KWP) (as defined in NIST SP 800-38F)*,
  - *AES-GCM (as defined in NIST SP 800-38D)*,
  - *AES-CCM (as defined in NIST SP 800-38C)*,
  - *AES-XTS (as defined in NIST SP 800-38E) mode*,
  - *AES-CCMP-256 (as defined in NIST SP800-38C and IEEE 802.11ac-2013)*,
  - *AES-GCMP-256 (as defined in NIST SP800-38D and IEEE 802.11ac-2013)*,
  - *no other modes*

  ]

and cryptographic key sizes 128-bit key sizes and [**selection**: *256-bit key sizes*, *no other key sizes*].

Application Note: For the first selection, the ST author should choose the mode or modes in which AES operates. For the second selection, the ST author should choose the key sizes that are supported by this functionality. 128-bit CBC and CCMP are required in order to comply with WLAN Client Extended Package.

Note that to comply with the WLAN Client EP, AES CCMP (which uses AES in CCM as specified in SP 800-38C) with cryptographic key size of 128 bits must be implemented. If CCM is only implemented to support CCMP for WLAN, AES-CCM does not need be selected. Optionally, AES-CCMP-256 or AES-GCMP-256 with cryptographic key size of 256 bits may be implemented.

Cryptographic operation

The TSF shall perform cryptographic hashing in accordance with a specified cryptographic algorithm SHA-1 and [**selection**: *SHA-256*, *SHA-384*, *SHA-512*, *no other algorithms*] and message digest sizes 160 and [**selection**: *256*, *384*, *512 bits*, *no other message digest sizes*] that meet the following: FIPS Pub 180-4.

Application Note: Per NIST SP 800-131A, SHA-1 for generating digital signatures is no longer allowed, and SHA-1 for verification of digital signatures is strongly discouraged as there may be risk in accepting these signatures. It is expected that vendors will implement SHA-2 algorithms in accordance with SP 800-131A.

SHA-1 is currently required in order to comply with the WLAN Client Extended Package. Vendors are strongly encouraged to implement updated protocols that support the SHA-2 family; until updated protocols are supported, this PP allows support for SHA-1 implementations in compliance with SP 800-131A.

The intent of this requirement is to specify the hashing function. The hash selection must support the message digest size selection. The hash selection should be consistent with the overall strength of the algorithm used (for example, SHA 256 for 128-bit keys).

The TSF hashing functions can be implemented in one of two modes. The first mode is the byteoriented mode. In this mode the TSF only hashes messages that are an integral number of bytes in length; i.e. the length (in bits) of the message to be hashed is divisible by 8. The second mode is the bitoriented mode. In this mode the TSF hashes messages of arbitrary length. The TSF may implement either bit-oriented or byte-oriented; both implementations are not required.

Cryptographic operation

The TSF shall perform cryptographic signature services (generation and verification) in accordance with a specified cryptographic algorithm[**selection**:

- *RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4,*

"Digital Signature Standard (DSS)", Section 4,
- *ECDSA schemes using "NIST curves" P-384 and [**selection**: P-256, P-521, no other curves] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5*,
- *No other algorithms*

].

Application Note: The ST Author should choose the algorithm implemented to perform digital signatures; if more than one algorithm is available, this requirement should be iterated to specify the functionality. For the algorithm chosen, the ST author should make the appropriate assignments/selections to specify the parameters that are implemented for that algorithm.

Cryptographic operation

The TSF shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm HMAC-SHA-1 and [**selection**: *HMAC-SHA-256*, *HMAC-SHA-384*, *HMAC-SHA-512*, *no other algorithms*] and cryptographic key sizes [**assignment**: *key size (in bits) used in HMAC*] and message digest sizes 160 and [**selection**: *256*, *384*, *512*, *no other*] bits that meet the following: FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code", and FIPS Pub 180-4, "Secure Hash Standard".

Application Note: The selection in this requirement must be consistent with the key size specified for the size of the keys used in conjunction with the keyed-hash message authentication. HMAC-SHA-1 is currently required in order to comply with the WLAN Client EP.

Cryptographic operation

The TSF shall perform conditioning in accordance with a specified cryptographic algorithm HMAC-[**selection**: *SHA-256*, *SHA-384*, *SHA-512*] using a salt, and [**selection**: *PBKDF2 with [**assignment**: number of iterations] iterations*, *[**assignment**: key stretching funtion]*, *no other function*] and output cryptographic key sizes [**selection**: *128*, *256*] that meet the following: [**selection**: *NIST SP 800-132*, *no standard*].

Application Note: The key cryptographic key sizes in the third selection should be made to correspond to the KEK key sizes selected in FCS_CKM_EXT.3.

This password must be conditioned into a string of bits that forms the submask to be used as input into the KEK. Conditioning can be performed using one of the identified hash functions and may include a key stretching function; the method used is selected by the ST Author. If selected, NIST SP 800-132 requires the use of a pseudo-random function (PRF) consisting of HMAC with an approved hash function. The ST author selects the hash function used, also includes the appropriate requirements for HMAC and the hash function.

Appendix A of NIST SP 800-132 recommends setting the iteration count in order to increase the computation needed to derive a key from a password and, therefore, increase the workload of performing a dictionary attack.

Extended: HTTPS Protocol

The TSF shall implement the HTTPS protocol that complies with RFC 2818.

The TSF shall implement HTTPS using TLS as defined in the Package for Transport Layer Security.

Application Note: The Package for Transport Layer Security shall be included in the ST, with the following selections made:

- FCS_TLS_EXT.1:
  - TLS shall be selected
  - Client shall be selected

The TSF shall notify the application and [**selection**: *not establish the connection*, *request application authorization to establish the connection*, *no other action*] if the peer certificate is deemed invalid.

Application Note: Validity is determined by the certificate path, the expiration date, and the revocation status in accordance with RFC 5280.

If "not establish the connection" is selected then "with no exceptions" shall be selected for FCS_TLSC_EXT.1.3 in the Package for Transport Layer Security. If "request application authorization to establish the connection" is selected then "except when override is authorized" shall be selected for FCS_TLSC_EXT.1.3 in the Package for Transport Layer Security. If "no other action" is selected either selection can be made in FCS_TLSC_EXT.1.3.

FMT_SMF_EXT.1 Function configures whether to allow/disallow the establishment of a trusted channel if the peer certificate is deemed invalid.

Extended: Initialization Vector Generation

The TSF shall generate IVs in accordance with : References and IV Requirements for NIST-approved Cipher Modes.

Application Note: lists the requirements for composition of IVs according to the NIST Special Publications for each cipher mode. The composition of IVs generated for encryption according to a cryptographic protocol is addressed by the protocol. Thus, this requirement addresses only IVs generated for key storage and data storage encryption.

Extended: Cryptographic Operation (Random Bit Generation)

The TSF shall perform all deterministic random bit generation services in accordance with NIST Special Publication 800-90A using [**selection**: *Hash_DRBG (any)*, *HMAC_DRBG (any)*, *CTR_DRBG (AES)*].

The deterministic RBG shall be seeded by an entropy source that accumulates entropy from [**selection**: *a software-based noise source*, *TSF-hardware-based noise source*] with a minimum of [**selection**: *128 bits*, *256 bits*] of entropy at least equal to the greatest security strength (according to NIST SP 800-57) of the keys and hashes that it will generate.

The TSF shall be capable of providing output of the RBG to applications running on the TSF that request random bits.

Application Note: SP 800-90A contains three different methods of generating random numbers; each of these, in turn, depends on underlying cryptographic primitives (hash functions/ciphers). The ST author will select the function used, and include the specific underlying cryptographic primitives used in the requirement or in the TSS. While any of the identified hash functions (SHA-224, SHA-256, SHA-384, SHA-512) are allowed for Hash_DRBG or HMAC_DRBG, only AES-based implementations for CTR_DRBG are allowed.

The ST author must also ensure that any underlying functions are included in the baseline requirements for the TOE.

Health testing of the DRBGs is performed in conjunction with the self-tests required in FPT_TST_EXT.1.1.

For the selection in FCS_RBG_EXT.1.2, the ST author selects the appropriate number of bits of entropy that corresponds to the greatest security strength of the algorithms included in the ST. Security strength is defined in Tables 2 and 3 of NIST SP 800-57A. For example, if the implementation includes 2048-bit RSA (security strength of 112 bits), AES 128 (security strength 128 bits), and HMAC-SHA-256 (security strength 256 bits), then the ST author would select 256 bits.

The ST author may select either software or hardware noise sources. A hardware noise source is a component that produces data that cannot be explained by a deterministic rule, due to its physical nature. In other words, a hardware based noise source generates sequences of random numbers from a physical process that cannot be predicted. For example, a sampled ring oscillator consists of an odd number of inverter gates chained into a loop, with an electrical pulse traveling from inverter to inverter around the loop. The inverters are not clocked, so the precise time required for a complete circuit around the loop varies slightly as various physical effects modify the small delay time at each inverter on the line to the next inverter. This variance results in an approximate natural frequency that contains drift and jitter over time. The output of the ring oscillator consists of the oscillating binary value sampled at a constant rate from one of the inverters – a rate that is significantly slower than the oscillator's natural frequency.

Extended: Cryptographic Algorithm Services

The TSF shall provide a mechanism for applications to request the TSF to perform the following cryptographic operations:

- All mandatory and [**selection**: *selected algorithms*, *selected algorithms with the exception of ECC over curve 25519-based algorithms*] in FCS_CKM.2/LOCKED
- The following algorithms in FCS_COP.1/ENCRYPT: AES-CBC, [**selection**: *AES Key Wrap*, *AES Key Wrap with Padding*, *AES-GCM*, *AES-CCM*, *no other modes*]
- All mandatory and selected algorithms in FCS_COP.1/SIGN
- All mandatory and selected algorithms in FCS_COP.1/HASH
- All mandatory and selected algorithms in FCS_COP.1/KEYHMAC
- [**selection**:
  - *All mandatory and [**selection**: selected algorithms, selected algorithms with the exception of ECC over curve 25519-based algorithms] in FCS_CKM.1*,
  - *The selected algorithms in FCS_COP.1/CONDITION*,
  - *No other cryptographic operations*

  ]

Application Note: For each of the listed FCS components in the bulleted list, the intent is that the TOE will make available all algorithms specified for that component in the ST. For example, if for FCS_COP.1/HASH the ST author selects SHA-256, then the TOE would have to make available an interface to perform SHA-1 (the "mandatory" portion of FCS_COP.1/HASH) and SHA-256 (the "selected" portion of FCS_COP.1/HASH).

The exception is for FCS_COP.1/ENCRYPT. The TOE is not required to make available AES_CCMP, AES_XTS, AES_GCMP-256, or AES_CCMP_256 even though they may be implemented to perform TSF-related functions. It is

acceptable for the platform to not provide AES Key Wrap (KW) and AES Key Wrap with Padding (KWP) to applications even if selected in FCS_COP.1/ENCRYPT. However, the ST author is expected to select AES-GCM and/or AES-CCM if it is selected in the ST for the FCS_COP.1/ENCRYPT component.

Extended: Cryptographic Key Storage

The TSF shall provide [**selection**: *mutable hardware*, *software-based*] secure key storage for asymmetric private keys and [**selection**: *symmetric keys*, *persistent secrets*, *no other keys*].

Application Note: A hardware keystore can be exposed to the TSF through a variety of interfaces, including embedded on the motherboard, USB, microSD, and Bluetooth.

Immutable hardware is considered outside of this requirement and will be covered elsewhere.

If the secure key storage is implemented in software that is protected as required by FCS_STG_EXT.2, the ST author shall select "software-based." If "software-based" is selected, the ST author shall select "all software-based key storage" in FCS_STG_EXT.2.

Support for secure key storage for all symmetric keys and persistent secrets will be required in future revisions.

The TSF shall be capable of importing keys/secrets into the secure key storage upon request of [**selection**: *the user*, *the administrator*] and [**selection**: *applications running on the TSF*, *no other subjects*].

Application Note: If the ST Author selects only user, the ST Author shall select function in FMT_MOF_EXT.1.1.

The TSF shall be capable of destroying keys/secrets in the secure key storage upon request of [**selection**: *the user*, *the administrator*].

Application Note: If the ST Author selects only user, the ST Author shall select function in FMT_MOF_EXT.1.1.

The TSF shall have the capability to allow only the application that imported the key/secret the use of the key/secret. Exceptions may only be explicitly authorized by [**selection**: *the user*, *the administrator*, *a common application developer*].

Application Note: If the ST Author selects user or administrator, the ST Author must also select function in FMT_SMF_EXT.1.1. If the ST Author selects only user, the ST Author shall select function in FMT_MOF_EXT.1.1.

The TSF shall allow only the application that imported the key/secret to request that the key/secret be destroyed. Exceptions may only be explicitly authorized by [**selection**: *the user*, *the administrator*, *a common application developer*].

Application Note: If the ST Author selects user or administrator, the ST Author must also select function in FMT_SMF_EXT.1.1. If the ST Author selects only user, the ST Author shall select function in FMT_MOF_EXT.1.1.

Extended: Encrypted Cryptographic Key Storage

The TSF shall encrypt all DEKs, KEKs, [**assignment**: *any long-term trusted channel key material*] and [**selection**: *all software-based key storage*, *no other keys*] by KEKs that are [**selection**:

- *Protected by the REK with [**selection**:*
  - *encryption by a REK,*
  - *encryption by a KEK chaining from a REK,*
  - *encryption by a KEK that is derived from a REK*

  ] ,
- *Protected by the REK and the password with [**selection**:*
  - *encryption by a REK and the password-derived KEK,*
  - *encryption by a KEK chaining to a REK and the password-derived or biometric-unlocked KEK,*
  - *encryption by a KEK that is derived from a REK and the password-derived or biometric-unlocked KEK*

  ]

].

Application Note: The ST author must select "all software-based key storage" if "software-based" is selected in FCS_STG_EXT.1.1. If the ST author selects "mutable hardware" in FCS_STG_EXT.1.1, the secure key storage is not subject to this requirement. REKs are not subject to this requirement.

A REK and the password-derived KEK may be combined to form a combined KEK (as described in FCS_CKM_EXT.3) in order to meet this requirement.

Sensitive data is protected by the REK and the password or biometric. Sensitive data includes some or all user or enterprise data. Software-based key storage itself shall be considered sensitive data and be protected accordingly, i.e. by the password or biometric and REK.

All keys must ultimately be protected by a REK. Sensitive data must be protected by the password or biometric (selection 2). In particular, has KEKs protected according to these requirements: DEK_1 meets 2a and would be

appropriate for sensitive data, DEK_2 meets 1b and would not be appropriate for sensitive data, K_1 meets 1a and is not considered a sensitive key, and K_2 meets 2b and is considered a sensitive key.

Long-term trusted channel key material includes WPA2 (PSKs), IPsec (PSKs and client certificates) and Bluetooth keys. These keys shall not be protected by the password, as they may be necessary in the locked state. For clarity, the ST author must assign any Long-term trusted channel key material supported by the TOE . At a minimum, a TOE must support at least WPA2 and Bluetooth keys.

The documentation of the product's encryption key management should be detailed enough that, after reading, the evaluator will thoroughly understand the product's key management and how it meets the requirements to ensure the keys are adequately protected. This documentation should include an essay and diagram(s). This documentation is not required to be part of the TSS - it can be submitted as a separate document and marked as developer proprietary.

DEKs, KEKs, [**assignment**: *any long-term trusted channel key material*] and [**selection**: *all software-based key storage*, *no other keys*] shall be encrypted using one of the following methods: [**selection**:

- *using a SP800-56B key establishment scheme*,
- *using AES in the [**selection**: Key Wrap (KW) mode, Key Wrap with Padding (KWP) mode, GCM, CCM, CBC mode]*

] .

Application Note: The ST author selects which key encryption schemes are used by the TOE. This requirement refers only to KEKs as defined this PP and does not refer to those KEKs specified in other standards. The ST author must assign the same Long-term trusted channel key material assigned in FCS_STG_EXT.2.1.

Extended: Integrity of encrypted key storage

The TSF shall protect the integrity of any encrypted DEKs and KEKs and [**selection**: *long-term trusted channel key material*, *all software-based key storage*, *no other keys*] by [**selection**:

- *[**selection**: GCM, CCM, Key Wrap, Key Wrap with Padding] cipher mode for encryption according to FCS_STG_EXT.2*,
- *a hash (FCS_COP.1/HASH) of the stored key that is encrypted by a key protected by FCS_STG_EXT.2*,
- *a keyed hash (FCS_COP.1/KEYHMAC) using a key protected by a key protected by FCS_STG_EXT.2*,
- *a digital signature of the stored key using an asymmetric key protected according to FCS_STG_EXT.2*,
- *an immediate application of the key for decrypting the protected data followed by a successful verification of the decrypted data with previously known information*

].

Application Note: The ST author must assign the same Long-term trusted channel key material assigned in FCS_STG_EXT.2.1.

The TSF shall verify the integrity of the [**selection**: *hash*, *digital signature*, *MAC*] of the stored key prior to use of the key.

Application Note: This requirement is not applicable to derived keys that are not stored. It is not expected that a single key will be protected from corruption by multiple of these methods; however, a product may use one integrity-protection method for one type of key and a different method for other types of keys. The explicit Assurance Activities for each of the options will be addressed in each of the requirements (FCS_COP.1.1/HASH, FCS_COP.1.1/KEYHMAC).

Key Wrapping shall be implemented per SP800-38F.

The documentation of the product's encryption key management should be detailed enough that, after reading, the evaluator will thoroughly understand the product's key management and how it meets the requirements to ensure the keys are adequately protected. This documentation should include an essay and diagram(s). This documentation is not required to be part of the TSS - it can be submitted as a separate document and marked as developer proprietary.

Extended: Security access control

The TSF shall provide a mechanism to restrict the system services that are accessible to an application.

Application Note: Examples of system services to which this requirement applies include:

- obtain data from camera and microphone input devices
- get current GPS location
- retrieve credentials from system-wide credential store
- retrieve contacts list / address book
- retrieve stored pictures
- retrieve text messages
- retrieve emails
- retrieve device identifier information
- obtain network access

FEL-ACP The TSF shall provide an access control policy that prevents [**selection**: *application*, *groups of applications*] from accessing [**selection**: *all*, *private*] data stored by other [**selection**: *application*, *groups of applications*]. Exceptions may only be explicitly authorized for such sharing by [**selection**: *the user*, *the administrator*, *a common application developer*, *no one*].

Application Note: Application groups may be designated Enterprise or Personal. Applications installed by the user default to being in the Personal application group unless otherwise designated by the administrator in function of FMT_SMF_EXT.1.1. Applications installed by the administrator default to being in the Enterprise application group (this category includes applications that the user requests the administrator install, for instance by selecting the application for installation through an enterprise application catalog) unless otherwise designated by the administrator in function of FMT_SMF_EXT.1.1. It is acceptable for the same application to have multiple instances installed, each in different application groups. Private data is defined as data that is accessible only by the application that wrote it. Private data is distinguished from data that an application may, by design, write to shared storage areas.

If "groups of applications" is selected, FDP_ACF_EXT.2 must be included in the ST.

Extended: Protected Data Encryption

Encryption shall cover all protected data.

Application Note: Protected data is all non-TSF data, including all user or enterprise data. Some or all of this data may be considered sensitive data as well.

Encryption shall be performed using DEKs with AES in the [**selection**: *XTS*, *CBC*, *GCM*] mode with key size [**selection**: *128*, *256*] bits.

Application Note: IVs shall be generated in accordance with FCS_IV_EXT.1.1.

Extended: Sensitive Data Encryption

The TSF shall provide a mechanism for applications to mark data and keys as sensitive.

Application Note: Data and keys that have been marked as sensitive will be subject to certain restrictions (through other requirements) in both the locked and unlocked states of the Mobile Device. This mechanism allows an application to choose those data and keys under its control to be subject to those requirements.

In the future, this PP may require that all data and key created by applications will default to the "sensitive" marking, requiring an explicit "non-sensitive" marking rather than an explicit "sensitive" marking.

The TSF shall use an asymmetric key scheme to encrypt and store sensitive data received while the product is locked.

Application Note: Sensitive data is encrypted according to FDP_DAR_EXT.1.2. The asymmetric key scheme must be performed in accordance with FCS_CKM.2/LOCKED.

The intent of this requirement is to allow the device to receive sensitive data while locked and to store the received data in such a way as to prevent unauthorized parties from decrypting it while in the locked state. If only a subset of sensitive data may be received in the locked state, this subset must be described in the TSS.

Key material must be cleared when no longer needed according to FCS_CKM_EXT.4. For keys (or key material used to derive those keys) protecting sensitive data received in the locked state, "no longer needed" includes "while in the locked state." For example, in the first key scheme, this includes the DEK protecting the received data as soon as the data is encrypted. In the second key scheme this includes the private key for the data asymmetric pair, the generated shared secret, and any generated DEKs. Of course, both schemes require that a private key of an asymmetric pair (the RSA private key and the device-wide private key, respectively) be cleared when transitioning to the locked state.

The TSF shall encrypt any stored symmetric key and any stored private key of the asymmetric key(s) used for the protection of sensitive data according to FCS_STG_EXT.2.1 selection 2.

Application Note: Symmetric keys used to encrypt sensitive data while the TSF is in the unlocked state must be encrypted with (or chain to a KEK encrypted with) the REK and password-derived or biometric-unlocked KEK. A stored private key of the asymmetric key scheme for encrypting data in the locked state must be encrypted with (or chain to a KEK encrypted with) the REK and password-derived or biometric-unlocked KEK.

The documentation of the product's encryption key management should be detailed enough that, after reading, the evaluator will thoroughly understand the product's key management and how it meets the requirements to ensure the keys are adequately protected. This documentation should include an essay and diagram(s). This documentation is not required to be part of the TSS - it can be submitted as a separate document and marked as developer proprietary.

The TSF shall decrypt the sensitive data that was received while in the locked state upon transitioning to the unlocked state using the asymmetric key scheme and shall re-encrypt that sensitive data using the symmetric key scheme.

Extended: Subset information flow control

The TSF shall [**selection**:

- *provide an interface which allows a VPN client to protect all IP traffic using IPsec*,
- *provide a VPN client which can protect all IP traffic using IPsec*

] with the exception of IP traffic required to establish the VPN connection.

Application Note: Typically, the traffic required to establish the VPN connection is referred to as "Control Plane" traffic; whereas, the IP traffic protected by the IPsec VPN is referred to as "Data Plane" traffic. All "Data Plane" traffic must flow through the VPN connection and the VPN must not split-tunnel.

If no native IPsec client is validated or third-party VPN clients may also implement the required Information Flow Control, the first option shall be selected. In these cases, the TOE provides an API to third-party VPN clients that allow them to configure the TOE's network stack to perform the required Information Flow Control.

The ST author shall select the second option if the TSF implements a native VPN client (IPsec is selected in FTP_ITC_EXT.1). Thus the TSF shall be validated against the PP-Module for VPN Client and the ST author shall also include FDP_IFC_EXT from the PP-Module for VPN Client.

It is optional for the VPN client to be configured to be always-on per FMT_SMF_EXT.1 Function . Always-on means the establishment of an IPsec trusted channel to allow any communication by the TSF.

Extended: User Data Storage

The TSF shall provide protected storage for the Trust Anchor Database.

Extended: Inter-TSF user data transfer protection

The TSF shall provide a means for non-TSF applications executing on the TOE to use

- mutually authenticated TLS as defined in the Package for Transport Layer Security,
- HTTPS,

and [**selection**:

- *mutually authenticated DTLS as defined in the Package for Transport Layer Security*,
- *IPsec in accordance with the PP-Module for VPN Client*,
- *no other protocol*

] to provide a protected communication channel between the non-TSF application and another IT product that is logically distinct from other communication channels, provides assured identification of its end points, protects channel data from disclosure, and detects modification of the channel data.

Application Note: The intent of this requirement is that one of the selected protocols is available for use by user applications running on the device for use in connecting to distant-end services that are not necessarily part of the enterprise infrastructure. It should be noted that the FTP_ITC_EXT.1 requires that all TSF communications be protected using the protocols indicated in that requirement, so the protocols required by this component ride "on top of" those listed in FTP_ITC_EXT.1.

It should also be noted that some applications are part of the TSF, and FTP_ITC_EXT.1 requires that TSF applications be protected by at least one of the protocols in first selection in FTP_ITC_EXT.1. It is not required to have two different implementations of a protocol, or two different protocols, to satisfy both this requirement (for non-TSF apps) and FTP_ITC_EXT.1 (for TSF apps), as long as the services specified are provided.

The ST author shall list which trusted channel protocols are implemented by the Mobile Device for use by non-TSF apps.

The TSF shall be validated against requirements from the Package for Transport Layer Security, with the following selections made:

- FCS_TLS_EXT.1:
  - TLS is selected
  - Client is selected
- FCS_TLSC_EXT.1.1:
  - The cipher suites selected must correspond with the algorithms and hash functions allowed in FCS_COP.1.
  - Mutual authentication must be selected
- FCS_TLSC_EXT.1.3
  - With no exceptions is selected.

If "mutually authenticated DTLS as defined in the Package for Transport Layer Security" is selected, the TSF shall be validated against requirements from the Package for Transport Layer Security, with the following selections made:

- FCS_TLS_EXT.1:
  - DTLS is selected
  - client is selected
- FCS_DTLSC_EXT.1.1:
  - The cipher suites selected must correspond with the algorithms and hash functions allowed in FCS_COP.1.
  - mutual authentication must be selected
- FCS_DTLSC_EXT.1.3
  - With no exceptions is selected.

If the ST author selects IPsec, the TSF shall be validated against the PP-Module for Virtual Private Network (VPN) Clients.

The TSF shall permit the non-TSF applications to initiate communication via the trusted channel.

Extended: Inter-TSF user data transfer protection - Bluetooth

The TSF shall provide a means for non-TSF applications executing on the TOE to use

- Bluetooth BR/EDR in accordance with the PP-Module for Bluetooth,

and [**selection**:

- *Bluetooth LE in accordance with the PP-Module for Bluetooth*,
- *no other protocol*

] to provide a protected communication channel between the non-TSF application and another IT product that is logically distinct from other communication channels, provides assured identification of its end points, protects channel data from disclosure, and detects modification of the channel data.

Application Note: If the TOE includes Bluetooth hardware this requirement shall be included in the ST. The intent of this requirement is that Bluetooth BR/EDR and optionally Bluetooth LE is available for use by user applications running on the device for use in connecting to distant-end services that are not necessarily part of the enterprise infrastructure. The ST author shall list which trusted channel protocols are implemented by the Mobile Device for use by non-TSF apps.

The TSF shall be validated against requirements from the PP-Module for Bluetooth. It should be noted that the FTP_ITC_EXT.1 requires that all TSF communications be protected using the protocols indicated in that requirement, so the protocols required by this component ride "on top of" those listed in FTP_ITC_EXT.1.

The TSF shall permit the non-TSF applications to initiate communication via the trusted channel.

Extended: Authentication failure handling

The TSF shall consider password and [**selection**: *fingerprint*, *iris*, *face*, *voice*, *vein*, *hybrid*, *no other*] as critical authentication mechanisms.

Application Note: A critical authentication mechanism is one in which countermeasures are triggered (i.e. wipe of the device) when the maximum number of unsuccessful authentication attempts is exceeded, rendering the other factors unavailable.

If no additional authentication mechanisms are selected in FIA_UAU.5.1, then ‘no other’ shall be selected. If an additional authentication mechanism is selected in FIA_UAU.5.1, then it shall only be selected in FIA_AFL_EXT.1.1 if surpassing the authentication failure threshold for biometric data causes a countermeasure to be triggered regardless of the failure status of the other authentication mechanisms.

If the TOE implements multiple Authentication Factor interfaces (for example, a DAR decryption interface, a lockscreen interface, an auxiliary boot mode interface), this component applies to all available interfaces. For example, a password is a critical authentication mechanism regardless of if it is being entered at the DAR decryption interface or at a lockscreen interface.

The TSF shall detect when a configurable positive integer within [**assignment**: *range of acceptable values for each authentication mechanism*] of [**selection**: *unique*, *non-unique*] unsuccessful authentication attempts occur related to last successful authentication for each authentication mechanism.

Application Note: The positive integer(s) is configured according to FMT_SMF_EXT.1.1 function .

An unique authentication attempt is defined as any attempt to verify a password or biometric sample, in which the input is different from a previous attempt. ‘Unique’ shall be selected if the authentication system increments the counter only for unique unsuccessful authentication attempts. For example, if the same incorrect password is attempted twice the authentication system increments the counter once. ‘Non-unique’ shall be selected if the authentication system increments the counter for each unsuccessful authentication attempt, regardless of if the input is unique. For example, if the same incorrect password is attempted twice the authentication system increments the

counter twice.

If hybrid authentication (i.e. a combination of biometric and pin/password) is supported, a failed authentication attempt can be counted as a single attempt, even if both the biometric and pin/password were incorrect.

If the TOE supports multiple authentication mechanisms per FIA_UAU.5.1, this component applies to all authentication mechanisms. It is acceptable for each authentication mechanism to utilize an independent counter or for multiple authentication mechanisms to utilize a shared counter. The interaction between the authentication factors in regards to the authentication counter shall be in accordance with FIA_UAU.5.2.

If the TOE implements multiple Authentication Factor interfaces (for example, a DAR decryption interface, a lockscreen interface, an auxiliary boot mode interface), this component applies to all available interfaces. However, it is acceptable for each Authentication Factor interface to be configurable with a different number of unsuccessful authentication attempts.

The TSF shall maintain the number of unsuccessful authentication attempts that have occurred upon power off.

Application Note: The TOE may implement an Authentication Factor interface that precedes another Authentication Factor interface in the boot sequence (for example, a volume DAR decryption interface which precedes the lockscreen interface) before the user can access the device. In this situation, because the user must successfully authenticate to the first interface to access the second, the number of unsuccessful authentication attempts need not be maintained for the second interface.

When the defined number of unsuccessful authentication attempts has exceeded the maximum allowed for a given authentication mechanism, all future authentication attempts will be limited to other available authentication mechanisms, unless the given mechanism is designated as a critical authentication mechanism.

Application Note: In accordance with FIA_AFL_EXT.1.3, this requirement also applies after the TOE is powered off and powered back on.

When the defined number of unsuccessful authentication attempts for the last available authentication mechanism or single critical authentication mechanism has been surpassed, the TSF shall perform a wipe of all protected data.

Application Note: Wipe is performed in accordance with FCS_CKM_EXT.5. Protected data is all non-TSF data, including all user or enterprise data. Some or all of this data may be considered sensitive data as well.

If the TOE implements multiple Authentication Factor interfaces (for example, a DAR decryption interface, a lockscreen interface, an auxiliary boot mode interface), this component applies to all available interfaces.

The TSF shall increment the number of unsuccessful authentication attempts prior to notifying the user that the authentication was unsuccessful.

Application Note: This requirement is to ensure that if power is cut to the device directly after an authentication attempt, the counter will be incremented to reflect that attempt.

Extended: Password Management

The TSF shall support the following for the Password Authentication Factor:

1. Passwords shall be able to be composed of any combination of [**selection**: *upper and lower case letters*, *[**assignment**: a character set of at least 52 characters]*], numbers, and special characters: [**selection**: *"!"*, *"@"*, *"#"*, *"$"*, *"%"*, *"^"*, *"&"*, *"*"*, *"("*, *")"*, *[**assignment**: other characters]*] ;
2. Password length up to [**assignment**: *an integer greater than or equal to 14*] characters shall be supported.

Application Note: While some corporate policies require passwords of 14 characters or better, the use of a REK for DAR protection and key storage protection and the anti-hammer requirement (FIA_TRT_EXT.1) addresses the threat of attackers with physical access using much smaller and less complex passwords.

The ST author selects the character set: either the upper and lower case Basic Latin letters or another assigned character set containing at least 52 characters. The assigned character set must be well defined: either according to an international encoding standard (such as Unicode) or defined in the assignment by the ST author. The ST author also selects the special characters that are supported by TOE; they may optionally list additional special characters supported using the assignment.

Extended: Authentication Throttling

The TSF shall limit automated user authentication attempts by [**selection**: *preventing authentication via an external port*, *enforcing a delay between incorrect authentication attempts*] for all authentication mechanisms selected in FIA_UAU.5.1. The minimum delay shall be such that no more than 10 attempts can be attempted per 500 milliseconds.

Application Note: The authentication throttling applies to all authentication mechanisms selected in FIA_UAU.5.1. The user authentication attempts in this requirement are attempts to guess the Authentication Factor. The developer

can implement the timing of the delays in the requirements using unequal or equal timing of delays. The minimum delay specified in this requirement provides defense against brute forcing.

Multiple Authentication Mechanisms

FEL-UAU The TSF shall provide password and [**selection**: *fingerprint*, *iris*, *face*, *voice*, *vein*, *hybrid*, *no other mechanism*] to support user authentication.

Application Note: The TSF must support a Password Authentication Factor and may optionally implement a BAF, in the form of a fingerprint, iris, face, voice and (finger/palm) vein. A hybrid authentication factor is where a user has to submit a combination of PIN/password and biometric sample where both have to pass and if either fails the user is not made aware of which factor failed.

If "hybrid" is selected, a biometric modality does not need to be selected, but should be selected if the biometric authentication can be used independent of the hybrid authentication, i.e. without having to enter a PIN/password.

If a biometric modality or "hybrid" is selected, then FIA_BMG_EXT.1 and FDP_PBA_EXT.1 must be included in the ST.

If "using a PIN as an additional factor" or "using a password as an additional factor" is selected in FDP_PBA_EXT.1.1, then "hybrid" shall be selected.

The Password Authentication Factor is configured according to FIA_PMG_EXT.1.

The TSF shall authenticate any user's claimed identity according to the [**assignment**: *rules describing how each authentication mechanism provides authentication*].

Application Note: For all authentication mechanisms specified in FIA_UAU.5.1, the TSS shall describe the rules as to how each authentication mechanism is used. Example rules are how the authentication mechanism authenticates the user (i.e. how does the TSF verify that the correct password or biometric sample was entered), the result of a successful authentication (i.e. is the user input used to derive or unlock a key) and which authentication mechanism can be used at which authentication factor interfaces (i.e. if there are times, for example, after a reboot, that only specific authentication mechanisms can be used). If multiple BAFs are supported per FIA_UAU.5.1, the interaction between the BAFs, shall be described. For example, if the multiple BAFs can be enabled at the same time. Rules regarding how the authentication factors interact in terms of unsuccessful authentication are covered in FIA_AFL_EXT.1.

Re-Authentication

The TSF shall re-authenticate the user via the Password Authentication Factor under the conditions attempted change to any supported authentication mechanisms.

Application Note: The password authentication factor must be entered before either the password or biometric authentication factor, if selected in FIA_UAU.5.1, can be changed.

The TSF shall re-authenticate the user via an authentication factor defined in FIA_UAU.5.1 under the conditions TSF-initiated lock, user-initiated lock, [**assignment**: *other conditions*].

Application Note: Depending on the selections made in FIA_UAU.5.1, either the password (at a minimum), biometric authentication or hybrid authentication mechanisms can be used to unlock the device. TSF- and user-initiated locking is described in FTA_SSL_EXT.1.

Protected Authentication Feedback

The TSF shall provide only obscured feedback to the device's display to the user while the authentication is in progress.

Application Note: This applies to all authentication methods specified in FIA_UAU.5.1. The TSF may briefly (1 second or less) display each character or provide an option to allow the user to unmask the password; however, the password must be obscured by default.

If a BAF is selected in FIA_UAU.5.1, the TSF shall not display sensitive information regarding the biometric that could aid an adversary in identifying and/or spoofing the respective biometric characteristics of a given human user. While it is true that biometric samples, by themselves, are not secret, the analysis performed by the respective biometric algorithms, as well as output data from these biometric algorithms, is considered sensitive and shall be kept secret. Where applicable, the TSF shall not reveal or make public the reason(s) for authentication failure.

Extended: Authentication for Cryptographic Operation

The TSF shall require the user to present the Password Authentication Factor prior to decryption of protected data and encrypted DEKs, KEKs and [**selection**: *long-term trusted channel key material*, *all software-based key storage*, *no other keys*] at startup.

Application Note: The intent of this requirement is to prevent decryption of protected data before the user has authorized to the device using the Password Authentication Factor. The Password Authentication Factor is also required in order derive the key used to decrypt sensitive data, which includes software-based secure key storage.

Extended: Timing of Authentication

The TSF shall allow [**selection**: *[**assignment**: list of actions]*, *no actions*] on behalf of the user to be performed before the user is authenticated.

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

Application Note: The security relevant actions allowed by unauthorized users in locked state must be listed. At a minimum the actions that correspond to the functions available to the user in FMT_SMF_EXT.1 and are allowed by unauthorized users in locked state should be listed. For example, if the user can enable/disable the camera per function of FMT_SMF_EXT.1 and unauthorized users can take a picture when the device is in locked state, this action must be listed.

Extended: X.509 Validation of certificates

The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation
- The certificate path must terminate with a certificate in the Trust Anchor Database
- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates
- The TSF shall validate the revocation status of the certificate using [**selection**: *the Online Certificate Status Protocol (OCSP) as specified in RFC 2560*, *a Certificate Revocation List (CRL) as specified in RFC 5759*]
- The TSF shall validate the extendedKeyUsage field according to the following rules:
    - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field
    - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field
    - (Conditional) Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the extendedKeyUsage field

Application Note: FIA_X509_EXT.1.1 lists the rules for validating certificates. The ST author shall select whether revocation status is verified using OCSP or CRLs. The WLAN Client EP to which a MDF TOE must also conform requires that certificates are used for EAP-TLS; this use requires that the extendedKeyUsage rules are verified. Certificates may optionally be used for trusted updates of system software and applications (FPT_TUD_EXT.2) and for integrity verification (FPT_TST_EXT.2/PREKERNEL) and, if implemented, must be validated to contain the Code Signing purpose extendedKeyUsage.

While FIA_X509_EXT.1.1 requires that the TOE perform certain checks on the certificate presented by a TLS server, there are corresponding checks that the authentication server will have to perform on the certificate presented by the client; namely that the extendedKeyUsage field of the client certificate includes "Client Authentication" and that the key agreement bit (for the Diffie-Hellman ciphersuites) or the key encipherment bit (for RSA ciphersuites) be set. Certificates obtained for use by the TOE will have to conform to these requirements in order to be used in the enterprise. This check is required to support EAP-TLS for the WLAN Client EP.

The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

Application Note: This requirement applies to certificates that are used and processed by the TSF and restricts the certificates that may be added to the Trust Anchor Database.

Extended: X.509 certificate authentication

FEL-CERT-AUTH The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for mutually authenticated TLS as defined in the Package for Transport Layer Security, HTTPS [**selection**: *IPsec in accordance with the PP-Module for VPN Client*, *mutually authenticated DTLS as defined in the Package for Transport Layer Security*], and [**selection**: *code signing for system software updates*, *code signing for mobile applications*, *code signing for integrity verification*, *[**assignment**: other uses]*, *no additional uses*].

Application Note: The ST author's first selection shall match the selection of FDP_UPC_EXT.1.1/NORMAL and FTP_ITC_EXT.1.1.

Certificates may optionally be used for trusted updates of system software (FPT_TUD_EXT.2.3) and mobile applications (FPT_TUD_EXT.4.1) and for integrity verification (FPT_TST_EXT.2/PREKERNEL). If "code signing for system software updates" or "code signing for mobile applications" is selected FPT_TUD_EXT.3.1 shall be included in the ST.

If FPT_TUD_EXT.4.1 is included in the ST, "code signing for mobile applications" must be included in the selection.

When the TSF cannot establish a connection to determine the revocation status of a certificate, the TSF shall [**selection**: *allow the administrator to choose whether to accept the certificate in these cases*, *allow the user to choose whether to accept the certificate in these cases*, *accept the certificate*, *not accept the certificate*].

Application Note: The TOE must not accept the certificate if it fails any of the other validation rules in FIA_X509_EXT.1. However, often a connection must be established to perform a verification of the revocation status of a certificate - either to download a CRL or to perform OCSP. The selection is used to describe the behavior in the event that such a connection cannot be established (for example, due to a network error). If the TOE has determined the certificate valid according to all other rules in FIA_X509_EXT.1, the behavior indicated in the selection shall determine the validity. If the administrator-configured or user-configured option is selected, the ST Author must also select function in FMT_SMF_EXT.1.

The TOE may behave differently depending on the trusted channel; for example, in the case of WLAN where connections are unlikely to be established, the TOE may accept the certificate even though certificates are not accepted for other channels. The ST author should select all applicable behaviors.

Extended: Request Validation of certificates

The TSF shall provide a certificate validation service to applications.

The TSF shall respond to the requesting application with the success or failure of the validation.

Application Note: In order to comply with all of the rules in FIA_X509_EXT.1, multiple API calls may be required; all of these calls should be clearly documented

Extended: Management of security functions behavior

The TSF shall restrict the ability to perform the functions in column 3 of to the user.

Application Note: The functions that have an "M" in the third column are mandatory for this component, thus are restricted to the user, meaning that the administrator cannot manage those functions. The functions that have an "O" in the third column are optional and may be selected; and those functions with a "-" in the third are not applicable and may not be selected. The ST author should select those security management functions that only the user may perform (i.e. the ones the administrator may not perform).

The ST author may not select the same function in both FMT_MOF_EXT.1.1 and FMT_MOF_EXT.1.2. A function cannot contain an "M" in both column 3 and column 5.

The ST author may use a table in the ST, indicating with clear demarcations (to be accompanied by an index) those functions that are restricted to the user (column 3). The ST author should iterate a row to indicate any variations in the selectable sub-functions or assigned values with respect to the values in the columns.

For functions that are mandatory, any sub-functions not in a selection are also mandatory and any assignments must contain at least one assigned value. For non-selectable sub-functions in an optional function, all sub-functions outside a selection must be implemented in order for the function to be listed.

The TSF shall restrict the ability to perform the functions in column 5 of to the administrator when the device is enrolled and according to the administrator-configured policy.

Application Note: As long as the device is enrolled in management, the administrator (of the enterprise) must be guaranteed that minimum security functions of the enterprise policy are enforced. Further restrictive policies can be applied at any time by the user on behalf of the user or other administrators.

The functions that have an "M" in the fifth column are mandatory for this component; the functions that have an "O" in the fifth column are optional and may be selected; and those functions with a "-" in the fifth are not applicable and may not be selected.

The ST author may not select the same function in both FMT_MOF_EXT.1.1 and FMT_MOF_EXT.1.2.

The ST author should select those security management functions that the administrator may restrict. The ST author may use a table in the ST, indicating with clear demarcations (to be accompanied by an index) those functions that are and are not implemented with APIs for the administrator (as in column 4). Additionally, the ST author should demarcate which functions the user is prevented from accessing or performing (as in column 5). The ST author should iterate a row to indicate any variations in the selectable sub-functions or assigned values with respect to the values in the columns.

For functions that are mandatory, any sub-functions not in a selection are also mandatory and any assignments must

contain at least one assigned value. For non-selectable sub-functions in an optional function, all sub-functions outside the selection must be implemented in order for the function to be listed.

Extended: Specification of Management Functions

The TSF shall be capable of performing the following management functions:

: Management Functions

Status Markers:
M - Mandatory
O - Optional/Objective

Management Function	Impl.	User Only	Admin	Admin Only
. configure password policy: a. minimum password length b. minimum password complexity c. maximum password lifetime	X	-	X	X
. configure session locking policy: a. screen-lock enabled/disabled b. screen lock timeout c. number of authentication failures	X	-	X	X
. enable/disable the VPN protection: a. across device [**selection**: • *b. on a per-app basis*, • *c. on a per-group of applications processes basis* , • *d. no other method*]	X	O	O	O
. enable/disable [**assignment**: *list of all radios*]	X	O	O	O
. enable/disable [**assignment**: *list of audio or visual collection devices*]: a. across device [**selection**: • *b. on a per-app basis*, • *c. on a per-group of applications processes basis* , • *d. no other method*]	X	O	O	O
. transition to the locked state	X	-	X	-
.TSF wipe of protected data	X	-	X	-
.configure application installation policy by [**selection**: • *a. restricting the sources of applications*, • *b. specifying a set of allowed applications based on [**assignment**: application characteristics] (an application whitelist)*, • *c. denying installation of applications*]	X	-	X	X
.import keys/secrets into the secure key storage	X	O	O	-
. destroy imported keys/secrets and [**selection**: *no other keys/secrets*, *[**assignment**: list of other categories of keys/secrets]*] in the secure key storage	X	O	O	-
.import X.509v3 certificates into the Trust Anchor Database	X	-	X	O
.remove imported X.509v3 certificates and [**selection**: *no other X.509v3 certificates*, *[**assignment**: list of other categories of X.509v3 certificates]*] in the Trust Anchor Database	X	O	O	-

. enroll the TOE in management	X	X	-	-
. remove applications	X	-	X	O
. update system software	X	-	X	O
. install applications	X	-	X	O
. remove Enterprise applications	X	-	X	-
. enable/disable display notification in the locked state of: [**selection**: • *a. email notifications*, • *b. calendar appointments*, • *c. contact associated with phone call notification*, • *d. text message notification*, • *e. other application-based notifications*, • *f. all notifications*]	X	O	O	O
. enable data-at rest protection	X	O	O	O
.enable removable media's data-at-rest protection	X	O	O	O
. enable/disable location services: a. across device [**selection**: • *b. on a per-app basis*, • *c. on a per-group of applications processes basis* , • *d. no other method*]	X	O	O	O
. Enable/disable the use of [**selection**: *Biometric Authentication Factor*, *Hybrid Authentication Factor*]	X	O	O	O
. configure whether to allow/disallow establishment of a trusted channel if the peer/server certificate is deemed invalid.	X	O	O	O
. enable/disable all data signaling over [**assignment**: *list of externally accessible hardware ports*]	O	O	O	O
. enable/disable [**assignment**: *list of protocols where the device acts as a server*]	O	O	O	O
. enable/disable developer modes	O	O	O	O
. enable/disable bypass of local user authentication	O	O	O	O
. wipe Enterprise data	O	O	O	-
. approve [**selection**: *import*, *removal*] by applications of X.509v3 certificates in the Trust Anchor Database	O	O	O	O
. configure whether to allow/disallow establishment of a trusted channel if the TSF cannot establish a connection to determine the validity of a certificate	O	O	O	O
. enable/disable the cellular protocols used to connect to cellular network base stations	O	O	O	O
. read audit logs kept by the TSF	O	O	O	-
. configure [**selection**: *certificate*, *public-key*] used to validate digital signature on applications	O	O	O	O
. approve exceptions for shared use of keys/secrets by multiple applications	O	O	O	O
. approve exceptions for destruction of keys/secrets by applications that did not import the key/secret	O	O	O	O
. configure the unlock banner	O	-	O	O
. configure the auditable items	O	-	O	O
. retrieve TSF-software integrity verification values	O	O	O	O
. enable/disable [**selection**: • *USB mass storage mode*,				

• *USB data transfer without user authentication*, • *USB data transfer without authentication of the connecting system*]	O	O	O	O
. enable/disable backup of [**selection**: *all applications*, *selected applications*, *selected groups of applications*, *configuration data*] to [**selection**: *locally connected system*, *remote system*]	O	O	O	O
. enable/disable [**selection**: • *Hotspot functionality authenticated by [**selection**: pre-shared key, passcode, no authentication]*, • *USB tethering authenticated by [**selection**: pre-shared key, passcode, no authentication]*]	O	O	O	O
. approve exceptions for sharing data between [**selection**: *application*, *groups of application*]	O	O	O	O
. place applications into application groups based on [**assignment**: *enterprise configuration settings*]	O	O	O	O
. unenroll the TOE from management	O	O	O	O
. enable/disable the Always On VPN protection	O	O	O	O
. revoke Biometric template	O	O	O	O
. [**assignment**: *list of other management functions to be provided by the TSF*]	O	O	O	O

Application Note: compares the management functions required by this Protection Profile.

The first column lists the management functions identified in the PP.

In the following columns:

- ‘M’ means Mandatory
- ‘O’ means Optional/Objective

The second column (FMT_SMF_EXT.1) indicates whether the function is to be implemented. The ST author should select which Optional functions are implemented.

The third column (FMT_MOF_EXT.1.1) indicates functions that are to be restricted to the user (i.e. not available to the administrator).

The fourth column (Administrator) indicates functions that are available to the administrator. The functions restricted to the user (column 3) cannot also be available to the administrator. Functions available to the administrator can still be available to the user, as long as the function is not restricted to the administrator (column 5). Thus, if the TOE must offer these functions to the administrator to perform the fourth column shall be selected.

The fifth column (FMT_MOF_EXT.1.2) indicates whether the function is to be restricted to the administrator when the device is enrolled and the administrator applies the indicated policy. If the function is restricted to the administrator the function is not available to the user. This does not prevent the user from modifying a setting to make the function stricter, but the user cannot undo the configuration enforced by the administrator.

The ST author may use a table in the ST, listing only those functions that are implemented. For functions that are mandatory, any sub-functions not in a selection are also mandatory and any assignments must contain at least one assigned value. For functions that are optional and contain an assignment or selection, at least one value must be assigned/selected to be included in the ST. For non-selectable sub-functions in an optional function, all sub-functions must be implemented in order for the function to be included. For functions with a "per-app basis" sub function and an assignment, the ST author must indicate which assigned features are manageable on a per-app basis and which are not by iterating the row.

Function-specific Application Notes:

For functions , and , the function must be implemented on a device-wide basis but may also be implemented on a per-app basis or on a per-group of applications basis in which the configuration includes the list of applications or groups of applications to which the enable/disable applies.

Function addresses enabling and disabling the IPsec VPN only. The configuration of the VPN Client itself (with information such as VPN Gateway, certificates, and algorithms) is addressed by the PP-Module for VPN Client. The administrator options should only be listed if the administrator can remotely enable/disable the VPN connection.

Function optionally allows the VPN to be configured per-app or per-groups of apps. If this configuration is selected, it does not void FDP_IFC_EXT.1. Instead FDP_IFC_EXT.1 is applied to the application or group of applications the VPN is applied to. In other words, all traffic destined for the VPN-enabled application or group of applications, must travel through the VPN, but traffic not destined for that application or group of applications can travel outside the VPN.

When the VPN is configured across the device FDP_IFC_EXT.1 applies to all traffic and the VPN must not split tunnel.

The assignment in function consists of all radios present on the TSF, such as Wi-Fi, GPS, cellular, NFC, Bluetooth BR/EDR, and Bluetooth LE, which can be enabled and disabled. In the future, if both Bluetooth BR/EDR and Bluetooth LE are supported, they will be required to be enabled and disabled separately. Disablement of the cellular radio does not imply that the radio may not be enabled in order to place emergency phone calls; however, it is not expected that a device in "airplane mode", where all radios are disabled, will automatically (without authorization) turn on the cellular radio to place emergency calls.

The assignment in function consists of at least one audio and/or visual device, such as camera and microphone, which can be enabled and disabled by either the user or administrator. Disablement of the microphone does not imply that the microphone may not be enabled in order to place emergency phone calls. If certain devices are able to be restricted to the enterprise (either device-wide, per-app or per-group of applications) and others are able to be restricted to users, then this function should be iterated in the table with the appropriate table entries.

Regarding functions and , disablement of a particular radio or audio/visual device must be effective as soon as the TOE has power. Disablement must also apply when the TOE is booted into auxiliary boot modes, for example, associated with updates or backup. If the TOE supports states in which security management policy is inaccessible, for example, due to data-at-rest protection, it is acceptable to meet this requirement by ensuring that these devices are disabled by default while in these states. That these devices are disabled during auxiliary boot modes does not imply that the device (particularly the cellular radio) may not be enabled in order to perform emergency phone calls.

Wipe of the TSF (function) is performed according to FCS_CKM_EXT.5. Protected data is all non-TSF data, including all user or enterprise data. Some or all of this data may be considered sensitive data as well.

The selection in function allows the ST author to select which mechanisms are available to the administrator through the MDM Agent to restrict the applications which the user may install. The ST author shall state if application whitelisting is applied device-wide or if it can be specified to apply to either the Enterprise and/or Personal applications.

- If the administrator can restrict the sources from which applications can be installed, the ST author selects option a.
- If the administrator can specify a whitelist of allowed applications, the ST author selects option b. The ST author should list any application characteristics (e.g. name, version, or developer) based on which the whitelist can be formed.
- If the administrator can prevent the user from installing additional applications, the ST author selects c.

In the future, function may require destruction or disabling of any default trusted CA certificates, excepting those CA certificates necessary for continued operation of the TSF, such as the developer’s certificate. At this time, the ST author shall indicate in the assignment whether pre-installed or any other category of X.509v3 certificates may be removed from the Trust Anchor Database.

For function , the enrollment function may be installing an MDM agent and includes the policies to be applied to the device. It is acceptable for the user approval notice to require the user to intentionally opt to view the policies (for example, by "tapping" on a "View" icon) rather than listing the policies in full in the notice.

For function , the administrator capability to update the system software may be limited to causing a prompt to the user to update rather than the ability to initiate the update itself. As the administrator is likely to be acting remotely, he/she would be unaware of inopportune situations, such as low power, which may cause the update to fail and the device to become inoperable. The user can refuse to accept the update in such situations. It is expected that system architects will be cognizant of this limitation and will enforce network access controls in order to enforce enterprise-critical updates.

Function addresses both installation and update. This protection profile does not distinguish between installation and update of applications because mobile devices typically completely overwrite the previous installation with a new installation during an application update.

For function , "Enterprise applications" are those applications that belong to the Enterprise application group. Applications installed by the enterprise administrator (including automatic installation by the administrator after being requested by the user from a catalog of enterprise applications) are by default placed in the Enterprise application group unless an exception has been made in function of FMT_SMF_EXT.1.1.

If the display of notifications in the locked state is supported, the configuration of these notifications (function) must be included in the selection.

Function must be included in the selection if data-at-rest protection is not natively enabled.

Function is implicitly met if the TSF does not support removable media.

For function , location services include location information gathered from GPS, cellular, and Wi-Fi.

Function is implicitly met if the TOE does not contain a BAF. This selection shall correspond with the selection made in FIA_UAU.5.1. If a BAF is selected in FIA_UAU.5.1, "Biometric Authentication Factor" shall be selected and the user or admin shall have the option to disable the use of it. If multiple BAFs are selected in FIA_UAU.5.1, this applies to all different modalities. If "hybrid" is selected in FIA_UAU.5.1 it shall be selected and the user or admin shall have the option to disable the use of it.

For function , the configuration can be different depending on the specific trusted channel.

The assignment in function consists of all externally accessible hardware ports, such as USB, the SD card, and HDMI, whose data transfer capabilities can be enabled and disabled by either the user or administrator. Disablement of data transfer over an external port must be effective during and after boot into the normal operative mode of the device. If the TOE supports states in which configured security management policy is inaccessible, for example, due to data-at-rest protection, it is acceptable to meet this requirement by ensuring that data transfer is disabled by default while in these states. Each of the ports may be enabled or disabled separately. The configuration policy need not disable all ports together. In the case of USB, chagrining is still allowed if data transfer capabilities have been disabled.

The assignment in function consists of all protocols where the TSF acts as a server, which can be enabled and disabled by either the user or administrator.

Function must be included in the selection if developer modes are supported by the TSF.

Function must be included in the selection if bypass of local user authentication, such as a "Forgot Password", password hint, or remote authentication feature, is supported.

Function must be included in the selection if the TSF allows applications, other than the MDM Agents, to import or remove X.509v3 certificates from the Trust Anchor Database. The MDM Agent is considered the administrator. This function does not apply to applications trusting a certificate for its own validations. The function only applies to situations where the application modifies the device-wide Trust Anchor Database, affecting the validations performed by the TSF for other applications. The user or administrator may be provided the ability to globally allow or deny any application requests in order to meet this requirement.

Function must be included in the ST if "administrator-configured option" is selection in FIA_X509_EXT.2.2.

Function should be included in the selection if FPT_TUD_EXT.4.1 is included in the ST and the configurable option is selected.

Function should be included in the selection if user or administrator is selected in FCS_STG_EXT.1.4.

Function should be included in the selection if user or administrator is selected in FCS_STG_EXT.1.5.

Function must be included in the selection if FTA_TAB.1 is included in the ST.

Function must be included in the selection if FAU_SEL.1 is included in the ST.

For function , hotspot functionality refers to the condition in which the mobile device is serving as an access point to other devices, not the connection of the TOE to external hotspots.

Functions and correspond to FDP_ACF_EXT.1.2.

For function , FMT_SMF_EXT.2.1 specifies actions to be performed when the TOE is unenrolled from management.

For function , shall be included in the ST if IPsec is selected in FTP_ITC_EXT.1 and the native IPsec VPN client can be configured to be Always-On. Always-On is defined as when the TOE has a network connection the VPN attempts to connect, all data leaving the device uses the VPN when the VPN is connected and no data leaves that device when the VPN is disconnected. The configuration of the VPN Client itself (with information such as VPN Gateway, certificates, and algorithms) is addressed by the PP-Module for VPN Client.

Extended: Specification of Remediation Actions

The TSF shall offer [**selection**: *wipe of protected data*, *wipe of sensitive data*, *remove Enterprise applications*, *remove all device-stored Enterprise resource data*, *remove Enterprise secondary authentication data*, ***[assignment**: list other available remediation actions]*] upon un-enrollment and [**selection**: ***[assignment**: other administrator-configured triggers]*, *no other triggers*].

Application Note: Un-enrollment may consist of removing the MDM agent or removing the administrator's policies. The functions in the selection are remediation actions that TOE provides (perhaps via APIs) to the administrator (perhaps via an MDM agent) that are performed upon un-enrollment. "Enterprise applications" refers to applications that are in the Enterprise application group. "Enterprise resource data" refers to all stored Enterprise data and the separate resources that are available to the Enterprise application group, per FDP_ACF_EXT.2.1. If FDP_ACF_EXT.2.1 is included in the ST, then "remove all device-stored Enterprise resource data" must be selected, and is defined to be all resources selected in FDP_ACF_EXT.2.1. If FIA_UAU_EXT.4.1 is included in the ST, then "remove Enterprise secondary authentication data" must be selected. If FIA_UAU_EXT.4.1 is not included in the ST, then "remove Enterprise secondary authentication data" cannot be selected. Enterprise secondary authentication data only refers to any data stored on the TOE that is specifically used as part of a secondary authentication mechanism to authenticate access to Enterprise applications and shared resources. Material that is used for the TOE's primary authentication mechanism or other purposes not related to authentication to or protection of Enterprise applications or shared resources should not be removed.

Protected data is all non-TSF data, including all user or enterprise data. Some or all of this data may be considered sensitive data as well. If "wipe of protected data" is selected it is assumed that the sensitive data is wiped as well. However, if "wipe of sensitive data" is selected, it does not imply that all non-TSF data (protected data) is wiped. If "wipe of protected data" or "wipe of sensitive data" is selected the wipe shall be in accordance with FCS_CKM_EXT.5.1. Thus cryptographically wiping the device is an acceptable remediation action.

Extended: Anti-Exploitation Services (ASLR)

The TSF shall provide address space layout randomization ASLR to applications.

The base address of any user-space memory mapping will consist of at least 8 unpredictable bits.

Application Note: The 8 unpredictable bits may be provided by the TSF RBG (as specified in FCS_RBG_EXT.1) but is not required.

Extended: Anti-Exploitation Services (Memory Page Permissions)

The TSF shall be able to enforce read, write, and execute permissions on every page of physical memory.

Extended: Anti-Exploitation Services (Overflow Protection)

TSF processes that execute in a non-privileged execution domain on the application processor shall implement stack-based buffer overflow protection.

Application Note: A "non-privileged execution domain" refers to the user mode (as opposed to kernel mode, for instance) of the processor. While not all TSF processes must implement such protection, it is expected that most of the processes (to include libraries used by TSF processes) do implement buffer overflow protections.

Extended: Domain Isolation

The TSF shall protect itself from modification by untrusted subjects.

The TSF shall enforce isolation of address space between applications.

Application Note: In addition to the TSF software (e.g., kernel image, device drivers, trusted applications) that resides in storage, the execution context (e.g., address space, processor registers, per-process environment variables) of the software operating in a privileged mode of the processor (e.g., kernel), as well as the context of the trusted applications is to be protected. In addition to the software, any configuration information that controls or influences the behavior of the TSF is also to be protected from modification by untrusted subjects.

Configuration information includes, but is not limited to, user and administrative management function settings, WLAN profiles, and Bluetooth data such as the service-level security requirements database.

Untrusted subjects include untrusted applications; unauthorized users who have access to the device while powered off, in a screen-locked state, or when booted into auxiliary boot modes; and, unauthorized users or untrusted software or hardware which may have access to the device over a wired interface, either when the device is in a screen-locked state or booted into auxiliary boot modes.

Extended: JTAG Disablement

The TSF shall [**selection**: *disable access through hardware*, *control access by a signing key*] to JTAG.

Application Note: This requirement means that access to JTAG shall be disabled either through hardware and/or restricted through the use of a signing key.

Extended: Key Storage

The TSF shall not store any plaintext key material in readable non-volatile memory.

Application Note: The intention of this requirement is that the TOE will not write plaintext keying material to persistent storage. For the purposes of this requirement, keying material refers to authentication data, passwords, secret/private symmetric keys, private asymmetric keys, data used to derive keys, etc. These values must be stored encrypted.

This requirement also applies to any value derived from passwords. Thus, the TOE cannot store plaintext password hashes for comparison purposes before protected data is decrypted, and the TOE should use key derivation and decryption to verify the Password Authentication Factor.

If a BAF is selected in FIA_UAU.5.1, keying material also refers to source biometric data (i.e. fingerprint), enrollment and authentication templates, the features an algorithm uses to perform biometric authentication for enrollment or

verification (i.e. location of minutia), threshold values used in making the match adjudication, intermediate calculations generated while building an enrollment or authentication template (i.e. direction maps, minutia counts, binarized and skeletonized representations of friction ridge patterns, etc.), and final match scores. Any images or metadata identifying the user for authentication shall be stored encrypted.

If "hybrid" is selected in FIA_UAU.5.1, in addition to the keying material included for the BAF, mentioned in the previous paragraph, keying material also refers to the PIN/password used as part of the hybrid authentication.

Extended: No Key Transmission

The TSF shall not transmit any plaintext key material outside the security boundary of the TOE.

Application Note: The intention of this requirement is to prevent the logging of plaintext key information to a service that transmits the information off-device. For the purposes of this requirement, key material refers to keys, passwords, and other material that is used to derive keys.

If a BAF is selected in FIA_UAU.5.1, keying material also refers to source biometric data (i.e. fingerprint), enrollment and authentication templates, the features an algorithm uses to perform biometric authentication for enrollment or verification (i.e. location of minutia), threshold values used in making the match adjudication, intermediate calculations generated while building an enrollment or authentication template (i.e. direction maps, minutia counts, binarized and skeletonized representations of friction ridge patterns), and final match scores.

If "hybrid" is selected in FIA_UAU.5.1, in addition to the keying material included for the BAF, mentioned in the previous paragraph, keying material also refers to the PIN/password used as part of the hybrid authentication.

In the future, this requirement will apply to symmetric and asymmetric private keys stored in the TOE secure key storage where applications are outside the boundary of the TOE. Thus, the TSF will be required to provide cryptographic key operations (signature, encryption, and decryption) on behalf of applications (FCS_SRV_EXT.2.1) that have access to those keys.

Extended: No Plaintext Key Export

The TSF shall ensure it is not possible for the TOE user(s) to export plaintext keys.

Application Note: Plaintext keys include DEKs, KEKs, and all keys stored in the secure key storage (FCS_STG_EXT.1). The intent of this requirement is to prevent the plaintext keys from being exported during a backup authorized by the TOE user or administrator.

Extended: Self-Test Notification

The TSF shall transition to non-operational mode and [**selection**: *log failures in the audit record*, *notify the administrator*, *[**assignment**: other actions]*, *no other actions*] when the following types of failures occur:

- failures of the self-test(s)
- TSF software integrity verification failures
- [**selection**: *no other failures*, *[**assignment**: other failures]*]

Reliable time stamps

The TSF shall be able to provide reliable time stamps for its own use.

Extended: TSF Cryptographic Functionality Testing

The TSF shall run a suite of self-tests during initial start-up (on power on) to demonstrate the correct operation of all cryptographic functionality.

Application Note: This requirement may be met by performing known answer tests and/or pair-wise consistency tests. The self-tests must be performed before the cryptographic functionality is exercised (for example, during the initialization of a process that utilizes the functionality).

The cryptographic functionality includes the cryptographic operations in FCS_COP, the key generation functions in FCS_CKM, and the random bit generation in FCS_RBG_EXT.

Extended: TSF Integrity Checking

The TSF shall verify the integrity of the bootchain up through the Application Processor OS kernel stored in mutable media prior to its execution through the use of [**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: The bootchain of the TSF is the sequence of firmware and software, including ROM, bootloader(s), and kernel, which ultimately result in loading the kernel on the Application Processor, regardless of which processor executes that code. Executable code that would be loaded after the kernel is covered in FPT_TST_EXT.2/POSTKERNEL.

In order to meet this requirement, the hardware protection may be transitive in nature: a hardware-protected public key, hash of an asymmetric key, or hash may be used to verify the mutable bootloader code which contains a key or hash used by the bootloader to verify the mutable OS kernel code, which contains a key or hash to verify the next layer of executable code, and so on.

The cryptographic mechanism used to verify the (initial) mutable executable code must be protected, such as being implemented in hardware or in read-only memory (ROM).

Extended: Trusted Update: TSF version query

The TSF shall provide authorized users the ability to query the current version of the TOE firmware/software.

The TSF shall provide authorized users the ability to query the current version of the hardware model of the device.

Application Note: The current version of the hardware model of the device is an identifier that is sufficient to indicate (in tandem with manufacturer documentation) the hardware which comprises the device.

The TSF shall provide authorized users the ability to query the current version of installed mobile applications.

Application Note: The current version of mobile applications is the name and published version number of each installed mobile application.

Extended: TSF Update Verification

The TSF shall verify software updates to the Application Processor system software and [**selection**: *[**assignment**: other processor system software]*, *no other processor system software*] using a digital signature verified by the manufacturer trusted key prior to installing those updates

Application Note: The digital signature mechanism is implemented in accordance with FCS_COP.1.1/SIGN.

At this time, this requirement does not required verification of software updates to the software operating outside the Application Processor.

Any change, via a supported mechanism, to software residing in non-volatile storage is deemed a software update. Thus, this requirement applies to TSF software updates regardless of how the software arrives or is delivered to the device. This includes over-the-air (OTA) updates as well as partition images containing software which may be delivered to the device over a wired interface.

The TSF shall [**selection**: *never update*, *update only by verified software*] the TSF boot integrity [**selection**: *key*, *hash*].

Application Note: The key or hash updated via this requirement is used for verifying software before execution in FPT_TST_EXT.2/PREKERNEL. The key or hash is verified as a part of the digital signature on an update, and the software which performs the update of the key or hash is verified by FPT_TST_EXT.2/PREKERNEL.

The TSF shall verify that the digital signature verification key used for TSF updates [**selection**: *is validated to a public key in the Trust Anchor Database*, *matches an immutable hardware public key*].

Application Note: The ST author shall indicate the method by which the signing key for system software updates is limited and, if selected in FPT_TUD_EXT.2.3, shall indicate how this signing key is protected by the hardware.

If certificates are used, certificates are validated for the purpose of software updates in accordance with FIA_X509_EXT.1 and should be selected in FIA_X509_EXT.2.1. Additionally, FPT_TUD_EXT.3.1 must be included in the ST.

The TSF shall verify mobile application software using a digital signature mechanism prior to installation.

Application Note: This requirement does not necessitate an X.509v3 certificate or certificate validation. X.509v3 certificates and certificate validation are addressed in FPT_TUD_EXT.4.1.

Extended: TSF- and User-initiated Locked State

The TSF shall transition to a locked state after a time interval of inactivity.

The TSF shall transition to a locked state after initiation by either the user or the administrator.

The TSF shall, upon transitioning to the locked state, perform the following operations:

a. clearing or overwriting display devices, obscuring the previous contents;
b. [**assignment**: *other actions performed upon transitioning to the locked state*].

Application Note: The time interval of inactivity is configured using FMT_SMF_EXT.1 function . The user/administrator-initiated lock is specified in FMT_SMF_EXT.1 function .

Extended: Trusted Channel Communication

The TSF shall use

- 802.11-2012 in accordance with the Extended Package for WLAN Clients,
- 802.1X in accordance with the Extended Package for WLAN Clients,
- EAP-TLS in accordance with the Extended Package for WLAN Clients,
- mutually authenticated TLS as defined in the Package for Transport Layer Security

and [**selection**:

- *IPsec in accordance with the PP-Module for VPN Client*,
- *mutually authenticated DTLS as defined in the Package for Transport Layer Security*,
- *HTTPS*

] protocol to provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels, provides assured identification of its end points, protects channel data from disclosure, and detects modification of the channel data.

Application Note: The intent of the mandatory portion of the above requirement is to use the cryptographic protocols identified in the requirement to establish and maintain a trusted channel between the TOE and an access point, VPN Gateway, or other trusted IT product.

The ST author shall list which trusted channel protocols are implemented by the Mobile Device.

The TSF shall be validated against the Extended Package for WLAN Clients to satisfy the mandatory trusted channels of 802.11-2012, 802.1X, and EAP-TLS.

To satisfy the mandatory trusted channel of TLS and if "mutually authenticated DTLS as defined in the Package for Transport Layer Security" is selected, the TSF shall be validated against the TLS Functional Package, with the following selections made:

- FCS_TLS_EXT.1:
  - either TLS or DTLS is selected as appropriate
  - client is selected
- FCS_TLSC_EXT.1.1 or FCS_DTLSC_EXT.1.1 (as appropriate):
  - The cipher suites selected must correspond with the algorithms and hash functions allowed in FCS_COP.1.
  - Mutual authentication must be selected
- FCS_TLSC_EXT.1.3 or FCS_DTLSC_EXT.1.3 (as appropriate):
  - With no exceptions is selected.

If the ST author selects IPsec, the TSF shall be validated against the PP-Module for VPN Client.

Appendix B - Selection-Based Requirements contains the requirements for implementing each of the other optional trusted channel protocols. The ST author must include the security functional requirements for the trusted channel protocol selected in FTP_ITC_EXT.1 in the main body of the ST.

Assured identification of endpoints is performed according to the authentication mechanisms used by the listed trusted channel protocols.

The TSF shall permit the TSF to initiate communication via the trusted channel.

The TSF shall initiate communication via the trusted channel for wireless access point connections, administrative communication, configured enterprise connections, and [**selection**: *OTA updates*, *no other connections*].

The developer shall provide a functional specification.

The developer shall provide a tracing from the functional specification to the SFRs.

Application Note: As indicated in the introduction to this section, the functional specification is comprised of the information contained in the AGD_OPE, AGD_PRE, and the API information that is provided to application developers, including the APIs that require privilege to invoke.

The developer may reference a website accessible to application developers and the evaluator. The API documentation shall include those interfaces required in this profile. The API documentation shall clearly indicate to which products and versions each available function applies.

The assurance activities in the functional requirements point to evidence that should exist in the documentation and TSS section; since these are directly associated with the SFRs, the tracing in element ADV_FSP.1.2D is implicitly already done and no additional documentation is necessary.

The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.

The functional specification shall provide rationale for the implicit categorization of interfaces as SFR-non-interfering.

The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

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

The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

The developer shall provide operational user guidance.

Application Note: The operational user guidance does not have to be contained in a single document. Guidance to users, administrators and application developers can be spread among documents or web pages. Where appropriate, the guidance documentation is expressed in the eXtensible Configuration Checklist Description Format (XCCDF) to support security automation.

Rather than repeat information here, the developer should review the assurance activities for this component to ascertain the specifics of the guidance that the evaluator will be checking for. This will provide the necessary information for the preparation of acceptable guidance.

The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

Application Note: User and administrator (e.g., MDM agent), and application developer are to be considered in the definition of user role.

The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

The operational user guidance shall identify all possible modes of operation of the OS (including operation following failure or operational error), their consequences, and implications for maintaining secure operation.

The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.

The operational user guidance shall be clear and reasonable.

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

The developer shall provide the TOE, including its preparative procedures.

Application Note: As with the operational guidance, the developer should look to the assurance activities to determine the required content with respect to preparative procedures.

The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

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

The evaluator shall apply the preparative procedures to confirm that the OS can be prepared securely for operation.

The developer shall provide the TOE and a reference for the TOE.

The TOE shall be labeled with a unique reference.

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

The developer shall provide a configuration list for the TOE.

The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.

The configuration list shall uniquely identify the configuration items.

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

The developer shall provide a description in the TSS of how timely security updates are made to the TOE.

The description shall include the process for creating and deploying security updates for the TOE software.

The description shall express the time window as the length of time, in days, between public disclosure of a vulnerability and the public availability of security updates to the TOE.

The description shall include the mechanisms publicly available for reporting security issues pertaining to the TOE.

The description shall include where users can seek information about the availability of new updates including details (e.g. CVE identifiers) of the specific public vulnerabilities corrected by each update.

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

The developer shall provide the TOE for testing.

The TOE shall be suitable for testing.

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

The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

The developer shall provide the TOE for testing.

The TOE shall be suitable for testing.

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

The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

Application Note: Public domain sources include the Common Vulnerabilities and Exposures (CVE) dictionary for publicly-known vulnerabilities.

The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

Selection-Based Security Functional Requirements

Extended: Cryptographic Key Support (REK)

A REK shall not be able to be read from or exported from the hardware.

Application Note: If "mutable-hardware" is selected in FCS_CKM_EXT.1.1, FCS_CKM_EXT.7 must be included in the ST. Note that if "immutable-hardware" is selected in FCS_CKM_EXT.1.1 it implicitly meets FCS_CKM_EXT.7.

The lack of a public/documented API for importing or exporting, when a private/undocumented API exists, is not sufficient to meet this requirement.

Extended: Security access control

The TSF shall provide a separate [**selection**: *address book*, *calendar*, *keystore*, *account credential database, [**assignment**: list of additional resources]*] for each application group and only allow applications within that process group to access the resource. Exceptions may only be explicitly authorized for such sharing by [**selection**: *the user*, *the administrator*, *no one*].

Application Note: If "groups of applications" is selected in FDP_ACF_EXT.1.2, FDP_ACF_EXT.2 must be included in the ST.

Extended: Storage of Critical Biometric Parameters

The TSF shall protect the authentication template [**selection**: *using a PIN as an additional factor*, *using a password as an additional factor*, *[**assignment**: other circumstances]*].

Application Note: If a BAF or "hybrid" is selected in FIA_UAU.5.1, FDP_PBA_EXT.1.1 must be included in the ST. If "hybrid" is selected in FIA_UAU.5.1, then "using a PIN as an additional factor" or "using a password as an additional factor" shall be selected. If "hybrid" is not selected in FIA_UAU.5.1, then the authentication template shall be secured by other means, which should be specified in the assignment. Since compromised authentication templates can be used in generating presentation/spoof attacks, it is important to utilize secure methods for protecting them.

Extended: Accuracy of Biometric Authentication

The one-attempt BAF False Accept Rate (FAR) for [**assignment**: *biometric modality selected in FIA_UAU.5.1*]shall not exceed [**assignment**: *claimed FAR no greater than 1:100*] with a one-attempt BAF False Reject Rate (FRR) not to exceed 1 in [**assignment**: *claimed FRR no greater than 1:10*].

Application Note: If a BAF or "hybrid" is selected in FIA_UAU.5.1, FIA_BMG_EXT.1.1 must be included in the ST. The assignment shall be completed for each biometric modality selected in FIA_UAU.5.1. If multiple biometric modalities are selected in FIA_UAU.5.1, it is acceptable for each modality to have a different FAR and FRR.

The False Accept Rate (FAR) is the measure of the likelihood that the biometric will incorrectly accept an authentication attempt by an unauthorized user. A system's FAR typically is stated as the proportion of verification transactions with wrongful claims of identity that are incorrectly confirmed.

The False Reject Rate (FRR) is the measure of the likelihood that the biometric security system will incorrectly reject an authentication attempt by an authorized user. A system's FRR typically is stated as the proportion of verification transactions with truthful claims of identity that are incorrectly denied.

Please note that without the use of hybrid authentication, multiple authentication attempts for a BAF that is claimed to have a one-attempt FAR between 1:100 and 1:500 inclusive will not produce an acceptable SAFAR in meeting FIA_BMG_EXT.1.2. More generally, depending on the number of authentication attempts allowed for the BAF, the claimed FAR must be strong (or equivalently, low) enough so that the SAFAR chosen in FIA_BMG_EXT.1.2 can be met within the 1% margin mandated.

Generally testing environments for a biometric system in a mobile device are based on a single legitimate user enrolling and test subjects attempt to authenticate. Since a thorough evaluation for FAR and FRR meeting all the conditions of statistical independence is not feasible in the timeframe of CC evaluations and in agreement with ISO/IEC 19795, the use of offline testing is acceptable even if this causes the biometric system to deviate slightly from the evaluated configuration. Additionally, full cross-comparison (i.e. all test subjects are compared to non-self) is acceptable.

Detailed explanations corresponding to the testing environments that are acceptable, to include the number of trials needed, can be found in .

The overall System Authentication False Accept Rate (SAFAR) shall be no greater than 1 in [**assignment**: *a SAFAR*

no greater than 1:500] within a 1% margin.

Application Note: If a BAF or "hybrid" is selected in FIA_UAU.5.1, FIA_BMG_EXT.1.2 must be included in the ST.

System Authentication False Accept Rate (SAFAR) is defined by the combination of individual error rates for each authentication factor and attempts used for access to a single session on the device.

Accessing a single session may involve a single authentication factor, in which case the SAFAR for a single attempt will be equal to the false accept rate (FAR) of that authentication factor and the SAFAR for n attempts will be $1-\left(1-FAR\right)^n$, assuming independence.

Accessing a single session on the device may involve the ability to use multiple authentication factors. It may be the case that only one authentication factor is needed to access a single session on the device (i.e. both a password and a BAF can be used, but only one is needed) or that both authentication factors are needed to access a single session on the device (i.e. both the BAF and a PIN must be entered). The full equations for calculating the SAFAR can be found in . A fully worked-out example that applies the equations in for calculating the SAFAR can be found in .

The worst-case scenario shall be used to calculate the SAFAR. Thus the authentication factor with the highest FAR shall be used for the maximum number of authentication attempts allowed for that factor. If any authentication attempts remain, then the authentication factor with the second highest FAR is used for the maximum number of authentication attempts allowed for that factor and so on. For example, the TOE supports a password and a BAF, the FAR for the BAF is higher than the FAR for the password and each authentication factor utilizes a shared counter per FIA_AFL_EXT.1. Then the worst-case scenario is the BAF is utilized for the maximum number of authentication attempts allowed for the BAF. For any remaining authentication attempts allowed the password is utilized.

Another example is the TOE supports a password and two BAFs, where the BAFs have different FARs, with both FARs being higher than the password FAR. Then the worst-case scenario is that the BAF with the highest FAR is used for the maximum number of authentication attempts allowed for that BAF, followed by the second BAF if any authentication attempts are allowed for that BAF. If any authentication attempts remain, then the password is utilized for those attempts.

The 1% margin is included for cases where a BAF is not a critical authentication factor and thus both BAF and password can be used in a session without exceeding the declared SAFAR.

Extended: TSF Integrity Testing

The TSF shall not execute code if the code signing certificate is deemed invalid.

Application Note: Certificates may optionally be used for code signing for integrity verification (FPT_TST_EXT.2.1/PREKERNEL). If "code signing for integrity verification" is selected in FIA_X509_EXT.2.1, FPT_TST_EXT.3.1 must be included in the ST.

Validity is determined by the certificate path, the expiration date, and the revocation status in accordance with RFC 5280.

Extended: Trusted Update Verification

The TSF shall not install code if the code signing certificate is deemed invalid.

Application Note: Certificates may optionally be used for code signing of system software updates (FPT_TUD_EXT.2.3) and of mobile applications (FPT_TUD_EXT.4.1). This element must be included in the ST if certificates are used for either update element. If either "code signing for system software updates" or "code signing for mobile applications" is selected in FIA_X509_EXT.2.1, FPT_TUD_EXT.3.1 must be included in the ST.

Validity is determined by the certificate path, the expiration date, and the revocation status in accordance with RFC 5280.

Objective Security Functional Requirements

Audit Review

The TSF shall provide the administrator with the capability to read all audited events and record contents from the audit records.

Application Note: The administrator shall have access to read the audit record, perhaps through an API or via an MDM Agent, which transfers the local records stored on the TOE to the MDM Server where the enterprise administrator may view them. If this requirement is included in the ST, function shall be included in the selection of FMT_SMF_EXT.1.

The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

Selective Audit

The TSF shall be able to select the set of events to be audited from the set of all auditable events based on the following attributes [**selection**:

- *event type*,
- *success of auditable security events*,
- *failure of auditable security events*,
- *[**assignment**: other attributes]*

].

Application Note: The intent of this requirement is to identify all criteria that can be selected to trigger an audit event. This can be configured through an interface on the TSF for a user or administrator to invoke. For the ST author, the assignment is used to list any additional criteria or "none".

Extended: Cryptographic Operation (Random Bit Generation)

The TSF shall save the state of the deterministic RBG at power-off, and shall use this state as input to the deterministic RBG at startup.

Application Note: The capability to add the state saved at power-off as input to the RBG prevents an RBG that is slow to gather entropy from producing the same output regularly and across reboots. Since there is no guarantee of the protections provided when the state is stored (or a requirement for any such protection), it is assumed that the state is 'known', and therefore cannot contribute entropy to the RBG, but can introduce enough variation that the initial RBG values are not predictable and exploitable.

Extended: Cryptographic Operation (Random Bit Generation)

The TSF shall allow applications to add data to the deterministic RBG using the Personalization String as defined in SP 800-90A.

Application Note: As specified in SP 800-90A the TSF shall not count data input from an application towards the entropy required by FCS_RBG_EXT.1. Thus, the TSF shall not allow the only input to the RBG seed to be from an application.

Extended: Cryptographic Algorithm Services

The TSF shall provide a mechanism for applications to request the TSF to perform the following cryptographic operations:

- Algorithms in FCS_COP.1/ENCRYPT
- Algorithms in FCS_COP.1/SIGN

by keys stored in the secure key storage.

Application Note: The TOE will, therefore, be required to perform cryptographic operations on behalf of applications using the keys stored in the TOE’s secure key storage.

Extended: Security attribute based access control

The TSF shall enforce an access control policy that prohibits an application from granting both write and execute permission to a file on the device except for [**selection**: *files stored in the application's private data folder*, *no exceptions*].

Extended: Application Backup

The TSF shall provide a mechanism for applications to mark [**selection**: *all application data*, *selected application data*] to be excluded from device backups.

Application Note: Device backups include any mechanism built into the TOE that allows stored application data to

be extracted over a physical port or sent over the network, but does not include any functionality implemented by a specific application itself if the application is not included in the TOE. The lack of a public/documented API for performing backups, when a private/undocumented API exists, is not sufficient to meet this requirement.

Extended: Biometric Enrollment

The TSF shall only use biometric samples of sufficient quality for enrollment. Sample data shall have [**assignment**: *quality metrics corresponding to each biometric modality*].

Application Note: Different biometric modalities utilize different quality standards. The quality standard for the each BAF selected in FIA_UAU.5 should be listed in the assignment. For example, fingerprint may utilize the NFIQ standard where NFIQ 1.0 scores of 1, 2, or 3 are required for use in hardware PIV, where 1 is the highest quality standard. NFIQ 2.0 is a newer version of the NFIQ standard that has not seen widespread adoption as of the publication of this PP but is being considered by the scientific community as well as by industry. Samples used to create the authentication template/profile at enrollment shall be mutually consistent. After the authentication template has been created, it shall be tested to determine whether or not it is of sufficient quality and if not, more quality samples shall be added until it is of sufficient quality.

Extended: Biometric Verification

The TSF shall only use biometric samples of sufficient quality for verification. As such, sample data shall have [**assignment**: *quality metrics corresponding to each biometric modality*].

Application Note: Different biometric modalities utilize different quality standards. The quality standard for the each BAF selected in FIA_UAU.5 should be listed in the assignment. For example, fingerprint may utilize the NFIQ standard where NFIQ 1.0 scores of 1, 2, or 3 are required for use in hardware PIV, where 1 is the highest quality standard. NFIQ 2.0 is a newer version of the NFIQ standard that has not seen widespread adoption as of the publication of this PP but is being considered by the scientific community as well as by industry.

Extended: Biometric Templates

The TSF shall only generate and use enrollment templates and/or authentication templates of sufficient quality for any subsequent authentication functions.

Application Note: If the vendor needs to develop an authentication template using multiple enrollment samples, they shall all be mutually consistent and correspond to the biometric characteristics of a single user and source. For the purposes of this requirement, enrollment templates are templates constructed from sample data, while authentication templates are generated based on sample data and/or enrollment templates and stored for matching/biometric verification purposes. One or more templates could be generated during enrollment without the user knowing how many.

Authentication templates may not have standard quality metrics, but vendor and/or labs still need to ensure that such templates have a sufficient feature set available to provide a desired identity assurance level. Examples include minimum number of fingerprint minutiae.

Extended: Handling Unusual Biometric Templates

The matching algorithm shall handle properly formatted enrollment templates and/or authentication templates, especially those with unusual data properties, appropriately. If such templates contain incorrect syntax, are of low quality, or contain enrollment data considered unrealistic for a given modality, then they shall be rejected by the matching algorithm and an error code shall be reported.

Application Note: While it is important to have properly formatted enrollment or authentication templates, it is equally important for the matching algorithm to correctly handle enrollment and/or authentication templates that have unusual data properties or are of low quality. If the matching algorithm detects templates that are of low quality, have low numbers of bits of complexity, or maintain unusual data properties, it shall return an error code or other indication in order to protect the system from possible spoofing or denial-of-service attacks. For the purposes of this requirement, enrollment templates are templates constructed from sample data, while authentication templates are stored for matching/biometric verification purposes.

Examples of unusual data properties that may cause fingerprint enrollment template rejection include, but are not limited to, minutia counts that are too high or too low, direction field maps that do not correspond to real fingerprint ridge flow maps, all detected minutia crowded to the extreme edges of the image area, and ridge widths that are too wide or too narrow.

Accordingly, if an enrollment template and/or authentication template meets the structural requirements but without proper syntax, the matching algorithm shall similarly return an error code or other indication to similar effect.

Extended: Spoof Detections for Biometrics

The TSF shall perform Presentation Attack Detection testing up to the attack potential of [**selection**: *basic*, *intermediate*, *advanced*] attacks, for each biometric modalities selected in FIA_UAU.5.1 on each enrollment and authentication attempt, rejecting detected spoofs. When an authentication attempt fails due to PAD testing, the TSF shall not indicate to the user the reason for failure to authenticate.

Application Note: Presentation Attack Detection (PAD) is also known as liveness detection or spoof detection. If multiple modalities are selected in FIA_UAU.5.1, then this SFR shall be iterated for each modality. For each modality, only one attack strength shall be selected.

Because Presentation Attack Detection (PAD) is an open-ended problem much like vulnerability testing, it is neither cost-effective nor feasible to create a complete list of attack vectors and perform testing on all of them during the timeframe for CC evaluations. Such a list would be ever-changing, and unlike code vulnerabilities (i.e. CVEs), the equipment, skill, time, and cost required to test highly sophisticated attacks is highly infeasible for a testing lab given the current timeframe for CC evaluations. Nevertheless, it is a known risk that has been documented by researchers for years.

Therefore, vendors are responsible for providing their own documentation specifying the measures the TSF takes to mitigate presentation attacks and the appropriate pen-testing (for example, red teaming and blue teaming) performed as proof.

To be specific, basic attacks (including basic and enhanced-basic [IBPC]) refer to attacks in literature of low skill that can be performed on a limited budget. This includes, but is not limited to, playback attacks of a spoken utterance using a different mobile device for voice authentication, taking a photograph of a fingerprint or facial and submitting it to the sensor, among other examples.

Intermediate (or moderate [IBPC]) attacks can include, but are not limited to, creating a foam finger to thwart fingerprint detection and using a higher quality playback device to thwart liveness detection.

Advanced (including high and beyond high [IBPC]) attacks can include, but are not limited to, creating a synthetic hand with the given fingerprint using an expensive 3D-printer and forcing someone to reveal one's credentials through coercion or threats that may cause harm (where detection of duress is required). Many of these attack techniques may be sensitive or government classified.

Extended: X509 certificate enrollment

The TSF shall use the Enrollment over Secure Transport (EST) protocol as specified in RFC 7030 to request certificate enrollment using the simple enrollment method described in RFC 7030 Section 4.2.

The TSF shall be capable of authenticating EST requests using an existing certificate and corresponding private key as specified by RFC 7030 Section 3.3.2.

The TSF shall be capable of authenticating EST requests using HTTP Basic Authentication with a username and password as specified by RFC 7030 Section 3.2.3.

The TSF shall perform authentication of the EST server using an Explicit Trust Anchor following the rules described in RFC 7030, section 3.6.1.

Application Note: EST also uses HTTPS as specified in FCS_HTTPS_EXT.1 to establish a secure connection to an EST server. The separate Trust Anchor Database dedicated to EST operations is described as Explicit Trust Anchors in RFC 7030.

The TSF shall be capable of requesting server-provided private keys as specified in RFC 7030 Section 4.4.

The TSF shall be capable of updating its EST-specific Trust Anchor Database using the "Root CA Key Update" process described in RFC 7030 Section 4.1.3.

The TSF shall generate a Certificate Request Message for EST as specified in RFC 2986 and be able to provide the following information in the request: public key and [**selection**: *device-specific information*, *Common Name*, *Organization*, *Organizational Unit*, *Country*].

Application Note: The public key referenced is the public key portion of the public-private key pair generated by the TOE as specified in FCS_CKM.1.

The TSF shall validate the chain of certificates from the Root CA certificate in the Trust Anchor Database to the EST Server CA certificate upon receiving a CA Certificates Response.

Extended: X509 certificate enrollment

The TSF shall generate a Certificate Request Message as specified in RFC 2986 and be able to provide the following

information in the request: public key and [**selection**: *device-specific information*, *Common Name*, *Organization*, *Organizational Unit*, *Country*].

Application Note: The public key referenced in FIA_X509_EXT.5.1 is the public key portion of the public-private key pair generated by the TOE as specified in FCS_CKM.1. The trusted channel requirements do not apply to communication with the CA for the certificate request/response messages.

As Enrollment over Secure Transport (EST) is a new standard that has not yet been widely adopted, this requirement is included as an interim objective requirement in order to allow developers to distinguish those products which have do have the ability to generate Certificate Request Messages but do not yet implement EST.

The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

Extended: Current Administrator

The TSF shall provide a mechanism that allows users to view a list of currently authorized administrators and the management functions that each administrator is authorized to perform.

Extended: Anti-Exploitation Services (ASLR)

The TSF shall provide address space layout randomization (ASLR) to the kernel.

The base address of any kernel-space memory mapping will consist of at least 4 unpredictable bits.

Application Note: The 4 unpredictable bits may be provided by the TSF RBG (as specified in FCS_RBG_EXT.1).

Extended: Anti-Exploitation Services (Memory Page Permissions)

The TSF shall prevent write and execute permissions from being simultaneously granted to any page of physical memory [**selection**: *with no exceptions*, *[**assignment**: specific exceptions]*].

Application Note: Memory used for just-in-time (JIT) compilation is anticipated as an exception in this requirement; if so, the ST author must address how this exception is permitted. It is expected that the memory management unit will transition the system to a non-operational state if any violation is detected in kernel memory space.

Extended: Anti-Exploitation Services (Overflow Protection)

The TSF shall include heap-based buffer overflow protections in the runtime environment it provides to processes that execute on the application processor.

Application Note: These heap-based buffer overflow protections are expected to ensure the integrity of heap metadata such as memory addresses or offsets recorded by the heap implementation to manage memory blocks. This includes chunk headers, look-aside lists, and other data structures used to track the state and location of memory blocks managed by the heap.

Extended: Application Processor Mediation

The TSF shall prevent code executing on any baseband processor (BP) from accessing application processor (AP) resources except when mediated by the AP.

Application Note: These resources include:

- Volatile and non-volatile memory
- Control of and data from integrated and non-integrated peripherals (e.g. USB controllers, touch screen controllers, LCD controller, codecs)
- Control of and data from integrated and non-integrated I/O sensors (e.g. camera, light, microphone, GPS, accelerometers, geomagnetic field sensors)

Mobile devices are becoming increasingly complex having an application processor that runs a rich operating system and user applications and separate baseband processor(s) that handle cellular and other wireless network connectivity.

- The application processor within most modern Mobile Devices is a system on a chip (SoC) that integrates, for example, CPU/GPU cores and memory interface electronics into a single, power-efficient package.
- Baseband processors are becoming increasingly complex themselves delivering voice encoding alongside

multiple independent radios (LTE, Wi-Fi, Bluetooth, FM, GPS) in a single package containing multiple CPUs and DSPs.

Thus, the baseband processor(s) in these requirements include such integrated SoCs and include any radio processors (integrated or not) on the Mobile Device.

All other requirements mostly, except where noted, apply to firmware/software on the application processor, but future requirements (notably, all Integrity, Access Control, and Anti-Exploitation requirements) will apply to application processors and baseband processors.

Extended: Limitation of Bluetooth Profile Support

The TSF shall disable support for [**assignment**: *list of Bluetooth profiles*] Bluetooth profiles when they are not currently being used by an application on the Mobile Device, and shall require explicit user action to enable them.

Application Note: Some Bluetooth services incur more serious consequences if unauthorized remote devices gain access to them. Such services should be protected by measures like disabling support for the associated Bluetooth profile unless it is actively being used by an application on the Mobile Device (in order to prevent discovery by a Service Discovery Protocol search), and then requiring explicit user action to enable those profiles in order to use the services. It may be further appropriate to require additional user action before granting a remote device access to that service.

For example, it may be appropriate to disable the OBEX Push Profile until a user on the Mobile Device pushes a button in an application indicating readiness to transfer an object. After completion of the object transfer, support for the OBEX profile should be suspended until the next time the user requests its use.

The ST author shall list all Bluetooth profiles which are disabled while not in use by an application and which need explicit user action in order to become enabled.

Extended: Self-Test Notification

The TSF shall [**selection**: *audit*, *provide the administrator with*] TSF-software integrity verification values.

Application Note: These notifications are typically called remote attestation and these integrity values are typically called measurements. The integrity values are calculated from hashes of critical memory and values, including executable code. The ST author shall select whether these values are logged as a part of FAU_GEN.1.1 or are provided to the administrator.

The TSF shall cryptographically sign all integrity verification values.

Application Note: The intent of this requirement is to provide assurance to the administrator that the responses provided are from the TOE and have not been modified or spoofed by a man-in-the-middle such as a network-based adversary or a malicious MDM Agent.

Extended: TSF Integrity Checking

The TSF shall verify the integrity of [**selection**: *all executable code stored in mutable media*, *[**assignment**: list of other executable code]*], stored in mutable media prior to its execution through the use of [**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*, *hardware-protected hash*].

Application Note: All executable code covered in this requirement is executed after the kernel is loaded.

If "all executable code in mutable media" is verified, implementation in hardware or in read-only memory is a natural logical consequence.

At this time, the verification of software executed on other processors stored in mutable media is not required; however, it may be added in the first assignment. If all executable code (including bootloader(s), kernel, device drivers, pre-loaded applications, user-loaded applications, and libraries) is verified, "all executable code stored in mutable media" should be selected.

Extended: Application Verification

The TSF shall by default only install mobile applications cryptographically verified by [**selection**: *a built-in X.509v3 certificate*, *a configured X.509v3 certificate*].

Application Note: The built-in certificate is installed by the manufacturer either at time of manufacture or as a part of system updates. The configured certificate used to verify the signature is set according to FMT_SMF_EXT.1 function
.

Extended: Trusted Update Verification

The TSF shall verify that software updates to the TSF are a current or later version than the current version of the TSF.

Application Note: A later version has a larger version number. The method for distinguishing newer software versions from older versions is determined by the manufacturer.

Default TOE Access Banners

Before establishing a user session, the TSF shall display an advisory warning message regarding unauthorized use of the TOE.

Application Note: This requirement may be met with the configuration of either text or an image containing the text of the desired message. The TSF shall minimally display this information at startup, but may also display the information at every unlock. The banner is configured according to FMT_SMF_EXT.1 function .

Optional Security Functional Requirements

Extended: Secondary User Authentication

The TSF shall provide a secondary authentication mechanism for accessing Enterprise applications and resources. The secondary authentication mechanism shall control access to the Enterprise application and shared resources and shall be incorporated into the encryption of protected and sensitive data belonging to Enterprise applications and shared resources.

Application Note: For the BYOD use case, Enterprise applications and data shall be protected using a different password than the user authentication to gain access to the personal applications and data, if configured.

This requirement shall be included in the ST if the TOE implements a container solution, in which there is a separate authentication, to separate user and Enterprise applications and resources.

The TSF shall require the user to present the secondary authentication factor prior to decryption of Enterprise application data and Enterprise shared resource data.

Application Note: This requirement must be selected if FIA_UAU_EXT.4.1 is selected. The intent of this requirement is to prevent decryption of protected Enterprise application data and Enterprise shared resource data before the user has authenticated to the device using the secondary authentication factor. Enterprise shared resource data consists of the FDP_ACF_EXT.2.1 selections.