Supporting Document
Mandatory Technical Document

Protection Profile for Mobile Devices
Version: 3.2
2021-04-15
National Information Assurance Partnership

Foreword

This is a Supporting Document (SD), intended to complement the Common Criteria version 3 and the associated Common Evaluation Methodology for Information Technology Security Evaluation.

SDs may be “Guidance Documents”, that highlight specific approaches and application of the standard to areas where no mutual recognition of its application is required, and as such, are not of normative nature, or “Mandatory Technical Documents”, whose application is mandatory for evaluations whose scope is covered by that of the SD. The usage of the latter class is not only mandatory, but certificates issued as a result of their application are recognized under the CCRA.

Technical Editor:
National Information Assurance Partnership (NIAP)

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
		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.

3.1	2017-04-05	• 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	2021-04-15	Removed TLS SFRs and utilized TLS Functional Package Removed Bluetooth SFRs and utilized Bluetooth Module. Bluetooth SFR moved to Implementation Dependent. FPT_TUD_EXT.2.4 renumbered to FPT_TUD_EXT.3.1. FPT_TUD_EXT.3 renumbered to FPT_TUD_EXT.4. FPT_TUD_EXT.4.1 renumbered to FPT_TUD_EXT.5.1. FPT_TUD_EXT.4.2 renumbered to FPT_TUD_EXT.6.1.

General Purpose:
The purpose of this SD is to define evaluation methods for the functional behavior of Mobile Device products.

Acknowledgments:
This SD was developed with support from NIAP Mobile Devices Technical Community members, with representatives from industry, government agencies, Common Criteria Test Laboratories, and members of academia.

Contents

1 Introduction
1.1 Technology Area and Scope of Supporting Document
1.2 Structure of the Document
1.3 Terms
1.3.1 Common Criteria Terms
1.3.2 Technical Terms
2 Evaluation Activities for SFRs
2.1 TOE SFR Evaluation Activities
2.1.1 Class: Security Audit (FAU)
2.1.2 Class: Cryptographic Support (FCS)
2.1.3 Cryptographic Storage (FCS_STG)
2.1.4 Class: User Data Protection (FDP)
2.1.5 Class: Identification and Authentication (FIA)
2.1.6 Class: Security Management (FMT)
2.1.7 Class: Protection of the TSF (FPT)
2.1.8 Class: TOE Access (FTA)
2.1.9 Class: Trusted Path/Channels (FTP)
2.2 Evaluation Activities for Strictly Optional SFRS
2.2.1 Class: Identification and Authentication (FIA)
2.3 Evaluation Activities for Objective SFRS
2.3.1 Class: Security Audit (FAU)
2.3.2 Class: Cryptographic Support (FCS)
2.3.3 Class: User Data Protection (FDP)
2.3.4 Class: Identification and Authentication (FIA)
2.3.5 Class: Security Management (FMT)
2.3.6 Class: Protection of the TSF (FPT)
2.3.7 Class: TOE Access (FTA)
2.4 Evaluation Activities for Implementation-based SFRS
2.4.1 Class: User Data Protection (FDP)
2.5 Evaluation Activities for Selection-based SFRS
2.5.1 Class: User Data Protection (FDP)
3 Evaluation Activities for SARs
4 Required Supplementary Information
Appendix A - Bibliography

1 Introduction

1.1 Technology Area and Scope of Supporting Document

The scope of the Protection Profile for Mobile Devices is to describe the security functionality of Mobile Devices products in terms of [CC] and to define functional and assurance requirements for them.

Although Evaluation Activities are defined mainly for the evaluators to follow, in general they also help

developers to prepare for evaluation by identifying specific requirements for their TOE. The specific requirements in Evaluation Activities may in some cases clarify the meaning of Security Functional Requirements (SFR), and may identify particular requirements for the content of Security Targets (ST) (especially the TOE Summary Specification), user guidance documentation, and possibly supplementary information (e.g. for entropy analysis or cryptographic key management architecture).

1.2 Structure of the Document

Evaluation Activities can be defined for both SFRs and Security Assurance Requirements (SAR), which are themselves defined in separate sections of the SD.

If any Evaluation Activity cannot be successfully completed in an evaluation, then the overall verdict for the evaluation is a 'fail'. In rare cases there may be acceptable reasons why an Evaluation Activity may be modified or deemed not applicable for a particular TOE, but this must be approved by the Certification Body for the evaluation.

In general, if all Evaluation Activities (for both SFRs and SARs) are successfully completed in an evaluation then it would be expected that the overall verdict for the evaluation is a ‘pass’. To reach a ‘fail’ verdict when the Evaluation Activities have been successfully completed would require a specific justification from the evaluator as to why the Evaluation Activities were not sufficient for that TOE.

Similarly, at the more granular level of assurance components, if the Evaluation Activities for an assurance component and all of its related SFR Evaluation Activities are successfully completed in an evaluation then it would be expected that the verdict for the assurance component is a ‘pass’. To reach a ‘fail’ verdict for the assurance component when these Evaluation Activities have been successfully completed would require a specific justification from the evaluator as to why the Evaluation Activities were not sufficient for that TOE.

1.3 Terms

The following sections list Common Criteria and technology terms used in this document.

1.3.1 Common Criteria Terms

Assurance	Grounds for confidence that a TOE meets the SFRs [CC].
Base Protection Profile (Base-PP)	Protection Profile used as a basis to build a PP-Configuration.
Collaborative Protection Profile (cPP)	A Protection Profile developed by international technical communities and approved by multiple schemes.
Common Criteria (CC)	Common Criteria for Information Technology Security Evaluation (International Standard ISO/IEC 15408).
Common Criteria Testing Laboratory	Within the context of the Common Criteria Evaluation and Validation Scheme (CCEVS), an IT security evaluation facility accredited by the National Voluntary Laboratory Accreditation Program (NVLAP) and approved by the NIAP Validation Body to conduct Common Criteria-based evaluations.
Common Evaluation Methodology (CEM)	Common Evaluation Methodology for Information Technology Security Evaluation.
Distributed TOE	A TOE composed of multiple components operating as a logical whole.
Extended Package (EP)	A deprecated document form for collecting SFRs that implement a particular protocol, technology, or functionality. See Functional Packages.
Functional Package (FP)	A document that collects SFRs for a particular protocol, technology, or functionality.
Operational Environment (OE)	Hardware and software that are outside the TOE boundary that support the TOE functionality and security policy.
Protection Profile (PP)	An implementation-independent set of security requirements for a category of products.
Protection Profile	

Configuration (PP-Configuration)	A comprehensive set of security requirements for a product type that consists of at least one Base-PP and at least one PP-Module.
Protection Profile Module (PP-Module)	An implementation-independent statement of security needs for a TOE type complementary to one or more Base-PPs.
Security Assurance Requirement (SAR)	A requirement to assure the security of the TOE.
Security Functional Requirement (SFR)	A requirement for security enforcement by the TOE.
Security Target (ST)	A set of implementation-dependent security requirements for a specific product.
Target of Evaluation (TOE)	The product under evaluation.
TOE Security Functionality (TSF)	The security functionality of the product under evaluation.
TOE Summary Specification (TSS)	A description of how a TOE satisfies the SFRs in an ST.

1.3.2 Technical Terms

Adaptive Template	A type of authentication template that evolves with each sample that is verified and introduced into the biometrics database or gallery.
Address Space Layout Randomization (ASLR)	An anti-exploitation feature, which loads memory mappings into unpredictable locations. ASLR makes it more difficult for an attacker to redirect control to code that they have introduced into the address space of a process or the kernel.
Administrator	The Administrator is responsible for management activities, including setting the policy that is applied by the enterprise on the Mobile Device. This administrator is likely to be acting remotely and could be the Mobile Device Management (MDM) Administrator acting through an MDM Agent. If the device is unenrolled, the user is the administrator.
Authentication Template	A digital representation of an individual’s distinct characteristics, representing information extracted from a biometric sample. Such templates are used during biometric authentication and verification as the basis for comparison. Unlike enrollment templates, these templates can be adaptive.
Auxiliary Boot Modes	Auxiliary boot modes are states in which the device provides power to one or more components to provide an interface that enables an unauthenticated user to interact with either a specific component or several components that exist outside of the device’s fully authenticated, operational state.
Biometric Authentication Factor (BAF)	Authentication factor, which uses biometric sample, matched to a biometric authentication template to help establish identity.
Biometric Data	Digital data created during a biometric process. It encompasses raw sensor observations, biometric samples, models, templates, and/or similarity scores, among other data. This data is used to describe the information collected during an enrollment, verification, or identification process, but does not apply to end user information such as user name, password (unless tied to the biometric modality), demographic information, and authorizations.
Biometric Sample	Information or computer data obtained from a biometric sensor device or captured from an individual to the sensor.
	Multiple individual components (such as sensor, matching algorithm, and result display) that combine to make a fully operational system completely contained within the TOE. A biometric system is automated and capable of:

Biometric System	1. Capturing a biometric sample from an end user 2. Extracting and processing the biometric data from that sample 3. Generating various templates based on processing of that sample during enrollment, or, if adaptive, during verification as well 4. Storing the extracted information in a database on the device 5. Comparing the biometric data with data contained in one or more authentication templates 6. Deciding how well they match and indicating whether or not an identification or verification of identity has been achieved.
Common Application Developer	Application developers (or software companies) often produce many applications under the same name. Mobile devices often allow shared resources by such applications where otherwise resources would not be shared.
Critical Security Parameter (CSP)	Security-related information whose disclosure or modification can compromise the security of a cryptographic module and/or authentication system.
Data	Program/application or data files that are stored or transmitted by a server or Mobile Device (MD).
Data Encryption Key (DEK)	A key used to encrypt data-at-rest.
Developer Modes	Developer modes are states in which additional services are available to a user in order to provide enhanced system access for debugging of software.
Encrypted Software Keys	These keys are stored in the main file system encrypted by another key and can be changed and sanitized.
Enrolled State	The state in which the Mobile Device is managed with active policy settings from the administrator.
Enrollment (Biometrics)	The process of collecting a biometric sample from an end user, converting it into an enrollment and/or authentication template, and storing it in the biometric system’s database. If an enrollment template is generated, it is used during the enrollment process for later comparison to other enrollment templates already stored. If there are multiple enrollment templates, they may be fused, averaged, or otherwise, in order to create authentication templates, which are used for later comparison in verification.
Enrollment Template	A digital representation of an individual’s distinct characteristics, representing information extracted from a biometric sample. Such templates are generated during the enrollment process and utilized in various ways (including averaging, fusion, etc.) in order to generate an authentication template.
Enterprise Applications	Applications that are provided and managed by the enterprise.
Enterprise Data	Enterprise data is any data residing in the enterprise servers, or temporarily stored on Mobile Devices to which the Mobile Device user is allowed access according to security policy defined by the enterprise and implemented by the administrator.
Ephemeral Keys	These keys are stored in volatile memory.
False Accept Rate (FAR)	A statistic used to measure biometric performance when operating in verification, defined as the percentage of times a system produces a false accept, which occurs when an individual is incorrectly matched to another individual’s existing biometric. For example, Mallory claims to be Alice and the system verifies the claim.
False Reject Rate (FRR)	A statistic used to measure biometric performance in verification, defined as the percentage of times the system produces a false reject. A false reject occurs when an individual is not matched to his or her own existing biometric template. For example, John claims to be John, but the system incorrectly denies the claim.
Feature(s) (Biometrics)	Distinctive mathematical characteristic(s) derived from a biometric sample, used to generate enrollment or authentication templates.
File Encryption Key (FEK)	A DEK used to encrypt a file or a director when File Encryption is used. FEKs are unique to each encrypted file or directory.
Hardware-	

Isolated Keys	The OS can only access these keys by reference, if at all, during runtime.
Hybrid Authentication	A hybrid authentication factor is one where a user has to submit a combination of a biometric sample and a PIN or password and both to pass. If either factor fails, the entire attempt fails. The user shall not be made aware of which factor failed, if either fails.
Immutable Hardware Key	These keys are stored as hardware-protected raw key and cannot be changed or sanitized.
Key Chaining	The method of using multiple layers of encryption keys to protect data. A top layer key encrypts a lower layer key, which encrypts the data; this method can have any number of layers.
Key Encryption Key (KEK)	A key used to encrypt other keys, such as DEKs or storage that contains keys.
Liveness Detection	A technique used to ensure that the biometric sample submitted is from an end user. A liveness detection method can help protect the system against some types of spoofing attacks.
Locked State	Powered on but most functionality is unavailable for use. User authentication is required to access functionality.
MDM Agent	The MDM Agent is installed on a Mobile Device as an application or is part of the Mobile Device's OS. The MDM Agent establishes a secure connection back to the MDM Server controlled by the administrator.
Minutia Point	Friction ridge characteristics that are used to individualize a fingerprint image. Minutia are the points where friction ridges begin, terminate, or split into two or more ridges. In many fingerprint systems, the minutia points are compared for recognition purposes.
Mobile Device (MD)	A device which is composed of a hardware platform and its system software. The device typically provides wireless connectivity and may include software for functions like secure messaging, email, web, VPN (Virtual Private Network) connection, and VoIP (Voice over IP), for access to the protected enterprise network, enterprise data and applications, and for communicating to other Mobile Devices.
Mobile Device Management (MDM)	Mobile device management (MDM) products allow enterprises to apply security policies to mobile devices. This system consists of two primary components: the MDM Server and the MDM Agent.
Mobile Device User (User)	The individual authorized to physically control and operate the Mobile Device. Depending on the use case, this can be the device owner or an individual authorized by the device owner.
Modality (Biometrics)	A type or class of biometric system, such as fingerprint recognition, facial recognition, iris recognition, voice recognition, signature/sign, and others.
Mutable Hardware Key	These keys are stored as hardware-protected raw key and can be changed or sanitized.
NIST Fingerprint Image Quality (NFIQ)	A machine-learning algorithm that reflects the predictive positive or negative contribution of an individual sample to the overall performance of a fingerprint matching system. NFIQ 1.0 scores are calculated on a scale from 1 to 5, where NFIQ = 1 indicates high quality samples and NFIQ = 5 indicates poor quality samples [NFIQ 1.0]. NFIQ 2.0 scores are calculated on a scale from 0 to 100, where NFIQ = 0 indicates poor quality samples and NFIQ = 100 indicates high quality samples [NFIQ 2.0].
Operating System (OS)	Software that runs at the highest privilege level and can directly control hardware resources. Modern Mobile Devices typically have at least two primary operating systems: one, which runs on the application processor and one, which runs on the cellular baseband processor. The OS of the application processor handles most user interactions and provides the execution environment for apps. The OS of the cellular baseband processor handles communications with the cellular network and may control other peripherals. The term OS, without context, may be assumed to refer to the OS of the application processor.
Password Authentication Factor	A type of authentication factor requiring the user to provide a secret set of characters to gain access.
PIN Authentication Factor	A PIN is a set of numeric or alphabetic characters that may be used in addition to a biometric factor to provide a hybrid authentication factor. At this time it is not considered as a stand-alone authentication mechanism. A PIN is distinct from a password in that the allowed character set and required length of a PIN is typically smaller than that of a password as it is designed to be input quickly.

Powered Off State	The device has been shut down such that no TOE function can be performed.
Presentation Attack Detection (PAD)	A technique used to ensure that the biometric sample submitted is from an end user. A presentation attack detection method can help protect the system against some types of spoofing attacks.
Protected Data (PD)	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.
Root Encryption Key (REK)	A key tied to the device used to encrypt other keys.
Sensitive data	Sensitive data shall be identified in the TSS section of the Security Target (ST) by the ST author. Sensitive data is a subset or all of the Protected data. Sensitive data may include all user or enterprise data or may be specific application data such as emails, messaging, documents, calendar items, and contacts. Sensitive data is protected while in the locked state (FDP_DAR_EXT.2).
Software Keys	The OS access the raw bytes of these keys during runtime.
Template (Biometrics)	A digital representation of an individual’s distinct characteristics, representing information extracted from a biometric sample. This PP further defines enrollment templates and authentication templates.
Threshold	A user setting for biometric systems operating in verification. Thresholds are also used in enrollment if enrollment templates are created and compared to each other. The acceptance or rejection of biometric data in verification is dependent on the match score falling above or below the threshold. The threshold is adjustable so that the biometric system can be more or less strict, depending on the requirements of any given biometric application.
Trust Anchor Database	A list of trusted root Certificate Authority certificates.
TSF Data	Data for the operation of the TSF upon which the enforcement of the requirements relies.
Unenrolled State	The state in which the Mobile Device is not managed.
Unlocked State	Powered on and device functionality is available for use. Implies user authentication has occurred (when so configured).
Verification (Biometrics)	A task where the biometric system attempts to confirm an individual’s claimed identity by comparing a submitted sample to one or more previously enrolled authentication templates.

2 Evaluation Activities for SFRs

The EAs presented in this section capture the actions the evaluator performs to address technology specific aspects covering specific SARs (e.g. ASE_TSS.1, ADV_FSP.1, AGD_OPE.1, and ATE_IND.1) – this is in addition to the CEM workunits that are performed in Section .

Regarding design descriptions (designated by the subsections labeled TSS, as well as any required supplementary material that may be treated as proprietary), the evaluator must ensure there is specific information that satisfies the EA. For findings regarding the TSS section, the evaluator’s verdicts will be associated with the CEM workunit ASE_TSS.1-1. Evaluator verdicts associated with the supplementary evidence will also be associated with ASE_TSS.1-1, since the requirement to provide such evidence is specified in ASE in the PP.

For ensuring the guidance documentation provides sufficient information for the administrators/users as it pertains to SFRs, the evaluator’s verdicts will be associated with CEM workunits ADV_FSP.1-7, AGD_OPE.1-4, and AGD_OPE.1-5.

Finally, the subsection labeled Tests is where the authors have determined that testing of the product in the context of the associated SFR is necessary. While the evaluator is expected to develop tests, there may be instances where it is more practical for the developer to construct tests, or where the developer may have existing tests. Therefore, it is acceptable for the evaluator to witness developer-generated tests in lieu of executing the tests. In this case, the evaluator must ensure the developer’s tests are executing both in the manner declared by the developer and as mandated by the EA. The CEM workunits that are associated with the EAs specified in this section are: ATE_IND.1-3, ATE_IND.1-4, ATE_IND.1-5, ATE_IND.1-6, and ATE_IND.1-7.

2.1 TOE SFR Evaluation Activities

2.1.1 Class: Security Audit (FAU)

FAU_GEN.1 Audit Data Generation

FAU_GEN.1
TSS
The evaluator shall check the TSS and ensure that it lists all of the auditable events and provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field. The evaluator shall check to make sure that every audit event type mandated by the PP is described and that the description of the fields contains the information required in FAU_GEN.1.2.

Guidance
The evaluator shall also make a determination of the administrative actions that are relevant in the context of this PP including those listed in the Management section. The evaluator shall examine the administrative guide and make a determination of which administrative commands are related to the configuration (including enabling or disabling) of the mechanisms implemented in the TOE that are necessary to enforce the requirements specified in the PP. The evaluator shall document the methodology or approach taken while determining which actions in the administrative guide are security relevant with respect to this PP. The evaluator may perform this activity as part of the activities associated with ensuring the AGD_OPE guidance satisfies the requirements.

Tests
The evaluator shall test the TOE's ability to correctly generate audit records by having the TOE generate audit records for the events listed in the provided table and administrative actions. This should include all instances of an event. The evaluator shall test that audit records are generated for the establishment and termination of a channel for each of the cryptographic protocols contained in the ST. For administrative actions, the evaluator shall test that each action determined by the evaluator above to be security relevant in the context of this PP is auditable. When verifying the test results, the evaluator shall ensure the audit records generated during testing match the format specified in the administrative guide, and that the fields specified in FAU_GEN.1.2 are contained in each audit record.

Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly. For example, testing performed to ensure that the administrative guidance provided is correct verifies that AGD_OPE.1 is satisfied and should address the invocation of the administrative actions that are needed to verify the audit records are generated as expected.

FAU_STG.1 Audit Storage Protection

FAU_STG.1
TSS
The evaluator shall ensure that the TSS lists the location of all logs and the access controls of those files such that unauthorized modification and deletion are prevented.

Guidance
There are no guidance evaluation activities for this component.

Tests

- Test FAU_STG.1:1:The evaluator shall attempt to delete the audit trail in a manner that the access controls should prevent (as an unauthorized user) and shall verify that the attempt fails.
- Test FAU_STG.1:2:The evaluator shall attempt to modify the audit trail in a manner that the access controls should prevent (as an unauthorized application) and shall verify that the attempt fails.

FAU_STG.4 Prevention of Audit Data Loss

FAU_STG.4
TSS
The evaluator shall examine the TSS to ensure that it describes the size limits on the audit records, the detection of a full audit trail, and the action(s) taken by the TSF when the audit trail is full. The evaluator shall ensure that the action(s) results in the deletion or overwrite of the oldest stored record.

Guidance
There are no guidance evaluation activities for this component.

Tests
There are no test evaluation activities for this component.

2.1.2 Class: Cryptographic Support (FCS)

FCS_CKM.1 Cryptographic Key Generation

FCS_CKM.1

TSS

The evaluator shall ensure that the TSS identifies the key sizes supported by the TOE. If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme.

Guidance

The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key generation scheme(s) and key size(s) for all uses defined in this PP.

Tests

Evaluation Activity Note: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products.

Key Generation for FIPS PUB 186-4 RSA Schemes

The evaluator shall verify the implementation of RSA Key Generation by the TOE using the Key Generation test. This test verifies the ability of the TSF to correctly produce values for the key components including the public verification exponent *e*, the private prime factors *p* and *q*, the public modulus n and the calculation of the private signature exponent *d*.

Key Pair generation specifies 5 ways (or methods) to generate the primes *p* and *q*. These include:

1. Random Primes:
    - Provable primes
    - Probable primes
2. Primes with Conditions:
    - Primes p1, p2, q1,q2, p and q shall all be provable primes
    - Primes p1, p2, q1, and q2 shall be provable primes and p and q shall be probable primes
    - Primes p1, p2, q1,q2, p and q shall all be probable primes

To test the key generation method for the Random Provable primes method and for all the Primes with Conditions methods, the evaluator must seed the TSF key generation routine with sufficient data to deterministically generate the RSA key pair. This includes the random seed(s), the public exponent of the RSA key, and the desired key length. For each key length supported, the evaluator shall have the TSF generate 25 key pairs. The evaluator shall verify the correctness of the TSF's implementation by comparing values generated by the TSF with those generated from a known good implementation.

If possible, the Random Probable primes method should also be verified against a known good implementation as described above. Otherwise, the evaluator shall have the TSF generate 10 keys pairs for each supported key length nlen and verify:

- n = p*q
- p and q are probably prime according to Miller-Rabin tests
- GCD(p-1,e) = 1
- GCD(q-1,e) = 1
- 2^16 < e < 2^256 and e is an odd integer
- |p-q| > 2^(nlen/2 – 100)
- p >= squareroot(2)*(2^(nlen/2 -1))
- q >= squareroot(2)*(2^(nlen/2 -1))
- 2^(nlen/2) < d < LCM(p-1,q-1)
- e*d = 1 mod LCM(p-1,q-1)

Key Generation for FIPS 186-4 Elliptic Curve Cryptography (ECC)

FIPS 186-4 ECC Key Generation Test

For each supported NIST curve, i.e. P-256, P-384 and P-521, the evaluator shall require the implementation under test (IUT) to generate 10 private/public key pairs. The private key shall be generated using an approved random bit generator (RBG). To determine correctness, the evaluator shall submit the generated key pairs to the public key verification (PKV) function of a known good implementation.

FIPS 186-4 Public Key Verification (PKV) Test

For each supported NIST curve, i.e. P-256, P-384 and P-521, the evaluator shall generate 10 private/public key pairs using the key generation function of a known good implementation and modify five of the public key values so that they are incorrect, leaving five values unchanged (i.e. correct). The evaluator shall obtain in response a set of 10 PASS/FAIL values.

Key Generation for Curve25519

The evaluator shall require the implementation under test (IUT) to generate 10 private/public key pairs. The private key shall be generated as specified in RFC 7748 using an approved random bit generator (RBG) and shall be written in little-endian order (least significant byte first). To determine correctness, the evaluator shall submit the generated key pairs to the public key verification (PKV) function of a known good

implementation.

Note: Assuming the PKV function of the good implementation will (using little-endian order):

a. confirm the private and public keys are 32-byte values
b. confirm the three least significant bits of the first byte of the private key are zero
c. confirm the most significant bit of the last byte is zero
d. confirm the second most significant bit of the last byte is one
e. calculate the expected public key from the private key and confirm it matches the supplied public key

The evaluator shall generate 10 private/public key pairs using the key generation function of a known good implementation and modify 5 of the public key values so that they are incorrect, leaving five values unchanged (i.e. correct). The evaluator shall obtain in response a set of 10 PASS/FAIL values.

Key Generation for Finite-Field Cryptography (FFC)
The evaluator shall verify the implementation of the Parameters Generation and the Key Generation for FFC by the TOE using the Parameter Generation and Key Generation test. This test verifies the ability of the TSF to correctly produce values for the field prime p, the cryptographic prime q (dividing p-1), the cryptographic group generator g, and the calculation of the private key x and public key y.
The Parameter generation specifies 2 ways (or methods) to generate the cryptographic prime q and the field prime p:

Cryptographic and Field Primes:

- Primes q and p shall both be provable primes
- Primes q and field prime p shall both be probable primes

and two ways to generate the cryptographic group generator g:

Cryptographic Group Generator:

- Generator g constructed through a verifiable process
- Generator g constructed through an unverifiable process

The Key generation specifies 2 ways to generate the private key x:

Private Key:

- len(q) bit output of RBG where 1 <= x <= q-1
- len(q) + 64 bit output of RBG, followed by a mod q-1 operation where 1<= x<=q-1

The security strength of the RBG must be at least that of the security offered by the FFC parameter set.

To test the cryptographic and field prime generation method for the provable primes method and/or the group generator g for a verifiable process, the evaluator must seed the TSF parameter generation routine with sufficient data to deterministically generate the parameter set.

For each key length supported, the evaluator shall have the TSF generate 25 parameter sets and key pairs. The evaluator shall verify the correctness of the TSF's implementation by comparing values generated by the TSF with those generated from a known good implementation. Verification must also confirm

- g != 0,1
- q divides p-1
- g^q mod p = 1
- g^x mod p = y

for each FFC parameter set and key pair. ***Diffie-Hellman Group 14 and FFC Schemes using "safe-prime" groups***
Testing for FFC Schemes using Diffie-Hellman group 14 and/or "safe-prime" groups is done as part of testing in FCS_CKM.2/UNLOCKED.

FCS_CKM.2/UNLOCKED Cryptographic Key Establishment

FCS_CKM.2/UNLOCKED
TSS
The evaluator shall ensure that the supported key establishment schemes correspond to the key generation schemes identified in FCS_CKM.1.1. If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme.

If Diffie-Hellman group 14 is selected from FCS_CKM.2/UNLOCKED, the TSS shall describe how the implementation meets RFC 3526 Section 3.

Guidance
The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key establishment scheme(s).

Tests
Evaluation Activity Note: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products.

The evaluator shall verify the implementation of the key establishment schemes supported by the TOE using the applicable tests below.

SP800-56A Revision 3 Key Establishment Schemes

The evaluator shall verify a TOE's implementation of SP800-56A Revision 3 key establishment schemes using the following Function and Validity tests. These validation tests for each key agreement scheme verify that a TOE has implemented the components of the key agreement scheme according to the specifications in the Recommendation. These components include the calculation of the DLC primitives (the shared secret value Z) and the calculation of the derived keying material (DKM) via the Key Derivation Function (KDF). If key confirmation is supported, the evaluator shall also verify that the components of key confirmation have been implemented correctly, using the test procedures described below. This includes the parsing of the DKM, the generation of MACdata and the calculation of MacTag.

Function Test

The Function test verifies the ability of the TOE to implement the key agreement schemes correctly. To conduct this test the evaluator shall generate or obtain test vectors from a known good implementation of the TOE supported schemes. For each supported key agreement scheme-key agreement role combination, KDF type, and, if supported, key confirmation role- key confirmation type combination, the tester shall generate 10 sets of test vectors. The data set consists of one set of domain parameter values (FFC) or the NIST approved curve (ECC) per 10 sets of public keys. These keys are static, ephemeral or both depending on the scheme being tested.

The evaluator shall obtain the DKM, the corresponding TOE’s public keys (static and/or ephemeral), the MAC tag(s), and any inputs used in the KDF, such as the Other Information field OI and TOE id fields.

If the TOE does not use a KDF defined in SP 800-56A Revision 3, the evaluator shall obtain only the public keys and the hashed value of the shared secret.

The evaluator shall verify the correctness of the TSF’s implementation of a given scheme by using a known good implementation to calculate the shared secret value, derive the keying material DKM, and compare hashes or MAC tags generated from these values.

If key confirmation is supported, the TSF shall perform the above for each implemented approved MAC algorithm.

Validity Test

The Validity test verifies the ability of the TOE to recognize another party’s valid and invalid key agreement results with or without key confirmation. To conduct this test, the evaluator shall obtain a list of the supporting cryptographic functions included in the SP800-56A Revision 3 key agreement implementation to determine which errors the TOE should be able to recognize. The evaluator generates a set of 24 (FFC) or 30 (ECC) test vectors consisting of data sets including domain parameter values or NIST approved curves, the evaluator’s public keys, the TOE’s public/private key pairs, MacTag, and any inputs used in the KDF, such as the other info and TOE id fields.

The evaluator shall inject an error in some of the test vectors to test that the TOE recognizes invalid key agreement results caused by the following fields being incorrect: the shared secret value Z, the DKM, the other information field OI, the data to be MACed, or the generated MacTag. If the TOE contains the full or partial (only ECC) public key validation, the evaluator will also individually inject errors in both parties’ static public keys, both parties’ ephemeral public keys and the TOE’s static private key to assure the TOE detects errors in the public key validation function and/or the partial key validation function (in ECC only). At least two of the test vectors shall remain unmodified and therefore should result in valid key agreement results (they should pass).

The TOE shall use these modified test vectors to emulate the key agreement scheme using the corresponding parameters. The evaluator shall compare the TOE’s results with the results using a known good implementation verifying that the TOE detects these errors.

SP800-56B Key Establishment Schemes

The evaluator shall verify that the TSS describes whether the TOE acts as a sender, a recipient, or both for

RSA-based key establishment schemes.

If the TOE acts as a sender, the following evaluation activity shall be performed to ensure the proper operation of every TOE supported combination of RSA-based key establishment scheme:
To conduct this test the evaluator shall generate or obtain test vectors from a known good implementation of the TOE supported schemes. For each combination of supported key establishment scheme and its options (with or without key confirmation if supported, for each supported key confirmation MAC function if key confirmation is supported, and for each supported mask generation function if KTS-OAEP is supported), the tester shall generate 10 sets of test vectors. Each test vector shall include the RSA public key, the plaintext keying material, any additional input parameters if applicable, the MacKey and MacTag if key confirmation is incorporated, and the outputted ciphertext. For each test vector, the evaluator shall perform a key establishment encryption operation on the TOE with the same inputs (in cases where key confirmation is incorporated, the test shall use the MacKey from the test vector instead of the randomly generated MacKey used in normal operation) and ensure that the outputted ciphertext is equivalent to the ciphertext in the test vector.

If the TOE acts as a receiver, the following evaluation activities shall be performed to ensure the proper operation of every TOE supported combination of RSA-based key establishment scheme:
To conduct this test the evaluator shall generate or obtain test vectors FCS_CKM.2.1/LOCKED from a known good implementation of the TOE supported schemes. For each combination of supported key establishment scheme and its options (with our without key confirmation if supported, for each supported key confirmation MAC function if key confirmation is supported, and for each supported mask generation function if KTS-OAEP is supported), the tester shall generate 10 sets of test vectors. Each test vector shall include the RSA private key, the plaintext keying material (KeyData), any additional input parameters if applicable, the MacTag in cases where key confirmation is incorporated, and the outputted ciphertext. For each test vector, the evaluator shall perform the key establishment decryption operation on the TOE and ensure that the outputted plaintext keying material (KeyData) is equivalent to the plaintext keying material in the test vector. In cases where key confirmation is incorporated, the evaluator shall perform the key confirmation steps and ensure that the outputted MacTag is equivalent to the MacTag in the test vector.

The evaluator shall ensure that the TSS describes how the TOE handles decryption errors. In accordance with NIST Special Publication 800-56B, the TOE must not reveal the particular error that occurred, either through the contents of any outputted or logged error message or through timing variations. If KTS-OAEP is supported, the evaluator shall create separate contrived ciphertext values that trigger each of the three decryption error checks described in NIST Special Publication 800-56B section 7.2.2.3, ensure that each decryption attempt results in an error, and ensure that any outputted or logged error message is identical for each. If KTS-KEM-KWS is supported, the evaluator shall create separate contrived ciphertext values that trigger each of the three decryption error checks described in NIST Special Publication 800-56B section 7.2.3.3, ensure that each decryption attempt results in an error, and ensure that any outputted or logged error message is identical for each.

RSAES-PKCS1-v1_5 Key Establishment Schemes

The evaluator shall verify the correctness of the TSF's implementation of RSAES-PKCS1-v1_5 by using a known good implementation for each protocol selected in FTP_ITC_EXT.1 that uses RSAES-PKCS1-v1_5.

Diffie-Hellman Group 14

The evaluator shall verify the correctness of the TSF's implementation of Diffie-Hellman group 14 by using a known good implementation for each protocol selected in FTP_ITC_EXT.1 that uses Diffie-Hellman Group 14.

FFC Schemes using "safe-prime" groups

The evaluator shall verify the correctness of the TSF's implementation of "safe-prime" groups by using a known good implementation for each protocol selected in FTP_ITC_EXT.1 that uses "safe-prime" groups. This test must be performed for each "safe-prime" group that each protocol uses.

FCS_CKM.2/LOCKED Cryptographic Key Establishment

FCS_CKM.2/LOCKED
TSS
There are no TSS evaluation activities for this component.

Guidance
There are no guidance evaluation activities for this component.

Tests
The test for SP800-56A Revision 3 and SP800-56B key establishment schemes is performed in association with FCS_CKM.2/UNLOCKED.

Curve25519 Key Establishment Schemes

The evaluator shall verify a TOE's implementation of the key agreement scheme using the following Function

and Validity tests. These validation tests for each key agreement scheme verify that a TOE has implemented the components of the key agreement scheme according to the specification. These components include the calculation of the shared secret K and the hash of K.

Function Test

The Function test verifies the ability of the TOE to implement the key agreement schemes correctly. To conduct this test the evaluator shall generate or obtain test vectors from a known good implementation of the TOE supported schemes. For each supported key agreement role and hash function combination, the tester shall generate 10 sets of public keys. These keys are static, ephemeral or both depending on the scheme being tested.

The evaluator shall obtain the shared secret value K, and the hash of K.

The evaluator shall verify the correctness of the TSF's implementation of a given scheme by using a known good implementation to calculate the shared secret value K and compare the hash generated from this value.

Validity Test

The Validity test verifies the ability of the TOE to recognize another party's valid and invalid key agreement results. To conduct this test, the evaluator generates a set of 30 test vectors consisting of data sets including the evaluator's public keys and the TOE's public/private key pairs.

The evaluator shall inject an error in some of the test vectors to test that the TOE recognizes invalid key agreement results caused by the following fields being incorrect: the shared secret value K or the hash of K. At least two of the test vectors shall remain unmodified and therefore should result in valid key agreement results (they should pass).

The TOE shall use these modified test vectors to emulate the key agreement scheme using the corresponding parameters. The evaluator shall compare the TOE's results with the results using a known good implementation verifying that the TOE detects these errors.

FCS_CKM_EXT.1 Cryptographic Key Support

FCS_CKM_EXT.1
TSS
The evaluator shall review the TSS to determine that a REK is supported by the TOE, that the TSS includes a description of the protection provided by the TOE for a REK, and that the TSS includes a description of the method of generation of a REK.

The evaluator shall verify that the description of the protection of a REK describes how any reading, import, and export of that REK is prevented. (For example, if the hardware protecting the REK is removable, the description should include how other devices are prevented from reading the REK.) The evaluator shall verify that the TSS describes how encryption/decryption/derivation actions are isolated so as to prevent applications and system-level processes from reading the REK while allowing encryption/decryption/derivation by the key.

The evaluator shall verify that the description includes how the OS is prevented from accessing the memory containing REK key material, which software is allowed access to the REK, how any other software in the execution environment is prevented from reading that key material, and what other mechanisms prevent the REK key material from being written to shared memory locations between the OS and the separate execution environment.

If key derivation is performed using a REK, the evaluator shall ensure that the TSS description includes a description of the key derivation function and shall verify the key derivation uses an approved derivation mode and key expansion algorithm according to FCS_CKM_EXT.3.2.

The evaluator shall verify that the generation of a REK meets the FCS_RBG_EXT.1.1 and FCS_RBG_EXT.1.2 requirements:

- If REK(s) is/are generated on-device, the TSS shall include a description of the generation mechanism including what triggers a generation, how the functionality described by FCS_RBG_EXT.1 is invoked, and whether a separate instance of the RBG is used for REK(s).
- If REK(s) is/are generated off-device, the TSS shall include evidence that the RBG meets FCS_RBG_EXT.1. This will likely necessitate a second set of RBG documentation equivalent to the documentation provided for the RBG Evaluation Activities. In addition, the TSS shall describe the manufacturing process that prevents the device manufacturer from accessing any REK(s).

Guidance
There are no guidance evaluation activities for this component.

Tests
There are no test evaluation activities for this component.

FCS_CKM_EXT.2 Cryptographic Key Random Generation

FCS_CKM_EXT.2
TSS
The evaluator shall ensure that the documentation of the product's encryption key management is detailed enough that, after reading, the product's key management hierarchy is clear and that it meets the requirements to ensure the keys are adequately protected. The evaluator shall ensure that the documentation includes both an essay and one or more diagrams. Note that this may also be documented as separate proprietary evidence rather than being included in the TSS.

The evaluator shall also examine the key hierarchy section of the TSS to ensure that the formation of all DEKs is described and that the key sizes match that described by the ST author. The evaluator shall examine the key hierarchy section of the TSS to ensure that each DEK is generated or combined from keys of equal or greater security strength using one of the selected methods.

- If the symmetric DEK is generated by an RBG, the evaluator shall review the TSS to determine that it describes how the functionality described by FCS_RBG_EXT.1 is invoked. The evaluator uses the description of the RBG functionality in FCS_RBG_EXT.1 or documentation available for the operational environment to determine that the key size being requested is greater than or equal to the key size and mode to be used for the encryption/decryption of the data.
- If the DEK is formed from a combination, the evaluator shall verify that the TSS describes the method of combination and that this method is either an XOR or a KDF to justify that the effective entropy of each factor is preserved. The evaluator shall also verify that each combined value was originally generated from an Approved DRBG described in FCS_RBG_EXT.1.
- If “concatenating the keys and using a KDF (as described in (SP 800-56C)” is selected, the evaluator shall ensure the TSS includes a description of the randomness extraction step.

The description must include how an approved untruncated MAC function is being used for the randomness extraction step and the evaluator must verify the TSS describes that the output length (in bits) of the MAC function is at least as large as the targeted security strength (in bits) of the parameter set employed by the key establishment scheme (see Tables 1-3 of SP 800-56C).

The description must include how the MAC function being used for the randomness extraction step is related to the PRF used in the key expansion and verify the TSS description includes the correct MAC function:

- If an HMAC-hash is used in the randomness extraction step, then the same HMAC-hash (with the same hash function hash) is used as the PRF in the key expansion step.
- If an AES-CMAC (with key length 128, 192, or 256 bits) is used in the randomness extraction step, then AES-CMAC with a 128-bit key is used as the PRF in the key expansion step.
- The description must include the lengths of the salt values being used in the randomness extraction step and the evaluator shall verify the TSS description includes correct salt lengths:
- If an HMAC-hash is being used as the MAC, the salt length can be any value up to the maximum bit length permitted for input to the hash function hash.
- If an AES-CMAC is being used as the MAC, the salt length shall be the same length as the AES key (i.e. 128, 192, or 256 bits).

(conditional) If a KDF is used, the evaluator shall ensure that the TSS includes a description of the key derivation function and shall verify the key derivation uses an approved derivation mode and key expansion algorithm according to SP 800-108 or SP 800-56C.

Guidance
The evaluator uses the description of the RBG functionality in FCS_RBG_EXT.1 or documentation available for the operational environment to determine that the key size being generated or combined is identical to the key size and mode to be used for the encryption/decryption of the data.

Tests
If a KDF is used, the evaluator shall perform one or more of the following tests to verify the correctness of the key derivation function, depending on the mode(s) that are supported. Table 20 maps the data fields to the notations used in SP 800-108 and SP 800-56C.

Table 20: : Notations used in SP 800-108 and SP 800-56C

Data Fields	**Notations**	
	SP 800-108	**SP 800-56C**
Pseudorandom function	PRF	PRF
Counter length	r	r
Length of output of PRF	h	h
Length of derived keying material	L	L

Length of input values	l length	l length
Pseudorandom input values I	K1 (key derivation key)	Z (shared secret)
Pseudorandom salt values	n/a	s
Randomness extraction MAC	n/a	MAC

Counter Mode Tests:

The evaluator shall determine the following characteristics of the key derivation function:

- One or more pseudorandom functions that are supported by the implementation (PRF).
- One or more of the values {8, 16, 24, 32} that equal the length of the binary representation of the counter (r).
- The length (in bits) of the output of the PRF (h).
- Minimum and maximum values for the length (in bits) of the derived keying material (L). These values can be equal if only one value of L is supported. These must be evenly divisible by h.
- Up to two values of L that are NOT evenly divisible by h.
- Location of the counter relative to fixed input data: before, after, or in the middle.
    - Counter before fixed input data: fixed input data string length (in bytes), fixed input data string value.
    - Counter after fixed input data: fixed input data string length (in bytes), fixed input data string value.
    - Counter in the middle of fixed input data: length of data before counter (in bytes), length of data after counter (in bytes), value of string input before counter, value of string input after counter.
- The length (I_length) of the input values I.

For each supported combination of I_length, MAC, salt, PRF, counter location, value of r, and value of L, the evaluator shall generate 10 test vectors that include pseudorandom input values I, and pseudorandom salt values. If there is only one value of L that is evenly divisible by h, the evaluator shall generate 20 test vectors for it. For each test vector, the evaluator shall supply this data to the TOE in order to produce the keying material output.

The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

Feedback Mode Tests:

The evaluator shall determine the following characteristics of the key derivation function:

- One or more pseudorandom functions that are supported by the implementation (PRF).
- The length (in bits) of the output of the PRF (h).
- Minimum and maximum values for the length (in bits) of the derived keying material (L). These values can be equal if only one value of L is supported. These must be evenly divisible by h.
- Up to two values of L that are NOT evenly divisible by h.
- Whether or not zero-length IVs are supported.
- Whether or not a counter is used, and if so:
    - One or more of the values {8, 16, 24, 32} that equal the length of the binary representation of the counter (r).
    - Location of the counter relative to fixed input data: before, after, or in the middle.
        - Counter before fixed input data: fixed input data string length (in bytes), fixed input data string value.
        - Counter after fixed input data: fixed input data string length (in bytes), fixed input data string value.
        - Counter in the middle of fixed input data: length of data before counter (in bytes), length of data after counter (in bytes), value of string input before counter, value of string input after counter.
- The length (I_length) of the input values I.

For each supported combination of I_length, MAC, salt, PRF, counter location (if a counter is used), value of r (if a counter is used), and value of L, the evaluator shall generate 10 test vectors that include pseudorandom input values I and pseudorandom salt values. If the KDF supports zero-length IVs, five of these test vectors will be accompanied by pseudorandom IVs and the other five will use zero-length IVs. If zero-length IVs are not supported, each test vector will be accompanied by an pseudorandom IV. If there is only one value of L that is evenly divisible by h, the evaluator shall generate 20 test vectors for it.

For each test vector, the evaluator shall supply this data to the TOE in order to produce the keying material output. The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

Double Pipeline Iteration Mode Tests:

The evaluator shall determine the following characteristics of the key derivation function:

- One or more pseudorandom functions that are supported by the implementation (PRF).
- The length (in bits) of the output of the PRF (h).
- Minimum and maximum values for the length (in bits) of the derived keying material (L). These values can be equal if only one value of L is supported. These must be evenly divisible by h.
- Up to two values of L that are NOT evenly divisible by h.
- Whether or not a counter is used, and if so:
    - One or more of the values {8, 16, 24, 32} that equal the length of the binary representation of the counter (r).
    - Location of the counter relative to fixed input data: before, after, or in the middle.
        - Counter before fixed input data: fixed input data string length (in bytes), fixed input data string value.
        - Counter after fixed input data: fixed input data string length (in bytes), fixed input data string value.
        - Counter in the middle of fixed input data: length of data before counter (in bytes), length of data after counter (in bytes), value of string input before counter, value of string input after counter.
- The length (I_length) of the input values I.

For each supported combination of I_length, MAC, salt, PRF, counter location (if a counter is used), value of r (if a counter is used), and value of L, the evaluator shall generate 10 test vectors that include pseudorandom input values I, and pseudorandom salt values. If there is only one value of L that is evenly divisible by h, the evaluator shall generate 20 test vectors for it.

For each test vector, the evaluator shall supply this data to the TOE in order to produce the keying material output. The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

FCS_CKM_EXT.3 Cryptographic Key Generation

FCS_CKM_EXT.3
TSS
The evaluator shall examine the key hierarchy section of the TSS to ensure that the formation of all KEKs are described and that the key sizes match that described by the ST author. The evaluator shall examine the key hierarchy section of the TSS to ensure that each key (DEKs, software-based key storage, and KEKs) is encrypted by keys of equal or greater security strength using one of the selected methods.

The evaluator shall review the TSS to verify that it contains a description of the conditioning used to derive KEKs. This description must include the size and storage location of salts. This activity may be performed in combination with that for FCS_COP.1/CONDITION.

(conditional) If the symmetric KEK is generated by an RBG, the evaluator shall review the TSS to determine that it describes how the functionality described by FCS_RBG_EXT.1 is invoked. The evaluator uses the description of the RBG functionality in FCS_RBG_EXT.1 or documentation available for the operational environment to determine that the key size being requested is greater than or equal to the key size and mode to be used for the encryption/decryption of the data.

(conditional) If the KEK is generated according to an asymmetric key scheme, the evaluator shall review the TSS to determine that it describes how the functionality described by FCS_CKM.1 is invoked. The evaluator uses the description of the key generation functionality in FCS_CKM.1 or documentation available for the operational environment to determine that the key strength being requested is greater than or equal to 112 bits.

(conditional) If the KEK is formed from a combination, the evaluator shall verify that the TSS describes the method of combination and that this method is either an XOR, a KDF, or encryption.

(conditional) If a KDF is used, the evaluator shall ensure that the TSS includes a description of the key derivation function and shall verify the key derivation uses an approved derivation mode and key expansion algorithm according to SP 800-108.

(conditional) If "concatenating the keys and using a KDF (as described in (SP 800-56C)" is selected, the evaluator shall ensure the TSS includes a description of the randomness extraction step. The description must include

- How an approved untruncated MAC function is being used for the randomness extraction step and the evaluator must verify the TSS describes that the output length (in bits) of the MAC function is at least as large as the targeted security strength (in bits) of the parameter set employed by the key establishment scheme (see Tables 1-3 of SP 800-56C).
- How the MAC function being used for the randomness extraction step is related to the PRF used in the key expansion and verify the TSS description includes the correct MAC function:

    - If an HMAC-hash is used in the randomness extraction step, then the same HMAC-hash (with the same hash function hash) is used as the PRF in the key expansion step.
    - If an AES-CMAC (with key length 128, 192, or 256 bits) is used in the randomness extraction step, then AES-CMAC with a 128-bit key is used as the PRF in the key expansion step.
- The lengths of the salt values being used in the randomness extraction step and the evaluator shall verify the TSS description includes correct salt lengths:
    - If an HMAC-hash is being used as the MAC, the salt length can be any value up to the maximum bit length permitted for input to the hash function hash.
    - If an AES-CMAC is being used as the MAC, the salt length shall be the same length as the AES key (i.e. 128, 192, or 256 bits).

The evaluator shall also ensure that the documentation of the product's encryption key management is detailed enough that, after reading, the product's key management hierarchy is clear and that it meets the requirements to ensure the keys are adequately protected. The evaluator shall ensure that the documentation includes both an essay and one or more diagrams. Note that this may also be documented as separate proprietary evidence rather than being included in the TSS.

Guidance
There are no guidance evaluation activities for this component.

Tests
If a KDF is used, the evaluator shall perform one or more of the following tests to verify the correctness of the key derivation function, depending on the mode(s) that are supported. Table 21 maps the data fields to the notations used in SP 800-108 and SP 800-56C.

Table 21: : Notations used in SP 800-108 and SP 800-56C

Data Fields	**Notations**	
	SP 800-108	**SP 800-56C**
Pseudorandom function	PRF	PRF
Counter length	r	r
Length of output of PRF	h	h
Length of derived keying material	L	L
Length of input values	I_length	I_length
Pseudorandom input values I	K_1 (key derivation key)	Z (shared secret)
Pseudorandom salt values	n/a	s
Randomness extraction MAC	n/a	MAC

Counter Mode Tests:

The evaluator shall determine the following characteristics of the key derivation function:

- One or more pseudorandom functions that are supported by the implementation (PRF).
- One or more of the values {8, 16, 24, 32} that equal the length of the binary representation of the counter (r).
- The length (in bits) of the output of the PRF (h).
- Minimum and maximum values for the length (in bits) of the derived keying material (L). These values can be equal if only one value of L is supported. These must be evenly divisible by h.
- Up to two values of L that are NOT evenly divisible by h.
- Location of the counter relative to fixed input data: before, after, or in the middle.
    - Counter before fixed input data: fixed input data string length (in bytes), fixed input data string value.
    - Counter after fixed input data: fixed input data string length (in bytes), fixed input data string value.
    - Counter in the middle of fixed input data: length of data before counter (in bytes), length of data after counter (in bytes), value of string input before counter, value of string input after counter.
- The length (I_length) of the input values I.

For each supported combination of I_length, MAC, salt, PRF, counter location, value of r, and value of L, the evaluator shall generate 10 test vectors that include pseudorandom input values I, and pseudorandom salt values. If there is only one value of L that is evenly divisible by h, the evaluator shall generate 20 test vectors for it. For each test vector, the evaluator shall supply this data to the TOE in order to produce the keying material output.

The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

Feedback Mode Tests:
The evaluator shall determine the following characteristics of the key derivation function:

- One or more pseudorandom functions that are supported by the implementation (PRF).
- The length (in bits) of the output of the PRF (h).
- Minimum and maximum values for the length (in bits) of the derived keying material (L). These values can be equal if only one value of L is supported. These must be evenly divisible by h.
- Up to two values of L that are NOT evenly divisible by h.
- Whether or not zero-length IVs are supported.
- Whether or not a counter is used, and if so:
    - One or more of the values {8, 16, 24, 32} that equal the length of the binary representation of the counter (r).
    - Location of the counter relative to fixed input data: before, after, or in the middle.
        - Counter before fixed input data: fixed input data string length (in bytes), fixed input data string value.
        - Counter after fixed input data: fixed input data string length (in bytes), fixed input data string value.
        - Counter in the middle of fixed input data: length of data before counter (in bytes), length of data after counter (in bytes), value of string input before counter, value of string input after counter.
- The length (I_length) of the input values I.

For each supported combination of I_length, MAC, salt, PRF, counter location (if a counter is used), value of r (if a counter is used), and value of L, the evaluator shall generate 10 test vectors that include pseudorandom input values I and pseudorandom salt values. If the KDF supports zero-length IVs, five of these test vectors will be accompanied by pseudorandom IVs and the other five will use zero-length IVs. If zero-length IVs are not supported, each test vector will be accompanied by an pseudorandom IV. If there is only one value of L that is evenly divisible by h, the evaluator shall generate 20 test vectors for it.

For each test vector, the evaluator shall supply this data to the TOE in order to produce the keying material output. The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

Double Pipeline Iteration Mode Tests:
The evaluator shall determine the following characteristics of the key derivation function:

- One or more pseudorandom functions that are supported by the implementation (PRF).
- The length (in bits) of the output of the PRF (h).
- Minimum and maximum values for the length (in bits) of the derived keying material (L). These values can be equal if only one value of L is supported. These must be evenly divisible by h.
- Up to two values of L that are NOT evenly divisible by h.
- Whether or not a counter is used, and if so:
    - One or more of the values {8, 16, 24, 32} that equal the length of the binary representation of the counter (r).
    - Location of the counter relative to fixed input data: before, after, or in the middle.
        - Counter before fixed input data: fixed input data string length (in bytes), fixed input data string value.
        - Counter after fixed input data: fixed input data string length (in bytes), fixed input data string value.
        - Counter in the middle of fixed input data: length of data before counter (in bytes), length of data after counter (in bytes), value of string input before counter, value of string input after counter.
- The length (I_length) of the input values I.

For each supported combination of I_length, MAC, salt, PRF, counter location (if a counter is used), value of r (if a counter is used), and value of L, the evaluator shall generate 10 test vectors that include pseudorandom input values I, and pseudorandom salt values. If there is only one value of L that is evenly divisible by h, the evaluator shall generate 20 test vectors for it.

For each test vector, the evaluator shall supply this data to the TOE in order to produce the keying material output. The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

FCS_CKM_EXT.4 Key Destruction

FCS_CKM_EXT.4

TSS

The evaluator shall check to ensure the TSS lists each type of plaintext key material (DEKs, software-based key storage, KEKs, trusted channel keys, passwords, etc.) and its generation and storage location.

The evaluator shall verify that the TSS describes when each type of key material is cleared (for example, on system power off, on wipe function, on disconnection of trusted channels, when no longer needed by the trusted channel per the protocol, when transitioning to the locked state, and possibly including immediately after use, while in the locked state, etc.).

The evaluator shall also verify that, for each type of key, the type of clearing procedure that is performed (cryptographic erase, overwrite with zeros, overwrite with random pattern, or block erase) is listed. If different types of memory are used to store the materials to be protected, the evaluator shall check to ensure that the TSS describes the clearing procedure in terms of the memory in which the data are stored.

Guidance

There are no guidance evaluation activities for this component.

Tests

Evaluation Activity Note: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products.

For each software and firmware key clearing situation (including on system power off, on wipe function, on disconnection of trusted channels, when no longer needed by the trusted channel per the protocol, when transitioning to the locked state, and possibly including immediately after use, while in the locked state) the evaluator shall repeat the following tests.

For these tests the evaluator shall utilize appropriate development environment (e.g. a Virtual Machine) and development tools (debuggers, simulators, etc.) to test that keys are cleared, including all copies of the key that may have been created internally by the TOE during normal cryptographic processing with that key.

- Test FCS_CKM_EXT.4:1: Applied to each key held as plaintext in volatile memory and subject to destruction by overwrite by the TOE (whether or not the plaintext value is subsequently encrypted for storage in volatile or non-volatile memory). In the case where the only selection made for the destruction method key was removal of power, then this test is unnecessary. The evaluator shall:
  1. Record the value of the key in the TOE subject to clearing.
  2. Cause the TOE to perform a normal cryptographic processing with the key from Step #1.
  3. Cause the TOE to clear the key.
  4. Cause the TOE to stop the execution but not exit.
  5. Cause the TOE to dump the entire memory of the TOE into a binary file.
  6. Search the content of the binary file created in Step #5 for instances of the known key value from Step #1.
  7. Break the key value from Step #1 into 3 similar sized pieces and perform a search using each piece.

  Steps 1-6 ensure that the complete key does not exist anywhere in volatile memory. If a copy is found, then the test fails.

  Step 7 ensures that partial key fragments do not remain in memory. If a fragment is found, there is a minuscule chance that it is not within the context of a key (e.g., some random bits that happen to match). If this is the case the test should be repeated with a different key in Step #1. If a fragment is found the test fails.

- Test FCS_CKM_EXT.4:2: Applied to each key held in non-volatile memory and subject to destruction by overwrite by the TOE. The evaluator shall use special tools (as needed), provided by the TOE developer if necessary, to view the key storage location:
  1. Record the value of the key in the TOE subject to clearing.
  2. Cause the TOE to perform a normal cryptographic processing with the key from Step #1.
  3. Cause the TOE to clear the key.
  4. Search the non-volatile memory the key was stored in for instances of the known key value from Step #1. If a copy is found, then the test fails.
  5. Break the key value from Step #1 into 3 similar sized pieces and perform a search using each piece. If a fragment is found then the test is repeated (as described for test 1 above), and if a fragment is found in the repeated test then the test fails.

- Test FCS_CKM_EXT.4:3: Applied to each key held as non-volatile memory and subject to destruction by overwrite by the TOE. The evaluator shall use special tools (as needed), provided by the TOE developer if necessary, to view the key storage location:
  1. Record the storage location of the key in the TOE subject to clearing.
  2. Cause the TOE to perform a normal cryptographic processing with the key from Step #1.
  3. Cause the TOE to clear the key.
  4. Read the storage location in Step #1 of non-volatile memory to ensure the appropriate pattern is utilized.

The test succeeds if correct pattern is used to overwrite the key in the memory location. If the pattern is not found the test fails.

FCS_CKM_EXT.5 TSF Wipe

FCS_CKM_EXT.5
TSS
The evaluator shall check to ensure the TSS describes how the device is wiped, the type of clearing procedure that is performed (cryptographic erase or overwrite) and, if overwrite is performed, the overwrite procedure (overwrite with zeros, overwrite three or more times by a different alternating pattern, overwrite with random pattern, or block erase).

If different types of memory are used to store the data to be protected, the evaluator shall check to ensure that the TSS describes the clearing procedure in terms of the memory in which the data are stored (for example, data stored on flash are cleared by overwriting once with zeros, while data stored on the internal persistent storage device are cleared by overwriting three times with a random pattern that is changed before each write).

Guidance
The evaluator shall verify that the AGD guidance describes how to enable encryption, if it is not enabled by default. Additionally the evaluator shall verify that the AGD guidance describes how to initiate the wipe command.

Tests
Evaluation Activity Note: The following test may require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on consumer Mobile Device products.

- Test FCS_CKM_EXT.5:1:The evaluator shall perform one of the following tests. The test before and after the wipe command shall be identical. This test shall be repeated for each type of memory used to store the data to be protected.
    - Test FCS_CKM_EXT.5:1.1:**For File-based Methods:**
      The evaluator shall enable encryption according to the AGD guidance. The evaluator shall create a user data (protected data or sensitive data) file, for example, by using an application. The evaluator shall use a tool provided by the developer to examine this data stored in memory (for example, by examining a decrypted files). The evaluator shall initiate the wipe command according to the AGD guidance provided for FMT_SMF_EXT.1. The evaluator shall use a tool provided by the developer to examine the same data location in memory to verify that the data has been wiped according to the method described in the TSS (for example, the files are still encrypted and cannot be accessed).
    - Test FCS_CKM_EXT.5:1.2:**For Volume-based Methods:**
      The evaluator shall enable encryption according to the AGD guidance. The evaluator shall create a unique data string, for example, by using an application. The evaluator shall use a tool provided by the developer to search decrypted data for the unique string. The evaluator shall initiate the wipe command according to the AGD guidance provided for FMT_SMF_EXT.1. The evaluator shall use a tool provided by the developer to search for the same unique string in decrypted memory to verify that the data has been wiped according to the method described in the TSS (for example, the files are still encrypted and cannot be accessed).
- Test FCS_CKM_EXT.5:2: The evaluator shall cause the device to wipe and verify that the wipe concludes with a power cycle.

FCS_CKM_EXT.6 Salt Generation

FCS_CKM_EXT.6
TSS
The evaluator shall verify that the TSS contains a description regarding the salt generation, including which algorithms on the TOE require salts. The evaluator shall confirm that the salt is generated using an RBG described in FCS_RBG_EXT.1. For PBKDF derivation of KEKs, this evaluation activity may be performed in conjunction with FCS_CKM_EXT.3.2.

Guidance
There are no guidance evaluation activities for this component.

Tests
There are no test evaluation activities for this component.

FCS_COP.1/ENCRYPT Cryptographic Operation

FCS_COP.1/ENCRYPT
TSS
There are no TSS evaluation activities for this component.

Guidance

There are no guidance evaluation activities for this component.

Tests
Evaluation Activity Note: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products.

- Test FCS_COP.1/ENCRYPT:1:AES-CBC Known Answer Tests

  There are four Known Answer Tests (KATs), described below. In all KATs, the plaintext, ciphertext, and IV values shall be 128-bit blocks. The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.
  - Test FCS_COP.1/ENCRYPT:1.1:KAT-1. To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 10 plaintext values and obtain the ciphertext value that results from AES-CBC encryption of the given plaintext using a key value of all zeros and an IV of all zeros. Five plaintext values shall be encrypted with a 128-bit all-zeros key, and the other five shall be encrypted with a 256-bit all-zeros key.

    To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using 10 ciphertext values as input and AES-CBC decryption.
  - Test FCS_COP.1/ENCRYPT:1.2:KAT-2. To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 10 key values and obtain the ciphertext value that results from AES-CBC encryption of an all-zeros plaintext using the given key value and an IV of all zeros. Five of the keys shall be 128-bit keys, and the other five shall be 256-bit keys.

    To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using an all-zero ciphertext value as input and AES-CBC decryption.
  - Test FCS_COP.1/ENCRYPT:1.3:KAT-3. To test the encrypt functionality of AES-CBC, the evaluator shall supply the two sets of key values described below and obtain the ciphertext value that results from AES encryption of an all-zeros plaintext using the given key value and an IV of all zeros. The first set of keys shall have 128 128-bit keys, and the second set shall have 256 256-bit keys. Key i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1,N].

    To test the decrypt functionality of AES-CBC, the evaluator shall supply the two sets of key and ciphertext value pairs described below and obtain the plaintext value that results from AES-CBC decryption of the given ciphertext using the given key and an IV of all zeros. The first set of key/ciphertext pairs shall have 128 128-bit key/ciphertext pairs, and the second set of key/ciphertext pairs shall have 256 256-bit key/ciphertext pairs. Key i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1,N]. The ciphertext value in each pair shall be the value that results in an all-zeros plaintext when decrypted with its corresponding key.
  - Test FCS_COP.1/ENCRYPT:1.4:KAT-4. To test the encrypt functionality of AES-CBC, the evaluator shall supply the set of 128 plaintext values described below and obtain the two ciphertext values that result from AES-CBC encryption of the given plaintext using a 128-bit key value of all zeros with an IV of all zeros and using a 256-bit key value of all zeros with an IV of all zeros, respectively. Plaintext value i in each set shall have the leftmost i bits be ones and the rightmost 128-i bits be zeros, for i in [1,128].

    To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using ciphertext values of the same form as the plaintext in the encrypt test as input and AES-CBC decryption.
- Test FCS_COP.1/ENCRYPT:2:AES-CBC Multi-Block Message Test

  The evaluator shall test the encrypt functionality by encrypting an i-block message where 1 < i <= 10. The evaluator shall choose a key, an IV and plaintext message of length i blocks and encrypt the message, using the mode to be tested, with the chosen key and IV. The ciphertext shall be compared to the result of encrypting the same plaintext message with the same key and IV using a known good implementation.

  The evaluator shall also test the decrypt functionality for each mode by decrypting an i-block message where 1 < i <= 10. The evaluator shall choose a key, an IV and a ciphertext message of length i blocks and decrypt the message, using the mode to be tested, with the chosen key and IV. The plaintext shall be compared to the result of decrypting the same ciphertext message with the same key and IV using a known good implementation.
- Test FCS_COP.1/ENCRYPT:3:AES-CBC Monte Carlo Tests

  The evaluator shall test the encrypt functionality using a set of 200 plaintext, IV, and key 3-tuples. 100 of these shall use 128 bit keys, and 100 shall use 256 bit keys. The plaintext and IV values shall be 128-bit blocks. For each 3-tuple, 1000 iterations shall be run as follows:

```
# Input: PT, IV, Key for i = 1 to 1000: if i == 1: CT[1] =
                    AES-CBC-Encrypt(Key, IV, PT) PT = IV else: CT[i] = AES-CBC-Encrypt(Key, PT) PT
                    = CT[i-1]
```

The ciphertext computed in the 1000^{th} iteration (i.e. CT[1000]) is the result for that trial. This result shall be compared to the result of running 1000 iterations with the same values using a known good implementation.

The evaluator shall test the decrypt functionality using the same test as for encrypt, exchanging CT and PT and replacing AES-CBC-Encrypt with AES-CBC-Decrypt.

- Test FCS_COP.1/ENCRYPT:4: The evaluator shall test the generation-encryption and decryption-verification functionality of AES-CCM for the following input parameter and tag lengths:

  128 bit and 256 bit keys

  Two payload lengths. One payload length shall be the shortest supported payload length, greater than or equal to zero bytes. The other payload length shall be the longest supported payload length, less than or equal to 32 bytes (256 bits).

  Two or three associated data lengths. One associated data length shall be 0, if supported. One associated data length shall be the shortest supported payload length, greater than or equal to zero bytes. One associated data length shall be the longest supported payload length, less than or equal to 32 bytes (256 bits). If the implementation supports an associated data length of 2^{16} bytes, an associated data length of 2^{16} bytes shall be tested.

  Nonce lengths. All supported nonce lengths between 7 and 13 bytes, inclusive, shall be tested.

  Tag lengths. All supported tag lengths of 4, 6, 8, 10, 12, 14 and 16 bytes shall be tested.

  To test the generation-encryption functionality of AES-CCM, the evaluator shall perform the following four tests:
  - Test FCS_COP.1/ENCRYPT:4.1:For EACH supported key and associated data length and ANY supported payload, nonce and tag length, the evaluator shall supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext.
  - Test FCS_COP.1/ENCRYPT:4.2:For EACH supported key and payload length and ANY supported associated data, nonce and tag length, the evaluator shall supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext.
  - Test FCS_COP.1/ENCRYPT:4.3:For EACH supported key and nonce length and ANY supported associated data, payload and tag length, the evaluator shall supply one key value and 10 associated data, payload and nonce value 3-tuples and obtain the resulting ciphertext.
  - Test FCS_COP.1/ENCRYPT:4.4:For EACH supported key and tag length and ANY supported associated data, payload and nonce length, the evaluator shall supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext.

  To determine correctness in each of the above tests, the evaluator shall compare the ciphertext with the result of generation-encryption of the same inputs with a known good implementation.

  To test the decryption-verification functionality of AES-CCM, for EACH combination of supported associated data length, payload length, nonce length and tag length, the evaluator shall supply a key value and 15 nonce, associated data and ciphertext 3-tuples and obtain either a FAIL result or a PASS result with the decrypted payload. The evaluator shall supply 10 tuples that should FAIL and 5 that should PASS per set of 15.

- Test FCS_COP.1/ENCRYPT:5:The evaluator shall test the encrypt functionality using a set of 10 key, plaintext, AAD, and IV tuples for each combination of parameter lengths above and obtain the ciphertext value and tag that results from AES-GCM authenticated encrypt. Each supported tag length shall be tested at least once per set of 10. The IV value may be supplied by the evaluator or the implementation being tested, as long as it is known.
- Test FCS_COP.1/ENCRYPT:6:The evaluator shall test the decrypt functionality using a set of 10 key, ciphertext, tag, AAD, and IV 5-tuples for each combination of parameter lengths above and obtain a Pass/Fail result on authentication and the decrypted plaintext if Pass. The set shall include five tuples that Pass and five that Fail.

- Test FCS_COP.1/ENCRYPT:7:The evaluator shall test the encrypt functionality of XTS-AES for each combination of the following input parameter lengths:

  256 bit (for AES-128) and 512 bit (for AES-256) keys

  Three data unit (i.e. plaintext) lengths. One of the data unit lengths shall be a non-zero integer multiple of 128 bits, if supported. One of the data unit lengths shall be an integer multiple of 128 bits, if supported. The third data unit length shall be either the longest supported data unit length or 216 bits, whichever is smaller.

using a set of 100 (key, plaintext and 128-bit random tweak value) 3-tuples and obtain the ciphertext that results from XTS-AES encrypt.

The evaluator may supply a data unit sequence number instead of the tweak value if the implementation supports it. The data unit sequence number is a base-10 number ranging between 0 and 255 that implementations convert to a tweak value internally.

- Test FCS_COP.1/ENCRYPT:8:The evaluator shall test the decrypt functionality of XTS-AES using the same test as for encrypt, replacing plaintext values with ciphertext values and XTS-AES encrypt with XTS-AES decrypt.

- Test FCS_COP.1/ENCRYPT:9:The evaluator shall test the authenticated encryption functionality of AES-KW for EACH combination of the following input parameter lengths:

  128 and 256 bit key encryption keys (KEKs)

  Three plaintext lengths. One of the plaintext lengths shall be two semi-blocks (128 bits). One of the plaintext lengths shall be three semi-blocks (192 bits). The third data unit length shall be the longest supported plaintext length less than or equal to 64 semi-blocks (4096 bits).

  using a set of 100 key and plaintext pairs and obtain the ciphertext that results from AES-KW authenticated encryption. To determine correctness, the evaluator shall use the AES-KW authenticated-encryption function of a known good implementation.
- Test FCS_COP.1/ENCRYPT:10:The evaluator shall test the authenticated-decryption functionality of AES-KW using the same test as for authenticated-encryption, replacing plaintext values with ciphertext values and AES-KW authenticated-encryption with AES-KW authenticated-decryption.
- Test FCS_COP.1/ENCRYPT:11:The evaluator shall test the authenticated-encryption functionality of AES-KWP using the same test as for AES-KW authenticated-encryption with the following change in the three plaintext lengths:
  - One plaintext length shall be one octet. One plaintext length shall be 20 octets (160 bits).
  - One plaintext length shall be the longest supported plaintext length less than or equal to 512 octets (4096 bits).
- Test FCS_COP.1/ENCRYPT:12:The evaluator shall test the authenticated-decryption functionality of AES-KWP using the same test as for AES-KWP authenticated-encryption, replacing plaintext values with ciphertext values and AES-KWP authenticated-encryption with AES-KWP authenticated-decryption.

FCS_COP.1/HASH Cryptographic Operation

FCS_COP.1/HASH
TSS
The evaluator shall check that the association of the hash function with other TSF cryptographic functions (for example, the digital signature verification function) is documented in the TSS. The evaluator shall check that the TSS indicates if the hashing function is implemented in bit-oriented and/or byte-oriented mode.

Guidance
The evaluator checks the AGD documents to determine that any configuration that is required to be done to configure the functionality for the required hash sizes is present.

Tests
Evaluation Activity Note: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products.

The evaluator shall perform all of the following tests for each hash algorithm implemented by the TSF and used to satisfy the requirements of this PP. As there are different tests for each mode, an indication is given in the following sections for the bitoriented vs. the byteoriented tests.

- Test FCS_COP.1/HASH:1:*Short Messages Test: Bit-oriented Mode*
  The evaluators devise an input set consisting of m+1 messages, where m is the block length of the hash algorithm. The length of the messages ranges sequentially from 0 to m bits. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.
- Test FCS_COP.1/HASH:2:*Short Messages Test: Byte-oriented Mode*
  The evaluators devise an input set consisting of m/8+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m/8 bytes, with each message being an integral number of bytes. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.
- Test FCS_COP.1/HASH:3:*Selected Long Messages Test: Bit-oriented Mode*
  The evaluators devise an input set consisting of m messages, where m is the block length of the hash algorithm. The length of the i^{th} message is 512 + 99*i, where $1 \leq i \leq m$. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

- Test FCS_COP.1/HASH:4:*Selected Long Messages Test: Byte-oriented Mode*
  The evaluators devise an input set consisting of m/8 messages, where m is the block length of the hash algorithm. The length of the i^{th} message is 512 + 8*99*i, where 1 ≤ i ≤ m/8. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.
- Test FCS_COP.1/HASH:5:*Pseudorandomly Generated Messages Test: Byte-oriented Mode*
  This test is for byteoriented implementations only. The evaluators randomly generate a seed that is n bits long, where n is the length of the message digest produced by the hash function to be tested. The evaluators then formulate a set of 100 messages and associated digests by following the algorithm provided in Figure 1 of SHAVS. The evaluators then ensure that the correct result is produced when the messages are provided to the TSF.

FCS_COP.1/SIGN Cryptographic Operation

FCS_COP.1/SIGN
TSS
There are no TSS evaluation activities for this component.

Guidance
There are no guidance evaluation activities for this component.

Tests
Evaluation Activity Note: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products.

- Test FCS_COP.1/SIGN:1:**[conditional] If "ECDSA schemes..." is selected in FCS_COP.1.1/SIGN**
  - Test FCS_COP.1/SIGN:1.1:**ECDSA FIPS 186-4 Signature Generation Test**
    For each supported NIST curve (i.e. P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate 10 1024-bit long messages and obtain for each message a public key and the resulting signature values R and S. To determine correctness, the evaluator shall use the signature verification function of a known good implementation.
  - Test FCS_COP.1/SIGN:1.2:**ECDSA FIPS 186-4 Signature Verification Test**
    For each supported NIST curve (i.e. P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate a set of 10 1024-bit message, public key and signature tuples and modify one of the values (message, public key or signature) in five of the 10 tuples. The evaluator shall obtain in response a set of 10 PASS/FAIL values.
- Test FCS_COP.1/SIGN:2:**[conditional] If "RSA schemes..." is selected in FCS_COP.1.1/SIGN**
  - Test FCS_COP.1/SIGN:2.1:**Signature Generation Test**
    The evaluator shall verify the implementation of RSA Signature Generation by the TOE using the Signature Generation Test. To conduct this test the evaluator must generate or obtain 10 messages from a trusted reference implementation for each modulus size/SHA combination supported by the TSF. The evaluator shall have the TOE use their private key and modulus value to sign these messages.

    The evaluator shall verify the correctness of the TSF's signature using a known good implementation and the associated public keys to verify the signatures.
  - Test FCS_COP.1/SIGN:2.2:**Signature Verification Test**
    The evaluator shall perform the Signature Verification test to verify the ability of the TOE to recognize another party's valid and invalid signatures. The evaluator shall inject errors into the test vectors produced during the Signature Verification Test by introducing errors in some of the public keys e, messages, IR format, and/or signatures. The TOE attempts to verify the signatures and returns success or failure.

    The evaluator shall use these test vectors to emulate the signature verification test using the corresponding parameters and verify that the TOE detects these errors.

FCS_COP.1/KEYHMAC Cryptographic Operation

FCS_COP.1/KEYHMAC
TSS
The evaluator shall examine the TSS to ensure that it specifies the following values used by the HMAC function: key length, hash function used, block size, and output MAC length used.

Guidance
There are no guidance evaluation activities for this component.

Tests
Evaluation Activity Note: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products.

For each of the supported parameter sets, the evaluator shall compose 15 sets of test data. Each set shall consist of a key and message data. The evaluator shall have the TSF generate HMAC tags for these sets of

test data. The resulting MAC tags shall be compared to the result of generating HMAC tags with the same key and IV using a known good implementation.

FCS_COP.1/CONDITION Cryptographic Operation

FCS_COP.1/CONDITION
TSS
The evaluator shall check that the TSS describes the method by which the password is first encoded and then fed to the SHA algorithm and verify the SHA algorithm matches the first selection.

If a key stretching function, such as PBKDF2, is selected the settings for the algorithm (padding, blocking, etc.) shall be described. The evaluator shall verify that the TSS contains a description of how the output of the hash function or key stretching function is used to form the submask that will be input into the function and is the same length as the KEK as specified in FCS_CKM_EXT.3.

If any manipulation of the key is performed in forming the submask that will be used to form the KEK, that process shall be described in the TSS.

Guidance
There are no guidance evaluation activities for this component.

Tests
There are no test evaluation activities for this component. No explicit testing of the formation of the submask from the input password is required.

FCS_HTTPS_EXT.1 HTTPS Protocol

FCS_HTTPS_EXT.1
TSS
There are no TSS evaluation activities for this component.

Guidance
There are no guidance evaluation activities for this component.

Tests

- Test FCS_HTTPS_EXT.1:1:The evaluator shall attempt to establish an HTTPS connection with a webserver, observe the traffic with a packet analyzer, and verify that the connection succeeds and that the traffic is identified as TLS or HTTPS.

  Other tests are performed in conjunction with testing in the Package for Transport Layer Security.

  Certificate validity shall be tested in accordance with testing performed for FIA_X509_EXT.1, and the evaluator shall perform the following test:
- Test FCS_HTTPS_EXT.1:2:The evaluator shall demonstrate that using a certificate without a valid certification path results in an application notification. Using the administrative guidance, the evaluator shall then load a certificate or certificates to the Trust Anchor Database needed to validate the certificate to be used in the function, and demonstrate that the function succeeds. The evaluator then shall delete one of the certificates, and show that the application is notified of the validation failure.

FCS_IV_EXT.1 Initialization Vector Generation

FCS_IV_EXT.1
TSS
The evaluator shall examine the key hierarchy section of the TSS to ensure that the encryption of all keys is described and the formation of the IVs for each key encrypted by the same KEK meets FCS_IV_EXT.1.

Guidance
There are no guidance evaluation activities for this component.

Tests
There are no test evaluation activities for this component.

FCS_RBG_EXT.1 Random Bit Generation

FCS_RBG_EXT.1
Documentation shall be produced and the evaluator shall perform the activities in accordance with D - , the "Clarification to the Entropy Documentation and Assessment".

The evaluator shall verify that the API documentation provided according to 5.2.2, includes the security functions described in FCS_RBG_EXT.1.3.

TSS

There are no TSS evaluation activities for this component.

Guidance
The evaluator shall also confirm that the operational guidance contains appropriate instructions for configuring the RNG functionality.

Tests
Evaluation Activity Note: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products.

The evaluator shall perform 15 trials for the RNG implementation. If the RNG is configurable, the evaluator shall perform 15 trials for each configuration.

If the RNG has prediction resistance enabled, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) generate a second block of random bits (4) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. "generate one block of random bits" means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP800-90A).

If the RNG does not have prediction resistance, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) reseed, (4) generate a second block of random bits (5) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The fifth value is additional input to the first call to generate. The sixth and seventh are additional input and entropy input to the call to reseed. The final value is additional input to the second generate call.

The following paragraphs contain more information on some of the input values to be generated/selected by the evaluator.

Entropy input: the length of the entropy input value must equal the seed length.
Nonce: If a nonce is supported (CTR_DRBG with no Derivation Function does not use a nonce), the nonce bit length is one-half the seed length.
Personalization string: The length of the personalization string must be seed length. If the implementation only supports one personalization string length, then the same length can be used for both values. If more than one string length is support, the evaluator shall use personalization strings of two different lengths. If the implementation does not use a personalization string, no value needs to be supplied.
Additional input: the additional input bit lengths have the same defaults and restrictions as the personalization string lengths.

FCS_SRV_EXT.1 Cryptographic Algorithm Services

FCS_SRV_EXT.1
The evaluator shall verify that the API documentation provided according to 5.2.2 includes the security functions (cryptographic algorithms) described in these requirements.

TSS
There are no TSS evaluation activities for this component.

Guidance
There are no guidance evaluation activities for this component.

Tests
The evaluator shall write, or the developer shall provide access to, an application that requests cryptographic operations by the TSF. The evaluator shall verify that the results from the operation match the expected results according to the API documentation. This application may be used to assist in verifying the cryptographic operation Evaluation Activities for the other algorithm services requirements.

2.1.3 Cryptographic Storage (FCS_STG)

FCS_STG_EXT.1 Cryptographic Key Storage

FCS_STG_EXT.1
The evaluator shall verify that the API documentation provided according to 5.2.2 includes the security functions (import, use, and destruction) described in these requirements. The API documentation shall include the method by which applications restrict access to their keys/secrets in order to meet FCS_STG_EXT.1.4.

TSS
The evaluator shall review the TSS to determine that the TOE implements the required secure key storage. The evaluator shall ensure that the TSS contains a description of the key storage mechanism that justifies the selection of "mutable hardware" or "software-based".

Guidance
The evaluator shall review the AGD guidance to determine that it describes the steps needed to import or destroy keys/secrets.

Tests
The evaluator shall test the functionality of each security function:

- Test FCS_STG_EXT.1:1:The evaluator shall import keys/secrets of each supported type according to the AGD guidance. The evaluator shall write, or the developer shall provide access to, an application that generates a key/secret of each supported type and calls the import functions. The evaluator shall verify that no errors occur during import.
- Test FCS_STG_EXT.1:2:The evaluator shall write, or the developer shall provide access to, an application that uses an imported key/secret:
  - For RSA, the secret shall be used to sign data.
  - For ECDSA, the secret shall be used to sign data

  In the future additional types will be required to be tested:
  - For symmetric algorithms, the secret shall be used to encrypt data.
  - For persistent secrets, the secret shall be compared to the imported secret.

  The evaluator shall repeat this test with the application-imported keys/secrets and a different application’s imported keys/secrets. The evaluator shall verify that the TOE requires approval before allowing the application to use the key/secret imported by the user or by a different application:
  - The evaluator shall deny the approvals to verify that the application is not able to use the key/secret as described.
  - The evaluator shall repeat the test, allowing the approvals to verify that the application is able to use the key/secret as described.

  If the ST author has selected "common application developer", this test is performed by either using applications from different developers or appropriately (according to API documentation) not authorizing sharing.
- Test FCS_STG_EXT.1:3:The evaluator shall destroy keys/secrets of each supported type according to the AGD guidance. The evaluator shall write, or the developer shall provide access to, an application that destroys an imported key/secret.

  The evaluator shall repeat this test with the application-imported keys/secrets and a different application’s imported keys/secrets. The evaluator shall verify that the TOE requires approval before allowing the application to destroy the key/secret imported by the administrator or by a different application:

  - The evaluator shall deny the approvals and verify that the application is still able to use the key/secret as described.
  - The evaluator shall repeat the test, allowing the approvals and verifying that the application is no longer able to use the key/secret as described.

  If the ST author has selected "common application developer", this test is performed by either using applications from different developers or appropriately (according to API documentation) not authorizing sharing.

FCS_STG_EXT.2 Encrypted Cryptographic Key Storage

FCS_STG_EXT.2
The evaluator shall review the TSS to determine that the TSS includes key hierarchy description of the protection of each DEK for data-at-rest, of software-based key storage, of long-term trusted channel keys, and of KEK related to the protection of the DEKs, long-term trusted channel keys, and software-based key storage. This description must include a diagram illustrating the key hierarchy implemented by the TOE in order to demonstrate that the implementation meets FCS_STG_EXT.2. The description shall indicate how the functionality described by FCS_RBG_EXT.1 is invoked to generate DEKs (FCS_CKM_EXT.2), the key size (FCS_CKM_EXT.2 and FCS_CKM_EXT.3) for each key, how each KEK is formed (generated, derived, or combined according to FCS_CKM_EXT.3), the integrity protection method for each encrypted key (FCS_STG_EXT.3), and the IV generation for each key encrypted by the same KEK (FCS_IV_EXT.1). More detail for each task follows the corresponding requirement.

The evaluator shall also ensure that the documentation of the product's encryption key management is detailed enough that, after reading, the product's key management hierarchy is clear and that it meets the requirements to ensure the keys are adequately protected. The evaluator shall ensure that the documentation includes both an essay and one or more diagrams. Note that this may also be documented as separate proprietary evidence rather than being included in the TSS.

There are no guidance evaluation activities for this element.

There are no test evaluation activities for this element.
TSS
The evaluator shall examine the key hierarchy description in the TSS section to verify that each DEK and software-stored key is encrypted according to FCS_STG_EXT.2.

Guidance
There are no guidance evaluation activities for this element.

Tests
There are no test evaluation activities for this element.

FCS_STG_EXT.3 Integrity of Encrypted Key Storage

FCS_STG_EXT.3
TSS
The evaluator shall examine the key hierarchy description in the TSS section to verify that each encrypted key is integrity protected according to one of the options in FCS_STG_EXT.3.

The evaluator shall also ensure that the documentation of the product's encryption key management is detailed enough that, after reading, the product's key management hierarchy is clear and that it meets the requirements to ensure the keys are adequately protected. The evaluator shall ensure that the documentation includes both an essay and one or more diagrams. Note that this may also be documented as separate proprietary evidence rather than being included in the TSS.

Guidance
There are no guidance evaluation activities for this component.

Tests
There are no test evaluation activities for this component.

2.1.4 Class: User Data Protection (FDP)

FDP_ACF_EXT.1 Access Control for System Services

FDP_ACF_EXT.1
The evaluator shall ensure the TSS lists all system services available for use by an application. The evaluator shall also ensure that the TSS describes how applications interface with these system services, and means by which these system services are protected by the TSF.

The TSS shall describe which of the following categories each system service falls in:

1. No applications are allowed access
2. Privileged applications are allowed access
3. Applications are allowed access by user authorization
4. All applications are allowed access

Privileged applications include any applications developed by the TSF developer. The TSS shall describe how privileges are granted to third-party applications. For both types of privileged applications, the TSS shall describe how and when the privileges are verified and how the TSF prevents unprivileged applications from accessing those services.

For any services for which the user may grant access, the evaluator shall ensure that the TSS identifies whether the user is prompted for authorization when the application is installed, or during runtime. The evaluator shall ensure that the operational user guidance contains instructions for restricting application access to system services.

There are no guidance evaluation activities for this element.

Evaluation Activity Note: The following tests require the vendor to provide access to a test platform that provides the evaluator with tools that are typically not found on consumer Mobile Device products.

The evaluator shall write, or the developer shall provide, applications for the purposes of the following tests.

- Test FDP_ACF_EXT.1:1:For each system service to which no applications are allowed access, the evaluator shall attempt to access the system service with a test application and verify that the application is not able to access that system service.
- Test FDP_ACF_EXT.1:2:For each system service to which only privileged applications are allowed access, the evaluator shall attempt to access the system service with an unprivileged application and verify that the application is not able to access that system service. The evaluator shall attempt to access the system service with a privileged application and verify that the application can access the service.
- Test FDP_ACF_EXT.1:3:For each system service to which the user may grant access, the evaluator shall

attempt to access the system service with a test application. The evaluator shall ensure that either the system blocks such accesses or prompts for user authorization. The prompt for user authorization may occur at runtime or at installation time, and should be consistent with the behavior described in the TSS.
- Test FDP_ACF_EXT.1:4:For each system service listed in the TSS that is accessible by all applications, the evaluator shall test that an application can access that system service.

TSS
The evaluator shall examine the TSS to verify that it describes which data sharing is permitted between applications, which data sharing is not permitted, and how disallowed sharing is prevented. It is possible to select both "applications" and "groups of applications", in which case the TSS is expected to describe the data sharing policies that would be applied in each case.

Guidance
There are no guidance evaluation activities for this element.

Tests

- Test FDP_ACF_EXT.1:1:The evaluator shall write, or the developer shall provide, two applications, one that saves data containing a unique string and the other, which attempts to access that data. If "groups of applications" is selected, the applications shall be placed into different groups. If "application" is selected, the evaluator shall install the two applications. If "private data" is selected, the application shall not write to a designated shared storage area. The evaluator shall verify that the second application is unable to access the stored unique string.

  If "the user" is selected, the evaluator shall grant access as the user and verify that the second application is able to access the stored unique string.

  If "the administrator" is selected, the evaluator shall grant access as the administrator and verify that the second application is able to access the stored unique string.

  If "a common application developer" is selected, the evaluator shall grant access to an, application with a common application developer to the first, and verify that the application is able to access the stored unique string.

FDP_DAR_EXT.1 Protected Data Encryption

FDP_DAR_EXT.1
TSS
The evaluator shall verify that the TSS section of the ST indicates which data is protected by the DAR implementation and what data is considered TSF data. The evaluator shall ensure that this data includes all protected data.

Guidance
The evaluator shall review the AGD guidance to determine that the description of the configuration and use of the DAR protection does not require the user to perform any actions beyond configuration and providing the authentication credential. The evaluator shall also review the AGD guidance to determine that the configuration does not require the user to identify encryption on a per-file basis.

Tests
Evaluation Activity Note: The following test requires the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on consumer Mobile Device products.

- Test FDP_DAR_EXT.1:1:The evaluator shall enable encryption according to the AGD guidance. The evaluator shall create user data (non-system) either by creating a file or by using an application. The evaluator shall use a tool provided by the developer to verify that this data is encrypted when the product is powered off, in conjunction with Test 1 for FIA_UAU_EXT.1.

FDP_DAR_EXT.2 Sensitive Data Encryption

FDP_DAR_EXT.2
The evaluator shall verify that the TSS includes a description of which data stored by the TSF (such as by native applications) is treated as sensitive. This data may include all or some user or enterprise data and must be specific regarding the level of protection of email, contacts, calendar appointments, messages, and documents.

The evaluator shall examine the TSS to determine that it describes the mechanism that is provided for applications to use to mark data and keys as sensitive. This description shall also contain information reflecting how data and keys marked in this manner are distinguished from data and keys that are not (for instance, tagging, segregation in a "special" area of memory or container, etc.).

There are no guidance evaluation activities for this element.

- Test FDP_DAR_EXT.2:1:The evaluator shall enable encryption of sensitive data and require user authentication according to the AGD guidance. The evaluator shall try to access and create sensitive data (as defined in the ST and either by creating a file or using an application to generate sensitive data) in order to verify that no other user interaction is required.

The evaluator shall review the TSS section of the ST to determine that the TSS includes a description of the process of receiving sensitive data while the device is in a locked state. The evaluator shall also verify that the description indicates if sensitive data that may be received in the locked state is treated differently than sensitive data that cannot be received in the locked state. The description shall include the key scheme for encrypting and storing the received data, which must involve an asymmetric key and must prevent the sensitive data-at-rest from being decrypted by wiping all key material used to derive or encrypt the data (as described in the application note). The introduction to this section provides two different schemes that meet the requirements, but other solutions may address this requirement.

There are no guidance evaluation activities for this element.

The evaluator shall perform the tests in FCS_CKM_EXT.4 for all key material no longer needed while in the locked state and shall ensure that keys for the asymmetric scheme are addressed in the tests performed when transitioning to the locked state.
The evaluator shall verify that the key hierarchy section of the TSS required for FCS_STG_EXT.2.1 includes the symmetric encryption keys (DEKs) used to encrypt sensitive data. The evaluator shall ensure that these DEKs are encrypted by a key encrypted with (or chain to a KEK encrypted with) the REK and password-derived or biometric-unlocked KEK.

The evaluator shall verify that the TSS section of the ST that describes the asymmetric key scheme includes the protection of any private keys of the asymmetric pairs. The evaluator shall ensure that any private keys that are not wiped and are stored by the TSF are stored encrypted by a key encrypted with (or chain to a KEK encrypted with) the REK and password-derived or biometric-unlocked KEK.

The evaluator shall also ensure that the documentation of the product's encryption key management is detailed enough that, after reading, the product's key management hierarchy is clear and that it meets the requirements to ensure the keys are adequately protected. The evaluator shall ensure that the documentation includes both an essay and one or more diagrams. Note that this may also be documented as separate proprietary evidence rather than being included in the TSS.

There are no guidance evaluation activities for this element.

There are no test evaluation activities for this element.
TSS
The evaluator shall verify that the TSS section of the ST that describes the asymmetric key scheme includes a description of the actions taken by the TSF for the purposes of DAR upon transitioning to the unlocked state. These actions shall minimally include decrypting all received data using the asymmetric key scheme and re-encrypting with the symmetric key scheme used to store data while the device is unlocked.

Guidance
There are no guidance evaluation activities for this element.

Tests
There are no test evaluation activities for this element.

FDP_IFC_EXT.1 Subset Information Flow Control

FDP_IFC_EXT.1
TSS
The evaluator shall verify that the TSS section of the ST describes the routing of IP traffic through processes on the TSF when a VPN client is enabled. The evaluator shall ensure that the description indicates which traffic does not go through the VPN and which traffic does and that a configuration exists for each baseband protocol in which only the traffic identified by the ST author as necessary for establishing the VPN connection (IKE traffic and perhaps HTTPS or DNS traffic) is not encapsulated by the VPN protocol (IPsec). The evaluator shall verify that the TSS section describes any differences in the routing of IP traffic when using any supported baseband protocols (e.g. Wi-Fi or, LTE).

Guidance
The evaluator shall verify that one (or more) of the following options is addressed by the documentation:

- The description above indicates that if a VPN client is enabled, all configurations route all Data Plane traffic through the tunnel interface established by the VPN client.
- The AGD guidance describes how the user and/or administrator can configure the TSF to meet this requirement.
- The API documentation includes a security function that allows a VPN client to specify this routing.

Tests

- Test FDP_IFC_EXT.1:1:If the ST author identifies any differences in the routing between Wi-Fi and

cellular protocols, the evaluator shall repeat this test with a base station implementing one of the identified cellular protocols.

Step 1: The evaluator shall enable a Wi-Fi configuration as described in the AGD guidance (as required by FTP_ITC_EXT.1). The evaluator shall use a packet sniffing tool between the wireless access point and an Internet-connected network. The evaluator shall turn on the sniffing tool and perform actions with the device such as navigating to websites, using provided applications, and accessing other Internet resources. The evaluator shall verify that the sniffing tool captures the traffic generated by these actions, turn off the sniffing tool, and save the session data.

Step 2: The evaluator shall configure an IPsec VPN client that supports the routing specified in this requirement, and if necessary, configure the device to perform the routing specified as described in the AGD guidance. The evaluator shall turn on the sniffing tool, establish the VPN connection, and perform the same actions with the device as performed in the first step. The evaluator shall verify that the sniffing tool captures traffic generated by these actions, turn off the sniffing tool, and save the session data.

Step 3: The evaluator shall examine the traffic from both step one and step two to verify that all Data Plane traffic is encapsulated by IPsec. The evaluator shall examine the Security Parameter Index (SPI) value present in the encapsulated packets captured in Step two from the TOE to the Gateway and shall verify this value is the same for all actions used to generate traffic through the VPN. Note that it is expected that the SPI value for packets from the Gateway to the TOE is different than the SPI value for packets from the TOE to the Gateway. The evaluator shall be aware that IP traffic on the cellular baseband outside of the IPsec tunnel may be emanating from the baseband processor and shall verify with the manufacturer that any identified traffic is not emanating from the application processor.

Step 4: The evaluator shall perform an ICMP echo from the TOE to the IP address of another device on the local wireless network and shall verify that no packets are sent using the sniffing tool. The evaluator shall attempt to send packets to the TOE outside the VPN tunnel (i.e. not through the VPN gateway), including from the local wireless network, and shall verify that the TOE discards them.

FDP_STG_EXT.1 User Data Storage

FDP_STG_EXT.1
TSS
The evaluator shall ensure the TSS describes the Trust Anchor Database implemented that contain certificates used to meet the requirements of this PP. This description shall contain information pertaining to how certificates are loaded into the store, and how the store is protected from unauthorized access (for example, UNIX permissions) in accordance with the permissions established in FMT_SMF_EXT.1 and FMT_MOF_EXT.1.1.

Guidance
There are no guidance evaluation activities for this component.

Tests
There are no test evaluation activities for this component.

FDP_UPC_EXT.1/APPS Inter-TSF User Data Transfer Protection (Applications)

FDP_UPC_EXT.1/APPS
The evaluator shall verify that the API documentation provided according to 5.2.2 includes the security functions (protection channel) described in these requirements, and verify that the APIs implemented to support this requirement include the appropriate settings/parameters so that the application can both provide and obtain the information needed to assure mutual identification of the endpoints of the communication as required by this component.

TSS
The evaluator shall examine the TSS to determine that it describes that all protocols listed in the TSS are specified and included in the requirements in the ST.

Guidance
The evaluator shall confirm that the operational guidance contains instructions necessary for configuring the protocol(s) selected for use by the applications.

Tests
Evaluation Activity Note: The following test requires the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on consumer Mobile Device products.

The evaluator shall write, or the developer shall provide access to, an application that requests protected channel services by the TSF. The evaluator shall verify that the results from the protected channel match the expected results according to the API documentation. This application may be used to assist in verifying the protected channel Evaluation Activities for the protocol requirements. The evaluator shall also perform the following tests:

- Test FDP_UPC_EXT.1/APPS:1:The evaluators shall ensure that the application is able to initiate communications with an external IT entity using each protocol specified in the requirement, setting up the connections as described in the operational guidance and ensuring that communication is successful.
- Test FDP_UPC_EXT.1/APPS:2:The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data are not sent in plaintext.

2.1.5 Class: Identification and Authentication (FIA)

FIA_AFL_EXT.1 Authentication Failure Handling

FIA_AFL_EXT.1
TSS
The evaluator shall ensure that the TSS describes that a value corresponding to the number of unsuccessful authentication attempts since the last successful authentication is kept for each Authentication Factor interface. The evaluator shall ensure that this description also includes if and how this value is maintained when the TOE loses power, either through a graceful powered off or an ungraceful loss of power. The evaluator shall ensure that if the value is not maintained, the interface is after another interface in the boot sequence for which the value is maintained.

If the TOE supports multiple authentication mechanisms, the evaluator shall ensure that this description also includes how the unsuccessful authentication attempts for each mechanism selected in FIA_UAU.5.1 is handled. The evaluator shall verify that the TSS describes if each authentication mechanism utilizes its own counter or if multiple authentication mechanisms utilize a shared counter. If multiple authentication mechanisms utilize a shared counter, the evaluator shall verify that the TSS describes this interaction.

The evaluator shall confirm that the TSS describes how the process used to determine if the authentication attempt was successful. The evaluator shall ensure that the counter would be updated even if power to the device is cut immediately following notifying the TOE user if the authentication attempt was successful or not.

Guidance
The evaluator shall verify that the AGD guidance describes how the administrator configures the maximum number of unique unsuccessful authentication attempts.

Tests

- Test FIA_AFL_EXT.1:1:The evaluator shall configure the device with all authentication mechanisms selected in FIA_UAU.5.1. The evaluator shall perform the following tests for each available authentication interface:

  Test 1a: The evaluator shall configure the TOE, according to the AGD guidance, with a maximum number of unsuccessful authentication attempts. The evaluator shall enter the locked state and enter incorrect passwords until the wipe occurs. The evaluator shall verify that the number of password entries corresponds to the configured maximum and that the wipe is implemented.

  Test 1b: [conditional] If the TOE supports multiple authentication mechanisms the previous test shall be repeated using a combination of authentication mechanisms confirming that the critical authentication mechanisms will cause the device to wipe and that when the maximum number of unsuccessful authentication attempts for a non-critical authentication mechanism is exceeded, the device limits authentication attempts to other available authentication mechanisms. If multiple authentication mechanisms utilize a shared counter, then the evaluator shall verify that the maximum number of unsuccessful authentication attempts can be reached by using each individual authentication mechanism and a combination of all authentication mechanisms that share the counter.
- Test FIA_AFL_EXT.1:2:The evaluator shall repeat test one, but shall power off (by removing the battery, if possible) the TOE between unsuccessful authentication attempts. The evaluator shall verify that the total number of unsuccessful authentication attempts for each authentication mechanism corresponds to the configured maximum and that the critical authentication mechanisms cause the device to wipe. Alternatively, if the number of authentication failures is not maintained for the interface under test, the evaluator shall verify that upon booting the TOE between unsuccessful authentication attempts another authentication factor interface is presented before the interface under test.

FIA_PMG_EXT.1 Password Management

FIA_PMG_EXT.1
TSS
There are no TSS evaluation activities for this component.

Guidance
The evaluator shall examine the operational guidance to determine that it provides guidance to security administrators on the composition of strong passwords, and that it provides instructions on setting the minimum password length. The evaluator shall also perform the following tests. Note that one or more of these tests can be performed with a single test case.

Tests

- Test FIA_PMG_EXT.1:1:The evaluator shall compose passwords that either meet the requirements, or fail to meet the requirements, in some way. For each password, the evaluator shall verify that the TOE supports the password. While the evaluator is not required (nor is it feasible) to test all possible compositions of passwords, the evaluator shall ensure that all characters, rule characteristics, and a minimum length listed in the requirement are supported, and justify the subset of those characters chosen for testing.

FIA_TRT_EXT.1 Authentication Throttling

FIA_TRT_EXT.1
TSS
The evaluator shall verify that the TSS describes the method by which authentication attempts are not able to be automated. The evaluator shall ensure that the TSS describes either how the TSF disables authentication via external interfaces (other than the ordinary user interface) or how authentication attempts are delayed in order to slow automated entry and shall ensure that this delay totals at least 500 milliseconds over 10 attempts for all authentication mechanisms selected in FIA_UAU.5.1.

Guidance
There are no guidance evaluation activities for this component.

Tests
There are no test evaluation activities for this component.

FIA_UAU.5 Multiple Authentication Mechanisms

FIA_UAU.5
TSS
The evaluator shall ensure that the TSS describes each mechanism provided to support user authentication and the rules describing how the authentication mechanism(s) provide authentication.

Specifically, for all authentication mechanisms specified in FIA_UAU.5.1, the evaluator shall ensure that the TSS describes 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 must be described. For example, whether the multiple BAFs can be enabled at the same time.

Guidance
The evaluator shall verify that configuration guidance for each authentication mechanism is addressed in the AGD guidance.

Tests

- Test FIA_UAU.5:1:For each authentication mechanism selected in FIA_UAU.5.1, the evaluator shall enable that mechanism and verify that it can be used to authenticate the user at the specified authentication factor interfaces.
- Test FIA_UAU.5:2:For each authentication mechanism rule, the evaluator shall ensure that the authentication mechanism(s) behave accordingly.

FIA_UAU.6 Re-Authentication

FIA_UAU.6
There are no TSS evaluation activities for this element.

There are no guidance evaluation activities for this element.

- Test FIA_UAU.6:1:The evaluator shall configure the TSF to use the Password Authentication Factor according to the AGD guidance. The evaluator shall change Password Authentication Factor according to the AGD guidance and verify that the TSF requires the entry of the Password Authentication Factor before allowing the factor to be changed.
- Test FIA_UAU.6:2:[conditional] For each BAF selected in FIA_UAU.5.1, the evaluator shall configure the TSF to use the BAF, which includes configuring the Password Authentication Factor, according to the AGD guidance. The evaluator shall change the BAF according to the AGD guidance and verify that the TSF requires the entry of the Password Authentication Factor before allowing the BAF to be changed.
- Test FIA_UAU.6:3:[conditional] If "hybrid" is selected in FIA_UAU.5.1, the evaluator shall configure the TSF to use the BAF and PIN or password, which includes configuring the Password Authentication Factor, according to the AGD guidance. The evaluator shall change the BAF and PIN according to the AGD guidance and verify that the TSF requires the entry of the Password Authentication Factor before allowing the factor to be changed.

TSS

There are no TSS evaluation activities for this element.

Guidance
There are no guidance evaluation activities for this element.

Tests

- Test FIA_UAU.6:1:The evaluator shall configure the TSF to transition to the locked state after a time of inactivity (FMT_SMF_EXT.1) according to the AGD guidance. The evaluator shall wait until the TSF locks and then verify that the TSF requires the entry of the Password Authentication Factor before transitioning to the unlocked state.
- Test FIA_UAU.6:2:[conditional] For each BAF selected in FIA_UAU.5.1, the evaluator shall repeat Test 1 verifying that the TSF requires the entry of the BAF before transitioning to the unlocked state.
- Test FIA_UAU.6:3:[conditional] If "hybrid" is selected in FIA_UAU.5.1, the evaluator shall repeat Test 1 verifying that the TSF requires the entry of the BAF and PIN/password before transitioning to the unlocked state.
- Test FIA_UAU.6:4:The evaluator shall configure user-initiated locking according to the AGD guidance. The evaluator shall lock the TSF and then verify that the TSF requires the entry of the Password Authentication Factor before transitioning to the unlocked state.
- Test FIA_UAU.6:5:[conditional] For each BAF selected in FIA_UAU.5.1, the evaluator shall repeat Test 4 verifying that the TSF requires the entry of the BAF before transitioning to the unlocked state.
- Test FIA_UAU.6:6:[conditional] If "hybrid" is selected in FIA_UAU.5.1, the evaluator shall repeat Test 4 verifying that the TSF requires the entry of the BAF and PIN/password before transitioning to the unlocked state.

FIA_UAU.7 Protected Authentication Feedback

FIA_UAU.7
TSS
The evaluator shall ensure that the TSS describes the means of obscuring the authentication entry, for all authentication methods specified in FIA_UAU.5.1.

Guidance
The evaluator shall verify that any configuration of this requirement is addressed in the AGD guidance and that the password is obscured by default.

Tests

- Test FIA_UAU.7:1:The evaluator shall enter passwords on the device, including at least the Password Authentication Factor at lock screen, and verify that the password is not displayed on the device.
- Test FIA_UAU.7:2:[conditional] For each BAF selected in FIA_UAU.5.1, the evaluator shall authenticate by producing a biometric sample at lock screen. As the biometric algorithms are performed, the evaluator shall verify that sensitive images, audio, or other information identifying the user are kept secret and are not revealed to the user. Additionally, the evaluator shall produce a biometric sample that fails to authenticate and verify that the reason(s) for authentication failure (user mismatch, low sample quality, etc.) are not revealed to the user. It is acceptable for the BAF to state that it was unable to physically read the biometric sample, for example, if the sensor is unclean or the biometric sample was removed too quickly. However, specifics regarding why the presented biometric sample failed authentication shall not be revealed to the user.

FIA_UAU_EXT.1 Authentication for Cryptographic Operation

FIA_UAU_EXT.1
TSS
The evaluator shall verify that the TSS section of the ST describes the process for decrypting protected data and keys. The evaluator shall ensure that this process requires the user to enter a Password Authentication Factor and, in accordance with FCS_CKM_EXT.3, derives a KEK, which is used to protect the software-based secure key storage and (optionally) DEK(s) for sensitive data, in accordance with FCS_STG_EXT.2.

Guidance
There are no guidance evaluation activities for this component.

Tests
The following tests may be performed in conjunction with FDP_DAR_EXT.1 and FDP_DAR_EXT.2.

Evaluation Activity Note: The following test require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on consumer Mobile Device products.

- Test FIA_UAU_EXT.1:1:The evaluator shall enable encryption of protected data and require user authentication according to the AGD guidance. The evaluator shall write, or the developer shall provide access to, an application that includes a unique string treated as protected data.

  The evaluator shall reboot the device, use a tool provided by developer to search for the unique string

within the application data, and verify that the unique string cannot be found. The evaluator shall enter the Password Authentication Factor to access full device functionality, use a tool provided by the developer to access the unique string within the application data, and verify that the unique string can be found.

- Test FIA_UAU_EXT.1:2:[conditional] The evaluator shall require user authentication according to the AGD guidance. The evaluator shall store a key in the software-based secure key storage.

  The evaluator shall lock the device, use a tool provided by developer to access the key within the stored data, and verify that the key cannot be retrieved or accessed. The evaluator shall enter the Password Authentication Factor to access full device functionality, use a tool provided by developer to access the key, and verify that the key can be retrieved or accessed.
- Test FIA_UAU_EXT.1:3:[conditional] The evaluator shall enable encryption of sensitive data and require user authentication according to the AGD guidance. The evaluator shall write, or the developer shall provide access to, an application that includes a unique string treated as sensitive data.

  The evaluator shall lock the device, use a tool provided by developer to attempt to access the unique string within the application data, and verify that the unique string cannot be found. The evaluator shall enter the Password Authentication Factor to access full device functionality, use a tool provided by developer to access the unique string within the application data, and verify that the unique string can be retrieved.

FIA_UAU_EXT.2 Timing of Authentication

FIA_UAU_EXT.2
TSS
The evaluator shall verify that the TSS describes the actions allowed by unauthorized users in the locked state.

Guidance
There are no guidance evaluation activities for this component.

Tests
The evaluator shall attempt to perform some actions not listed in the selection while the device is in the locked state and verify that those actions do not succeed.

FIA_X509_EXT.1 X.509 Validation of Certificates

FIA_X509_EXT.1
TSS
The evaluator shall ensure the TSS describes where the check of validity of the certificates takes place. The evaluator ensures the TSS also provides a description of the certificate path validation algorithm.

Guidance
There are no guidance evaluation activities for this component.

Tests
The tests described must be performed in conjunction with the other Certificate Services evaluation activities, including the use cases in FIA_X509_EXT.2.1 and FIA_X509_EXT.3. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. The evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA.

- Test FIA_X509_EXT.1:1: The evaluator shall demonstrate that validating a certificate without a valid certification path results in the function failing, for each of the following reasons, in turn:
  - by establishing a certificate path in which one of the issuing certificates is not a CA certificate,
  - by omitting the basicConstraints field in one of the issuing certificates,
  - by setting the basicConstraints field in an issuing certificate to have CA=False,
  - by omitting the CA signing bit of the key usage field in an issuing certificate, and
  - by setting the path length field of a valid CA field to a value strictly less than the certificate path.

  The evaluator shall then establish a valid certificate path consisting of valid CA certificates, and demonstrate that the function succeeds. The evaluator shall then remove trust in one of the CA certificates, and show that the function fails.
- Test FIA_X509_EXT.1:2:The evaluator shall demonstrate that validating an expired certificate results in the function failing.
- Test FIA_X509_EXT.1:3:The evaluator shall test that the TOE can properly handle revoked certificates-conditional on whether CRL, OCSP, OSCP stapling, or OCSP multi-stapling is selected; if multiple methods are selected, then the following tests shall be performed for each method:

  The evaluator shall test revocation of the node certificate.

  The evaluator shall also test revocation of the intermediate CA certificate (i.e. the intermediate CA certificate should be revoked by the root CA). For the test of the WLAN use case, only pre-stored CRLs are used. If OCSP stapling per RFC 6066 is the only supported revocation method, this test is omitted.

  The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The

evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails.
- Test FIA_X509_EXT.1:4:If any OCSP option is selected, the evaluator shall configure the OCSP server or use a man-in-the-middle tool to present a certificate that does not have the OCSP signing purpose and verify that validation of the OCSP response fails. If CRL is selected, the evaluator shall configure the CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set, and verify that validation of the CRL fails.
- Test FIA_X509_EXT.1:5:The evaluator shall modify any byte in the first eight bytes of the certificate and demonstrate that the certificate fails to validate (the certificate will fail to parse correctly).
- Test FIA_X509_EXT.1:6:The evaluator shall modify any bit in the last byte of the signature algorithm of the certificate and demonstrate that the certificate fails to validate (the signature on the certificate will not validate).
- Test FIA_X509_EXT.1:7:The evaluator shall modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate (the signature on the certificate will not validate).
- Test FIA_X509_EXT.1:8:
  - Test FIA_X509_EXT.1:8.1:(Conditional on support for EC certificates as indicated in FCS_COP.1(3)). The evaluator shall establish a valid, trusted certificate chain consisting of an EC leaf certificate, an EC Intermediate CA certificate not designated as a trust anchor, and an EC certificate designated as a trusted anchor, where the elliptic curve parameters are specified as a named curve. The evaluator shall confirm that the TOE validates the certificate chain.
  - Test FIA_X509_EXT.1:8.2:(Conditional on support for EC certificates as indicated in FCS_COP.1(3)). The evaluator shall replace the intermediate certificate in the certificate chain for Test 8a with a modified certificate, where the modified intermediate CA has a public key information field where the EC parameters uses an explicit format version of the Elliptic Curve parameters in the public key information field of the intermediate CA certificate from Test 8a, and the modified Intermediate CA certificate is signed by the trusted EC root CA, but having no other changes. The evaluator shall confirm the TOE treats the certificate as invalid.

FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2
TSS
The evaluator shall check the TSS to ensure that it describes how the TOE chooses which certificates to use, and any necessary instructions in the administrative guidance for configuring the operating environment so that the TOE can use the certificates.

The evaluator shall examine the TSS to confirm that it describes the behavior of the TOE when a connection cannot be established during the validity check of a certificate used in establishing a trusted channel. The evaluator shall verify that any distinctions between trusted channels are described.

Guidance
If the requirement that the administrator is able to specify the default action, then the evaluator shall ensure that the operational guidance contains instructions on how this configuration action is performed.

Tests
The evaluator shall perform the following test for each trusted channel:

- Test FIA_X509_EXT.2:1:The evaluator shall demonstrate that using a valid certificate that requires certificate validation checking to be performed in at least some part by communicating with a non-TOE IT entity. The evaluator shall then manipulate the environment so that the TOE is unable to verify the validity of the certificate, and observe that the action selected in FIA_X509_EXT.2.2 is performed. If the selected action is administrator-configurable, then the evaluator shall follow the operational guidance to determine that all supported administrator-configurable options behave in their documented manner.

FIA_X509_EXT.3 Request Validation of Certificates

FIA_X509_EXT.3
The evaluator shall verify that the API documentation provided according to 5.2.2 includes the security function (certificate validation) described in this requirement. This documentation shall be clear as to which results indicate success and failure.

TSS
There are no TSS evaluation activities for this component.

Guidance
There are no guidance evaluation activities for this component.

Tests
The evaluator shall write, or the developer shall provide access to, an application that requests certificate validation by the TSF. The evaluator shall verify that the results from the validation match the expected results according to the API documentation. This application may be used to verify that import, removal, modification, and validation are performed correctly according to the tests required by FDP_STG_EXT.1, FTP_ITC_EXT.1, FMT_SMF_EXT.1, and FIA_X509_EXT.1.

2.1.6 Class: Security Management (FMT)

FMT_MOF_EXT.1 Management of Security Functions Behavior

FMT_MOF_EXT.1
The evaluator shall verify that the TSS describes those management functions that may only be performed by the user and confirm that the TSS does not include an Administrator API for any of these management functions. This activity will be performed in conjunction with FMT_SMF_EXT.1.

There are no guidance evaluation activities for this component.

There are no test evaluation activities for this component.
TSS
The evaluator shall verify that the TSS describes those management functions that may be performed by the Administrator, to include how the user is prevented from accessing, performing, or relaxing the function (if applicable), and how applications/APIs are prevented from modifying the Administrator configuration. The TSS also describes any functionality that is affected by administrator-configured policy and how. This activity will be performed in conjunction with FMT_SMF_EXT.1.

Guidance
There are no guidance evaluation activities for this component.

Tests

- Test FMT_MOF_EXT.1:1:The evaluator shall use the test environment to deploy policies to Mobile Devices.
- Test FMT_MOF_EXT.1:2:The evaluator shall create policies which collectively include all management functions which are controlled by the (enterprise) administrator and cannot be overridden/relaxed by the user as defined in FMT_MOF_EXT.1.2. The evaluator shall apply these policies to devices, attempt to override/relax each setting both as the user (if a setting is available) and as an application (if an API is available), and ensure that the TSF does not permit it. Note that the user may still apply a more restrictive policy than that of the administrator.
- Test FMT_MOF_EXT.1:3:Additional testing of functions provided to the administrator are performed in conjunction with the testing activities for FMT_SMF_EXT.1.1.

FMT_SMF_EXT.1 Specification of Management Functions

FMT_SMF_EXT.1
The evaluator shall verify the TSS defines the allowable policy options: the range of values for both password length and lifetime, and a description of complexity to include character set and complexity policies (e.g., configuration and enforcement of number of uppercase, lowercase, and special characters per password).
The evaluator shall exercise the TSF configuration as the administrator and perform positive and negative tests, with at least two values set for each variable setting, for each of the following:

- minimum password length
- minimum password complexity
- maximum password lifetime

The evaluator shall verify the TSS defines the range of values for both timeout period and number of authentication failures for all supported authentication mechanisms.
The evaluator shall exercise the TSF configuration as the administrator. The evaluator shall perform positive and negative tests, with at least two values set for each variable setting, for each of the following:

- screen-lock enabled/disabled
- screen lock timeout
- number of authentication failures (may be combined with test for FIA_AFL_EXT.1)

The evaluator shall perform the following tests:

a. The evaluator shall exercise the TSF configuration to enable the VPN protection. These configuration actions must be used for the testing of the FDP_IFC_EXT.1.1 requirement.
b. [conditional] If "per-app basis" is selected, the evaluator shall create two applications and enable one to use the VPN and the other to not use the VPN. The evaluator shall exercise each application (attempting to access network resources; for example, by browsing different websites) individually while capturing packets from the TOE. The evaluator shall verify from the packet capture that the traffic from the VPN-enabled application is encapsulated in IPsec and that the traffic from the VPN-disabled application is not encapsulated in IPsec.
c. [conditional] If "per-groups of application basis" is selected, the evaluator shall create two applications and the applications shall be placed into different groups. Enable one application group to use the VPN and the other to not use the VPN. The evaluator shall exercise each application (attempting to access network resources; for example, by browsing different websites) individually while capturing packets from the TOE. The evaluator shall verify from the packet capture that the traffic from the application in the VPN-enabled group is encapsulated in IPsec and that the traffic from the application in the VPN-disabled group is not encapsulated in IPsec.

The evaluator shall verify that the TSS includes a description of each radio and an indication of if the radio can be enabled/disabled along with what role can do so. In addition the evaluator shall verify that the frequency ranges at which each radio operates is included in the TSS. The evaluator shall verify that the TSS includes at what point in the boot sequence the radios are powered on and indicates if the radios are used as part of the initialization of the device.
The evaluator shall confirm that the AGD guidance describes how to perform the enable/disable function for each radio.

The evaluator shall ensure that minimal signal leakage enters the RF shielded enclosure (i.e. Faraday bag, Faraday box, RF shielded room) by performing the following steps:

Step 1: Place the antenna of the spectrum analyzer inside the RF shielded enclosure.

Step 2: Enable "Max Hold" on the spectrum analyzer and perform a spectrum sweep of the frequency range between 300 MHz – 6000 MHz, in I kHz steps (this range should encompass 802.11, 802.15, GSM, UMTS, and LTE). This range will not address NFC 13.56.MHz, another test should be set up with similar constraints to address NFC.

If power above -90 dBm is observed, the Faraday box has too great of signal leakage and shall not be used to complete the test for Function 4. The evaluator shall exercise the TSF configuration as the administrator and, if not restricted to the administrator, the user, to enable and disable the state of each radio (e.g. Wi-Fi, cellular, NFC, Bluetooth). Additionally, the evaluator shall repeat the steps below, booting into any auxiliary boot mode supported by the device. For each radio, the evaluator shall:

Step 1: Place the antenna of the spectrum analyzer inside the RF shielded enclosure. Configure the spectrum analyzer to sweep desired frequency range for the radio to be tested (based on range provided in the TSS)). The ambient noise floor shall be set to -110 dBm. Place the TOE into the RF shielded enclosure to isolate them from all other RF traffic.

Step 2: The evaluator shall create a baseline of the expected behavior of RF signals. The evaluator shall power on the device, ensure the radio in question is enabled, power off the device, enable "Max Hold" on the spectrum analyzer and power on the device. The evaluator shall wait 2 minutes at each Authentication Factor interface prior to entering the necessary password to complete the boot process, waiting 5 minutes after the device is fully booted. The evaluator shall observe that RF spikes are present at the expected uplink channel frequency. The evaluator shall clear the "Max Hold" on the spectrum analyzer.

Step 3: The evaluator shall verify the absence of RF activity for the uplink channel when the radio in question is disabled. The evaluator shall complete the following test five times. The evaluator shall power on the device, ensure the radio in question is disabled, power off the device, enable "Max Hold" on the spectrum analyzer and power on the device. The evaluator shall wait 2 minutes at each Authentication Factor interface prior to entering the necessary password to complete the boot process, waiting 5 minutes after the device is fully booted. The evaluator shall clear the "Max Hold" on the spectrum analyzer. If the radios are used for device initialization, then a spike of RF activity for the uplink channel can be observed initially at device boot. However, if a spike of RF activity for the uplink channel of the specific radio frequency band is observed after the device is fully booted or at an Authentication Factor interface it is deemed that the radio is enabled.
The evaluator shall verify that the TSS includes a description of each collection device and an indication of if it can be enabled/disabled along with what role can do so. The evaluator shall confirm that the AGD guidance describes how to perform the enable/disable function.
The evaluator shall perform the following test(s):

a. The evaluator shall exercise the TSF configuration as the administrator and, if not restricted to the administrator, the user, to enable and disable the state of each audio or visual collection devices (e.g. camera, microphone) listed by the ST author. For each collection device, the evaluator shall disable the device and then attempt to use its functionality. The evaluator shall reboot the TOE and verify that disabled collection devices may not be used during or early in the boot process. Additionally, the evaluator shall boot the device into each available auxiliary boot mode and verify that the collection device cannot be used.
b. [conditional] If "per-app basis" is selected, the evaluator shall create two applications and enable one to use access the A/V device and the other to not access the A/V device. The evaluator shall exercise each application attempting to access the A/V device individually. The evaluator shall verify that the enabled application is able to access the A/V device and the disabled application is not able to access the A/V device.
c. [conditional] If "per-groups of application basis" is selected, the evaluator shall create two applications and the applications shall be placed into different groups. Enable one group to access the A/V device and the other to not access the A/V device. The evaluator shall exercise each application attempting to access the A/V device individually. The evaluator shall verify that the application in the enabled group is able to access the A/V device and the application in the disabled group is not able to access the A/V device.

The evaluator shall use the test environment to instruct the TSF, both as a user and as the administrator, to command the device to transition to a locked state, and verify that the device transitions to the locked state upon command.
The evaluator shall verify the TSS describes the allowable application installation policy options based on the selection included in the ST. If the application allowlist is selected, the evaluator shall verify that the TSS includes a description of each application characteristic upon which the allowlist may be based.
The evaluator shall exercise the TSF configuration as the administrator to restrict particular applications,

sources of applications, or application installation according to the AGD guidance. The evaluator shall attempt to install unauthorized applications and ensure that this is not possible. The evaluator shall, in conjunction, perform the following specific tests:

a. [conditional] The evaluator shall attempt to connect to an unauthorized repository in order to install applications.
b. [conditional] The evaluator shall attempt to install two applications (one allowlisted, and one not) from a known allowed repository and verify that the application not on the allowlist is rejected. The evaluator shall also attempt to side-load executables or installation packages via USB connections to determine that the white list is still adhered to

The evaluator shall verify that the TSS describes each category of keys/secrets that can be imported into the TSF's secure key storage.
The test of these functions is performed in association with FCS_STG_EXT.1.
The evaluator shall review the AGD guidance to determine that it describes the steps needed to import, modify, or remove certificates in the Trust Anchor database, and that the users that have authority to import those certificates (e.g., only administrator, or both administrators and users) are identified.
The evaluator shall import certificates according to the AGD guidance as the user and/or as the administrator, as determined by the administrative guidance. The evaluator shall verify that no errors occur during import.
The evaluator should perform an action requiring use of the X.509v3 certificate to provide assurance that installation was completed properly.
The evaluator shall verify that the TSS describes each additional category of X.509 certificates and their use within the TSF.
The evaluator shall remove an administrator-imported certificate and any other categories of certificates included in the assignment of function 14 from the Trust Anchor Database according to the AGD guidance as the user and as the administrator.
The evaluator shall examine the TSS to ensure that it contains a description of each management function that will be enforced by the enterprise once the device is enrolled. The evaluator shall examine the AGD guidance to determine that this same information is present.
The evaluator shall verify that user approval is required to enroll the device into management.
The evaluator shall verify that the TSS includes an indication of what applications (e.g., user-installed applications, Administrator-installed applications, or Enterprise applications) can be removed along with what role can do so. The evaluator shall examine the AGD guidance to determine that it details, for each type of application that can be removed, the procedures necessary to remove those applications and their associated data. For the purposes of this Evaluation Activity, "associated data" refers to data that are created by the app during its operation that do not exist independent of the app's existence, for instance, configuration data, or e-mail information that's part of an e-mail client. It does not, on the other hand, refer to data such as word processing documents (for a word processing app) or photos (for a photo or camera app).
The evaluator shall attempt to remove applications according to the AGD guidance and verify that the TOE no longer permits users to access those applications or their associated data.
The evaluator shall attempt to update the TSF system software following the procedures in the AGD guidance and verify that updates correctly install and that the version numbers of the system software increase.
The evaluator shall attempt to install an application following the procedures in the AGD guidance and verify that the application is installed and available on the TOE.
The evaluator shall attempt to remove any Enterprise applications from the device by following the administrator guidance. The evaluator shall verify that the TOE no longer permits users to access those applications or their associated data.
The evaluator shall examine the AGD Guidance to determine that it specifies, for at least each category of information selected for Function 18, how to enable and disable display information for that type of information in the locked state.
For each category of information listed in the AGD guidance, the evaluator shall verify that when that TSF is configured to limit the information according to the AGD, the information is no longer displayed in the locked state.
The evaluator shall exercise the TSF configuration as the administrator and, if not restricted to the administrator, the user, to enable system-wide data-at-rest protection according to the AGD guidance. The evaluator shall ensure that all Evaluation Activities for DAR (FDP_DAR) are conducted with the device in this configuration.
The evaluator shall exercise the TSF configuration as the administrator and, if not restricted to the administrator, the user, to enable removable media's data-at-rest protection according to the AGD guidance. The evaluator shall ensure that all Evaluation Activities for DAR (FDP_DAR) are conducted with the device in this configuration.
The evaluator shall perform the following tests.

a. The evaluator shall enable location services device-wide and shall verify that an application (such as a mapping application) is able to access the TOE's location information. The evaluator shall disable location services device-wide and shall verify that an application (such as a mapping application) is unable to access the TOE's location information.
b. [conditional] If "per-app basis" is selected, the evaluator shall create two applications and enable one to use access the location services and the other to not access the location services. The evaluator shall exercise each application attempting to access location services individually. The evaluator shall verify that the enabled application is able to access the location services and the disabled application is not able to access the location services.

The evaluator shall verify that the TSS states if the TOE supports a BAF and/or hybrid authentication. If the TOE does not include a BAF and/or hybrid authentication this test is implicitly met.

a. [conditional] If a BAF is selected the evaluator shall verify that the TSS describes the procedure to enable/disable the BAF. If the TOE includes multiple BAFs, the evaluator shall verify that the TSS describes how to enable/disable each BAF, specifically if the different modalities can be individually enabled/disabled. The evaluator shall configure the TOE to allow each supported BAF to authenticate and verify that successful authentication can be achieved using the BAF. The evaluator shall configure the TOE to disable the use of each supported BAF for authentication and confirm that the BAF cannot be used to authenticate.
b. [conditional] If "Hybrid" is selected the evaluator shall verify that the TSS describes the procedure to enable/disable the hybrid (biometric credential and PIN/password) authentication. The evaluator shall configure the TOE to allow hybrid authentication to authenticate and confirm that successful authentication can be achieved using the hybrid authentication. The evaluator shall configure the TOE to disable the use of hybrid authentication and confirm that the hybrid authentication cannot be used to authenticate.

The test of this function is performed in conjunction with FIA_X509_EXT.2.2, FCS_TLSC_EXT.1.3 in the Package for Transport Layer Security.
The evaluator shall verify that the TSS includes a list of each externally accessible hardware port and an indication of if data transfer over that port can be enabled/disabled. AGD guidance will describe how to perform the enable/disable function.
The evaluator shall exercise the TSF configuration to enable and disable data transfer capabilities over each externally accessible hardware ports (e.g. USB, SD card, HDMI) listed by the ST author. The evaluator shall use test equipment for the particular interface to ensure that no low-level signaling is occurring on all pins used for data transfer when they are disabled. For each disabled data transfer capability, the evaluator shall repeat this test by rebooting the device into the normal operational mode and verifying that the capability is disabled throughout the boot and early execution stage of the device.
The evaluator shall verify that the TSS describes how the TSF acts as a server in each of the protocols listed in the ST, and the reason for acting as a server.
The evaluator shall attempt to disable each listed protocol in the assignment. The evaluator shall verify that remote devices can no longer access the TOE or TOE resources using any disabled protocols.
The evaluator shall exercise the TSF configuration as the administrator and, if not restricted to the administrator, the user, to enable and disable any developer mode. The evaluator shall test that developer mode access is not available when its configuration is disabled. The evaluator shall verify the developer mode remains disabled during device reboot.
The evaluator shall examine the AGD guidance to determine that it describes how to enable and disable any "Forgot Password", password hint, or remote authentication (to bypass local authentication mechanisms) capability.
For each mechanism listed in the AGD guidance that provides a "Forgot Password" feature or other means where the local authentication process can be bypassed, the evaluator shall disable the feature and ensure that they are not able to bypass the local authentication process.
The evaluator shall attempt to wipe Enterprise data resident on the device according to the administrator guidance. The evaluator shall verify that the data is no longer accessible by the user.
The evaluator shall verify that the TSS describes how approval for an application to perform the selected action (import, removal) with respect to certificates in the Trust Anchor Database is accomplished (e.g., a pop-up, policy setting, etc.).

The evaluator shall also verify that the API documentation provided according to 5.2.2 includes any security functions (import, modification, or destruction of the Trust Anchor Database) allowed by applications.
The evaluator shall perform one of the following tests:

a. [conditional] If applications may import certificates to the Trust Anchor Database, the evaluator shall write, or the developer shall provide access to, an application that imports a certificate into the Trust Anchor Database. The evaluator shall verify that the TOE requires approval before allowing the application to import the certificate:
    - The evaluator shall deny the approvals to verify that the application is not able to import the certificate. Failure of import shall be tested by attempting to validate a certificate that chains to the certificate whose import was attempted (as described in the evaluation activity for FIA_X509_EXT.1).
    - The evaluator shall repeat the test, allowing the approval to verify that the application is able to import the certificate and that validation occurs.
b. [conditional] If applications may remove certificates in the Trust Anchor Database, the evaluator shall write, or the developer shall provide access to, an application that removes certificates from the Trust Anchor Database. The evaluator shall verify that the TOE requires approval before allowing the application to remove the certificate:
    - The evaluator shall deny the approvals to verify that the application is not able to remove the certificate. Failure of removal shall be tested by attempting to validate a certificate that chains to the certificate whose removal was attempted (as described in the evaluation activity for FIA_X509_EXT.1).

The evaluator shall repeat the test, allowing the approval to verify that the application is able to remove/modify the certificate and that validation no longer occurs.
The test of this function is performed in conjunction with FIA_X509_EXT.2.2.
The evaluator shall ensure that the TSS describes which cellular protocols can be disabled.
The evaluator shall confirm that the AGD guidance describes the procedure for disabling each cellular protocol identified in the TSS.
The evaluator shall attempt to disable each cellular protocol according to the administrator guidance. The

evaluator shall attempt to connect the device to a cellular network and, using network analysis tools, verify that the device does not allow negotiation of the disabled protocols.
The evaluator shall attempt to read any device audit logs according to the administrator guidance and verify that the logs may be read. This test may be performed in conjunction with the evaluation activity of FAU_GEN.1.
The test of this function is performed in conjunction with FPT_TUD_EXT.5.1.
The evaluator shall verify that the TSS describes how the approval for exceptions for shared use of keys/secrets by multiple applications is accomplished (e.g., a pop-up, policy setting, etc.).
The test of this function is performed in conjunction with FCS_STG_EXT.1.
The evaluator shall verify that the TSS describes how the approval for exceptions for destruction of keys/secrets by applications that did not import the key/secret is accomplished (e.g., a pop-up, policy setting, etc.).
The test of this function is performed in conjunction with FCS_STG_EXT.1.
The evaluator shall verify that the TSS describes any restrictions in banner settings (e.g., character limitations).
The test of this function is performed in conjunction with FTA_TAB.1.
The test of this function is performed in conjunction with FAU_SEL.1.
The test of this function is performed in conjunction with FPT_NOT_EXT.2.1.
The evaluator shall verify that the TSS includes a description of how data transfers can be managed over USB.
The evaluator shall perform the following tests based on the selections made in the table:

a. [conditional] The evaluator shall disable USB mass storage mode, attach the device to a computer, and verify that the computer cannot mount the TOE as a drive. The evaluator shall reboot the TOE and repeat this test with other supported auxiliary boot modes.
b. [conditional] The evaluator shall disable USB data transfer without user authentication, attach the device to a computer, and verify that the TOE requires user authentication before the computer can access TOE data. The evaluator shall reboot the TOE and repeat this test with other supported auxiliary boot modes.
c. [conditional] The evaluator shall disable USB data transfer without connecting system authentication, attach the device to a computer, and verify that the TOE requires connecting system authentication before the computer can access TOE data. The evaluator shall then connect the TOE to another computer and verify that the computer cannot access TOE data. The evaluator shall then connect the TOE to the original computer and verify that the computer can access TOE data.

The evaluator shall verify that the TSS includes a description of available backup methods that can be enabled/disabled. If "selected applications" or "selected groups of applications" are selected the TSS shall include which applications of groups of applications backup can be enabled/disabled.
If "all applications" is selected, the evaluator shall disable each selected backup location in turn and verify that the TOE cannot complete a backup. The evaluator shall then enable each selected backup location in turn and verify that the TOE can perform a backup.

If "selected applications" is selected, the evaluator shall disable each selected backup location in turn and verify that for the selected application the TOE prevents backup from occurring. The evaluator shall then enable each selected backup location in turn and verify that for the selected application the TOE can perform a backup.

If "selected groups of applications" is selected, the evaluator shall disable each selected backup location in turn and verify that for a group of applications the TOE prevents the backup from occurring. The evaluator shall then enable each selected backup location in turn and verify for the group of application the TOE can perform a backup.

If "configuration data" is selected, the evaluator shall disable each selected backup location in turn and verify that the TOE prevents the backup of configuration data from occurring. The evaluator shall then enable each selected backup location in turn and verify that the TOE can perform a backup of configuration data.
The evaluator shall verify that the TSS includes a description of Hotspot functionality and USB tethering to include any authentication for these.
The evaluator shall perform the following tests based on the selections in Function 41.

a. [conditional] The evaluator shall enable hotspot functionality with each of the of the support authentication methods. The evaluator shall connect to the hotspot with another device and verify that the hotspot functionality requires the configured authentication method.
b. [conditional] The evaluator shall enable USB tethering functionality with each of the of the support authentication methods. The evaluator shall connect to the TOE over USB with another device and verify that the tethering functionality requires the configured authentication method.

The test of this function is performed in conjunction with FDP_ACF_EXT.1.2.
The evaluator shall set a policy to cause a designated application to be placed into a particular application group. The evaluator shall then install the designated application and verify that it was placed into the correct group.
The evaluator shall attempt to unenroll the device from management and verify that the steps described in FMT_SMF_EXT.2.1 are performed. This test should be performed in conjunction with the FMT_SMF_EXT.2.1 evaluation activity.
The evaluator shall verify that the TSS contains guidance to configure the VPN as Always-On.
The evaluator shall configure the VPN as Always-On and perform the following test.

a. The evaluator shall verify that when the VPN is connected all traffic is routed through the VPN. This test is performed in conjunction with FDP_IFC_EXT.1.1.
b. The evaluator shall verify that when the VPN is not established, that no traffic leaves the device. The evaluator shall ensure that the TOE has network connectivity and that the VPN is established. The evaluator shall use a packet sniffing tool to capture the traffic leaving the TOE. The evaluator shall disable the VPN connection on the server side. The evaluator shall perform actions with the device such as navigating to websites, using provided applications, and accessing other Internet resources and verify that no traffic leaves the device.
c. The evaluator shall verify that the TOE has network connectivity and that the VPN is established. The evaluator shall disable network connectivity (i.e. Airplane Mode) and verify that the VPN disconnects. The evaluator shall re-establish network connectivity and verify that the VPN automatically reconnects.

The evaluator shall verify that the TSS describes the procedure to revoke a biometric credential stored on the TOE.
The evaluator shall configure the TOE to use BAF and confirm that the biometric can be used to authenticate to the device. The evaluator shall revoke the biometric credential's ability to authenticate to the TOE and confirm that the same BAF cannot be used to authenticate to the device.
The evaluator shall verify that the TSS describes all assigned security management functions and their intended behavior.
The evaluator shall design and perform tests to demonstrate that the function may be configured and that the intended behavior of the function is enacted by the TOE.
TSS
The evaluator shall verify that the TSS describes all management functions, what role(s) can perform each function, and how these functions are (or can be) restricted to the roles identified by FMT_MOF_EXT.1.

The following activities are organized according to the function number in the table. These activities include TSS Evaluation Activities, AGD Evaluation Activities, and test activities.

Test activities specified below shall take place in the test environment described in the evaluation activity for FPT_TUD_EXT.1.

Guidance
The evaluator shall consult the AGD guidance to perform each of the specified tests, iterating each test as necessary if both the user and administrator may perform the function. The evaluator shall verify that the AGD guidance describes how to perform each management function, including any configuration details. For each specified management function tested, the evaluator shall confirm that the underlying mechanism exhibits the configured setting.

FMT_SMF_EXT.2 Specification of Remediation Actions

FMT_SMF_EXT.2
TSS
The evaluator shall verify that the TSS describes all available remediation actions, when they are available for use, and any other administrator-configured triggers. The evaluator shall verify that the TSS describes how the remediation actions are provided to the administrator.

Guidance
There are no guidance evaluation activities for this component.

Tests
The evaluator shall use the test environment to iteratively configure the device to perform each remediation action in the selection. The evaluator shall configure the remediation action per how the TSS states it is provided to the administrator. The test environment could be a MDM agent application, but can also be an application with administrator access.

2.1.7 Class: Protection of the TSF (FPT)

FPT_AEX_EXT.1 Application Address Space Layout Randomization

FPT_AEX_EXT.1
TSS
The evaluator shall ensure that the TSS section of the ST describes how the 8 bits are generated and provides a justification as to why those bits are unpredictable.

Guidance
There are no guidance evaluation activities for this component.

Tests
Evaluation Activity Note: The following test require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on consumer Mobile Device products.

- Test FPT_AEX_EXT.1:1:The evaluator must select 3 apps included with the TSF. These must include any web browser or mail client included with the TSF. For each of these apps, the evaluator shall launch the

same app on two separate Mobile Devices of the same type and compare all memory mapping locations. The evaluator must ensure that no memory mappings are placed in the same location on both devices.

If the rare (at most 1/256) chance occurs that two mappings are the same for a single app and not the same for the other two apps, the evaluator shall repeat the test with that app to verify that in the second test the mappings are different.

FPT_AEX_EXT.2 Memory Page Permissions

FPT_AEX_EXT.2
TSS
The evaluator shall ensure that the TSS describes of the memory management unit (MMU), and ensures that this description documents the ability of the MMU to enforce read, write, and execute permissions on all pages of virtual memory.

Guidance
There are no guidance evaluation activities for this component.

Tests
There are no test evaluation activities for this component.

FPT_AEX_EXT.3 Stack Overflow Protection

FPT_AEX_EXT.3
TSS
The evaluator shall determine that the TSS contains a description of stack-based buffer overflow protections implemented in the TSF software which runs in the non-privileged execution mode of the application processor. The exact implementation of stack-based buffer overflow protection will vary by platform. Example implementations may be activated through compiler options such as "-fstack-protector-all", "-fstack-protector", and "/GS" flags. The evaluator shall ensure that the TSS contains an inventory of TSF binaries and libraries, indicating those that implement stack-based buffer overflow protections as well as those that do not. The TSS must provide a rationale for those binaries and libraries that are not protected in this manner.

Guidance
There are no guidance evaluation activities for this component.

Tests
There are no test evaluation activities for this component.

FPT_AEX_EXT.4 Domain Isolation

FPT_AEX_EXT.4
TSS
The evaluator shall ensure that the TSS describes the mechanisms that are in place that prevents non-TSF software from modifying the TSF software or TSF data that governs the behavior of the TSF. These mechanisms could range from hardware-based means (e.g. "execution rings" and memory management functionality); to software-based means (e.g. boundary checking of inputs to APIs). The evaluator determines that the described mechanisms appear reasonable to protect the TSF from modification.

The evaluator shall ensure the TSS describes how the TSF ensures that the address spaces of applications are kept separate from one another.

The evaluator shall ensure the TSS details the USSD and MMI codes available from the dialer at the locked state or during auxiliary boot modes that may alter the behavior of the TSF. The evaluator shall ensure that this description includes the code, the action performed by the TSF, and a justification that the actions performed do not modify user or TSF data. If no USSD or MMI codes are available, the evaluator shall ensure that the TSS provides a description of the method by which actions prescribed by these codes are prevented.

The evaluator shall ensure the TSS documents any TSF data (including software, execution context, configuration information, and audit logs) which may be accessed and modified over a wired interface in auxiliary boot modes. The evaluator shall ensure that the description includes data, which is modified in support of update or restore of the device. The evaluator shall ensure that this documentation includes the auxiliary boot modes in which the data may be modified, the methods for entering the auxiliary boot modes, the location of the data, the manner in which data may be modified, the data format and packaging necessary to support modification, and software and/or hardware tools, if any, which are necessary for modifying the data.

The evaluator shall ensure that the TSS provides a description of the means by which unauthorized and undetected modification (that is, excluding cryptographically verified updates per FPT_TUD_EXT.2) of the TSF data over the wired interface in auxiliary boots modes is prevented. The lack of publicly available tools is not sufficient justification. Examples of sufficient justification include auditing of changes, cryptographic verification in the form of a digital signature or hash, disabling the auxiliary boot modes, and access control mechanisms that prevent writing to files or flashing partitions.

Guidance
There are no guidance evaluation activities for this component.

Tests
Evaluation Activity Note: The following tests require the vendor to provide access to a test platform that provides the evaluator with tools that are typically not found on consumer Mobile Device products. In addition, the vendor provides a list of files (e.g., system files, libraries, configuration files, audit logs) that make up the TSF data. This list could be organized by folders/directories (e.g., /usr/sbin, /etc), as well as individual files that may exist outside of the identified directories.

- Test FPT_AEX_EXT.4:1:The evaluator shall create and load an app onto the Mobile Device. This app shall attempt to traverse over all file systems and report any locations to which data can be written or overwritten. The evaluator must ensure that none of these locations are part of the OS software, device drivers, system and security configuration files, key material, or another untrusted application's image/data. For example, it is acceptable for a trusted photo editor app to have access to the data created by the camera app, but a calculator application shall not have access to the pictures.
- Test FPT_AEX_EXT.4:2:For each available auxiliary boot mode, the evaluator shall attempt to modify a TSF file of their choosing using the software and/or hardware tools described in the TSS. The evaluator shall verify that the modification fails.

FPT_JTA_EXT.1 JTAG Disablement

FPT_JTA_EXT.1
TSS
If "disable access through hardware" is selected:
The evaluator shall examine the TSS to determine the location of the JTAG ports on the TSF, to include the order of the ports (i.e. Data In, Data Out, Clock, etc.).

If "control access by a signing key" is selected:
The evaluator shall examine the TSS to determine how access to the JTAG is controlled by a signing key. The evaluator shall examine the TSS to determine when the JTAG can be accessed, i.e. what has the access to the signing key.

Guidance
There are no guidance evaluation activities for this component.

Tests
Evaluation Activity Note: The following test requires the developer to provide access to a test platform that provides the evaluator with chip level access.

If "disable access through hardware" is selected:
The evaluator shall connect a packet analyzer to the JTAG ports. The evaluator shall query the JTAG port for its device ID and confirm that the device ID cannot be retrieved.

FPT_KST_EXT.1 Key Storage

FPT_KST_EXT.1
TSS
The evaluator shall consult the TSS section of the ST in performing the Evaluation Activities for this requirement.

In performing their review, the evaluator shall determine that the TSS contains a description of the activities that happen on power-up and password authentication relating to the decryption of DEKs, stored keys, and data.

The evaluator shall ensure that the description also covers how the cryptographic functions in the FCS requirements are being used to perform the encryption functions, including how the KEKs, DEKs, and stored keys are unwrapped, saved, and used by the TOE so as to prevent plaintext from being written to non-volatile storage. The evaluator shall ensure that the TSS describes, for each power-down scenario how the TOE ensures that all keys in non-volatile storage are not stored in plaintext.

The evaluator shall ensure that the TSS describes how other functions available in the system (e.g., regeneration of the keys) ensure that no unencrypted key material is present in persistent storage.

The evaluator shall review the TSS to determine that it makes a case that key material is not written unencrypted to the persistent storage.

For each BAF selected in FIA_UAU.5.1:

The evaluator shall determine that the TSS also contains a description of the activities that happen on biometric authentication, relating to the decryption of DEKs, stored keys, and data. In addition how the system ensures that the biometric keying material is not stored unencrypted in persistent storage.

Guidance
There are no guidance evaluation activities for this component.

Tests
There are no test evaluation activities for this component.

FPT_KST_EXT.2 No Key Transmission

FPT_KST_EXT.2
TSS
The evaluator shall consult the TSS section of the ST in performing the Evaluation Activities for this requirement. The evaluator shall ensure that the TSS describes the TOE security boundary. The cryptographic module may very well be a particular kernel module, the Operating System, the Application Processor, or up to the entire Mobile Device.

In performing their review, the evaluator shall determine that the TSS contains a description of the activities that happen on power-up and password authentication relating to the decryption of DEKs, stored keys, and data.

The evaluator shall ensure that the TSS describes how other functions available in the system (e.g., regeneration of the keys) ensure that no unencrypted key material is transmitted outside the security boundary of the TOE.

The evaluator shall review the TSS to determine that it makes a case that key material is not transmitted outside the security boundary of the TOE.

For each BAF selected in FIA_UAU.5.1:

In performing their review, the evaluator shall determine that the TSS contains a description of the activities that happen on biometric authentication, including how any plaintext material, including critical security parameters and results of biometric algorithms, are protected and accessed.

The evaluator shall ensure that the TSS describes how functions available in the biometric algorithms ensure that no unencrypted plaintext material, including critical security parameters and intermediate results, is transmitted outside the security boundary of the TOE or to other functions or systems that transmit information outside the security boundary of the TOE.

Guidance
There are no guidance evaluation activities for this component.

Tests
There are no test evaluation activities for this component.

FPT_KST_EXT.3 No Plaintext Key Export

FPT_KST_EXT.3
TSS
The ST author will provide a statement of their policy for handling and protecting keys. The evaluator shall check to ensure the TSS describes a policy in line with not exporting either plaintext DEKs, KEKs, or keys stored in the secure key storage.

Guidance
There are no guidance evaluation activities for this component.

Tests
There are no test evaluation activities for this component.

FPT_NOT_EXT.1 Self-Test Notification

FPT_NOT_EXT.1
TSS
The evaluator shall verify that the TSS describes critical failures that may occur and the actions to be taken upon these critical failures.

Guidance
There are no guidance evaluation activities for this component.

Tests
Evaluation Activity Note: The following test require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on consumer Mobile Device products.

- Test FPT_NOT_EXT.1:1:The evaluator shall use a tool provided by the developer to modify files and

processes in the system that correspond to critical failures specified in the second list. The evaluator shall verify that creating these critical failures causes the device to take the remediation actions specified in the first list.

FPT_STM.1 Reliable Time Stamps

FPT_STM.1
TSS
The evaluator shall examine the TSS to ensure that it lists each security function that makes use of time. The TSS provides a description of how the time is maintained and considered reliable in the context of each of the time related functions. This documentation must identify whether the TSF uses a NTP server or the carrier's network time as the primary time sources.

Guidance
The evaluator examines the operational guidance to ensure it describes how to set the time.

Tests

- Test FPT_STM.1:1:The evaluator uses the operational guide to set the time. The evaluator shall then use an available interface to observe that the time was set correctly.

FPT_TST_EXT.1 TSF Cryptographic Functionality Testing

FPT_TST_EXT.1
TSS
The evaluator shall examine the TSS to ensure that it specifies the self-tests that are performed at start-up. This description must include an outline of the test procedures conducted by the TSF (e.g., rather than saying "memory is tested", a description similar to "memory is tested by writing a value to each memory location and reading it back to ensure it is identical to what was written" shall be used). The TSS must include any error states that they TSF may enter when self-tests fail, and the conditions and actions necessary to exit the error states and resume normal operation. The evaluator shall verify that the TSS indicates these self-tests are run at start-up automatically, and do not involve any inputs from or actions by the user or operator.

The evaluator shall inspect the list of self-tests in the TSS and verify that it includes algorithm self-tests. The algorithm self-tests will typically be conducted using known answer tests.

Guidance
There are no guidance evaluation activities for this component.

Tests
There are no test evaluation activities for this component.

FPT_TST_EXT.2/PREKERNEL TSF Integrity Checking (Pre-Kernel)

FPT_TST_EXT.2/PREKERNEL
TSS
The evaluator shall verify that the TSS section of the ST includes a description of the boot procedures, including a description of the entire bootchain, of the software for the TSF's Application Processor. The evaluator shall ensure that before loading the bootloader(s) for the operating system and the kernel, all bootloaders and the kernel software itself is cryptographically verified. For each additional category of executable code verified before execution, the evaluator shall verify that the description in the TSS describes how that software is cryptographically verified.

The evaluator shall verify that the TSS contains a justification for the protection of the cryptographic key or hash, preventing it from being modified by unverified or unauthenticated software. The evaluator shall verify that the TSS contains a description of the protection afforded to the mechanism performing the cryptographic verification.

The evaluator shall verify that the TSS describes each auxiliary boot mode available on the TOE during the boot procedures. The evaluator shall verify that, for each auxiliary boot mode, a description of the cryptographic integrity of the executed code through the kernel is verified before each execution.

Guidance
There are no guidance evaluation activities for this component.

Tests
Evaluation Activity Note: The following tests require the vendor to provide access to a test platform that provides the evaluator with tools that are typically not found on consumer Mobile Device products.

The evaluator shall perform the following tests:

- Test FPT_TST_EXT.2/PREKERNEL:1:The evaluator shall perform actions to cause TSF software to load and observe that the integrity mechanism does not flag any executables as containing integrity errors

and that the TOE properly boots.
- Test FPT_TST_EXT.2/PREKERNEL:2:The evaluator shall modify a TSF executable that is integrity protected and cause that executable to be successfully loaded by the TSF. The evaluator observes that an integrity violation is triggered and the TOE does not boot. (Care must be taken so that the integrity violation is determined to be the cause of the failure to load the module, and not the fact that the module was modified so that it was rendered unable to run because its format was corrupt).
- Test FPT_TST_EXT.2/PREKERNEL:3:[conditional] If the ST author indicates that the integrity verification is performed using a public key, the evaluator shall verify that the update mechanism includes a certificate validation according to FIA_X509_EXT.1. The evaluator shall digitally sign the TSF executable with a certificate that does not have the Code Signing purpose in the extendedKeyUsage field and verify that an integrity violation is triggered. The evaluator shall repeat the test using a certificate that contains the Code Signing purpose and verify that the integrity verification succeeds. Ideally, the two certificates should be identical except for the extendedKeyUsage field.

FPT_TUD_EXT.1 Trusted Update: TSF Version Query

FPT_TUD_EXT.1
The evaluator shall establish a test environment consisting of the Mobile Device and any supporting software that demonstrates usage of the management functions. This can be test software from the developer, a reference implementation of management software from the developer, or other commercially available software. The evaluator shall set up the Mobile Device and the other software to exercise the management functions according to the provided guidance documentation.

TSS
There are no TSS evaluation activities for this component.

Guidance
There are no guidance evaluation activities for this component.

Tests

- Test FPT_TUD_EXT.1:1:Using the AGD guidance provided, the evaluator shall test that the administrator and user can query:
  - the current version of the TSF operating system and any firmware that can be updated separately
  - the hardware model of the TSF
  - the current version of all installed mobile applications

The evaluator must review manufacturer documentation to ensure that the hardware model identifier is sufficient to identify the hardware which comprises the device.

FPT_TUD_EXT.2 TSF Update Verification

FPT_TUD_EXT.2
TSS
The evaluator shall verify that the TSS section of the ST describes all TSF software update mechanisms for updating the system software. The evaluator shall verify that the description includes a digital signature verification of the software before installation and that installation fails if the verification fails. The evaluator shall verify that all software and firmware involved in updating the TSF is described and, if multiple stages and software are indicated, that the software/firmware responsible for each stage is indicated and that the stage(s) which perform signature verification of the update are identified.

The evaluator shall verify that the TSS describes the method by which the digital signature is verified and that the public key used to verify the signature is either hardware-protected or is validated to chain to a public key in the Trust Anchor Database. If hardware-protection is selected, the evaluator shall verify that the method of hardware-protection is described and that the ST author has justified why the public key may not be modified by unauthorized parties.

[conditional] If the ST author indicates that software updates to system software running on other processors is verified, the evaluator shall verify that these other processors are listed in the TSS and that the description includes the software update mechanism for these processors, if different than the update mechanism for the software executing on the Application Processor.

[conditional] If the ST author indicates that the public key is used for software update digital signature verification, the evaluator shall verify that the update mechanism includes a certificate validation according to FIA_X509_EXT.1 and a check for the Code Signing purpose in the extendedKeyUsage.

Guidance
There are no guidance evaluation activities for this component.

Tests
The evaluator shall verify that the developer has provided evidence that the following tests were performed for each available update mechanism:

- Test FPT_TUD_EXT.2:1: The tester shall try to install an update without the digital signature and shall

verify that installation fails. The tester shall attempt to install an update with digital signature, and verify that installation succeeds.
- Test FPT_TUD_EXT.2:2:The tester shall digitally sign the update with a key disallowed by the device and verify that installation fails. The tester shall attempt to install an update signed with the allowed key and verify that installation succeeds.
- Test FPT_TUD_EXT.2:3:[conditional] The tester shall digitally sign the update with an invalid certificate and verify that update installation fails. The tester attempt to install an update that was digitally signed using a valid certificate and a certificate that contains the purpose and verify that the update installation succeeds.
- Test FPT_TUD_EXT.2:4:[conditional] The tester shall repeat these test for the software executing on each processor listed in the first selection. The tester shall attempt to install an update without the digital signature and shall verify that installation fails. The tester shall attempt to install an update with digital signature, and verify that installation succeeds.

FPT_TUD_EXT.3 Application Signing

FPT_TUD_EXT.3
TSS
The evaluator shall verify that the TSS describes how mobile application software is verified at installation. The evaluator shall ensure that this method uses a digital signature.

Guidance
There are no guidance evaluation activities for this component.

Tests
Evaluation Activity Note: The following test does not have to be tested using the commercial application store.

- Test FPT_TUD_EXT.3:1:The evaluator shall write, or the developer shall provide access to, an application. The evaluator shall try to install this application without a digitally signature and shall verify that installation fails. The evaluator shall attempt to install a digitally signed application, and verify that installation succeeds.

2.1.8 Class: TOE Access (FTA)

FTA_SSL_EXT.1 TSF- and User-initiated Locked State

FTA_SSL_EXT.1
TSS
The evaluator shall verify the TSS describes the actions performed upon transitioning to the locked state.

Guidance
The evaluation shall verify that the AGD guidance describes the method of setting the inactivity interval and of commanding a lock. The evaluator shall verify that the TSS describes the information allowed to be displayed to unauthorized users.

Tests

- Test FTA_SSL_EXT.1:1:The evaluator shall configure the TSF to transition to the locked state after a time of inactivity (FMT_SMF_EXT.1) according to the AGD guidance. The evaluator shall wait until the TSF locks and verify that the display is cleared or overwritten and that the only actions allowed in the locked state are unlocking the session and those actions specified in FIA_UAU_EXT.2.
- Test FTA_SSL_EXT.1:2:The evaluator shall command the TSF to transition to the locked state according to the AGD guidance as both the user and the administrator. The evaluator shall wait until the TSF locks and verify that the display is cleared or overwritten and that the only actions allowed in the locked state are unlocking the session and those actions specified in FIA_UAU_EXT.2.

2.1.9 Class: Trusted Path/Channels (FTP)

FTP_ITC_EXT.1 Trusted Channel Communication

FTP_ITC_EXT.1
TSS
The evaluator shall examine the TSS to determine that it describes the details of the TOE connecting to access points, VPN Gateways, and other trusted IT products in terms of the cryptographic protocols specified in the requirement, along with TOE-specific options or procedures that might not be reflected in the specifications. The evaluator shall also confirm that all protocols listed in the TSS are specified and included in the requirements in the ST.

If OTA updates are selected, the TSS shall describe which trusted channel protocol is initiated by the TOE and is used for updates.

Guidance
The evaluator shall confirm that the operational guidance contains instructions for establishing the connection to access points, VPN Gateways, and other trusted IT products.

Tests
The evaluator shall also perform the following tests for each protocol listed:

- Test FTP_ITC_EXT.1:1:The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data are not sent in plaintext and that a protocol analyzer identifies the traffic as the protocol under testing.
- Test FTP_ITC_EXT.1:2:[conditional] If IPsec is selected, the evaluator shall ensure that the TOE is able to initiate communications with a VPN Gateway, setting up the connections as described in the operational guidance and ensuring that communication is successful.
- Test FTP_ITC_EXT.1:3:[conditional]If OTA updates are selected, the evaluator shall trigger an update request according to the operational guidance and shall ensure that the communication is successful.
- Test FTP_ITC_EXT.1:4:For any other selected protocol (not tested in Test 1, 2, or 3), the evaluator shall ensure that the TOE is able to initiate communications with a trusted IT product using the protocol, setting up the connection as described in the operational guidance and ensuring that the communication is successful.

2.2 Evaluation Activities for Strictly Optional SFRS

2.2.1 Class: Identification and Authentication (FIA)

FIA_UAU_EXT.4 Secondary User Authentication

FIA_UAU_EXT.4
There are no TSS evaluation activities for this element.

There are no guidance evaluation activities for this element.

The Evaluation Activities for any selected requirements related to device authentication must be separately performed for the secondary authentication mechanism (in addition to activities performed for the primary authentication mechanism). The requirements are: FIA_UAU.6, FIA_PMG_EXT.1, FIA_TRT_EXT.1, FIA_UAU.7, FIA_UAU_EXT.2, FTA_SSL_EXT.1, FCS_STG_EXT.2, FMT_SMF_EXT.1/FMT_MOF_EXT.1 #1, #2, #8, #21, and #36.

Additionally, FIA_AFL_EXT.1 must be met, except that in FIA_AFL_EXT.1.2 the separate test is performed with the text "wipe of all protected data" changed to "wipe of all Enterprise application data and all Enterprise shared resource data."
TSS
The evaluator shall verify that the TSS section of the ST describes the process for decrypting Enterprise application data and shared resource data. The evaluator shall ensure that this process requires the user to enter an Authentication Factor and, in accordance with FCS_CKM_EXT.3, derives a KEK which is used to protect the software-based secure key storage and (optionally) DEK(s) for sensitive data, in accordance with FCS_STG_EXT.2.

Guidance
There are no guidance evaluation activities for this element.

Tests
There are no test evaluation activities for this element.

2.3 Evaluation Activities for Objective SFRS

2.3.1 Class: Security Audit (FAU)

FAU_SAR.1 Audit Review

FAU_SAR.1
TSS
There are no TSS evaluation activities for this component.

Guidance
There are no guidance evaluation activities for this component.

Tests
The evaluation activity for this requirement is performed in conjunction with test for function 32 of FMT_SMF_EXT.1.

FAU_SEL.1 Selective Audit

FAU_SEL.1
TSS
There are no TSS evaluation activities for this component.

Guidance
The evaluator shall review the administrative guidance to ensure that the guidance itemizes all event types, as well as describes all attributes that are to be selectable in accordance with the requirement, to include those attributes listed in the assignment. The administrative guidance shall also contain instructions on how to set the pre-selection as well as explain the syntax (if present) for multi-value pre-selection. The administrative guidance shall also identify those audit records that are always recorded, regardless of the selection criteria currently being enforced.

Tests
The evaluator shall also perform the following tests:

- Test FAU_SEL.1:1:For each attribute listed in the requirement, the evaluator shall devise a test to show that selecting the attribute causes only audit events with that attribute (or those that are always recorded, as identified in the administrative guidance) to be recorded.
- Test FAU_SEL.1:2:[conditional] If the TSF supports specification of more complex audit pre-selection criteria (e.g., multiple attributes, logical expressions using attributes) then the evaluator shall devise tests showing that this capability is correctly implemented. The evaluator shall also, in the test plan, provide a short narrative justifying the set of tests as representative and sufficient to exercise the capability.

2.3.2 Class: Cryptographic Support (FCS)

FCS_RBG_EXT.2 Random Bit Generator State Preservation

FCS_RBG_EXT.2
TSS
The evaluation activity for this requirement is captured in the RBG documentation for D - . The evaluator shall verify that the documentation describes how the state is generated so as to be available for the next startup, how the state is used as input to the DRBG, and any protection measures used for the state while the TOE is powered off.

Guidance
There are no guidance evaluation activities for this component.

Tests
There are no test evaluation activities for this component.

FCS_RBG_EXT.3 Support for Personalization String

FCS_RBG_EXT.3
The evaluator shall verify that this function is included as an interface to the RBG in the documentation required by D - and that the behavior of the RBG following a call to this interface is described. The evaluator shall also verify that the documentation of the RBG describes the conditions of use and possible values for the Personalization String input to the SP 800-90A specified DRBG.

TSS
There are no TSS evaluation activities for this component.

Guidance
There are no guidance evaluation activities for this component.

Tests

- Test FCS_RBG_EXT.3:1:The evaluator shall write, or the developer shall provide, an application that adds data to the RBG via the Personalization String. The evaluator shall verify that the request succeeds.

FCS_SRV_EXT.2 Cryptographic Algorithm Services

FCS_SRV_EXT.2
The evaluator shall verify that the API documentation for the secure key storage includes the cryptographic operations by the stored keys.

TSS
There are no TSS evaluation activities for this component.

Guidance
There are no guidance evaluation activities for this component.

Tests

The evaluator shall write, or the developer shall provide access to, an application that requests cryptographic operations of stored keys by the TSF. The evaluator shall verify that the results from the operation match the expected results according to the API documentation. The evaluator shall use these APIs to test the functionality of the secure key storage according to the Evaluation Activities in FCS_STG_EXT.1.

2.3.3 Class: User Data Protection (FDP)

FDP_ACF_EXT.3 Security Attribute Based Access Control

FDP_ACF_EXT.3
TSS
There are no TSS evaluation activities for this component.

Guidance
There are no guidance evaluation activities for this component.

Tests
Evaluation Activity Note: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on consumer Mobile Device products.

- Test FDP_ACF_EXT.3:1:The evaluator shall write, or the developer shall provide, an application that attempts to store a file with both write and execute permissions. If the selection is "no exceptions", then the evaluator shall verify that this action fails and that the permissions on the file are not simultaneously write and execute. If the selection is "application's private data folder", then the evaluator shall ensure that the attempt to store the file is outside of the application's private data folder.
- Test FDP_ACF_EXT.3:2:The evaluator shall traverse the file system examining the permission on each TSF file to verify that no file has both write and execute permissions set. If the selection is "application's private data folder", then only files outside of this folder need to be examined by the evaluator for this test.

FDP_BCK_EXT.1 Application Backup

FDP_BCK_EXT.1
TSS
There are no TSS evaluation activities for this component.

Guidance
There are no guidance evaluation activities for this component.

Tests
If "all application data" is selected, the evaluator shall install an application that has marked all of its application data to be excluded from backups. The evaluator shall cause data to be placed into the application’s storage area. The evaluator shall attempt to back up the application data and verify that the backup fails or that the application’s data was not included in the backup.

If "selected application data" is selected, the evaluator shall install an application that has marked selected application data to be excluded from backups. The evaluator shall cause data covered by "selected application data" to be placed into the application’s storage area. The evaluator shall attempt to backup that selected application data and verify that either the backup fails or that the selected data is excluded from the backup.

FDP_BLT_EXT.1 Limitation of Bluetooth Device Access

FDP_BLT_EXT.1
TSS
The evaluator shall ensure that the TSS describes the mechanism used to prevent unrestricted access to paired Bluetooth devices (and/or their communication data) by every application with access to the Bluetooth system service on the TOE. The evaluator shall verify that this method either restricts access to a single application or provides explicit control of the applications that may communicate with the paired Bluetooth device.

Guidance
The evaluator shall verify that the AGD contains the steps to configure which applications are allowed to communicate with a given Bluetooth peripheral.

Tests
The evaluator shall establish a Bluetooth connection with any peripheral. The evaluator shall verify that an application that is allowed to communicate with the Bluetooth peripheral is able to and that an application that is not allowed to communicate with that Bluetooth peripheral is unable to communicate with the peripheral.

2.3.4 Class: Identification and Authentication (FIA)

FIA_BMG_EXT.2 Biometric Enrollment

FIA_BMG_EXT.2
TSS
The evaluator shall verify that the TSS describes how the quality of samples used to create the authentication template at enrollment are verified. As well as the quality standard that the validation method uses to perform the assessment.

Guidance
The evaluator shall verify that the AGD guidance describes how to enroll a user for each biometric modality supported.

Tests
The evaluator shall input biometric samples for enrollment. Upon inputting biometric samples a fixed number of times as specified in the prompts, one or more authentication templates will be generated. The evaluator shall verify that the device only accepts samples of sufficient quality or requests additional samples if the authentication template is not of sufficient quality. For all quality metrics, the evaluator shall ensure that biometric samples achieving a worse quality score than the prescribed threshold are rejected.

FIA_BMG_EXT.3 Biometric Verification

FIA_BMG_EXT.3
TSS
The evaluator shall verify that the TSS describes how the quality of samples used to verify authentication are verified. As well as the quality standard that the validation method uses to perform the assessment. The evaluator shall enroll a user for each biometric modality supported. The evaluator will then input biometric samples for verification and ensure that the device only accepts samples of sufficient quality. The evaluator shall ensure that biometric samples achieving a worse quality score than the prescribed threshold are rejected.

Guidance
There are no guidance evaluation activities for this component.

Tests
There are no test evaluation activities for this component.

FIA_BMG_EXT.4 Biometric Templates

FIA_BMG_EXT.4
TSS
The evaluator shall verify that the TSS describes how the samples used to create the authentication template at enrollment are mutually consistent and how the mutual consistency is validated, both in terms of the method of validation as well as the quality standard that the validation method uses to perform the assessment.

The evaluator shall input biometric samples for enrollment. In doing so, the evaluator shall verify the enrollment templates generated are of sufficient quality. Upon inputting biometric samples a fixed number of times as specified in the prompts, the evaluator shall additionally verify that any enrollment and authentication templates generated are of sufficient quality. That is, they shall all be mutually consistent and correspond to the biometric characteristics of a single user and source (e.g. the same finger from the same person).

Guidance
There are no guidance evaluation activities for this component.

Tests
There are no test evaluation activities for this component.

FIA_BMG_EXT.5 Handling Unusual Biometric Templates

FIA_BMG_EXT.5
TSS
The evaluator shall verify that the TSS how the matching algorithm addresses properly formatted templates with unusual data properties, incorrect syntax, or low quality. The evaluator shall ensure that these claims are sound through appropriate testing based on test programs provided by the vendor.

Guidance
There are no guidance evaluation activities for this component.

Tests
There are no test evaluation activities for this component.

FIA_BMG_EXT.6 Spoof Detections for Biometrics

FIA_BMG_EXT.6
TSS
The testing methodology specified in ISO 19989 Information technology — Security evaluation of presentation attack detection for biometrics [ISO 19989] is to be used to determine the efficacy of the PAD for the selected attack potential.

Guidance
There are no guidance evaluation activities for this component.

Tests
Evaluation Activity Note: ISO 19989 is in draft status at the time of publication of this PP. Once the ISO standard is published, it shall be used to meet the evaluation activity for this requirement. Henniger, Scheuermann, and Kniess [IBPC], provide a description of attack potential calculation with examples. Until such time as ISO 19989 is published, the vendor shall provide to the lab a description of the PAD processing implemented in the TSF, test procedures used to validate successful operation of PAD, and test data with results of the PAD validation testing. The lab may analyze the test procedures and data to validate vendor test results or, optionally, may conduct its own testing.

If the lab performs its own testing, it is highly recommended that the vendor provides spoof testing tools, as it is not expected for the lab to create a test procedure for modalities outside of established standards and easily implemented procedures. Labs can also expedite the testing process by purchasing the appropriate spoof kits and recipes from specialized biometrics testing labs.

FIA_X509_EXT.4 X.509 Certificate Enrollment

FIA_X509_EXT.4
TSS
There are no TSS evaluation activities for this component.

Guidance
The evaluator shall check to ensure that the operational guidance contains instructions on requesting certificates from an EST server, including generating a Certificate Request Message.

Tests
The evaluator shall also perform the following tests. Other tests are performed in conjunction with the evaluation activity listed in the Package for Transport Layer Security.

- Test FIA_X509_EXT.4:1:The evaluator shall use the operational guidance to cause the TOE to request certificate enrollment from an EST server using the simple enrollment method described in RFC 7030 Section 4.2, authenticating the certificate request to the server using an existing certificate and private key as described by RFC 7030 Section 3.3.2. The evaluator shall confirm that the resulting certificate is successfully obtained and installed in the TOE key store.
- Test FIA_X509_EXT.4:2:The evaluator shall use the operational guidance to cause the TOE to request certificate enrollment from an EST server using the simple enrollment method described in RFC 7030 Section 4.2, authenticating the certificate request to the server using a username and password as described by RFC 7030 Section 3.2.3. The evaluator shall confirm that the resulting certificate is successfully obtained and installed in the TOE key store.
- Test FIA_X509_EXT.4:3:The evaluator shall modify the EST server to return a certificate containing a different public key than the key included in the TOE’s certificate request. The evaluator shall use the operational guidance to cause the TOE to request certificate enrollment from an EST server. The evaluator shall confirm that the TOE does not accept the resulting certificate since the public key in the issued certificate does not match the public key in the certificate request.
- Test FIA_X509_EXT.4:4:The evaluator shall configure the EST server or use a man-in-the-middle tool to present a server certificate to the TOE that is present in the TOE general Trust Anchor Database but not its EST-specific Trust Anchor Database. The evaluator shall cause the TOE to request certificate enrollment from the EST server. The evaluator shall verify that the request is not successful.
- Test FIA_X509_EXT.4:5:The evaluator shall configure the EST server or use a man-in-the-middle tool to present an invalid certificate. The evaluator shall cause the TOE to request certificate enrollment from the EST server. The evaluator shall verify that the request is not successful The evaluator shall configure the EST server or use a man-in-the-middle tool to present a certificate that does not have the CMC RA purpose and verify that requests to the EST server fail. The tester shall repeat the test using a valid certificate and a certificate that contains the CMC RA purpose and verify that the certificate enrollment requests succeed.
- Test FIA_X509_EXT.4:6:The evaluator shall use a packet sniffing tool between the TOE and an EST server. The evaluator shall turn on the sniffing tool and cause the TOE to request certificate enrollment from an EST server. The evaluator shall verify that the EST protocol interaction occurs over a Transport Layer Security (TLS) protected connection. The evaluator is not expected to decrypt the connection but rather observe that the packets conform to the TLS protocol format.
- Test FIA_X509_EXT.4:7:The evaluator shall use the operational guidance to cause the TOE to request a server-provided private key and certificate from an EST server. The evaluator shall confirm that the resulting private key and certificate are successfully obtained and installed in the TOE key store.
- Test FIA_X509_EXT.4:8:The evaluator shall modify the EST server to, in response to a server-provided private key and certificate request, return a private key that does not correspond with the public key in the returned certificate. The evaluator shall use the operational guidance to cause the TOE to request a

server-provided private key and certificate. The evaluator shall confirm that the TOE does not accept the resulting private key and certificate since the private key and public key do not correspond.
- Test FIA_X509_EXT.4:9:The evaluator shall configure the EST server to provide a "Root CA Key Update" as described in RFC 7030 Section 4.1.3. The evaluator shall cause the TOE to request CA certificates from the EST server and shall confirm that the EST-specific Trust Anchor Database is updated with the new trust anchor.
- Test FIA_X509_EXT.4:10:The evaluator shall configure the EST server to provide a "Root CA Key Update" as described in RFC 7030 Section 4.1.3, but shall modify part of the NewWithOld certificate’s generated signature. The evaluator shall cause the TOE to request CA certificates from the EST server and shall confirm that the EST-specific Trust Anchor Database is not updated with the new trust anchor since the signature did not verify.
- Test FIA_X509_EXT.4:11:The evaluator shall use the operational guidance to cause the TOE to generate a certificate request message. The evaluator shall capture the generated message and ensure that it conforms to the format specified by RFC 2986. The evaluator shall confirm that the certificate request provides the public key and other required information, including any necessary user-input information.

FIA_X509_EXT.5 X.509 Certificate Requests

FIA_X509_EXT.5
TSS
If the ST author selects "device-specific information", the evaluator shall verify that the TSS contains a description of the device-specific fields used in certificate requests.

Guidance
The evaluator shall check to ensure that the operational guidance contains instructions on generating a Certificate Request Message. If the ST author selects "Common Name", "Organization", "Organizational Unit", or "Country", the evaluator shall ensure that this guidance includes instructions for establishing these fields before creating the certificate request message.

Tests
The evaluator shall also perform the following tests:

- Test FIA_X509_EXT.5:1:The evaluator shall use the operational guidance to cause the TOE to generate a certificate request message. The evaluator shall capture the generated message and ensure that it conforms to the format specified. The evaluator shall confirm that the certificate request provides the public key and other required information, including any necessary user-input information.
- Test FIA_X509_EXT.5:2:The evaluator shall demonstrate that validating a certificate response message without a valid certification path results in the function failing. The evaluator shall then load a certificate or certificates as trusted CAs needed to validate the certificate response message, and demonstrate that the function succeeds. The evaluator shall then delete one of the certificates, and show that the function fails.

2.3.5 Class: Security Management (FMT)

FMT_SMF_EXT.3 Current Administrator

FMT_SMF_EXT.3
TSS
There are no TSS evaluation activities for this component.

Guidance
There are no guidance evaluation activities for this component.

Tests
The evaluator shall cause the TOE to be enrolled into management. The evaluator shall then invoke this mechanism and verify the ability to view that the device has been enrolled, and view the management functions that the administrator is authorized to perform.

2.3.6 Class: Protection of the TSF (FPT)

FPT_AEX_EXT.5 Kernel Address Space Layout Randomization

FPT_AEX_EXT.5
TSS
The evaluator shall ensure that the TSS section of the ST describes how the bits are generated and provides a justification as to why those bits are unpredictable.

Guidance
There are no guidance evaluation activities for this component.

Tests
Evaluation Activity Note: The following test require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on consumer Mobile Device products.

- Test FPT_AEX_EXT.5:1:The evaluator shall reboot the TOE six times. For each of these reboots, the evaluator shall examine memory mapping locations of the kernel. The evaluator must ensure that for at least five reboots the memory mappings are not placed in the same location on both devices.

FPT_AEX_EXT.6 Write or Execute Memory Page Permissions

FPT_AEX_EXT.6
TSS
The evaluator shall ensure that the TSS describes how the operating system of the application processor prevents all processes executing in a non-privileged execution domain from achieving write and execute permissions on any page of memory (with only specified exceptions). The evaluator shall ensure that the TSS describes how such processes are unable to request pages of memory with such permissions, and how they are unable to change permissions to both write and execute on any pages already allocated to them.

Guidance
There are no guidance evaluation activities for this component.

Tests
There are no test evaluation activities for this component.

FPT_AEX_EXT.7 Heap Overflow Protection

FPT_AEX_EXT.7
TSS
The evaluator shall verify that the TSS enumerates the heap implementations provided to userspace processes. The evaluator shall ensure that the TSS lists all types of heap metadata and identifies how the integrity of each type of metadata is ensured. The evaluator shall ensure that the TSS identifies all fields within each type of metadata and identifies how the integrity of these fields is ensured. The evaluator shall verify that the TSS identifies the manner in which an error condition is entered when a heap overflow is detected and the resulting actions taken by the TSF.

Guidance
There are no guidance evaluation activities for this component.

Tests
For each heap implementation, the evaluator shall write, or the developer shall provide access to, an application, which allocates memory from the heap and then writes arbitrary data significantly beyond the end of the allocated buffer. The evaluator shall attempt to execute this application and verify that the write is not allowed.

FPT_BBD_EXT.1 Application Processor Mediation

FPT_BBD_EXT.1
TSS
The evaluator shall ensure that the TSS section of the ST describes at a high level how the processors on the Mobile Device interact, including which bus protocols they use to communicate, any other devices operating on that bus (peripherals and sensors), and identification of any shared resources. The evaluator shall verify that the design described in the TSS does not permit any BPs from accessing any of the peripherals and sensors or from accessing main memory (volatile and non-volatile) used by the AP. In particular, the evaluator shall ensure that the design prevents modification of executable memory of the AP by the BP.

Guidance
There are no guidance evaluation activities for this component.

Tests
There are no test evaluation activities for this component.

FPT_BLT_EXT.1 Limitation of Bluetooth Profile Support

FPT_BLT_EXT.1
TSS
The evaluator shall ensure that the TSS lists all Bluetooth profiles that are disabled while not in use by an application and which need explicit user action in order to become enabled.

Guidance
There are no guidance evaluation activities for this component.

Tests
The evaluator shall perform the following tests:

- Test FPT_BLT_EXT.1:1:While the service is not in active use by an application on the TOE, the evaluator shall attempt to discover a service associated with a "protected" Bluetooth profile (as specified by the requirement) on the TOE via a Service Discovery Protocol search. The evaluator shall verify that the

service does not appear in the Service Discovery Protocol search results. Next, the evaluator shall attempt to gain remote access to the service from a device that does not currently have a trusted device relationship with the TOE. The evaluator shall verify that this attempt fails due to the unavailability of the service and profile.
- Test FPT_BLT_EXT.1:2:The evaluator shall repeat Test 1 with a device that currently has a trusted device relationship with the TOE and verify that the same behavior is exhibited.

FPT_NOT_EXT.2 Self-Test Notification

FPT_NOT_EXT.2
The evaluator shall verify that the TSS describes which critical memory is measured for these integrity values and how the measurement is performed (including which TOE software performs these generates these values, how that software accesses the critical memory, and which algorithms are used).

If the integrity values are provided to the administrator, the evaluator shall verify that the AGD guidance contains instructions for retrieving these values and information for interpreting them. For example, if multiple measurements are taken, what those measurements are and how changes to those values relate to changes in the device state.

Evaluation Activity Note: The following test may require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on consumer Mobile Device products.

The evaluator shall repeat the following test for each measurement:

- Test FPT_NOT_EXT.2:1:The evaluator shall boot the device in an approved state and record the measurement taken (either from the log or by using the administrative guidance to retrieve the value via an MDM Agent). The evaluator shall modify the critical memory or value that is measured. The evaluator shall boot the device and verify that the measurement changed.

TSS
The evaluator shall verify that the TSS describes which key the TSF uses to sign the responses to queries and the certificate used to prove ownership of the key, and the method of associating the certificate with a particular device manufacturer and model.

Guidance
There are no guidance evaluation activities for this element.

Tests
The evaluator shall perform the following test:

- Test FPT_NOT_EXT.2:1:The evaluator shall write, or the developer shall provide, a management application that queries either the audit logs or the measurements. The evaluator shall verify that the responses to these queries are signed and verify the signatures against the TOE's certificate.

FPT_TST_EXT.2/POSTKERNEL TSF Integrity Checking (Post-Kernel)

FPT_TST_EXT.2/POSTKERNEL
TSS
There are no TSS evaluation activities for this component.

Guidance
There are no guidance evaluation activities for this component.

Tests
The evaluation activity shall be completed in conjunction with FPT_TST_EXT.2/PREKERNEL for all executable code specified.

FPT_TUD_EXT.5 Application Verification

FPT_TUD_EXT.5
TSS
The evaluator shall verify that the TSS describes how mobile application software is verified at installation. The evaluator shall ensure that this method uses a digital signature by a code signing certificate.

Guidance
There are no guidance evaluation activities for this component.

Tests

- Test FPT_TUD_EXT.5:1:The evaluator shall write, or the developer shall provide access to, an application. The evaluator shall try to install this application without a digitally signature and shall verify that installation fails. The evaluator shall attempt to install an application digitally signed with an appropriate certificate, and verify that installation succeeds.
- Test FPT_TUD_EXT.5:2:The evaluator shall digitally sign the application with an invalid certificate and

verify that application installation fails. The evaluator shall digitally sign the application with a certificate that does not have the Code Signing purpose and verify that application installation fails. This test may be performed in conjunction with the Evaluation Activities for FIA_X509_EXT.1.
- Test FPT_TUD_EXT.5:3:If necessary, the evaluator shall configure the device to limit the public keys that can sign application software according to the AGD guidance. The evaluator shall digitally sign the application with a certificate disallowed by the device or configuration and verify that application installation fails. The evaluator shall attempt to install an application digitally signed with an authorized certificate and verify that application installation succeeds.

FPT_TUD_EXT.6 Trusted Update Verification

FPT_TUD_EXT.6
TSS
The evaluator shall verify that the TSS describes the mechanism that prevents the TSF from installing software updates that are an older version that the currently installed version.

Guidance
There are no guidance evaluation activities for this component.

Tests
The evaluator shall repeat the following tests to cover all allowed software update mechanisms as described in the TSS. For example, if the update mechanism replaces an entire partition containing many separate code files, the evaluator does not need to repeat the test for each individual file.

- Test FPT_TUD_EXT.6:1:The evaluator shall attempt to install an earlier version of software (as determined by the manufacturer). The evaluator shall verify that this attempt fails by checking the version identifiers or cryptographic hashes of the privileged software against those previously recorded and checking that the values have not changed.
- Test FPT_TUD_EXT.6:2:The evaluator shall attempt to install a current or later version and shall verify that the update succeeds.

2.3.7 Class: TOE Access (FTA)

FTA_TAB.1 Default TOE Access Banners

FTA_TAB.1
TSS
The TSS shall describe when the banner is displayed.

Guidance
There are no guidance evaluation activities for this component.

Tests
The evaluator shall also perform the following test:

- Test FTA_TAB.1:1:The evaluator follows the operational guidance to configure a notice and consent warning message. The evaluator shall then start up or unlock the TSF. The evaluator shall verify that the notice and consent warning message is displayed in each instance described in the TSS.

2.4 Evaluation Activities for Implementation-based SFRS

2.4.1 Class: User Data Protection (FDP)

FDP_UPC_EXT.1/BLUETOOTH Inter-TSF User Data Transfer Protection (Bluetooth)

FDP_UPC_EXT.1/BLUETOOTH
The evaluator shall verify that the API documentation provided according to 5.2.2 includes the security functions (protection channel) described in these requirements, and verify that the APIs implemented to support this requirement include the appropriate settings/parameters so that the application can both provide and obtain the information needed to assure mutual identification of the endpoints of the communication as required by this component.

TSS
The evaluator shall examine the TSS to determine that it describes that all protocols listed in the TSS are specified and included in the requirements in the ST.

Guidance
The evaluator shall confirm that the operational guidance contains instructions necessary for configuring the protocol(s) selected for use by the applications.

Tests
Evaluation Activity Note: The following test requires the developer to provide access to a test platform that

provides the evaluator with tools that are typically not found on consumer Mobile Device products.

The evaluator shall write, or the developer shall provide access to, an application that requests protected channel services by the TSF. The evaluator shall verify that the results from the protected channel match the expected results according to the API documentation. This application may be used to assist in verifying the protected channel Evaluation Activities for the protocol requirements. The evaluator shall also perform the following tests:

- Test FDP_UPC_EXT.1/BLUETOOTH:1:The evaluators shall ensure that the application is able to initiate communications with an external IT entity using each protocol specified in the requirement, setting up the connections as described in the operational guidance and ensuring that communication is successful.
- Test FDP_UPC_EXT.1/BLUETOOTH:2:The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data are not sent in plaintext.

2.5 Evaluation Activities for Selection-based SFRS

2.5.1 Class: User Data Protection (FDP)

FDP_UPC_EXT.1/BLUETOOTH Inter-TSF User Data Transfer Protection (Bluetooth)

FDP_UPC_EXT.1/BLUETOOTH
The evaluator shall verify that the API documentation provided according to 5.2.2 includes the security functions (protection channel) described in these requirements, and verify that the APIs implemented to support this requirement include the appropriate settings/parameters so that the application can both provide and obtain the information needed to assure mutual identification of the endpoints of the communication as required by this component.

TSS
The evaluator shall examine the TSS to determine that it describes that all protocols listed in the TSS are specified and included in the requirements in the ST.

Guidance
The evaluator shall confirm that the operational guidance contains instructions necessary for configuring the protocol(s) selected for use by the applications.

Tests
Evaluation Activity Note: The following test requires the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on consumer Mobile Device products.

The evaluator shall write, or the developer shall provide access to, an application that requests protected channel services by the TSF. The evaluator shall verify that the results from the protected channel match the expected results according to the API documentation. This application may be used to assist in verifying the protected channel Evaluation Activities for the protocol requirements. The evaluator shall also perform the following tests:

- Test FDP_UPC_EXT.1/BLUETOOTH:1:The evaluators shall ensure that the application is able to initiate communications with an external IT entity using each protocol specified in the requirement, setting up the connections as described in the operational guidance and ensuring that communication is successful.
- Test FDP_UPC_EXT.1/BLUETOOTH:2:The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data are not sent in plaintext.

3 Evaluation Activities for SARs

The sections below specify EAs for the Security Assurance Requirements (SARs) included in the related cPPs (see section 1.1 above). The EAs in Section 2 (Evaluation Activities for SFRs), Section 3 (Evaluation Activities for Optional Requirements), and Section 4 (Evaluation Activities for Selection-Based Requirements) are an interpretation of the more general CEM assurance requirements as they apply to the specific technology area of the TOE.

In this section, each SAR that is contained in the cPP is listed, and the EAs that are not associated with an SFR are captured here, or a reference is made to the CEM, and the evaluator is expected to perform the CEM work units.

4 Required Supplementary Information

This Supporting Document has no required supplementary information beyond the ST, operational guidance, and testing.

Appendix A - Bibliography

Identifier	Title
[ANSI 409.1]	ANSI/CITS 409.1-2005. Biometrics Performance Testing and Reporting—Part 1: Principles and Findings." Annex B. ANSI/CITS, 2005.
[NFIQ 2.0]	Biometric Quality: The push towards zero error biometrics., Tabassi, Elham et al. International Biometrics Performance Conference (IBPC), 2016. Retrieved May 30, 2016.
[CC]	Common Criteria for Information Technology Security Evaluation - • Part 1: Introduction and General Model, CCMB-2017-04-001, Version 3.1 Revision 5, April 2017. • Part 2: Security Functional Components, CCMB-2017-04-002, Version 3.1 Revision 5, April 2017. • Part 3: Security Assurance Components, CCMB-2017-04-003, Version 3.1 Revision 5, April 2017.
[CEM]	Common Evaluation Methodology for Information Technology Security - Evaluation Methodology, CCMB-2012-09-004, Version 3.1, Revision 5, April 2017.
[ISO 19989]	ISO/IEC NP 19989: Evaluation of presentation attack detection for biometrics International Organization for Standardization (ISO), 2014.
[BROWN]	Interval Estimation for a Binomial Proportion.Brown, Cai, and DasGupta.
[NFIQ 1.0]	NIST Fingerprint Image Quality and relation to PIV, Tabassi, Elham. NIST Information Technology Laboratory, 2005. Retrieved June 13, 2015.
[IBPC]	On security evaluation of fingerprint recognition systems-- IBPC Presentation., Henniger, Scheuermann, and Kniess.International Biometric Performance Testing Conference (IBPC), 2010. Retrieved June 12, 2015.
[NIST]	The NIST speaker recognition evaluation—Overview, methodology, systems, results, perspective, Doddington, Przybocki, Martin, and Reynolds. Speech Communication 31: Elsevier, 2000, Retrieved June 10, 2015.