Security Requirements for Mobile Operating Systems

Information Assurance Directorate

25 January 2013

Version 1.0

Table of Contents

List of Tables

List of Figures

Revision History

Version	Date	Description
1.0	*22 January 2013*	Initial release

1 INTRODUCTION

1 This Mobile Operating System (OS) Protection Profile (PP) presents a minimal baseline set of requirements that target well defined and described threats. It represents an evolution of “traditional” Protection Profiles and the associated evaluation of the defined requirements. This introduction will describe the features of a compliant Target of Evaluation (TOE), and will also discuss the evolutionary aspects of the PP as a guide to readers of the document.

1.1 First Generation Protection Profiles

2 What makes security for mobility different than other technologies? Regardless of the actual technical security features of individual devices, there is implied security if the physical environment where the device resides is protected by guards, dogs and fences with a wired computing or communications device. For mobility, these traditional physical protections are irrelevant. Not only are the wireless communication channels more readily available to adversaries, but the devices themselves are also expected to be multipurpose and used for both work and personal use. Mobility clearly brings new security challenges.

3 To keep up with rapidly-evolving mobility market place, the Information Assurance Directorate (IAD) intends to manage the risks of missing or imperfectly implemented mobility security features by issuing first generation Mobility PPs. IAD’s near term objective is to create and disseminate first generation Mobility PPs as a means to communicate the initial set of critical security functionality to the vendor community, which will in turn provide the pool of commercial products implementing the security features IAD requires. The first generation Mobility PPs consist of the Mobile OS PP (this document); SIP Server PP, and the Mobility App (VoIP) PP. The goal of these PPs is to present plainly what is possible today (and required today) so that a clear direction is given for security critical components to improve enterprise security that includes mobile devices.

4 Monitoring/Auditing capabilities, virtualization and other potentially desirable features for the Mobile OS will not be addressed in this version of the OS PP and are being considered for incorporation into future PP versions. IAD will work with vendors to determine how and when to obtain products with these features, and whether /when to create the corresponding PPs.

5 The set of requirements in this PP is intentionally limited in scope in order to promote quicker, less costly evaluations. Security Targets (STs) that include additional functionality (and requirements) will not be accepted into evaluation.

1.2 Compliant Targets of Evaluation

6 The Mobile OS in the context of this PP is part of the cell phone workspace that the enterprise can administer and control. The OS is principally similar to a general-purpose OS found on a desktop or laptop however; its features are simpler and include controls for third-party applications, limited processing resources, limited storage capacity (i.e. no hard discs, no CD-ROM or DVD reader, etc.), and mobile-related functionality. A Target of Evaluation (TOE) that is defined to be a mobile operating system for smartphones should claim compliance to this PP. Other mobile operating systems that come with personal digital assistants (PDAs), tablet computers, and other information appliances are not suitable to claim conformance to this PP.

7 In the near term, the focus of this PP is using a smartphone over a 3G/4G network for secure end-to-end voice communications, and for secure data communications with the enterprise. This means the only

communication path will be either with an enterprise's VPN Gateway or emergency 911 calls. Wi-Fi and Bluetooth communications are assumed to be configured off/disabled. The emphasis is on secure communications with limited protected data storage concerns. Future PPs will include additional requirements for data encryption. If the TOE provides data encryption relevant to requirements defined in Annex C, then applicable SFRs from Annex C should be included.

8 TOE functionality is detailed in later sections, but it is useful to give a brief description of the general mobile network architecture here. Compliant TOEs will provide security functionality that addresses threats to the TOE. Compliant TOEs must protect communications to and from the enterprise by use of an enterprise Virtual Private Network (VPN). The TOE will drop IP data originating outside the VPN, except data necessary to establish and maintain the VPN. All data between the TOE and the enterprise is protected in an IPSec VPN tunnel. The IPSec VPN connection must be established before connections to enterprise services are permitted. A VPN client that cannot be successfully identified or authenticated is denied access to the enterprise infrastructure and services. The protocols required by this PP make use of certificates and the TOE must securely store certificates and private keys. The TOE must protect the workspace from being exposed to the carrier infrastructure. The TOE must also provide the ability to verify the source of updates to the TOE. Figure 1 shows the TOE in relation to other mobility system components and the environment.

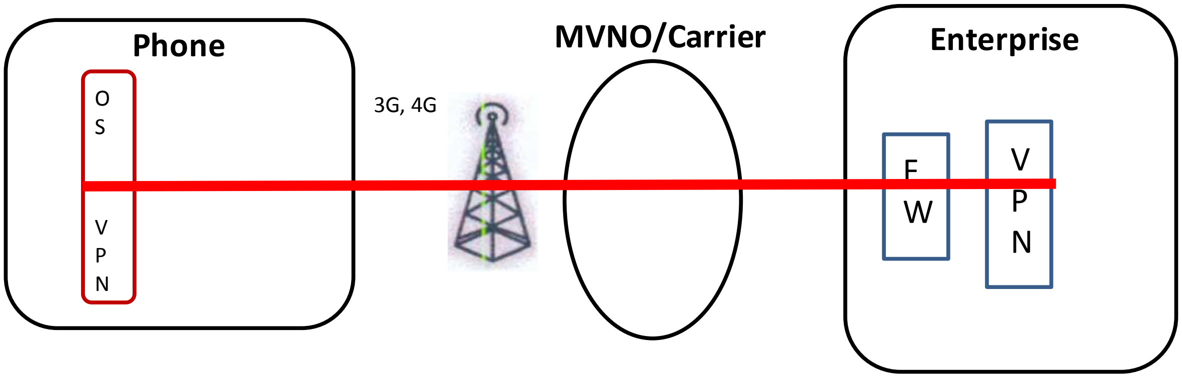

Figure 1 Mobile System Components

2 SECURITY PROBLEM DESCRIPTION

9 As detailed in the previous section, the security problem to be addressed by compliant TOEs is described by threats that are common to a mobility OS, as opposed to those that might be targeted at the specific functionality of a specific type of mobility OS. The Common Criteria defines a threat as "an adverse action performed by a threat agent on an asset." In this version of the PP the primary asset being protected in local storage is user credentials (passwords) and key material used for authentication and update validation. In addition, user credentials and certificates must be protected from malware and other malicious attacks since this data provides the attacker with access to the enterprise network. Annex A: Supporting Tables presents the Security Problem Description (SPD) in a more "traditional" form. The following sections detail the problems that compliant TOEs will address; references to the "traditional" statements in Annex A are included.

2.1 Unattended, Misplaced, or Stolen Device

10 Having a device that is portable and small like a smartphone can lead to the phone being lost or stolen very easily. This puts all of the keys and credentials that are needed to access the enterprise VPN at risk for extraction by a malicious user. A misplaced phone may also be left temporarily unlocked. If data such as email, contacts, and user credentials is left unencrypted, then all enterprise data may be accessed by malicious users. A VPN connection that is left running also allows an adversary to access the enterprise network without needing credentials. However, an adversary making physical changes to the actual smartphone, such as modifying the SIM card, is not a threat addressed by this PP.

[T. UNAUTHORIZED_ACCESS]

2.2 Web-based and network-based attacks

11 Unfortunately malware can easily be installed onto a mobile device. Once installed, malicious software may be used to extract the user's web surfing history, login credentials including passwords, and other potentially harmful information such as credit card numbers.[1]

12 Devices that are corrupted by malware may also have their data at risk. Any information that is saved on the phone has a potential for a data integrity threat. The data on the phone may be altered by malware and become unrecoverable. Additionally, the data can be extracted by the originator of the malware.

[T.INSTALL_MALICIOUS_SOFTWARE]

2.3 User Data[2] Sent to Unintended Destination

13 Data traversing the TOE could inadvertently be disclosed to malicious users; since this data may be sensitive, this may cause a compromise that is unacceptable. The specific threat that must be addressed concerns data that is retained in memory by the TOE in the course of processing network traffic. If the TOE retains any information between sessions with different end points the information from the first connection could be inadvertently released to a second user during a subsequent connection. Since the TOE operates on a thin client, residual data that is saved to local memory such as credentials, add an unnecessary risk to enterprise level information being exposed to malicious actions.

[T.USER_DATA_REUSE]

2.4 Unauthorized Updates

14 A common method of attack is exploiting unpatched versions of software. A malformed update prepared by a malicious party may be applied that compromises the enterprise network. Any

[1] This description of threat was taken from the Symantec paper – "A Window into Mobile Device Security".

[2] This PP is for enterprise use only phones (not BYOD – bring your own device). Therefore, user data is considered to be enterprise user data, not personal user data.

credentials or saved certificates on the device may then be used to fake the identity of the device's owner and gain access to the enterprise.

15 Especially critical to maintaining the security controls of the system are the components that make up the operating system and the controls that provide separation. Updates to these components must come from a trusted source.

[T. UNAUTHORIZED_UPDATE]

2.5 Insecure Workspace

16 The workspace of a smartphone contains the mobile OS, infrastructure, and the enterprise data. A risk exists if enterprise and personal cell phone data is not separated. The enterprise has no way of controlling what a user does with enterprise level information if there are no restrictions to where and how the data is stored.

17 The risk exists that a malicious hardware or applications that do not have proper permissions and separation from enterprise data may exploit other applications on the device that do have access to the enterprise data. For example, an application with all permissions on a device may gain access to the email or calendar applications, which can contain locally saved enterprise data. This newly acquired enterprise data can be sent back to the owner of the malicious application without the user's knowledge.

[T. INSECURE_WORKSPACE]

2.6 TSF Failure

18 Security mechanisms of the TOE generally build up from a primitive set of mechanisms (e.g., memory management, privileged modes of process execution) to more complex sets of mechanisms. Failure of the primitive mechanisms could lead to a compromise in more complex mechanisms, resulting in a compromise of the TOE Security Functionality (TSF).

[T. TSF_FAILURE]

3 SECURITY OBJECTIVES

19 Compliant TOEs will provide security functionality that address threats to the TOE and implement policies that are imposed by law or regulation. The following sections provide a description of this functionality in light of the threats previously discussed that motivate its inclusion in compliant TOEs. The security functionality provided includes protected communications to and between elements of the TOE; administrative access to the TOE and its configuration capabilities; system monitoring for detection of security relevant events; and the ability to verify the source of updates to the TOE.

3.1 Verifiable Updates

20 As described in section 2.4, "Unauthorized Updates", unsecured device drivers and unpatched exploits can be used to attack a mobile device. To ensure that an update or new device driver comes from a trusted a source, a VPN must be used. This ensures that any update the MDM applies is coming from a trusted network and is approved by the enterprise. This includes not only updates to the TOE itself, but any applications running on the TOE.

21 Section 2.5 delineates the threat of malicious applications, which is mitigated by verifying the signatures of signed applications at installation. The same mechanisms will be used to verify the updates prior to installing them: only updates signed by an authorized source will be accepted and allowed to be installed.

(FPT_TUD_EXT.1(1), FPT_TUD_EXT.1(2), FPT_TST_EXT.1, FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_IPSEC_EXT.1)

3.2 Secure Storage

22 In Section 2.1, the threat of a lost or stolen device is described. To protect the data on a phone that is misplaced or stolen and authentication data such as passwords and private keys stored on the phone, this data must be encrypted using an approved encryption algorithm with a key that is generated using a random bit generator (RBG) that provides sufficient entropy. None of the keys on the phone should be stored in plain text. The key encryption key (KEK) is hashed with a password to encrypt/decrypt the credentials (e.g., private key stored in the X.509 certificate) that are stored in the TOE.

23 A phone that is left on and unattended by a user may have access to the enterprise and thus the data stored there. Preventing this can be done using a lock screen that is brought up after a configurable idle time or intentionally locked by the user. The user will be required to re-authenticate prior to regaining access to the enterprise services (with the exception of completing 911 calls).

(FCS_CKM.1, FCS_CKM_EXT.2, FCS_CKM_EXT.4, FCS_COP.1(1), FCS_COP.1(3), FCS_RBG_EXT.1, FDP_ACC.1, FDP_ACF.1, FPT_KST_EXT.1, FTA_SSL_EXT.1, FTA_SSL_EXT.2)

3.3 Secure Workspace

24 Section 2.5 discussed the threats of an insecure workspace. These include not knowing what a user does with enterprise data, a backdoor in the SIM card, and applications having unwanted permissions. Ideally, the TOE shall be separated from the hardware on the mobile device such that a secure workspace is maintained. In the future, virtualization for smartphones may be employed to isolate/separate the device workspace.

25 An administrator must have control over what networks users can connect to on the mobile device. This prevents exposing enterprise data to other networks since a user will only be able to access the enterprise network. Access to wifi networks are not allowed such that the TOE is configured to have wifi access turned off.

26 All communication originating from or accepted by the TOE must be done over a VPN to prevent exposing sensitive data to the mobile carrier. The VPN client is configured such that the only communication path is via trusted VPN Gateways, which can prevent users from having direct internet access via an unsecure ISP (going to websites that have a malicious agenda as described in threat 2.2).

27 The OS, through the enterprise, can prevent malicious applications on the device from invading enterprise data stored on the device. Only authorized applications[3] that are signed and approved by the enterprise may be installed on the device. This increases the difficulty of applications that have a malicious purpose from ever making it into the secure workspace of a device.

(FCS_CKM.1, FCS_CKM_EXT.4, FCS_COP.1(*),FCS_RBG_EXT.1, FDP_ACC.1, FDP_ACF.1, FTP_ITC.1, FCS_IPSEC_EXT.1, FIA_AFL.1, FIA_UIA_EXT.1, FTA_SSL_EXT.1, FTA_SSL_EXT.2)

3.4 TOE Management

28 In order to provide a trusted means for management of the TOE's functions, users that are authorized to use the device by virtual of having the proper authorization factors are not allowed to configure the TOE, or manage critical configuration data. The management of these critical functions is restricted to an authorized mobile device manager (MDM).

(FMT_MOF.1(1), FMT_MOF.1(2), FMT_SMF.1)

3.5 Residual Information Clearing

29 In order to counter the threat that user data is inadvertently included in network traffic not intended by the original sender, the TSF ensures that network IP packets sent from the TOE do not include data "left over" from the processing of previous information that includes any information between sessions with different end points, residual data that is saved to local memory such as credentials, and etc.

(FDP_RIP.2)

3.6 TSF Self Test

30 In order to detect some number of failures of underlying security mechanisms used by the TSF, the TSF will perform self-tests during the initial start-up (power on). The extent of this self testing is left to the product developer, but a more comprehensive set of self tests should result in a more trustworthy device on which to connect to the enterprise architecture.

(FPT_TST_EXT.1)

[3] This only applies to applications, not scripts as part of applications (i.e. the browser itself must be signed and approved as an application, but scripts running on it will not be signed). However, the user will be restricted to browse only certain pre-selected sites.

4 SECURITY REQUIREMENTS

31 The Security Functional Requirements included in this section are derived from Part 2 of the *Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 3*, with additional extended functional components.

4.1 Conventions

32 The CC defines operations on Security Functional Requirements: assignments, selections, assignments within selections and refinements. This document uses the following font conventions to identify the operations defined by the CC:

- Assignment: Indicated with *italicized* text;
- Refinement made by PP author: Indicated by the word "Refinement" in **bold text** after the element number with additional text in **bold** text and strikethroughs, if necessary;
- Selection: Indicated with underlined text;
- Assignment within a Selection: Indicated with *italicized and underlined* text;
- Iteration: Indicated by appending the iteration number in parenthesis, e.g., (1), (2), (3).

33 Extended SFRs are identified by having a label 'EXT' after the requirement name for TOE SFRs.

4.2 TOE Security Functional Requirements

34 This section identifies the Security Functional Requirements for the TOE.

4.2.1 Cryptographic Support (FCS)

35 The cryptographic requirements make reference to standards describing the algorithms; most of these standards are available from US NIST as Special Publications (800-xxx) or Federal Information Processing Standards (FIPS). The assurance requirements detail how the implementation of these requirements is to be verified. Each scheme has the option of specifying the process under which the cryptographic assurance activities may be considered satisfied. All cryptographic functionality specified below must be implemented within the TOE. Guidance for testing the requirements will depend on selections performed by the ST author. This may require the evaluator to have a trusted reference implementation of the algorithms implemented that can produce test vectors that are verifiable for ATE_IND.1 tests.

36 It is important to note that while the TOE contains the same IPsec requirements and similar X.509 requirements as the VPN Client, and important difference to note is currently the TOE is not required to generate an asymmetric key pair that is used for authentication. Rather, the TOE receives its certificate and key pair from a Certificate Authority.

37

FCS_CKM.1 Cryptographic key generation (KEK)

FCS_CKM.1.1 A password used to generate a submask shall contain at least [assignment: positive integer of 8 or more] characters in the set of {upper case characters, lower case characters, numbers, and the following special characters: "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")", and [assignment: other supported special characters]} and shall be conditioned [selection:

- using [selection: SHA-1, SHA-256, SHA-384, SHA-512];
- using NIST SP 800-132 with a salt generated using a Random Bit Generator as specified in FCS_RBG_EXT.1, an iteration count of [assignment: *number greater than or equal to 1000*], and HMAC using [selection: SHA-1, SHA-256, SHA-512];

such that the output of the conditioning function is greater than or equal to [assignment: cryptographic key size specified in FCS_COP.1(1).

Application Note:

38 *The TOE supports different methods for authorizing the user. Local authorization is the used to "unlock" the smartphone in order to use it and is described later. Credential authorization is an additional authorization (password) to the phone used to protect secure credential storage (such as VPN certificates and private keys).*

39 *The password is represented as a sequence of characters whose encoding depends on the TOE. This sequence must be conditioned into a string of bits that is to be used as input into the KEK. Conditioning can be performed using a PBKDF as defined in NIST SP 800-132 with one of the identified hash functions; the function used is selected by the ST Author. It should be noted that the hash function required by the key derivation function must be implemented by TOE.*

40 *The ST author shall choose between SHA-1, SHA-256, SHA-384, or SHA-512 when conditioning a password for the KEK. If the output of the conditioning function is greater than the size of the KEK, then the password shall be truncated to equal the size of the KEK.*

41 *In subsequent publications of this PP, it is likely that SHA-1 will no longer be an approved algorithm for cryptographic hashing.*

Assurance Activity:

42 *There are two aspects of this component that require evaluation: passwords of at least 8 characters are supported, and that the characters that are input are subject to the selected conditioning function. These activities are separately addressed in the text below.*

Support for password lengths of 8 characters

43 *The evaluators shall check the TSS section to determine that it specifies that a capability exists to accept passwords with the maximum number of characters specified in the ST in this assignment statement, and that the number specified is at least 8. The evaluators shall also check the operational guidance to determine that there are instructions for generating such passwords, and that guidance indicates how the passwords are entered into the TOE.*

44 *In addition to the analysis above, the evaluator shall also perform the following tests on a TOE configured according to the AGD_PRE guidance:*

Test 1: Ensure that the TOE supports passwords having a character length that is equal to or greater than the number specified in the ST for the first assignment.

Test 2: Ensure that the TOE supports passwords of shorter lengths consistent with what is specified in the operational guidance supplied by the vendor (for instance, if the guidance specifies that passwords have a minimum length of 16 characters, this test would minimally determine that 16-character passwords were accepted by the TOE).

Test 3: Ensure that the TOE contains support for passwords composed as specified in guidance contained in the AGD_OPR or AGD_PRE guidance. For instance, if the guidance specifies that passwords must contain a special character, this test would fail if the TOE only supported letters and numbers.

Password Conditioning

45 *The evaluator shall perform the following analysis using the information in the TSS section of the ST. It should be noted that this analysis is based entirely on claims in the TSS section of the ST; the evaluator is not expected to do any low-level testing to determine that the low-level cryptographic operations specified in the requirement are actually being performed.*

46 *For SHA-based conditioning of the password, the evaluator performs the following activities. The evaluator shall check that the TSS describes the method by which the password is first encoded and then fed to the SHA algorithm. The settings for the algorithm (padding, blocking, etc.) shall be described, and the evaluator shall verify that these are supported by the selections in this component as well as the selections in FCS_COP.1(3) concerning the hash function itself. The evaluator shall verify that the TSS contains a description of how the output of the hash function is used to form the authorization factor.*

FCS_CKM_EXT.2 Extended: Cryptographic Key Handling and Storage

FCS_CKM_EXT.2.1 The TSF shall store certificates, persistent secret and private keys when not in use in encrypted form as specified in FCS_COP.1(1).

Assurance Activity:

47 *The evaluator shall examine the TSS to ensure it describes in detail how user credentials, certificates, persistent secret and private keys are stored and encrypted. The evaluator reviews the TSS to determine that it makes a case that key material is not written unencrypted to persistent memory.*

FCS_CKM_EXT.4 Cryptographic key material destruction (Key Material)

FCS_CKM_EXT.4.1 The TSF shall destroy cryptographic keys contained within the TSF in accordance with the following rules:

a) Zeroization of all keying material shall be performed when it is no longer needed, on shutdown or when initialized.
b) For non-volatile memory other than EEPROM and flash, the zeroization shall be executed by overwriting three or more times using a different alternating data pattern each time.
c) For non-volatile EEPROM, the zeroization shall be executed by a single direct overwrite consisting of a pseudo random pattern based on the RBG specified in FCS_RBG_EXT.1, followed by a read-verify.
d) For volatile memory and non-volatile flash memory, the zeroization shall be executed by a single direct overwrite consisting of a single direct overwrite with zeros, followed by a read-verify.

that meets the following: no standard.

Application Note:

48 *The intent of this requirement is to ensure that key material does not remain in memory, enabling an attacker to not have to exhaust the AES key space. For added security a cryptographic key may be split*

into multiple key components, individually sharing no knowledge of the original key, which can be subsequently input into, or output from, a cryptographic module by separate entities and combined to recreate the original cryptographic key.

49 *Although verification of this zeroization of a plaintext key/critical cryptographic security parameter is desired here (by checking for the final known alternating data pattern), it is not required at this time. However, vendors are highly encouraged to incorporate this verification whenever possible into their implementations.*

50 *Compliance to Guidelines for Media Sanitization (SP 800-88 Rev1, dated Sept 2012) Appendix A, Table A-3 Mobile Device Sanitization is recommended.*

Assurance Activity:

51 *The evaluator shall check to ensure the TSS describes each of the secret keys (keys used for symmetric encryption), private keys, and Critical Security Parameters (CSPs) used to generate key; when they are zeroized (for example, immediately after use, on system shutdown, etc.); and the type of zeroization procedure that is performed (overwrite with zeros, overwrite three times with random pattern, etc.). 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 zeroization procedure in terms of the memory in which the data are stored (for example, "secret keys stored on flash are zeroized by overwriting once with zeros, while secret keys stored on the internal hard drive are zeroized by overwriting three times with a random pattern that is changed before each write").The evaluator shall ensure that the TSS contains a description of how the cryptographic functions in the FCS requirements are being used to perform the encryption functions, including how the keys are unwrapped and stored in the TOE. The evaluator shall ensure that the TSS describes how the key material is used to show that none of the key material is written to persistent memory in plaintext.*

52 *The evaluator reviews the TSS to determine that it makes a case that key material is not written unencrypted to persistent memory.*

FCS_COP.1 (1) Cryptographic operation (Key Masking)

FCS_COP.1.1(1) **Refinement:** The TSF shall perform **key masking** in accordance with a specified cryptographic algorithm **[selection: AES used in ECB mode; AES Key Wrap; AES Key Wrap with Padding]** and the cryptographic key size **[selection: 128 bits, 256 bits]** that meet the following: [**selection: "FIPS PUB 197, *Advanced Encryption Standard (AES)* and NIST SP 800-38A" for AES used in ECB mode; NIST SP 800-38F for Key Wrap].**

Application Note:
In the first selection, the ST author chooses the method by which the KEK is used to mask the DEK: either, using AES in ECB mode, or using one of the two AES-based Key Wrap methods specified in NIST SP 800-38F. Based on this selection, the last selection should be used to select the appropriate. The second selection should be made to reflect the size of the KEK.

53 *NIST SP 800-38F is currently in pre-publication; once formally published, the published version applies to new evaluations using this PP.*

Assurance Activity:
If AES in ECB mode is used, then the following assurance activities will be performed.

The evaluator shall ensure that the vendor has described the method/algorithm by which the KEK is used to mask the DEK using AES (for example, any options specified by the FIPS documents are identified, methods of padding the input, truncating the output, etc.).

The evaluator shall perform the following tests. If multiple modes are supported, the evaluator examines the TSS and guidance documentation to determine how ECB and the specified key-size is chosen by the end user. The evaluator then tests each key size in the manner found in the following sections, as appropriate. Note that some of these tests will require a reference implementation of the algorithms that is acceptable to the evaluation facility's Scheme.

ECB Mode

The ECB mode tests reference *The Advanced Encryption Standard Algorithm Validation Suite (AESAVS)* [AESAVS], available from http://csrc.nist.gov/groups/STM/cavp/documents/aes/AESAVS.pdf.

The evaluators shall run a set of known answer tests for each key size supported by the TSF. Inputs are a key and either plaintext to be encrypted or ciphertext to be decrypted. All of the test vectors (both encrypt and decrypt) for ECB mode in the supported key lengths from http://csrc.nist.gov/groups/STM/cavp/documents/aes/KAT_AES.zip shall be used to perform these tests

The evaluators shall perform a multi-block message test for each key length supported. To perform this test, the evaluators generated 10 data sets for encryption and 10 data sets for decryption. Each data set consists of key and plaintext (for encryption) or ciphertext (for decryption). The length of a block shall be 128 bits; the length of the plaintext/ ciphertext shall be block length * i, where i indicates the data set number and i ranges from 1 to 10 (so messages will range from 128 bits to 1280 bits).

The evaluators shall perform a Monte Carlo test. The evaluators shall generate 10 sets of starting values for encryption (values for the key and plaintext) and 10 sets of starting values for decryption (values for the key and ciphertext). The length of the plaintext/ciphertext shall be 128 bits. Each set of starting values is used to generate and perform 100 tests; the algorithm for generating the 100 test values (per set of starting values) is contained in section 6.4.1 of [AESAVS].

If AES Key Wrap or AES Key Wrap with padding is used, then the following assurance activities shall be performed.

The evaluation team shall check the TSS to ensure it vigorously asserts that the Key Wrap specification is met.

FCS_COP.1(2) Cryptographic operation (Signature Verification)

FCS_COP.1.1(2) **Refinement:** The TSF shall perform *cryptographic signature services* in accordance with **the following** specified cryptographic algorithms ***[selection:***

(1) Digital Signature Algorithm (DSA) with a key size (modulus) of 2048 bits or greater;
(2) RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 2048 bits or greater;
(3) Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of 256 bits and 384 bits;]

that meets the following:

Case: Digital Signature Algorithm

- *[selection: FIPS PUB 186-3, "Digital Signature Standard" , FIPS PUB 186-2, "Digital Signature Standard"]*

Case: RSA Digital Signature Algorithm

- *[selection: FIPS PUB 186-3, "Digital Signature Standard" , FIPS PUB 186-2, "Digital Signature Standard"]*

Case: Elliptic Curve Digital Signature Algorithm

- *[selection: FIPS PUB 186-3, "Digital Signature Standard" , FIPS PUB 186-2, "Digital Signature Standard"]*
- *The TSF shall implement "NIST curves" P-256, P-384 and [selection: P-521, no other curves] (as defined in FIPS PUB 186-3, "Digital Signature Standard")*

Application Note:

54 *Future versions of the PP will only allow ECDSA 256 and 384.*

55 *There are three instances where digital signatures are to be used in the TOE; updates to the TOE, installation and updates to the mobile applications, and verifying the integrity of the TOE's software when loaded upon startup. If different algorithms are employed when performing these functions, than the requirement must be iterated for each instance.*

56 *The ST author should choose the algorithm implemented to perform digital signatures. For the algorithm chosen, the ST author should make the appropriate assignments/selections to specify the parameters that are implemented for that algorithm.*

57 *For the elliptic curve-base schemes, the key size refers to the log_2 of the order of the base point. As the preferred approach for the digital signatures ECDDSA will be required in future publications of this PP.*

Assurance Activity:

58 *The evaluator shall use the signature verification portions of "The Digital Signature Algorithm Validation System" (DSAVS or DSA2VS), "The Elliptic Curve Digital Signature Algorithm Validation System" (ECDSAVS or ECDSA2VS), and "The RSA Validation System" (RSAVS) as a guide in testing the requirement above The Validation System used shall comply with the conformance standard identified in the ST (i.e. FIPS 186-2 or FIPS PUB 186-3). This will require that the evaluator have reference implementation of the algorithms known to be that can produce test vectors that are verifiable during the test.*

FCS_COP.1(3) Cryptographic operation (Cryptographic Hashing)

FCS_COP.1.1(3) **Refinement:** The TSF shall perform *cryptographic hashing services* in accordance with a specified cryptographic algorithm *[selection: SHA-1, SHA-224, SHA-256, SHA-384, SHA-512]* and **message digest** sizes [**selection: 160, 224, 256, 384, 512**] **bits** that meet the following: *FIPS Pub 180-3, "Secure Hash Standard."*

Application Note:

59 *The intent of this requirement is to specify the hashing function used when password conditioning for key encryption key (KEK) derivation and the digital signature checking associated with trusted updates and*

certificate and CRL signing. The hash selection must support the message digest size selection. The hash selection should be consistent with the overall strength of the algorithm used for FCS_COP.1(1) and FCS_COP.1(2).

Assurance Activity:

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

61 The cryptographic hashing tests reference The Secure Hash Algorithm Validation System (SHAVS) [SHAVS], available from *http://csrc.nist.gov/groups/STM/cavp/documents/shs/SHAVS.pdf*.

62 *The TSF hashing functions can be implemented in one of two modes. The first mode is the byte-oriented mode. In this mode the TSF only hashes messages that are an integral number of bytes in length; i.e., the length (in bits) of the message to be hashed is divisible by eight. The second mode is the bit-oriented mode. In this mode the TSF hashes messages of arbitrary length. As there are different tests for each mode, an indication is given in the following sections for the bit-oriented vs. the byte-oriented tests.*

63 *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.*

Short Messages Test - Bit-oriented Mode

64 *The evaluator devises an input set consisting of m+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m bits. The message text shall be psuedorandomly generated. The evaluator computes the message digest for each of the messages and ensures that the correct result is produced when the messages are provided to the TSF.*

Short Messages Test - Byte-oriented Mode

65 *The evaluator devises 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 psuedorandomly generated. The evaluator computes the message digest for each of the messages and ensures that the correct result is produced when the messages are provided to the TSF.*

Selected Long Messages Test - Bit-oriented Mode

66 *The evaluator devises 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 ≤ i ≤ m. The message text shall be psuedorandomly generated. The evaluator computes the message digest for each of the messages and ensures that the correct result is produced when the messages are provided to the TSF.*

Selected Long Messages Test - Byte-oriented Mode

67 *The evaluator devises 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 psuedorandomly generated. The evaluator computes the message digest for each of the messages and ensures that the correct result is produced when the messages are provided to the TSF.*

Pseudorandomly Generated Messages Test

68 *This test is for byte-oriented implementations only. The evaluator randomly generates 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 evaluator then formulates a set of 100 messages and associated digests by following the algorithm provided in Figure 1 of [SHAVS]. The evaluator then ensures that the correct result is produced when the messages are provided to the TSF.*

FCS_COP.1(4) Cryptographic Operation (For keyed-hash Message Authentication)

FCS_COP.1.1(4) **Refinement:** The TSF shall perform *keyed-hash message authentication* in accordance with a specified cryptographic algorithm *HMAC-[selection: SHA-1, SHA-256, SHA-384, SHA-512],* key sizes *[assignment: key size (in bits) used in HMAC],* **and message digest sizes** [**selection: 160, 256, 384, 512**] **bits** that meet the following: *FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code, and FIPS Pub 180-3, "Secure Hash Standard."*

Assurance Activity:

69 *The evaluator shall use "The Keyed-Hash Message Authentication Code (HMAC) Validation System (HMACVS) " as a guide in testing the requirement above. This will require that the evaluator have a reference implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.*

70

FCS_COP.1(5) Cryptographic Operation (Data Encryption/Decryption)

FCS_COP.1.1(5) **Refinement:** The TSF shall perform [*encryption and decryption*] in accordance with a specified cryptographic algorithm *AES operating in* **GCM, CBC,** [**assignment:** ***one or more modes, no other modes***]] and cryptographic key sizes 128-bits, 256-bits, and [**selection: 192 bits, no other key sizes**] that meets the following:

- **FIPS PUB 197, "Advanced Encryption Standard (AES)"**
- **NIST SP 800-38D, NIST SP 800-38A [selection:, NIST SP 800-38B, NIST SP 800-38C, NIST SP 800-38E, no other standards**

  ]

Application Note:

71 *This PP requires the modes GCM and CBC to be used in the IPsec and IKE protocols (FCS_IPSEC_EXT.1.4, FCS_IPSEC_EXT.1.6).*

Assurance Activity:

The evaluator shall use tests appropriate to the modes selected in the above requirement from "The Advanced Encryption Standard Algorithm Validation Suite (AESAVS)", "The XTS-AES Validation System (XTSVS)", The CMAC Validation System (CMACVS)", "The Counter with Cipher Block Chaining-Message Authentication Code (CCM) Validation System (CCMVS)", and "The Galois/Counter Mode (GCM) and GMAC Validation System (GCMVS)" (these documents are available from http://csrc.nist.gov/groups/STM/cavp/index.html) as a guide in testing the requirement above. This will *require that the evaluator have a trusted reference implementation of the algorithms that can produce test vectors that are verifiable during the test.*

FCS_RBG_EXT.1 Extended: Cryptographic operation (Random Bit Generation)

FCS_RBG_EXT.1.1 The TSF shall perform all random bit generation (RBG) services in accordance with [selection, choose one of: NIST Special Publication 800-90A using [selection: Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES), Dual_EC_DRBG (any)]; FIPS Pub 140-2 Annex C: X9.31 Appendix 2.4 using AES] seeded by an entropy source that accumulates entropy from [selection: a software-based noise source; TSF-hardware-based noise sources].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded with a minimum of [selection, choose one of: 128 bits, 256 bits] of entropy at least equal to the greatest bit length of the keys and authorization factors that it will generate.

Application Note:

72 *NIST Special Pub 800-90A, Appendix C describes the minimum entropy measurement that will probably be required future versions of FIPS-140. If possible this should be used immediately.*

73 *For the first selection in FCS_RBG_EXT.1.1, the ST author should select the standard to which the RBG services comply (either 800-90A or 140-2 Annex C).*

74 *SP 800-90A contains four different methods of generating random numbers; each of these, in turn, depends on underlying cryptographic primitives (hash functions/ciphers). The ST author will select the function used (if 800-90A is selected), and include the specific underlying cryptographic primitives used in the requirement or in the TSS. While any of the identified hash functions (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512) are allowed for Hash_DRBG or HMAC_DRBG, only AES-based implementations for CTR_DRBG are allowed. While any of the curves defined in 800-90A are allowed for Dual_EC_DRBG, the ST author not only must include the curve chosen, but also the hash algorithm used.*

75 *For the third selection in FCS_RBG_EXT.1.1, the ST author indicates whether the source of entropy are software-based, hardware-based, or both. If there are multiple sources of entropy, the ST author will elaborate each entropy sources and whether it is hardware- or software –based. Hardware-based noise sources are preferred.*

76 *Note that for FIPS Pub 140-2 Annex C, currently only the method described in NIST-Recommended Random Number Generator Based on ANSI X9.31 Appendix A.2.4 Using the 3-Key Triple DES and AES Algorithm, Section 3 is valid. If the key length for the AES implementation used here is different than that used to encrypt the credentials, then FCS_COP.1 may have to be adjusted or iterated to reflect the different key length. For the selection in FCS_RBG_EXT.1.2, the ST author selects the minimum number of bits of entropy that is used to seed the RBG.*

77 *The ST author also ensures that any underlying functions are included in the baseline requirements for the TOE. Hash functions are required for Hash_DRBG, HMAC_DRBG, and Dual_EC_DRBG.*

Assurance Activity:

Documentation shall be produced—and the evaluator shall perform the activities—in accordance with Annex D, Entropy Documentation and Assessment.

78 *The evaluator shall also perform the following tests, depending on the standard to which the RBG conforms.*

79 ***Test***

80 *The evaluator shall also perform the following tests, depending on the standard to which the RBG conforms.*

Implementations Conforming to FIPS 140-2, Annex C

81 *The reference for the tests contained in this section is The Random Number Generator Validation System (RNGVS) [RNGVS]. The evaluators shall conduct the following two tests. Note that the "expected values" are produced by a reference implementation of the algorithm that is known to be correct. Proof of correctness is left to each Scheme.*

82 *The evaluators shall perform a Variable Seed Test. The evaluators shall provide a set of 128 (Seed, DT) pairs to the TSF RBG function, each 128 bits. The evaluators shall also provide a key (of the length appropriate to the AES algorithm) that is constant for all 128 (Seed, DT) pairs. The DT value is incremented by 1 for each set. The seed values shall have no repeats within the set. The evaluators ensure that the values returned by the TSF match the expected values.*

83 *The evaluators shall perform a Monte Carlo Test. For this test, they supply an initial Seed and DT value to the TSF RBG function; each of these is 128 bits. The evaluators shall also provide a key (of the length appropriate to the AES algorithm) that is constant throughout the test. The evaluators then invoke the TSF RBG 10,000 times, with the DT value being incremented by 1 on each iteration, and the new seed for the subsequent iteration produced as specified in NIST-Recommended Random Number Generator Based on ANSI X9.31 Appendix A.2.4 Using the 3-Key Triple DES and AES Algorithms, Section 3. The evaluators ensure that the 10,000th value produced matches the expected value.*

Implementations Conforming to NIST Special Publication 800-90A

84 *The evaluator shall perform 15 trials for the RBG implementation. If the RBG is configurable, the evaluator shall perform 15 trials for each configuration. The evaluator shall also confirm that the operational guidance contains appropriate instructions for configuring the RBG functionality.*

85 *If the RBG has prediction resistance enabled, each trial consists of (1) instantiate the deterministic RBG (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 the number of returned bits equal to the Output Block Length (as defined in NIST SP 800-90A).*

86 *If the RBG 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 re-seed. The final value is additional input to the second generate call.*

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

88 ***Entropy input:*** *the length of the entropy input value must equal the seed length.*

89 ***Nonce:*** *If a nonce is supported (CTR_DRBG with no derivation function (df) does not use a nonce), the nonce bit length is one-half the seed length.*

90 ***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 supported, 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.*

91 ***Additional input:*** *the additional input bit lengths have the same defaults and restrictions as the personalization string lengths.*

FCS_IPSEC_EXT.1 Extended: Internet Protocol Security (FCS_IPSEC_EXT)

FCS_IPSEC_EXT.1.1 The TSF shall implement the IPsec architecture as specified in RFC 4301.

Assurance Activity:

In order to show that the TSF implements the RFCs correctly, the evaluator shall perform the assurance activities listed below. In future versions of this EP, assurance activities may be augmented, or new ones introduced that cover more aspects of RFC compliance than is currently described in this publication.

TSS

Nothing is done in addition to determining that the TOE's implementation is conformant to RFC 4301 as described above.

Guidance

The evaluator shall examine the operational guidance to verify it instructs the MDM how to construct entries into the SPD that specify a rule for DISCARD, BYPASS and PROTECT.

Test

The evaluator uses the operational guidance to configure the TOE to carry out the following tests:

Test 1: The evaluator shall configure the TOE's SPD such that there is a rule for DISCARD, BYPASS, PROTECT. The selectors used in the construction of the rule shall be different such that the evaluator can send in three network packets with the appropriate fields in the packet header that each packet will match one of the three rules. The evaluator observes via the audit trail, and packet captures that the TOE exhibited the expected behavior: appropriate packet was dropped, allowed through without modification, was encrypted by the IPsec implementation.

Test 2: The evaluator shall devise two equal SPD entries with alternate operations – BYPASS and PROTECT. The entries should then be deployed in two distinct orders and in each case the evaluator shall ensure that the first entry is enforced in both cases by generating applicable packets and using packet capture and logs for confirmation.

Test 3: The evaluator shall repeat the procedure above, except that the two entries should be devised where one is a subset of the other (e.g., a specific address vs. a network segment). Again, the evaluator should test both orders to ensure that the first is enforced regardless of the specificity of the rule.

FCS_IPSEC_EXT.1.2 The TSF shall implement [selection, choose at least one of: tunnel mode, transport mode].

Assurance Activity:
TSS
The evaluator checks the TSS to ensure it states that the TOE can operate in tunnel mode and/or transport mode (as selected).
Guidance
The evaluator shall confirm that the operational guidance instructs the MDM how the TOE is configured in each mode selected.
Test
Test 1 (conditional): If tunnel mode is selected, the evaluator uses the operational guidance to configure the TOE to operate in tunnel mode and also configures a VPN GW to operate in tunnel mode. The evaluator configures the TOE and the VPN GW to use any of the allowable cryptographic algorithms, authentication methods, etc. to ensure an allowable SA can be negotiated. The evaluator shall then initiate a connection from the TOE to connect to the VPN GW peer. The evaluator observes in the audit trail and the captured packets that a successful connection was established using the tunnel mode.
Test 2 (conditional): If transport mode is selected, the evaluator uses the operational guidance to configure the TOE to operate in transport mode and also configures a VPN GW to operate in transport mode. The evaluator configures the TOE and the VPN GW to use any of the allowed cryptographic algorithms, authentication methods, etc. to ensure an allowable SA can be negotiated. The evaluator then initiates a connection from the TOE to connect to the VPN GW. The evaluator observes in the audit trail and the captured packets that a successful connection was established using the transport mode.

FCS_IPSEC_EXT.1.3 The TSF shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched, and discards it.

Assurance Activity:
TSS
The evaluator shall examine the TSS to verify that the TSS provides a description of how a packet is processed against the SPD and that if no “rules” are found to match, that a final rule exists, either implicitly or explicitly, that causes the network packet to be discarded.
Guidance
The evaluator checks that the operational guidance provides instructions on how to construct the SPD and uses the guidance to configure the TOE for the following tests.
Test
Test 1: The evaluator shall configure the TOE’s SPD, such that it has entries that contain operations that DISCARD, BYPASS, and PROTECT network packets. The evaluator also configures the TOE so that all auditable events with respect to FCS_IPSEC_EXT.1 are enabled. The evaluator may use the SPD that was created for verification of FCS_IPSEC_EXT.1.1. The evaluator shall construct a network packet that matches a BYPASS entry, and send that packet to the TOE. The evaluator should observe that the network packet is passed by the TOE to the proper destined interface with no modification. The evaluator shall then modify a field in the packet header; such that it no longer matches the evaluator created entries (there may be a “TOE created” final entry that discards packets that do not match any previous entries). The evaluator sends the packet to the TOE, and observes that the packet was not permitted to flow to any of the TOE’s interfaces. The evaluator shall verify that an audit record is generated that specifies that the packet was discarded as expected.

FCS_IPSEC_EXT.1.4 The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms AES-GCM-128, AES-GCM-256 as specified in RFC 4106, [selection: AES-CBC-128, AES-CBC-256 (both specified by RFC 3602) together with a Secure Hash Algorithm (SHA)-based HMAC, no other algorithms].

Assurance Activity:
TSS
The evaluator shall examine the TSS to verify that the algorithms AES-GCM-128 and AES-GCM-256 are implemented. If the ST author has selected either AES-CBC-128 or AES-CBC-256 in the requirement, then the evaluator verifies the TSS describes these as well. In addition, the evaluator ensures that the SHA-based HMAC algorithm conforms to the algorithms specified in FCS_COP.1(4) Cryptographic Operations (for keyed-hash message authentication).
Guidance
The evaluator checks the operational guidance to ensure it provides instructions on how to configure the TOE to use the AES-GCM-128, and AES-GCM-256 algorithms, and if either AES-CBC-128 or AES-CBC-256 have been selected the guidance instructs how to use these as well.
Test
Test 1: The evaluator shall configure the TOE as indicated in the operational guidance configuring the TOE to using each of the AES-GCM-128, and AES-GCM-256 algorithms, and attempt to establish a connection using ESP. If the ST Author has selected either AES-CBC-128 or AES-CBC-256, the TOE is configured to use those algorithms and the evaluator attempts to establish a connection using ESP for those algorithms selected.

FCS_IPSEC_EXT.1.5 The TSF shall implement the protocol: [selection, choose at least one of: IKEv1 as defined in RFCs 2407, 2408, 2409, RFC 4109, [selection: no other RFCs for extended sequence numbers, RFC 4304 for extended sequence numbers], and [selection: no other RFCs for hash functions, RFC 4868 for hash functions]; IKEv2 as defined in RFCs 5996 (with mandatory support for NAT traversal as specified in section 2.23), 4307, and [selection: no other RFCs for hash functions, RFC 4868 for hash functions]].

Assurance Activity:
TSS
The evaluator shall examine the TSS to verify that IKEv1 and/or IKEv2 are implemented.
Guidance
The evaluator checks the operational guidance to ensure it instructs the MDM how to configure the TOE to use IKEv1 and/or IKEv2 (as selected), and uses the guidance to configure the TOE to perform NAT traversal for the following test.
Test
Test 1: The evaluator shall configure the TOE so that it will perform NAT traversal processing as described in the TSS and RFC 5996, section 2.23. The evaluator shall initiate an IPsec connection and determine that the NAT is successfully traversed.

FCS_IPSEC_EXT.1.6 The TSF shall ensure the encrypted payload in the [selection, choose at least one of: IKEv1, IKEv2] protocol uses the cryptographic algorithms AES-CBC-128, AES-CBC-256 as specified in RFC 6379 and [selection: AES-GCM-128, AES-GCM-256 as specified in RFC 5282, no other algorithm].

Assurance Activity:
TSS

The evaluator shall ensure the TSS identifies the algorithms used for encrypting the IKEv1 and/or IKEv2 payload, and that the algorithms AES-CBC-128, AES-CBC-256 are specified, and if others are chosen in the selection of the requirement, those are included in the TSS discussion.

Guidance

The evaluator ensures that the operational guidance describes how the TOE can be configured to use the mandated algorithms, as well as any additional algorithms selected in the requirement. The guidance is then used to configure the TOE to perform the following test.

Test

Test 1: The evaluator shall configure the TOE to use AES-CBC-128 to encrypt the IKEv1 and/or IKEv2 payload and establish a connection with a peer device, which is configured to only accept the payload encrypted using AES-CBC-128. The evaluator will consult the audit trail to confirm the algorithm was that used in the negotiation.

FCS_IPSEC_EXT.1.7 The TSF shall ensure that IKEv1 Phase 1 exchanges use only main mode.

Assurance Activity:

TSS

The evaluator shall examine the TSS to ensure that, in the description of the IPsec protocol supported by the TOE, it states that aggressive mode is not used for IKEv1 Phase 1 exchanges, and that only main mode is used. It may be that this is a configurable option.

Guidance

If the mode requires configuration of the TOE prior to its operation, the evaluator shall check the operational guidance to ensure that instructions for this configuration are contained within that guidance.

Test

Test 1 (conditional): The evaluator shall configure the TOE as indicated in the operational guidance, and attempt to establish a connection using an IKEv1 Phase 1 connection in aggressive mode. This attempt should fail. The evaluator should then show that main mode exchanges are supported. This test is not applicable if IKEv1 is not selected above in the FCS_IPSEC_EXT.1.5 protocol selection.

FCS_IPSEC_EXT.1.8 The TSF shall ensure that [selection: IKEv2 SA lifetimes can be configured by [selection: a MDM, VPN Gateway] based on number of packets/number of bytes or length of time, where the time values can be limited to: 24 hours for Phase 1 SAs and 8 hours for Phase 2 SAs, IKEv1 SA lifetimes can be configured by an [selection: a MDM, VPN Gateway] based on number of packets/number of bytes or length of time, where the time values can be limited to: 24 hours for Phase 1 SAs and 8 hours for Phase 2 SAs].

Application Note:

The ST Author is afforded a selection based on the version of IKE in their implementation. There is a further selection within this selection that allows the ST Author to specify which entity is responsible for "configuring" the life of the SA. An implementation that allows an MDM to configure the client or a VPN gateway that pushes the SA lifetime down to the client are both acceptable.

As far as SA lifetimes are concerned, the TOE can limit the lifetime based on the number of bytes transmitted, or the number of packets transmitted. Either packet-based or volume-based SA lifetimes are acceptable.

Assurance Activity:

TSS

How the lifetimes are established and enforced is described in the RFCs and the evaluator examines the TSS as stated at the beginning of this section.

Guidance

The evaluator verifies that the values for SA lifetimes can be configured and that the instructions for doing so are located in the operational guidance. The evaluator ensures that either the MDM or VPN Gateway are able to configurable Phase 1 SAs values for 24 hours and 8 hours for Phase 2 SAs. Currently there are no values mandated for the number of packets or number of bytes, the evaluator just ensures that this can be configured.

Test

When testing this, the evaluator needs to ensure that both sides are configured appropriately. From the RFC "A difference between IKEv1 and IKEv2 is that in IKEv1 SA lifetimes were negotiated. In IKEv2, each end of the SA is responsible for enforcing its own lifetime policy on the SA and rekeying the SA when necessary. If the two ends have different lifetime policies, the end with the shorter lifetime will end up always being the one to request the rekeying. If the two ends have the same lifetime policies, it is possible that both will initiate a rekeying at the same time (which will result in redundant SAs). To reduce the probability of this happening, the timing of rekeying requests SHOULD be jittered."

Each of the following tests shall be performed for each version of IKE selected in the FCS_IPSEC_EXT.1.5 protocol selection:

Test 1: The evaluator shall configure a maximum lifetime in terms of the # of packets (or bytes) allowed following the operational guidance. The evaluator shall establish an SA and determine that once the allowed # of packets (or bytes) through this SA is exceeded, the connection is closed.

Test 2: The evaluator shall construct a test where a Phase 1 SA is established and attempted to be maintained for more than 24 hours before it is renegotiated. The evaluator shall observe that this SA is closed or renegotiated in 24 hours or less. If such an action requires that the TOE be configured in a specific way, the evaluator shall implement tests demonstrating that the configuration capability of the TOE works as documented in the operational guidance.

Test 3: The evaluator shall perform a test similar to Test 1 for Phase 2 SAs, except that the lifetime will be 8 hours instead of 24.

FCS_IPSEC_EXT.1.9 The TSF shall generate the secret value x used in the IKE Diffie-Hellman key exchange ("x" in g^x mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least [assignment: (one or more) number(s) of bits that is at least twice the "bits of security" value associated with the negotiated Diffie-Hellman group as listed in Table 2 of NIST SP 800-57, *Recommendation for Key Management – Part 1: General*] bits.

Assurance Activity:

The evaluator shall check to ensure that, for each DH group supported by the TSF, the TSS describes the process for generating "x" (as defined in FCS_IPSEC_EXT.1.9) and each nonce. The evaluator shall verify that the TSS indicates that the random number generated that meets the requirements in this PP is used, and that the length of "x" and the nonces meet the stipulations in the requirement.

FCS_IPSEC_EXT.1.10 The TSF shall generate nonces used in IKE exchanges in a manner such that the probability that a specific nonce value will be repeated during the life a specific IPsec SA is less than 1 in 2^[assignment: (one or more) "bits of security" value(s) associated with the negotiated Diffie-Hellman group as listed in Table 2 of NIST SP 800-57, *Recommendation for Key Management – Part 1: General*] .

Assurance Activity:

The evaluator shall check to ensure that, for each DH group supported by the TSF, the TSS describes the process for generating "x" (as defined in FCS_IPSEC_EXT.1.9) and each nonce. The evaluator shall verify that the TSS indicates that the random number generated that meets the requirements in this PP is used, and that the length of "x" and the nonces meet the stipulations in the requirement.

FCS_IPSEC_EXT.1.11 The TSF shall ensure that all IKE protocols implement DH Groups 14 (2048-bit MODP), 19 (256-bit Random ECP), and [selection: 5 (1536-bit MODP), 24 (2048-bit MODP with 256-bit POS), 20 (384-bit Random ECP), [assignment: other DH groups that are implemented by the TOE], no other DH groups].

Assurance Activity:

The evaluator shall check to ensure that the DH groups specified in the requirement are listed as being supported in the TSS. If there is more than one DH group supported, the evaluator checks to ensure the TSS describes how a particular DH group is specified/negotiated with a peer. The evaluator shall also perform the following test:

Test 1: For each supported DH group, the evaluator shall test to ensure that all IKE protocols can be successfully completed using that particular DH group.

FCS_IPSEC_EXT.1.12 The TSF shall ensure that all IKE protocols perform peer authentication using a [selection, choose at least one of: *RSA, ECDSA*] that use X.509v3 certificates that conform to RFC 4945 and [selection: Pre-shared Keys, no other method].

Assurance Activity:

TSS

The evaluator ensures that the TSS identifies RSA and/or ECDSA as being used to perform peer authentication. The description must be consistent with the algorithms as specified in FCS_COP.1(2) Cryptographic Operations (for cryptographic signature).

If pre-shared keys are chosen in the selection, the evaluator shall check to ensure that the TSS describes how pre-shared keys are established and used in authentication of IPsec connections. The evaluator shall check that the operational guidance describes how pre-shared keys are to be generated and established for a TOE. The description in the TSS and the operational guidance shall also indicate how pre-shared key establishment is accomplished for both TOEs that can generate a pre-shared key as well as TOEs that simply use a pre-shared key.

Guidance

The evaluator ensures the operational guidance describes how to set up the TOE to use the cryptographic algorithms RSA and/or ECDSA.

In order to construct the environment and configure the TOE for the following tests, the evaluator will ensure that the operation guidance also describes how to configure the TOE to connect to a trusted CA, and ensure a valid certificate for that CA is loaded into the TOE and marked "trusted".

Test

For efficiency sake, the testing that is performed here has been combined with aspects of the testing for FIA_X509_EXT.1 Extended: X.509 Certificates, specifically FIA_X509_EXT.1.4, and FIA_X509_EXT.1.5.

The following tests shall be repeated for each peer authentication protocol selected in the FCS_IPSEC_EXT.1.12 selection above:

Test 1: The evaluator shall have the TOE generate a public-private key pair, and submit a CSR (Certificate Signing Request) to a CA (trusted by both the TOE and the peer VPN used to establish a connection) for

its signature. The values for the DN (Common Name, Organization, Organizational Unit, and Country) will also be passed in the request.
Test 2: The evaluator shall use a certificate signed using the RSA or ECDSA algorithm to authenticate the remote peer during the IKE exchange. This test ensures the remote peer has the certificate for the trusted CA that signed the TOE's certificate and it will do a bit-wise comparison on the DN. This bit-wise comparison of the DN ensures that not only does the peer have a certificate signed by the trusted CA, but the certificate is from the DN that is expected. The evaluator will configure the TOE to associate a certificate (e.g., a certificate map in some implementations) with a VPN connection. This is what the DN is checked against.
Test 3: The evaluator shall test that the TOE can properly handle revoked certificates – conditional on whether CRL or OCSP is selected; if both are selected, and then a test is performed for each method. For this draft of the EP, the evaluator has to only test one up in the trust chain (future drafts may require to ensure the validation is done up the entire chain). The evaluator shall ensure that a valid certificate is used, and that the SA is established. The evaluator then attempts the test with a certificate that will be revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the TOE will not establish an SA.
Test 4: The evaluator shall test that given a signed certificate from a trusted CA, that when the DN does not match – any of the four fields can be modified such that they do not match the expected value, that an SA does not get established.
Test 5: The evaluator shall ensure that the TOE is configurable to either establish an SA, or not establish an SA if a connection to the certificate validation entity cannot be reached. For each method selected for certificate validation, the evaluator attempts to validate the certificate – for the purposes of this test, it does not matter if the certificate is revoked or not. For the "mode" where an SA is allowed to be established, the connection is made. Where the SA is not to be established, the connection is refused.
Test 6 [conditional]: The evaluator shall generate a pre-shared key and use it, as indicated in the operational guidance, to establish an IPsec connection between the TOE and the VPN GW peer. If the TOE supports generation of the pre-shared key, the evaluator shall ensure that establishment of the key is carried out for an instance of the TOE generating the key as well as an instance of the TOE merely taking in and using the key.

FCS_IPSEC_EXT.1.13 The TSF shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [selection: *IKEv1 Phase 1, IKEv2 IKE_SA*] connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [selection: *IKEv1 Phase 2, IKEv2 CHILD_SA*] connection.

Assurance Activity:

TSS
The evaluator shall check that the TSS describes the potential strengths (in terms of the number of bits in the symmetric key) of the algorithms that are allowed for the IKE and ESP exchanges. The TSS shall also describe the checks that are done when negotiating IKEv1 Phase 2 and/or IKEv2 CHILD_SA suites to ensure that the strength (in terms of the number of bits of key in the symmetric algorithm) of the negotiated algorithm is less than or equal to that of the IKE SA this is protecting the negotiation.

Guidance
The evaluator simply follows the guidance to configure the TOE to perform the following tests.

Test

Test 1: This test shall be performed for each version of IKE supported by the TOE. The evaluator shall successfully negotiate an IPsec connection using each of the supported algorithms and hash functions identified in the requirements.

Test 2: This test shall be performed for each version of IKE supported by the TOE. The evaluator shall attempt to establish an SA for ESP that selects an encryption algorithm with more strength than that being used for the IKE SA (i.e., symmetric algorithm with a key size larger than that being used for the IKE SA). Such attempts should fail.

Test 3: This test shall be performed for each version of IKE supported by the TOE. The evaluator shall attempt to establish an IKE SA using an algorithm that is not one of the supported algorithms and hash functions identified in the requirements. Such an attempt should fail.

Test 4: This test shall be performed for each version of IKE supported by the TOE. The evaluator shall attempt to establish an SA for ESP (assumes the proper parameters where used to establish the IKE SA) that selects an encryption algorithm that is not identified in FCS_IPSEC_EXT.1.4. Such an attempt should fail.

Application Note:

92 *FCS_IPSEC_EXT.1.7* is only applicable if IKEv1 is selected.

93 *FCS_IPSEC_EXT.1.8: The ST author chooses either the IKEv1 requirements or IKEv2 requirements (or both, depending on the selection in FCS_IPSEC_EXT.1.5. The IKEv1 requirement can be accomplished either by providing MDM-configurable lifetimes (with appropriate instructions in documents mandated by AGD_OPE), or by "hard coding" the limits in the implementation. For IKEv2, there are no hardcoded limits, but in this case it is required than an MDM be able to configure the values. In general, instructions for setting the parameters of the implementation, including lifetime of the SAs, should be included in the administrative guidance generated for AGD_OPE. It is appropriate to refine the requirement in terms of number of MB/KB instead of number of packets, as long as the TOE is capable of setting a limit on the amount of traffic that is protected by the same key (the total volume of all IPsec traffic protected by that key). It is also appropriate to refine the requirement such that SA lifetime management and enforcement occurs external to the TOE (i.e.: on a VPN Gateway), however, even when the requirement is refined in this manner, the evaluator shall conduct the associated assurance activities described above.*

94 *Since the implementation may allow different Diffie-Hellman groups to be negotiated for use in forming the SAs, the assignments in FCS_IPSEC_EXT.1.9 and FCS_IPSEC_EXT.1.10 may contain multiple values. For each DH group supported, the ST author consults Table 2 in 800-57 to determine the "bits of security" associated with the DH group. Each unique value is then used to fill in the assignment (for 1.9 they are doubled; for 1.10 they are inserted directly into the assignment). For example, suppose the implementation supports DH group 14 (2048-bit MODP) and group 20 (ECDH using NIST curve P-384). From Table 2, the bits of security value for group 14 is 112, and for group 20 it is 192. For FCS_IPSEC_EXT.1.9, then, the assignment would read "[224, 384]" and for FCS_IPSEC_EXT.1.10 it would read "[112,192]" (although in this case the requirement should probably be refined so that it makes sense mathematically).*

95 *FCS_IPSEC_EXT.1.11: The selection is used to specify additional DH groups supported. This applies to IKEv1 and IKEv2 exchanges. In future versions of this PP, DH Group 20 (384-bit RandomECP) will be required. It should be noted that if any additional DH groups are specified, they must comply with the requirements (in terms of the ephemeral keys that are established) listed in FCS_CKM.1.*

96 *FCS_IPSEC_EXT.1.12: At least one public-key-based Peer Authentication method is required for conformant TOEs; one or more of the public key schemes is chosen by the ST author to reflect what is implemented by the TOE. The ST author also ensures that appropriate FCS requirements reflecting the algorithms used (and key generation capabilities, if provided) are listed to support those methods. Note that the TSS will elaborate on the way in which these algorithms are to be used (for example, 2409 specifies three authentication methods using public keys; each one supported will be described in the TSS).*

FCS_IPSEC_EXT.1.13: The ST author chooses either or both of the IKE selections based on what is implemented by the TOE. Obviously, the IKE version(s) chosen should be consistent not only in this element, but with other choices for other elements in this component. While it is acceptable for a TOE to allow this capability to be configurable, the default configuration in the evaluated configuration (either "out of the box" or by configuration guidance in the OPE documentation) must enable this functionality.

4.2.2 User Data Protection (FDP)

FDP_ACC.1(1) Subset Access Control (Mobility Applications)

FDP_ACC.1.1(1) The TSF shall enforce the *Mobility Application Access Control Policy* on *mobile applications, [assignment: list of objects covered by the SFP], and [assignment: list of operations among subjects and objects covered by the SFP].*

Application Note:

97 *This requirement ensures that the TOE implements the Mobility Application Access Control Policy that governs the access mobile applications have to the objects residing on the device. The ability to influence the access applications have is limited to the MDM.*

Assurance Activity:

98 *The evaluator shall examine the TSS to verify that it describes how the Mobility Application Access Control Policy described in FDP_ACF.1 is implemented.*

99 *The evaluator shall ensure that all access controls for the enumerated objects work as described. The evaluator shall also perform tests described in the assurance activity for the FDP_ACF.1.*

FDP_ACF.1 Security Attribute Based Access Control (Mobility Application)

FDP_ACF.1.1 The TSF shall enforce the *Mobility Application Access Control Policy* to objects based on the following: *mobile applications* [assignment: list objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects].

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [assignment: rules, based on security attributes, that explicitly authorize access of subjects to objects].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [assignment: rules, based on security attributes, that explicitly deny access of subjects to objects].

Application Note:

100 *This requirement ensures that the TOE implements the Mobility Application Access Policy which defines explicit permissions that mobile applications can have on objects within the TOE.*

Assurance Activity:

TSS

101 *The evaluator shall examine the TSS to verify that it describes how the Mobility Application Access Control Policy is implemented. This includes a description of the objects that are subject to the policy, how the policy in configured, whether it can be changed or is hardwired, what the modes of access allow an application to do with a given object (e.g., write access allows a deletion of the object)*

Test

The open assignments for the list of objects, operations, and rules, make it impossible to list the set of tests the evaluator must run. In this case, the evaluator must include in their test plan the tests they intend to exercise given the completed operations of the SFRs in the ST. The overseeing scheme will then make a determination of the adequacy of the evaluator's proposed tests.

FDP_RIP.2 Full Residual Information Protection

FDP_RIP.2.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the [selection: allocation of the resource to, deallocation of the resource from] all objects.

Application Note:

102 *The selection is for the ST author to choose whether to make a resource unavailable on allocation or deallocation of the resource.*

103 *"Resources" in the context of this requirement are the abstractions that the mobile OS makes available to mobile applications. The concern here is a mobile application gaining access to data that is not intended for that application and is contrary to the Mobile Application SFP.*

Assurance Activity:

TSS
The evaluator examines the TSS to ensure it describes how the OS manages resources to prevent residual data being available to applications for which would otherwise not have access. Acceptable methods of ensuring residual data is unavailable could include zeroization of memory or buffer management schemes that ensure any data that was not zeroized was not accessible.

4.2.3 Identification and Authentication (FIA)

The device does not have to maintain the notion of an individual "user", therefore it is not necessary to have the device authenticate an identity. So many of the requirements in this section are used to express how a user is authorized to use the device. There are two authorization factors: the PIN, which unlocks the device, and the Password, which unwraps the credentials to allow an IPsec communication channel to be established.

FIA_AFL.1 Authentication Failure Handling

FIA_AFL.1.1 **Refinement:** The TSF shall detect when **an MDM configurable positive integer of consecutive** unsuccessful **authorization** attempts occur related to *entry of the PIN Authorization Factor.*

FIA_AFL.1.2 When the defined number of unsuccessful PIN authorization attempts has been [selection: met, surpassed], the TSF shall *lock the device for an MDM configurable amount of time* or [selection: no other action, wipe device persistent memory].

Application Note:

104 *This requirement defines how the TSF handles failed authorization attempts regarding a user attempting to unlock the phone. The first selection is used to define when the failed attempts limit is satisfied (when it is met or surpassed). If additionally wiping the device persistent memory is supported then that selection should be made, otherwise select "no other action."*

Assurance Activity:

105 *The evaluators shall check the TSS section to determine that it specifies how the failed attempts limit is satisfied. The evaluators shall perform the following tests:*

> *Test 1: The evaluator shall enter invalid PINs until the failed attempts limit is satisfied and verify that the device is locked.*
>
> *Test 2: The evaluator shall wait the MDM configured amount of time and verify that the time period is as configured and the device is unlocked.*
>
> *Test 3: [Conditional] If the ST author has selected the wipe device persistent memory, the evaluator shall enter invalid PINs until the failed attempts limit is satisfied. At this point the device's memory should be wiped. The evaluator then enters the correct PIN and should see that the device is no longer functional.*

FIA_PIN_EXT.1 Extended: PIN Management (PIN Authorization Factor)

FIA_PIN_EXT.1.1 The TSF shall provide the following PIN management capabilities:

1. PINs shall be able to be composed of any combination of numbers and [selection: no other characters, [assignment: other supported characters]];
2. Minimum PIN length shall be settable by the MDM and support PINs of [selection: 4 characters, [assignment: number of characters the device supports greater than 4]];
3. PINs shall have a maximum lifetime, configurable by the MDM.

Assurance Activity:

TSS

The evaluator examines the TSS to determine it describes how the PIN authorization factor is constructed and what it uses as the character set. It also describes how the minimum PIN length is configurable and what the maximum character length is.

Guidance

The evaluator ensures the guidance documentation describes how to configure the PIN length. The guidance documentation also describes how the maximum life time is configured and what the maximum lifetime that can be configured is.

Test

The evaluator tests to ensure that the minimum PIN length is enforced, by trying to create/change a PIN that is shorter than the required minimum length. The evaluator also tests to ensure the maximum lifetime is enforced.

FIA_PMG_EXT.1 Extended: Password Management (Credential Authorization Factor)

FIA_PMG_EXT.1.1 The TSF shall require the password be changed at an MDM configurable time period in a range of [assignment: *minimum period* to *maximum period that is at least 120 days*].

FIA_PMG _EXT.1.2 The TSF shall not allow the ten previous passwords to be reused.

Application Note:
The construction of the password itself is already covered by FCS_CKM.1. Here, the requirements simply mandate the frequency in which the password must be changed by the authorized user, and rules regarding password reuse.

FIA_UAU_EXT.6 Re-Authorizing

FIA_UAU_EXT.6.1: The TSF shall require the user to enter the correct PIN Authorization Factor when the user changes their PIN, following TSF-initiated locking (FTA_SSL_EXT.1), following user-initiated locking (FTA_SSL_EXT.2), [selection: [assignment: other conditions], no other conditions].

FIA_UAU_EXT.6.2: The TSF shall require the user to enter the correct Password Authorization Factor when changing to a new Password Authorization Factor.

Application Note:

106 *If the TOE is requiring the user to change authorization data upon having just provided the proper authorization factor (e.g., initial unlock, session unlock), the user is considered to be re-authorized.*

Assurance Activity:

107 *The evaluator shall perform the following test:*

> *Test 1: The evaluator shall attempt to change their PIN as directed by the operational guidance. While making this attempt, the evaluator shall verify that re-authorization is required.*
>
> *Test 2: The evaluator shall attempt to unlock a TSF-initiated session lock using an invalid PIN. The attempt shall fail.*

Test 3: The evaluator shall attempt to unlock a TSF-initiated session lock using a valid PIN. The attempt shall succeed.

Test 4: The evaluator shall attempt to unlock a user-initiated session lock using an invalid PIN. The attempt shall fail.

Test 5: The evaluator shall attempt to unlock a user-initiated session lock using a valid PIN. The attempt shall succeed.

Test 6: The evaluator shall attempt to change their Password Authorization Factor as directed by the operational guidance. While making this attempt, the evaluator shall verify that re-authorization is required.

FIA_UIA_EXT.1 Extended: User Identification and Authentication

FIA_UIA_EXT.1.1 The TSF shall allow emergency calls, [selection: text notification, voicemail notification, calendar notification, alert/alarm, no other services] on behalf of the user to be performed before the user has presented the authorization factor to unlock the device.
FIA_UIA_EXT.1.2 The TSF shall require each user to enter the correct authorization factor before allowing any other TSF-mediated actions on behalf of that user.

Application Note:

108 *This requirement applies to TOE provided services available before the correct authorization is provided to unlock the device. The ST author should consider the selection made in FTA_SSL_EXT.1.2. The same principles apply – notification information, rather than the data itself.*

Assurance Activity:

109 *The evaluator shall examine the TSS to verify that it describes how this requirement is implemented.*

110 *The evaluator shall verify that only the specified services in FIA_UIA_EXT.1.1 can be performed prior to user being authorized.*

111 *The evaluator shall verify that the TOE services not specified in FIA_UIA_EXT.1.1 can only be performed after successful authorization of the user.*

X509 Certificates (FIA_X509_EXT)

FIA_X509_EXT.1 Extended: X.509 Certificates

FIA_X509_EXT.1.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for IPsec connections.

Application Note:

It should be noted that RFC 5280 defines certificate validation and certification path validation requirements that must be implemented by the TOE as per this requirement.

FIA_X509_EXT.1.2 The TSF shall store and protect certificate(s) from unauthorized deletion and modification.

FIA_X509_EXT.1.3 The TSF shall provide the capability for the MDM to load X.509v3 certificates into the TOE for use by the security functions specified in this PP.

FIA_X509_EXT.1.4 The TSF shall validate the certificate using [selection: the Online Certificate Status Protocol (OCSP) as specified in RFC 2560, a Certificate Revocation List (CRL) as specified in RFC 5759].

FIA_X509_EXT.1.5 The TSF shall not establish an SA if a certificate is deemed invalid.

FIA_X509_EXT.1.6 The TSF shall not establish an SA if the distinguished name (DN) contained in a certificate does not match the expected DN for the entity attempting to establish a connection.

FIA_X509_EXT.1.7 When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall, at the option of the MDM, establish an SA or disallow the establishment of an SA.

Application Note:

The intent of FIA_X509_EXT.1.7 is that the TOE is configurable to allow or disallow session establishment if the TOE cannot connect to an entity responsible for providing certificate validation information. For instance, if a CRL cannot be obtained because a machine is down, or the network path is broken, the MDM may elect to configure the TOE to allow sessions to continued to be established, rather than terminate the TOE's ability to establish any new SAs because it cannot reach the CA.

Assurance Activity:

112 ***TSS***

113 *The evaluator shall ensure the TSS describes all certificate stores implemented that contain certificates used to meet the requirements of this PP. This description shall contain information pertaining to how certificates are loaded into storage, and how the storage is protected from unauthorized access.*

114 ***Guidance***

The evaluator shall examine the guidance documentation to ensure it describes how to configure either the TOE or the environment to prevent unauthorized modification or deletion of the certificates.

115 ***Test***

116 *The evaluator shall perform the following tests for each function in the system that requires the use of certificates:*

Test 1: The evaluator shall demonstrate that using a certificate without a valid certification path results in the function failing. The evaluator shall then load a certificate or certificates 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 function fails.

*The testing to ensure the requirements are satisfied is performed in conjunction with the IPsec requirement FCS_*IPSEC_EXT.1.12.

4.2.4 Security Management (FMT)

The management of the TOE is done initially as part of a provisioning process and is performed by an MDM. There are some management functions that may be done via the device connecting to the MDM using the IPsec connection to the enterprise. The ST author will make clear in their ST which functions can be performed by the MDM via the trusted connection to the enterprise. A separate Protection Profile specifies the requirements on the MDM.

FMT_MOF.1(1) Management of Security Functions Behavior (IPsec)

FMT_MOF.1.1(1) **Refinement:** The TSF shall restrict the ability to [selection: determine the behavior of, disable, enable, modify the behavior of] the IPsec protocol to *the MDM*.

Application Note:

117 *This requirement ensures that the ability to manage the TOEs IPsec configuration is restricted to the MDM.*

Assurance Activity:

118 ***Guidance***

119 *The evaluator shall examine the operational guidance to verify that it describes what and how the TOE's IPsec behavior is determined and modified. The evaluator shall verify the AGD guidance includes detailed and accurate instructions of how to determine and modify the TOE VPN behavior.*

120 ***Test***

121 *The evaluator shall test the TOE to ensure that determining and modifying the IPsec behavior is restricted to the MDM.*

FMT_MOF.1(2) Management of TSF Data (Mobile Applications)

FMT_MOF.1.1(2) **Refinement:** The TSF shall restrict the ability to *<u>install and update</u> mobile applications* to *the MDM*.

Application Note:

122 *This requirement ensures that the ability to install and update mobile applications is restricted to the MDM.*

Assurance Activity:

123 *The evaluator shall verify the AGD guidance includes detailed and accurate mobile application installation instructions. The evaluator shall verify that mobile application installation is restricted to the MDM.*

FMT_SMF.1 Specification of Management Functions

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

a. *ability to configure TOE cryptographic functions, including cryptographic protocols;*
b. *ability to query the current version of TOE and initiate software updates;*

c. *ability to query the current version of a mobile application and initiate an application update;*
d. *ability to install mobile applications;*
e. *ability to load X.509 certificates;*
f. *ability to configure the failed authentication attempt limit and the lockout time period;*
g. *[selection: no other functions, Mobility Application Access Control Policy, [assignment: other management functions provided by the TSF]].*

Application Note:

124 *This requirement identifies the capabilities the device must provide such that a MDM can manage the device. There are functions that an authorized user can perform, such as changing the PIN, or the password that is used to wrap the credentials required for establishment of the IPsec tunnel, but those are not considered management functions and are specified elsewhere.*

125 *For the selection, the ST author may add management capabilities that are not required, but due to implementation/design choices exist in the product. For example, the Mobility Application Access Control Policy does not mandate that the policy be configurable by the MDM. The platform may have a "hardcoded" policy regarding an applications access to resources.*

Assurance Activity:

126 *The evaluator shall examine the TSS to ensure that it describes each management function listed.*

127 *The evaluator shall verify the AGD guidance includes detailed instructions of what options are available and how to configure each TSF functional capability listed.*

128 *The evaluator shall verify the ability to configure each TSF functional capability listed via the MDM.*

4.2.5 Protection of the TSF (FPT)

FPT_KST_EXT.1 Extended: Key Storage

FPT_KST_EXT.1.1 No unencrypted key material used by the TOE shall be written to persistent memory.

FPT_KST_EXT.1.2 The TSF shall not store keys in plaintext on non-volatile storage.

***Assurance Activity**:*

129 *The evaluator shall check the TSS section to confirm that it describes how this requirement is implemented in the TOE and as specified in FCS_COP.1(1) (Key Masking).*

FPT_NOT_EXT.1 Extended: Event Notification

FPT_NOT_EXT.1.1 The TSF shall notify the user when the following types of failures occur:

- failures resulting from self tests performed per FPT_TST_EXT.1
- [selection: no other failures, [assignment: other failures]].

FPT_NOT_EXT.1.2 The TSF shall take the following actions:

- for failures specified in FPT_NOT_EXT.1.1, the TSF shall preserve a secure state and transition to non-operational mode,
- [selection: no other actions, [assignment: other actions]].

Application Note:

130 *This requirement lists what events trigger a notification to be sent to the user. In addition, the second element defines the actions to be taken by the TSF when a specified event occurs. A non-operational mode means the user is not able to unlock the credentials to establish a IPsec connection.*

Assurance Activity:

131 *The evaluator shall examine the TSS to ensure that it details the functionality to notify the device when the listed failures occur. The evaluator shall verify that the AGD guidance describes the actions that would be taken by the authorized user when such failures occur. The evaluator shall create a test that causes the self-test to fail and verify that the user is notified.*

Extended: Trusted Update (FPT_TUD_EXT TOE)

FPT_TUD_EXT.1(1) Extended: Trusted Update

FPT_TUD_EXT.1.1(1) The TSF shall provide authorized users the ability to query the current version of the TOE firmware/software.

FPT_TUD_EXT.1.2(1) The TSF shall provide authorized users the ability to initiate updates to TOE firmware/software.
FPT_TUD_EXT.1.3(1) The TSF shall verify software updates to the TOE using a digital signature mechanism implemented by the TOE prior to installing those updates.

FPT_TUD_EXT.1.4(1) The TSF shall only accept software updates signed by an authorized source.

Application Note:

132 *Since the only data communication path allowed is to the enterprise via the IPsec connection, the updates are assured to come from the enterprise. This requirement mandates that the updates are verified before they can be applied to the device. The digital signature mechanism referenced in the third element is the one specified in FCS_COP.1(2).*

Assurance Activity:

133 *Updates to the TOE software are signed by an authorized source. The definition of an authorized source is contained in the TSS, along with a description of how the certificates used by the update verification mechanism are contained on the device. The evaluator ensures this information is contained in the TSS. The evaluator also ensures that the TSS (or the operational guidance) describes how the candidate updates are obtained; the processing associated with verifying the digital signature of the updates; and the actions that take place for successful (signature was verified) and unsuccessful (signature could not be verified) cases. The location of the software/firmware that is performing the processing must also be described in the TSS and verified by the evaluators. The evaluators shall perform the following tests:*

Test 1: The evaluator performs the version verification activity to determine the current version of the product. The evaluator obtains a legitimate update using procedures described in the operational guidance and verifies that it is successfully installed on the TOE. Then, the evaluator performs a subset of other assurance activity tests to demonstrate that the update functions as expected. After the update, the evaluator performs the version verification activity again to verify the version correctly corresponds to that of the update.

Test 2: The evaluator performs the version verification activity to determine the current version of the product. The evaluator obtains or produces an illegitimate update, and attempts to install it on the TOE. The evaluator verifies that the TOE rejects the update.

Extended: Trusted Update (FPT_TUD_EXT Mobile Application)

FPT_TUD_EXT.1(2) Extended: Trusted Update

FPT_TUD_EXT.1.1(2) The TSF shall provide the MDM the ability to query the current version of the mobile applications.

FPT_TUD_EXT.1.2(2) The TSF shall provide the MDM the ability to initiate updates to the mobile application.

FPT_TUD_EXT.1.3(2) The TSF shall verify software updates to the mobile application using a digital signature mechanism prior to installing those updates.

FPT_TUD_EXT.1.4(2) The TSF shall only accept software updates signed by an authorized source.

Application Note:
As stated above, the only data path is via the IPsec connection, so the updates to applications will be coming from the enterprise. The notion of update, is left to the application developer and may be simply installing another version of the application, which would be done by the MDM. This requirement ensures that the original source (e.g., the vendor) has signed the application and it has not been modified. The digital signature mechanism referenced in the third element is the one specified in FCS_COP.1(2).

FPT_TST_EXT.1 Extended: TSF Testing

FPT_TST_EXT.1.1 The TSF shall run a suite of self tests during initial start-up (on power on) to demonstrate the correct operation of the TSF.

FPT_TST_EXT.1.2 The TSF shall provide the capability to verify the integrity of stored TSF executable code when it is loaded for execution through the use of the TSF-provided cryptographic service specified in FCS_COP.1(2).

Application Note:

134 *The TSF is expected to perform self tests to confirm correct operations of the hardware functions which the TOE depends upon for enforcing the SFRs, which includes the crypto-related functionality, including FCS_RBG_EXT.1 is implemented according to NIST SP 800-90A, the RBG health tests are consistent with*

section 11.3 of NIST SP 800-90A. In addition, the TOE uses cryptography to ensure the integrity of the software is maintained prior to it loading.

Assurance Activity:

135 *The TSS shall describe the known-answer self-tests for all tested cryptographic functions.*

136 *The evaluator shall verify that the TSS describes, for some set of non-cryptographic functions affecting the correct operation of the TSF, the method by which those functions are tested. The TSS will describe, for each of these functions, the method by which correct operation of the function/component is verified. The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly.*

137 *The evaluator shall determine that all of the identified functions/components are adequately tested on start-up.*

4.2.6 TOE Access (FTA)

FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1.1 The TSF shall lock an interactive session after [assignment: time interval of user inactivity] by clearing or overwriting display devices, making the current contents unreadable.

FTA_SSL_EXT.1.2 The TSF shall allow the following events to occur prior to unlocking the session: emergency calls, [selection: text notification, voicemail notification, calendar notification, alert/alarm, no other services].

Application Note:

138 *The assignment in the first element is used to specify the period of inactivity (i.e. the user is not browsing internet, reading emails, talking on the phone, etc. for a period of time) . The selection in FTA_SSL_EXT.1.2 should take into consideration the selection choices made in FIA_UIA_EXT.1.1. The intent is that there are certain things the device can do and present to the user without the user unlocking the device. These things are of a “notification” flavor, and do not provide detailed information (e.g., a text notification, rather than the text message itself).*

Assurance Activity:

139 *The evaluator shall examine the TSS section to confirm that it describes how this requirement is implemented in the TOE. The evaluator should ensure that if the TOE has an inactivity period timeout and only the allowed events can take place before the device is unlocked.*

FTA_SSL_EXT.2 User-initiated Locking

FTA_SSL_EXT.2.1 The TSF shall allow user-initiated locking of the user’s own interactive session, by clearing or overwriting display devices, making the current contents unreadable.

FTA_SSL_EXT.2.2 The TSF shall allow the following events to occur prior to unlocking the session: emergency calls, [selection: text notification, voicemail notification, calendar notification, alert/alarm, no other services].

Application Note:

140 *As with FTA_SSL_EXT.1.2, the selection is used to specify the actions that can be performed before the session can be unlocked.*

Assurance Activity:

The evaluator shall examine the TSS section to confirm that it describes how this requirement is implemented in the TOE. The evaluator should ensure that if the TOE has an inactivity period timeout and only the allowed events can take place before the device is unlocked.

4.2.7 Trusted Path/Channel (FTP)

FTP_ITC.1 Inter-TSF Trusted Channel (Protection from Modification or Disclosure)

FTP_ITC.1.1 **Refinement:** The TSF shall provide a **IPsec** communication channel between itself and **the enterprise VPN Gateway** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2 **Refinement:** The TSF shall permit the TSF, **or the enterprise VPN server** to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for *all incoming or outgoing data connections*.

Application Note:

141 *This requirement addresses the case where the TOE establishes communications with the enterprise VPN server. No communication will be attempted outside the enterprise VPN except data necessary to establish and maintain the VPN, and emergency calling. If a connection with the OEM (or any other service provider) is attempted and cannot be established, the mobile device will continue to function.*

Assurance Activity:

142 *The evaluator shall check the TSS section to confirm that it describes how this requirement is implemented in the TOE.*

143 *The evaluator shall verify that communication can be initiated from both the TSF and the enterprise VPN server. The evaluator shall attempt to establish communication outside the VPN and verify that this cannot be accomplished.*

4.3 Security Assurance Requirements

144 The Security Objectives for the TOE in Section 3 were constructed to address threats identified in Section 2. The Security Functional Requirements (SFRs) in Section 4.2 are a formal instantiation of the Security Objectives. The PP draws from EAL1 the Security Assurance Requirements (SARs) to frame the extent to which the evaluator assesses the documentation applicable for the evaluation and performs independent testing.

145 While this section contains the complete set of SARs from the CC, the Assurance Activities to be performed by an evaluator are detailed both in Section 4.2 as well as in this section.

146 The general model for evaluation of TOEs against STs written to conform to this PP is as follows:

147 After the ST has been approved for evaluation, the Common Criteria Testing Laboratory (CCTL) will obtain the TOE, supporting environmental IT, and the administrative guides for the TOE. The Assurance Activities listed in the ST (which will be refined by the CCTL to be TOE-specific, either within the ST or in a separate document) will then be performed by the CCTL. The CCTL is also expected to perform all of the actions mandated by the Common Evaluation Methodology (CEM) for EAL1. The results of these activities will be documented and presented (along with the administrative guidance used) for validation.

148 For each family, “Developer Notes” are provided on the developer action elements to clarify what, if any, additional documentation/activity needs to be provided by the developer. For the content/presentation and evaluator activity elements, additional assurance activities (to those already contained in Section 4.2 and the CEM for EAL1) are described as a whole for the family, rather than for each element. Additionally, the assurance activities described in this section are complementary to those specified in Section 4.2.

149 The TOE security assurance requirements, summarized in Table 2, identify the management and evaluative activities required to address the threats identified in Section 2 of this PP.

Table 1: TOE Security Assurance Requirements

Assurance Class	Assurance Components	Assurance Components Description
Development	ADV_FSP.1	Basic Functional Specification
Guidance Documents	AGD_OPE.1	Operational User Guidance
	AGD_PRE.1	Preparative User Guidance
Tests	ATE_IND.1	Independent Testing - Conformance

Assurance Class	Assurance Components	Assurance Components Description
Vulnerability Assessment	AVA_VAN.1	Vulnerability Analysis
Life Cycle Support	ALC_CMC.1	Labeling of the TOE
	ALC_CMS.1	TOE CM Coverage

4.3.1 Class ADV: Development

150 At EAL1, the information about the TOE is contained in the guidance documentation available to the end user as well as the TOE Summary Specification (TSS) portion of the ST. While it is not required that the TOE developer write the TSS, the TOE developer must concur with the description of the product that is contained in the TSS as it relates to the functional requirements. The Assurance Activities contained in Section 4.2 should provide the ST authors with sufficient information to determine the appropriate content for the TSS section.

4.3.1.1 ADV_FSP.1 Basic functional specification

151 The functional specification describes the TOE Security Functionality Interfaces (TSFIs). At EAL1, it is not necessary to have a formal or complete specification of these interfaces. Additionally, because TOEs conforming to this PP will necessarily have interfaces to the Operational Environment that are not directly invokable by TOE users, at EAL1 there is little point specifying that such interfaces be described in and of themselves since only indirect testing of such interfaces may be possible. For this PP, the activities for this family should focus on understanding the interfaces presented in the TSS in response to the functional requirements and the interfaces presented in the AGD documentation. No additional "functional specification" documentation is necessary to satisfy the assurance activities specified.

152 The interfaces that need to be evaluated are characterized through the information needed to perform the assurance activities listed, rather than as an independent, abstract list.

Developer action elements:

ADV_FSP.1.1D The developer shall provide a functional specification.

ADV_FSP.1.2D The developer shall provide a tracing from the functional specification to the SFRs.

Developer Note: As indicated in the introduction to this section, the functional specification is comprised of the information contained in the AGD_OPR and AGD_PRE documentation, coupled with the information provided in the TSS of the ST. The assurance activities in the functional requirements point to evidence that should exist in the documentation and TSS section; since these are directly associated with the SFRs, the tracing in element ADV_FSP.1.2D is implicitly already done and no additional documentation is necessary.

Content and presentation elements:

ADV_FSP.1.1C The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.2C The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.3C The functional specification shall provide rationale for the implicit categorization of interfaces as SFR-non-interfering.

ADV_FSP.1.4C The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

Evaluator action elements:

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

ADV_ FSP.1.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

Assurance Activity:

153 *There are no specific assurance activities associated with these SARs. The functional specification documentation is provided to support the evaluation activities described in Section 4.2, and other activities described for AGD, ATE, and AVA SARs. The requirements on the content of the functional specification information is implicitly assessed by virtue of the other assurance activities being performed; if the evaluator is unable to perform an activity because the there is insufficient interface information, then an adequate functional specification has not been provided.*

4.3.2 Class AGD: Guidance Documents

154 The guidance documents will be provided with the developer’s security target. Guidance must include a description of how the authorized user verifies that the Operational Environment can fulfill its role for the security functionality. The documentation should be in an informal style and readable by an authorized user.

155 Guidance must be provided for every operational environment that the product supports as claimed in the ST. This guidance includes

- instructions to successfully install the TOE in that environment; and
- instructions to manage the security of the TOE as a product and as a component of the larger operational environment.

156 Guidance pertaining to particular security functionality is also provided; specific requirements on such guidance are contained in the assurance activities specified in Section 4.2.

4.3.2.1 AGD_OPE.1 Operational User Guidance

Developer action elements:

AGD_OPE.1.1D The developer shall provide operational user guidance.

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

Content and presentation elements:

AGD_OPE.1.1C The operational user guidance shall describe what the authorized user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2C The operational user guidance shall describe, for the authorized user, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C The operational user guidance shall describe, for the authorized user, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4C The operational user guidance shall, for the authorized user, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6C The operational user guidance shall, for the authorized user, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C The operational user guidance shall be clear and reasonable.

Evaluator action elements:

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

Assurance Activity:

157 *Some of the contents of the operational guidance will be verified by the assurance activities in Section 4.2 and evaluation of the TOE according to the CEM. The following additional information is also required.*

158 *The operational guidance shall at a minimum list the processes running (or that could run) on the TOE in its evaluated configuration during its operation that are capable of processing data received on the network interfaces (there are likely more than one of these, and this is not limited to the process that "listens" on the network interface). It is acceptable to list all processes running (or that could run) on the TOE in its evaluated configuration instead of attempting to determine just those that process the network data. For each process listed, the administrative guidance will contain a short (e.g., one- or two-line) description of the process' function, and the privilege with which the service runs. "Privilege" includes the hardware privilege level (e.g., ring 0, ring 1), any software privileges specifically associated with the process, and the privileges associated with the user role the process runs as or under.*

159 *The operational guidance shall contain instructions for configuring the cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE.*

160 *The documentation must describe the process for verifying updates to the TOE, either by checking the hash or by verifying a digital signature. The evaluator shall verify that this process includes the following steps:*

1. *For hashes, a description of where the hash for a given update can be obtained. For digital signatures, instructions for obtaining the certificate that will be used by the FCS_COP.1(2) mechanism to ensure that a signed update has been received from the certificate owner. This may be supplied with the product initially, or may be obtained by some other means.*
2. *Instructions for obtaining the update itself. This should include instructions for making the update accessible to the TOE (e.g., placement in a specific directory).*
3. *Instructions for initiating the update process, as well as discerning whether the process was successful or unsuccessful. This includes generation of the hash/digital signature.*

4.3.2.2 AGD_PRE.1 Preparative procedures

Developer action elements:

AGD_PRE.1.1D The developer shall provide the TOE including its preparative procedures.

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

Content and presentation elements:

AGD_PRE.1.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2C The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

Evaluator action elements:

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

AGD_PRE.1.2E The evaluator *shall apply* the preparative procedures to confirm that the TOE can be prepared securely for operation.

Assurance Activity:

161 *As indicated in the introduction above, there are significant expectations with respect to the documentation—especially when configuring the operational environment to support TOE functional requirements. The evaluator shall check to ensure that the guidance provided for the TOE adequately addresses all platforms (that is, combination of hardware and operating system) claimed for the TOE in the ST.*

4.3.3 Class ATE: Tests

162 Testing is specified for functional aspects of the system as well as aspects that take advantage of design or implementation weaknesses. The former is done through ATE_IND family, while the latter is through the AVA_VAN family. At the assurance level specified in this PP, testing is based on advertised functionality and interfaces with dependency on the availability of design information. One of the primary outputs of the evaluation process is the test report as specified in the following requirements.

4.3.3.1 ATE_IND.1 Independent testing - Conformance

163 Testing is performed to confirm the functionality described in the TSS as well as the administrative (including configuration and operation) documentation provided. The focus of the testing is to confirm that the requirements specified in Section 4.2 are being met, although some additional testing is specified for SARs in Section 4.3. The Assurance Activities identify the additional testing activities associated with these components. The evaluator produces a test report documenting the plan for and results of testing, as well as coverage arguments focused on the platform/TOE combinations that are claiming conformance to this PP.

Developer action elements:

ATE_IND.1.1D The developer shall provide the TOE for testing.

Content and presentation elements:

ATE_IND.1.1C The TOE shall be suitable for testing.

Evaluator action elements:

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

ATE_IND.1.2E The evaluator *shall test* a subset of the TSF to confirm that the TSF operates as specified.

Assurance Activity:

164 *The evaluator shall prepare a test plan and report documenting the testing aspects of the system. The test plan covers all of the testing actions contained in the CEM and the body of this PP's Assurance Activities. While it is not necessary to have one test case per test listed in an Assurance Activity, the evaluators must document in the test plan that each applicable testing requirement in the ST is covered.*

165 *The Test Plan identifies the platforms to be tested, and for those platforms not included in the test plan but included in the ST, the test plan provides a justification for not testing the platforms. This justification must address the differences between the tested platforms and the untested platforms, and make an argument that the differences do not affect the testing to be performed. It is not sufficient to merely assert that the differences have no affect; rationale must be provided. If all platforms claimed in the ST are tested, then no rationale is necessary.*

166 *The test plan describes the composition of each platform to be tested, and any setup that is necessary beyond what is contained in the AGD documentation. It should be noted that the evaluators are expected to follow the AGD documentation for installation and setup of each platform either as part of a test or as a standard pre-test condition. This may include special test drivers or tools. For each driver or tool, an argument (not just an assertion) is provided that the driver or tool will not adversely affect the performance of the functionality by the TOE and its platform. This also includes the configuration of the cryptographic engine to be used. The cryptographic algorithms implemented by this engine are those specified by this PP and used by the cryptographic protocols being evaluated (IPsec).*

167 *The test plan identifies high-level test objectives as well as the test procedures to be followed to achieve those objectives. These procedures include expected results. The test report (which could just be an annotated version of the test plan) details the activities that took place when the test procedures were executed, and includes the actual results of the tests. This shall be a cumulative account, so if there was a test run that resulted in a failure; a fix installed; and then a successful re-run of the test, the report would show a "fail" and "pass" result (and the supporting details), and not just the "pass" result.*

4.3.4 Class AVA: Vulnerability assessment

168 For the first generation of this protection profile, the evaluation lab is expected to survey open sources to learn what vulnerabilities have been discovered in these types of products. In most cases, these vulnerabilities will require sophistication beyond that of a basic attacker. Until penetration tools are created and uniformly distributed to the evaluation labs, evaluators will not be expected to test for these vulnerabilities in the TOE. The labs will be expected to comment on the likelihood of these

vulnerabilities given the documentation provided by the vendor. This information will be used in the development of penetration testing tools and for the development of future protection profiles.

4.3.4.1 AVA_VAN.1 Vulnerability survey

Developer action elements:

AVA_VAN.1.1D The developer shall provide the TOE for testing.

Content and presentation elements:

AVA_VAN.1.1C The TOE shall be suitable for testing.

Evaluator action elements:

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

AVA_VAN.1.2E The evaluator *shall perform* a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.1.3E The evaluator *shall conduct* penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

Assurance Activity:

169 *As with ATE_IND the evaluator shall generate a report to document their findings with respect to this requirement. This report could physically be part of the overall test report mentioned in ATE_IND, or a separate document. The evaluator performs a search of public information to determine the vulnerabilities that have been found in Mobility Operating Systems in general, as well as those that pertain to the particular TOE. The evaluator documents the sources consulted and the vulnerabilities found in the report. For each vulnerability found, the evaluator either provides a rationale with respect to its non-applicability or the evaluator formulates a test (using the guidelines provided in ATE_IND) to confirm the vulnerability, if suitable. Suitability is determined by assessing the attack vector needed to take advantage of the vulnerability. For example, if the vulnerability can be detected by pressing a key combination on boot-up, a test would be suitable at the assurance level of this PP. If exploiting the vulnerability requires expert skills and an electron microscope, for instance, then a test would not be suitable and an appropriate justification would be formulated.*

4.3.5 Class ALC: Life-cycle support

170 At the assurance level provided for TOEs conformant to this PP, life-cycle support is limited to end-user-visible aspects of the life-cycle, rather than an examination of the TOE vendor’s development and configuration management process. This is not meant to diminish the critical role that a developer’s practices play in contributing to the overall trustworthiness of a product; rather, it’s a reflection on the information to be made available for evaluation at this assurance level.

4.3.5.1 ALC_CMC.1 Labeling of the TOE

171 This component is targeted at identifying the TOE such that it can be distinguished from other products or version from the same vendor and can be easily specified when being procured by an end user.

Developer action elements:

ALC_CMC.1.1D The developer shall provide the TOE and a reference for the TOE.

Content and presentation elements:

ALC_CMC.1.1C The TOE shall be labeled with its unique reference.

Evaluator action elements:

ALC_CMC.2.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

Assurance Activity:

172 The evaluator shall check the ST to ensure that it contains an identifier (such as a product name/version number) that specifically identifies the version that meets the requirements of the ST. Further, the evaluator shall check the AGD guidance and TOE samples received for testing to ensure that the version number is consistent with that in the ST. If the vendor maintains a web site advertising the TOE, the evaluator shall examine the information on the web site to ensure that the information in the ST is sufficient to distinguish the product.

4.3.5.2 ALC_CMS.1 TOE CM coverage

173 Given the scope of the TOE and its associated evaluation evidence requirements, this component's assurance activities are covered by the assurance activities listed for ALC_CMC.1.

Developer action elements:

ALC_CMS.2.1D The developer shall provide a configuration list for the TOE.

Content and presentation elements:

ALC_CMS.2.1C The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.

ALC_CMS.2.2C The configuration list shall uniquely identify the configuration items.

Evaluator action elements:

ALC_CMS.2.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

Assurance Activity:

174 *The "evaluation evidence required by the SARs" in this PP is limited to the information in the ST coupled with the guidance provided to administrators and users under the AGD requirements. By ensuring that the TOE is specifically identified and that this identification is consistent in the ST and in the AGD guidance (as done in the assurance activity for ALC_CMC.1), the evaluator implicitly confirms the information required by this component.*

RATIONALE

175 The rationale tracing the threats to the objectives and the objectives to the requirements is contained in the prose in Sections 2.0 and 3.0. The only outstanding mappings are those for the Assumptions and Organizational Security Policies; those are contained in Annex A below.

ANNEX A: SUPPORTING TABLES

176 In this Protection Profile, the focus in the initial sections of the document is to use a narrative presentation in an attempt to increase the overall understandability of the threats to mobile operating systems; the methods used to mitigate those threats; and the extent of the mitigation achieved by compliant TOEs. This presentation style does not readily lend itself to a formalized evaluation activity, so this Annex contains the tabular artifacts that can be used for the evaluation activities associated with this document.

Assumptions

177 The specific conditions listed in the following subsections are assumed to exist in the TOE's Operational Environment. These assumptions include both practical realities in the development of the TOE security requirements and the essential environmental conditions on the use of the TOE.

178 ST authors should ensure that the assumptions still hold for their particular technology; the table should be modified as appropriate.

Table 2: TOE Assumptions

Assumption Name	Assumption Name
A.AUTHORIZED_USER	An authorized user of the TOE is well-trained, not actively working against the protection of the data, and will follow all provided guidance.
A.ENTERPRISE	The enterprise is responsible for signing applications and configuring the mobile device before issue to the user. The enterprise is also responsible for signing all OS and application updates and pushing the updates to the device.
A.UNSECURE_COMM	Wifi and Bluetooth capability are disabled (turned off) and will not be activated by an authorized user.
A.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

Threats

179 The following threats should be integrated into the threats that are specific to the technology by the ST authors when including the requirements described in this document. Modifications, omissions, and additions to the requirements may impact this list, so the PP author should modify or delete these threats as appropriate.

Table 3: Threats

Threat	Description of Threat
T.INSECURE_WORKSPACE	Malicious hardware or applications may exfiltrate sensitive data from the TOE.
T.USER_DATA_REUSE	Residual user data may be inadvertently sent to a destination not intended by the original sender.
T.TSF_FAILURE	Security mechanisms of the TOE may fail, leading to a compromise of the TSF.
T.UNAUTHORIZED_ACCESS	A user may gain unauthorized access to the TOE data and TOE executable code. A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources. A malicious user, process, or external IT entity may misrepresent itself as the TOE to obtain identification and authentication data.
T.UNAUTHORIZED_UPDATE	A malicious party attempts to supply the end user with an update to the product that may compromise the security features of the TOE.
T. INSTALL_MALICIOUS_SOFTWARE	Malicious software installed onto device may attempt to extract and/or alternate user's data.

Security Objectives for the TOE

Table 4: Security Objectives for the TOE

Objective	Objective Description
O.CLEAR_RESIDUAL_DATA	The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated.
O.DISCRETIONARY_ACCESS	The TSF must control access of subjects and/or users to named resources based on identity of the object. The TSF must allow authorized users to specify for each access mode which users/subjects are allowed to access a specific named object in that access mode.

O.SECURE_STORAGE	The TOE shall provide protected storage of key data stored on the TOE. If FCS_CKM.1(3) and FCS_COP.1(5) from Appendix C are included by the PP/ST author, this Objective shall apply to user data stored on the TOE.
O.SECURE_WORKSPACE	The TOE shall be separated from the hardware on the mobile device such that a secure workspace is maintained.
O.TOE_ADMINISTRATION	The TOE will provide mechanisms to ensure that only administrators are able to log in and configure the TOE, and provide protections for logged-in administrators.
O.TRUSTED_CHANNEL	The TSF must be designed and implemented in a manner that allows for establishing a trusted channel between the TOE and a remote trusted IT system that protects the user data and TSF data transferred over this channel from disclosure and undetected modification and prevents masquerading of the remote trusted IT system.
O.TSF_SELF_TEST	The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly.
O.VERIFIABLE_UPDATES	The TOE will provide the capability to help ensure that any updates to the TOE can be verified by the administrator to be unaltered and from a trusted source.

180 The following table contains objectives for the Operational Environment. As assumptions are added to the PP, these objectives should be augmented to reflect such additions.

Table 5: Security Objectives for the Operational Environment

Objective	Objective Description
OE.AUTHORIZED_USERS	Users of the phone are trained to securely use the phone and apply all guidance in a trusted manner.
OE.ENTERPRISE	The enterprise provides the administration services for signing of applications and update patches and configuration and update of the mobile device.
OE.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

ANNEX B: NIST SP 800-53/CNSS 1253 MAPPING

181 Several of the NIST SP 800-53/CNSS 1253 controls are either fully or partially addressed by compliant TOEs. This section outlines the requirements that are addressed, and can be used by certification personnel to determine what, if any, additional testing is required when the TOE is incorporated into its operational configuration.

182 *Application Note: In this version, only a simple mapping is provided. In future versions, additional narrative will be included that will provide further information for the certification team. This additional information will include details regarding the SFR to control mapping discussing what degree of compliance is provided by the TOE (e.g., fully satisfies the control, partially satisfies the control). In addition, a comprehensive review of the specified assurance activities, and those evaluation activities that occur as part of satisfying the SARs will be summarized to provide the certification team information regarding how compliance was determined (e.g., document review, vendor assertion, degree of testing/verification). This information will indicate to the certification team what, if any, additional activities they need to perform to determine the degree of compliance to specified controls.*

183 *Since the ST will make choices as far as selections, and will be filling in assignments, a final story cannot necessarily be made until the ST is complete and evaluated. Therefore, this information should be included in the ST in addition to the PP. Additionally, there may be some necessary interpretation (e.g.,"modification") to the activities performed by the evaluator based on a specific implementation. The scheme could have the oversight personnel (e.g., Validators) fill in this type of information, or could have this done by the evaluator as part of the assurance activities. The verification activities are a critical piece of information that must be provided so the certification team can determine what, if anything, they need to do in addition to the work of the evaluation team*.

Identifier	Name	Applicable SFRs
AC-3	Access Enforcement	FDP_ACF.1(*), FMT_MTD.1(*)
AC-4	Information Flow Enforcement	FDP_IFC.2, FDP_IFF.1
AC-6	Least Privilege	FDP_ACC.1(*), FMT_MOF.1(*), FMT_MTD.1(*), FMT_SMR.1
AC-7	Unsuccessful Login Attempts	FIA_AFL.1
AC-11	Session Lock	FTA_SSL.1 , FTA_SSL.2
AC-14	Permitted Actions without Identification or Authentication	FIA_UIA_EXT.1
AU-10	Non-Repudiation	FCS_COP.1(2)
CM-5	Access Restrictions for Change	FMT_MOF.1(*), FMT_MTD.1(*), FTP_TUD_EXT.1
CM-6	Configuration Settings	FMT_SMF.1
IA-2	Identification and Authentication	FIA_UAU_EXT.5
IA-5	Authenticator Management	FCS_PPH_EXT.1, FIA_PMG_EXT.1. FIA_UAU.6
MP-4	Media Protection	FTP_KST_EXT.1

SC-4	Information in Shared Resources	FDP_RIP.2
SC-8	Transmission Integrity	FCS_IPSEC_EXT.1, FTP_ITC.1, FCS_COP.1(2), FCS_COP.1(4)
SC-9	Transmission Confidentiality	FCS_IPSEC_EXT.1, FTP_ITC.1, FCS_COP.1(4)
SC-12	Cryptographic Key Establishment and Management	FCS_CKM.1(*), FCS_CKM_EXT.2 FCS_CKM_EXT.4, FCS_IPSEC_EXT.1, FCS_COP.1(4)
SC-13	Use of Cryptography	FCS_COP.1(*), FCS_RBG_EXT.1, FCS_IPSEC_EXT.1, FPT_TUD_EXT.1, FCS_COP.1(4)
SI-3	Malicious Code Protection	FPT_TUD_EXT.1
SI-6	Security Functionality Verification	FPT_NOT_EXT.1, FPT_TST_EXT.1

ANNEX C: ADDITIONAL REQUIREMENTS

184 For this draft of the PP, this appendix contains additional components without supporting threats, objectives, rationale, or assurance activities (although some guidance is given for selected components). In tandem with the current review cycle, this supporting information will be developed and incorporated into the next release of the PP. Comments on the information contained in this section (both on whether the requirements contained are applicable to the potential conformant TOEs as well as requirements that are not contained in this appendix that are widely applicable Mobile OS products) are welcome and solicited.

185 As indicated in the introduction to this PP, there are several capabilities that a TOE may implement and still be conformant to this PP. These capabilities are not required, creating a dependency on the Operational Environment. However, if a TOE does implement such capabilities, the ST author will take the following information and include it in their ST.

186 The focus of this PP is secure communications with limited data protection functionality. If the TOE provides user data encryption, the following requirements must be included in the ST.

187 Note that minor adjustments to the narrative information in the beginning of the ST may be required depending on the selections performed.

C.1 Requirements

FCS_CKM.1(3) Cryptographic key generation (DEK)

FCS_CKM.1.1(3) **Refinement:** The TSF shall generate **DEK** cryptographic keys **using a Random Bit Generator as specified in FCS_RBG_EXT.1 and** specified cryptographic key sizes **[selection: 128 bit, 256 bit]** that meet the following: [**No Standard]**.

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

188 ***Assurance Activity:***

189 *The evaluator shall review the TSS to determine that it describes how the functionality described by FCS_RBG_EXT.1 is invoked. If the RBG is provided by the Operational Environment, then the evaluator checks to ensure that--for each platform identified in the ST--the TSS describes the interface used by the TOE to invoke this functionality. The evaluator uses the description of the RBG functionality in FCS_RBG_EXT or documentation available for the operational environment to determine that the key size being requested is identical to the key size and mode to be used for the encryption/decryption of the user data.*

FCS_COP.1(5) Cryptographic Operation (Data Encryption and Decryption)

FCS_COP.1.1(5) The TSF shall perform *data encryption and decryption* in accordance with a specified cryptographic algorithm *NIST-approved implementation of AES used in [selection: CBC, CCM, CFB, CTR, GCM, OFB, XTS] mode* and cryptographic key sizes *[selection: 128 bits, 256 bits]* that meet the following: *FIPS PUB 197, "Advanced Encryption Standard (AES)" and [selection: NIST SP 800-38A, NIST SP 800-38C, NIST SP 800-38D, NIST SP 800-38E]***.**

Application Notes:

190 *This version of the PP is focused on VoIP only and no data remains that must be protected. This will change in the future and when data protected becomes necessary, data encryption will be required.*

191 *The intent of this requirement is to specify the approved AES modes that the ST author may select for AES encryption of the appropriate information on the file encryption software. For the first selection, the ST author should indicate the mode or modes supported by the TOE implementation. The second selection indicates the key size to be used, which is identical to that specified for FCS_CKM.1(1). The third selection must agree with the mode or modes chosen in the first selection. If multiple modes are supported, it may be clearer in the ST if this component was iterated.*

192 *The overall algorithm strength shall be used consistently. For example, if AES-256 is used, then SHA-384 shall be used, not SHA-256.*

193 *Future versions of this PP may include new cryptographic modes as they are reviewed and approved by NIST.*

Assurance Activity:

194 *If multiple modes are supported, the evaluator examines the TSS and guidance documentation to determine how a specific mode/key-size is chosen by the end user. The evaluator then tests each mode/key size combination in the manner found in the following sections, as appropriate. Note that some of these tests will require a reference implementation of the algorithms that is acceptable to the evaluation facility's Scheme.*

CBC Mode

195 *The CBC mode tests reference The Advanced Encryption Standard Algorithm Validation Suite (AESAVS) [AESAVS], available from http://csrc.nist.gov/groups/STM/cavp/documents/aes/AESAVS.pdf.*

196 *The evaluator shall examine the TSS to confirm that it describes how the IV values for this mode are derived/obtained, and ensure that the described implementation satisfies the required property that the IVs are unpredictable.*

197 *The evaluators shall run a set of known answer tests for each key size supported by the TSF. Inputs are a key, IV, and either plaintext to be encrypted or ciphertext to be decrypted. All of the test vectors (both encrypt and decrypt) for CBC mode in the supported key lengths from http://csrc.nist.gov/groups/STM/cavp/documents/aes/KAT_AES.zip shall be used to perform these tests*

198 *The evaluators shall perform a multi-block message test for each key length supported. To perform this test, the evaluators generated 10 data sets for encryption and 10 data sets for decryption. Each data set consists of key, an IV, and plaintext (for encryption) or ciphertext (for decryption). The length of a block*

*shall be 128 bits; the length of the plaintext/ ciphertext shall be block length * i, where i indicates the data set number and i ranges from 1 to 10 (so messages will range from 128 bits to 1280 bits).*

199 *The evaluators shall perform a Monte Carlo test. The evaluators shall generate 10 sets of starting values for encryption (values for the key, IV, and plaintext) and 10 sets of starting values for decryption (values for the key, IV, and ciphertext). The length of the plaintext/ciphertext shall be 128 bits. Each set of starting values is used to generate and perform 100 tests; the algorithm for generating the 100 test values (per set of starting values) is contained in section 6.4.2 of [AESAVS].*

CCM Mode

200 *The CCM mode tests reference The CCM Validation System (CCMVS) [CCMVS], available from http://csrc.nist.gov/groups/STM/cavp/documents/mac/CCMVS.pdf.*

201 *The evaluators shall examine the TSS to ensure the lengths for the payload, associated data, nonce, and tags (as well as the key length) are specified. These values shall be used in constructing the tests described in the next section. If multiple values are supported, then the evaluator shall examine the operational guidance to determine how the values are selected by the user.*

202 *The evaluator shall perform the following five tests for each key length supported by the file encryption software.*

203 *The evaluator shall perform a variable associated data test. For each associated data length supported, the evaluators shall devise 10 sets of input data. Each set of input data shall use the same key and nonce, and have the same tag (MAC) length. For each of the 10 sets, a unique string of associated data and payload data shall be used. The evaluators shall calculate the correct ciphertext for the inputs, and then ensure that the TSF calculates the same value for all input sets for all supported associated data lengths. An example of the input sets (for a 256-bit key) can be found in the VADT256.txt file from the archive http://csrc.nist.gov/groups/STM/cavp/documents/mac/ccmtestvectors.zip.*

204 *The evaluator shall perform a variable payload test. For each payload length supported, the evaluators shall devise 10 sets of input data. Each set of input data shall use the same key and nonce, and have the same tag (MAC) length. For each of the 10 sets, a unique string of associated data and payload data shall be used. The evaluators shall calculate the correct ciphertext for the inputs, and then ensure that the TSF calculates the same value for all input sets for all supported payload lengths. An example of the input sets (for a 256-bit key) can be found in the VPT256.txt file from the archive http://csrc.nist.gov/groups/STM/cavp/documents/mac/ccmtestvectors.zip.*

205 *The evaluator shall perform a variable nonce test. For each nonce length supported, the evaluators shall devise 10 sets of input data. Each set of input data shall use the same key and have the same tag (MAC) length. For each of the 10 sets, a unique nonce and unique strings of associated data and payload data shall be used. The evaluators shall calculate the correct ciphertext for the inputs, and then ensure that the TSF calculates the same value for all input sets for all supported nonce lengths. An example of the input sets (for a 256-bit key) can be found in the VNT256.txt file from the archive http://csrc.nist.gov/groups/STM/cavp/documents/mac/ccmtestvectors.zip.*

206 *The evaluator shall perform a variable tag test. For each tag length supported, the evaluators shall devise 10 sets of input data. Each set of input data shall use the same key and nonce. For each of the 10 sets, a unique string of associated data and payload data shall be used. The evaluators shall calculate the correct ciphertext for the inputs, and then ensure that the TSF calculates the same value for all input sets for all supported tag lengths. An example of the input sets (for a 256-bit key) can be found in the*

VTT256.txt file from the archive http://csrc.nist.gov/groups/STM/cavp/documents/mac/ccmtestvectors.zip.

207 *The final test the evaluators shall perform is the decryption-verification process test. This test is performed for each combination of associated data length, payload length, nonce length, and tag length supported by the TSF. For each combination, 15 sets of input data are provided to the TSF. The input data consists of a key, associated data, payload data, nonce, and ciphertext. The evaluators should ensure that between 1/3 and 2/3 of the ciphertext values do not pass the MAC check for a variety of error types. The inputs are supplied to the TSF and the evaluators verify that the TSF correctly identifies erroneous MAC values as well as passing values. An example of the input sets (for a 256-bit key) can be found in the VTT256.txt file from the archive http://csrc.nist.gov/groups/STM/cavp/documents/mac/ccmtestvectors.zip.*

CTR Mode

208 *The CTR mode tests reference The Advanced Encryption Standard Algorithm Validation Suite (AESAVS) [AESAVS], available from http://csrc.nist.gov/groups/STM/cavp/documents/aes/AESAVS.pdf.*

209 *The evaluator shall examine the TSS to confirm that it describes how the counter values for this mode are derived/obtained, and ensure that the described implementation satisfies the required property that each counter value is associated with only one plaintext block that is ever encrypted with a given key.*

210 *The evaluator shall run a set of known answer tests for each key size supported by the TSF. Inputs are a key, IV, and either plaintext to be encrypted or ciphertext to be decrypted. All of the test vectors (both encrypt and decrypt) for CTR mode in the supported key lengths from http://csrc.nist.gov/groups/STM/cavp/documents/aes/KAT_AES.zip shall be used to perform these tests.*

211 *The evaluator shall perform a multi-block message test for each key length supported. To perform this test, the evaluator shall generate 10 data sets for encryption and 10 data sets for decryption. Each data set consists of key, an IV, and plaintext (for encryption) or ciphertext (for decryption). The length of a block shall be 128 bits; the length of the plaintext/ ciphertext shall be block length * i, where i indicates the data set number and i ranges from 1 to 10 (so messages will range from 128 bits to 1280 bits).*

212 *The evaluator shall perform a Monte Carlo test. The evaluator shall generate 10 sets of starting values for encryption (values for the key, IV, and plaintext) and 10 sets of starting values for decryption (values for the key, IV, and ciphertext). The length of the plaintext/ciphertext shall be 128 bits. Each set of starting values is used to generate and perform 100 tests; the algorithm for generating the 100 test values (per set of starting values) is contained in section 6.4.1 of [AESAVS].*[4]

FMT_MOF.1(x) Management of Security Functions Behavior (SFP)
This requirement is considered optional, since it is not mandated that the access control policy be configurable or is a management function. If a device provides this capability, then this requirement would be added to the body of the ST.

[4] CTR mode uses the ECB Monte Carlo Algorithm.

FMT_MOF.1.1(x) **Refinement:** The TSF shall restrict the ability to determine the behavior of and modify the behavior of the *SFP specified in FDP_ACC.1* to *the MDM*.

.

Assurance Activity:

213 *The evaluator shall examine the TSS to verify that it describes what and how the SFP behavior is determined and modified. The evaluator shall verify the AGD guidance includes detailed and accurate instructions of how to determine and modify the SFP behavior, and how the MDM would interact with the device.*

The evaluator shall verify that determining and modifying the SFP behavior is restricted to the MDM.

ANNEX D: ENTROPY DOCUMENTATION AND ASSESSMENT

The documentation of the entropy source should be detailed enough that, after reading, the evaluator will thoroughly understand the entropy source and why it can be relied upon to provide entropy. This documentation should include multiple detailed sections: design description, entropy justification, operating conditions, and health testing. This documentation is not required to be part of the TSS.

Design Description

Documentation shall include the design of the entropy source as a whole, including the interaction of all entropy source components. It will describe the operation of the entropy source to include how it works, how entropy is produced, and how unprocessed (raw) data can be obtained from within the entropy source for testing purposes. The documentation should walk through the entropy source design indicating where the random comes from, where it is passed next, any post-processing of the raw outputs (hash, XOR, etc.), if/where it is stored, and finally, how it is output from the entropy source. Any conditions placed on the process (e.g., blocking) should also be described in the entropy source design. Diagrams and examples are encouraged.

This design must also include a description of the content of the security boundary of the entropy source and a description of how the security boundary ensures that an adversary outside the boundary cannot affect the entropy rate.

Entropy Justification

There should be a technical argument for where the unpredictability in the source comes from and why there is confidence in the entropy source exhibiting probabilistic behavior (an explanation of the probability distribution and justification for that distribution given the particular source is one way to describe this). This argument will include a description of the expected entropy rate and explain how you ensure that sufficient entropy is going into the TOE randomizer seeding process. This discussion will be part of a justification for why the entropy source can be relied upon to produce bits with entropy.

Operating Conditions

Documentation will also include the range of operating conditions under which the entropy source is expected to generate random data. It will clearly describe the measures that have been taken in the system design to ensure the entropy source continues to operate under those conditions. Similarly, documentation shall describe the conditions under which the entropy source is known to malfunction or become inconsistent. Methods used to detect failure or degradation of the source shall be included.

Health Testing

More specifically, all entropy source health tests and their rationale will be documented. This will include a description of the health tests, the rate and conditions under which each health test is performed (e.g., at startup, continuously, or on-demand), the expected results for each health test, and rationale indicating why each test is believed to be appropriate for detecting one or more failures in the entropy source.