# Protection Profile for Mobile Device Management

31 December 2014

Version 2.0

## REVISION HISTORY

| Version | Date | Description |
|---|---|---|
| 1.0 | 21 October 2013 | Initial Release |
| 1.1 | 7 February 2014 | Typographical changes and clarifications to front-matter |
| 2.0 | 31 December 2014 | Separation of MDM Agent SFRs<br>Updated cryptography, protocol, X.509 requirements.<br>Updated management functions to match MDFPPv2.0.<br>Included SSH as a remote administration protocol.<br>Removed IPsec as protocol to communicate to MDM Agent.<br>Added X509 enrollment objective requirement.<br>Added Optional Mobile Application Store requirements. |

## CONTENTS

## TABLE OF TABLES

# 1. INTRODUCTION

Mobile device management (MDM) products allow enterprises to apply security policies to mobile devices, such as smartphones and tablets. The purpose of these policies is to establish a security posture adequate to permit mobile devices to process enterprise data and connect to enterprise network resources.

This document provides a baseline set of Security Functional Requirements (SFRs) for an MDM System, which is the Target of Evaluation (TOE). The MDM System is only one component of an enterprise deployment of mobile devices. Other components, such as the mobile device platforms, which enforce the security policies, and network access control servers, are out of scope.

## 1.1. Compliant Targets of Evaluation

The Mobile Device Management (MDM) system consists of two primary components: the MDM Server software and the MDM Agent. Optionally, the MDM system may consist of a separate Mobile Application Store (MAS) server.

The MDM system operational environment consists of the mobile device on which the MDM Agent resides, the platform on which the MDM Server runs, and an untrusted wireless network over which they communicate, as pictured below.

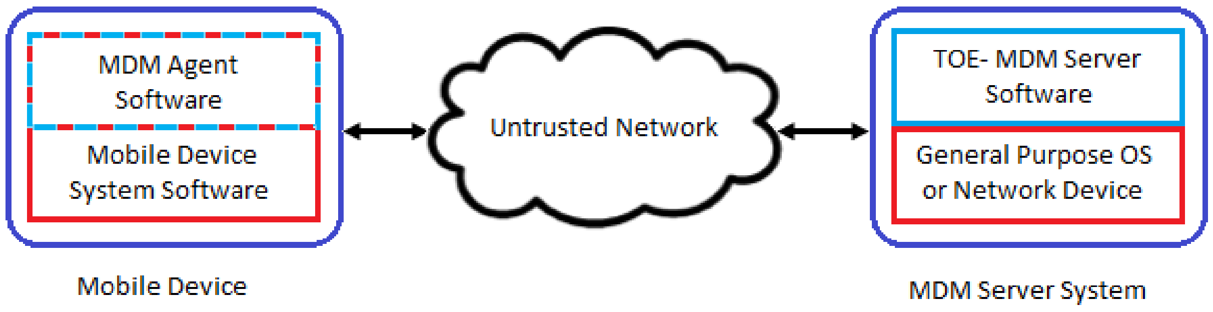

**Figure 1: MDM System Operating Environment**

The **MDM Server** is an application on a general-purpose platform or on a network device, executing in a trusted network environment. The MDM Server provides administration of the mobile device policies and reporting on mobile device behavior. The MDM Server is responsible for managing device enrollment, configuring and sending policies to the MDM Agents, collecting reports on device status, and sending commands to the Agents. The platform on which the MDM Server software runs is either a general-purpose platform or a network device.

The **MDM Agent** establishes a secure connection back to the MDM Server controlled by an enterprise administrator and configures the mobile device per the administrator's policies. Optionally, the MDM Agent may interact with the MAS Server to download and install enterprise applications. The MDM Agent is addressed in the *Extended Package for MDM Agents*. If the MDM Agent is installed on a mobile device as an application developed by the MDM developer, the EP extends this PP and is included in the TOE. In this case, the TOE security functionality specified in this PP must be addressed by the MDM Agent in addition to the MDM Server. Otherwise, the MDM Agent is provided by the mobile device vendor and is out of scope of this PP; however, MDMs are required to indicate the mobile platforms supported by the MDM Server and must be tested against the native MDM agent of those platforms.

The **MAS Server** is an application on a general-purpose platform or on a network device, executing in a trusted network environment. The MAS Server may be separate to or included in the MDM Server. The MAS server hosts applications for the enterprise, authenticates Agents, and securely transmits applications to enrolled mobile devices.

## 1.2. TOE Usage

Requirements in this Protection Profile are designed to address the security problem in the following use cases. Each use case involves acceptance of differing levels and types of risk.

### 1.2.1. Enterprise-owned device for general-purpose enterprise use

An Enterprise-owned device for general-purpose business use is commonly called Corporately Owned, Personally Enabled (COPE). This use case entails a significant degree of Enterprise control over configuration and software inventory. Enterprise administrators use an MDM product to establish policies on the mobile devices prior to user issuance. Users may use Internet connectivity to browse the web or access corporate mail or run Enterprise applications, but this connectivity may be under significant control of the Enterprise. The user may also be expected to store data and use applications for personal, non-enterprise use. The Enterprise administrator uses the MDM product to deploy security policies and query mobile device status. The MDM may issue commands for remediation actions.

### 1.2.2. Enterprise-owned device for specialized, high-security use

An Enterprise-owned device with intentionally-limited network connectivity, tightly-controlled configuration, and limited software inventory is appropriate for specialized, high-security use cases. As in the previous use case, the MDM product is used to establish such policies on mobile devices prior to issuance to users. The device may not be permitted connectivity to any external peripherals. It may only be able to communicate via its WiFi or cellular radios with the Enterprise-run network, which may not even permit connectivity to the Internet. Use of the device may require compliance with usage policies that are more restrictive than those in any general-purpose use case, yet may mitigate risks to highly sensitive information.

### 1.2.3. Personally-owned device for personal and enterprise use

A personally-owned device which is used for both personal activities and Enterprise data is commonly called Bring Your Own Device (BYOD). Unlike in the Enterprise-owned cases, in this scenario the Enterprise is limited in what security policies it can push to the device because the user purchased the device primarily for personal use and is unlikely to accept policies that limit the functionality of the device. However, because the Enterprise allows the user full (or nearly full) access to the Enterprise network, the Enterprise will require certain security policies, for example a password or screenlock policy, and health reporting, such as the integrity of the mobile device system software, before allowing access. The administrator of the MDM can establish remediation actions, such as wipe of the Enterprise data, for non-compliant devices.

# 2. Security Problem Description

Appendix A 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 Appendix A are included.

## 2.1. Threats

### 2.1.1. Malicious and Flawed Applications

Malicious or flawed application (app) threats exist because apps loaded onto a Mobile Device may include malicious or exploitable code. This code could be included unwittingly by its developer, perhaps as part of a software library. Malicious apps may attempt to exfiltrate data to which they have access. Malicious apps may be able to control the device's sensors (geolocation, camera, microphone, etc.) to gather intelligence about the user's surroundings even when those activities do not involve data resident or transmitted from the device. Flawed apps may give an attacker access to perform network-based or physical attacks that otherwise would have been prevented.

[T.MALICIOUS_APPS]

### 2.1.2. Network Attack

An attacker may position themselves on a wireless communications channel or elsewhere on the network infrastructure. From this vantage point, an attacker may initiate communication with the mobile device or alter communication between elements of the operating environment and other endpoints. By altering this communication, the attacker may be able to spoof endpoints such as the MDM Agent or administrative computers.

[T.NETWORK_ATTACK]

### 2.1.3. Network Eavesdropping

In a similar manner to the network attack threat, an attacker may position themselves on a wireless communications channel or elsewhere on the network infrastructure. The attacker may then monitor or gain access to data exchanged between the MDM Server and MDM Agents. By monitoring this data, the attacker may intercept security critical data including cryptographic keys and human-user authentication data.

[T.NETWORK_EAVESDROP]

### 2.1.4. Physical Access

Loss or theft of the underlying mobile device platform may give rise to loss of confidentiality of user data, including, most importantly, credentials. Physical access attacks involve attempts to access the device through external hardware ports, through its user interface, or through direct and possible destructive access to its storage media. Such attacks are intended to gain access to data from a lost or stolen mobile device that it is not expected to be returned to its owner. Although these attacks are primarily directed against the mobile device platform, the TOE configures features which address these threats.

[T.PHYSICAL_ACCESS]

## 2.2. Assumptions

The assumptions for the MDM are defined in Appendix A.1.1: Assumptions.

## 2.3. Organization Security Policy

The organization security policies for the MDM are defined in Appendix A.1.3: Organizational Security Policies.

# 3. SECURITY OBJECTIVES

## 3.1. Security Objectives for the TOE

Compliant TOEs will provide security functionality to address threats to it and to implement policies that address additional threats to the enterprise from inclusion of mobile devices. 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 the TOE and the MDM agent, administrative access to the MDM Server, configuration of security policies for mobile devices, and system reporting for detection of security relevant events.

### 3.1.1. Protected Communications

To address the issues concerning transmitting sensitive data to the TOE (e.g., remote administration to the MDM Server)and between the TOE and the MDM Agent, described in Section 2.1.3, compliant TOEs will use a trusted communication path. The trusted channel between the MDM Server and MDM Agent is implemented using one (or more) of these standard protocols: DTLS, HTTPS, or TLS. If remote administration to the MDM Server is provided, it is implemented using one or more of these standard protocols: IPsec, HTTPS, TLS, or SSH.

To address the threat of network attacks described in Section 2.1.2, the protocols described in this document provide encryption and mutual authentication of each endpoint in a cryptographically secure manner; thus, any attempt by a malicious attacker to represent himself to either endpoint of the communications path as the other party would be detected.

O.DATA_PROTECTION_TRANSIT -> (FAU_STG_EXT.1, FCS_CKM_EXT.4, FCS_CKM.1, FCS_CKM.2, FCS_COP.1(*), FCS_DTLS_EXT.1, FCS_HTTPS_EXT.1, FCS_IPSEC_EXT.1, FCS_IV_EXT.1, FCS_RBG_EXT.1, FCS_SSHS_EXT.1, FCS_STG_EXT.1, FCS_STG_EXT.2, FCS_TLSC_EXT.1, FCS_TLSS_EXT.1, FIA_X509_EXT.1, FIA_X509_EXT.2, FIA_X509_EXT.3, FIA_X509_EXT.4, FPT_ITT.1(*), FTP_ITC.1(*), FTP_TRP.1(*))

### 3.1.2. System Reporting

To ensure that information exists that allows administrators to discover unintentional issues with the configuration and operation of the Server, compliant TOEs have the capability of generating reports which may indicate such issues. Auditing of administrative activities provides information that may hasten corrective action.

O.ACCOUNTABILITY -> (FAU_ALT_EXT.1, FAU_GEN.1, FAU_NET_EXT.1, FAU_SAR.1, FAU_SEL.1, FAU_STG_EXT.1, FAU_STG_EXT.2)

### 3.1.3. Mobile Device Configuration

Mobile devices can accept security policies defined by the Enterprise in order to ensure protection of enterprise data that they may store or process. The MDM Server is responsible for configuring and sending these policies to MDM Agents on devices, sending commands to the MDM Agents, and collecting reports from devices.

O.APPLY_POLICY -> (FIA_ENR_EXT.1, FMT_MOF.1(2), FMT_POL_EXT.1, FMT_SMF.1(1))

### 3.1.4. Administration of Management Capability

To ensure the MDM Server software is operated only by authorized parties, it provides access controls around its management functionality. This includes authentication prior to administrative actions, as well as protections for its own integrity.

O.MANAGEMENT -> (FIA_UAU.1, FMT_MOF.1(*), FMT_SMF.1(*), FMT_SMR.1(*), FTA_TAB.1)

### 1.1.1 MDM Integrity

To ensure integrity of the MDM Server is maintained, conformant TOEs will perform self-tests to ensure correct operation of the MDM Server. In addition, to address the issue of an MDM Server containing malicious or flawed code, the integrity of stored executable code will be verified prior to installation/execution on the device.

O.INTEGRITY-> (FIA_X509_EXT.2, FPT_TST_EXT.1, FPT_TUD_EXT.1)

## 3.2. Security Objectives for the Operational Environment

The objectives that are required to be met by the TOE's operational environment are defined in Appendix A.2.2: Security Objectives for the Operational Environment.

# 4. SECURITY FUNCTIONAL REQUIREMENTS

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 4*, with additional extended functional components.

## 4.1. Conventions

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 deletions with 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).

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

## 4.2. Test Environment for Assurance Activities

As described in the assurance activities below, the ST for an MDM system is required to list all the supported MDM Agents/MD platforms with which an MDM Server operates. The identified assurance activities for testing that includes the use of an Agent should be completed for each MDM Agent/platform listed in the ST.

The evaluator's activities for test that include use of an Agent will ensure that the Server interacts appropriately with the Agent (i.e. sends a policy update to the Agent), but will not ensure that the Agent handles the received data correctly (i.e. appropriately applies the policy to the device), as that is accounted for in the Assurance Activities in the Mobile Device Fundamentals PP or the Extended Package for Mobile Device Management Agents.

## 4.3. TOE Security Functional Requirements

### 4.3.1. Security Audit (FAU)

#### 4.3.1.1. Server Alerts

**FAU_ALT_EXT.1.1** The MDM Server shall alert the administrators in the event of any of the following:

a. change in enrollment status;
b. failure to apply policies to a mobile device;
c. [selection: [assignment: *other events*], no other events]

*Application Note:*

*The MDM Agent is required to report to the MDM Server on successful application of policies on a managed mobile device, and failures can be inferred from the absence of such alerts. This requirement is* *intended to ensure that the MDM Server notifies administrators when policies are not properly installed. Failure to properly install policy updates does not affect the enrollment status of the mobile device.*

***Assurance Activity:***

*TSS*

*The evaluator shall examine the TSS and verify that it describes how the alert system is implemented. The evaluator shall also verify that a description of each assigned event is provided in the TSS.*

*TEST*

*For each MDM Agent/platform listed as supported in the ST:*

*Test 1: The evaluator shall enroll a device and ensure that the MDM server alerts the administrator of the change in enrollment status. The evaluator shall unenroll (retire) a device and ensure that the MDM server alerts the administrator of the change in enrollment status.*

*Test 2: The evaluator shall configure policies, which the MDM agent should not be able to apply. These policies shall include:*

- *a setting which is configurable on the MDM Server interface but not supported by the platform on which the MDM Agent runs, if any such settings exist*
- *an invalid configuration setting, which may require manual modification of the policy prior to transmission to the device*
- *a valid configuration setting with an invalid parameter, which may also require manual modification of the policy prior to transmission to the device*

*The evaluator shall deploy such policies and verify that the MDM server alerts the administrator about the failed application of the policy.*

*Test 3: (Conditional) The evaluator shall trigger each of the events listed and ensure that the MDM Server alerts the administrator.*

#### 4.3.1.2. Audit Data Generation

**FAU_GEN.1.1(1) Refinement:** The **MDM Server** shall be able to generate an audit record of the following auditable events:

a. Start-up and shutdown of the MDM Server software;
b. All administrative actions;
c. Commands issued from the MDM Server to an MDM Agent;
d. Specifically defined auditable events listed in
e. Table 1; and
f. [assignment: other events].

*Application Note:*

*This requirement outlines the information to be included in audit records generated by the MDM Server software. These audit records may be written by the MDM Server software or may be dispatched to the operating system on which it runs. The ST author can include other auditable events in the assignment; they are not limited to the list presented. All audits must contain at least the information mentioned in FAU_GEN.1.2, but may contain more information which can be assigned.*

*Item b above requires all administrative actions to be auditable, so no additional specification for the auditability of these actions is specified in*

*Table 1 aside from those that require additional record content. Administrative actions refer to any management functions specified by FMT_MOF.1(1).*

*Item c includes those commands, which may be performed automatically based on triggers or on a schedule.*

*Depending on the specific requirements selected by the ST author from Security Functional Requirements, Optional Requirements, Selection-Based Requirements, and Objective Requirements, the ST author should include the appropriate auditable events from*

*Table 1 in the ST for the requirements selected.*

***Assurance Activity:***

*TSS*

*The evaluator shall check the TSS and ensure that it lists all of the auditable events. The evaluator shall check to make sure that every audit event type mandated by the PP is described in the TSS.*

*GUIDANCE*

*The evaluator shall check the administrative guide and ensure that it lists all of the auditable events. The evaluator shall check to make sure that every audit event type mandated by the PP is described.*

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

*TEST*

*The evaluator shall test the TOE's ability to correctly generate audit records by having the TOE generate audit records for the events listed in the provided table and administrative actions. This should include all instances of an event. The evaluator shall test that audit records are generated for the establishment and termination of a channel for each of the cryptographic protocols contained in the ST. For administrative actions, the evaluator shall test that each action determined by the evaluator above to be security relevant in the context of this PP is auditable.*

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

**Table 1: Auditable Events**

| **Requirement** | **Auditable Events** | **Additional Audit Record Contents** |
|---|---|---|
| FAU_ALT_EXT.1 | Type of alert. | Identity of Mobile Device that sent alert. |
| FAU_GEN.1 | None. | |

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FAU_NET_EXT.1 | None. | |
| FAU_SAR.1 | None. | |
| FAU_SEL.1 | All modifications to the audit configuration that occur while the audit collection functions are operating. | No additional information. |
| FAU_STG_EXT.1 | None. | |
| FAU_STG_EXT.2 | None. | |
| FCS_CKM_EXT.4 | None. | |
| FCS_CKM.1 | Failure of key generation activity for authentication keys. | No additional information. |
| FCS_CKM.2 | None. | |
| FCS_COP.1(*) | None. | |
| FCS_DTLS_EXT.1 | Failure of the certificate validity check. | |
| FCS_HTTPS_EXT.1 | Failure of the certificate validity check. | |
| FCS_IPSEC_EXT.1 | Failure to establish an IPsec SA. | Reason for failure. |
| FCS_IV_EXT.1 | None. | |
| FCS_RBG_EXT.1 | Failure of the randomization process. | No additional information. |
| FCS_SSHS_EXT.1 | Failure to establish an SSH session.<br>Successful SSH re-key. | Reason for failure.<br>Non-TOE endpoint of connection. |
| FCS_STG_EXT.1 | None. | |
| FCS_STG_EXT.2 | None. | |
| FCS_TLSC_EXT.1 | Failure to establish a TLS session.<br>Failure to verify presented identifier. | Reason for failure.<br>Presented identifier and reference identifier. |
| FCS_TLSS_EXT.1 | Failure to establish a TLS session. | Reason for failure. |
| FIA_ENR_EXT.1 | Failure of MD user authentication. | Presented credentials. |
| FIA_UAU.1 | None. | |
| FIA_X509_EXT.1 | Failure to validate X.509 certificate | Reason for failure. |
| FIA_X509_EXT.2 | Failure to establish connection to determine revocation status. | No additional information. |
| FIA_X509_EXT.3 | Generation of Certificate Request Message.<br>Success or failure of verification. | Content of Certificate Request Message.<br>Issuer and Subject name of added certificate or reason for failure. |
| FIA_X509_EXT.4 | Generation of Certificate Enrollment Request.<br>Success or failure of enrollment.<br>Update of EST Trust Anchor Database. | Issuer and Subject name of EST Server. Method of authentication.<br>Issuer and Subject name of certificate used to authenticate. Content of Certificate Request Message.<br>Issuer and Subject name of added certificate or reason for failure.<br>Subject name of added Root CA. |
| FMT_MOF.1(1) | Issuance of command to perform function.<br>Change of policy settings. | Command sent and identity of MDM Agent recipient. Query responses.<br>Policy changed and value or full policy. |

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FMT_MOF.1(2) | Enrollment by a user. | Identity of user. |
| FMT_POL_EXT.1 | None. | |
| FMT_SMF.1(1) | None. | |
| FMT_SMF.1(2) | Success or failure of function. | No additional information. |
| FMT_SMR.1 | None. | |
| FPT_ITT.1 | Initiation and termination of the trusted channel. | Trusted channel protocol. Identity of initiator and recipient. |
| FPT_TST_EXT.1 | Initiation of self-test. Failure of self-test. Detected integrity violation. | Algorithm that caused failure.<br>The TSF code file that caused the integrity violation. |
| FPT_TUD_EXT.1 | Success or failure of signature verification. | |
| FTA_TAB.1 | Change in banner setting. | No additional information. |
| FTP_ITC.1 | Initiation and termination of the trusted channel. | Trusted channel protocol. Non-TOE endpoint of connection. |
| FTP_TRP.1(1) | Initiation and termination of the trusted channel. | Trusted channel protocol. Identity of administrator. |
| FTP_TRP.1(2) | Initiation and termination of the trusted channel. | Trusted channel protocol. |

#### 4.3.1.3. Network Reachability Review

**FAU_NET_EXT.1.1** The MDM Server shall provide authorized administrators with the capability to read the network connectivity status of an enrolled agent.

*Application Note:*

*The MDM Server establishes the network connectivity status of enrolled agents using periodic reachability event alerts from the agents according to FAU_ALT_EXT.2.1 in the MDM Agent EP. This status may be determined using "polls" from the MDM Server which the Agent is required to respond to or using scheduled periodic notifications of connectivity initiated by the MDM Agent.*

***Assurance Activity:***

*TSS*

*The evaluator ensures that the TSS describes how reachability events are implemented, for each supported mobile platform. The evaluator verifies that this description clearly indicates who (MDM Agent or MDM Server) initiates reachability events.*

*GUIDANCE*

*The evaluator shall verify that the guidance instructs administrators on the method of determining the network connectivity status of an enrolled agent.*

*TEST*

*For each MDM Agent/platform listed as supported in the ST:*

*The evaluator shall configure the MDM Agent/platform to perform a network reachability test, both with and without such connectivity and shall ensure that by following the guidance, the evaluator can determine results that reflect both.*

### 4.3.2. Identification and Authentication (FIA)

#### 4.3.2.1. Enrollment of Mobile Device into Management

**FIA_ENR_EXT.1.1** The MDM Server shall authenticate the remote user over a trusted channel during the enrollment of a mobile device.

*Application Note:*

*The MDM Server may use its own directory or a directory server to perform the authentication decision for users performing the remote enrollment of a mobile device.*

***Assurance Activity:***

*TSS*

*The evaluator shall examine the TSS and verify that it describes the process of enrollment for each MDM Agent/platform listed as supported in the ST. This description shall include the trusted path used for enrollment (FTP_TRP.1(2)), the method of user authentication (username/password, token, etc.), the method of authentication decision (local or remote authentication services), and the actions performed on the MDM Server upon successful authentication.*

*TEST*

*For each MDM Agent/platform listed as supported in the ST:*

*The evaluator shall perform the following tests:*

*Test 1: The evaluator shall attempt to enroll a device without providing correct credentials. The evaluator shall verify that the device is not enrolled and that the described enrollment actions are not taken.*

*Test 2: The evaluator shall attempt to enroll the device providing correct credentials. The evaluator shall verify that the device is enrolled and that the described enrollment actions are taken.*

**FIA_ENR_EXT.1.2** The MDM Server shall limit the user's enrollment of devices to [selection: specific devices, specific device models, a number of devices, specific time period].

*Application Note:*

*This requirement is designed to permit the enterprise to restrict users' enrollment of devices.*

***Assurance Activity:***

*TSS*

*The evaluator shall examine the TSS and verify that it implements a policy to limit the user's enrollment of devices.*

*GUIDANCE*

*The evaluator shall ensure that the administrative guidance describes the method(s) of restricting user enrollment and that it instructs the administrator how to configure the restrictions.*

*TEST*

*For each type of policy selected, the evaluator shall perform the following test:*

*Test: The evaluator shall attempt to configure the MDM Server according to the administrative guidance in order to prevent enrollment. The evaluator shall verify that the user cannot enroll a device outside of the configured limitation. (For example, the evaluator may try to enroll a disallowed device, or may try to enroll additional devices beyond the number allowed.)*

### 4.3.3. Security Management (FMT)

#### 4.3.3.1. Management of Functions in MDM Server

**FMT_MOF.1.1(1) Refinement:** The **MDM Server** shall restrict the ability to perform the functions

- listed in FMT_SMF.1(1)
- enable, disable, and modify policies listed in FMT_SMF.1(1)
- listed in FMT_SMF.1(2)

to *Authorized Administrators*.

*Application Note:*

*This requirement outlines the functions that administrators will have the power to enable, disable, modify, and monitor functions and policies listed in FMT_SMF.1(1). It also includes functions necessary to maintain and configure the MDM Server itself.*

***Assurance Activity:***

*TSS*

*The evaluator shall examine the TSS and user documents to ensure that it describes what security management functions are restricted to the administrator and what actions can be taken for each management function. The evaluator shall verify that the security management functions are restricted to authorized administrators and the administrator is only able to take the actions as described in the user documents.*

*TEST*

*The evaluator shall attempt to access the functions and policies in FMT_SMF.1(1) as an unauthorized user and verify that the attempt fails.*

#### 4.3.3.2. Management of Enrollment Function

**FMT_MOF.1.1(2) Refinement:** The **MDM Server** shall restrict the ability to initiate the enrollment process to *Authorized Administrators and MD users*.

*Application Note:*

*This requirement outlines the enrollment functions that both administrators and MD users may perform. The enrollment actions are identified in the TSS as a part of FIA_ENR_EXT.1.*

*The authorized administrator does not remotely initiate enrollment of the mobile devices that are in the possession of users but may enroll mobile devices when they are in the possession of the administrator, for example, before distributing the mobile devices to the users.*

***Assurance Activity:***

*TSS*

*The evaluator shall examine the TSS and verify that it describes how unauthorized users are prevented from enrolling in the MDM services.*

*TEST*

*The test of this function is performed in conjunction with FIA_ENR_EXT.1.*

#### 4.3.3.3. Specification of Management Functions (Server configuration of Agent)

**FMT_SMF.1.1(1) Refinement:** The **MDM Server** shall be capable of **communicating the** following **commands to the MDM Agent**:

1. transition to the locked state, (*MDF Function 8)*
2. full wipe of protected data, (*MDF Function 9)*
3. unenroll from management,
4. install policies,
5. query connectivity status,
6. query the current version of the MD firmware/software
7. query the current version of the hardware model of the device
8. query the current version of installed mobile applications
9. import X.509v3 certificates into the Trust Anchor Database, (*MDF Function 13)*
10. install applications, (*MDF Function 18)*
11. update system software, (*MDF Function 17)*
12. remove applications, (*MDF Function 16)*
13. remove Enterprise applications, (*MDF Function 19)*

**and the following commands to the MDM Agent:** [selection:

14. wipe Enterprise data, (*MDF Function 28)*
15. remove imported X.509v3 certificates and [selection: no other X.509v3 certificates, [assignment: list of other categories of X.509v3 certificates]] in the Trust Anchor Database, (*MDF Function 14)*
16. alert the administrator,
17. import keys/secrets into the secure key storage, (*MDF Function 11)*
18. destroy imported keys/secrets and [selection: no other keys/secrets, [assignment: list of other categories of keys/secrets]] in the secure key storage, (*MDF Function 12)*
19. read audit logs kept by the MD, (*MDF Function 32)*
20. retrieve MD-software integrity verification values, (*MDF Function 38)*
21. approve exceptions for sharing data between [selection: application processes, group of application processes], (*MDF Function 42)*
22. place applications into application process groups based on [assignment: application characteristics], (*MDF Function 43)*
23. [assignment: list of other management functions to be provided by the MD], no other management functions]

**and the following MD configuration policies:**

24. password policy:
    a. minimum password length
    b. minimum password complexity
    c. maximum password lifetime (*MDF Function 1)*
25. session locking policy:

    a. screen-lock enabled/disabled
    b. screen lock timeout
    c. number of authentication failures (*MDF Function 2)*

26. wireless networks (SSIDs) to which the MD may connect (*MDF Function 6)*
27. security policy for each wireless network:
    a. [selection: specify the CA(s) from which the MD will accept WLAN authentication server certificate(s), specify the FQDN(s) of acceptable WLAN authentication server certificate(s)]
    b. ability to specify security type
    c. ability to specify authentication protocol
    d. specify the client credentials to be used for authentication
    e. [assignment: any additional WLAN management functions] (*MDF Function 7)*
28. application installation policy by [selection:
    a. specifying authorized application repository(s),
    b. specifying a set of allowed applications and versions (an application whitelist)
    c. denying application installation], (*MDF Function 10)*
29. enable/disable policy for [assignment: list of audio or visual collection devices] across MD, [selection: on a per-app basis, no other method], (*MDF Function 5)*

**and the following MD configuration policies:** [selection:

30. enable/disable policy for the VPN protection across MD, [selection: on a per-app basis, no other method], (*MDF Function 3)*
31. enable/disable policy for [assignment: list of radios], (*MDF Function 4)*
32. enable/disable policy for data signaling over [assignment: list of externally accessible hardware ports], (*MDF Function 22)*
33. enable/disable policy for [assignment: list of protocols where the device acts as a server], (*MDF Function 23)*
34. enable/disable policy for developer modes, (*MDF Function 24)*
35. enable policy for data-at rest protection, (*MDF Function 25)*
36. enable policy for removable media's data-at-rest protection, (*MDF Function 26)*
37. enable/disable policy for local authentication bypass, (*MDF Function 27)*
38. the Bluetooth trusted channel policy:
    a. enable/disable the Discoverable mode (for BR/EDR)
    b. change the Bluetooth device name

    [selection:
    c. allow/disallow additional wireless technologies to be used with Bluetooth
    d. disable/enable Advertising (for LE)
    e. disable/enable the Connection mode
    f. disable/enable the Bluetooth services and/or profiles available on the device
    g. specify minimum level of security for each pairing
    h. configure allowable methods of Out of Band pairing
    i. no other Bluetooth configuration] (*MDF Function 20)*
39. enable/disable policy for display notification in the locked state of [selection:
    a. email notifications,
    b. calendar appointments,
    c. contact associated with phone call notification,
    d. text message notification,

    e. other application-based notifications,
    f. none] (*MDF Function 21)*

40. policy for establishing a trusted channel or disallowing establishment if the MD cannot establish a connection to determine the validity of a certificate, (*MDF Function 30)*
41. enable/disable policy for the cellular protocols used to connect to cellular network base stations, (*MDF Function 31)*
42. policy for import and removal by applications of X.509v3 certificates in the Trust Anchor Database, (*MDF Function 29)*
43. [selection: certificate, public-key] used to validate digital signature on applications, (*MDF Function 33)*
44. policy for exceptions for shared use of keys/secrets by multiple applications, (*MDF Function 34)*
45. policy for exceptions for destruction of keys/secrets by applications that did not import the key/secret, (*MDF Function 35)*
46. the unlock banner policy, (*MDF Function 36)*
47. configure the auditable items (*MDF Function 37)*
48. enable/disable [selection:
    a. USB mass storage mode,
    b. USB data transfer without user authentication,
    c. USB data transfer without authentication of the connection system] (*MDF Function 39)*
49. enable/disable backup to [selection: locally connected system, remote system] (*MDF Function 40)*
50. enable/disable [selection:
    a. Hotspot functionality authenticated by [selection: pre-shared key, passcode, no authentication],
    b. USB tethering authenticated by [selection: pre-shared key, passcode, no authentication]] (*MDF Function 41)*
51. enable/disable location services:
    a. across device

    [selection:

    b. on a per-app basis,
    c. no other method] (*MDF Function 44)*
52. enable/disable policy for user unenrollment
53. [assignment: list of other policies to be provided by the MD], no other policies].

*Application Note:*

*This requirement captures all the configuration functionality the TSF provides the administrator to configure the MDM Agent. This requirement is broken into two configurable areas: MDM Agent commands and MDM Agent policies. The ST author can add more commands and configuration policies by completing the appropriate assignment statements*

*The ST author shall not claim any functionality not provided by the Mobile Device. All selections and assignments performed by the ST author in this requirement should match the selections and assignments of the validated Mobile Device ST.*

*Function-specific Application Notes:*

*Function specific application notes reference Mobile Device Fundamentals (MDF) PP v 2.0.*

*The audit records read according to Function 19 are to be transmitted to an external audit server according to FAU_STG_EXT.1. The MDM Server is not expected to retain those logs.*

***Assurance Activity:***

*TSS*

*The evaluator shall examine the TSS to ensure that it describes each management function listed. The evaluator shall examine the TSS to verify that any differences between management functions and policies for each supported MDM Agent/platform listed. The evaluator shall also examine the ST of the claimed Mobile Device to verify that the selections and assignments in the functions and policies in the TSS do not exceed the capabilities of the supported MD.*

*The evaluator shall examine the TSS to ensure that it identifies the management functions implemented for each supported MDM Agent/platform, which are likely to be subsets of the all of the management functions available to the administrator on the MDM Server.*

*TEST*

*For each MDM Agent/platform listed as supported in the ST:*

*TEST: The evaluator shall verify the ability to command each MDM Agent functional capability and configure each MDM Agent policy listed above.*

#### 4.3.3.4. Specification of Management Functions (Server Configuration of Server)

**FMT_SMF.1.1(2) Refinement:** The **MDM Server** shall be capable of **performing** the following management functions:

a. configure X.509v3 certificates for MDM Server use
b. configure the [selection: specific devices, specific device models, a number of devices, specific time period] allowed for enrollment

[selection:

c. allow the administrator to choose whether to accept the certificate when a connection cannot be made to establish validity,
d. configure the TOE unlock banner,
e. configure periodicity of the following commands to the agent: [assignment: list of commands],
f. [assignment: additional functions required to support SFRs],
g. no other management functions

].

*Application Note:*

*This requirement captures all the configuration functionality in the MDM Server to configure the underlying MDM Server. The ST author can add more commands and configuration policies by completing the assignment statement. The selection in b corresponds to the selection in FIA_ENR_EXT.1.2. The selection in d includes a function that corresponds to the selection in FIA_X509_EXT.2.2.*

*Function e allows the administrator to configure periodicity of assigned commands, for example "read audit logs kept by the Mobile Device". In this way the administrator can configure the MDM system to retrieve audit logs from the Mobile Device on a periodic, such as daily, basis in order to ensure freshness of log data and to minimize loss of audit logs.*

***Assurance Activity:***

*TSS*

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

*GUIDANCE*

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

*TEST*

The test of functions b, c, and d are performed in conjunction with the use of the function. The evaluator shall perform the following test:

*Test 1: The evaluator shall configure the TSF authentication certificate(s) and verify that the certificate is used in established trusted connections (FPT_ITT.1, FTP_ITC.1, and FTP_TRP.1).*

*Test 2: (conditional) The evaluator shall configure the periodicity for the assigned list of commands to the agent for several configured time periods and shall verify that the MDM Server performs the commands schedule.*

*Test 3: (conditional) The evaluator shall design and perform tests to demonstrate that the assigned function may be configured and that the intended behavior of the function is enacted by the MDM Server.*

#### 4.3.3.5. Security Management Roles

**FMT_SMR.1.1(1) Refinement:** The **MDM Server** shall maintain the roles *administrator, MD user, and [assignment: additional authorized identified roles]*.

**FMT_SMR.1.2(1) Refinement:** The **MDM Server** shall be able to associate users with roles.

*Application Note:*

*It is envisioned that the MDM Server will be configured and maintained by different user roles. The assignment is used by the ST author to list the roles that are supported. At a minimum, one administrative role shall be supported. If no additional roles are supported, then "no additional roles" is stated. The MD user role is used for enrollment of mobile devices to the MDM according to FIA_ENR_EXT.1.*

***Assurance Activity:***

*TSS*

*The evaluator shall examine the TSS to verify that it describes the administrator role and the powers granted to and limitations of the role.*

*GUIDANCE*

*The evaluator shall review the operational guidance to ensure that it contains instructions for administering the TOE and which interfaces are supported.*

*TEST*

*In the course of performing the testing activities for the evaluation, the evaluator shall use all supported interfaces, although it is not necessary to repeat each test involving an administrative action with each interface. The evaluator shall ensure, however, that each supported method of administering the TOE* *that conforms to the requirements of this PP be tested; for instance, if the TOE can be administered through a local hardware interface or TLS/HTTPS then both methods of administration must be exercised during the evaluation team's test activities.*

#### 4.3.3.6. Trusted Update

**FPT_TUD_EXT.1.1** The MDM Server shall provide Authorized Administrators the ability to query the current version of the MDM Server software.

***Assurance Activity:***

*GUIDANCE*

*The evaluator shall ensure that the administrator guidance includes instructions for determining the current version of the TOE.*

*TEST*

*The evaluator shall query the TSF for the current version of the software according to the AGD guidance and shall verify that the current version matches that of the documented and installed version.*

### 4.3.4. Protection of the TSF (FPT)

The ST author must include at least one FPT_ITT.1(2) or FTP_ITC.1(2) in the ST. If the TOE includes an MDM Agent, the ST author shall include FPT_ITT.1(2). If the TOE interoperates with an evaluated MDM Agent built into a mobile device, the ST author shall include FTP_ITC.1(2).

If the TOE includes separate components (such as a MDM Server and a MAS Server), the ST author shall include FPT_ITT.1(1).

### 4.3.5. Trusted Path/Channels (FTP)

The ST author must include at least one FPT_ITT.1(2) or FTP_ITC.1(2) in the ST. If the TOE includes an MDM Agent, the ST author shall include FPT_ITT.1(2). If the TOE interoperates with an evaluated MDM Agent built into a mobile device, the ST author shall include FTP_ITC.1(2).

## 4.4. TOE or Platform Security Functional Requirements

This section identifies the SFRs that must be performed by the MDM Server or by the MDM Server's platform. Each requirement includes a selection for the ST author to indicate whether the MDM Server or the MDM Server's platform performs the functionality in the requirement. The assurance activity for those requirements for which the platform has been selected shall comply with the scheme's policy for platform-dependent products.

### 4.4.1. Security Audit (FAU)

#### 4.4.1.1. Audit Data Generation

**FAU_GEN.1.2(1) Refinement:** The [<u>selection: *MDM Server, MDM Server platform*</u>] shall record within each TSF audit record at least the following information:

- date and time of the event,
- type of event,
- subject identity**,**
- (if relevant) the outcome (success or failure) of the event,

- additional information in
- Table 1,
- [assignment: *other audit relevant information*].

*Application Note:*

*All audits must contain at least the information mentioned in FAU_GEN.1.2, but may contain more information which can be assigned. The ST author shall identify in the TSS which information of the audit record that is performed by the TSF and that which is performed by the TOE platform.*

***Assurance Activity:***

*TSS*

*The evaluator shall check the TSS and ensure that it provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field.*

*GUIDANCE*

*The evaluator shall check the administrative guide and ensure that it provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field. The evaluator shall check to make sure that the description of the fields contains the information required in FAU_GEN.1.2.*

*TEST*

*When verifying the test results from FAU_GEN.1.1, the evaluator shall ensure the audit records generated during testing match the format specified in the administrative guide, and that the fields in each audit record have the proper entries.*

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

#### 4.4.1.2. External Audit Trail Storage

**FAU_STG_EXT.1.1** The [selection: MDM Server, MDM Server platform] shall be able to transmit audit data to an external IT entity using a trusted channel implementing the [selection: IPsec, SSH, TLS, TLS/HTTPS] protocol.

*Application Note:*

*The TOE may rely on a non-TOE audit server for storage and review of audit records. Although the TOE generates audit records and receives audit records from managed mobile devices, the storage of these audit records and the ability to allow the administrator to review these audit records is provided by the operational environment. The TSF may rely on the underlying operating system for this functionality, and the first selection should be made appropriately.*

*In the second selection, the ST author chooses the means by which this connection is protected. The ST author also ensures that the supporting protocol requirement matching the selection is included in the ST.*

***Assurance Activity:***

*TSS*

*The evaluator shall examine the TSS to ensure it describes the means by which the audit data are transferred to the external audit server, and how the trusted channel is provided.*

*GUIDANCE*

*The evaluator shall also examine the operational guidance to determine that it describes the relationship between the local audit data and the audit data that are sent to the audit log server. For example, when an audit event is generated, is it simultaneously sent to the external server and the local store, or is the local store used as a buffer and "cleared" periodically by sending the data to the audit server.*

*The evaluator shall also examine the operational guidance to ensure it describes how to establish the trusted channel to the audit server, as well as describe any requirements on the audit server (particular audit server protocol, version of the protocol required, etc.), as well as configuration of the TOE needed to communicate with the audit server.*

*TEST*

*Testing of the trusted channel mechanism will be performed as specified in the associated assurance activities for the particular trusted channel mechanism.*

*The evaluator shall perform the following test for this requirement:*

*Test: The evaluator shall establish a session between the TOE and the audit server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the audit server and the TOE during several activities of the evaluator's choice designed to generate audit data to be transferred to the audit server. The evaluator shall observe that these data are not able to be viewed in the clear during this transfer, and that they are successfully received by the audit server. The evaluator shall record the particular software (name, version) used on the audit server during testing.*

### 4.4.2. Cryptographic Support (FCS)

#### 4.4.2.1. Cryptographic Key Generation

**FCS_CKM.1.1 Refinement:** The [selection: **TSF, TOE platform**] shall generate **asymmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm [selection:

- **RSA schemes using** cryptographic key sizes of **2048-bit or greater** that meet the following: [**selection:**
  - **FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;**
  - **ANSI X9.31-1998, Section 4.1];**
- **ECC schemes using "NIST curves" P-256, P-384 and [selection: P-521, no other curves]** that meet the following: **FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;**
- **FFC schemes using** cryptographic key sizes of **2048-bit or greater** that meet the following: **FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1**

].

***Application Note:*** *The ST author shall select all key generation schemes used for key establishment and MDM authentication. When key generation is used for key establishment, the schemes in FCS_CKM.2.1* *and selected cryptographic protocols must match the selection. When key generation is used for MDM authentication, the public key is expected to be associated with an X.509v3 certificate.*

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

*The ANSI X9.31-1998 option will be removed from the selection in a future publication of this document. Presently, the selection is not exclusively limited to the FIPS PUB 186-4 options in order to allow industry some further time to complete the transition to the modern FIPS PUB 186-4 standard.*

*ECC schemes will be required for products entering evaluation after Quarter 3, 2015.*

***Assurance Activity:***

**Requirement met by the platform**

*TSS*

*For each platform listed in the ST, the evaluator shall examine the ST of the platform to ensure that the key generation claimed in that platform's ST contains the key generation requirement in the MDM Server's ST. The evaluator shall also examine the TSS of the MDM Server's ST to verify that it describes (for each supported platform) how the key generation functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this assurance activity).*

**Requirement met by the TOE**

*TSS*

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

*GUIDANCE*

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

*TEST*

***Key Generation for FIPS PUB 186-4 RSA Schemes***

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

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

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

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

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

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

***Key Generation for ANSI X9.31-1998 RSA Schemes***

*If the TSF implements the ANSI X9.31-1998 scheme, the evaluator shall check to ensure that the TSS describes how the key-pairs are generated. In order to show that the TSF implementation complies with ANSI X9.31-1998, the evaluator shall ensure that the TSS contains the following information:*

*The TSS shall list all sections of the standard to which the TOE complies;*

*For each applicable section listed in the TSS, for all statements that are not "shall" (that is, "shall not", "should", and "should not"), if the TOE implements such options it shall be described in the TSS. If the included functionality is indicated as "shall not" or "should not" in the standard, the TSS shall provide a rationale for why this will not adversely affect the security policy implemented by the TOE;*

*For each applicable section of Appendix B, any omission of functionality related to "shall" or "should" statements shall be described.*

***Key Generation for Elliptic Curve Cryptography (ECC)***

*FIPS 186-4 ECC Key Generation Test*

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

*FIPS 186-4 Public Key Verification (PKV) Test*

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

***Key Generation for Finite-Field Cryptography (FFC)***

*The evaluator shall verify the implementation of the Parameters Generation and the Key Generation for FFC by the TOE using the Parameter Generation and Key Generation test. This test verifies the ability of the TSF to correctly produce values for the field prime p, the cryptographic prime q (dividing p-1), the cryptographic group generator g, and the calculation of the private key x and public key y.*

*The Parameter generation specifies 2 ways (or methods) to generate the cryptographic prime q and the field prime p:*

*Cryptographic and Field Primes:*

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

*and two ways to generate the cryptographic group generator g:*

*Cryptographic Group Generator:*

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

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

*Private Key:*

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

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

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

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

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

*for each FFC parameter set and key pair.*

#### 4.4.2.2. Cryptographic Key Establishment

**FCS_CKM.2.1 Refinement** The [selection: **TSF, TOE platform**] shall **perform** cryptographic **key establishment** in accordance with a specified cryptographic key **establishment** method: [selection:

- **RSA-based key establishment schemes** that meets the following: **NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography"**;

- **Elliptic curve-based key establishment schemes** that meets the following: **NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"**;

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

].

*Application Note:*

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

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

*The elliptic curves used for the key establishment scheme shall correlate with the curves specified in FCS_CKM.1.1. Elliptic curve-based schemes will be required for products entering evaluation after Quarter 3, 2015.*

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

***Assurance Activity:***

**Requirement met by the platform**

*TSS*

*For each platform listed in the ST, the evaluator shall examine the ST of the platform to ensure that the key establishment claimed in that platform's ST contains the key establishment requirement in the MDM Server's ST. The evaluator shall also examine the TSS of the MDM Server's ST to verify that it describes (for each supported platform) how the key establishment functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this assurance activity).*

**Requirement met by the TOE**

*TSS*

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

*GUDIANCE*

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

*TEST*

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

***SP800-56A Key Establishment Schemes***

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

*<u>Function Test</u>*

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

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

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

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

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

*<u>Validity Test</u>*

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

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

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

***SP800-56B Key Establishment Schemes***

*The evaluator shall verify that the TSS describes whether the TOE acts as a sender, a recipient, or both for RSA-based key establishment schemes.*

*If the TOE acts as a sender, the following assurance activity shall be performed to ensure the proper operation of every TOE supported combination of RSA-based key establishment scheme:*

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

*If the TOE acts as a receiver, the following assurance activities shall be performed to ensure the proper operation of every TOE supported combination of RSA-based key establishment scheme:*

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

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

### 4.4.2.3. Cryptographic Key Destruction

**FCS_CKM_EXT.4.1** The [selection: TSF, TOE platform] shall destroy plaintext keying material and critical security parameters in accordance with the following rules:

- For volatile memory, the destruction shall be executed by a single direct overwrite [selection: consisting of a pseudo-random pattern using the TSF's RBG, consisting of zeroes] following by a read-verify.
- For non-volatile EEPROM, the destruction shall be executed by a single direct overwrite consisting of a pseudo random pattern using the TSF's RBG (as specified in FCS_RBG_EXT.1) followed by a read-verify.
- For non-volatile flash memory, the destruction shall be executed [selection: by a direct overwrite of all copies consisting of zeros followed by a read-verify, by a block erase followed by a read-verify].
- For non-volatile memory other than EEPROM and flash, the destruction shall be executed by overwriting three or more times with a random pattern that is changed before each write.

*Application Note:*

*The ST author should select the platform if the MDM Server performs no operations using plaintext secret, private cryptographic keys, and CSPs.*

*Any security related information (such as keys, authentication data, and passwords) must be zeroized when no longer in use to prevent the disclosure or modification of security critical data.*

*The zeroization indicated above applies to each intermediate storage area for plaintext key and Cryptographic Service Provider (CSP) (i.e., any storage, such as memory buffers, that is included in the path of such data) upon the transfer of the key/CSP to another location.*

*Since the TOE does not include the host IT environment, the extent of this capability is necessarily somewhat limited. For the purposes of this requirement, it is sufficient for the TOE to invoke the correct underlying functions of the host to perform the zeroization--it does not imply that the TOE has to include a kernel-mode memory driver to ensure the data are zeroized. It is assumed that the host platform appropriately performs zeroization of key material in its internal processes.*

**FCS_CKM_EXT.4.2** The TSF shall destroy all plaintext keying material and critical security parameters (CSP) when no longer needed.

*Application Note:*

*For the purposes of this requirement, plaintext keying material refers to authentication data, authorization data, secret/private symmetric keys, data used to derive keys, etc. Key destruction procedures are performed in accordance with FCS_CKM.4.1.*

***Assurance Activity:***

**Requirement met by the TOE platform**

*TSS*

*The evaluator shall check to ensure the TSS describes each of the secret keys (keys used for symmetric encryption), private keys, and CSPs used to generate key that are not otherwise covered by the FCS_CKM_EXT.4 requirement levied on the TOE.*

*For each platform listed in the ST, the evaluator shall examine the TSS of the ST of the platform to ensure that each of the secret keys, private keys, and CSPs used to generate key listed above are covered.*

**Requirement met by TSF**

*TSS*

*The evaluator shall check to ensure the TSS lists each type of plaintext key material and CSP (authentication data, authorization data, secret/private symmetric keys, data used to derive keys, etc.) and its origin and storage location.*

*The evaluator shall verify that the TSS describes when each type of key material and CSP is cleared.*

*The evaluator shall also verify that, for each type, the type of clearing procedure that is performed (overwrite with zeros, overwrite three or more times by a different alternating pattern, overwrite with random pattern, or block erase). If different types of memory are used to store the materials to be protected, the evaluator shall check to ensure that the TSS describes the clearing procedure in terms of the memory in which the data are stored (for example, "secret keys stored on flash are cleared by overwriting once with zeros, while secret keys stored on the internal persistent storage device are cleared by overwriting three times with a random pattern that is changed before each write"). For block erases, the evaluator shall also ensure that the block erase command used is listed and shall verify that the command used also addresses any copies of the plaintext key material and that may be created in order to optimize the use of flash memory.*

*TEST*

*For each software and firmware key clearing situation the evaluator shall repeat the following tests. Note that at this time hardware-bound keys are explicitly excluded from testing.*

*Test 1: The evaluator shall utilize appropriate combinations of specialized operational environment and development tools (debuggers, simulators, etc.) for the TOE and instrumented TOE builds to test that keys are cleared correctly, including all intermediate copies of the key that may have been created internally by the TOE during normal cryptographic processing with that key.*

*Cryptographic TOE implementations in software shall be loaded and exercised under a debugger to perform such tests. The evaluator shall perform the following test for each key subject to clearing, including intermediate copies of keys that are persisted encrypted by the TOE:*

1. *Load the instrumented TOE build in a debugger.*
2. *Record the value of the key in the TOE subject to clearing.*
3. *Cause the TOE to perform a normal cryptographic processing with the key from #1.*
4. *Cause the TOE to clear the key.*
5. *Cause the TOE to stop the execution but not exit.*
6. *Cause the TOE to dump the entire memory footprint of the TOE into a binary file.*
7. *Search the content of the binary file created in #4 for instances of the known key value from #1.*

*The test succeeds if no copies of the key from #1 are found in step #7 above and fails otherwise.*

*The evaluator shall perform this test on all keys, including those persisted in encrypted form, to ensure intermediate copies are cleared.*

*Test 2: In cases where the TOE is implemented in firmware and operates in a limited operating environment that does not allow the use of debuggers, the evaluator shall utilize a simulator for the TOE on a general purpose operating system. The evaluator shall provide a rationale explaining the instrumentation of the simulated test environment and justifying the obtained test results.*

#### 4.4.2.4. Cryptographic Operation (Confidentiality Algorithms)

**FCS_COP.1.1(1) Refinement:** The [selection: **TSF, TOE platform**] shall perform **encryption/decryption** in accordance with a specified cryptographic algorithm [selection:

- **<u>AES-CBC (as defined in FIPS PUB 197, and NIST SP 800-38A) mode,</u>**
- **<u>AES-GCM (as defined in NIST SP 800-38D),</u>**
- **<u>AES Key Wrap (KW) (as defined in NIST SP 800-38F),</u>**
- **<u>AES Key Wrap with Padding (KWP) (as defined in NIST SP 800-38F),</u>**
- **<u>AES-CCM (as defined in NIST SP 800-38C),</u>**

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

*Application Note:*

*For the first selection of FCS_COP.1.1(1), the ST author should choose the mode or modes in which AES operates in the trusted channel protocols.. For the second selection, the ST author should choose the key sizes that are supported by this functionality.*

*Support for 256-bit key sizes will be required for products entering evaluation after Quarter 3, 2015.*

***Assurance Activity:***

**Requirement met by the platform**

*TSS*

*For each platform listed in the ST, the evaluator shall examine the ST of the platform to ensure that the encryption/decryption function(s) claimed in that platform's ST contains the encryption/decryption function(s) in the MDM Server's ST. The evaluator shall also examine the TSS of the MDM Server's ST to verify that it describes (for each supported platform) how the encryption/decryption functionality is invoked for each mode and key size selected in the MDM Server's ST (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this assurance activity).*

**Requirement met by the MDM Server**

*TEST*

***AES-CBC Tests***

*<u>AES-CBC Known Answer Tests</u>*

*There are four Known Answer Tests (KATs), described below. In all KATs, the plaintext, ciphertext, and IV values shall be 128-bit blocks. The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine* *correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.*

> ***KAT-1.*** *To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 10 plaintext values and obtain the ciphertext value that results from AES-CBC encryption of the given plaintext using a key value of all zeros and an IV of all zeros. Five plaintext values shall be encrypted with a 128-bit all-zeros key, and the other five shall be encrypted with a 256-bit all-zeros key.*
>
> *To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using 10 ciphertext values as input and AES-CBC decryption.*
>
> ***KAT-2.*** *To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 10 key values and obtain the ciphertext value that results from AES-CBC encryption of an all-zeros plaintext using the given key value and an IV of all zeros. Five of the keys shall be 128-bit keys, and the other five shall be 256-bit keys.*
>
> *To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using an all-zero ciphertext value as input and AES-CBC decryption.*
>
> ***KAT-3.*** *To test the encrypt functionality of AES-CBC, the evaluator shall supply the two sets of key values described below and obtain the ciphertext value that results from AES encryption of an all-zeros plaintext using the given key value and an IV of all zeros. The first set of keys shall have 128 128-bit keys, and the second set shall have 256 256-bit keys. Key i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1,N].*
>
> *To test the decrypt functionality of AES-CBC, the evaluator shall supply the two sets of key and ciphertext value pairs described below and obtain the plaintext value that results from AES-CBC decryption of the given ciphertext using the given key and an IV of all zeros. The first set of key/ciphertext pairs shall have 128 128-bit key/ciphertext pairs, and the second set of key/ciphertext pairs shall have 256 256-bit key/ciphertext pairs. Key i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1,N]. The ciphertext value in each pair shall be the value that results in an all-zeros plaintext when decrypted with its corresponding key.*
>
> ***KAT-4.*** *To test the encrypt functionality of AES-CBC, the evaluator shall supply the set of 128 plaintext values described below and obtain the two ciphertext values that result from AES-CBC encryption of the given plaintext using a 128-bit key value of all zeros with an IV of all zeros and using a 256-bit key value of all zeros with an IV of all zeros, respectively. Plaintext value i in each set shall have the leftmost i bits be ones and the rightmost 128-i bits be zeros, for i in [1,128].*
>
> *To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using ciphertext values of the same form as the plaintext in the encrypt test as input and AES-CBC decryption.*

*<u>AES-CBC Multi-Block Message Test</u>*

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

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

*<u>AES-CBC Monte Carlo Tests</u>*

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

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

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

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

***AES-GCM Test***

*The evaluator shall test the authenticated encrypt functionality of AES-GCM for each combination of the following input parameter lengths:*

- *<u>128 bit and 256 bit keys</u>*
- *<u>Two plaintext lengths.</u> One of the plaintext lengths shall be a non-zero integer multiple of 128 bits, if supported. The other plaintext length shall not be an integer multiple of 128 bits, if supported.*
- *<u>Three AAD lengths.</u> One AAD length shall be 0, if supported. One AAD length shall be a non-zero integer multiple of 128 bits, if supported. One AAD length shall not be an integer multiple of 128 bits, if supported.*
- *<u>Two IV lengths.</u> If 96 bit IV is supported, 96 bits shall be one of the two IV lengths tested.*

*The evaluator shall test the encrypt functionality using a set of 10 key, plaintext, AAD, and IV tuples for each combination of parameter lengths above and obtain the ciphertext value and tag that results from AES-GCM authenticated encrypt. Each supported tag length shall be tested at least once per set of 10. The IV value may be supplied by the evaluator or the implementation being tested, as long as it is known.*

*The evaluator shall test the decrypt functionality using a set of 10 key, ciphertext, tag, AAD, and IV 5-tuples for each combination of parameter lengths above and obtain a Pass/Fail result on authentication and the decrypted plaintext if Pass. The set shall include five tuples that Pass and five that Fail.*

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

***AES-CCM Tests***

*The evaluator shall test the generation-encryption and decryption-verification functionality of AES-CCM for the following input parameter and tag lengths:*

- *128 bit and 256 bit keys*
- *Two payload lengths. One payload length shall be the shortest supported payload length, greater than or equal to zero bytes. The other payload length shall be the longest supported payload length, less than or equal to 32 bytes (256 bits).*
- *Two or three associated data lengths. One associated data length shall be 0, if supported. One associated data length shall be the shortest supported payload length, greater than or equal to zero bytes. One associated data length shall be the longest supported payload length, less than or equal to 32 bytes (256 bits). If the implementation supports an associated data length of $2^{16}$ bytes, an associated data length of $2^{16}$ bytes shall be tested.*
- *Nonce lengths. All supported nonce lengths between 7 and 13 bytes, inclusive, shall be tested.*
- *Tag lengths. All supported tag lengths of 4, 6, 8, 10, 12, 14 and 16 bytes shall be tested.*

*To test the generation-encryption functionality of AES-CCM, the evaluator shall perform the following four tests:*

*Test 1. For EACH supported key and associated data length and ANY supported payload, nonce and tag length, the evaluator shall supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext.*

*Test 2. For EACH supported key and payload length and ANY supported associated data, nonce and tag length, the evaluator shall supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext.*

*Test 3. For EACH supported key and nonce length and ANY supported associated data, payload and tag length, the evaluator shall supply one key value and 10 associated data, payload and nonce value 3-tuples and obtain the resulting ciphertext.*

*Test 4. For EACH supported key and tag length and ANY supported associated data, payload and nonce length, the evaluator shall supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext.*

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

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

***AES Key Wrap (AES-KW) and Key Wrap with Padding (AES-KWP) Test***

*The evaluator shall test the authenticated encryption functionality of AES-KW for EACH combination of the following input parameter lengths:*

- *<u>128 and 256 bit key encryption keys (KEKs)</u>*
- *<u>Three plaintext lengths</u>. One of the plaintext lengths shall be two semi-blocks (128 bits). One of the plaintext lengths shall be three semi-blocks (192 bits). The third data unit length shall be the longest supported plaintext length less than or equal to 64 semi-blocks (4096 bits).*

*using a set of 100 key and plaintext pairs and obtain the ciphertext that results from AES-KW authenticated encryption. To determine correctness, the evaluator shall use the AES-KW authenticated-encryption function of a known good implementation.*

*The evaluator shall test the authenticated-decryption functionality of AES-KW using the same test as for authenticated-encryption, replacing plaintext values with ciphertext values and AES-KW authenticated-encryption with AES-KW authenticated-decryption.*

*The evaluator shall test the authenticated-encryption functionality of AES-KWP using the same test as for AES-KW authenticated-encryption with the following change in the three plaintext lengths:*

- *One plaintext length shall be one octet. One plaintext length shall be 20 octets (160 bits).*
- *One plaintext length shall be the longest supported plaintext length less than or equal to 512 octets (4096 bits).*

*The evaluator shall test the authenticated-decryption functionality of AES-KWP using the same test as for AES-KWP authenticated-encryption, replacing plaintext values with ciphertext values and AES-KWP authenticated-encryption with AES-KWP authenticated-decryption.*

#### 4.4.2.5. Cryptographic Operation (Hashing)

**FCS_COP.1.1(2) Refinement:** The [selection: **<u>TSF, TOE platform</u>**] shall perform **cryptographic hashing** in accordance with a specified cryptographic algorithm [selection*:* **<u>SHA-1, SHA-256, SHA-384, SHA-512</u>**] and **message digest** sizes [selection*:* **<u>160, 256, 384, 512</u>] bits** that meet the following: ***FIPS Pub 180-4***.

*Application Note:*

*Per NIST SP 800-131A, SHA-1 for generating digital signatures is no longer allowed, and SHA-1 for verification of digital signatures is strongly discouraged as there may be risk in accepting these signatures. It is expected that vendors will implement SHA-2 algorithms in accordance with SP 800-131A, and products entering into evaluation after Quarter 3, 2015 will be required to include SHA-2 algorithms.*

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

**Assurance Activity:**

**Requirement met by the TOE platform**

*TSS*

*For each platform listed in the ST, the evaluator shall examine the ST of the platform to ensure that the hash function(s) claimed in that platform's ST contains the hash function(s) in the MDM Server's ST. The evaluator shall also examine the TSS of the MDM Server's ST to verify that it describes (for each supported platform) how the hash functionality is invoked for each digest size selected in the MDM*

*Server's ST (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this assurance activity).*

**Requirement met by the TSF**

*TSS*

*The evaluator shall check that the association of the hash function with other TSF cryptographic functions (for example, the digital signature verification function) is documented in the TSS.The evaluator checks the AGD documents to determine that any configuration that is required to be done to configure the functionality for the required hash sizes is present. 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 8. 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 testmacs.*

*TEST*

*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*

*The evaluators devise an input set consisting of m+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m bits. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.*

*Short Messages Test - Byte-oriented Mode*

*The evaluators devise an input set consisting of m/8+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m/8 bytes, with each message being an integral number of bytes. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.*

*Selected Long Messages Test - Bit-oriented Mode*

*The evaluators devise an input set consisting of m messages, where m is the block length of the hash algorithm. The length of the ith message is 512 + 99*i, where 1 ≤ i ≤ m. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.*

*Selected Long Messages Test - Byte-oriented Mode*

*The evaluators devise an input set consisting of m/8 messages, where m is the block length of the hash algorithm. The length of the ith message is 512 + 8*99*i, where 1 ≤ i ≤ m/8. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.*

*Pseudorandomly Generated Messages Test*

*This test is for byte-oriented implementations only. The evaluators randomly generate a seed that is n bits long, where n is the length of the message digest produced by the hash function to be tested. The evaluators then formulate a set of 100 messages and associated digests by following the algorithm* *provided in Figure 1 of [SHAVS]. The evaluators then ensure that the correct result is produced when the messages are provided to the TSF.*

#### 4.4.2.6. Cryptographic Operation (Digital Signature)

**FCS_COP.1.1(3) Refinement:** The [selection: **TSF, TOE platform**] shall perform **cryptographic signature services (generation and verification)** in accordance with a specified cryptographic algorithm [selection:

- **RSA schemes** using cryptographic key sizes **of 2048-bit or greater** that meet the following: **FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 4**;
- **ECDSA schemes** using **"NIST curves" P-256, P-384 and [selection: P-521, no other curves]** that meet the following: **FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5**;

].

*Application Note:*

*The ST Author should choose the algorithm implemented to perform digital signatures. The MDM Server must perform digital signatures in accordance with the trusted channel protocols. The MDM Server is required to validate any signed policies and policy updates sent by the MDM Server.*

*ECDSA schemes will be required for products entering evaluation after Quarter 3, 2015.*

**Assurance Activity:**

**Requirement met by the TOE platform**

*TSS*

*For each platform listed in the ST, the evaluator shall examine the ST of the platform to ensure that the digital signature functions claimed in that platform's ST contains the digital signature functions in the MDM Server's ST. The evaluator shall also examine the TSS of the MDM Server's ST to verify that it describes (for each supported platform) how the digital signature functionality is invoked for each operation they are used for in the MDM Server (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this assurance activity).*

**Requirement met by the TSF**

*TEST*

***ECDSA Algorithm Tests***

*ECDSA FIPS 186-4 Signature Generation Test*

*For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate 10 1024-bit long messages and obtain for each message a public key and the resulting signature values R and S. To determine correctness, the evaluator shall use the signature verification function of a known good implementation.*

*ECDSA FIPS 186-4 Signature Verification Test*

*For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate a set of 10 1024-bit message, public key and signature tuples and modify one of* *the values (message, public key or signature) in five of the 10 tuples. The evaluator shall obtain in response a set of 10 PASS/FAIL values.*

***RSA Signature Algorithm Tests***

*Signature Generation Test*

*The evaluator shall verify the implementation of RSA Signature Generation by the TOE using the Signature Generation Test. To conduct this test the evaluator must generate or obtain 10 messages from a trusted reference implementation for each modulus size/SHA combination supported by the TSF. The evaluator shall have the TOE use their private key and modulus value to sign these messages.*

*The evaluator shall verify the correctness of the TSF's signature using a known good implementation and the associated public keys to verify the signatures.*

*Signature Verification Test*

*The evaluator shall perform the Signature Verification test to verify the ability of the TOE to recognize another party's valid and invalid signatures. The evaluator shall inject errors into the test vectors produced during the Signature Verification Test by introducing errors in some of the public keys e, messages, IR format, and/or signatures. The TOE attempts to verify the signatures and returns success or failure.*

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

#### 4.4.2.7. Cryptographic Operation (Keyed-Hash Message Authentication)

**FCS_COP.1.1(4) Refinement:** The [selection: **TSF, TOE platform**] 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 180-4, "Secure Hash Standard."**

*Application Note:*

*The intent of this requirement is to specify the keyed-hash message authentication function used when used for key establishment purposes for the various cryptographic protocols used by the TOE (e.g., trusted channel). 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(3).*

***Assurance Activity:***

**Requirement met by the TOE platform**

*TSS*

*For each platform listed in the ST, the evaluator shall examine the ST of the platform to ensure that the keyed-hash function(s) claimed in that platform's ST contains the keyed-hash function(s) in the MDM Server's ST. The evaluator shall also examine the TSS of the MDM Server's ST to verify that it describes (for each supported platform) how the keyed-hash functionality is invoked for each mode and key size selected in the MDM Server's ST (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this assurance activity).*

**Requirement met by the TSF**

*TSS*

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

*TEST*

*For each of the supported parameter sets, the evaluator shall compose 15 sets of test data. Each set shall consist of a key and message data. The evaluator shall have the TSF generate HMAC tags for these sets of test data. The resulting MAC tags shall be compared to the result of generating HMAC tags with the same key and IV using a known good implementation.*

#### 4.4.2.8. Extended: Random Bit Generation

**FCS_RBG_EXT.1.1**The [selection: TSF, TOE platform] shall perform all deterministic random bit generation services in accordance with [selection, *choose one of:* NIST Special Publication 800-90A using [selection: Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES)]; FIPS Pub 140-2 Annex C: X9.31 Appendix 2.4 using AES].

*Application Note:*

*The ST author should select whether the Server provides its own DRBG or uses the platform's. The ST author should select the standard to which the RBG services comply (either SP 800-90A or FIPS 140-2 Annex C).*

*SP 800-90A contains three different methods of generating random numbers; each of these, in turn, depends on underlying cryptographic primitives (hash functions/ciphers). The STauthor will select the function used (if SP 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.*

*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, Section 3 is valid. Use of this DRBG is disallowed after 2015 per NIST SP 800-131A. The PP will be updated to reflect this; however, developers should begin transitioning from this DRBG as soon as possible.*

***Assurance Activity***

**Requirement met by the TOE platform**

*TSS*

*For each platform listed in the ST, the evaluator shall examine the ST of the platform to ensure that the RBG functions claimed in that platform's ST contains the RBG functions in the MDM Server's ST. The evaluator shall also examine the TSS of the MDM Server's ST to verify that it describes (for each supported platform) how the RBG functionality is invoked for each operation they are used for in the MDM Server (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this assurance activity).*

**Requirement met by the TSF**

*TEST*

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

***Implementations Conforming to FIPS 140-2 Annex C***

*The reference for the tests contained in this section is The Random Number Generator Validation System (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.*

*Test 1: 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.*

*Test 2: 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***

*Test 1: The evaluator shall perform 15 trials for the RNG implementation. If the RNG 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 RNG functionality.*

*If the RNG has prediction resistance enabled, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) generate a second block of random bits (4) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. "generate one block of random bits" means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP 800-90A).*

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

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

> *Entropy input:* the length of the entropy input value must equal the seed length.*
>
> *Nonce:* If a nonce is supported (CTR_DRBG with no Derivation Function does not use a nonce), the nonce bit length is one-half the seed length.*

*<u>Personalization string:</u> The length of the personalization string must be less than or equal to seed length. If the implementation only supports one personalization string length, then the same length can be used for both values. If more than one string length is support, the evaluator shall use personalization strings of two different lengths. If the implementation does not use a personalization string, no value needs to be supplied.*

*<u>Additional input</u>: the additional input bit lengths have the same defaults and restrictions as the personalization string lengths.*

**FCS_RBG_EXT.1.2** The deterministic RBG shall be seeded by an entropy source that accumulates entropy from [selection: <u>a software-based noise source, a platform-based RBG , a hardware-based noise source, no other sources</u>] with a minimum of [selection: <u>128 bits, 256 bits</u>] of entropy at least equal to the greatest security strength (according to NIST SP 800-57) of the keys and hashes that it will generate.

*Application Note:*

*For the first selection in this requirement, the ST author selects 'software-based noise source' if any additional noise sources are used as input to the application's DRBG. Note that the application must use the platform's DRBG to seed its DRBG.*

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

***Assurance Activity:***

*Documentation shall be produced—and the evaluator shall perform the activities—in accordance with Appendix E: Entropy Documentation and Assessment and the "Clarification to the Entropy Documentation and Assessment Annex".*

*In the future, specific statistical testing (in line with NIST SP 800-90B) will be required to verify the entropy estimates.*

#### 4.4.2.9. Cryptographic Key Storage

**FCS_STG_EXT.1.1** The [selection: TSF, TOE platform] shall store persistent secrets and private keys when not in use, in [selection: platform-provided key storage, as specified in FCS_STG_EXT.2].

*Application Note:*

*This requirement ensures that persistent secrets (credentials, secret keys) and private keys are stored securely when not in use. If some secrets/keys are manipulated by the TOE and others are manipulated by the platform, then both of the selections can be specified by the ST author and the ST author must identify in the TSS those keys which are manipulated by the TOE and those by the platform.*

*If the TSF is an application, and not a dedicated server, then it should store its private keys in the platform-provided key storage.*

*The ST author is responsible for selecting the manner in which the keys are stored and where they are stored in the selections above.*

***Assurance Activity:***

*TSS*

*Regardless of whether this requirement is met by the TSF or the TOE platform, the evaluator will check the TSS to ensure that it lists each persistent secret (credential, secret key) and private key needed to meet the requirements in the ST. For each of these items, the evaluator will confirm that the TSS lists for what purpose it is used, and how it is stored. The evaluator than performs the following actions.*

**Persistent secrets and private keys manipulated by the TOE platform**

*For each platform listed in the ST, the evaluator shall examine the ST of the platform to ensure that the persistent secrets and private keys listed as being stored by the platform in the MDM Server ST are identified as being protected in that platform's ST.*

**Persistent secrets and private keys manipulated by the TSF**

*The evaluator reviews the TSS to determine that it makes a case that, for each item listed as being manipulated by the TOE, it is not written unencrypted to persistent memory, and that the item is stored by the platform.*

### 4.4.3. Identification and Authentication (FIA)

#### 4.4.3.1. Timing of Authentication

**FIA_UAU.1.1 Refinement**: The [selection: TSF, TOE platform] shall allow [assignment: *list of* **TSF** *mediated actions*] on behalf of the user to be performed before the user is authenticated **with the Server.**

**FIA_UAU.1.2 Refinement**: The [selection: TSF, TOE platform] shall require each user to be successfully authenticated **with the Server** before allowing any other **TSF**-mediated actions on behalf of that user.

*Application Note:*

*This requirement ensures that any user attempting to access the TSFTSF must be authenticated. These users may be administrators attempting to administer the TOE or ordinary users attempting to enroll for management by the MDM system. The ST author is responsible for assigning the list of actions that can take place before this authentication. The TSFTSF or TOETOE platform may utilize enterprise authentication to meet this requirement.*

***Assurance Activity:***

*TSS*

*The evaluator shall examine the TSS and verify that it describes the actions that can and cannot be performed before authentication.*

*TEST*

*The evaluator shall perform the following tests:*

*Test 1: The evaluator shall attempt to perform the prohibited actions before authentication. The evaluator shall verify the actions cannot be performed.*

*Test 2: The evaluator shall attempt to perform the prohibited actions after authentication. The evaluator shall verify the actions can be performed.*

#### 4.4.3.2. X509 Validation

**FIA_X509_EXT.1.1** The [selection: *TSF, TOE platform*] shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The TSF shall validate the revocation status of the certificate using [selection: the Online Certificate Status Protocol (OCSP) as specified in RFC 2560, a Certificate Revocation List (CRL) as specified in RFC 5759].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
    - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
    - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
    - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
    - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.
    - Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the extendedKeyUsage field.

*Application Note:*

*FIA_X509_EXT.1.1 lists the rules for validating certificates. The ST author shall select whether revocation status is verified using OCSP or CRLs. FIA_X509_EXT.2 requires that certificates are used for trusted channels; this use requires that the extendedKeyUsage rules are verified. Certificates may optionally be used for code signing and policy signing and, if implemented, must be validated to contain the corresponding extendedKeyUsage.*

*Regardless of the selection of TSF or TOE platform, the validation is expected to end in a trusted root CA certificate in a root store managed by the platform.*

***Assurance Activity:***

*TSS*

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

*TEST*

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

*Test 1: The evaluator shall load a certificate or certificates as trusted CAs needed to validate the certificate to be used in the function, and demonstrate that the function succeeds. The evaluator shall then delete one of the certificates, and show that the function fails.*

*Test 2: The evaluator shall demonstrate that validating an expired certificate results in the function failing.*

*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, then a test shall be performed for each method. The evaluator shall test revocation of the node certificate and revocation of the intermediate CA certificate (i.e. the intermediate CA certificate should be revoked by the root CA). The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails.*

*Test 4: If OCSP is selected, the evaluator shall configure the OCSP server or use a man-in-the-middle tool to present a certificate that does not have the OCSP signing purpose and verify that validation of the OCSP response fails. If CRL is selected, the evaluator shall configure the CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set, and verify that validation of the CRL fails.*

*Test 5: The evaluator shall modify any byte in the first eight bytes of the certificate and demonstrate that the certificate fails to validate. (The certificate will fail to parse correctly.)*

*Test 6: The evaluator shall modify any byte in the last byte of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)*

*Test 7: The evaluator shall modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)*

**FIA_X509_EXT.1.2** The [selection: *TSF, TOE platform*] shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

*Application Note:*

*This requirement applies to certificates that are used and processed by the TOE or platform and restricts the certificates that may be added as trusted CA certificates.*

***Assurance Activity:***

*TEST*

*The tests described must be performed in conjunction with the other certificate services assurance activities, including the functions in FIA_X509_EXT.2.1. The evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA.*

*Test 1: The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOE's certificate does not contain the basicConstraints extension. The validation of the certificate path fails.*

*Test 2: The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOE's certificate has the cA flag in the basicConstraints extension not set. The validation of the certificate path fails.*

*Test 3: The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOE's certificate has the cA flag in the basicConstraints extension set to TRUE. The validation of the certificate path succeeds.*

#### 4.4.3.3. X509 Authentication

**FIA_X509_EXT.2.1** The [selection: *TSF, TOE platform*] shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [selection*: IPsec, TLS, HTTPS, DTLS*]], and [selection: *code signing for system software updates, code signing for integrity verification, policy signing, no additional uses*].

*Application Note:*

*The ST author's selection shall match the selection of FTP_TRP.1, FTP_ITC.1, and FPT_ITT.1. Certificates may optionally be used for trusted updates of system software (FPT_TUD_EXT.1.3) and software integrity verification (FPT_TST_EXT.1.2). If FMT_POL_EXT.1 is included in the main body, policy signing must be selected. If some authentication services are provided by the TOE and others by the platform, the ST author shall clearly identify which services are provided by the TOE and which by the platform.*

**FIA_X509_EXT.2.2** When the [selection: *TSF, TOE platform*] cannot establish a connection to determine the validity of a certificate, the [selection: *TSF, TOE platform*] shall [selection: *allow the administrator to choose whether to accept the certificate in these cases, accept the certificate, not accept the certificate*].

*Application Note:*

*Often a connection must be established to perform a verification of the revocation status of a certificate - either to download a CRL or to perform OCSP. The selection is used to describe the behavior in the event that such a connection cannot be established (for example, due to a network error). If the TOE has determined the certificate valid according to all other rules in FIA_X509_EXT.1, the behavior indicated in the second selection shall determine the validity. The TOE must not accept the certificate if it fails any of the other validation rules in FIA_X509_EXT.1. If the administrator-configured option is selected by the ST Author, the ST Author must also select function d in FMT_SMF.1(2).*

***Assurance Activity***

*TSS*

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

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

*GUIDANCE*

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

*TEST*

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

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

***FIA_X509_EXT.2.3** The [selection: TSF, TOE platform] shall require a unique certificate for each client device.*

*Application Note:*

*Each client device will have a unique X.509v3 certificate for use by the MDM Agent; the certificate is not to be reused among clients. This requirement is to ensure that the MDM Server verifies that each client certificate is unique.*

*Assurance Activity:*

*TEST*

*For each MDM Agent/platform listed as supported in the ST:*

*The evaluator shall enroll a client device with the MDM Server using certificates. Then attempt to enroll a second device with the same client certificate, verifying that enrolment was unsuccessful.*

### 4.4.4. Protection of the TSF (FPT)

#### 4.4.4.1. TSF Testing

**FPT_TST_EXT.1.1** The [selection: MDM Server, MDM Server platform] shall run a suite of self tests during initial start-up (on power on) to demonstrate correct operation of the TSF.

**FPT_TST_EXT.1.2** The [selection: MDM Server, MDM Server platform] shall provide the capability to verify the integrity of stored TSF executable code when it is loaded for execution through the use of the [selection: TSF, TOE platform]-provided cryptographic services.

*Application Note:*

*While the TOE is typically a software package running in the IT Environment, it is still capable of performing the self-test activities required above. It should be understood, however, that there is a significant dependency on the host environment in assessing the assurance provided by the tests mentioned above (meaning that if the host environment is compromised, the self tests will not be meaningful).*

***Assurance Activity:***

***TSS***

*The evaluator shall examine the TSS to ensure that it details the self tests that are run by the TSF on start-up; this description should include an outline of what the tests are actually doing (e.g., rather than saying "memory is tested", a description similar to "memory is tested by writing a value to each memory location and reading it back to ensure it is identical to what was written" shall be used). The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly.*

*The evaluator shall examine the TSS to ensure that it describes how to verify the integrity of stored TSF executable code when it is loaded for execution. The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the integrity of stored TSF executable code has not been compromised. The evaluator also ensures that the TSS (or the operational guidance) describes the actions that take place for successful (e.g. hash verified) and unsuccessful (e.g., hash not verified) cases.*

*TEST*

*The evaluator shall perform the following tests:*

*Test 1: The evaluator performs the integrity check on a known good TSF executable and verifies that the check is successful.*

*Test 2: The evaluator modifies the TSF executable, performs the integrity check on the modified TSF executable and verifies that the check fails.*

#### 4.4.4.2. Trusted Update

**FPT_TUD_EXT.1.2** The [selection: MDM Server, MDM Server platform] shall provide Authorized Administrators the ability to initiate updates to TSF software.

**FPT_TUD_EXT.1.3** The [selection: MDM Server, MDM Server platform] shall provide a means to verify software updates to the TSF using a digital signature mechanism prior to installing those updates.

*Application Note:*

*The software on the TSF will occasionally need to be updated. This requirement is intended to ensure that the TSF only installs updates provided by the vendor, as updates provided by another source may contain malicious code. If the server is not an appliance, the update will be verified by the platform on which the server software runs. If the server is an appliance, the update must be verified by the TSF software or hardware.*

***Assurance Activity:***

***TSS***

*The evaluator shall examine the TSS and verify that it describes the standards by which the updates are digitally signed and how the signature verification process is implemented.*

*GUIDANCE*

*The evaluator shall examine the AGD guidance to verify that it describes how to query the current version of the TSF software and how to initiate updates.*

*TEST*

*The evaluator shall perform the following tests:*

*Test 1: The evaluator shall attempt to initiate an update digitally signed by the vendor and verify that the update is successfully installed.*

*Test 2: The evaluator shall attempt to install an update not digitally signed by the vendor and verify that the update is not installed.*

### 4.4.5. Trusted Path/Channels (FTP)

#### 4.4.5.1. Inter-TSF Trusted Channel (Authorized IT Entities)

**FTP_ITC.1.1(1) Refinement:** The [selection: **MDM Server, MDM Server platform**] shall **use [selection: IPsec, SSH, TLS, TLS/HTTPS]** to provide a **trusted** communication channel between itself and **authorized IT entities supporting the following capabilities: audit server, [selection: authentication server, [assignment: *other capabilities*]]** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data **from disclosure and detection of modification of the channel data**.

*Application Note:*

*The intent of the mandatory portion of the above requirement is to use the cryptographic protocols identified in the requirement to establish and maintain a trusted channel with authorized IT entities that the TOE interacts with to perform its functions.*

*Protection (by one of the listed protocols) is required at least for communications with the server that collects the audit information. If it communicates with an authentication server (e.g., RADIUS), then the ST author chooses "authentication server" in FTP_ITC.1.1(1) and this connection must be protected by one of the listed protocols. If other authorized IT entities (e.g., NTP server) are protected, the ST author makes the appropriate assignments (for those entities) and selections (for the protocols that are used to protect those connections). After the ST author has made the selections, they are to select the detailed requirements in Appendix C corresponding to their protocol selection to put in the ST. To summarize, the connection to an external audit collection server is required to be protected by one of the listed protocols. If an external authentication server is supported, then it is required to protect that connection with one of the listed protocols. For any other external server, external communications are not required to be protected, but if protection is claimed, then it must be protected with one of the identified protocols.*

*The trusted channel uses IPsec, TLS, DTLS, or HTTPS as the protocol that preserves the confidentiality and integrity of MDM communications. The ST author chooses the mechanism or mechanisms supported by the TOE, and then ensures the detailed requirements in Appendix C corresponding to their selection are copied to the ST if not already present.*

*Protocol, RBG, Certificate validation, algorithm, and similar services may be met with platform provided services.*

*The requirement implies that not only are communications protected when they are initially established, but also on resumption after an outage. It may be the case that some part of the TOE setup involves manually setting up tunnels to protect other communication, and if after an outage the TOE attempts to re-establish the communication automatically with (the necessary) manual intervention, there may be a window created where an attacker might be able to gain critical information or compromise a connection.*

**FTP_ITC.1.2(1)** The TSF shall permit the **MDM Server or other authorized IT entities** to initiate communication via the trusted channel.

**FTP_ITC.1.3(1)** The TSF shall initiate communication via the trusted channel for **[assignment: list of services for which the TSF is able to initiate communications].**

*Application Note:*

*While there are no requirements on the party initiating the communication, the ST author lists in the assignment for FTP_ITC.1.3 the services for which the TOE can initiate the communication with the authorized IT entity.*

***Assurance Activity:***

*TSS*

*The evaluator shall examine the TSS to determine that the methods of communication with authorized IT entities are indicated, along with how those communications are protected.*

*GUIDANCE*

*The evaluator shall confirm that the operational guidance contains instructions for configuring the communication channel between the MDM Server and authorized IT entities for each supported method.*

*TEST*

*Test 1: The evaluators shall ensure that communications using each specified (in the operational guidance) communication method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.*

*Test 2: The evaluator shall ensure, for each method of communication, the channel data is not sent in plaintext.*

*Test 3: The evaluator shall ensure, for each communication channel with the MDM Server, that a protocol analyzer identifies the traffic as the protocol under testing.*

*Further assurance activities are associated with the specific protocols.*

#### 4.4.5.2. Trusted Path for Remote Administration

**FTP_TRP.1.1(1) Refinement:** The [selection: **MDM Server, MDM Server platform**] shall **use [selection: IPsec, TLS, TLS/HTTPS, SSH] to** provide a trusted communication path between itself and **remote administrators** that is logically distinct from other communication paths and provides assured identification of its endpoints and protection of the communicated data from *disclosure and detection of modification of the communicated data.*

**FTP_TRP.1.2(1) Refinement:** The [selection: **MDM Server, MDM Server platform**] shall permit **remote administrators** to initiate communication via the trusted path.

**FTP_TRP.1.3(1) Refinement:** The [selection: **MDM Server, MDM Server platform**] shall require the use of the trusted path for all remote administration actions.

*Application Note:*

*This requirement ensures that authorized remote administrators initiate all communication with the TOE via a trusted path, and that all communications with the TOE by remote administrators is performed over this path. The data passed in this trusted communication channel are encrypted as defined the protocol chosen in the first selection. The ST author chooses the mechanism or mechanisms supported by the TOE, and then ensures the detailed requirements in Appendix C corresponding to their selection are copied to the ST if not already present.*

***Assurance Activity:***

*TSS*

*The evaluator shall examine the TSS to determine that the methods of remote TOE administration are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST.*

*GUIDANCE*

*The evaluator shall confirm that the operational guidance contains instructions for establishing the remote administrative sessions for each supported method.*

*TEST*

*The evaluator shall also perform the following tests:*

*Test 1: The evaluators shall ensure that communications using each specified (in the operational guidance) remote administration method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.*

*Test 2: For each method of remote administration supported, the evaluator shall follow the operational guidance to ensure that there is no available interface that can be used by a remote user to establish remote administrative sessions without invoking the trusted path.*

*Test 3: The evaluator shall ensure, for each method of remote administration, the channel data is not sent in plaintext.*

*Further assurance activities are associated with the specific protocols.*

#### 4.4.5.3. Trusted Path for Enrollment

**FTP_TRP.1.1(2) Refinement:** The [selection: **MDM Server, MDM Server platform**] shall **use [selection: TLS, TLS/HTTPS] to** provide a trusted communication path between itself and **MD users** that is logically distinct from other communication paths and provides assured identification of its endpoints and protection of the communicated data from *disclosure and detection of modification of the communicated data.*

**FTP_TRP.1.2(2) Refinement:** The [selection: **MDM Server, MDM Server platform**] shall permit **MD users** to initiate communication via the trusted path.

**FTP_TRP.1.3(2) Refinement:** The [selection: **MDM Server, MDM Server platform**] shall require the use of the trusted path for all MD user actions.

*Application Note:*

*This requirement ensures that authorized MD users initiate all communication with the TOE via a trusted path, and that all communications with the TOE by MD users is performed over this path. The purpose of this connection is for enrollment by the MD user. The data passed in this trusted communication channel are encrypted as defined the protocol chosen in the first selection. The ST author chooses the mechanism or mechanisms supported by the TOE, and then ensures the detailed requirements in Appendix C corresponding to their selection are copied to the ST if not already present.*

***Assurance Activity:***

*TSS*

*The evaluator shall examine the TSS to determine that the methods of remote enrollment are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of enrollment are consistent with those specified in the requirement, and are included in the requirements in the ST.*

*GUIDANCE*

*The evaluator shall confirm that the operational guidance contains instructions for establishing the enrollment sessions for each supported method.*

*TEST*

*For each MDM Agent/platform listed as supported in the ST:*

*Test 1: The evaluators shall ensure that communications using each specified (in the operational guidance) enrollment method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.*

*Test 2: For each method of enrollment supported, the evaluator shall follow the operational guidance to ensure that there is no available interface that can be used by a remote user to establish enrollment sessions without invoking the trusted path.*

*Test 3: The evaluator shall ensure, for each method enrollment, the channel data is not sent in plaintext.*

*Further assurance activities are associated with the specific protocols.*

# 5. SECURITY ASSURANCE REQUIREMENTS

The Security Objectives for the TOE in Section 3 were constructed to address threats identified in Section 2.1. The Security Functional Requirements (SFRs) in Section 4 are a formal instantiation of the Security Objectives. The PP identifies the Security Assurance Requirements (SARs) to frame the extent to which the evaluator assesses the documentation applicable for the evaluation and performs independent testing.

This section lists the set of SARs from CC part 3 that are required in evaluations against this PP. Individual Assurance Activities (Assurance Activities) to be performed are specified both in Section 4 as well as in this section.

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

After the ST has been approved for evaluation, the ITSEF will obtain the TOE, supporting environmental IT, and the administrative/user guides for the TOE. The ITSEF is expected to perform actions mandated by the Common Evaluation Methodology (CEM) for the ASE and ALC SARs. The ITSEF also performs the Assurance Activities contained within Section 4, which are intended to be an interpretation of the other CEM assurance requirements as they apply to the specific technology instantiated in the TOE. The Assurance Activities that are captured in Section 4 also provide clarification as to what the developer needs to provide to demonstrate the TOE is compliant with the PP.

The TOE security assurance requirements are identified in Table 2.

**Table 2 Security Assurance Requirements**

| Assurance Class | Assurance Components |
|---|---|
| Security Target (ASE) | ST introduction (ASE_INT.1) |
| | Conformance claims (ASE_CCL.1) |
| | Security objectives for the operational environment (ASE_OBJ.1) |
| | Extended components definition (ASE_ECD.1) |
| | Stated security requirements (ASE_REQ.1) |
| | TOE summary specification (ASE_TSS.1) |
| Development (ADV) | Basic functional specification (ADV_FSP.1) |
| Guidance documents (AGD) | Operational user guidance (AGD_OPE.1) |
| | Preparative procedures (AGD_PRE.1) |
| Life cycle support (ALC) | Labeling of the TOE (ALC_CMC.1) |
| | TOE CM coverage (ALC_CMS.1) |
| Tests (ATE) | Independent testing – sample (ATE_IND.1) |
| Vulnerability assessment (AVA) | Vulnerability survey (AVA_VAN.1) |

## 5.1. Class ASE: Security Target

The ST is evaluated as per ASE activities defined in the CEM. In addition, there may be Assurance Activities specified within Section 4 that call for necessary descriptions to be included in the TSS that are specific to the TOE technology type.

## 5.2. Class ADV: Development

The design information about the TOE is contained in the guidance documentation available to the end user as well as the TSS portion of the ST, and any additional information required by this PP that is not to be made public (e.g., Entropy Essay).

### 5.2.1. Basic Functional Specification (ADV_FSP)

The functional specification describes the Target Security Functions Interfaces (TSFIs). 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, 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.

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.

*Application Note:*

*As indicated in the introduction to this section, the functional specification is comprised of the information contained in the AGD_OPE and AGD_PRE documentation.*

*The developer may reference a website accessible to application developers and the evaluator.*

*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:***

*There are no specific assurance activities associated with these SARs, except ensuring the information is provided. The functional specification documentation is provided to support the evaluation activities described in Sections 4.2, 4.3, and 4.4, 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.*

## 5.3. Class AGD: Guidance Documentation

The guidance documents will be provided with the ST. Guidance must include a description of how the IT personnel verifies that the Operational Environment can fulfill its role for the security functionality. The documentation should be in an informal style and readable by the IT personnel.

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 TSF in that environment; and
- instructions to manage the security of the TSF as a product and as a component of the larger operational environment; and
- instructions to provide a protected administrative capability.

Guidance pertaining to particular security functionality must also be provided; requirements on such guidance are contained in the assurance activities specified with each requirement.

### 5.3.1. Operational User Guidance (AGD_OPE)

**Developer action elements:**

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

*Application Note:*

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

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

**Content and presentation elements:**

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

*Application Note:*

*User and administrator are to be considered in the definition of user role.*

AGD_OPE.1.2C The operational user guidance shall describe, for each user role, 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 each user role, 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 each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

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

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:***

*Some of the contents of the operational guidance will be verified by the assurance activities in Sections 4.2, 4.3, and 4.4and evaluation of the TOE according to the CEM. The following additional information is also required.*

*If cryptographic functions are provided by the TOE, 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.*

*The documentation must describe the process for verifying updates to the TOE by verifying a digital signature – this may be done by the TOE or the underlying platform. The evaluator shall verify that this process includes the following steps:*

*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).*

*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.*

*The TOE will likely contain security functionality that does not fall in the scope of evaluation under this PP. The operational guidance shall make it clear to an administrator which security functionality is covered by the evaluation activities.*

### 5.3.2. Preparative Procedures (AGD_PRE)

**Developer action elements:**

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

*Application Note:*

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

**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:***

*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 claimed for the TOE in the ST.*

## 5.4. Class ALC: Life-cycle Support

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.

### 5.4.1. Labeling of the TOE (ALC_CMC)

This component is targeted at identifying the TOE such that it can be distinguished from other products or versions 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.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

***Assurance Activity:***

*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.*

### 5.4.2. TOE CM Coverage (ALC_CMS)

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.1.1D The developer shall provide a configuration list for the TOE.

**Content and presentation elements:**

ALC_CMS.1.1C The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.

ALC_CMS.1.2C The configuration list shall uniquely identify the configuration items.

**Evaluator action elements:**

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

***Assurance Activity:***

*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.*

*Life-cycle support is targeted aspects of the developer's life-cycle and instructions to providers of applications for the developer's devices, rather than an in-depth examination of the TSF manufacturer'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.*

***Assurance Activity:***

*The evaluator shall ensure that the developer has identified (in public-facing development documentation for their platform) one or more development environments appropriate for use in developing applications for the developer's platform. For each of these development environments, the developer shall provide information on how to configure the environment to ensure that buffer overflow protection mechanisms in the environment(s) are invoked (e.g., compiler flags). The evaluator shall ensure that this documentation also includes an indication of whether such protections are on by default, or have to be specifically enabled.*

*The evaluator shall ensure that the TSF is uniquely identified (with respect to other products from the TSF vendor), and that documentation provided by the developer in association with the requirements in the ST is associated with the TSF using this unique identification.*

### 5.4.3. Class ATE: Tests

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 the 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.

Since many of the APIs are not exposed at the user interface (e.g., touch screen), the ability to stimulate the necessary interfaces requires a developer's test environment. This test environment will allow the evaluator, for example, to access APIs and view file system information that is not available on consumer mobile devices.

### 5.4.4. Independent Testing – Conformance (ATE_IND)

Testing is performed to confirm the functionality described in the TSS as well as the administrative (including configuration and operational) documentation provided. The focus of the testing is to confirm that the requirements specified in Sections 4.2, 4.3, and 4.4 being met, although some additional testing is specified for SARs in Section 4.5. 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:***

*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 evaluator must document in the test plan that each applicable testing requirement in the ST is covered.*

*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.*

*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 evaluator is 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) should be 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, TLS/HTTPS, SSH).*

*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.*

## 5.5. Class AVA: Vulnerability Assessment

For the first generation of this protection profile, the evaluation lab is expected to survey open sources to discover 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, the evaluator 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.

### 5.5.1. Vulnerability Survey (AVA_VAN)

**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:***

*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 network infrastructure devices and the implemented communication protocols 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. 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.*

# A. RATIONALE

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 MDM 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 Appendix contains the tabular artifacts that can be used for the evaluation activities associated with this document.

## A.1. Security Problem Definition

### A.1.1. Assumptions

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.

**Table 3: TOE Assumptions**

| **Assumption Name** | **Assumption Name** |
|---|---|
| A.CONNECTIVITY | The TOE relies on network connectivity to carry out its management activities. The TOE will robustly handle instances when connectivity is unavailable or unreliable. |
| A.MDM_SERVER_PLATFORM | The MDM Server relies upon a trustworthy platform and local network from which it provides administrative capabilities.<br><br>The MDM server relies on the this platform to provide a range of security-related services including reliable timestamps, user and group account management, logon and logout services via a local or network directory service, remote access control, and audit log management services to include offloading of audit logs to other servers. The platform is expected to be configured specifically to provide MDM services, employing features such as a host-based firewall, which limits its network role to providing MDM functionality. |
| A.PROPER_ADMIN | One or more competent, trusted personnel who are not careless, willfully negligent, or hostile, are assigned and authorized as the TOE Administrators, and do so using and abiding by guidance documentation. |
| A.PROPER_USER | Mobile device users are not willfully negligent or hostile, and use the device within compliance of a reasonable Enterprise security policy. |

### A.1.2. Threats

The threats to the mobile device listed below are addressed by Mobile Device Management systems

**Table 4: Threats**

| **Threat** | **Description of Threat** |
|---|---|
| T.MALICIOUS_APPS | An administrator of the MDM or mobile device user may inadvertently import malicious code, or an attacker may insert |

| | |
|---|---|
| | malicious code into the TOE or OE, resulting in the compromise of TOE or TOE data. |
| T.NETWORK_ATTACK | An attacker may masquerade as MDM Server and attempt to compromise the integrity of the mobile device by sending malicious management commands. |
| T.NETWORK_EAVESDROP | Unauthorized entities may intercept communications between the MDM and mobile devices to monitor, gain access to, disclose, or alter remote management commands. Unauthorized entities may intercept unprotected wireless communications between the mobile device and the Enterprise to monitor, gain access to, disclose, or alter TOE data. |
| T.PHYSICAL_ACCESS | The mobile device may be lost or stolen, and an unauthorized individual may attempt to access OE data. |

### A.1.3. Organizational Security Policies

An organizational security policy is a set of rules, practices, and procedures imposed by an organization to address its security needs. The following OSPs must be enforced by the TOE or its operational environment.

**Table 5: Organizational Security Policies**

| Policy Name | Policy Definition |
|---|---|
| P.ADMIN | The configuration of the mobile device security functions must adhere to the Enterprise security policy. |
| P.DEVICE_ENROLL | A mobile device must be enrolled for a specific user by the administrator of the MDM prior to being used in the Enterprise network by the user. |
| P.NOTIFY | The mobile user must immediately notify the administrator if a mobile device is lost or stolen so that the administrator may apply remediation actions via the MDM system. |
| P.ACCOUNTABILITY | Personnel operating the TOE shall be accountable for their actions within the TOE. |

## A.2. Security Objectives

### A.2.1. Security Objectives For the TOE

The following table identifies security objectives for the Mobile Device Management system.

**Table 6: Security Objectives for the TOE**

| Objective | Objective Description |
|---|---|
| O.APPLY_POLICY | The TOE must facilitate configuration and enforcement of enterprise security policies on mobile devices via interaction with the MDM Agent. This will include the initial enrollment of the device into management, through its lifecycle including policy updates and through its possible unenrollment from management services. |
| O.ACCOUNTABILITY | The TOE must provide logging facilities which record management actions undertaken by its administrators |

| | |
|---|---|
| O.DATA_PROTECTION_TRANSIT | Data exchanged between the MDM Server and the MDM Agent and between the MDM Server and its operating environment must be protected from being monitored, accessed and altered. |
| O.MANAGEMENT | The TOE provides access controls around its management functionality. |
| O.INTEGRITY | The TOE will provide the capability to perform self-tests to ensure the integrity of critical functionality, software/firmware and data has been maintained. The TOE will also provide a means to verify the integrity of downloaded updates. |

### A.2.2. Security Objectives for the Operational Environment

The following table contains objectives for the Operational Environment.

**Table 7: Security Objectives for the Operational Environment**

| Objective | Objective Description |
|---|---|
| OE.IT_ENTERPRISE | The Enterprise IT infrastructure provides security for a network that is available to the TOE and mobile devices that prevents unauthorized access. |
| OE.MDM_SERVER_PLATFORM | The MDM Server relies upon a trustworthy platform and local network from which it provides administrative capabilities. |
| OE.PROPER_ADMIN | TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner. |
| OE.PROPER_USER | Users of the mobile device are trained to securely use the mobile device and apply all guidance in a trusted manner. |
| OE.WIRELESS_NETWORK | A wireless network will be available to the mobile devices. |
| OE.TIMESTAMP | Reliable timestamp is provided by the operational environment for the TOE. |

## A.3. Correspondence

### A.3.1. Security Problem Definition Correspondence

The following table serves to map the threats and assumptions defined in this PP to the security objectives also defined or identified in this PP.

**Table 8: Security Problem Definition Correspondence**

| Threat or Assumption | Security Objective |
|---|---|
| A.CONNECTIVITY | OE.WIRELESS_NETWORK |
| A.MDM_SERVER_PLATFORM | OE.MDM_SERVER_PLATFORM, OE.TIMESTAMP |
| A.PROPER_ADMIN | OE.PROPER_ADMIN, |
| A.PROPER_USER | OE.PROPER_USER |
| T.MALICIOUS_APPS | O.APPLY_POLICY, O.INTEGRITY |
| T.NETWORK_ATTACK | O.DATA_PROTECTION_TRANSIT |
| T.NETWORK_EAVESDROP | O.DATA_PROTECTION_TRANSIT |
| T.PHYSICAL_ACCESS | O.APPLY_POLICY |
| P.ADMIN | OE.PROPER_ADMIN, O.MANAGEMENT |
| P.DEVICE_ENROLL | O.MANAGEMENT, OE.IT_ENTERPRISE |

| | |
|---|---|
| P.NOTIFY | OE.PROPER_USER |
| P.ACCOUNTABILITY | O.ACCOUNTABILITY |

### A.3.2. Security Objective Correspondence

The correspondence between the Security Functional Requirements (SFRs) and the Security Objectives identified or defined in the PP is provided in Section 3.1

# B. OPTIONAL REQUIREMENTS

As indicated in the introduction to this PP, the baseline requirements (those that must be performed by the TOE or its underlying platform) are contained in the body of this PP. Additionally, there are three other types of requirements specified in Appendices B, C, and D.

The first type (in this Appendix) are requirements that can be included in the ST, but do not have to be in order for a TOE to claim conformance to this PP. The second type (in Appendix C) are requirements based on selections in the body of the PP: if certain selections are made, then additional requirements in that appendix will need to be included. The third type (in Appendix D) are components that are not required in order to conform to this PP, but will be included in the baseline requirements in future versions of this PP, so adoption by MDM vendors is encouraged. Note that the ST author is responsible for ensuring that requirements that may be associated with those in Appendix B, Appendix C, and/or Appendix D but are not listed (e.g., FMT-type requirements) are also included in the ST.

This Appendix is divided into three subsections: optional requirements that may be performed by the TSF, optional requirements that may be performed by the MDM Server or its underlying platform, and optional requirements to support a Mobile Application Store Server.

## B.1. Optional TSF Requirements

### B.1.1. Security Audit (FAU)

#### B.1.1.1 Security Audit Event Selection

**FAU_SEL.1.1** The **MDM Server** shall be able to select the set of events to be audited from the set of all auditable events based on the following attributes:

a. **event type;**
b. **success of auditable security events;**
c. **failure of auditable security events; and**
d. **[assignment: other attributes].**

*Application Note:*

*The intent of this requirement is to identify all criteria that can be selected to trigger an audit event. The ST author must select whether the TSF or the platform maintains the audit record. For the ST author, the assignment is used to list any additional criteria or "none".*

***Assurance Activity:***

*GUIDANCE*

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

*TEST*

*The evaluator shall also perform the following tests:*

*Test 1: For each attribute listed in the requirement, the evaluator shall devise a test to show that selecting the attribute causes only audit events with that attribute (or those that are always recorded, as identified in the administrative guidance) to be recorded.*

*Test 2: [conditional] If the TSF supports specification of more complex audit pre-selection criteria (e.g., multiple attributes, logical expressions using attributes) then the evaluator shall devise tests showing that this capability is correctly implemented. The evaluator shall also, in the test plan, provide a short narrative justifying the set of tests as representative and sufficient to exercise the capability.*

### B.1.2. Protection of the TSF (FPT)

#### B.1.2.1 Basic Internal TSF Data Transfer Protection (MDM Server)

**FPT_ITT.1.1(1) Refinement:** The TSF shall protect **all data** from disclosure and modification **through use of [selection: TLS, HTTPS, DTLS]** when it is transferred between **the MDM Agent and MDM Server.**

*Application Note:*

*This requirement is to ensure that the transmission of any audit logs, mobile device information data (software version, hardware model, and application versions), and configuration data collected by the MDM Agent and sent from the MDM Agent to the MDM Server, when commanded, or at configurable intervals, is properly protected. This trusted channel also protects any commands and policies sent by the MDM Server to the MDM Agent. Either the MDM Agent or the MDM Server is able to initiate the connection.*

*The trusted channel uses secure protocols that preserve the confidentiality and integrity of MDM communications. The ST author chooses the mechanism or mechanisms supported by the TOE, and then ensures the detailed requirements in Appendix C: Selection-Based Requirements corresponding to their selection are copied to the ST if not already present.*

*Protocol, RBG, Certificate validation, algorithm, and similar services may be met with platform provided services.*

***Assurance Activity:***

*TSS*

*The evaluator shall examine the TSS to determine that for each supported Agent listed in the ST, the methods of Agent-Server communication are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of remote TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST.*

*GUIDANCE*

*The evaluator shall confirm that the operational guidance contains instructions for configuring the communication channel between each supported MDM Agent and the MDM Server for each supported method.*

*TEST*

*For each supported MDM Agent/platform listed in the ST:*

*Test 1: The evaluators shall ensure that communications using each specified (in the operational guidance) Agent-Server communication method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.*

*Test 2: The evaluator shall ensure, for each method of Agent-Server communication, the channel data is not sent in plaintext.*

*Further assurance activities are associated with the specific protocols.*

### B.1.3. TOE Access (FTA)

#### B.1.3.1 Default TOE Access Banners

**FTA_TAB.1.1** Before establishing a user session, the [selection: MDM Server, MDM Server *platform*] shall display an Administrator-specified advisory notice and consent warning message regarding use of the TOE.

***Assurance Activity:***

*TSS*

*The TSS shall describe when the banner is displayed. The evaluator shall also perform the following test:*

*TEST*

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

### B.1.4. Trusted Path/Channels (FTP)

#### B.1.4.1 Inter-TSF Trusted Channel (MDM Agent)

**FTP_ITC.1.1(2)** The TSF shall **use [selection: *TLS, DTLS, HTTPS*]** to provide a **trusted** communication channel between itself and another trusted IT product that is logically distinct from other communication channels, provides assured identification of its end points, protects channel data from disclosure, and detects modification of the channel data.

*Application Note:*

*The intent of the mandatory portion of the above requirement is to use the cryptographic protocols identified in the requirement to establish and maintain a trusted channel between the TOE and the MDM Agent. Only TLS, DTLS, or HTTPS are used in this trusted channel.*

*This requirement is to ensure that the transmission of any audit logs, mobile device information data (software version, hardware model, and application versions), and configuration data collected by the MDM Agent and sent from the MDM Agent to the MDM Server, when commanded, or at configurable intervals, is properly protected. This trusted channel also protects any commands and policies sent by the MDM Server to the MDM Agent. Either the MDM Agent or the MDM Server is able to initiate the connection.*

*This trusted channel protects both the connection between an enrolled MDM Agent and the MDM Server and the connection between an unenrolled MDM Agent and the MDM Server during the enrollment operation. Different protocols can be used for these two connections, and the description in the TSS should make this difference clear.*

*The trusted channel uses TLS, DTLS, or HTTPS as the protocol that preserves the confidentiality and integrity of MDM communications. The ST author chooses the mechanism or mechanisms supported by the TOE, and then ensures the detailed requirements in Appendix C corresponding to their selection are copied to the ST if not already present.*

*Protocol, RBG, Certificate validation, algorithm, and similar services may be met with platform provided services.*

**FTP_ITC.1.2(2)** The TSF shall permit the **TSF and MDM Agent** to initiate communication via the trusted channel.

**FTP_ITC.1.3(2)** The TSF shall initiate communication via the trusted channel for **all communication between the MDM Server and the MDM Agent and [selection: <u>all communication between the MAS Server and the MDM Agent, no other communication</u>].**

*Application Note:*

*If the TOE includes a separate MAS Server, this requirement also addresses the communication between the MAS Server and the MDM Agent.*

***Assurance Activity:***

*TSS*

*The evaluator shall examine the TSS to determine that the methods of Agent-Server communication are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of remote TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST.*

*GUIDANCE*

*The evaluator shall confirm that the operational guidance contains instructions for configuring the communication channel between the MDM Agent and the MDM Server for each supported method.*

*TEST*

*Test 4: The evaluators shall ensure that communications using each specified (in the operational guidance) Agent-Server communication method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.*

*Test 5: The evaluator shall ensure, for each method of Agent-Server communication, the channel data is not sent in plaintext.*

*Test 6: The evaluator shall ensure, for each communication channel with the MDM Server, that a protocol analyzer identifies the traffic as the protocol under testing.*

*Further assurance activities are associated with the specific protocols.*

## B.2. Optional TOE or Platform Requirements

### B.2.1. Security Audit (FAU)

#### B.2.1.1 Audit Review

**FAU_SAR.1.1 Refinement:** The <u>[selection: **MDM Server, MDM Server platform**]</u> shall provide **Authorized Administrators** with the capability to read **all audit data** from the audit records.

**FAU_SAR.1.2 Refinement:** The <u>[selection: **MDM Server, MDM Server platform**]</u> shall provide the audit records in a manner suitable for the Authorized Administrators to interpret the information.

*Application Note:*

*The intent of this requirement is to ensure that the administrator can view and interpret the audit records and to prevent unauthorized users from accessing the logs.*

***Assurance Activity:***

*GUIDANCE*

*The evaluator shall check the AGD guidance and ensure that it describes how the administrator accesses the audit data and describes the format of the audit record.*

*TEST*

*The evaluator shall attempt to view the audit record as the authorized administrator and verify that the action succeeds. The evaluator shall ensure the audit records generated during testing match the format specified in the administrative guide.*

#### B.2.1.2 Audit Event Storage

**FAU_STG_EXT.2.1** The [<u>selection: MDM Server, MDM Server platform</u>] shall protect the stored audit records in the audit trail from unauthorized modification.

*Application Note:*

*The purpose of this requirement is to ensure that audit records are stored securely. The ST author is responsible for selecting whether audit records are maintained when audit storage or failure occurs. The ST author must choose a means by which audit records are saved and select the events during which the records will be saved. The TSF may rely on the underlying operating system for this functionality, and the first selection should be made appropriately.*

***Assurance Activity:***

*TSS*

*The evaluator shall ensure that the TSS describes how the audit records are protected from unauthorized modification or deletion. The evaluator shall ensure that the TOE uses audit trail specific protection mechanisms.*

*TEST*

*The evaluator shall perform the following tests:*

*Test 1: The evaluator shall access the audit trail as an unauthorized user and attempt to modify and delete the audit records. The evaluator shall verify that these attempts fail.*

*Test 2: The evaluator shall access the audit trail as an authorized user and attempt to modify and delete the audit records. The evaluator shall verify that these attempts succeed. The evaluator shall verify that only the records intended for modification and deletion are modified and deleted.*

### B.2.2. Cryptographic Support (FCS)

**FCS_TLSC_EXT.1.1** The [selection: <u>TSF, TOE platform</u>] shall implement [selection: <u>TLS 1.0 (RFC 3246), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)</u>] supporting the following ciphersuites: [

- Mandatory Ciphersuites:
  - TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246
- [selection: Optional Ciphersuites:
  - TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246

- TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- no other ciphersuite]].

*Application Note:*

*The TLS Client is required for MDM Agents in the TOE and may be included in MDM Servers in order to support Enrollment over Secure Transport (Appendix D.2.2.2).*

*The ciphersuites to be tested in the evaluated configuration are limited by this requirement. The ST author should select the optional ciphersuites that are supported; if there are no ciphersuites supported other than the mandatory suites, then "None" should be selected. It is necessary to limit the ciphersuites that can be used in an evaluated configuration administratively on the server in the test environment. The Suite B algorithms listed above (RFC 6460) are the preferred algorithms for implementation. TLS_RSA_WITH_AES_128_CBC_SHA is required in order to ensure compliance with RFC 5246.*

*These requirements will be revisited as new TLS versions are standardized by the IETF.*

*If any ciphersuites are selected using ECDHE, then FCS_TLSC_EXT.1.5 is required.*

***Assurance Activity:***

*TSS*

*The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the ciphersuites supported are specified. The evaluator shall check the TSS to ensure that the ciphersuites specified include those listed for this component. The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS. If the Agent supports multiple platforms, the ST shall make clear any differences in the TLS implementation.*

*TEST*

*Test 1: The evaluator shall establish a TLS connection using each of the ciphersuites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an EAP session. It is sufficient to observe the successful negotiation of a ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).*

*Test 2: The evaluator shall attempt to establish the connection using a server with a server certificate that contains the Server Authentication purpose in the extendedKeyUsage field and verify that a* *connection is established. The evaluator will then verify that the client rejects an otherwise valid server certificate that lacks the Server Authentication purpose in the extendedKeyUsage field and a connection is not established. Ideally, the two certificates should be identical except for the extendedKeyUsage field.*

*Test 3: The evaluator shall send a server certificate in the TLS connection that the does not match the server-selected ciphersuite (for example, send a ECDSA certificate while using the TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite or send a RSA certificate while using one of the ECDSA ciphersuites.) The evaluator shall verify that the TOE disconnects after receiving the server's Certificate handshake message.*

*Test 4: The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the client denies the connection.*

*Test 5: The evaluator shall perform the following modifications to the traffic:*

- *Change the TLS version selected by the server in the Server Hello to a non-supported TLS version (for example 1.3 represented by the two bytes 03 04) and verify that the client rejects the connection.*
- *Modify at least one byte in the server's nonce in the Server Hello handshake message, and verify that the client rejects the Server Key Exchange handshake message (if using a DHE or ECDHE ciphersuite) or that the server denies the client's Finished handshake message.*
- *Modify the server's selected ciphersuite in the Server Hello handshake message to be a ciphersuite not presented in the Client Hello handshake message. The evaluator shall verify that the client rejects the connection after receiving the Server Hello.*
- *Modify the signature block in the Server's Key Exchange handshake message, and verify that the client rejects the connection after receiving the Server Key Exchange message.*
- *Modify a byte in the Server Finished handshake message, and verify that the client sends a fatal alert upon receipt and does not send any application data.*
- *Send a garbled message from the Server after the Server has issued the ChangeCipherSpec message and verify that the client denies the connection.*

**FCS_TLSC_EXT.1.2** The [selection: <u>TSF, TOE platform</u>] shall verify that the presented identifier matches the reference identifier according to RFC 6125.

*Application Note:*

*The rules for verification of identify are described in Section 6 of RFC 6125. The reference identifier is established by the user (e.g. entering a URL into a web browser or clicking a link), by configuration (e.g. configuring the name of a mail server or authentication server), or by an application (e.g. a parameter of an API) depending on the application service. Based on a singular reference identifier's source domain and application service type (e.g. HTTP, SIP, LDAP), the client establishes all reference identifiers which are acceptable, such as a Common Name for the Subject Name field of the certificate and a (case-insensitive) DNS name, URI name, and Service Name for the Subject Alternative Name field. The client then compares this list of all acceptable reference identifiers to the presented identifiers in the TLS server's certificate.*

*The preferred method for verification is the Subject Alternative Name using DNS names, URI names, or Service Names. Verification using the Common Name is required for the purposes of backwards compatibility. Additionally, support for use of IP addresses in the Subject Name or Subject Alternative* *name is discouraged as against best practices but may be implemented. Finally, the client should avoid constructing reference identifiers using wildcards. However, if the presented identifiers include wildcards, the client must follow the best practices regarding matching; these best practices are captured in the assurance activity.*

***Assurance Activity:***

*TSS*

*The evaluator shall ensure that the TSS describes the client's method of establishing all reference identifiers from the user/administrator/application-configured reference identifier, including which types of reference identifiers are supported (e.g Common Name, DNS Name, URI Name, Service Name, or other application-specific Subject Alternative Names) and whether IP addresses and wildcards are supported. The evaluator shall ensure that this description identifies whether and the manner in which certificate pinning is supported or used by the TOE.*

*GUIDANCE*

*The evaluator shall verify that the AGD guidance includes instructions for setting the reference identifier to be used for the purposes of certificate validation in TLS.*

*TEST*

*The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection:*

*Test 1: The evaluator shall present a server certificate that does not contain an identifier in either the Subject Alternative Name (SAN) or Common Name (CN) that matches the reference identifier. The evaluator shall verify that the connection fails.*

*Test 2: The evaluator shall present a server certificate that contains a CN that matches the reference identifier, contains the SAN extension, but does not contain an identifier in the SAN that matches the reference identifier. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each supported SAN type.*

*Test 3: The evaluator shall present a server certificate that contains a CN that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds.*

*Test 4: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier but does contain an identifier in the SAN that matches. The evaluator shall verify that the connection succeeds.*

*Test 5: The evaluator shall perform the following wildcard tests with each supported type of reference identifier:*

- *The evaluator shall present a server certificate containing a wildcard that is not in the left-most label of the presented identifier (e.g. foo.*.example.com) and verify that the connection fails.*
- *The evaluator shall present a server certificate containing a wildcard in the left-most label but not preceding the public suffix (e.g. *.example.com). The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.example.com) and verify that the connection succeeds. The evaluator shall configure the reference identifier without a left-most label as in the certificate (e.g. example.com) and verify that the connection fails. The evaluator shall configure*

*the reference identifier with two left-most labels (e.g. bar.foo.example.com) and verify that the connection fails.*

- *The evaluator shall present a server certificate containing a wildcard in the left-most label immediately preceding the public suffix (e.g. *.com). The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.com) and verify that the connection fails. The evaluator shall configure the reference identifier with two left-most labels (e.g. bar.foo.com) and verify that the connection fails.*

*Test 6: [conditional] If URI or Service name reference identifiers are supported, the evaluator shall configure the DNS name and the service identifier. The evaluator shall present a server certificate containing the correct DNS name and service identifier in the URIName or SRVName fields of the SAN and verify that the connection succeeds. The evaluator shall repeat this test with the wrong service identifier (but correct DNS name) and verify that the connection fails.*

*Test 7: [conditional] If pinned certificates are supported the evaluator shall present a certificate that does not match the pinned certificate and verify that the connection fails.*

**FCS_TLSC_EXT.1.3** The [selection: <u>TSF, TOE platform</u>] shall only establish a trusted channel if the peer certificate is valid.

*Application Note:*

*Validity is determined by the identifier verification, certificate path, the expiration date, and the revocation status in accordance with RFC 5280. Certificate validity shall be tested in accordance with testing performed for FIA_X509_EXT.1.*

*For TLS connections, this channel shall not be established if the peer certificate is invalid.*

***Assurance Activity:***

*TEST*

*The evaluator shall use TLS as a function to verify that the validation rules in FIA_X509_EXT.1.1 are adhered to and shall perform the following additional test:*

*Test 1: The evaluator shall demonstrate that a peer using a certificate without a valid certification path results in an authenticate failure. Using the administrative guidance, the evaluator shall then load the trusted CA certificate(s) needed to validate the peer's certificate, and demonstrate that the connection succeeds. The evaluator then shall delete one of the CA certificates, and show that the connection fails.*

**FCS_TLSC_EXT.1.4** The [selection: <u>TSF, TOE platform</u>] shall support mutual authentication using X.509v3 certificates.

*Application Note:*

*The use of X.509v3 certificates for TLS is addressed in FIA_X509_EXT.2.1. This requirement adds that this use must include the client must be capable of presenting a certificate to a TLS server for TLS mutual authentication.*

***Assurance Activity:***

*TSS*

*The evaluator shall ensure that the TSS description required per FIA_X509_EXT.2.1 includes the use of client-side certificates for TLS mutual authentication.*

*GUIDANCE*

*The evaluator shall verify that the AGD guidance required per FIA_X509_EXT.2.1 includes instructions for configuring the client-side certificates for TLS mutual authentication.*

*TEST*

*The evaluator shall perform the following modification to the traffic:*

- *Configure the server to require mutual authentication and then modify a byte in a CA field in the Server's Certificate Request handshake message. The modified CA field must not be the CA used to sign the client's certificate. The evaluator shall verify the connection is unsuccessful.*

## B.3. Optional Requirements to Support MAS Server

### B.3.1. Security Audit (FAU)

#### B.3.1.1 Audit Generation (MAS Server)

**FAU_GEN.1.1(2) Refinement**: The **MAS Server** shall be able to generate an audit record of the following auditable events:

a. Failure to push a new application on a managed mobile device;
b. Failure to update an existing application on a managed mobile device.

*Application Note:*

*The MDM Agent is required to report to the MAS Server on successful receipt of an application or update on a managed mobile device, and failures can be inferred from the absence of such alerts.*

***Assurance Activity:***

*TSS*

*The evaluator shall check the TSS and ensure that it provides a format for audit records.*

*GUIDANCE*

*The evaluator shall check the administrative guide and ensure that it provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field.*

*TEST*

*The evaluator shall verify that when an application or update push fails, that the audit records generated match the format specified in the guidance and that the fields in each audit record have the proper entries.*

**FAU_GEN.1.2(2) Refinement:** The [selection: *MAS Server, MAS Server platform*] shall record within each TSF audit record at least the following information:

- date and time of the event,
- type of event,
- mobile device identity**,**
- [assignment: *other audit relevant information*].

*Application Note:*

*All audits must contain at least the information mentioned in FAU_GEN.1.2, but may contain more information which can be assigned. The ST author shall identify in the TSS which information of the audit record that is performed by the TSF and that which is performed by the TOE platform.*

***Assurance Activity:***

*TSS*

*The evaluator shall check the TSS and ensure that it provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field.*

*GUIDANCE*

*The evaluator shall check the administrative guide and ensure that it provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field. The evaluator shall check to make sure that the description of the fields contains the information required in FAU_GEN.1.2.*

*TEST*

*When verifying the test results from FAU_GEN.1.1, the evaluator shall ensure the audit records generated during testing match the format specified in the administrative guide, and that the fields in each audit record have the proper entries.*

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

### B.3.1.2 External Audit Trail Storage (MAS Server)

**FAU_STG_EXT.1.1(2)** The [selection: MAS Server, MAS Server platform] shall be able to transmit audit data to an external IT entity using a trusted channel implementing the [selection: IPsec, SSH, TLS, TLS/HTTPS] protocol.

*Application Note:*

*The TOE may rely on a non-TOE audit server for storage and review of audit records. Although the TOE generates audit records and receives audit records from managed mobile devices, the storage of these audit records and the ability to allow the administrator to review these audit records is provided by the operational environment. The TSF may rely on the underlying operating system for this functionality, and the first selection should be made appropriately.*

*In the second selection, the ST author chooses the means by which this connection is protected. The ST author also ensures that the supporting protocol requirement matching the selection is included in the ST.*

***Assurance Activity:***

*TSS*

*The evaluator shall examine the TSS to ensure it describes the means by which the audit data are transferred to the external audit server, and how the trusted channel is provided.*

*GUIDANCE*

*The evaluator shall also examine the operational guidance to determine that it describes the relationship between the local audit data and the audit data that are sent to the audit log server. For example, when an audit event is generated, is it simultaneously sent to the external server and the local store, or is the local store used as a buffer and "cleared" periodically by sending the data to the audit server.*

*The evaluator shall also examine the operational guidance to ensure it describes how to establish the trusted channel to the audit server, as well as describe any requirements on the audit server (particular audit server protocol, version of the protocol required, etc.), as well as configuration of the TOE needed to communicate with the audit server.*

*TEST*

*Testing of the trusted channel mechanism will be performed as specified in the associated assurance activities for the particular trusted channel mechanism.*

*The evaluator shall perform the following test for this requirement:*

*Test: The evaluator shall establish a session between the TOE and the audit server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the audit server and the TOE during several activities of the evaluator's choice designed to generate audit data to be transferred to the audit server. The evaluator shall observe that these data are not able to be viewed in the clear during this transfer, and that they are successfully received by the audit server. The evaluator shall record the particular software (name, version) used on the audit server during testing.*

## B.3.2. Security Management (FMT)

### B.3.2.1 Management of Functions in MAS Server

**FMT_MOF.1.1(3) Refinement:** The **MAS Server** shall restrict the ability to **configure user groups for user-access to specific applications** to **the administrator**.

*Application Note*

*This requirement is to ensure that the MAS Server can create groups to configure which applications a user can access based on which group they are in. If the MAS Server uses the groups defined by the MDM, then it must communicate with the MDM Server (if separate server) to determine which applications the user can access.*

***Assurance Activity***

*TSS*

*The evaluator shall examine the TSS to determine if the MAS Server creates its own groups or relies on the groups specified by the MDM Server.*

*GUIDANCE*

*The evaluator shall confirm that the operational guidance contains how to create and define user groups and how to specify which applications are accessible by which group.*

*TEST*

*The evaluator shall ensure that the MAS client can only access the applications specified for the group they are enrolled in. The evaluator shall create a user group and do not define the user to be part of the group. Verify that an application accessible to that group cannot be accessed. The evaluator shall include the user in the group and assure that the application can be accessed.*

#### B.3.2.2 Management of Download Function in MAS Server

**FMT_MOF.1.1(4) Refinement:** The **MAS Server** shall restrict the ability to **download applications** to **enrolled mobile devices that are compliant with MDM policies and assigned to a user in the application access group**.

*Assurance Activity:*

*TSS*

*The evaluator shall examine the TSS to determine that all methods of initiating an application download or update push are specified.*

*GUIDANCE*

*The evaluator shall confirm that the operational guidance contains how to initiate an application download or update push.*

*TEST*

*The evaluator shall ensure that the MAS Server verifies that the mobile device is enrolled in the MDM Server and is in a compliant state. The evaluator shall verify that an application cannot be downloaded from the MAS Server prior to enrolling the device with the MDM. The evaluator shall partially enroll the mobile device, so the device is connected to the MDM Server, but is not complaint and verify that applications cannot be downloaded.*

#### B.3.2.3 Specification of Management Functions (MAS Server)

**FMT_SMF.1.1(3) Refinement:** The **MAS Server** shall be capable of **performing** the following management functions:

a. Configure application access groups,
b. Download applications,
c. [selection: [assignment: other MAS management functions], no other functions].

*Application Note:*

*This requirement captures all the configuration functionality in the MAS Server to configure the underlying MAS Server. The ST author can add more commands and configuration policies by completing the assignment statement.*

***Assurance Activity:***

*TSS*

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

*GUIDANCE*

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

*TEST*

The test of the functions are performed in conjunction with the restriction of the function in FMT_MOF.1.

#### B.3.2.4 Security Management Roles

**FMT_SMR.1.1(2) Refinement:** The **MAS Server** shall maintain the roles *administrator, MD user, enrolled mobile devices, application access groups, and [assignment: additional authorized identified roles]*.

**FMT_SMR.1.2(2) Refinement:** The **MAS Server** shall be able to associate users with roles.

*Application Note:*

*It is envisioned that the MAS Server will be configured and maintained by different user roles. The assignment is used by the ST author to list the roles that are supported. At a minimum, one administrative role shall be supported. If no additional roles are supported, then "no additional roles" is stated. The MD user role is used for enrollment of mobile devices to the MAS according to FIA_ENR_EXT.1.*

***Assurance Activity:***

*TSS*

*The evaluator shall examine the TSS to verify that it describes the administrator role and the powers granted to and limitations of the role.*

*GUIDANCE*

*The evaluator shall review the operational guidance to ensure that it contains instructions for administering the TOE and which interfaces are supported.*

*TEST*

*In the course of performing the testing activities for the evaluation, the evaluator shall use all supported interfaces, although it is not necessary to repeat each test involving an administrative action with each interface. The evaluator shall ensure, however, that each supported method of administering the TOE that conforms to the requirements of this PP be tested; for instance, if the TOE can be administered through a local hardware interface or TLS/HTTPS then both methods of administration must be exercised during the evaluation team's test activities.*

### B.3.3. Protection of the TSF (FPT)

#### B.3.3.1 Basic Internal TSF Data Transfer Protection (Distributed TOE)

**FPT_ITT.1.1(2) Refinement:** The TSF shall protect **all data** from <u>disclosure and modification</u> **through use of [selection: <u>IPsec, TLS, HTTPS, DTLS</u>]** when it is transferred between separate parts of the TOE.

*Application Note:*

*This requirement ensures all communications between components of a distributed TOE (such as between the MDM Server and the MAS Server) is protected through the use of an encrypted communications channel. The data passed in this trusted communication channel are encrypted as defined the protocol chosen in the first selection.*

*The trusted channel uses secure protocols that preserve the confidentiality and integrity of MDM communications. The ST author chooses the mechanism or mechanisms supported by the TOE, and then ensures the detailed requirements in Appendix C: Selection-Based Requirements corresponding to their selection are copied to the ST if not already present.*

*Protocol, RBG, Certificate validation, algorithm, and similar services may be met with platform provided services.*

***Assurance Activity:***

*TSS*

*The evaluator shall examine the TSS to determine that the methods and protocols used to protect distributed TOE components are described. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST.*

*GUIDANCE*

*The evaluator shall confirm that the operational guidance contains instructions for establishing the communication paths for each supported method.*

*TEST*

*Test 3: The evaluators shall ensure that communications using each specified (in the operational guidance) communication method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.*

*Test 4: The evaluator shall ensure, for each method of communication, the channel data is not sent in plaintext.*

*Further assurance activities are associated with the specific protocols.*

#### B.3.3.2 Basic Internal TSF Data Transfer Protection (MAS Server)

**FPT_ITT.1.1(3) Refinement:** The TSF shall protect **all data** from disclosure and modification **through use of [selection: TLS, HTTPS, DTLS]** when it is transferred between **the MDM Agent and MAS Server.**

*Application Note:*

*This requirement is to ensure that the transmission of any audit logs, applications, and configuration data collected by the MDM Agent and sent from the MDM Agent to the MAS Server, when commanded, or at configurable intervals, is properly protected. This trusted channel also protects any commands and policies sent by the MAS Server to the MDM Agent. Either the MDM Agent or the MAS Server is able to initiate the connection.*

*The trusted channel uses secure protocols that preserve the confidentiality and integrity of MAS communications. The ST author chooses the mechanism or mechanisms supported by the TOE, and then ensures the detailed requirements in Appendix C: Selection-Based Requirements corresponding to their selection are copied to the ST if not already present.*

*Protocol, RBG, Certificate validation, algorithm, and similar services may be met with platform provided services.*

***Assurance Activity:***

*TSS*

*The evaluator shall examine the TSS to determine that for each supported Agent listed in the ST, the methods of Agent-Server communication are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of remote TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST.*

*GUIDANCE*

*The evaluator shall confirm that the operational guidance contains instructions for configuring the communication channel between each supported MDM Agent and the MAS Server for each supported method.*

*TEST*

*For each supported MAS Agent/platform listed in the ST:*

*Test 1: The evaluators shall ensure that communications using each specified (in the operational guidance) Agent-Server communication method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.*

*Test 2: The evaluator shall ensure, for each method of Agent-Server communication, the channel data is not sent in plaintext.*

*Further assurance activities are associated with the specific protocols.*

### B.3.4. Trusted Path/Channels (FTP)

#### B.3.4.1 Inter-TSF Trusted Channel (Authorized IT Entities)

**FTP_ITC.1.1(3) Refinement:** The [selection: **MAS Server, MAS Server platform**] shall **use [selection: IPsec, SSH, TLS, TLS/HTTPS]** to provide a **trusted** communication channel between itself and **authorized IT entities supporting the following capabilities: audit server, [selection: authentication server, [assignment: *other capabilities*]]** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data **from disclosure and detection of modification of the channel data**.

*Application Note:*

*The intent of the mandatory portion of the above requirement is to use the cryptographic protocols identified in the requirement to establish and maintain a trusted channel with authorized IT entities that the MAS Server interacts with to perform its functions.*

*Protection (by one of the listed protocols) is required at least for communications with the server that collects the audit information. If it communicates with an authentication server (e.g., RADIUS), then the ST author chooses "authentication server" in FTP_ITC.1.1(1) and this connection must be protected by one of the listed protocols. If other authorized IT entities (e.g., NTP server) are protected, the ST author makes the appropriate assignments (for those entities) and selections (for the protocols that are used to protect those connections). After the ST author has made the selections, they are to select the detailed requirements in Appendix C corresponding to their protocol selection to put in the ST. To summarize, the connection to an external audit collection server is required to be protected by one of the listed protocols. If an external authentication server is supported, then it is required to protect that connection with one of the listed protocols. For any other external server, external communications are not required to be protected, but if protection is claimed, then it must be protected with one of the identified protocols.*

*The trusted channel uses IPsec, TLS, DTLS, or HTTPS as the protocol that preserves the confidentiality and integrity of MAS communications. The ST author chooses the mechanism or mechanisms supported by the TOE, and then ensures the detailed requirements in Appendix C corresponding to their selection are copied to the ST if not already present.*

*Protocol, RBG, Certificate validation, algorithm, and similar services may be met with platform provided services.*

*The requirement implies that not only are communications protected when they are initially established, but also on resumption after an outage. It may be the case that some part of the TOE setup involves manually setting up tunnels to protect other communication, and if after an outage the TOE attempts to re-establish the communication automatically with (the necessary) manual intervention, there may be a window created where an attacker might be able to gain critical information or compromise a connection.*

**FTP_ITC.1.2(3)** The TSF shall permit the **MAS Server or other authorized IT entities** to initiate communication via the trusted channel.

**FTP_ITC.1.3(3)** The TSF shall initiate communication via the trusted channel for **[assignment: list of services for which the TSF is able to initiate communications].**

*Application Note:*

*While there are no requirements on the party initiating the communication, the ST author lists in the assignment for FTP_ITC.1.3 the services for which the TOE can initiate the communication with the authorized IT entity.*

***Assurance Activity:***

*TSS*

*The evaluator shall examine the TSS to determine that the methods of communication with authorized IT entities are indicated, along with how those communications are protected.*

*GUIDANCE*

*The evaluator shall confirm that the operational guidance contains instructions for configuring the communication channel between the MAS Server and authorized IT entities for each supported method.*

*TEST*

*Test 7: The evaluators shall ensure that communications using each specified (in the operational guidance) communication method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.*

*Test 8: The evaluator shall ensure, for each method of communication, the channel data is not sent in plaintext.*

*Test 9: The evaluator shall ensure, for each communication channel with the MAS Server, that a protocol analyzer identifies the traffic as the protocol under testing.*

*Further assurance activities are associated with the specific protocols.*

# C. SELECTION-BASED REQUIREMENTS

As indicated in the introduction to this PP, the baseline requirements (those that must be performed by the TOE or its underlying platform) are contained in the body of this PP. There are additional requirements based on selections in the body of the PP: if certain selections are made, then additional requirements below will need to be included.

This Appendix is divided into two subsections: selection-based requirements that may be performed by the TSF and selection-based requirements that may be performed by the TSF or its underlying platform.

## C.1. Selection-Based TSF Requirements

### C.1.1. Cryptographic Support (FCS)

#### C.1.1.1 Initialization Vector Generation

**FCS_IV_EXT.1.1** The MDM Servershall generate IVs in accordance with Table 9.

*Application Note:*

*Table 9 lists the requirements for composition of IVs according to the corresponding NIST Special Publications for each cipher mode. The composition of IVs generated for encryption according to a cryptographic protocol is addressed by the protocol. Thus, this requirement addresses only IVs generated for key storage encryption.*

***Assurance Activity:***

*TSS*

*The evaluator shall examine the TSS to ensure that it details the encryption of user credentials, persistent secrets, and private keys and the generation of the IVs used for that encryption.*

*TEST*

*The evaluator shall ensure that the generation of IVs for each key encrypted by the same KEK meets Table 9.*

#### C.1.1.2 Encrypted Cryptographic Key Storage

**FCS_STG_EXT.2.1** The MDM Server shall encrypt all keys using AES in the [selection: *Key Wrap (KW) mode, Key Wrap with Padding (KWP) mode, GCM, CCM, CBC mode*].

*Application Note:*

*This requirement states that keys used by the TSF shall not be kept in plaintext. The intent of this requirement is to ensure that the private keys, credentials, and persistent secrets cannot be accessed in the TOE in an unencrypted state, allowing an attacker to access keys without having to exhaust the AES key space. This requirement must be including in the ST if the selection in FCS_STG_EXT.1 indicates that the TSF is protecting private keys and persistent secrets with encryption rather than the platform-provided key storage.*

*If this requirement is included in the ST, FCS_IV_EXT.1 must also be included.*

***Assurance Activity:***

*TSS*

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

## C.2. Selection-Based TOE or Platform Requirements

### C.2.1. Cryptographic Support (FCS)

#### C.2.1.1 DTLS Protocol

**FCS_DTLS_EXT.1.1** The [selection: MDM Server, MDM Server platform] shall implement the DTLS protocol in accordance with DTLS 1.2 (RFC 6347).

**FCS_DTLS_EXT.1.2** The [selection: MDM Server, MDM Server platform] shall implement the requirements in TLS (FCS_TLSS_EXT.2) for the DTLS implementation, except where variations are allowed according to DTLS 1.2 (RFC 6347).

*Application Note:*

*Differences between DTLS 1.2 and TLS 1.2 are outlined in RFC 6347; otherwise the protocols are the same. In particular, for the applicable security characteristics defined for the TSF, the two protocols do not differ. Therefore, all application notes and assurance activities that are listed for TLS apply to the DTLS implementation.*

**FCS_DTLS_EXT.1.3** The [selection: MDM Server, MDM Server platform] shall not establish a trusted communication channel if the peer certificate is deemed invalid.

*Application Note:*

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

***Assurance Activity:***

*TEST*

*Test 1: The evaluator shall attempt to establish a connection with a DTLS server, observe the traffic with a packet analyzer, and verify that the connection succeeds and that the traffic is identified as DTLS.*

*Other tests are performed in conjunction with the Assurance Activity listed for FCS_TLSS_EXT.1.*

*Certificate validity shall be tested in accordance with testing performed for FIA_X509_EXT.1, and the evaluator shall perform the following test.*

*Test 2: The evaluator shall demonstrate that using a certificate without a valid certification path results in the function failing. Using the administrative guidance, the evaluator shall then load a certificate or certificates to the Trust Anchor Database needed to validate the certificate to be used in the function, and demonstrate that the function succeeds. The evaluator then shall delete one of the certificates, and show that the function fails.*

#### C.2.1.2 HTTPS Protocol

**FCS_HTTPS_EXT.1.1** The [selection: MDM Server, MDM Server platform] shall implement the HTTPS protocol that complies with RFC 2818.

**FCS_HTTPS_EXT.1.2** The [selection: MDM Server, MDM Server platform] shall implement HTTPS using TLS as specified in FCS_TLSS_EXT.1.

**FCS_HTTPS_EXT.1.3** The [selection: MDM Server, MDM Server platform] shall [selection: not establish the connection, request authorization to establish the connection, no other action] if the peer certificate is deemed invalid.

*Application Note:*

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

***Assurance Activity***

*TEST*

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

*Other tests are performed in conjunction with the TLS evaluation activities.*

*Certificate validity shall be tested in accordance with testing performed for FIA_X509_EXT.1, and the evaluator shall perform the following test:*

*Test 2: The evaluator shall demonstrate that using a certificate without a valid certification path results in an application notification. Using the administrative guidance, the evaluator shall then load a valid certificate and certification path, and demonstrate that the function succeeds. The evaluator then shall delete one of the certificates, and show that the selection listed in the ST occurs.*

### C.2.1.3 IPsec Protocol

**FCS_IPSEC_EXT.1.1** The [selection: MDM Server, MDM Server platform] shall implement the IPsec architecture as specified in RFC 4301.

*Application Note:*

*RFC 4301 calls for an IPsec implementation to protect IP traffic through the use of a Security Policy Database (SPD). The SPD is used to define how IP packets are to be handled: PROTECT the packet (e.g., encrypt the packet), BYPASS the IPsec services (e.g., no encryption), or DISCARD the packet (e.g., drop the packet). The SPD can be implemented in various ways, including router access control lists, firewall rulesets, a "traditional" SPD, etc. Regardless of the implementation details, there is a notion of a "rule" that a packet is "matched" against and a resulting action that takes place.*

*While there must be a means to order the rules, a general approach to ordering is not mandated, as long as the SPD can distinguish the IP packets and apply the rules accordingly. There may be multiple SPDs (one for each network interface), but this is not required.*

***Assurance Activity:***

*TSS*

*The evaluator shall examine the TSS and determine that it describes what takes place when a packet is processed by the TOE, e.g., the algorithm used to process the packet. The TSS describes how the SPD is implemented and the rules for processing both inbound and outbound packets in terms of the IPsec policy. The TSS describes the rules that are available and the resulting actions available after matching a rule. The TSS describes how those rules and actions form the SPD in terms of the BYPASS (e.g., no* *encryption), DISCARD (e.g., drop the packet), and PROTECT (e.g., encrypt the packet) actions defined in RFC 4301.*

*As noted in section 4.4.1 of RFC 4301, the processing of entries in the SPD is non-trivial and the evaluator shall determine that the description in the TSS is sufficient to determine which rules will be applied given the rule structure implemented by the TOE. For example, if the TOE allows specification of ranges, conditional rules, etc., the evaluator shall determine that the description of rule processing (for both inbound and outbound packets) is sufficient to determine the action that will be applied, especially in the case where two different rules may apply. This description shall cover both the initial packets (that is, no SA is established on the interface or for that particular packet) as well as packets that are part of an established SA.*

*GUIDANCE*

*The evaluator shall examine the operational guidance to verify it instructs the Administrator how to construct entries into the SPD that specify a rule for processing a packet. The description includes all three cases – a rule that ensures packets are encrypted/decrypted, dropped, and flow through the TOE without being encrypted. The evaluator shall determine that the description in the operational guidance is consistent with the description in the TSS, and that the level of detail in the operational guidance is sufficient to allow the administrator to set up the SPD in an unambiguous fashion. This includes a discussion of how ordering of rules impacts the processing of an IP packet.*

*TEST*

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

*Test 1: The evaluator shall configure the SPD such that there is a rule for dropping a packet, encrypting a packet, and allowing a packet to flow in plaintext. The selectors used in the construction of the rule shall be different such that the evaluator can generate a packet and send packets to the gateway with the appropriate fields (fields that are used by the rule - e.g., the IP addresses, TCP/UDP ports) in the packet header. The evaluator performs both positive and negative test cases for each type of rule (e.g. a packet that matches the rule and another that does not match the rule). The evaluator observes via the audit trail, and packet captures that the TOE exhibited the expected behavior: appropriate packets were dropped, allowed to flow without modification, encrypted by the IPsec implementation.*

*Test 2: The evaluator shall devise several tests that cover a variety of scenarios for packet processing. As with Test 1, the evaluator ensures both positive and negative test cases are constructed. These scenarios must exercise the range of possibilities for SPD entries and processing modes as outlined in the TSS and operational guidance. Potential areas to cover include rules with overlapping ranges and conflicting entries, inbound and outbound packets, and packets that establish SAs as well as packets that belong to established SAs. The evaluator shall verify, via the audit trail and packet captures, for each scenario that the expected behavior is exhibited, and is consistent with both the TSS and the operational guidance.*

**FCS_IPSEC_EXT.1.2** The [selection: MDM Server, MDM Server platform] shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched, and discards it.

***Assurance Activity:***

*TEST*

*The assurance activity for this element is performed in conjunction with the activities for FCS_IPSEC_EXT.1.1.*

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

*Test: The evaluator shall configure the SPD such that there is a rule for dropping a packet, encrypting a packet, and allowing a packet to flow in plaintext. 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 the rule to allow the packet to flow in plaintext and send that packet. The evaluator should observe that the network packet is passed to the proper destination 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/platform created" final entry that discards packets that do not match any previous entries). The evaluator sends the packet, and observes that the packet was dropped.*

**FCS_IPSEC_EXT.1.3** The [selection: MDM Server, MDM Server platform] shall implement transport mode and [selection: tunnel mode, no other mode].

***Assurance Activity:***

*TSS*

*The evaluator checks the TSS to ensure it states that the VPN can be established to operate in tunnel mode and/or transport mode (as identified in FCS_IPSEC_EXT.1.3).*

*GUIDANCE*

*The evaluator shall confirm that the operational guidance contains instructions on how to configure the connection in each mode selected.*

*TEST*

*The evaluator shall perform the following test(s) based on the selections chosen:*

*Test 1: The evaluator uses the operational guidance to configure the TOE/platform to operate in transport mode and also configures a VPN peer to operate in transport mode. The evaluator configures the TOE/platform and the VPN peer 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/platform to connect to the VPN peer. The evaluator observes (for example, in the audit trail and the captured packets) that a successful connection was established using the transport mode.*

*Test 2: (conditional) If tunnel mode is selected, the evaluator uses the operational guidance to configure the TOE/platform to operate in tunnel mode and also configures a VPN peer to operate in tunnel mode. The evaluator configures the TOE/platform and the VPN peer 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 peer. The evaluator observes (for example, in the audit trail and the captured packets) that a successful connection was established using the tunnel mode.*

**FCS_IPSEC_EXT.1.4** The [selection: MDM Server, MDM Server platform] 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/platform 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*

*The evaluator shall configure the TOE/platform as indicated in the operational guidance configuring the TOE/platform to use each of the supported algorithms, attempt to establish a connection using ESP, and verify that the attempt succeeds.*

**FCS_IPSEC_EXT.1.5** The [selection: MDM Server, MDM Server platform] shall implement the protocol: [selection:

- 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 and [selection: with support for NAT traversal (NAT-T) as specified in RFC 5996 Section 2.23, RFC 4868 for hash functions, no other RFCs]

].

*Application Note:*

*If the TOE implements SHA-2 hash algorithms for IKEv1 or IKEv2, the ST author shall select RFC 4868. If the ST author selects IKEv1, FCS_IPSEC_EXT.1.16 must also be included in the ST. IKEv2 will be required for those TOEs entering evaluation after Quarter 3, 2016.*

***Assurance Activity:***

*TSS*

*The evaluator shall examine the TSS to verify that IKEv1 and/or IKEv2 are implemented.*

*GUIDANCE*

*The evaluator shall check the operational guidance to ensure it instructs the administrator how to configure the TOE/platform to use IKEv1 and/or IKEv2 (as selected), and uses the guidance to configure the TOE/platform to perform NAT traversal for the following test (if selected).*

*TEST*

*Tests are performed in conjunction with the other IPsec evaluation activities.*

*Test (conditional): The evaluator shall configure the TOE/platform 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 [selection: MDM Server, MDM Server platform] shall ensure the encrypted payload in the [selection: IKEv1, IKEv2] protocol uses the cryptographic algorithms AES-CBC-128, AES-CBC-256 as specified in RFC 3602 and [selection: AES-GCM-128, AES-GCM-256 as specified in RFC 5282, no other algorithm].

*Application Note:*

*AES-GCM-128 and AES-GCM-256 may only be selected if IKEv2 is also selected, as there is no RFC defining AES-GCM for IKEv1. Note that FCS_IPSEC_EXT.1.4 for IPsec and FCS_IPSEC_EXT.1.6 for IKE mandate different AES modes.*

***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 the configuration of the mandated algorithms, as well as any additional algorithms selected in the requirement. The guidance is then used to configure the TOE/platform to perform the following test for each ciphersuite selected.*

*TEST*

*The evaluator shall configure the TOE/platform to use the ciphersuite under test 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 the indicated ciphersuite. The evaluator will confirm the algorithm was that used in the negotiation.*

**FCS_IPSEC_EXT.1.7** The [selection: MDM Server, MDM Server platform] shall ensure that [selection:

- IKEv1 Phase 1 SA lifetimes can be configured by an Authorized Administrator based on [selection:
  - number of packets/number of bytes;
  - length of time, where the time values can configured within [assignment: *integer range including 24*] hours

  ];
- IKEv2 SA lifetimes can be configured by an Authorized Administrator based on [selection:
  - number of packets/number of bytes;
  - length of time, where the time values can configured within [assignment: *integer range including 24*] hours

  ]

].

*Application Note:*

*The ST author chooses either the IKEv1 requirements or IKEv2 requirements (or both, depending on the selection in FCS_IPSEC_EXT.1.5. The ST author chooses either packet/volume-based lifetimes or time-based lifetimes. This requirement must be accomplished by providing Authorized Administrator-configurable lifetimes (with appropriate instructions in documents mandated by AGD_OPE). Hardcoded limits are not acceptable. In general, instructions for setting the parameters of the implementation, including lifetime of the SAs, should be included in the operational guidance generated for AGD_OPE.*

***Assurance Activity:***

*GUIDANCE*

*The evaluator shall verify that the values for SA lifetimes can be configured and that the instructions for doing so are located in the operational guidance. If time-based limits are supported, the evaluator ensures that the Administrator is able to configure SA values for 24 hours. 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 functionality, 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: (Conditional) The evaluator shall configure a maximum lifetime in terms of the number of packets (or bytes) allowed following the operational guidance. The evaluator shall configure a test peer with a packet/byte lifetime that exceeds the configured lifetime of the TOE. The evaluator shall establish an SA between the TOE and the test peer, and determine that once the allowed number of packets (or bytes) through this SA is exceeded, a new SA is negotiated. The evaluator shall verify that the TOE initiates a Phase 1 negotiation.*

*Test 2: (Conditional) The evaluator shall configure a maximum lifetime of 24 hours for the Phase 1 SA following the operational guidance. The evaluator shall configure a test peer with a lifetime that exceeds the configured lifetime of the TOE. The evaluator shall establish an SA between the TOE and the test peer, maintain the Phase 1 SA for 24 hours, and determine that once 24 hours has elapsed, a new Phase 1 SA is negotiated. The evaluator shall verify that the TOE initiates a Phase 1 negotiation.*

**FCS_IPSEC_EXT.1.8** The [selection: MDM Server, MDM Server platform] shall ensure that [selection:

- IKEv1 Phase 2 SA lifetimes can be configured by an Authorized Administrator based on [selection:
  - number of packets/number of bytes;
  - length of time, where the time values can be configured within [assignment: integer range including 8] hours;

  ];

- IKEv2 Child SA lifetimes can be configured by an Authorized Administrator based on [selection:
  - number of packets/number of bytes;
  - length of time, where the time values can be configured within [assignment: integer range including 8] hours;

  ]

].

*Application Note:*

*The ST author chooses either the IKEv1 requirements or IKEv2 requirements (or both, depending on the selection in FCS_IPSEC_EXT.1.5). The ST author chooses either packet/volume-based lifetimes or time-based lifetimes. This requirement must be accomplished by providing Authorized Administrator-configurable lifetimes (with appropriate instructions in documents mandated by AGD_OPE). Hardcoded limits are not acceptable. In general, instructions for setting the parameters of the implementation, including lifetime of the SAs, should be included in the operational guidance generated for AGD_OPE.*

***Assurance Activity:***

*GUIDANCE*

*The evaluator shall verify that the values for SA lifetimes can be configured and that the instructions for doing so are located in the operational guidance. If time-based limits are supported, the evaluator ensures that the Administrator is able to configure Phase 2/Child SA values for 8 hours. 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 functionality, 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: (Conditional) The evaluator shall configure a maximum lifetime in terms of the number of packets (or bytes) allowed following the operational guidance. The evaluator shall configure a test peer with a packet/byte lifetime that exceeds the lifetime of the TOE. The evaluator shall establish an SA between the TOE and the test peer, and determine that once the allowed number of packets (or bytes) through this SA is exceeded, a new SA is negotiated. The evaluator shall verify that the TOE initiates a Phase 2 negotiation.*

*Test 2: (Conditional) The evaluator shall configure a maximum lifetime of 8 hours for the Phase 2 SA following the operational guidance. The evaluator shall configure a test peer with a lifetime that exceeds the lifetime of the TOE. The evaluator shall establish an SA between the TOE and the test peer, maintain the Phase 2 SA for 8 hours, and determine that once 8 hours has elapsed, a new Phase 2 SA is negotiated. The evaluator shall verify that the TOE initiates a Phase 2 negotiation.*

**FCS_IPSEC_EXT.1.9** The [selection: MDM Server, MDM Server platform] 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 security strength of the negotiated Diffie-Hellman group*] bits.

*Application Note:*

*For DH groups 19 and 20, the "x" value is the point multiplier for the generator point G.*

*Since the implementation may allow different Diffie-Hellman groups to be negotiated for use in forming the SAs, the assignment in FCS_IPSEC_EXT.1.9 may contain multiple values. For each DH group supported, the ST author consults Table 2 in NIST SP 800-57 "Recommendation for Key Management – Part 1: General" to determine the security strength ("bits of security") associated with the DH group. Each unique value is then used to fill in 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.*

***Assurance Activity:***

*TSS*

*The evaluator shall check to ensure that, for each DH group supported, the TSS describes the process for generating "x" (as defined in FCS_IPSEC_EXT.1.8). 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" meets the stipulations in the requirement.*

**FCS_IPSEC_EXT.1.10** The [selection: MDM Server, MDM Server platform] shall generate nonces used in [selection: *IKEv1, IKEv2*] exchanges of length [selection:

- [assignment: security strength associated with the negotiated Diffie-Hellman group];
- at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash

] .

*Application Note:*

*The ST author must select the second option for nonce lengths if IKEv2 is also selected (as this is mandated in RFC 5996). The ST author may select either option for IKEv1.*

*For the first option for nonce lengths, since the implementation may allow different Diffie-Hellman groups to be negotiated for use in forming the SAs, the assignment in FCS_IPSEC_EXT.1.9 may contain multiple values. For each DH group supported, the ST author consults Table 2 in NIST SP 800-57 "Recommendation for Key Management –Part 1: General" to determine the security strength ("bits of security") associated with the DH group. Each unique value is then used to fill in 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 the assignment would read "[112,192]".*

*Because nonces may be exchanged before the DH group is negotiated, the nonce used should be large enough to support all TOE-chosen proposals in the exchange.*

***Assurance Activity:***

*TSS*

*(conditional) If the first selection is chosen, the evaluator shall check to ensure that, for each DH group supported, the TSS describes the process for generating 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 the nonces meet the stipulations in the requirement.*

*(conditional) If the second selection is chosen, the evaluator shall check to ensure that, for each PRF hash supported, the TSS describes the process for generating 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 the nonces meet the stipulations in the requirement.*

**FCS_IPSEC_EXT.1.11** The [selection: MDM Server, MDM Server platform] shall ensure that all IKE protocols implement DH Groups 14 (2048-bit MODP), and [selection: 19 (256-bit Random ECP), 5 (1536-bit MODP), 24 (2048-bit MODP with 256-bit POS), 20 (384-bit Random ECP), no other DH groups].

*Application Note:*

*The selection is used to specify additional DH groups supported. This applies to IKEv1 and IKEv2 exchanges. For products entering into evaluation after Quarter 3, 2015, DH Group 19 (256-bit Random ECP) and DH Group 20 (384-bit Random ECP) 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.*

***Assurance Activity:***

*TSS*

*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.*

*TEST*

*For each supported DH group:*

*Test: The evaluator shall test to ensure that all supported IKE protocols can be successfully completed using that particular DH group.*

**FCS_IPSEC_EXT.1.12** The [selection: MDM Server, MDM Server platform] 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.

*Application Note:*

*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 this capability to be configurable, the default configuration in the evaluated configuration (either "out of the box" or by configuration guidance in the AGD documentation) must enable this functionality.*

***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.*

*TEST*

*The evaluator simply follows the guidance to configure the TOE/platform to perform the following tests.*

*Test 1: This test shall be performed for each version of IKE supported. 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. 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. 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. 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.*

**FCS_IPSEC_EXT.1.13** The [selection: MDM Server, MDM Server platform] shall ensure that all IKE protocols perform peer authentication using a [selection: RSA, ECDSA] that use X.509v3 certificates that conform to RFC 4945 and [selection: Pre-shared Keys, no other method].

*Application Note:*

*At least one public-key-based Peer Authentication method is required in order to conform to this PP; one or more of the public key schemes is chosen by the ST author to reflect what is implemented. 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). Peer authentication using ECDSA X.509v3 certificates will be required for TOEs entering evaluation after Quarter 3, 2015.*

***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. The description in the TSS and the operational guidance shall also indicate how pre-shared key establishment is accomplished for 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 certificates with RSA and/or ECDSA signatures and public keys.*

*In order to construct the environment and configure the TOE for the following tests, the evaluator will ensure that the operational guidance 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 may be combined with the testing for FIA_X509_EXT.1, FIA_X509_EXT.2 (for IPsec connections), and FCS_IPSEC_EXT.1.13. The following tests shall be repeated for each peer authentication selected in the FCS_IPSEC_EXT.1.12 selection above:*

*Test 1: The evaluator shall configure the TOE to use a private key and associated certificate signed by a trusted CA and shall establish an IPsec connection with the peer.*

*Test 2: [conditional] The evaluator shall generate a pre-shared key off-TOE and use it, as indicated in the operational guidance, to establish an IPsec connection with the peer.*

**FCS_IPSEC_EXT.1.14** The [selection: MDM Server, MDM Server platform] shall only establish a trusted channel to peers with valid certificates.

*Application Note:*

*Supported peer certificate algorithms are the same as FCS_IPSEC_EXT.1.12.*

***Assurance Activity:***

*TEST*

*The evaluator shall use IPsec as a function to verify that TOE adheres to the validation rules in FIA_X509_EXT.1.1.*

**FCS_IPSEC_EXT.1.15** The [selection: MDM Server, MDM Server platform] 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.

*Application Note:*

*The DN may be the FQDN or the Common Name in the certificate. The expected DN for the peer is the configured name of the peer. Matching should be performed by a bit-wise comparison.*

***Assurance Activity:***

*TSS*

*The evaluator shall verify that the TSS describes how the DN in the certificate is compared to the expected DN.*

*GUIDANCE*

*The evaluator shall ensure that the operational guidance includes configuration of the expected DN for the connection.*

*TEST*

*The evaluator shall, if necessary, configure the expected DN according to the operational guidance. The evaluator shall send a peer certificate signed by a trusted CA with a DN that does not match an expected DN and verify that the TOE denies the connection.*

**FCS_IPSEC_EXT.1.16** The [selection: MDM Server, MDM Server platform] shall ensure that IKEv1 Phase 1 exchanges use only main mode.

*Application Note:*

*FCS_IPSEC_EXT.1.16 is only applicable if IKEv1 is selected in FCS_IPSEC_EXT.1.5.*

***Assurance Activity:***

*TSS*

*The evaluator shall examine the TSS to ensure that, in the description of the IPsec protocol, 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/platform prior to its operation, the evaluator shall check the operational guidance to ensure that instructions for this configuration are contained within that guidance.*

*TEST*

*The evaluator shall configure the TOE/platform 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.*

#### C.2.1.4 SSH Protocol

**FCS_SSHS_EXT.1.1** The [selection: MDM Server, MDM Server platform] shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254, and [selection: 5647, 5656, 6187, 6668, no other RFCs].

*Application Note:*

*The ST author selects which of the additional RFCs to which conformance is being claimed. Note that these need to be consistent with selections in later elements of this component (e.g., cryptographic algorithms permitted). RFC 4253 indicates that certain cryptographic algorithms are "REQUIRED". This means that the implementation must include support, not that the algorithms must be enabled for use. Ensuring that algorithms indicated as "REQUIRED" but not listed in the later elements of this component are implemented is out of scope of the assurance activity for this requirement.*

**FCS_SSHS_EXT.1.2** The [selection: MDM Server, MDM Server platform] shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, password-based.

***Assurance Activity**:*

*TSS*

*The evaluator shall check to ensure that the TSS contains a description of the public key algorithms that are acceptable for use for authentication, that this list conforms to FCS_SSHS_EXT.1.5, and ensure that password-based authentication methods are also allowed.*

*TEST*

*The evaluator shall also perform the following tests:*

*Test 1: The evaluator shall, for each public key algorithm supported, show that the TOE supports the use of that public key algorithm to authenticate a user connection. Any configuration activities required to support this test shall be performed according to instructions in the operational guidance.*

*Test 2: The evaluator shall choose one public key algorithm supported by the TOE. The evaluator shall generate a new key pair for that algorithm without configuring the TOE to recognize the public key for authentication. The evaluator shall use an SSH client to attempt to connect to the TOE with the new key pair and demonstrate that authentication fails.*

*Test 3: Using the operational guidance, the evaluator shall configure the TOE to accept password-based authentication, and demonstrate that a user can be successfully authenticated to the TOE over SSH using a password as an authenticator.*

*Test 4: The evaluator shall use an SSH client, enter an incorrect password to attempt to authenticate to the TOE, and demonstrate that the authentication fails.*

**FCS_SSHS_EXT.1.3** The [selection: MDM Server, MDM Server platform] shall ensure that, as described in RFC 4253, packets greater than [*assignment: number of bytes*] bytes in an SSH transport connection are dropped.

*Application Note:*

*RFC 4253 provides for the acceptance of "large packets" with the caveat that the packets should be of "reasonable length" or dropped. The assignment should be filled in by the ST author with the maximum packet size accepted, thus defining "reasonable length" for the TOE.*

***Assurance Activity:***

*TSS*

*The evaluator shall check that the TSS describes how "large packets" in terms of RFC 4253 are detected and handled.*

*TEST*

*The evaluator shall demonstrate that if the TOE receives a packet larger than that specified in this component, that packet is dropped.*

**FCS_SSHS_EXT.1.4** The [selection: MDM Server, MDM Server platform] shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: aes128-cbc, aes256-cbc, [selection: AEAD_AES_128_GCM, AEAD_AES_256_GCM, no other algorithms].

*Application Note:*

*RFC 5647 specifies the use of the AEAD_AES_128_GCM and AEAD_AES_256_GCM algorithms in SSH. As described in RFC 5647, AEAD_AES_128_GCM and AEAD_AES_256_GCM can only be chosen as encryption algorithms when the same algorithm is being used as the MAC algorithm. In the assignment, the ST author can select the AES-GCM algorithms, or "no other algorithms" if AES-GCM is not supported. If AES-GCM is selected, there should be corresponding FCS_COP entries in the ST.*

***Assurance Activity:***

*TSS*

*The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that optional characteristics are specified, and the encryption algorithms supported are specified as well. The evaluator shall check the TSS to ensure that the encryption algorithms specified are identical to those listed for this component.*

*GUIDANCE*

*The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that SSH conforms to the description in the TSS (for instance, the set of algorithms advertised by the TOE may have to be restricted to meet the requirements).*

*TEST*

*The evaluator shall also perform the following tests:*

*Test 1: The evaluator shall establish a SSH connection using each of the encryption algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.*

*Test 2: The evaluator shall configure an SSH client to only allow the 3des-cbc encryption algorithm and no other encryption algorithms. The evaluator shall attempt to establish an SSH connection from the SSH client to the TOE and observe that the connection is rejected.*

**FCS_SSHS_EXT.1.5** The [selection: MDM Server, MDM Server platform] shall ensure that the SSH transport implementation uses [selection: ssh-rsa, ecdsa-sha2-nistp256] and [selection: ecdsa-sha2-nistp384, x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384, no other public key algorithms] as its public key algorithm(s) and rejects all other public key algorithms.

*Application Note:*

*Implementations that select only ssh-rsa will not achieve the 112-bit security strength in the digital signature generation for SSH authentication as is recommended in NIST SP 800-131A. Future versions of this profile may remove ssh-rsa as a selection.*

***Assurance Activity:***

*TSS*

*The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that optional characteristics are specified, and the public key algorithms supported are specified as well. The evaluator shall check the TSS to ensure that the public key algorithms specified are identical to those listed for this component.*

*GUIDANCE*

*The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that SSH conforms to the description in the TSS (for instance, the set of algorithms advertised by the TOE may have to be restricted to meet the requirements).*

*TEST*

*The evaluator shall also perform the following test:*

*Test 1: The evaluator shall establish a SSH connection using each of the public key algorithms specified by the requirement to authenticate the TOE to an SSH client. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.*

*Test 2: The evaluator shall configure an SSH client to only allow the ssh-dsa public key algorithm and no other public key algorithms. The evaluator shall attempt to establish an SSH connection from the SSH client to the TOE and observe that the connection is rejected.*

**FCS_SSHS_EXT.1.6** The [selection: MDM Server, MDM Server platform] shall ensure that the SSH transport implementation uses [selection: hmac-sha1, hmac-sha1-96, hmac-sha2-256, hmac-sha2-512] and [selection: AEAD_AES_128_GCM, AEAD_AES_256_GCM, no other MAC algorithms] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

*Application Note:*

*RFC 5647 specifies the use of the AEAD_AES_128_GCM and AEAD_AES_256_GCM algorithms in SSH. As described in RFC 5647, AEAD_AES_128_GCM and AEAD_AES_256_GCM can only be chosen as MAC algorithms when the same algorithm is being used as the encryption algorithm. RFC 6668 specifies the use of the sha2 algorithms in SSH.*

***Assurance Activity:***

*TSS*

*The evaluator shall check the TSS to ensure that it lists the supported data integrity algorithms, and that that list corresponds to the list in this component.*

*GUIDANCE*

*The evaluator shall also check the operational guidance to ensure that it contains instructions to the administrator on how to ensure that only the allowed data integrity algorithms are used in SSH connections with the TOE (specifically, that the "none" MAC algorithm is not allowed).*

*TEST*

*The evaluator shall also perform the following test:*

*Test 1: The evaluator shall establish a SSH connection using each of the integrity algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.*

*Test 2: The evaluator shall configure an SSH client to only allow the none MAC algorithm. The evaluator shall attempt to connect from the SSH client to the TOE and observe that the attempt fails.*

*Test 3: The evaluator shall configure an SSH client to only allow the hmac-md5 MAC algorithm. The evaluator shall attempt to connect from the SSH client to the TOE and observe that the attempt fails.*

**FCS_SSHS_EXT.1.7** The [selection: MDM Server, MDM Server platform] shall ensure that [selection: diffie-hellman-group14-sha1, ecdh-sha2-nistp256] and [selection: ecdh-sha2-nistp384, ecdh-sha2-nistp521, no other methods] are the only allowed key exchange methods used for the SSH protocol.

***Assurance Activity:***

*TSS*

*The evaluator shall check the TSS to ensure that it lists the supported key exchange algorithms, and that that list corresponds to the list in this component.*

*GUIDANCE*

*The evaluator shall also check the operational guidance to ensure that it contains instructions to the administrator on how to ensure that only the allowed key exchange algorithms are used in SSH connections with the TOE.*

*TEST*

*The evaluator shall also perform the following tests:*

*Test 1: The evaluator shall configure an SSH client to only allow the diffie-hellman-group1-sha1 key exchange. The evaluator shall attempt to connect from the SSH client to the TOE and observe that the attempt fails.*

*Test 2: For each allowed key exchange method, the evaluator shall configure an SSH client to only allow that method for key exchange, attempt to connect from the client to the TOE, and observe that the attempt succeeds.*

**FCS_SSHS_EXT.1.8** The MDM Server shall ensure that the SSH connection be rekeyed after no more than 2^28 packets have been transmitted using that key.

***Assurance Activity:***

*TEST*

*The evaluator shall configure the TOE to create a log entry when a rekey occurs. The evaluator shall connect to the TOE with an SSH client and cause 2^28 packets to be transmitted from the client to the TOE, and subsequently review the audit log to ensure that a rekey occurred.*

#### C.2.1.5 TLS Client Protocol

**FCS_TLSC_EXT.1.5** The [selection: TSF, TOE platform] shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [selection: secp256r1, secp384r1, secp521r1] and no other curves.

*Application Note:*

*This requirement limits the elliptic curves allowed for authentication and key agreement to the NIST curves from FCS_COP.1(3) and FCS_CKM.1 and FCS_CKM.2. This extension is required for clients supporting Elliptic Curve ciphersuites.*

***Assurance Activity:***

*TSS*

*The evaluator shall verify that TSS describes the Supported Elliptic Curves Extension and whether the required behavior is performed by default or may be configured.*

*GUIDANCE*

*If the TSS indicates that the Supported Elliptic Curves Extension must be configured to meet the requirement, the evaluator shall verify that AGD guidance includes configuration of the Supported Elliptic Curves Extension.*

*TEST*

*Test 1: The evaluator shall configure the server to perform an ECDHE key exchange message in the TLS connection using a non-supported ECDHE curve (for example, P-192) and shall verify that the TOE disconnects after receiving the server's Key Exchange handshake message.*

#### C.2.1.6 TLS Server Protocol

**FCS_TLSS_EXT.1.1** The [selection: MDM Server, MDM Server platform] shall implement [selection: *TLS 1.0 (RFC 2246), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)*] supporting the following ciphersuites: [

- Mandatory Ciphersuites:

  - TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246
- [selection: Optional Ciphersuites:
  - *TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246*
  - *TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246*
  - *TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246*
  - *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492*
  - *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492*
  - *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492*
  - *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492*
  - *TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246*
  - *TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246*
  - *TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256 as defined in RFC 5246*
  - *TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246*
  - *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289*
  - *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289*
  - *TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*
  - *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*
  - *no other ciphersuite*]].

*Application Note:*

*The MDM Server must support all versions of TLS supported by evaluated Agents listed in the ST as supported in the MDM System. TLS 1.0 and TLS 1.1 are currently allowed due to lack of support for TLS 1.2. TLS 1.0 and TLS 1.1 do not have the extensions necessary to assure a connection with security strength of 112-bits or better. TLS 1.2 will be required for those TOEs entering evaluation after Quarter 3 2015.*

*The ciphersuites to be tested in the evaluated configuration are limited by this requirement. The ST author should select the optional ciphersuites that are supported; if there are no ciphersuites supported other than the mandatory suites, then "None" should be selected. If administrative steps need to be taken so that the suites negotiated by the implementation are limited to those in this requirement, the appropriate instructions need to be contained in the guidance called for by AGD_OPE. FMT_SMF.1 addresses configuration of the ciphersuite to be used for connections. The Suite B algorithms listed above (RFC 6460) are the preferred algorithms for implementation.*

*These requirements will be revisited as new TLS versions are standardized by the IETF.*

*If any ciphersuites are selected using DHE or ECDHE, then FCS_TLSS_EXT.1.6 is required.*

***Assurance Activity:***

*TSS*

*The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the ciphersuites supported are specified. The evaluator shall check the TSS to ensure that the ciphersuites specified are identical to those listed for this component. The evaluator shall check the TSS to account for all versions of TLS supported by an evaluated client.*

*GUIDANCE*

*The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS (for instance, the set of ciphersuites advertised by the TOE may have to be restricted to meet the requirements).*

*TEST*

*The evaluator shall also perform the following tests:*

*Test 1: The evaluator shall establish a TLS connection using each of the ciphersuites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an EAP session. It is sufficient to observe the successful negotiation of a ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).*

*Test 2: The evaluator shall send a Client Hello to the server with a list of ciphersuites that does not contain any of the ciphersuites in the server's ST and verify that the server denies the connection. Additionally, the evaluator shall send a Client Hello to the server containing only the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the server denies the connection.*

*Test 3: The evaluator shall use a client to send a key exchange message in the TLS connection that the does not match the server-selected ciphersuite (for example, send an ECDHE key exchange while using the TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite or send a RSA key exchange while using one of the ECDSA ciphersuites.) The evaluator shall verify that the TOE disconnects after the receiving the key exchange message.*

*Test 4: The evaluator shall perform the following modifications to the traffic:*

- *Modify at a byte in the client's nonce in the Client Hello handshake message, and verify that the server rejects the client's Certificate Verify handshake message (if using mutual authentication) or that the server denies the client's Finished handshake message.*
- *Modify the signature block in the Client's Key Exchange handshake message, and verify that the server rejects the client's Certificate Verify handshake message (if using mutual authentication) or that the server denies the client's Finished handshake message.*
- *Modify a byte in the Client Finished handshake message, and verify that the server rejects the connection and does not send any application data.*
- *After generating a fatal alert by sending a Finished message from the client before the client sends a ChangeCipherSpec message, send a Client Hello with the session identifier from the previous test, and verify that the server denies the connection.*
- *Send a garbled message from the client after the client has issued the ChangeCipherSpec message and verify that the Server denies the connection.*

**FCS_TLSS_EXT.1.2** The [selection: MDM Server, MDM Server platform] shall deny connections from clients requesting SSL 1.0, SSL 2.0, SSL 3.0 and [selection: TLS 1.0, TLS 1.1, no other TLS version].

***Assurance Activity:***

*TSS*

*The evaluator shall verify that the TSS contains a description of the denial of old SSL and TLS versions, and any configuration necessary to meet the requirement must be contained in the AGD guidance.*

*TEST*

*The evaluator shall send a Client Hello requesting a connection with version SSL 1.0 and verify that the server denies the connection. The evaluator shall repeat this test with SSL 2.0 and SSL 3.0 and any selected TLS versions.*

**FCS_TLSS_EXT.1.3** The [selection: MDM Server, MDM Server platform] shall support mutual authentication of TLS clients using X.509v3 certificates.

**FCS_TLSS_EXT.1.4** The [selection: MDM Server, MDM Server platform] shall not establish a trusted channel if the peer certificate is invalid.

*Application Note:*

*The use of X.509v3 certificates for TLS is addressed in FIA_X509_EXT.2.1. This requirement adds that this use must include support for client-side certificates for TLS mutual authentication.*

*Validity is determined by the certificate path, the expiration date, and the revocation status in accordance with RFC 5280. Certificate validity shall be tested in accordance with testing performed for FIA_X509_EXT.1.*

***Assurance Activity:***

*TSS*

*The evaluator shall ensure that the TSS description required per FIA_X509_EXT.2.1 includes the use of client-side certificates for TLS mutual authentication.*

*GUIDANCE*

*The evaluator shall verify that the AGD guidance required per FIA_X509_EXT.2.1 includes instructions for configuring the client-side certificates for TLS mutual authentication.*

*TEST*

*The evaluator shall perform the following mutual authentication tests:*

*Test 1: The evaluator shall configure the server to send a certificate request to the client and shall attempt a connection without sending a certificate from the client. The evaluator shall verify that the connection is denied.*

*Test 2: The evaluator shall configure the server to send a certificate request to the client without the supported_signature_algorithm used by the client's certificate. The evaluator shall attempt a connection using the client certificate and verify that the connection is denied.*

*Test 3: The evaluator shall demonstrate that using a certificate without a valid certification path results in the function failing. Using the administrative guidance, 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.*

*Test 4: The evaluator shall configure the client to send a certificate that does not chain to one of the Certificate Authorities (either a Root or Intermediate CA) in the server's Certificate Request message. The evaluator shall verify that the attempted connection is denied.*

*Test 5: The evaluator shall configure the client to send a certificate with the Client Authentication purpose in the extendedKeyUsage field and verify that the server accepts the attempted connection. The evaluator shall repeat this test without the Client Authentication purpose and shall verify that the server* *denies the connection. Ideally, the two certificates should be identical except for the Client Authentication purpose.*

*Test 6: The evaluator shall perform the following modifications to the traffic:*

- *Configure the server to require mutual authentication and then modify a byte in the client's certificate. The evaluator shall verify that the server rejects the connection.*
- *Configure the server to require mutual authentication and then modify a byte in the client's Certificate Verify handshake message. The evaluator shall verify that the server rejects the connection.*

**FCS_TLSS_EXT.1.5** The [selection: MDM Server, MDM Server platform] shall not establish a trusted channel if the distinguished name (DN) or Subject Alternative Name (SAN) contained in a certificate does not match the expected identifier for the peer.

*Application Note:*

*This requirement only applies to those TOEs performing mutually-authenticated TLS (FCS_TLSS_EXT.1.4). The peer identifier may be in the Subject field or the Subject Alternative Name extension of the certificate. The expected identifier may either be configured, may be compared to the Domain Name, IP address, username, or email address used by the peer, or may be passed to a directory server for comparison. Matching should be performed by a bit-wise comparison.*

***Assurance Activity:***

*TSS*

*If the TOE implements mutual authentication, the evaluator shall verify that the TSS describes how the DN and SAN in the certificate is compared to the expected identifier.*

*GUIDANCE*

*If the DN is not compared automatically to the Domain Name, IP address, username, or email address, the evaluator shall ensure that the AGD guidance includes configuration of the expected identifier or the directory server for the connection.*

*TEST*

*The evaluator shall send a client certificate with an identifier that does not match an expected identifier and verify that the server denies the connection.*

**FCS_TLSS_EXT.1.6** The [selection: MDM Server, MDM Server platform] shall generate key agreement parameters [selection: *over NIST curves [selection: secp256r1, secp384r1] and no other curves; Diffie-Hellman parameters of size 2048 bits and [selection: 3072 bits, no other size]*].

*Application Note:*

*This element may be omitted if no DHE or ECDHE ciphersuites are selected in FCS_TLSS_EXT.1.1.*

*If the ST lists a DHE ciphersuite in FCS_TLSS_EXT.1.1, the ST must include the Diffie-Hellman selection in the requirement. FMT_SMF.1 requires the configuration of the key agreement parameters in order to establish the security strength of the TLS connection.*

***Assurance Activity:***

*TSS*

*The evaluator shall verify that the TSS describes the key agreement parameters of the server key exchange message.*

*GUIDANCE*

*The evaluator shall verify that any configuration guidance necessary to meet the requirement must be contained in the AGD guidance.*

*TEST*

*The evaluator shall attempt a connection using an ECDHE ciphersuite and a configured curve and, using a packet analyzer, verify that the key agreement parameters in the Key Exchange message are the ones configured. (Determining that the size matches the expected size for the configured curve is sufficient.) The evaluator shall repeat this test for each supported NIST Elliptic Curve and each supported Diffie-Hellman key size.*

# D. OBJECTIVE REQUIREMENTS

As indicated in the introduction to this PP, the baseline requirements (those that must be performed by the TOE or its underlying platform) are contained in the body of this PP. There are additional requirements that specify security functionality that is desirable and these requirements are contained in this Appendix. It is expected that these requirements will transition from objective requirements to baseline requirements in future versions of this PP.

At any time these may be included in the ST such that the TOE is still conformant to this PP.

This Appendix is divided into two subsections: objective requirements that may be performed by the TSF and objective requirements that may be performed by the MDM Server or its underlying platform.

## D.1. Objective TSF Requirements

### D.1.1. Security Audit (FAU)

#### D.1.1.1 Support for Compliance Reporting of Mobile Device Configuration

**FAU_CRP_EXT.1.1** The MDM Server shall provide [selection: an interface that provides responses to queries about the configuration of enrolled devices, an interface that permits the export of data about the configuration of enrolled devices] to authorized entities over an authenticated [selection: TLS/HTTPS, TLS, DTLS, IPsec, SSH] trusted communication channel. The provided information for each enrolled mobile device includes:

a. The current version of the MD firmware/software
b. The current version of the hardware model of the device
c. The current version of installed mobile applications
d. List of MD configuration policies that are in place on the device (as defined in FMT_SMF.1.1(1))
e. [selection: [assignment: list of other available information about enrolled devices], no other information].

*Application Note:*

*The intent of this requirement is that the MDM Server be able to provide compliance information about enrolled mobile devices for use by other enterprise security infrastructure systems. There are active standards efforts underway by the Internet Engineering Task Force (IETF) Security Automation and Continuous Monitoring (SACM) Working Group and others to define protocols and standards to assess and report upon endpoint device posture. We expect that this requirement will evolve in future versions of this Protection Profile as standards efforts mature.*

***Assurance Activity:***

*GUIDANCE*

*The evaluator shall check to ensure that the operational guidance contains instructions on how to access the MDM Server's compliance reporting interface.*

*TEST*

*Test 1: Using the operational guidance, the evaluator shall demonstrate the ability to access the compliance reporting interface from an authorized entity and successfully obtain information about enrolled devices.*

*Test 2: The evaluator shall attempt to access the compliance reporting interface from an unauthorized entity and demonstrate that the attempt is denied.*

### D.1.2. Security Management (FMT)

#### D.1.2.1 Trusted Policy Update

**FMT_POL_EXT.1.1** The MDM Server shall provide digitally signed policies and policy updates to the MDM Agent.

*Application Note:*

*The intent of this requirement is to cryptographically tie the policies to the enterprise that mandated the policy, not to protect the policies in transit (as they are already protected by FPT_ITT.1(2) or FTP_ITC.1(2)). This is especially critical for users who connect to multiple enterprises.*

***Assurance Activity:***

*TSS*

*Policies must be digitally signed by the enterprise using the algorithms in FCS_COP.1(1). The evaluator shall ensure that the TSS describes how policies are digitally signed by the TSF.*

*GUIDANCE*

*If applicable, the evaluator shall verify that the AGD guidance instructs administrators on configuring the Enterprise certificate to be used for signing policies or signing the policies before applying them.*

*TEST*

*The evaluator shall perform a policy update in accordance with FMT_SMF.1(1). The evaluator shall examine the policy either at the MDM Server, in transmission, or at the MDM agent, and verify the TSF signs the update and provides it to the MDM Agent.*

## D.2. Objective TOE or Platform Requirements

### D.2.1. Cryptographic Support (FCS)

#### D.2.1.1 TLS Client Protocol

**FCS_TLSC_EXT.1.6** The [selection: TSF, TOE platform] shall present the signature_algorithms extension in the Client Hello with the supported_signature_algorithms value containing the following hash algorithms: [selection: SHA256, SHA384, SHA512] and no other hash algorithms.

*Application Note:*

*This requirement limits the hashing algorithms supported for the purpose of digital signature verification by the client and limits the server to the supported hashes for the purpose of digital signature generation by the server. The signature_algorithm extension is only supported by TLS 1.2.*

***Assurance Activity:***

*TSS*

*The evaluator shall verify that TSS describes the signature_algorithm extension and whether the required behavior is performed by default or may be configured.*

*GUIDANCE*

*If the TSS indicates that the signature_algorithm extension must be configured to meet the requirement, the evaluator shall verify that AGD guidance includes configuration of the signature_algorithm extension.*

*TEST*

*The evaluator shall configure the server to send a certificate in the TLS connection that is not supported according to the Client's HashAlgorithm enumeration within the signature_algorithms extension (for example, send a certificate with a SHA-1 signature). The evaluator shall verify that the TOE disconnects after receiving the server's Certificate handshake message.*

**FCS_TLSC_EXT.1.7** The [selection: TSF, TOE platform] shall support secure renegotiation through use of the "renegotiation_info" TLS extension in accordance with RFC 5746.

**FCS_TLSC_EXT.1.8** The [selection: TSF, TOE platform] shall include [selection: choose only one of: renegotiation_info extension, TLS_EMPTY_RENEGOTIATION_INFO_SCSV ciphersuite] in the ClientHello message.

*Application Note:*

*RFC 5746 defines an extension to TLS that binds renegotiation handshakes to the cryptography in the original handshake.*

*The ciphersuite included in the selection is a means for clients to be compatible with servers that don't support the extension. It is recommended that client implementations support both the ciphersuite and the extension.*

***Assurance Activity:***

*TEST*

*Test 1: The evaluator shall use a network packet analyzer/sniffer to capture the traffic between the two TLS endpoints. The evaluator shall verify that either the "renegotiation_info" field or the SCSV ciphersuite is included in the ClientHello packet during the initial handshake.*

*Test 2: The evaluator shall verify the Client's handling of ServerHello messages received during the initial handshake that include the "renegotiation_info" extension. The evaluator shall modify the length portion of this field in the ServerHello message to be non-zero and verify that the client sends a failure and terminates the connection. The evaluator shall verify that a properly formatted field results in a successful TLS connection.*

*Test 3: The evaluator shall verify that ServerHello messages received during secure renegotiation contain the "renegotiation_info" extension. The evaluator shall modify either the "client_verify_data" or "server_verify_data" value and verify that the client terminates the connection.*

#### D.2.1.2 TLS Server Protocol

**FCS_TLSS_EXT.1.7** The [selection: MDM Server, MDM Server platform] shall present the HashAlgorithm enumeration in supported_signature_algorithms in the Certificate Request with the following hash algorithms: [selection: *SHA256, SHA384, SHA512*] and no other hash algorithms.

*Application Note:*

*This requirement limits the hashing algorithms supported for the purpose of digital signature verification by the server and limits the client to the supported hashes for the purpose of digital signature generation by the client. The supported_signature_algorithms is only supported by TLS 1.2.*

***Assurance Activity:***

*TSS*

*(Conditional) The evaluator shall verify that TSS describes the supported_signature_algorithms field of the Certificate Request and whether the required behavior is performed by default or may be configured. If the TSS indicates that the supported_signature_algorithms field must be configured to meet the requirement, the evaluator shall verify that AGD guidance includes configuration of the supported_signature_algorithms field.*

*TEST*

*The evaluator shall configure a client to send the signature_algorithms extension indicating that hash algorithm used by the server's certificate is not supported. The evaluator shall attempt a connection and verify that the server denies the client's connection.*

**FCS_TLSS_EXT.1.8** The [selection: MDM Server, MDM Server platform] shall support the "renegotiation_info" TLS extension in accordance with RFC 5746.

**FCS_TLSS_EXT.1.9** The [selection: MDM Server, MDM Server platform] shall include the renegotiation_info extenstion in ServerHello messages.

*Application Note:*

*RFC 5746 defines an extension to TLS that binds renegotiation handshakes to the cryptography in the original handshake.*

***Assurance Activity:***

*TEST*

*The following tests require connection with a client that supports secure renegotiation and the "renegotiation_info" extension.*

*Test 1: The evaluator shall use a network packet analyzer/sniffer to capture the traffic between the two TLS endpoints. The evaluator shall verify that the "renegotiation_info" field is included in the ServerHello packet.*

*Test 2: The evaluator shall modify the length portion of the field in the ClientHello message in the initial handshake to be non-zero and verify that the client sends a failure and terminates the connection. The evaluator shall verify that a properly formatted field results in a successful TLS connection.*

*Test 3: The evaluator shall modify the "client_verify_data" or "server_verify_data" value in the ClientHello message received during secure renegotiation and verify that the server terminates the connection.*

### D.2.2. Identification and Authentication

#### D.2.2.1 X509 Enrollment

**FIA_X509_EXT.3.1** The [selection: MDM Server, MDM Server platform] shall generate a Certificate Request Message as specified by RFC 2986 and be able to provide the following information in the request: public key and [selection: device-specific information, Common Name, Organization, Organizational Unit, Country].

*Application Note:*

*The public key is the public key portion of the public-private key pair generated by the TOE as specified in FCS_CKM.1.1.*

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

**FIA_X509_EXT.3.2** The [selection: MDM Server, MDM Server platform] shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

***Assurance Activity:***

*TSS*

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

*GUIDANCE*

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

*TEST*

*Test 1: The evaluator shall use the operational guidance to cause the TOE to generate a certificate request message. The evaluator shall capture the generated message and ensure that it conforms to the format specified. The evaluator shall confirm that the certificate request provides the public key and other required information, including any necessary user-input information.*

*Test 2: The evaluator shall demonstrate that validating a certificate response message without a valid certification path results in the function failing. The evaluator shall then load a certificate or certificates as trusted CAs needed to validate the certificate response message, and demonstrate that the function succeeds. The evaluator shall then delete one of the certificates, and show that the function fails.*

#### D.2.2.2 Alternate X509 Enrollment

**FIA_X509_EXT.4.1** The TSF shall use the Enrollment over Secure Transport (EST) protocol as specified in RFC 7030 to request certificate enrollment using the simple enrollment method described in RFC 7030 Section 4.2.

**FIA_X509_EXT.4.2** The TSF shall be capable of authenticating EST requests using an existing certificate and corresponding private key as specified by RFC 7030 Section 3.3.2.

**FIA_X509_EXT.4.3** The TSF shall be capable of authenticating EST requests using HTTP Basic Authentication with a username and password as specified by RFC 7030 Section 3.2.3.

**FIA_X509_EXT.4.4** The TSF shall perform authentication of the EST server using an Explicit Trust Anchor following the rules described in RFC 7030, section 3.6.1.

*Application Note:*

EST also uses HTTPS as specified in FCS_HTTPS_EXT.1 to establish a secure connection to an EST server, and thus, the ST author must also include FCS_HTTPS_EXT.1 and FCS_TLSC_EXT.1 (in Appendix D) in the main body of the ST. The separate Trust Anchor Database dedicated to EST operations is described as Explicit Trust Anchors in RFC 7030.

**FIA_X509_EXT.4.5** The TSF shall be capable of requesting server-provided private keys as specified in RFC 7030 Section 4.4.

**FIA_X509_EXT.4.6** The TSF shall be capable of updating its EST-specific Trust Anchor Database using the “Root CA Key Update” process described in RFC 7030 Section 4.1.3.

**FIA_X509_EXT.4.7** The TSF shall generate a Certificate Request Message for EST as specified in RFC 2986 and be able to provide the following information in the request: public key and [selection: device-specific information; Common Name, Organization, Organizational Unit, and Country].

**FIA_X509_EXT.4.8** The TSF shall validate the chain of certificates from the Root CA certificate in the Trust Anchor Database to the EST Server CA certificate upon receiving a CA Certificates Response.

*Application Note:*

The public key referenced in FIA_X509_EXT.4.7 is the public key portion of the public-private key pair generated by the TOE as specified in FCS_CKM.1(2).

***Assurance Activity:***

*GUIDANCE*

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

*TEST*

*The evaluator shall also perform the following tests. Other tests are performed in conjunction with the Assurance Activity listed for FCS_TLSC_EXT.1.*

*Test 1: The evaluator shall use the operational guidance to cause the TOE to request certificate enrollment from an EST server using the simple enrollment method described in RFC 7030 Section 4.2, authenticating the certificate request to the server using an existing certificate and private key as described by RFC 7030 Section 3.3.2. The evaluator shall confirm that the resulting certificate is successfully obtained and installed in the TOE key store.*

*Test 2: The evaluator shall use the operational guidance to cause the TOE to request certificate enrollment from an EST server using the simple enrollment method described in RFC 7030 Section 4.2, authenticating the certificate request to the server using a username and password as described by RFC 7030 Section 3.2.3. The evaluator shall confirm that the resulting certificate is successfully obtained and installed in the TOE key store.*

*Test 3: The evaluator shall modify the EST server to return a certificate containing a different public key than the key included in the TOE’s certificate request. The evaluator shall use the operational guidance to cause the TOE to request certificate enrollment from an EST server. The evaluator shall confirm that* *the TOE does not accept the resulting certificate since the public key in the issued certificate does not match the public key in the certificate request.*

*Test 4: The evaluator shall configure the EST server or use a man-in-the-middle tool to present a server certificate to the TOE that is present in the TOE general Trust Anchor Database but not its EST-specific Trust Anchor Database. The evaluator shall cause the TOE to request certificate enrollment from the EST server. The evaluator shall verify that the request is not successful.*

*Test 5: The evaluator shall configure the EST server or use a man-in-the-middle tool to present an invalid certificate. The evaluator shall cause the TOE to request certificate enrollment from the EST server. The evaluator shall verify that the request is not successful. The evaluator shall configure the EST server or use a man-in-the-middle tool to present a certificate that does not have the CMC RA purpose and verify that requests to the EST server fail. The tester shall repeat the test using a valid certificate and a certificate that contains the CMC RA purpose and verify that the certificate enrollment requests succeed.*

*Test 6: The evaluator shall use a packet sniffing tool between the TOE and an EST server. The evaluator shall turn on the sniffing tool and cause the TOE to request certificate enrollment from an EST server. The evaluator shall verify that the EST protocol interaction occurs over a Transport Layer Security (TLS) protected connection. The evaluator is not expected to decrypt the connection but rather observe that the packets conform to the TLS protocol format.*

*Test 7: The evaluator shall use the operational guidance to cause the TOE to request a server-provided private key and certificate from an EST server. The evaluator shall confirm that the resulting private key and certificate are successfully obtained and installed in the TOE key store.*

*Test 8: The evaluator shall modify the EST server to, in response to a server-provided private key and certificate request, return a private key that does not correspond with the public key in the returned certificate. The evaluator shall use the operational guidance to cause the TOE to request a server-provided private key and certificate. The evaluator shall confirm that the TOE does not accept the resulting private key and certificate since the private key and public key do not correspond.*

*Test 9: The evaluator shall configure the EST server to provide a "Root CA Key Update" as described in RFC 7030 Section 4.1.3. The evaluator shall cause the TOE to request CA certificates from the EST server and shall confirm that the EST-specific Trust Anchor Database is updated with the new trust anchor.*

*Test 10: The evaluator shall configure the EST server to provide a "Root CA Key Update" as described in RFC 7030 Section 4.1.3, but shall modify part of the NewWithOld certificate's generated signature. The evaluator shall cause the TOE to request CA certificates from the EST server and shall confirm that the EST-specific Trust Anchor Database is not updated with the new trust anchor since the signature did not verify.*

*Test 11: The evaluator shall use the operational guidance to cause the TOE to generate a certificate request message. The evaluator shall capture the generated message and ensure that it conforms with the format specified by RFC 2986. The evaluator shall confirm that the certificate request provides the public key and other required information, including any necessary user-input information.*

# E. ENTROPY DOCUMENTATION AND ASSESSMENT

This appendix describes the required supplementary information for each entropy source used by the TOE.

The documentation of the entropy source(s) should be detailed enough that, after reading, the evaluator will thoroughly understand the entropy source and why it can be relied upon to provide sufficient 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.

## E.1. Design Description

Documentation shall include the design of each entropy source as a whole, including the interaction of all entropy source components. Any information that can be shared regarding the design should also be included for any third-party entropy sources that are included in the product.

The documentation will describe the operation of the entropy source to include, 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 entropy comes from, where the entropy output 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.

If implemented, the design description shall include a description of how third-party applications can add entropy to the RBG. A description of any RBG state saving between power-off and power-on shall be included.

## E.2. 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 delivering sufficient entropy for the uses made of the RBG output (by this particular TOE). This argument will include a description of the expected min-entropy rate (i.e. the minimum entropy (in bits) per bit or byte of source data) and explain 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.

The amount of information necessary to justify the expected min-entropy rate depends on the type of entropy source included in the product.

For developer provided entropy sources, in order to justify the min-entropy rate, it is expected that a large number of raw source bits will be collected, statistical tests will be performed, and the min-entropy rate determined from the statistical tests. While no particular statistical tests are required at this time, it is expected that some testing is necessary in order to determine the amount of min-entropy in each output.

For third party provided entropy sources, in which the TOE vendor has limited access to the design and raw entropy data of the source, the documentation will indicate an estimate of the amount of min-entropy obtained from this third-party source. It is acceptable for the vendor to "assume" an amount of min-entropy, however, this assumption must be clearly stated in the documentation provided. In particular, the min-entropy estimate must be specified and the assumption included in the ST.

Regardless of type of entropy source, the justification will also include how the DRBG is initialized with the entropy stated in the ST, for example by verifying that the min-entropy rate is multiplied by the amount of source data used to seed the DRBG or that the rate of entropy expected based on the amount of source data is explicitly stated and compared to the statistical rate. If the amount of source data used to seed the DRBG is not clear or the calculated rate is not explicitly related to the seed, the documentation will not be considered complete.

The entropy justification shall not include any data added from any third-party application or from any state saving between restarts.

## E.3. Operating Conditions

The entropy rate may be affected by conditions outside the control of the entropy source itself. For example, voltage, frequency, temperature, and elapsed time after power-on are just a few of the factors that may affect the operation of the entropy source. As such, documentation will also include the range of operating conditions under which the entropy source is expected to generate random data. Similarly, documentation shall describe the conditions under which the entropy source is no longer guaranteed to provide sufficient entropy. Methods used to detect failure or degradation of the source shall be included.

## E.4. 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, TOE behavior upon entropy source failure, and rationale indicating why each test is believed to be appropriate for detecting one or more failures in the entropy source.

# F. GLOSSARY

## F.1. Technical Definitions

| | |
|---|---|
| Administrator | The Administrator is responsible for management activities, including setting the policy that is applied by the enterprise on the Mobile Device. This administrator is the Mobile Device Management (MDM) Administrator, acting through an MDM Agent. |
| Data | Program/application or data files that are stored or transmitted by a server or mobile device (MD). |
| Developer Modes | Developer modes are states in which additional services are available to a user in order to provide enhanced system access for debugging of software. Developer modes are states in which additional services are available to a user in order to provide enhanced system access for debugging of software. For the purpose of this profile, these modes also include boot modes which are not verified according to FPT_TUD_EXT.2. |
| Enrolled state | The state in which a Mobile Device is managed by a policy from an MDM system. |
| Enterprise Applications | Applications that are provided and managed by the enterprise. |
| Enterprise Data | Enterprise data is any data residing in the enterprise servers, or temporarily stored on mobile devices to which the mobile device user is allowed access according to security policy defined by the enterprise and implemented by the administrator. |
| Key Encryption Key (KEK) | A key used to encrypt other keys, such as DEKs or storage that contains keys. |
| Locked State | Powered on but most functionality is unavailable for use. User authentication is required to access functionality (when so configured). |
| MD | Mobile Device |
| MDM Agent | The MDM Agent is installed on a mobile device as an application or is part of the mobile device's OS. The MDM Agent establishes a secure connection back to the MDM Server controlled by the administrator. |
| Mobile Device User (User) | This is the person who uses and is held responsible for the mobile device's physical control and operation. |
| Operating System (OS) | Software which runs at the highest privilege level and can directly control hardware resources. Modern mobile devices typically have at least two primary operating systems: one which runs on the cellular baseband processor and one which runs on the application processor. The platform of the application processor handles most user interaction and provides the execution environment for apps. The platform of the cellular baseband processor handles communications with the cellular network and may control other peripherals. The term OS, without context, may be assumed to refer to the platform of the application processor. |
| Powered-Off State | The device has been shutdown. |
| Protected Data | Protected data is all non-TSF data on the MD, including all user or enterprise data. Protected data is encrypted while the MD is powered off. Protected data includes all keys in software-based secure key storage. Some or all of this data may be considered sensitive data as well. |
| Root Encryption Key (REK) | A key tied to the device used to encrypt other keys. |

| | |
|---|---|
| Sensitive data | Sensitive data shall be identified by the MD's TSS. Sensitive data may include all user or enterprise data or may be specific application data such as emails, messaging, documents, calendar items, and contacts. Sensitive data is optionally protected while in the locked state by the MD. Sensitive data must minimally include some or all keys in software-based key storage. |
| Trust Anchor Database | A list of trusted root Certificate Authority certificates. |
| TSF Data | Data for the operation of the TSF upon which the enforcement of the requirements relies. |
| Unenrolled state | The state in which the Mobile Device is not managed by an MDM system. |
| Unlocked State | Powered on and device functionality is available for use. Implies user authentication has occurred (when so configured). |

## F.2. Common Criteria Definitions

| | |
|---|---|
| Assurance | Grounds for confidence that a TOE meets the SFRs [CC1]. |
| CC | Common Criteria |
| CM | Configuration Management |
| PP | Protection Profile |
| SAR | Security Assurance Requirement |
| SFR | Security Functional Requirement |
| Security Target (ST) | Implementation-dependent statement of security needs for a specific identified TOE. |
| Target of Evaluation (TOE) | Set of software, firmware and hardware under evaluation, possibly accompanied by guidance. |
| TOE Security Functionality (TSF) | Combined functionality of all hardware, software, and firmware of a TOE that must be relied upon for the correct enforcement of the SFRs. |
| TOE Summary Specification (TSS) | Documentation which provides evaluators with a description of the implementation of SFRs in the TOE. |

## G. INITIALIZATION VECTOR REQUIREMENTS

**Table 9: References and IV Requirements for NIST-approved Cipher Modes**

| Cipher Mode | Reference | IV Requirements |
|---|---|---|
| Electronic Codebook (ECB) | SP 800-38A | No IV |
| Counter (CTR) | SP 800-38A | "Initial Counter" shall be non-repeating. No counter value shall be repeated across multiple messages with the same secret key. |
| Cipher Block Chaining (CBC) | SP 800-38A | IVs shall be unpredictable. Repeating IVs leak information about whether the first one or more blocks are shared between two messages, so IVs should be non-repeating in such situations. |
| Output Feedback (OFB) | SP 800-38A | IVs shall be non-repeating and shall not be generated by invoking the cipher on another IV. |
| Cipher Feedback (CFB) | SP 800-38A | IVs should be non-repeating as repeating IVs leak information about the first plaintext block and about common shared prefixes in messages. |
| XEX (XOR Encrypt XOR) Tweakable Block Cipher with Ciphertext Stealing (XTS) | SP 800-38E | No IV. Tweak values shall be non-negative integers, assigned consecutively, and starting at an arbitrary non-negative integer. |
| Cipher-based Message Authentication Code (CMAC) | SP 800-38B | No IV |
| Key Wrap and Key Wrap with Padding | SP 800-38F | No IV |
| Counter with CBC-Message Authentication Code (CCM) | SP 800-38C | No IV. Nonces shall be non-repeating. |
| Galois Counter Mode (GCM) | SP 800-38D | IV shall be non-repeating. The number of invocations of GCM shall not exceed 2^32 for a given secret key unless an implementation only uses 96-bit IVs (default length). |