MobileIron Platform 11 Security Target

Version 0.6
August 31, 2021

Prepared for:

MobileIron, an Ivanti Company

401 East Middlefield Road
Mountain View, CA 94043

Prepared By:

www.gossamersec.com

LIST OF TABLES

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is MobileIron Platform provided by MobileIron. The TOE is being evaluated as a Mobile Device Management (MDM) solution consisting of an MDM server (MobileIron Core) and associated MDM mobile device agents (MobileIron Client).

The Security Target contains the following additional sections:

- Conformance Claims (Section 2)
- Security Objectives (Section 3)
- Extended Components Definition (Section 4)
- Security Requirements (Section 5)
- TOE Summary Specification (Section 6)

Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
    - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a parenthetical number placed at the end of the component. For example, FDP_ACC.1(1) and FDP_ACC.1(2) indicate that the ST includes two iterations of the FDP_ACC.1 requirement. Alternately, iterations are identified with a prefix in the requirement labels representing the source Protection Profile or Module (e.g., **MDMPP40:**FCS_HTTPS_EXT.1 vs. **MDMA10:**FCS_HTTPS_EXT.1).
    - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
    - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
    - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.1 Security Target Reference

ST Title – MobileIron Platform 11 Security Target

ST Version – Version 0.6

ST Date – August 31, 2021

1.2 TOE Reference

TOE Identification – MobileIron Platform

- MobileIron Core, Version 11
- MobileIron Client – Mobile@Work for Android, Version 11

TOE Developer – MobileIron

Evaluation Sponsor – MobileIron

1.3 TOE Overview

The Target of Evaluation (TOE) is the MobileIron Core server 11 (deployed as a VM as identified later) and associated MobileIron Client 11 agents for Android devices (Mobile@Work for Android) that are part of their MobileIron Platform. Note that the MobileIron Platform product consists of other components (MobileIron Sentry and additional mobile device applications, e.g., Mobile@Work for iOS, Web@work, Docs@work, AppConnect container and Secure Application Manager) that do not play a role in enforcing the security functions included in this Security Target.

Also, while no iOS device agent security function claims are made within this Security Target, the MobileIron Core server supports the enrollment of Apple iPad and iPhone Mobile Devices with iOS 13 (https://www.niap-ccevs.org/Product/Compliant.cfm?PID=11036) and subsequent management of those devices via interaction with their own built in MDM agents.

1.4 TOE Description

The TOE is an MDM solution where the claimed security functions are implemented in a central MDM server – MobileIron Core - and distributed MDM agents – MobileIron Client.

MobileIron Core (http://www.mobileiron.com/en/products/core) integrates with backend enterprise IT systems and enables IT to define security and management policies for mobile apps, content and devices independent of the operating system. MobileIron Core enables mobile devices (including both Android and iOS mobile devices identified in section 1.4.1.1 below), application, and content management.

- Mobile device management capabilities are the primary focus of this evaluation and enable IT to securely manage mobile devices across mobile operating systems and provide secure corporate email, automatic device configuration, certificate-based security, and selective wiping of enterprise data from both corporate-owned as well as user-owned devices.
- Mobile application management capabilities are a secondary focus of this evaluation and help IT manage the entire application lifecycle, from making the applications available in the enterprise app storefront, facilitating deployment of applications to mobile devices, and retiring applications as necessary. Note that this capability is referred to as MAS – Mobile Application Store – Server later in this ST.
- Mobile content management functions are included in the MobileIron Platform, but no claims are made about those capabilities in this Security Target.

MobileIron Client– also known as Mobile@Work for Android – is an app downloaded by end users onto their mobile devices. It configures the device to function in an enterprise environment by enforcing the configuration and security policies set by the IT department. Once installed, it creates a secure MobileIron container to protect enterprise data and applications.

- The MobileIron Client works with MobileIron Core to configure corporate email, Wi-Fi, VPN, and security certificates and to create a clear separation between personal and business information. This allows IT to selectively wipe only the enterprise data on the device if the user leaves or if the device falls out of compliance or is lost.
- The MobileIron Client also enables additional enterprise device controls that are not subject to security claims and hence are outside the scope of the evaluation related to this Security Target.

Note that MobileIron distributes a Mobile@Work for iOS application, however, given restrictions on the associated Apple iOS mobile devices it is incapable of implementing the required MDM agent security functions. Rather, Mobile@Work for iOS is an optional component and serves only to direct the built-in iOS MDM agent to the MobileIron Core MDM server for enrollment. As such, this component does not implement any security functions. Mobile@Work for iOS is not required to enroll an iOS device with the MobileIron Core MDM server – the Safari browser built into iOS devices can be used to enroll with the MobileIron Core MDM server with no other application support.

1.4.1 TOE Architecture

MobileIron offers a MobileIron Platform solution comprising MobileIron Core, MobileIron Sentry, MobileIron Client, and mobile end user products (e.g., smartphones and tablets).

Of these components, the TOE is a central MobileIron Core server and MobileIron Clients installed on distributed end user mobile Android devices (Mobile@Work for Android). Mobile@Work for iOS and MobileIron Sentry are security products not within scope of this evaluation (i.e., the MDMPP40 requirements are not applicable), but can freely be used with the TOE in its evaluated configuration.

1.4.1.1 Physical Boundaries

As indicated above the TOE consists of two software components: MobileIron Core and MobileIron Client.

MobileIron Core is a server based on a CentOS 7.6 Linux operating system (OS) with Apache 2.4 (or later) that runs on an Intel x64 architecture server platform. MobileIron supports the MobileIron Core operating as virtual deployments in VMWare ESXi (6.5, 6.7 or 7.0).

MobileIron Core can optionally be configured to utilize an external LDAP server via a secure TLS channel to authenticate users.

MobileIron Client consists of apps deployed on Android mobile devices. These components are identified later in this ST as the "MDM Agent".

NIAP requires that MDM agents must be installed on NIAP-evaluated mobile devices in order to be evaluated using the MDMA10. At present there are a number of evaluated Samsung Galaxy mobile Android devices ranging from Android version 9 to 11 that can be used with the Android version of the MobileIron Client.

- (NIAP VID 11042, https://www.niap-ccevs.org/Product/Compliant.cfm?PID=11042) Samsung Galaxy Devices on Android 10: Samsung Galaxy S20
- (NIAP VID 11109, https://www.niap-ccevs.org/Product/Compliant.cfm?PID=11109) Samsung Galaxy Devices on Android 10: Samsung Galaxy A71
- (NIAP VID 11160, https://www.niap-ccevs.org/Product/Compliant.cfm?PID=11160) Samsung Galaxy Devices on Android 11

MobileIron Core can manage devices with the iOS MDM agent developed and evaluated by Apple Inc. – that agent has been evaluated on Apple iPad and iPhone Mobile Devices with iOS 13 (NIAP VID 11036).

1.4.1.2 Logical Boundaries

This section summarizes the security functions provided by MobileIron Platform:

- Security audit
- Cryptographic support
- Identification and authentication
- Security management
- Protection of the TSF
- TOE access
- Trusted path/channels

1.4.1.2.1 Security audit

The MDM Server can generate and store audit records for security-relevant events as they occur. These events are stored and protected by the MDM Server and can be reviewed by an authorized administrator. The MDM Server can be configured to export the audit records in either in CSV (comma separated values) format, text format, or a compressed archive format utilizing TLS for protection of the records on the network. The MDM Server also supports the ability to query information about MDM agents and export MDM configuration information.

The MDM Agent can generate audit records for security-relevant events and includes the ability to indicate (i.e., respond) when it has been enrolled and when policies are successfully applied to the MDM Agent. The MDM Server can be configured to alert an administrator based on its configuration. For example, it can be configured to alert the administrator when a policy update fails or an MDM Agent has been enrolled.

1.4.1.2.2 Cryptographic support

The MDM Server and MDM Agent both include and/or utilize cryptographic modules with certified algorithms for a wide range of cryptographic functions including: asymmetric key generation and establishment, encryption/decryption, cryptographic hashing and keyed-hash message authentication. These functions are supported with suitable random bit generation, initialization vector generation, secure key storage, and key and protected data destruction.

The primitive cryptographic functions are used to implement security communication protocols: TLS and HTTPS used for communication between the MDM Server and MDM Agent and between the MDM Server and remote administrators.

1.4.1.2.3 Identification and authentication

The MDM Server requires mobile device users (MD users) and administrators to be authenticated prior to allowing any security-related functions to be performed. This includes MD users enrolling their device in the MDM Server using a corresponding MDM Agent as well as an administrator logging on to manage the MDM Server configuration, MDM policies for mobile devices, etc.

In addition, both the MDM Server and MDM Agent utilize X.509 certificates, including certificate validation checking, in conjunction with TLS to secure communications between the MDM Server and MDM Agents as well as between the MDM Server and administrators using a web-based user interface for remote administrative access.

1.4.1.2.4 Security management

The MDM Server is designed to include at least two distinct user roles: administrator and mobile device user (MD user). The former interacts directly with the MDM Server while the latter is the user of a mobile device hosting an MDM Agent. The MDM Server further supports the fine-grain assignment of role (access to management function) to defined users allowing the definition of multiple user and administrator roles with different capabilities and responsibilities.

The MDM Server provides all the function necessary to manage its own security functions as well as to manage mobile device policies that are sent to MDM Agents. In addition, the MDM Server ensures that security management functions are limited to authorized administrators while allowing MD users to perform only necessary functions such as enrolling in the MDM Server.

The MDM Agents provide the functions necessary to securely communicate with and enroll in a MDM Server, implement policies received from an enrolled MDM Server, and report the results of applying policies.

1.4.1.2.5 Protection of the TSF

The MDM Server and MDM Agent work together to ensure that all security related communication between those components is protected from disclosure and modification.

Both the MDM Server and MDM Agent include self-testing capabilities to ensure that they are functioning properly. The MDM Server also has the ability to cryptographically verify during start-up that its executable image has not been corrupted.

The MDM Server also includes mechanisms (i.e., verification of the digital signature of each new image) so that the TOE itself can be updated while ensuring that the updates will not introduce malicious or other unexpected changes in the TOE.

1.4.1.2.6 TOE access

The MDM Server has the capability to display an advisory banner when users attempt to login in order to manage the TOE using the web-based and command-line based user interfaces.

1.4.1.2.7 Trusted path/channels

The MDM Server uses TLS/HTTPS to secure communication channels between itself and remote administrators accessing the TOE via a web-based user interface.

The MDM Server can optionally be configured to use TLS to communicate with an LDAP server for user authentication.

It also uses TLS to secure communication channels between itself and mobile device users (MD users). In this latter case, the protected communication channel is established between the MDM Server and applicable MDM Agent on the user's mobile device.

In addition, the MDM Server implements a restricted shell (CLISH) that is accessible via a console CLI to provide access to low level management functions.

1.4.2 TOE Documentation

MobileIron has developed an extensive document set for the MobileIron Platform. However, for the purpose of evaluation, the following guides were examined:

- Core and Android and iOS Client Mobile Device Management Protection Profile Guide for Release 11, August 2021
- On-Premise Installation Guide for MobileIron Core and Enterprise Connector 11.0.0.0, December 3, 2020
- Getting Started with MobileIron Core 11.0.0.0, December 3, 2020
- MobileIron Core 11.0.0.0 Device Management Guide for Android and Android enterprise Devices, December 3, 2020
- MobileIron Core 11.0.0.0 Device Management Guide for iOS and macOS Devices, December 3, 2020
- MobileIron Core 11.0.0.0 System Manager Guide, December 3, 2020
- MobileIron Core 11.0.0.0 Apps@Work Guide, November 19, 2020

2. Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, April 2017.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1, Revision 5, April 2017.
    - Part 3 Conformant
- Package Claims:
    - Protection Profile for Mobile Device Management, Version 4.0, 2019-04-25 (MDMPP40)
    - PP-Module for MDM Agents, Version 1.0, 2019-04-25 (MDMA10)
    - Functional Package for Transport Layer Security (TLS), Version 1.1, 1 March 2019 (PKGTLS11)

NIAP Technical Decisions		Package	Applied?
TD0552	SFR Rationale and Implicitly Satisfied SFRs	MDMPP40	Yes
TD0524	Updates to Certificate Revocation (FIA_X509_EXT.1)	MDMPP40	Yes
TD0479	FMT_SMF.1(1) Reliance on MDF Evals	MDMPP40	Yes
TD0462	MDM Distributed TOE: Registration Channel Updates	MDMPP40	Yes
TD0461	Security Audit for Distributed TOEs	MDMPP40	Yes
TD0462	MDM Distributed TOE: Registration Channel Updates	MDMPP40	Yes
TD0438	TST and TUD on the MDM Agent	MDMPP40	Yes
TD0497	SFR Rationale, Consistency of SPD, and Implicitly Satisfied SFRs	MDMA10	Yes
TD0491	Update to FMT_SMF_EXT.4 Test 2	MDMA10	Yes
TD0513	CA Certificate loading	PKGTLS11	Yes
TD0499	Testing with pinned certificates	PKGTLS11	Yes
TD0469	Modification of test activity for FCS_TLSS_EXT.1.1 test 4.1	PKGTLS11	Yes
TD0442	Updated TLS Ciphersuites for TLS Package	PKGTLS11	Yes

2.1 Conformance Rationale

The ST conforms to the MDMPP40/MDMA10/PKGTLS11. As explained previously, the security problem definition, security objectives, and security requirements have been drawn from the identified requirement documents.

3. Security Objectives

The Security Problem Definition may be found in the MDMPP40/MDMA10/PKGTLS11 and this section reproduces only the corresponding Security Objectives for operational environment for reader convenience. The MDMPP40/MDMA10/PKGTLS11 offers additional information about the identified security objectives, but that has not been reproduced here and the MDMPP40/MDMA10/PKGTLS11 should be consulted if there is interest in that material.

In general, the MDMPP40/MDMA10/PKGTLS11 has defined Security Objectives appropriate for Mobile Device Management (MDM) product and as such are applicable to the MobileIron Platform TOE.

3.1 Security Objectives for the Operational Environment

OE.COMPONENTS_RUNNING For distributed TOEs the administrator ensures that the availability of every TOE component is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. The administrator also ensures that it is checked as appropriate for every TOE component that the audit functionality is running properly.

OE.DATA_PROPER_ADMIN/OE.PROPER_ADMIN TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

OE.DATA_PROPER_USER/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.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.MOBILE_DEVICE_PLATFORM The MDM Agent relies upon the trustworthy mobile platform and hardware to provide policy enforcement as well as cryptographic services and data protection. The mobile platform provides trusted updates and software integrity verification of the MDM Agent.

OE.TIMESTAMP Reliable timestamp is provided by the operational environment for the TOE.

OE.WIRELESS_NETWORK A wireless network will be available to the mobile devices.

4. Extended Components Definition

All of the extended requirements in this ST have been drawn from the MDMPP40/MDMA10/PKGTLS11. The MDMPP40/MDMA10/PKGTLS11 defines the following extended requirements and since they are not redefined in this ST the MDMPP40/MDMA10/PKGTLS11 should be consulted for more information in regard to those CC extensions.

Extended SFRs:

- MDMPP40:FAU_ALT_EXT.1: Server Alerts
- MDMA10:FAU_ALT_EXT.2: Agent Alerts
- MDMPP40:FAU_CRP_EXT.1: Support for Compliance Reporting of Mobile Device Configuration
- MDMPP40:FAU_NET_EXT.1: Network Reachability Review
- MDMPP40:FAU_STG_EXT.1: External Trail Storage
- MDMPP40:FAU_STG_EXT.2: Audit Event Storage
- MDMPP40:FCS_CKM_EXT.4: Cryptographic Key Destruction
- MDMPP40:FCS_HTTPS_EXT.1: HTTPS Protocol
- MDMPP40:FCS_RBG_EXT.1: Extended: Random Bit Generation
- MDMPP40:FCS_STG_EXT.1: Cryptographic Key Storage
- MDMA10:FCS_STG_EXT.1(2): Cryptographic Key Storage
- PKGTLS11:FCS_TLS_EXT.1: TLS Protocol
- PKGTLS11:FCS_TLSC_EXT.1: TLS Client Protocol
- PKGTLS11:FCS_TLSC_EXT.2: TLS Client Support for Mutual Authentication
- PKGTLS11:FCS_TLSC_EXT.5: TLS Client Support for Supported Groups Extension
- PKGTLS11:FCS_TLSS_EXT.1: TLS Server Protocol
- PKGTLS11:FCS_TLSS_EXT.2: TLS Server Support for Mutual Authentication
- MDMPP40:FIA_ENR_EXT.1: Enrollment of Mobile Device into Management
- MDMA10:FIA_ENR_EXT.2: Agent Enrollment of Mobile Device into Management
- MDMPP40:FIA_X509_EXT.1(1): X.509 Certificate Validation
- MDMPP40:FIA_X509_EXT.2: X.509 Certificate Authentication
- MDMPP40:FIA_X509_EXT.5: X.509 Unique Certificate
- MDMPP40:FMT_POL_EXT.1: Trusted Policy Update
- MDMA10:FMT_POL_EXT.2: Agent Trusted Policy Update
- MDMPP40:FMT_SAE_EXT.1: Security Attribute Expiration
- MDMA10:FMT_SMF_EXT.4: Specification of Management Functions
- MDMA10:FMT_UNR_EXT.1: User Unenrollment Prevention
- MDMPP40:FPT_API_EXT.1: Use of Supported Services and APIs
- MDMPP40:FPT_LIB_EXT.1: Use of Third Party Libraries
- MDMPP40:FPT_TST_EXT.1: Functionality Testing
- MDMPP40:FPT_TUD_EXT.1: Trusted Update
- MDMPP40:FTP_ITC_EXT.1: Trusted Channel

5. Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) that serve to represent the security functional claims for the Target of Evaluation (TOE) and to scope the evaluation effort.

The SFRs have all been drawn from the MDMPP40/MDMA10/PKGTLS11. The refinements and operations already performed in the MDMPP40/MDMA10/PKGTLS11 are not identified (e.g., highlighted) here, rather the requirements have been copied from the MDMPP40/MDMA10/PKGTLS11 and any residual operations have been completed herein. Of particular note, the MDMPP40/MDMA10/PKGTLS11 made a number of refinements and completed some of the SFR operations defined in the Common Criteria (CC) and that PP should be consulted to identify those changes if necessary.

The SARs are also drawn from the MDMPP40/MDMA10/PKGTLS11 which includes all the SARs for EAL 1. However, the SARs are effectively refined since requirement-specific 'Assurance Activities' are defined in the MDMPP40/MDMA10/PKGTLS11 that serve to ensure corresponding evaluations will yield more practical and consistent assurance than the EAL 1 assurance requirements alone. The MDMPP40/MDMA10/PKGTLS11 should be consulted for the assurance activity definitions.

5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by MobileIron Platform TOE.

Requirement Class	Requirement Component
FAU: Security audit	MDMPP40:FAU_ALT_EXT.1: Server Alerts
	MDMA10:FAU_ALT_EXT.2: Agent Alerts
	MDMPP40:FAU_CRP_EXT.1: Support for Compliance Reporting of Mobile Device Configuration
	MDMPP40:FAU_GEN.1(1): Audit Data Generation
	MDMA10:FAU_GEN.1(2): Audit Data Generation
	MDMPP40:FAU_GEN.1(2): Audit Generation (MAS Server)
	MDMPP40:FAU_NET_EXT.1: Network Reachability Review
	MDMPP40:FAU_SAR.1: Audit Review
	MDMA10:FAU_SEL.1(2): Security Audit Event Selection
	MDMPP40:FAU_STG_EXT.1: External Trail Storage
	MDMPP40:FAU_STG_EXT.2: Audit Event Storage
FCS: Cryptographic support	MDMPP40:FCS_CKM.1: Cryptographic Key Generation
	MDMPP40:FCS_CKM.2: Cryptographic Key Establishment
	MDMPP40:FCS_CKM_EXT.4: Cryptographic Key Destruction
	MDMPP40:FCS_COP.1(1): Cryptographic Operation (Confidentiality Algorithms)
	MDMPP40:FCS_COP.1(2): Cryptographic Operation (Hashing Algorithms)
	MDMPP40:FCS_COP.1(3): Cryptographic Operation (Signature Algorithms)
	MDMPP40:FCS_COP.1(4): Cryptographic Operation (Keyed-Hash Message Authentication)
	MDMPP40:FCS_HTTPS_EXT.1: HTTPS Protocol
	MDMPP40:FCS_RBG_EXT.1: Extended: Random Bit Generation
	MDMPP40:FCS_STG_EXT.1: Cryptographic Key Storage
	MDMA10:FCS_STG_EXT.1(2): Cryptographic Key Storage
	PKGTLS11:FCS_TLS_EXT.1: TLS Protocol
	PKGTLS11:FCS_TLSC_EXT.1: TLS Client Protocol
	PKGTLS11:FCS_TLSC_EXT.2: TLS Client Support for Mutual Authentication
	PKGTLS11:FCS_TLSC_EXT.5: TLS Client Support for Supported Groups Extension
	PKGTLS11:FCS_TLSS_EXT.1: TLS Server Protocol
	PKGTLS11:FCS_TLSS_EXT.2: TLS Server Support for Mutual Authentication

Requirement Class	Requirement Component
FIA: Identification and authentication	MDMPP40:FIA_ENR_EXT.1: Enrollment of Mobile Device into Management
	MDMA10:FIA_ENR_EXT.2: Agent Enrollment of Mobile Device into Management
	MDMPP40:FIA_UAU.1: Timing of Authentication
	MDMPP40:FIA_X509_EXT.1(1): X.509 Certificate Validation
	MDMPP40:FIA_X509_EXT.2: X.509 Certificate Authentication
	MDMPP40:FIA_X509_EXT.5: X.509 Unique Certificate
FMT: Security management	MDMPP40:FMT_MOF.1(1): Management of Functions Behavior
	MDMPP40:FMT_MOF.1(2): Management of Functions Behavior (Enrollment)
	MDMPP40:FMT_MOF.1(3): Management of Functions in (MAS Server Downloads)
	MDMPP40:FMT_POL_EXT.1: Trusted Policy Update
	MDMA10:FMT_POL_EXT.2: Agent Trusted Policy Update
	MDMPP40:FMT_SAE_EXT.1: Security Attribute Expiration
	MDMPP40:FMT_SMF.1(1): Specification of Management Functions (Server configuration of Agent)
	MDMPP40:FMT_SMF.1(2): Specification of Management Functions (Server Configuration of Server)
	MDMPP40:FMT_SMF.1(3): Specification of Management Functions (MAS Server)
	MDMA10:FMT_SMF_EXT.4: Specification of Management Functions
	MDMPP40:FMT_SMR.1(1): Security Management Roles
	MDMPP40:FMT_SMR.1(2): Security Management Roles (MAS Server)
	MDMA10:FMT_UNR_EXT.1: User Unenrollment Prevention
FPT: Protection of the TSF	MDMPP40:FPT_API_EXT.1: Use of Supported Services and APIs
	MDMPP40:FPT_ITT.1(2): Internal TOE TSF Data Transfer (MDM Agent)
	MDMPP40:FPT_LIB_EXT.1: Use of Third Party Libraries
	MDMPP40:FPT_TST_EXT.1: Functionality Testing
	MDMPP40:FPT_TUD_EXT.1: Trusted Update
FTA: TOE access	MDMPP40:FTA_TAB.1: Default TOE Access Banners
FTP: Trusted path/channels	MDMPP40:FTP_ITC.1(1): Inter-TSF Trusted Channel (Authorized IT Entities)
	MDMPP40:FTP_ITC.1(2): Inter-TSF Trusted Channel (MDM Agent)
	MDMPP40:FTP_ITC_EXT.1: Trusted Channel
	MDMPP40:FTP_TRP.1(1): Trusted Path (for Remote Administration)
	MDMPP40:FTP_TRP.1(2): Trusted Path (for Enrollment)

Table 1 TOE Security Functional Components

5.1.1 Security audit (FAU)

5.1.1.1 Server Alerts (MDMPP40:FAU_ALT_EXT.1)

MDMPP40:FAU_ALT_EXT.1.1

The TSF 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. [***no other events***].

5.1.1.2 Agent Alerts (MDMA10:FAU_ALT_EXT.2)

MDMA10:FAU_ALT_EXT.2.1

The MDM Agent shall provide an alert via the trusted channel to the MDM Server in the event of any of the following audit events: successful application of policies to a mobile device, [***receiving***] periodic reachability events, [***change in enrollment state***].

MDMA10:FAU_ALT_EXT.2.2

The MDM Agent shall queue alerts if the trusted channel is not available.

5.1.1.3 Support for Compliance Reporting of Mobile Device Configuration (MDMPP40:FAU_CRP_EXT.1)

MDMPP40:FAU_CRP_EXT.1.1

The TSF shall provide [***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 a channel that meets the secure channel requirements in FTP_ITC.1(1). 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. [***no other information***].

5.1.1.4 Audit Data Generation (MDMPP40:FAU_GEN.1(1))

MDMPP40:FAU_GEN.1.1(1)

Refinement: The TSF shall [***implement functionality***] to generate an audit record of the following auditable events:

a. Start up and shut down of the MDM System
b. All administrative actions
c. [***Commands issued to the MDM Agent***]
d. Specifically defined auditable events listed in **Table 2 Server Auditable Events**
e. [***no other events***].

MDMPP40:FAU_GEN.1.2(1)

The TSF 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 2 Server Auditable Events**; [**no other audit relevant information**].

Note: The Strikethrough text in the following table indicate events for requirements that are not claimed.

Requirement	Auditable Events	Additional Content
MDMPP40:FAU_ALT_EXT.1	Type of alert.	Identity of Mobile Device that sent alert.
MDMPP40:FAU_CRP_EXT.1	None.	None
MDMPP40:FAU_GEN.1(1)	None.	None
MDMPP40:FAU_GEN.1(2)	None.	None
MDMPP40:FAU_NET_EXT.1	None.	None
MDMPP40:FAU_SAR.1	None.	None
~~**MDMPP40:FAU_SEL.1**~~	~~All modifications to the audit configuration that occur while the audit collection functions are operating.~~	~~No additional information.~~
MDMPP40:FAU_STG_EXT.1	None.	None
MDMPP40:FAU_STG_EXT.2	None.	None
~~**MDMPP40:FCO_CPC_EXT.1**~~	~~Enabling or disabling communications between a pair of components.~~	~~Identities of the endpoints pairs enabled or disabled.~~
MDMPP40:FCS_CKM.1	[selection: Failure of key generation activity for authentication keys, None]	No additional information.
MDMPP40:FCS_CKM.2	None.	None
MDMPP40:FCS_CKM_EXT.4	None.	None
MDMPP40:FCS_COP.1(1)	None.	None
MDMPP40:FCS_COP.1(2)	None.	None
MDMPP40:FCS_COP.1(3)	None.	None
MDMPP40:FCS_COP.1(4)	None.	None

Requirement	Auditable Events	Additional Content
~~**PKGTLS11:FCS_DTLSC_EXT.1**~~	~~Failure of the certificate validity check.~~	~~Issuer Name and Subject Name of certificate~~
~~**PKGTLS11:FCS_DTLSC_EXT.2**~~	~~None.~~	~~None~~
~~**PKGTLS11:FCS_DTLSS_EXT.1**~~	~~Failure of the certificate validity check.~~	~~Issuer Name and Subject Name of certificate~~
~~**PKGTLS11:FCS_DTLSS_EXT.2**~~	~~None.~~	~~None~~
MDMPP40:FCS_HTTPS_EXT.1	Failure of the certificate validity check.	Issuer Name and Subject Name of certificate. [selection: User's authorization decision, no additional information]
~~**MDMPP40:FCS_IV_EXT.1**~~	~~None.~~	~~None~~
MDMPP40:FCS_RBG_EXT.1	Failure of the randomization process.	No additional information.
MDMPP40:FCS_STG_EXT.1	None.	None
~~**MDMPP40:FCS_STG_EXT.2**~~	~~None.~~	~~None~~
PKGTLS11:FCS_TLS_EXT.1	None.	None
PKGTLS11:FCS_TLSC_EXT.1	(server) Failure to establish a TLS session. (server) Failure to verify presented identifier. (agent) Failure to establish a TLS session. (agent) Failure to verify presented identifier. (agent) Establishment/termination of a TLS session.	(server) Reason for failure. (server) Presented identifier and reference identifier. (agent) Reason for failure. (agent) Presented identifier and reference identifier. (agent) Non-TOE endpoint of connection.
PKGTLS11:FCS_TLSC_EXT.2	None.	None
~~**PKGTLS11:FCS_TLSC_EXT.3**~~	~~None.~~	~~None~~
~~**PKGTLS11:FCS_TLSC_EXT.4**~~	~~None.~~	~~None~~
PKGTLS11:FCS_TLSC_EXT.5	None.	None
PKGTLS11:FCS_TLSS_EXT.1	Failure to establish a TLS session.	Reason for failure.
PKGTLS11:FCS_TLSS_EXT.2	None.	None.
~~**PKGTLS11:FCS_TLSS_EXT.3**~~	~~None~~	~~None~~
~~**PKGTLS11:FCS_TLSS_EXT.4**~~	~~None.~~	~~None~~
MDMPP40:FIA_ENR_EXT.1	Failure of MD user authentication.	Presented username.
MDMPP40:FIA_UAU.1	None.	None
~~**MDMPP40:FIA_UAU_EXT.4(1)**~~	~~Attempt to reuse enrollment data.~~	~~Enrollment data.~~
~~**MDMPP40:FIA_UAU_EXT.4(2)**~~	~~Attempt to reuse enrollment data.~~	~~Enrollment data.~~
MDMPP40:FIA_X509_EXT.1(1)	Failure to validate X.509 certificate	Reason for failure.
~~**MDMPP40:FIA_X509_EXT.1(2)**~~	~~Failure to validate X.509 certificate~~	~~Reason for failure.~~
MDMPP40:FIA_X509_EXT.2	Failure to establish connection to determine revocation status.	No additional information.
~~**MDMPP40:FIA_X509_EXT.3**~~	~~Generation of Certificate Request Message. Success or failure of verification.~~	~~Content of Certificate Request Message. Issuer and Subject name of added certificate or reason for failure.~~
~~**MDMPP40:FIA_X509_EXT.4**~~	~~Generation of Certificate Enrollment Request. Success or failure of enrollment. Update of EST Trust Anchor Database.~~	~~Issuer and Subject name of EST Server. Method of authentication. Issuer and Subject name of certificate used to authenticate. Content of Certificate Request Message. Issuer and Subject name of added certificate or reason for failure. Subject name of added Root CA.~~
MDMPP40:FIA_X509_EXT.5	None.	None

Requirement	Auditable Events	Additional Content
MDMPP40:FMT_MOF.1(1)	Issuance of command to perform function. Change of policy settings.	Command sent and identity of MDM Agent recipient(s). Policy changed and value or full policy.
MDMPP40:FMT_MOF.1(2)	Enrollment by a user.	Identity of user.
MDMPP40:FMT_MOF.1(3)	None.	None
MDMPP40:FMT_POL_EXT.1	None.	None
MDMPP40:FMT_SAE_EXT.1	Enrollment attempted after expiration of authentication data.	Identity of user.
MDMPP40:FMT_SMF.1(1)	None.	None
MDMPP40:FMT_SMF.1(2)	Success or failure of function.	No additional information.
MDMPP40:FMT_SMF.1(3)	None.	None
MDMPP40:FMT_SMR.1(1)	None.	None
MDMPP40:FMT_SMR.1(2)	None.	None
MDMPP40:FPT_API_EXT.1	None.	None
~~**MDMPP40:FPT_ITT.1(1)**~~	~~Initiation and termination of the trusted channel.~~	~~Trusted channel protocol. Identity of initiator and recipient.~~
MDMPP40:FPT_ITT.1(2)	Initiation and termination of the trusted channel.	Trusted channel protocol. Identity of initiator and recipient.
MDMPP40:FPT_LIB_EXT.1	None.	None
MDMPP40:FPT_TST_EXT.1	Initiation of self-test. Failure of self-test. Detected integrity violation.	Algorithm that caused failure. The TSF code file that caused the integrity violation.
MDMPP40:FPT_TUD_EXT.1	Success or failure of signature verification.	No additional information.
MDMPP40:FTA_TAB.1	Change in banner setting.	No additional information.
MDMPP40:FTP_ITC.1(1)	Initiation and termination of the trusted channel.	Trusted channel protocol. Non-TOE endpoint of connection
MDMPP40:FTP_ITC.1(2)	Initiation and termination of the trusted channel.	Trusted channel protocol. Non-TOE endpoint of connection.
MDMPP40:FTP_ITC_EXT.1	None.	None
MDMPP40:FTP_TRP.1(1)	Initiation and termination of the trusted channel.	Trusted channel protocol. Identity of administrator.
MDMPP40:FTP_TRP.1(2)	Initiation and termination of the trusted channel.	Trusted channel protocol.
~~**MDMPP40:FTP_TRP.1(3)**~~	~~Initiation and termination of the trusted channel.~~	~~Trusted channel protocol.~~

Table 2 Server Auditable Events

5.1.1.5 Audit Data Generation (MDMA10:FAU_GEN.1(2))

MDMA10:FAU_GEN.1.1(2)

Refinement: The MDM Agent shall [***implement functionality***] to generate an MDM Agent audit record of the following auditable events: a. Startup and shutdown of the MDM Agent; b. All auditable events for not specified level of audit; and c. MDM policy updated, any modification commanded by the MDM Server, specifically defined auditable events listed in **Table 3 Client Auditable Events**, and [***no other events***].

MDMA10:FAU_GEN.1.2(2)

Refinement: The [***TSF***] shall record within each MDM Agent audit record at least the following information: a. Date and time of the event, type of event, subject identity, (if relevant) the outcome

(success or failure) of the event, and additional information in **Table 3 Client Auditable Events**; and b. For each audit event type, based on the auditable event definitions of the functional components included in the PP-Module/ST, [**no other audit relevant information**].

Requirement	Auditable Events	Additional Content
MDMA10:FAU_ALT_EXT.2	Success/failure of sending alert.	No additional information.
MDMPP40:FAU_GEN.1(1)	None.	
MDMA10:FAU_GEN.1(2)	None.	
MDMA10:FAU_SEL.1(2)	All modifications to the audit configuration that occur while the audit collection functions are operating.	No additional information.
MDMPP40:FAU_STG_EXT.1	None.	
MDMA10:FCS_STG_EXT.1(2)	None.	
MDMPP40:FAU_STG_EXT.2	None.	
MDMPP40:FCS_CKM.1	[***None***]	No additional information.
MDMPP40:FCS_CKM.2	None.	
MDMPP40:FCS_CKM_EXT.4	None.	
MDMPP40:FCS_COP.1(1)	None.	
MDMPP40:FCS_COP.1(2)	None.	
MDMPP40:FCS_COP.1(3)	None.	
MDMPP40:FCS_COP.1(4)	None.	
~~**MDMPP40:FCS_IV_EXT.1**~~	~~None.~~	
MDMPP40:FCS_STG_EXT.1	None.	
MDMA10:FIA_ENR_EXT.2	Enrollment in management.	Reference identifier of MDM Server.
MDMA10:FMT_POL_EXT.2	Failure of policy validation.	Reason for failure of validation.
MDMA10:FMT_SMF_EXT.4	Outcome (Success/failure) of function.	No additional information.
MDMA10:FMT_UNR_EXT.1	[***none***]	No additional information.
MDMPP40:FPT_API_EXT.1	None.	
MDMPP40:FPT_LIB_EXT.1	None.	
MDMPP40:FPT_ITT.1(2)	Initiation and termination of the trusted channel.	Trusted channel protocol. Identity of initiator and recipient.

Table 3 Client Auditable Events

5.1.1.6 Audit Generation (MAS Server) (MDMPP40:FAU_GEN.1(2))

MDMPP40: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.

MDMPP40:FAU_GEN.1.2(2)

Refinement: The [***MAS Server***] shall record within each TSF audit record at least the following information: date and time of the event, type of event, mobile device identity, [**no other audit relevant information**].

5.1.1.7 Network Reachability Review (MDMPP40:FAU_NET_EXT.1)

MDMPP40:FAU_NET_EXT.1.1

The TSF shall provide authorized administrators with the capability to read the network connectivity status of an enrolled agent.

5.1.1.8 Audit Review (MDMPP40:FAU_SAR.1)

MDMPP40:FAU_SAR.1.1

Refinement: The TSF shall [***implement functionality***] to provide Authorized Administrators with the capability to read all audit data from the audit records.

MDMPP40:FAU_SAR.1.2

Refinement: The TSF shall [***implement functionality***] to provide the audit records in a manner suitable for the Authorized Administrators to interpret the information.

5.1.1.9 Security Audit Event Selection (MDMA10:FAU_SEL.1(2))

MDMA10:FAU_SEL.1.1(2)

Refinement: The TSF shall [***invoke platform-provided functionality***] 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, failure of auditable security events, [**no other attributes**].

5.1.1.10 External Trail Storage (MDMPP40:FAU_STG_EXT.1)

MDMPP40:FAU_STG_EXT.1.1

The TSF shall be able to use a trusted channel per FTP_ITC.1(1) to transmit audit data to an external IT entity and [***store audit data locally***].

5.1.1.11 Audit Event Storage (MDMPP40:FAU_STG_EXT.2)

MDMPP40:FAU_STG_EXT.2.1

The TSF shall [***implement functionality***] to protect the stored audit records in the audit trail from unauthorized modification.

5.1.2 Cryptographic support (FCS)

5.1.2.1 Cryptographic Key Generation (MDMPP40:FCS_CKM.1)

MDMPP40:FCS_CKM.1.1

Refinement: The TSF shall [***implement functionality***] to generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm [

- ***ECC schemes using 'NIST curves' P-384 and [P-256, P-521, no other curves] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.4***].

5.1.2.2 Cryptographic Key Establishment (MDMPP40:FCS_CKM.2)

MDMPP40:FCS_CKM.2.1

Refinement: The TSF shall [***implement functionality***] to perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [

- ***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'***].

5.1.2.3 Cryptographic Key Destruction (MDMPP40:FCS_CKM_EXT.4)

MDMPP40:FCS_CKM_EXT.4.1

The TSF shall destroy plaintext keying material and critical security parameters by [***implementing key destruction in accordance with the following rules:***

- ***For volatile memory, the destruction shall be executed by a single direct overwrite [consisting of zeroes] ,***
- ***For non-volatile EEPROM, the destruction shall be executed by a single direct overwrite consisting of a pseudo-random pattern using the TSF/Platform RBG (as specified in FCS_RBG_EXT.1), followed by a read-verify,***

- ***For non-volatile flash memory, that is not wear-leveled, the destruction shall be executed [by a single direct overwrite consisting of zeros followed by a read-verify],***
- ***For non-volatile flash memory, that is wear-leveled, the destruction shall be executed [by a single direct overwrite consisting of zeros] ,***
- ***For non-volatile memory other than EEPROM and flash, the destruction shall be executed by a single direct overwrite with a random pattern that is changed before each write***].

MDMPP40:FCS_CKM_EXT.4.2

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

5.1.2.4 Cryptographic Operation (Confidentiality Algorithms) (MDMPP40:FCS_COP.1(1))

MDMPP40:FCS_COP.1.1(1)

Refinement: The TSF shall [***implement functionality***] to perform encryption/decryption in accordance with a specified cryptographic algorithm: [

- ***AES-CBC (as defined in FIPS PUB 197 and NIST SP 800-38A) mode,***
- ***AES-GCM (as defined in NIST SP 800-38D)***]

and cryptographic key sizes [***128-bit, 256-bit***].

5.1.2.5 Cryptographic Operation (Hashing Algorithms) (MDMPP40:FCS_COP.1(2))

MDMPP40:FCS_COP.1.1(2)

Refinement: The TSF shall [***implement functionality***] to perform cryptographic hashing in accordance with a specified cryptographic algorithm [***SHA-256, SHA-384, SHA-512***] and message digest sizes [***256, 384, 512***] bits that meet the following: FIPS Pub 180-4.

5.1.2.6 Cryptographic Operation (Signature Algorithms) (MDMPP40:FCS_COP.1(3))

MDMPP40:FCS_COP.1.1(3)

Refinement: The TSF shall [***implement functionality***] to perform cryptographic signature services (generation and verification) in accordance with a specified cryptographic algorithm [

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

5.1.2.7 Cryptographic Operation (Keyed-Hash Message Authentication) (MDMPP40:FCS_COP.1(4))

MDMPP40:FCS_COP.1.1(4)

Refinement: The TSF shall [***implement functionality***] to perform keyed-hash message authentication in accordance with a specified cryptographic algorithm HMAC- [***SHA-256, SHA-384, SHA-512***] , key sizes [**equal to hash size**] , and message digest sizes [***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.'

5.1.2.8 HTTPS Protocol (MDMPP40:FCS_HTTPS_EXT.1)

MDMPP40:FCS_HTTPS_EXT.1.1

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

MDMPP40:FCS_HTTPS_EXT.1.2

The TSF shall implement HTTPS using TLS in accordance with the Package for Transport Layer Security.

5.1.2.9 Extended: Random Bit Generation (MDMPP40:FCS_RBG_EXT.1)

MDMPP40:FCS_RBG_EXT.1.1

The TSF shall [***implement functionality***] to perform all deterministic random bit generation services in accordance with NIST Special Publication 800-90A using [***CTR_DRBG (AES)***].

MDMPP40:FCS_RBG_EXT.1.2

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

5.1.2.10 Cryptographic Key Storage (MDMPP40:FCS_STG_EXT.1)

MDMPP40:FCS_STG_EXT.1.1

The TSF shall utilize [***platform-provided key storage***] for all persistent secrets and private keys.

5.1.2.11 Cryptographic Key Storage (MDMA10:FCS_STG_EXT.1(2))

MDMA10:FCS_STG_EXT.1.1(2)

Refinement: The MDM Agent shall use the platform-provided key storage for all persistent secret and private keys.

5.1.2.12 TLS Protocol (PKGTLS11:FCS_TLS_EXT.1)

PKGTLS11:FCS_TLS_EXT.1.1

The product shall implement [***TLS as a client, TLS as a server***].

5.1.2.13 TLS Client Protocol (PKGTLS11:FCS_TLSC_EXT.1)

PKGTLS11:FCS_TLSC_EXT.1.1

The product shall implement TLS 1.2 (RFC 5246) and [***no earlier TLS versions***] as a client that supports the cipher suites [
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289] and also supports functionality for [***mutual authentication***] (TD0442 applied)

PKGTLS11:FCS_TLSC_EXT.1.2

The product shall verify that the presented identifier matches the reference identifier according to RFC 6125.

PKGTLS11:FCS_TLSC_EXT.1.3

The product shall not establish a trusted channel if the server certificate is invalid [***with no exceptions***].

5.1.2.14 TLS Client Support for Mutual Authentication (PKGTLS11:FCS_TLSC_EXT.2)

PKGTLS11:FCS_TLSC_EXT.2.1

The product shall support mutual authentication using X.509v3 certificates.

5.1.2.15 TLS Client Support for Supported Groups Extension (PKGTLS11:FCS_TLSC_EXT.5)

PKGTLS11:FCS_TLSC_EXT.5.1

The product shall present the Supported Groups Extension in the Client Hello with the supported groups [***secp256r1, secp384r1, secp521r1***].

5.1.2.16 TLS Server Protocol (PKGTLS11:FCS_TLSS_EXT.1)

PKGTLS11:FCS_TLSS_EXT.1.1
The product shall implement TLS 1.2 (RFC 5246) and [***no earlier TLS versions***] as a server that supports the cipher suites [
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289] and no other cipher suites, and also supports functionality for [***mutual authentication, session resumption based on session tickets according to RFC 5077***] (TD0442 applied)

PKGTLS11:FCS_TLSS_EXT.1.2
The product shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0 and [***TLS 1.1***]

PKGTLS11:FCS_TLSS_EXT.1.3
The product shall perform key establishment for TLS using [***ECDHE parameters using elliptic curves [secp256r1, secp384r1, secp521r1] and no other curves***].

5.1.2.17 TLS Server Support for Mutual Authentication (PKGTLS11:FCS_TLSS_EXT.2)

PKGTLS11:FCS_TLSS_EXT.2.1
The product shall support authentication of TLS clients using X.509v3 certificates.

PKGTLS11:FCS_TLSS_EXT.2.2
The product shall not establish a trusted channel if the client certificate is invalid.

PKGTLS11:FCS_TLSS_EXT.2.3
The product shall not establish a trusted channel if the Distinguished Name (DN) or Subject Alternative Name (SAN) contained in a certificate does not match one of the expected identifiers for the client.

5.1.3 Identification and authentication (FIA)

5.1.3.1 Enrollment of Mobile Device into Management (MDMPP40:FIA_ENR_EXT.1)

MDMPP40:FIA_ENR_EXT.1.1
The TSF shall authenticate the remote users over a trusted channel during the enrollment of a mobile device.

MDMPP40:FIA_ENR_EXT.1.2
The TSF shall limit the user's enrollment of devices to devices specified by [***[serial number]***] and [***a number of devices***].

5.1.3.2 Agent Enrollment of Mobile Device into Management (MDMA10:FIA_ENR_EXT.2)

MDMA10:FIA_ENR_EXT.2.1
The MDM Agent shall record the reference identifier of the MDM Server during the enrollment process.

5.1.3.3 Timing of Authentication (MDMPP40:FIA_UAU.1)

MDMPP40:FIA_UAU.1.1
Refinement: The TSF shall [***implement functionality***] to allow [**no list of TSF mediated actions**] on behalf of the user to be performed before the user is authenticated with the Server.

MDMPP40:FIA_UAU.1.2
Refinement: The TSF shall [***implement functionality***] that requires each user to be successfully authenticated with the Server before allowing any other TSF-mediated actions on behalf of that user.

5.1.3.4 X.509 Certificate Validation (MDMPP40:FIA_X509_EXT.1(1))

MDMPP40:FIA_X509_EXT.1.1(1)

The TSF shall [***implement functionality***] to 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, that the CA flag is set to TRUE for all CA certificates, and that any path constraints are met.
- The TSF shall validate that any CA certificate includes caSigning purpose in the key usage field.
- The TSF shall validate the revocation status of the certificate using [***a CRL as specified in RFC 5759 Section 5***].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
    - o 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.
    - o 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.
    - o 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 EKU field.
    - o CSP 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 EKU field.
    - o 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 EKU field.

(TD0524 applied, supersedes TD0467)

MDMPP40:FIA_X509_EXT.1.2(1)

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

5.1.3.5 X.509 Certificate Authentication (MDMPP40:FIA_X509_EXT.2)

MDMPP40:FIA_X509_EXT.2.1

The TSF shall [***implement functionality to use X.509v3 certificates as defined by RFC 5280 to support authentication for [TLS as defined in the Package for Transport Layer Security] , and [no additional uses]***].

MDMPP40:FIA_X509_EXT.2.2

When the [***TSF***] cannot establish a connection to determine the validity of a certificate, the TSF shall [***implement functionality***] to [***not accept the certificate***].

5.1.3.6 X.509 Unique Certificate (MDMPP40:FIA_X509_EXT.5)

MDMPP40:FIA_X509_EXT.5.1

The TSF shall [***implement functionality***] to require a unique certificate for each client device.

5.1.4 Security management (FMT)

5.1.4.1 Management of Functions Behavior (MDMPP40:FMT_MOF.1(1))

MDMPP40:FMT_MOF.1.1(1)

Refinement: The TSF shall restrict the ability to perform the functions
- listed in FMT_SMF.1(1),
- enable, disable, and modify policies listed inFMT_SMF.1(1),
- listed in FMT_SMF.1(2),
- [***no other functions***]

to authorized administrators.

5.1.4.2 Management of Functions Behavior (Enrollment) (MDMPP40:FMT_MOF.1(2))

MDMPP40:FMT_MOF.1.1(2)

Refinement: The MDM Server shall restrict the ability to initiate the enrollment process to authorized administrators and MD users.

5.1.4.3 Management of Functions in (MAS Server Downloads) (MDMPP40:FMT_MOF.1(3))

MDMPP40:FMT_MOF.1.1(3)

Refinement: The MAS Server shall restrict the ability to download applications, allowing only enrolled mobile devices that are compliant with MDM policies and assigned to a user in the application access group to perform this function.

5.1.4.4 Trusted Policy Update (MDMPP40:FMT_POL_EXT.1)

MDMPP40:FMT_POL_EXT.1.1

The TSF shall provide digitally signed policies and policy updates to the MDM Agent.

5.1.4.5 Agent Trusted Policy Update (MDMA10:FMT_POL_EXT.2)

MDMA10:FMT_POL_EXT.2.1

The MDM Agent shall only accept policies and policy updates that are digitally signed by a certificate that has been authorized for policy updates by the MDM Server.

MDMA10:FMT_POL_EXT.2.2

The MDM Agent shall not install policies if the policy-signing certificate is deemed invalid.

5.1.4.6 Security Attribute Expiration (MDMPP40:FMT_SAE_EXT.1)

MDMPP40:FMT_SAE_EXT.1.1

The TSF shall be capable to specify a configurable expiration time for enrollment authentication data.

MDMPP40:FMT_SAE_EXT.1.2

The TSF shall be able to deny enrollment after the expiration time for the enrollment authentication data has passed.

5.1.4.7 Specification of Management Functions (Server configuration of Agent) (MDMPP40:FMT_SMF.1(1))

MDMPP40: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 6)
2. full wipe of protected data (MDF Function 7)
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 11)
10. install applications (MDF Function 16)
11. update system software (MDF Function 15)
12. remove applications (MDF Function 14)

and the following commands to the MDM Agent:

[***13. remove Enterprise applications (MDF Function 17),***
14. wipe Enterprise data (MDF Function 28),
15. remove imported X.509v3 certificates and [no other X.509v3 certificates] in the Trust Anchor Database (MDF Function 12),
17. import keys/secrets into the secure key storage (MDF Function 9),

18. destroy imported keys/secrets and [no other keys/secrets] in the secure key storage (MDF Function 10),
19. read audit logs kept by the MD (MDF Function 32) – Samsung Only]
and the following MD configuration policies:
25. password policy:
    a. minimum password length
    b. minimum password complexity
    c. maximum password lifetime (MDF Function 1)
26. session locking policy:
    a. screen-lock enabled/disabled
    b. screen lock timeout
    c. number of authentication failures (MDF Function 2)
27. wireless networks (SSIDs) to which the MD may connect (MDF Function 2)
28. security policy for each wireless network:
    a. [***specify the CA(s) from which the MD will accept 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. [**no any additional WLAN management functions**] (WLAN Client Function 1)
29. application installation policy by [
    - specifying authorized application repository(s) – Samsung Only
    - denying application installation – iOS Only] (MDF Function 8)
30. enable/disable policy for [**camera and microphone – *Camera-only on iOS***] across device and [***no other method***] (MDF Function 5),

and the following MD configuration policies:
[***31. enable/disable policy for the VPN protection across MD and [no other method] (MDF Function 3) – iOS only,***
32. enable/disable policy for [assignment: list of radios] (MDF Function 4) – Samsung Only,
34. enable/disable policy for [assignment: list of protocols where the device acts as a server] (MDF Function 25) – Samsung Only,
35. enable/disable policy for developer modes (MDF Function 26), – Samsung Only
36. enable policy for data-at-rest protection (MDF Function 20) – Samsung Only,
37. enable policy for removable media's data-at-rest protection (MDF Function 21) – Samsung Only,
40. enable/disable policy for display notification in the locked state of [all notifications[1]] (MDF Function 19) - iOS only,
47. the unlock banner policy (MDF Function 36),
48. configure the auditable items (MDF Function 37) – Samsung Only,
49. enable/disable [USB mass storage mode] (MDF Function 39) – Samsung Only,
51. enable/disable [Hotspot functionality authenticated by [pre-shared key], USB tethering authenticated by [no authentication]] (MDF Function 41)) – Samsung Only,
52. enable/disable location services: [no other method] (MDF Function 22) – Samsung Only,
55. enable/disable policy for use of Biometric Authentication Factor (MDF Function 23)].

[1] The iOS Security Target claims "all notifications", so while the MDMPP40 indicates the choices are *a. email notifications, b. calendar appointments, c. contact associated with phone call notification, d. text message notification, e. other application-based notifications, f. none*, this Security Target is effectively refining the requirement to indicate "all notifications" to match the claim of the applicable mobile device.

5.1.4.8 Specification of Management Functions (Server Configuration of Server) (MDMPP40:FMT_SMF.1(2))

MDMPP40:FMT_SMF.1.1(2)

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

a. choose X.509v3 certificates for MDM Server use

b. configure the [***devices specified by [[serial number]], a number of devices***] and [***[configure server session lock timeout]***] allowed for enrollment

c. [

2. configure the TOE unlock banner,

3. configure periodicity of the following commands to the agent: [

1. query connectivity status;

2. query the current version of the MD firmware/software;

3. query the current version of the hardware model of the device;

4. query the current version of installed mobile applications]].

5.1.4.9 Specification of Management Functions (MAS Server) (MDMPP40:FMT_SMF.1(3))

MDMPP40: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. [***no other functions***].

5.1.4.10 Specification of Management Functions (MDMA10:FMT_SMF_EXT.4)

MDMA10:FMT_SMF_EXT.4.1

The MDM Agent shall be capable of interacting with the platform to perform the following functions:

Import the certificates to be used for authentication of MDM Agent communications;

[***administrator-provided device management functions in MDM PP***];

[***no additional functions***].

MDMA10:FMT_SMF_EXT.4.2

The MDM Agent shall be capable of performing the following functions: Enroll in management; Configure whether users can unenroll from management; [***configure periodicity of reachability events***].

5.1.4.11 Security Management Roles (MDMPP40:FMT_SMR.1(1))

MDMPP40:FMT_SMR.1.1(1)

Refinement: The TSF shall maintain the roles administrator, MD user, and [***[Server primary administrator, Security configuration administrator, Device user group administrator, Auditor]***].

MDMPP40:FMT_SMR.1.2(1)

The TSF shall be able to associate users with roles.

5.1.4.12 Security Management Roles (MAS Server) (MDMPP40:FMT_SMR.1(2))

MDMPP40:FMT_SMR.1.1(2)

Refinement: The TSF shall additionally maintain the roles enrolled mobile devices, application access groups, and [**no additional authorized identified roles**].

MDMPP40:FMT_SMR.1.2(2)

Refinement: The MAS Server shall be able to associate users with roles.

5.1.4.13 User Unenrollment Prevention (MDMA10:FMT_UNR_EXT.1)

MDMA10:FMT_UNR_EXT.1.1

The MDM Agent shall provide a mechanism to enforce the following behavior upon an attempt to unenroll the mobile device from management: [***prevent the unenrollment from occurring***].

5.1.5 Protection of the TSF (FPT)

5.1.5.1 Use of Supported Services and APIs (MDMPP40:FPT_API_EXT.1)

MDMPP40:FPT_API_EXT.1.1
The TSF shall use only documented platform API's.

5.1.5.2 Internal TOE TSF Data Transfer (MDM Agent) (MDMPP40:FPT_ITT.1(2))

MDMPP40:FPT_ITT.1.1(2)
Refinement: The TSF shall [***implement functionality using [mutually authenticated TLS as defined in the Package for Transport Layer Security]***] to protect all data from disclosure and modification when it is transferred between the TSF and MDM Agent.

5.1.5.3 Use of Third Party Libraries (MDMPP40:FPT_LIB_EXT.1)

MDMPP40:FPT_LIB_EXT.1.1
The MDM software shall be packaged with only [**Bouncy Castle and OpenSSL**].

5.1.5.4 Functionality Testing (MDMPP40:FPT_TST_EXT.1)

MDMPP40:FPT_TST_EXT.1.1
The TSF shall run a suite of self -tests during initial start-up (power on) to demonstrate correct operation of the TSF.

MDMPP40:FPT_TST_EXT.1.2
The TSF shall [***implement functionality***] to provide the capability to verify the integrity of stored TSF executable code when it is loaded for execution through the use of the [***TOE platform***] -provided cryptographic services.

5.1.5.5 Trusted Update (MDMPP40:FPT_TUD_EXT.1)

MDMPP40:FPT_TUD_EXT.1.1
The TSF shall provide Authorized Administrators the ability to query the current version of the software. (TD0438 applied)

MDMPP40:FPT_TUD_EXT.1.2
The TSF shall [***implement functionality***] to provide Authorized Administrators the ability to initiate updates to TSF software.

MDMPP40:FPT_TUD_EXT.1.3
The TSF shall [***implement functionality***] to provide a means to verify software updates to the TSF using a digital signature mechanism prior to installing those updates.

5.1.6 TOE access (FTA)

5.1.6.1 Default TOE Access Banners (MDMPP40:FTA_TAB.1)

MDMPP40:FTA_TAB.1.1
Refinement: Before establishing a user session, the TSF shall [***implement functionality***] to display an Administrator-specified advisory notice and consent warning message regarding use of the TOE.

5.1.7 Trusted path/channels (FTP)

5.1.7.1 Inter-TSF Trusted Channel (Authorized IT Entities) (MDMPP40:FTP_ITC.1(1))

MDMPP40:FTP_ITC.1.1(1)
Refinement: The TSF shall [***implement functionality using [mutually authenticated TLS as defined in the Package for Transport Layer Security, HTTPS in accordance with FCS_HTTPS_EXT.1]***] to provide a trusted communication channel between itself and authorized

IT entities supporting the following capabilities: audit server, [***authentication server***] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification and disclosure.

MDMPP40:FTP_ITC.1.2(1)

Refinement: The TSF shall [***implement functionality***] to permit the MDM Server or other authorized IT entities to initiate communication via the trusted channel.

MDMPP40:FTP_ITC.1.3(1)

Refinement: The TSF shall [***implement functionality***] to initiate communication via the trusted channel for [**exporting audit logs and LDAP authentication requests**].

5.1.7.2 Inter-TSF Trusted Channel (MDM Agent) (MDMPP40:FTP_ITC.1(2))

MDMPP40:FTP_ITC.1.1(2)

Refinement: The TSF shall [***implement functionality using [mutually authenticated TLS as defined in the Package for Transport Layer Security]***] to provide a trusted communication channel between itself (as a server) and the MDM Agent 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.

MDMPP40:FTP_ITC.1.2(2)

Refinement: The TSF shall [***functionality***] to permit the TSF and MDM Agent to initiate communication via the trusted channel.

MDMPP40:FTP_ITC.1.3(2)

Refinement: The TSF shall [***implement functionality***] to initiate communication via the trusted channel for all communication between the TSF and the MDM Agent.

5.1.7.3 Trusted Channel (MDMPP40:FTP_ITC_EXT.1)

MDMPP40:FTP_ITC_EXT.1.1

The TSF shall provide a communication channel between itself and [***an MDM Agent that is internal to the TOE, an MDM Agent that is external to the TOE***] that is logically distinct from other communication channels, as specified in [***FPT_ITT.1(2), FTP_ITC.1(2)***].

5.1.7.4 Trusted Path (for Remote Administration) (MDMPP40:FTP_TRP.1(1))

MDMPP40:FTP_TRP.1.1(1)

Refinement: The TSF shall [***implement functionality using [TLS as defined in the Package for Transport Layer Security, HTTPS in accordance with FCS_HTTPS_EXT.1]***] to provide a trusted communication path between itself as a [***server***] 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 modification and disclosure.

MDMPP40:FTP_TRP.1.2(1)

Refinement: The TSF shall [***implement functionality***] to permit remote administrators to initiate communication via the trusted path.

MDMPP40:FTP_TRP.1.3(1)

Refinement: The TSF shall [***implement functionality***] to require the use of the trusted path for all remote administration actions.

5.1.7.5 Trusted Path (for Enrollment) (MDMPP40:FTP_TRP.1(2))

MDMPP40:FTP_TRP.1.1(2)

Refinement: The TSF shall [***implement functionality using [selection: TLS as defined in the Package for Transport Layer Security, HTTPS in accordance with FCS_HTTPS_EXT.1]***] to provide a trusted communication path between itself (as a server) 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 from modification and disclosure.

MDMPP40:FTP_TRP.1.2(2)

Refinement: The TSF shall [***implement functionality***] to permit MD users to initiate communication via the trusted path.

MDMPP40:FTP_TRP.1.3(2)

Refinement: The TSF shall [***implement functionality***] to require the use of the trusted path for all MD user actions.

5.2 TOE Security Assurance Requirements

The SARs for the TOE are the components as specified in Part 3 of the Common Criteria. Note that the SARs have effectively been refined with the assurance activities explicitly defined in association with both the SFRs and SARs.

Requirement Class	Requirement Component
ADV: Development	ADV_FSP.1: Basic Functional Specification
AGD: Guidance documents	AGD_OPE.1: Operational User Guidance
	AGD_PRE.1: Preparative Procedures
ALC: Life-cycle support	ALC_CMC.1: Labelling of the TOE
	ALC_CMS.1: TOE CM Coverage
ATE: Tests	ATE_IND.1: Independent Testing - Conformance
AVA: Vulnerability assessment	AVA_VAN.1: Vulnerability Survey

Table 4 Assurance Components

5.2.1 Development (ADV)

5.2.1.1 Basic Functional Specification (ADV_FSP.1)

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.

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.

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.

5.2.2 Guidance documents (AGD)

5.2.2.1 Operational User Guidance (AGD_OPE.1)

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

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.

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.

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

5.2.2.2 Preparative Procedures (AGD_PRE.1)

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

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.

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.

5.2.3 Life-cycle support (ALC)

5.2.3.1 Labelling of the TOE (ALC_CMC.1)

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

ALC_CMC.1.1c
The TOE shall be labelled with its unique reference.

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

5.2.3.2 TOE CM Coverage (ALC_CMS.1)

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

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.

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

5.2.4 Tests (ATE)

5.2.4.1 Independent Testing - Conformance (ATE_IND.1)

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

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

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.

5.2.5 Vulnerability assessment (AVA)

5.2.5.1 Vulnerability Survey (AVA_VAN.1)

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

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

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.

6. TOE Summary Specification

This chapter describes the security functions:

- Security audit
- Cryptographic support
- Identification and authentication
- Security management
- Protection of the TSF
- TOE access
- Trusted path/channels

6.1 Security audit

MDMPP40:FAU_ALT_EXT.1: The MDM Server component of the TOE can be configured to alert the administrator on essentially anything in the device configuration database. When configured per guidance, the MDM server generates alerts for changes to enrolled devices (i.e., enroll and retire actions) and failed policy deployments. Note that while a device could effectively be unenrolled by being factory reset, there is no explicit unenroll function available to the MDM Agent, though a mobile device can be retired (and hence unenrolled) at the MDM Server. Alerts can be configured and can be displayed on the administrator console, can be sent to a configured e-mail address, text (SMS), or a push notification to a configured device.

MDMA10:FAU_ALT_EXT.2: The TOE supports the ability to periodically synchronize the MDM Server and MDM Agents. The synchronization includes retrieving information about the policies that are installed, thereby ensuring the MDM Server is informed about which policies have been applied. The MDM Server can force a check-in at any time and that would serve to determine MDM Agent connectivity status.

When a mobile device checks in, the mobile device performs a compliance check based on configured policies. If any of the settings have not been reported to and acknowledged by the MDM server, the mobile device reports those changes. Hence, if something happens, such as a network disruption, that prevents the MDM server from receiving the mobile device compliance information or prevents the mobile device from receiving an acknowledgement from the MDM server, that information will be sent the next time the device connects until it is finally acknowledged.

The MDM Agent will store unsent alerts in the application storage available on the mobile device and that space is limited only by the space available on the mobile device non-volatile flash.

MDMPP40:FAU_CRP_EXT.1: The MDM Server component of the TOE maintains a configuration database that can be queried to determine the current configuration of mobile devices and can be used to export that configuration information. The MDM server maintains a database of device information including OS versions, device model, installed applications, applied policies, etc. and this information can be exported by an administrator in CSV (comma separated values) format via the administrator web interface.

See also section 6.4, MDMPP40:FMT_POL_EXT.1 and MDMA10:FMT_POL_EXT.2 for more information about how policies are received and checked prior to application.

MDMPP40:FAU_GEN.1(1)/MDMPP40:FAU_GEN.1(2): The MDM Server component of the TOE can generate audit events for a wide range of security-relevant operations including start-up and shutdown, administrator functions, commands sent to MDM Agents, and others (see Table 2 Server Auditable Events), Failure to push a new application on a managed mobile device, and Failure to update an existing application on a managed mobile device. In addition to identifying the security event, each audit records incudes a time stamp, identify of the responsible user (where applicable), and the outcome of the event (success or failure).

MDMA10:FAU_GEN.1(2): The MDM Agent component of the TOE is able to generate records of the following auditable events: start-up and shutdown of the MDM Agent, change in MDM policy, any modification commanded by the MDM Server, and the auditable events listed in Table 3 Client Auditable Events. The MDM Agent component of the TOE includes in each audit record the following information: date and time of the event, type of event, subject identity, (if relevant) the outcome (success or failure) of the event, and the additional information identified in Table 3 Client Auditable Events.

MDMPP40:FAU_NET_EXT.1: The MDM Server component of the TOE provides the ability for an administrator to determine the connectivity status of any MDM Agent. Device check-in normally occurs periodically, where an administrator configured the period. Device check-ins can also be initiated by the mobile device user using the MDM Agent or by an administrator using the MDM Server web interface to cause an immediate check-in to ensure or determine the current connectivity status. While an administrator can check the last check-in status of any device via the administrator web interface, the administrator can also configure a policy to send an alert if a device has not checked-in for a configured number of days.

MDMPP40:FAU_SAR.1/MDMPP40:FAU_STG_EXT.1/MDMPP40:FAU_STG_EXT.2: The MDM Server component of the TOE provides the functions necessary for an administrator to review all of the collected audit records, while ensuring that the audit records cannot be modified. The MDM Server doesn't offer any functions that allow the audit log or individual audit records therein to be modified or inserted. The MDM server does offer an administrator a way to clear the MICS and MIFS audit data. The MDM Server component of the TOE also provides the ability to export audit records via a function available in the administrator web interface and the exported records are protected via the HTTPS/TLS connection to that interface. This export transmits audit data in either in CSV (comma separated values) format, text format, or a compressed archive format, depending upon the specific audit data being exported.

MDMA10:FAU_SEL.1(2): The MDM Agent component of the TOE stores most audit records in the mobile device audit log and thereby leverages the selection functions of the mobile device to allow audited events to be selected based on the following attributes: event type; success of auditable security events; failure of auditable security events. The remaining events (e.g., policy signature related audits and agent alerts) not stored in the platform audit log are always audited (i.e., cannot be de-selected) and are stored locally by the agent in its data space until delivered to the MDM server.

6.2 Cryptographic support

MDMPP40:FCS_CKM.1/MDMPP40:FCS_CKM.2/MDMPP40:FCS_COP.1(*): The TOE makes use of available cryptographic modules to perform cryptographic operations to support higher level functions (such as communication protocols). The TOE use a byte-oriented mode for hashing.

The MDM Server component includes the Bouncy Castle (1.0.2) cryptographic library – operating in the Java SE Runtime Environment 8 - and also utilizes the Red Hat Enterprise Linux OpenSSL Module (openssl-libs-1.0.2k-16.el7_6.1.x86_64) cryptographic functions.

The MDM Agent component includes its own OpenSSL () cryptographic library.

In the event of decryption errors, particularly for communication (e.g., key establishment), the associated function for both the server and agent will fail and be logged, where appropriate, as a higher level session failure with no specific details about the decryption failure being disclosed.

The available cryptographic functions are as follows:

Requirement	Component	Function	Details	FIPS Algorithm Certificate
MDMPP40:FCS_CKM.1	MDM Server Bouncy Castle	generate asymmetric cryptographic keys used for authentication	• ECC schemes using 'NIST curves' P-384 and [P-256, P-521, no other curves] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.4	A1336
MDMPP40:FCS_CKM.1	MDM Server OpenSSL	generate asymmetric cryptographic keys used for authentication	• ECC schemes using 'NIST curves' P-384 and [P-256, P-521, no other curves] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.4	A1337

Requirement	Component	Function	Details	FIPS Algorithm Certificate
MDMPP40:FCS_CKM.1	MDM Agent OpenSSL	generate asymmetric cryptographic keys used for authentication	• ECC schemes using 'NIST curves' P-384 and [P-256, P-521, no other curves] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.4	A1397
MDMPP40:FCS_CKM.2	MDM Server Bouncy Castle	perform cryptographic key establishment	• 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'	A1336
MDMPP40:FCS_CKM.2	MDM Server OpenSSL	perform cryptographic key establishment	• 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'	A1337
MDMPP40:FCS_CKM.2	MDM Agent OpenSSL	perform cryptographic key establishment	• 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'	A1397
MDMPP40:FCS_COP.1(1)	MDM Server Bouncy Castle	perform encryption/decryption	• AES-CBC (as defined in FIPS PUB 197 and NIST SP 800-38A) mode • AES-GCM (as defined in NIST SP 800-38D)	A1336
MDMPP40:FCS_COP.1(1)	MDM Server OpenSSL	perform encryption/decryption	• AES-CBC (as defined in FIPS PUB 197 and NIST SP 800-38A) mode • AES-GCM (as defined in NIST SP 800-38D)	A1337
MDMPP40:FCS_COP.1(1)	MDM Agent OpenSSL	perform encryption/decryption	• AES-CBC (as defined in FIPS PUB 197 and NIST SP 800-38A) mode • AES-GCM (as defined in NIST SP 800-38D)	A1397
MDMPP40:FCS_COP.1(2)	MDM Server Bouncy Castle	perform cryptographic hashing	SHA-256, SHA-384, SHA-512	A1336

Requirement	Component	Function	Details	FIPS Algorithm Certificate
MDMPP40:FCS_COP.1(2)	MDM Server OpenSSL	perform cryptographic hashing	SHA-256, SHA-384, SHA-512	A1337
MDMPP40:FCS_COP.1(2)	MDM Agent OpenSSL	perform cryptographic hashing	SHA-256, SHA-384, SHA-512	A1397
MDMPP40:FCS_COP.1(3)	MDM Server Bouncy Castle	perform cryptographic signature services (generation and verification)	● RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 4 ● ECDSA schemes using 'NIST curves' P-384 and [P-256, P-521, no other curves] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 5	A1336
MDMPP40:FCS_COP.1(3)	MDM Server OpenSSL	perform cryptographic signature services (generation and verification)	● RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 4 ● ECDSA schemes using 'NIST curves' P-384 and [P-256, P-521, no other curves] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 5	A1337
MDMPP40:FCS_COP.1(3)	MDM Agent OpenSSL	perform cryptographic signature services (generation and verification)	● RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 4 ● ECDSA schemes using 'NIST curves' P-384 and [P-256, P-521, no other curves] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 5	A1397
MDMPP40:FCS_COP.1(4)	MDM Server Bouncy Castle	perform keyed-hash message authentication	HMAC-SHA-256, HMAC-384, HMAC-SHA-512 with key length equal to hash size, and digest size of 256, 283, or 512 respectively	A1336
MDMPP40:FCS_COP.1(4)	MDM Server OpenSSL	perform keyed-hash message authentication	HMAC-SHA-256, HMAC-384, HMAC-SHA-512 with key length equal to hash size, and digest size of 256, 283, or 512 respectively	A1337

Requirement	Component	Function	Details	FIPS Algorithm Certificate
MDMPP40:FCS_COP.1(4)	MDM Agent OpenSSL	perform keyed-hash message authentication	HMAC-SHA-256, HMAC-384, HMAC-SHA-512 with key length equal to hash size, and digest size of 256, 283, or 512 respectively	A1397
MDMPP40:FCS_RBG_EXT.1	MDM Server Bouncy Castle	DRBG	CTR_DRBG (AES) 256-bits	A1336
MDMPP40:FCS_RBG_EXT.1	MDM Server OpenSSL	DRBG	CTR_DRBG (AES) 256-bits	A1337
MDMAEP30:FCS_RBG_EXT.1	MDM Agent OpenSSL	DRBG	CTR_DRBG (AES) 256-bits	A1397

Table 5 Algorithm Certificates

MDMPP40:FCS_CKM_EXT.4: The TOE destroys cryptographic keys when they are no longer in use. Keys stored in memory are overwritten with zeros, while keys stored on non-volatile media are overwritten as required by FIPS 140-2.

The following keys and CSPs are managed by the MDM Server and MDM Agent.

Component	Key or CSP	Where Stored	Destruction
MDM Agent	Certificate private key	Platform key storage	Not stored on media in plain text. Replaced when a new certificate is received
MDM Agent	Certificate private key	Volatile memory (in use)	Overwritten with zeroes when the Agent terminates.
MDM Agent	TLS session key	Volatile memory	Overwritten with zeroes when TLS session terminates
MDM Server	Web Portal certificate private key	Non-volatile storage	Not stored on media in plain text. Replaced when a new certificate is loaded
MDM Server	Web Portal certificate private key	Volatile memory (in use)	Overwritten with zeroes when the MDM server terminates.
MDM Server	Local CA certificate issuing certificate private keys	Non-volatile storage	Not stored on media in plain text. Replaced when a new certificate is loaded
MDM Server	Local CA certificate issuing certificate private keys	Volatile memory (in use)	Overwritten with zeroes when the MDM server terminates.
MDM Server	TLS session key	Volatile memory	Overwritten with zeroes when TLS session terminates

Table 6 Keys and CSPs

MDMPP40:FCS_HTTPS_EXT.1: Both the MDM Server and MDM Agent components of the TOE included the ability to support the HTTPS protocol (compliant with RFC 2818) so that remote users can securely connect using the web-based user interface and the server and agent can communicate with each other using HTTPS.

MDMPP40:FCS_RBG_EXT.1: The MDM Server component of the TOE includes a cryptographic module (Bouncy Castle) and accesses a second cryptographic module (OpenSSL). In the case of OpenSSL, the library provides an AES-256 CTR_DRBG (cert #1567, 1578, 1593, 1596) deterministic random bit generation that is seeded with 384-bits of entropy with a security strength of 256-bits. In the case of Bouncy Castle, the MDM Server implements a SHA-256 HMAC-DRBG (cert # 1636) also seeded with 384-bits of entropy with a security strength of 256-bits.

The MDM Agent component of the TOE incudes a cryptographic module (OpenSSL). The MDM Agent provides an AES-256 CTR-DRBG (cert #2311) seeded using 384-bits (also forming a 256-bit entropy input and 128-bit nonce in conformance with SP 800-90A) of seeding material and further conditioned using an OpenSSL conditioning function.

MDMPP40:FCS_STG_EXT.1/MDMA10:FCS_STG_EXT.1(2): Both the MDM Server and MDM Agent components of the TOE use platform provided storage to store persistent and private keys.

The MDM Server stores each of its keys and certificates in flat files that are assigned permissions to restrict those keys to the components/processes that use them. The certificates are further protected using PCKS #12 encryption. The MDM Agent utilizes the Android keystore API on the mobile device to store its keys and certificates both utilizing the platform storage and the platform encryption of the key storage.

PKGTLS11:FCS_TLS_EXT.1/PKGTLS11:FCS_TLSC_EXT.1/PKGTLS11:FCS_TLSS_EXT.1/PKGTLS11:FCS_TLSC_EXT.2/PKGTLS11:FCS_TLSS_EXT.2/PKGTLS11:FCS_TLSC_EXT.5/PKGTLS11:FCS_TLSS_EXT.4: The MDM Server component of the TOE includes the ability to support the TLS protocol (TLS 1.2 (RFC 5246) only) for the purposes of protecting audit records exported to an external server, protecting communication channels between the MDM Server and MDM Agent, and protecting (in conjunction with HTTPS) remote web-based administrator sessions. It will reject any attempts to use other TLS or SSL versions and TLS sessions will not be established if an applicable certificate is found to be invalid or if the remote network entity doesn't match the distinguished name (DN) in the certificate. The DN can be an exact match or a match with a wildcard in accordance with RFC 6125. Note that when a mobile device is enrolled it is assigned a unique UUID that is associated with the serial number of the device's certificate. When the mobile device connects and presents its certificate, the UUID in the certificate is used to look up the certificate serial number to ensure it matches the certificate that was presented. The MDM Server generates key agreement parameters using NIST curves secp256r1, secp384r1 and secp521r1 when using elliptic curve ciphers. The specific curve parameters always corresponds to the server certificate that is configured for client access and can only be changed by configuring a different server certificate. Note also that neither URI Service names nor pinned certificates are not supported by the MDM Server or Agent.

The MDM Server component of the TOE also includes an LDAP client that acts as a TLS client. It supports the TLS protocol (TLS 1.2 (RFC 5246)) for the purpose of protecting the communication channel between the MDM Server and secure LDAP server for authentication. TLS sessions will not be established if an applicable certificate is found to be invalid or if the remote LDAP server doesn't match the distinguished name (DN) in the certificate (i.e., FQDN). Note that the DN can appear in the certificate CN or SAN and any value in the SAN (when present) always takes precedent over values in the CN. The LDAP client supports only NIST curves secp256r1, secp384r1, and secp521r1 when using elliptic curve ciphers. This is the default behavior and there is no means to enable additional curves in the LDAP client.

The MDM Agent component of the TOE includes the ability to support the TLS protocol TLS 1.2 (RFC 5246) for the purposed of protecting the communication channel between the MDM Server and MDM Agent. TLS sessions will not be established if an applicable certificate if found to be invalid or if the remote network entity doesn't match the distinguished name (DN) in the certificate (i.e., FQDN). The MDM Agent supports only NIST curves secp256r1, secp384r1, and secp521r1 when using elliptic curve ciphers. This is the default behavior and there is no means to enable additional curves in the MDM Agent.

Each TLS implementation supports mutual authentication although the administrative web console is not configured to use mutual authentication. The MDM Server supports TLS session resumption using session tickets according to RFC 5077 on its TLS Server interface.

The following ciphers are supported by the MDM Server when acting as a TLS Server and by the MDM Agent when communicating with the MDM Server:

- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289.

The following ciphers are supported by the MDM Server when acting as a TLS Client and by the MDM Agent:

- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289.[2]

The LDAP client supports all the ciphers listed above.

6.3 Identification and authentication

MDMPP40:FIA_ENR_EXT.1: When a new device is being enrolled, the initial connection is made using TLS where only the MDM Server certificate is authenticated by the MDM Agent. Once the TLS connection is established, the mobile device user must be authenticated with a PIN or username and password that are configured on the MDM Server. Only after the MDM Server determines that the device can be enrolled (including ensuring the user has not exceeded the maximum number of enrolled devices and that any serial-number based device restrictions are met), the device is provisioned with an X.509v3 certificate so that subsequent communication between the MDM Agent and MDM Server will be protected by mutually authenticated TLS. When enrolled (and later when synchronizing), the MDM Server checks the OS version on the mobile device and will quarantine the device if it doesn't meet any configured restrictions. Note that quarantine removes most enterprise configurations, but leaves the ability to check-in. If the device is determined to be in compliance during a subsequent check-in, the quarantined accesses will be restored.

MDMA10:FIA_ENR_EXT.2: During enrollment, the MDM Agent records the unique URL (FQDN) of the MDM Server for future communication purposes. This value is initially configured by the mobile device user when attempting to enroll the mobile device.

MDMPP40:FIA_UAU.1: The MDM Server component of the TOE requires users to login (be authenticated) before they can perform any security-related functions. Mobile device users are initially authenticated using a provided PIN or password and after that the MDM Agent uses its provisioned X.509v3 certificate to authenticate itself to the MDM Server. Administrators are assigned usernames and passwords used for authentication when they access the MDM Server via the local console or web-based user interfaces.

MDMPP40:FIA_X509_EXT.1(1)/MDMPP40:FIA_X509_EXT.2/MDMPP40:FIA_X509_EXT.5: The MDM Server component of the TOE configures X.509v3 certificates as part of TLS authentication and will not establish TLS sessions if the applicable certificates cannot be determined to be valid. Certificate validation includes checking the validation path, basicConstraints, revocation, and extendedKeyUsage properties (see MDMPP40:FIA_X509_EXT.1.1). Certificates must also chain to a trusted root that is configured by the administrator.

The MDM Agent component of the TOE uses X.509v3 certificates as part of TLS authentication and will not establish TLS sessions if the applicable certificate is not valid. It also performs the validation checks including checking the validation path, basicConstraints, revocation, and extendedKeyUsage properties (see MDMPP40:FIA_X509_EXT.1.1).

[2] While the TOE MD agent is capable of using the four ECDHE_RSA based ciphersuites, because the TOE MDM server cannot be configured to use an RSA local CA certificate, the MD agent will never receive an RSA certificate for communication with an MDM server.

The MDM Agents receive unique certificates[3] during their enrollment process and they store the received certificates in the Android keystore (which stores the keys with permissions to only allow the applications themselves to access the keys).

6.4 Security management

MDMPP40:FMT_MOF.1(1)/MDMPP40:FMT_MOF.1(3): The MDM Server component of the TOE restricts all security management functions (identified below for FMT_SMF.1(1)/ FMT_SMF.1(2)/FMT_SMF.1(3)) to an authorized administrator. This is accomplished by role-based access controls (described below) assigned to each of the functions. Note that applications and application updates are generally handled just like policies – they are packaged using XML and digitally signed along with any other policy elements and sent securely from the MDM server to the MDM agent via secure TLS. Applications and updates can be initiated by the MDM server by creating a configuration policy that includes the application or update – in this case the application will be sent to the MDM agent during its next check-in. Alternately, the MDM server can be configured with a list of available applications and a mobile device user can use the MDM agent to query that list to select and install any available applications or associated updates.

MDMPP40:FMT_MOF.1(2): While most security management functions are restricted to an authorized administrator, the authorized administrator can enable mobile device users to enroll their mobile device. An authorized administrator registers a given device and provides the mobile device user a 6-12 digit PIN or password (that can be more than 15 characters) that will allow them to enroll the device.

MDMPP40:FMT_POL_EXT.1: The MDM Server provides digitally signed policies and policy updates to the MDM Agent. Policies and policy updates (including policy settings, configurations, and applications) are formatted using XML and the entire XML structure is hashed using SHA-256. The hash is then signed by the MDM server using the configured portal certificate (RSA-2048 or ECDSA-P384) and the signed hash is added to the policy structure prior to being sent to any agents via a secure TLS channel.

MDMA10:FMT_POL_EXT.2: The MDM Agent only accepts policies and policy updates that are digitally signed by the Enterprise. When the MDM Agent checks-in with the MDM server, the MDM server will send any policy or policy updates to the MDM agent via a secure TLS channel if the policy or policy update has not yet been applied by the MDM agent. When the MDM agent receives the policy or policy update, it creates a SHA-256 hash of its contents (less the signed hash), verifies the hash matches and then verifies that the hash was signed using the same MDM server portal certificate that was used to receive the policy or policy update. The MDM agent will only install policies and policy updates if both the hash matches and the hash was signed by the expected certificate. If the verification fails, the policy or policy update is discarded and an alert is sent to the MDM server. This process is repeated during the next checking since the policy current on the MDM server has not been applied.

MDMPP40:FMT_SAE_EXT.1: The MDM Server component of the TOE allows PINs to be issued to users for device enrollment. The PINs are assigned expiration periods after which they can no longer be used by the user to enroll a mobile device. Note that there is no similar expiration limit for usernames and passwords that might alternately be used for device enrollment.

MDMPP40:FMT_SMF.1(1): A primary function of the MDM Server component of the TOE is to manage mobile devices via installed MDM Agent components. The MDM Server supports the following commands and configuration policies on the mobile devices identified in section 1.4.1.1:

- transition to the locked state
- full wipe of protected data
- unenroll from management
- install policies
- query connectivity status
- query the current version of the MD firmware/software
- query the current version of the hardware model of the device
- query the current version of installed mobile applications
- import X.509v3 certificates into the Trust Anchor Database

[3] Certificates are unique because the MDM server's local CA issues a new certificate to each enrolled device during enrollment.

- install applications
- update system software
- remove applications and Enterprise applications
- wipe enterprise data as part of a full wipe of all protected data
- remove imported X.509v3 certificates and default X.509v3 certificates in the Trust Anchor Database
- import keys/secrets into the secure key storage
- destroy imported keys/secrets in the secure key storage
- read audit logs kept by the mobile device (Samsung only)
- password Policy:
  - minimum password length
  - minimum password complexity
  - maximum password lifetime
- session locking Policy:
  - screen-lock enabled/disabled
  - screen lock timeout
  - number of authentication failures
- wireless networks (SSIDs) to which the MD may connect protection (Samsung only)
- security Policy for each wireless network:
  - specify the CA(s) from which the MD will accept WLAN authentication server certificate(s)
  - ability to specify security type
  - ability to specify authentication protocol
  - specify the client credentials to be used for authentication
- application installation Policy by
  - specifying authorized application repository(s) (Google Playstore and MDM application server) (Samsung only)
- enable/disable policy for the camera and microphone (Camera-only on iOS)
- enable/disable policy for the VPN (iOS only)
- enable/disable policy for NFC, Bluetooth, Wi-Fi, and cellular radios (Samsung only)
- enable/disable policy for protocols supporting remote access (Hotspot, USB, and Bluetooth tethering) (Samsung only)
- enable/disable policy for developer modes (Samsung only)
- enable policy for data-at rest protection (Samsung only)
- enable policy for removable media's data-at-rest protection (Samsung only[4])
- enable/disable policy for display notification in the locked state of all notifications (iOS only)
- certificate used to validate digital signature on applications (iOS only)
- unlock banner policy
- configure the auditable items (Samsung only)
- enable/disable USB mass storage mode (Samsung only)
- enable/disable Hotspot functionality authenticated by pre-shared key and USB tethering with no authenticated (Samsung only)
- enable/disable location services (Samsung only)
- enable/disable policy for use of Biometric Authentication Factor

MDMPP40:FMT_SMF.1(2): In addition to managing mobile devices, the MDM Server component of the TOE supports the security management functions to configure and manage itself, including configuring a login banner. Among the available security management functions are the ability to configure X.509v3 certificates, managing the device registration process (enrolling specific devices and limiting the number of devices a user can enroll), configure server session lock timeout, configure periodicity of the following commands to the agent, query connectivity status, query the current version of the MD firmware/software, query the current version of the hardware model of the device, and query the current version of installed mobile applications.

[4] This policy is only enforced by devices that have removable media.

MDMPP40:FMT_SMF.1(3): Furthermore, in support of application hosting, the MDM server (which also serves as a MAS server) supports the configuration of application groups in the form of labels assigned to individual apps and devices. It also supports the ability to download applications for deployment.

MDMA10:FMT_SMF_EXT.4/MDMA10:FMT_UNR_EXT.1: The MDM Agent component of the TOE can be configured with an X.509v3 certificate suitable to facilitate secure communication with the MDM Server. This certificate is provisioned during device enrollment. The MDM Server can be configured to use an external CA or to use a local CA using an administrator-configured certificate to sign CSRs from the MDM Agent during enrollment. Once secure communication is enabled and the device is enrolled, the MDM Agent accepts commands and policies from the enrolled MDM Server and implements those commands and policies (identified above). The TOE supports two enrollment methods for Samsung devices and two for iOS. Samsung devices can enroll using a QR code or through an automatic enrollment of a Knox Mobile Enrollment (KME) registered device. iOS devices can enroll using the MDM Server web enrollment method or through an automatic enrollment of an Apple Device Enrollment Program (DEP) registered device. The MDM Agent can also be unenrolled from an enrolled MDM Server. This is accomplished by retiring the mobile device on the MDM Server or alternately by using the Android Settings - Device Administrators function to remove the MDM Agent Administrator access on the mobile device. Note that the MDM Agent can restrict the ability to unenroll and enrolled device using policy settings. Specifically, the mobile device user can be denied the permission to take away the administrator privileges from the MDM agent.

Once an MDM Agent is enrolled, it offers a limited set of trouble shooting functions to the mobile device user: force check-in, send agent log, reinstall certificates, and show warnings.

MDMPP40:FMT_SMR.1(1)/MDMPP40:FMT_SMR.1(2): The MDM Server component of the TOE implements a role-based access mechanism. Initially, there is only one security management role by default – administrator – and additional roles can be configured. The administrator has access to the local console command-line interface to perform low-level management functions as well as access to the web-based System Manager and Admin Portals. Within the Admin Portal additional users can be defined and assigned specific user and management roles as follows.

Users can connect to a user portal and perform functions on devices assigned to that user depending on their assigned roles. The following list of roles can be individually assigned to each user:

- Wipe Device (cause the registered device to be reset)
- Lock Device (lock the device)
- Unlock Device (unlock the device)
- Locate Device (retrieve location information from the device)
- Retire Device (unenroll the device)
- Register Device (initiate a PIN registration request or send registration instructions for a device)
- Change Device Ownership (change the device ownership to another defined user)

User added in the Admin Portal cannot access the web-based System Manager or console command-line interfaces. Furthermore, added users can access the Admin Portal only if they have been assigned an administrator role. The following administrator roles can be individually assigned to each user:

- View dashboard, device page, device details
- Manage devices
- Manage devices, restricted
- Wipe device
- Add device
- Manage ActiveSync device
- Manage AppTunnel
- Manage device enrollment (iOS)
- Delete retired device
- View apps in device details
- Locate device
- View label
- Manage Label
- View user
- Manage user

- Manage app
- Apply and remove application label
- View configuration
- Manage configuration
- Apply and remove configuration
- View policy
- Manage policy
- Apply and remove policy label
- View settings
- Manage settings
- View logs and events
- Manage logs and events
- Manage administrators and device spaces
- Manage content
- View content, apply and remove content labels

The role settings allow fine-grained definition of administrative users.

Minimally the following roles are supported in the TOE:

- Administrator – This is the initial administrator that has full control of all functions.
- Mobile device user – This is a user added in the Admin Portal, but not assigned ant administrative roles.
- Enrolled mobile device – This is a mobile device that has been enrolled by a mobile device user.
- Application access group – The TOE allows "Labels" to be defined and apps to be assigned. Devices can also be assigned to labels and the associated services to restrict or allow access to corresponding apps.
- Server primary administrator – Same as Administrator above.
- Security configuration administrator – This is a user with access to the web-based System Manager and console command-line interfaces, as well as most of the Admin Portal manage roles.
- Device user group administrator – This is a user that minimally has the manage user Admin Portal role, but is not assigned the manage administrators and device spaces role.
- Auditor – This is a user that minimally has the manage logs and events Admin Portal role. If necessary for a given deployment, an auditor can be provided access to the System Manager Portal to have access to low level protocol audit records (application logs).

6.5 Protection of the TSF

MDMPP40:FPT_API_EXT.1: The TOE server is essentially an appliance that incorporates CentOS. The TOE Android client makes use of the following platform APIs:

- Knox APIs (Samsung-only)
- Android for Work APIs
- Java Cryptographic APIs

MDMPP40:FPT_ITT.1(2): : The TOE provides a trusted channel between the MDM Server and Android MDM Agent that is protected through the use of TLS. During enrollment, only the MDM Server is authenticated by the MDM Agent. Once enrolled, all communication between the MDM Server and MDM Agent is protected using this channel using mutual authentication. Note that the MDM server also performs the MAS server functions, so the only distributed TOE components are the MDM server and associated MDM agents.

MDMPP40:FPT_LIB_EXT.1: The TOE server utilizes Bouncy Castle and OpenSSL for secure TLS communication for each of its secure connections. The TOE Android client utilizes OpenSSL for secure TLS communication for each of its secure connections.

MDMPP40:FPT_TST_EXT.1: The MDM Server utilizes FIPS 140-2 certified cryptographic modules that include self-tests to ensure the available cryptographic operations (including AES, RSA, ECDSA, HMAC-SHA2 functions) are performing correctly when starting up.

The MDM Server uses the RPM Package Manager (RPM) functions of the TOE to verify the integrity of its own executable files during start-up. The RPM Package Manager maintains a database of all installed RPMs, including the TSF, and during boot each RPM is checked for integrity with the standard CentOS (Red Hat) RPM integrity checking functions. Any errors are reported on the console and the MDM Server will attempt to restart if an error is detected without intervention by an administrator.

MDMPP40:FPT_TUD_EXT.1: The MDM Server component of the TOE provides functions to query and update the MDM Server software version. When updating the MDM Server software, each new update is installed as an RPM package and is verified using a MobileIron digital signature prior to installation.

RPMs are signed with either the MobileIron signing key which is an RSA 2048-bit key or the CentOS signing key which is RSA 4096-bit key. When a package is installed, the signature on the package is validated by the RPM tool. The RPM includes digests of the files within the RPM. These digests are stored in a database on the system during package install. During boot, the contents of each file are verified against the stored digests.

6.6 TOE access

MDMPP40:FTA_TAB.1: The MDM Server component of the TOE can be configured to display an administrator-defined message when an administrator is logging onto the MDM Server to access available security functions.

The web-based user interfaces (including the System Manager and Admin Portal) can be configured with a textual banner up to 2048 characters that is displayed on the login screen. The local console interface can be configured separately with a textual banner that is displayed immediately upon connecting, but before the user logs in (when using a password).

6.7 Trusted path/channels

MDMPP40:FTP_ITC.1(1)/MDMPP40:FTP_ITC.1(2)/MDMPP40:FTP_ITC_EXT.1: The MDM Server uses TLS to secure communication when exporting audit records (this is really part of MDMPP40:FTP_TRP.1(1) below) as well as an external LDAP authentication server.

The TOE also provides a trusted channel between the MDM Server and the Android MDM Agent that is protected through the use of TLS. This channel is internal to the TOE. The TOE also provides a trusted channel between the MDM Server and iOS MDM Agent that is protected through the use of TLS. The channel to the iOS MDM agent is external to the TOE. During enrollment, only the MDM Server is authenticated by the MDM Agent. Once enrolled, all communication between the MDM Server and MDM Agent is protected using this channel using mutual authentication.

MDMPP40:FTP_TRP.1(1): The MDM Server provides a web-based user interface (including a System Manager and Admin Portal) to the MDM Server for remote administration. Each web-based session can be initiated by administrators and is protected through the use of HTTPS/TLS. The MDM Server Platform also provides a restricted shell (CLISH) for low level management of the server accessible at a local console.

MDMPP40:FTP_TRP.1(2): The MDM Server provides a TLS interface as described above for MDMPP40:FPT_ITT.1(2) that can be accessed by mobile device users via the MDM Agent.