Samsung SDS Co., LTD EMM (MDMPP20/MDMAEP20) Security Target

Version 1.0
2018/07/04

Prepared for:

Samsung SDS Co., LTD
Samsung SDS Tower, 125, Olympic-ro 35-gil, Songpa-gu, Seoul, Korea 138-240

Prepared By:

Gossamer Laboratories
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 SDS EMM provided by Samsung SDS. The TOE is being evaluated as a mobile device management.

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.
    - 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 – Samsung SDS Co., LTD EMM (MDMPP20/MDMAEP20) Security Target

ST Version – Version 1.0

ST Date – 2018/07/04

1.2 TOE Reference

TOE Identification – Samsung SDS EMM v1.5.1 with patch for iOS11

TOE Developer – Samsung SDS Co., LTD

Evaluation Sponsor – Samsung SDS Co., LTD

1.3 TOE Overview

The Target of Evaluation (TOE) is Samsung SDS Co., LTD EMM.

The SDS EMM consists of an EMM Server and Agent, where the Server provides centralized management of mobile devices and the Agent software (installed on each device) enforces the policies of the Server on each device.

1.4 TOE Description

Samsung SDS offers the EMM Server as a software installation for Java 1.8 and Tomcat 7.0 running on the Microsoft Windows Server 2012 or Windows Server 2012 R2 operating system. Once installed, the EMM Server allows administrators to configure policies for devices. Administrators connect securely to the EMM Server using a web browser (whether local to the Server itself or remote) and through the EMM Server's web interface can enroll, audit, lock, unlock, manage, and set policies for enrolled mobile devices. The EMM Server includes the RSA Crypto-J 6.2 cryptographic module as part of its software, and the EMM Server's Microsoft Windows platform includes SQL server 2008-2014 and a Microsoft CA certificate authority.

Samsung SDS provides the EMM Agent software for evaluated Samsung mobile devices and provides a software interface to the evaluated Apple MDM Agent embedded within evaluated Apple mobile devices. The Agent software, once installed and enrolled with the EMM Server, will apply and enforce administrator configured policies communicated through the EMM to the Agent's running on the mobile devices.

During evaluation testing Gossamer testing the EMM Server and EMM Agent in the following configuration:

1) The EMM Server (version 1.5.1) installed upon the Microsoft Windows 2012 R2 operating system with Oracle JRE 1.8, Microsoft SQL Server 2014, and Microsoft's Certificate Authority (CA).
2) The EMM Client version 1.5.1 APKs (EMM Agent, PushAgent, and EMM Agent Resource) installed upon the following evaluated Android devices:
    a. (NIAP VID 10898) Samsung Galaxy Devices on Android 8: Samsung Galaxy S8, S8+, S8 Active, Note 8, S9 and S9+.
    b. (NIAP VID 10849) Samsung Galaxy Devices on Android 7.1: Samsung Galaxy Note8 and Tab Active 2.
    c. (NIAP VID 10809) Samsung Galaxy Devices with Android 7: Samsung Galaxy S6, S6 Active, S6 Edge, S6 Edge+, Note 5, S7, S7 Active, S7 Edge, S8, S8 Active, S8+, and Tab S3.
3) The EMM Client version 1.5.1 application installed upon an evaluated iPhone (see NIAP VID 10851).

Please note that the MDMPP20 requirements apply to the EMM Server and its ability to manage both Android and iOS devices, while the MDMAEP20 requirements apply only to the EMM Agent for Android (and do not apply to the iOS Agent, as that Agent was evaluated separately under VID 10851).

1.4.1 TOE Architecture

The EMM Server actually consists of the following different servers:

1. EMM Server – the main server running to which remote administrators connect. The EMM Server bears responsibility for all logic needed to manage mobile devices.
2. Push Server – the Push Server accepts connections from mobile devices and then relays the messages to and from the EMM Server (for example, to send policies to an agent, or to send back a reply from an agent). One can install multiple Push Servers, in order to allow the overall solution to scale the supported number of mobile devices (a single Push Server configuration was used during testing).
3. AppTunnel Server – this server accepts connections from the EMM Client (one of the three portions of the agent software on Android) and allows the Client to upload log files or download mobile applications to be installed by the agent.

The EMM Server allows administrators to create and enforce two different types of profiles:

An MDM Profile – to control all MDM configurable extensions (for example enforcing password complexity requirements); and

EMM Client profile – controls only the configuration of the SDS client app itself (e.g., how a user logs in).

The EMM Agent consists of three different components on evaluated Android platforms:

1. The EMM Client – at the highest level, this provides a UI through which the user may enroll their mobile device. This Client is also responsible for uploading audit logs to the EMM Server and for downloading mobile applications that the Server directs the agent to install.
2. The EMM Agent – this component provides most of the agent's core functionality including the application of policies, reporting policy event triggers to the Server, installation of applications, communication with the Server, among other things. The Agent operates without user intervention and enforces the policies of the Server.
3. The Push Agent – this lowest level component facilitates Push communications with a Push server. It allows both the EMM Agent and other mobile applications to send and receive Push messages.

The EMM Agent consists of a single component on evaluated iOS platforms:

1. The EMM application – this iOS application provides a user interface to allow the user to enroll their phone with their organization's SDS EMM Server. The application relies upon the evaluated, embedded Apple agent for all agent functionality.

The EMM Client presents the UI to allow users to start the enrollment process and, once enrolled, to log in and log out.

1.4.1.1 Physical Boundaries

The physical boundaries of the SDS EMM are the physical perimeter of the servers hosting the EMM Server and the physical perimeter of the mobile devices being managed by the EMM Server (put another way, the mobile devices running the EMM Agent).

The EMM Server also interacts with Microsoft SQL server and a MS CA.

1.4.1.2 Logical Boundaries

This section summarizes the security functions provided by EMM:

- 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 EMM Server can generate and store audit records for security-relevant events as they occur. These events are stored and protected by the EMM Server and can be reviewed by an authorized administrator. The EMM Server can export the majority of audit events directly through the HTTPS protected GUI in a CSV format. Some low-level events are maintained in text files on the TOE platform and can be exported via RDP using the TOE platform.. In both cases, the EMM Server protects the exported audit records using TLS (as part of HTTPS and RDP). The EMM Server also supports the ability to query information about MDM agents and export MDM configuration information.

The EMM Agent includes the ability to indicate (i.e., respond) to the EMM Server when it has been enrolled and when it applies policies successfully. The EMM 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 EMM Server and EMM Agent both include and have access to cryptographic modules with Cryptographic Algorithm Validation Program (CAVP) certified algorithms for a wide range of cryptographic functions including: asymmetric key generation and establishment, encryption/decryption, and 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 Server and Agent and between the Server and remote administrators.

1.4.1.2.3 Identification and authentication

The EMM Server authenticates mobile device users (MD users) and administrators prior to allowing those operators to perform any functions. This includes MD users enrolling their device with the EMM Server using the EMM Agent as well as an administrator logging on to manage the EMM Server configuration, MDM policies for mobile devices, etc.

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

1.4.1.2.4 Security management

The EMM Server is designed with two distinct user roles: administrator and mobile device user (MD user). The former interacts directly with the EMM Server through HTTPS (using a browser) while the latter is the user of a mobile device with the EMM Agent installed.

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

The EMM Agents provide the functions necessary to securely communicate and enroll with the EMM Server, apply policies received from the EMM Server, and report the results of applying policies.

1.4.1.2.5 Protection of the TSF

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

Both the EMM Server and Agent include self-testing capabilities to ensure that they are functioning properly as well as to cryptographically verify that their executable images have not been corrupted.

The EMM 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.

1.4.1.2.7 Trusted path/channels

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

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 EMM Server and EMM Agent.

1.4.2 TOE Documentation

Samsung SDS EMM Administrator's Guide, Version 1.5.1, December 2016

Samsung SDS EMM Installation Guide, Version 1.5.1, December 2016

Samsung SDS Push Installation Guide, Version 1.5.1, December 2016

Samsung SDS AppTunnel Installation Guide, Version 1.5.1, December 2016

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 4, September 2012.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 4, September 2012.
    - Part 3 Conformant
- Package Claims:
    - Protection Profile for Mobile Device Management, Version 2.0, 31 December 2014 (MDMPP20) with the following extended package:
        - Extended Package for Mobile Device Management Agents, Version 2.0, 31 December 2014 (MDMAEP20)

2.1 Conformance Rationale

The ST conforms to the MDMPP20/MDMAEP20. The security problem definition, security objectives, and security requirements have been drawn from the PP.

3. Security Objectives

The Security Problem Definition may be found in the MDMPP20/MDMAEP20 and this section reproduces only the corresponding Security Objectives for operational environment for reader convenience. The MDMPP20/MDMAEP20 offers additional information about the identified security objectives, but that has not been reproduced here and the MDMPP20/MDMAEP20 should be consulted if there is interest in that material.

In general, the MDMPP20/MDMAEP20 has defined Security Objectives appropriate for a mobile device management and as such are applicable to the SDS EMM TOE.

3.1 Security Objectives for the Operational Environment

OE.IT_ENTERPRISE The Enterprise IT infrastructure provides security for a network that is available to the TOE and mobile devices that prevents unauthorized access.

OE.MDM_SERVER_PLATFORM The MDM Server relies upon a trustworthy platform and local network from which it provides administrative capabilities.

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

OE.PROPER_USER Users of the mobile device are trained to securely use the mobile device and apply all guidance in a trusted manner.

OE.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.

T.NETWORK_EAVESDROP Unauthorized entities may intercept communications between the MDM and mobile devices to monitor, gain access to, disclose, or alter remote management commands. Unauthorized entities may intercept unprotected wireless communications between the mobile device and the Enterprise to monitor, gain access to, disclose, or alter TOE data.

4. Extended Components Definition

All of the extended requirements in this ST have been drawn from the MDMPP20/MDMAEP20. The MDMPP20/MDMAEP20 defines the following extended requirements and since they are not redefined in this ST the MDMPP20/MDMAEP20 should be consulted for more information in regard to those CC extensions.

Extended SFRs:

- FAU_ALT_EXT.1: Server Alerts
- FAU_ALT_EXT.2: Agent Alerts
- FAU_CRP_EXT.1: Support for Compliance Reporting of Mobile Device Configuration
- FAU_NET_EXT.1: Network Reachability Review
- FAU_STG_EXT.1: External Audit Trail Storage
- FAU_STG_EXT.2: Audit Event Storage
- FCS_CKM_EXT.4: Cryptographic Key Destruction
- FCS_HTTPS_EXT.1: HTTPS Protocol
- FCS_IV_EXT.1: Initialization Vector Generation
- FCS_RBG_EXT.1: Extended: Random Bit Generation
- FCS_STG_EXT.1: Cryptographic Key Storage
- FCS_STG_EXT.2: Encrypted Cryptographic Key Storage
- FCS_STG_EXT.4: Cryptographic Key Storage
- FCS_TLSC_EXT.1: Cryptographic Support (FCS)
- FCS_TLSS_EXT.1: TLS Server Protocol
- FIA_ENR_EXT.1: Enrollment of Mobile Device into Management
- FIA_ENR_EXT.2: Enrollment of Mobile Device into Management
- FIA_X509_EXT.1: X509 Validation
- FIA_X509_EXT.2: X509 Authentication
- FMT_SMF_EXT.3: Specification of Management Functions
- FMT_UNR_EXT.1: User Unenrollment Prevention
- FPT_TST_EXT.1: Protection of the TSF (FPT)
- FPT_TUD_EXT.1: Trusted Update

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 MDMPP20/MDMAEP20. The refinements and operations already performed in the MDMPP20/MDMAEP20 are not identified (e.g., highlighted) here, rather the requirements have been copied from the MDMPP20/MDMAEP20 and any residual operations have been completed herein. Of particular note, the MDMPP20/MDMAEP20 made a number of refinements and completed some of the SFR operations defined in the Common Criteria (CC) and that PP/EP should be consulted to identify those changes if necessary.

The SARs are also drawn from the MDMPP20/MDMAEP20 which includes all the SARs for EAL 1. However, the SARs are effectively refined since requirement-specific 'Assurance Activities' are defined in the MDMPP20/MDMAEP20 that serve to ensure corresponding evaluations will yield more practical and consistent assurance than the EAL 1 assurance requirements alone. The MDMPP20/MDMAEP20 should be consulted for the assurance activity definitions.

5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by the SDS EMM TOE.

Requirement Class	Requirement Component
FAU: Security audit	FAU_ALT_EXT.1: Server Alerts
	FAU_ALT_EXT.2: Agent Alerts
	FAU_CRP_EXT.1: Support for Compliance Reporting of Mobile Device Configuration
	FAU_GEN.1(1): Audit Data Generation
	FAU_GEN.1(2): Audit Generation (MAS Server)
	FAU_GEN.1(3): Audit Data Generation (MDM Agent)
	FAU_NET_EXT.1: Network Reachability Review
	FAU_SAR.1: Audit Review
	FAU_STG_EXT.1: External Audit Trail Storage
	FAU_STG_EXT.1(2): External Audit Trail Storage (MAS Server)
	FAU_STG_EXT.2: Audit Event Storage
FCS: Cryptographic support	FCS_CKM.1: Cryptographic Key Generation - Server
	FCS_CKM.1(1): Cryptographic Key Generation - Agent
	FCS_CKM.2: Cryptographic Key Establishment - Server
	FCS_CKM.2(1): Cryptographic Key Establishment - Agent
	FCS_CKM_EXT.4: Cryptographic Key Destruction - Server
	FCS_CKM_EXT.4(1): Cryptographic Key Destruction - Agent
	FCS_COP.1(1): Cryptographic Operation (Confidentiality Algorithms) - Server
	FCS_COP.1(2): Cryptographic Operation (Hashing) - Server
	FCS_COP.1(3): Cryptographic Operation (Digital Signature) - Server
	FCS_COP.1(4): Cryptographic Operation (Keyed-Hash Message Authentication) - Server
	FCS_COP.1(5): Cryptographic Operation (Confidentiality Algorithms) - Agent
	FCS_COP.1(6): Cryptographic Operation (Hashing) - Agent
	FCS_COP.1(7): Cryptographic Operation (Digital Signature) - Agent
	FCS_COP.1(8): Cryptographic Operation (Keyed-Hash Message Authentication) - Agent
	FCS_HTTPS_EXT.1: HTTPS Protocol - Server
	FCS_IV_EXT.1: Initialization Vector Generation
	FCS_RBG_EXT.1: Extended: Random Bit Generation - Server

Requirement Class	Requirement Component
	FCS_RBG_EXT.1(1): Extended: Random Bit Generation - Agent
	FCS_STG_EXT.1: Cryptographic Key Storage
	FCS_STG_EXT.2: Encrypted Cryptographic Key Storage
	FCS_STG_EXT.4: Cryptographic Key Storage
	FCS_TLSC_EXT.1: Cryptographic Support (FCS)
	FCS_TLSS_EXT.1: TLS Server Protocol
FIA: Identification and authentication	FIA_ENR_EXT.1: Enrollment of Mobile Device into Management
	FIA_ENR_EXT.2: Enrollment of Mobile Device into Management
	FIA_UAU.1: Timing of Authentication
	FIA_X509_EXT.1: X509 Validation - Server
	FIA_X509_EXT.1(1): X509 Validation - Agent
	FIA_X509_EXT.2: X509 Authentication - Server
	FIA_X509_EXT.2(1): X509 Authentication - Agent
FMT: Security management	FMT_MOF.1(1): Management of Functions in MDM Server
	FMT_MOF.1(2): Management of Enrollment Function
	FMT_MOF.1(3): Management of Functions in MAS Server
	FMT_MOF.1(4): Management of Download Function in MAS Server
	FMT_SMF.1(1): Specification of Management Functions (Server configuration of Agent)
	FMT_SMF.1(2): Specification of Management Functions (Server Configuration of Server)
	FMT_SMF.1(3): Specification of Management Functions (MAS Server)
	FMT_SMF_EXT.3: Specification of Management Functions
	FMT_SMR.1(1): Security Management Roles - Agent
	FMT_SMR.1(2): Security Management Roles - Server
	FMT_UNR_EXT.1: User Unenrollment Prevention
FPT: Protection of the TSF	FPT_ITT.1(1): Basic Internal TSF Data Transfer Protection (MDM Server)
	FPT_ITT.1(2): Basic Internal TSF Data Transfer Protection (Distributed TOE)
	FPT_ITT.1(3): Basic Internal TSF Data Transfer Protection (MAS Server)
	FPT_TST_EXT.1: Protection of the TSF (FPT) - Server
	FPT_TST_EXT.1(1): Protection of the TSF (FPT)
	FPT_TUD_EXT.1: Trusted Update
FTA: TOE access	FTA_TAB.1: Default TOE Access Banners
FTP: Trusted path/channels	FTP_ITC.1(1): Inter-TSF Trusted Channel (Authorized IT Entities) - Server
	FTP_ITC.1(3): Inter-TSF Trusted Channel (Authorized IT Entities)
	FTP_TRP.1(1): Trusted Path for Remote Administration
	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 (FAU_ALT_EXT.1)

FAU_ALT_EXT.1.1

The MDM Server shall alert the administrators in the event of any of the following:
a. change in enrollment status;
b. failure to apply policies to a mobile device;
c. [***[no other events]***].

5.1.1.2 Agent Alerts (FAU_ALT_EXT.2)

FAU_ALT_EXT.2.1

The MDM **Android** Agent shall provide an alert via the trusted channel to the MDM Server in the event of any of the following:
a. successful application of policies to a mobile device;
b. [***receiving, generating***] periodic reachability events;
[***c. change in enrollment state,***
d. failure to install an application from the MAS Server,
e. failure to update an application from the MAS Server,
f. [***no other events***]].

FAU_ALT_EXT.2.2

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

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

FAU_CRP_EXT.1.1

The MDM Server shall provide [***an interface that permits the export of data about the configuration of enrolled devices***] to authorized entities over an authenticated [***TLS/HTTPS***] trusted communication channel. The provided information for each enrolled mobile device includes:
a. The current version of the MD firmware/software
b. The current version of the hardware model of the device
c. The current version of installed mobile applications
d. List of MD configuration policies that are in place on the device (as defined in FMT_SMF.1.1(1))
e. [***no other information***].

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

FAU_GEN.1(1).1

Refinement: The MDM Server shall be able to generate an audit record of the following auditable events:
a. Start-up and shutdown of the MDM Server software;
b. All administrative actions;
c. Commands issued from the MDM Server to an MDM Agent;
d. Specifically defined auditable events listed in **Table 2 Auditable Events**; and
e. [**no other events**].

FAU_GEN.1(1).2

Refinement: The [***MDM Server***] shall record within each MDM Server 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 Auditable Events**,
- [**no other audit relevant information**].

Requirement	Auditable Events	Additional Content
FAU_ALT_EXT.1	Type of alert.	Identity of Mobile Device that sent alert.
FAU_ALT_EXT.2	Type of alert.	None
FAU_CRP_EXT.1	None	None
FAU_GEN.1(1)	(Removed per TD0082)	None
FAU_GEN.1(2)	None	None
FAU_NET_EXT.1	None	None

Requirement	Auditable Events	Additional Content
FAU_SAR.1	None	None
FAU_STG_EXT.1	None	None
FAU_STG_EXT.1(2)	None	None
FAU_STG_EXT.2	None	None
FCS_CKM.1	Failure of key generation activity for authentication keys.	None
FCS_CKM.1(1)	None	None
FCS_CKM.2	None	None
FCS_CKM.2(1)	None	None
FCS_CKM_EXT.4	None	None
FCS_CKM_EXT.4(1)	None	None
FCS_COP.1(1)	None	None
FCS_COP.1(2)	None	None
FCS_COP.1(3)	None	None
FCS_COP.1(4)	None	None
FCS_COP.1(5)	None	None
FCS_COP.1(6)	None	None
FCS_COP.1(7)	None	None
FCS_COP.1(8)	None	None
FCS_HTTPS_EXT.1	Failure of the certificate validity check.	None
FCS_IV_EXT.1	None	None
FCS_RBG_EXT.1	Failure of the randomization process.	None
FCS_RBG_EXT.1(1)	None	None
FCS_STG_EXT.1	None	None
FCS_STG_EXT.2	None	None
FCS_STG_EXT.4	None	None
FCS_TLSC_EXT.1	Failure to establish a TLS session. Failure to verify presented identifier.	Reason for failure. Presented identifier and reference identifier.
FCS_TLSS_EXT.1	Failure to establish a TLS session.	Reason for failure.
FIA_ENR_EXT.1	Failure of MD user authentication.	Presented credentials.
FIA_UAU.1	None	None
FIA_X509_EXT.1	Failure to validate X.509 certificate.	Reason for failure.
FIA_X509_EXT.1(1)	None	None
FIA_X509_EXT.2	Failure to establish connection to determine revocation status.	None
FIA_X509_EXT.2(1)	None	None
FMT_MOF.1(1)	Issuance of command to perform function. Change of policy settings.	Command sent and identity of MDM Agent recipient. Query responses. Policy changed and value or full policy.
FMT_MOF.1(2)	Enrollment by a user.	Identity of user.
FMT_MOF.1(3)	None	None
FMT_MOF.1(4)	None	None
FMT_SMF.1(1)	None	None
FMT_SMF.1(2)	Success or failure of function.	None
FMT_SMF.1(3)	None	None
FMT_SMF_EXT.3	Success or failure of function.	None
FMT_SMR.1(1)	None	None
FMT_SMR.1(2)	None	None
FMT_UNR_EXT.1	Attempt to unenroll.	None
FPT_ITT.1(1)	None	None
FPT_ITT.1(2)	None	None

Requirement	Auditable Events	Additional Content
FPT_ITT.1(3)	None	None
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.
FPT_TST_EXT.1(1)	None	None
FPT_TUD_EXT.1	Success or failure of signature verification.	None
FTA_TAB.1	Change in banner setting.	None
FTP_ITC.1(1)	None	None
FTP_ITC.1(3)	None	None
FTP_TRP.1(1)	Initiation and termination of the trusted channel.	Trusted channel protocol. Identity of administrator.
FTP_TRP.1(2)	Initiation and termination of the trusted channel.	Trusted channel protocol.

Table 2 Auditable Events

5.1.1.5 Audit Generation (MAS Server) (FAU_GEN.1(2))

FAU_GEN.1(2).1

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.

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**].

5.1.1.6 Audit Data Generation (MDM Agent) (FAU_GEN.1(3))

FAU_GEN.1(3).1

Refinement: The MDM **Android** Agent shall be able to generate an MDM Agent audit record of the following auditable events:
a. Start-up and Shutdown of the audit functions
b. Change in MDM policy; and
c. Any modification commanded by the MDM Server,
d. Specifically defined auditable events listed in **Table 2 Auditable Events**
e. [assignment: other events].

FAU_GEN.1(3).2

Refinement: The [selection: TSF, TOE platform] shall record within each MDM Agent 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 Auditable Events** [assignment: other audit relevant information].

5.1.1.7 Network Reachability Review (FAU_NET_EXT.1)

FAU_NET_EXT.1.1

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

5.1.1.8 Audit Review (FAU_SAR.1)

FAU_SAR.1.1

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

FAU_SAR.1.2

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

5.1.1.9 External Audit Trail Storage (FAU_STG_EXT.1)

FAU_STG_EXT.1.1

The [***MDM Server***] shall be able to transmit the generated audit data to an external IT entity using a trusted channel implementing the [***TLS***] protocol.

5.1.1.10 External Audit Trail Storage (MAS Server) (FAU_STG_EXT.1(2))

FAU_STG_EXT.1(2).1

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.

5.1.1.11 Audit Event Storage (FAU_STG_EXT.2)

FAU_STG_EXT.2.1

The [***MDM Server***] shall 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 (FCS_CKM.1) - Server

FCS_CKM.1.1

Refinement: The [***TSF***] shall generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm
[***- RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.3 (TD0107 applied),***
- ECC schemes using 'NIST curves' P-384 and [P-256] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.4 (TD0084 applied)].

5.1.2.2 Cryptographic Key Generation (FCS_CKM.1(1)) - Agent

FCS_CKM.1(1).1

Refinement: The [***TSF***] shall generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm
[[***- RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: [o FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.3] (TD0107 applied),***
- ECC schemes using 'NIST curves' P-384 and [P-256] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.4]. ***TD0084 applied)***

5.1.2.3 Cryptographic Key Establishment (FCS_CKM.2) - Server

FCS_CKM.2.1

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

[***- RSA-based key establishment schemes that meets the following: NIST Special Publication 800-56B, 'Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography',***
- Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography'].

5.1.2.4 Cryptographic Key Establishment (FCS_CKM.2(1)) - Agent

FCS_CKM.2(1).1

Refinement: The [***TOE platform***] shall perform cryptographic key establishment in accordance with a specified cryptographic key establishment method:
[***- RSA-based key establishment schemes that meets the following: NIST Special Publication 800-56B, 'Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography',***
- Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography'].

5.1.2.5 Cryptographic Key Destruction (FCS_CKM_EXT.4) - Server

FCS_CKM_EXT.4.1

The [***TSF***] shall destroy cryptographic keys in accordance with the specified cryptographic key destruction methods:
- by clearing the KEK encrypting the target key,
- in accordance with the following rules:
  - o For volatile memory, the destruction shall be executed by a single direct overwrite [***consisting of zeroes***].
  - o For non-volatile EEPROM, the destruction shall be executed by a single direct overwrite consisting of a pseudo random pattern using the TSF's RBG (as specified in FCS_RBG_EXT.1), followed a read-verify.
  - o 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***]. (TD0057 applied)
  - o For non-volatile flash memory that is wear-leveled, the destruction shall be executed [***by a single direct overwrite consisting of zeros***].
  - o For non-volatile memory other than EEPROM and flash, the destruction shall be executed by overwriting three or more times with a random pattern that is changed before each write.

(TD0047 applied)

FCS_CKM_EXT.4.2

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

5.1.2.6 Cryptographic Key Destruction (FCS_CKM_EXT.4(1)) - Agent

FCS_CKM_EXT.4(1).1

The [***TOE Platform***] shall destroy cryptographic keys in accordance with the specified cryptographic key destruction methods:
- by clearing the KEK encrypting the target key,
- in accordance with the following rules:

o For volatile memory, the destruction shall be executed by a single direct overwrite [***consisting of zeroes***].
o For non-volatile EEPROM, the destruction shall be executed by a single direct overwrite consisting of a pseudo random pattern using the TSF's RBG (as specified in FCS_RBG_EXT.1), followed a read-verify.
o 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]. (TD0057 applied)
o For non-volatile flash memory that is wear-leveled, the destruction shall be executed [***by a single direct overwrite consisting of zeros***].
o For non-volatile memory other than EEPROM and flash, the destruction shall be executed by overwriting three or more times with a random pattern that is changed before each write. (TD0047 applied)

FCS_CKM_EXT.4(1).2

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

5.1.2.7 Cryptographic Operation (Confidentiality Algorithms) (FCS_COP.1(1)) - Server

FCS_COP.1(1).1

Refinement: The [***TSF***] shall perform encryption/decryption in accordance with a specified cryptographic algorithm
[***- AES-CBC (as defined in FIPS PUB 197 C98:C101and NIST SP 800-38A) mode,***
- AES-GCM (as defined in NIST SP 800-38D)]
and cryptographic key sizes 128-bit key sizes and [***256-bit key sizes***].

5.1.2.8 Cryptographic Operation (Hashing) (FCS_COP.1(2)) - Server

FCS_COP.1(2).1

Refinement: The [***TSF***] shall perform cryptographic hashing in accordance with a specified cryptographic algorithm [***SHA-1, SHA-256, SHA-384, SHA-512***] and message digest sizes [***160, 256, 384, 512***] bits that meet the following: FIPS Pub 180-4.

5.1.2.9 Cryptographic Operation (Digital Signature) (FCS_COP.1(3)) - Server

FCS_COP.1(3).1

Refinement: The [***TSF***] shall 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] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 5 (TD0084 applied)***].

5.1.2.10 Cryptographic Operation (Keyed-Hash Message Authentication) (FCS_COP.1(4)) - Server

FCS_COP.1(4).1

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

5.1.2.11 Cryptographic Operation (Confidentiality Algorithms) (FCS_COP.1(5)) - Agent

FCS_COP.1(5).1

Refinement: The [***TOE platform***] shall perform encryption/decryption in accordance with a specified cryptographic algorithm
[***- AES-CBC (as defined in FIPS PUB 197 C98:C101and NIST SP 800-38A) mode,***
- AES-GCM (as defined in NIST SP 800-38D)]
and cryptographic key sizes 128-bit key sizes and [***256-bit key sizes***].

5.1.2.12 Cryptographic Operation (Hashing) (FCS_COP.1(6)) - Agent

FCS_COP.1(6).1

Refinement: The [***TOE platform***] shall perform cryptographic hashing in accordance with a specified cryptographic algorithm [***SHA-1, SHA-256, SHA-384, SHA-512***] and message digest sizes [***160, 256, 384, 512***] bits that meet the following: FIPS Pub 180-4.

5.1.2.13 Cryptographic Operation (Digital Signature) (FCS_COP.1(7)) - Agent

FCS_COP.1(7).1

Refinement: The [***TOE platform***] shall 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-256, P-384 and [no other curves] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 5***].

5.1.2.14 Cryptographic Operation (Keyed-Hash Message Authentication) (FCS_COP.1(8)) - Agent

FCS_COP.1(8).1

Refinement: The [***TOE platform***] shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm HMAC-[***SHA-1, SHA-256, SHA-384***], key sizes [**160, 256, 384**], and message digest sizes [***160, 256, 384***] bits that meet the following: FIPS Pub 198-1, 'The Keyed-Hash Message Authentication Code', and FIPS Pub 180-4, 'Secure Hash Standard.'

5.1.2.15 HTTPS Protocol (FCS_HTTPS_EXT.1) - Server

FCS_HTTPS_EXT.1.1

The [***MDM Server***] shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2

The [***MDM Server***] shall implement HTTPS using TLS as specified in FCS_TLSS_EXT.1.

FCS_HTTPS_EXT.1.3

The [***MDM Server***] shall [***not establish the connection***] if the peer certificate is deemed invalid.

5.1.2.16 Initialization Vector Generation (FCS_IV_EXT.1)

FCS_IV_EXT.1.1

The MDM Server shall generate IVs in accordance with Table 9 of MDMPP20.

5.1.2.17 Extended: Random Bit Generation (FCS_RBG_EXT.1) - Server

FCS_RBG_EXT.1.1

The [***TSF***] shall perform all deterministic random bit generation services in accordance with NIST Special Publication 800-90A using [***HMAC_DRBG (SHA-256)***]. (TD0079 Applied)

FCS_RBG_EXT.1.2

The deterministic RBG shall be seeded by an entropy source that accumulates entropy from [***a TSF software-based noise source***] with a minimum of [***256-bit***] 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.

5.1.2.18 Extended: Random Bit Generation (FCS_RBG_EXT.1(1)) - Agent

FCS_RBG_EXT.1(1).1

The [***TOE platform***] shall perform all deterministic random bit generation services in accordance with NIST Special Publication 800-90A using [***CTR_DRBG (AES)***]. (TD0079 Applied)

FCS_RBG_EXT.1(1).2

The deterministic RBG shall be seeded by an entropy source that accumulates entropy from [***a***

platform-based RBG] with a minimum of [***256-bit***] 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.

5.1.2.19 Cryptographic Key Storage (FCS_STG_EXT.1)

FCS_STG_EXT.1.1

The [***TSF***] shall store persistent secrets and private keys when not in use, in [***as specified in FCS_STG_EXT.2***].

5.1.2.20 Encrypted Cryptographic Key Storage (FCS_STG_EXT.2)

FCS_STG_EXT.2.1

The MDM Server shall encrypt all keys using AES in the [***CBC mode***].

5.1.2.21 Cryptographic Key Storage (FCS_STG_EXT.4)

FCS_STG_EXT.4.1

The MDM **Android** Agent shall store persistent secrets and private keys when not in use in platform-provided key storage.

5.1.2.22 Cryptographic Support (FCS) (FCS_TLSC_EXT.1)

FCS_TLSC_EXT.1.1

The [***TSF***] shall implement [***TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)***] supporting the following ciphersuites:
- Mandatory Ciphersuites:
  o TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246;
- Optional Ciphersuites:
  [***o TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246,***
  o TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,
  o TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,
  o TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,
  o TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,
  o TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,
  o TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246,
  o TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
  o TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
  o TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
  o TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289].

FCS_TLSC_EXT.1.2

The [***TSF, TSF Platform***] shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.1.3

The [***TSF, TSF Platform***] shall only establish a trusted channel if the peer certificate is valid.

FCS_TLSC_EXT.1.4

The [***TSF, TSF Platform***] shall support mutual authentication using X.509v3 certificates.

FCS_TLSC_EXT.1.5

The [***TSF, TSF Platform***] shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [***secp256r1, secp384r1, secp521r1***] and no other curves.

5.1.2.23 TLS Server Protocol (FCS_TLSS_EXT.1)

FCS_TLSS_EXT.1.1

The [***MDM Server, MDM Server platform***] shall implement [***TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)***] supporting the following ciphersuites:
- Mandatory Ciphersuites:

o TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246;
- Optional Ciphersuites:
[***o TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246,***
o TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,
o TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,
o TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,
o TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,
o TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,
o TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246,
o TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
o TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
o TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
o TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289].

FCS_TLSS_EXT.1.2
The [***MDM Server***] shall deny connections from clients requesting SSL 1.0, SSL 2.0, SSL 3.0 and [***TLS 1.0***].

FCS_TLSS_EXT.1.3
The [***MDM Server***] shall support mutual authentication of TLS clients using X.509v3 certificates.

FCS_TLSS_EXT.1.4
The [***MDM Server***] shall not establish a trusted channel if the peer certificate is invalid.

FCS_TLSS_EXT.1.5
The [***MDM Server***] shall not establish a trusted channel if the distinguished name (DN) or Subject Alternative Name (SAN) contained in a certificate does not match the expected identifier for the peer.

FCS_TLSS_EXT.1.6
The [***MDM Server***] shall generate key agreement parameters [***over NIST curves [secp256r1, secp384r1] and no other curves***].

5.1.3 Identification and authentication (FIA)

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

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

FIA_ENR_EXT.1.2
The MDM Server shall limit the user's enrollment of devices to [***a number of devices***].

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

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

5.1.3.3 Timing of Authentication (FIA_UAU.1)

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

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

5.1.3.4 X509 Validation (FIA_X509_EXT.1) - Server

FIA_X509_EXT.1.1

The [***TSF***] shall validate certificates in accordance with the following rules:
- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The TSF shall validate the revocation status of the certificate using [***a Certificate Revocation List (CRL) as specified in RFC 5759***].
- 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 extendedKeyUsage field.
  o OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.
  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 extendedKeyUsage field.

FIA_X509_EXT.1.2

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

5.1.3.5 X509 Validation (FIA_X509_EXT.1(1)) - Agent

FIA_X509_EXT.1(1).1

The [***TOE platform***] shall validate certificates in accordance with the following rules:
- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The TSF shall validate the revocation status of the certificate using [***a Certificate Revocation List (CRL) as specified in RFC 5759***].
- 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 extendedKeyUsage field.
  o OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.
  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 extendedKeyUsage field.

FIA_X509_EXT.1(1).2

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

5.1.3.6 X509 Authentication (FIA_X509_EXT.2) - Server

FIA_X509_EXT.2.1

The [***TSF***] shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [***TLS, HTTPS***], and [***no additional uses***].

FIA_X509_EXT.2.2
When the [***TSF***] cannot establish a connection to determine the validity of a certificate, the [***TSF***] shall [***not accept the certificate***].

FIA_X509_EXT.2.3
The [***TSF***] shall require a unique certificate for each client device.

5.1.3.7 X509 Authentication (FIA_X509_EXT.2(1)) - Agent

FIA_X509_EXT.2(1).1
The [***TSF, TOE platform***] shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [***TLS***], and [***no additional uses***].

FIA_X509_EXT.2(1).2
When the [***TSF***] cannot establish a connection to determine the validity of a certificate, the [***TSF***] shall [***not accept the certificate***].

FIA_X509_EXT.2(1).3
The [***TSF, TOE platform***] shall require a unique certificate for each client device.

5.1.4 Security management (FMT)

5.1.4.1 Management of Functions in MDM Server (FMT_MOF.1(1))

FMT_MOF.1(1).1
Refinement: The MDM Server shall restrict the ability to perform the functions
- listed in FMT_SMF.1(1)
- enable, disable, and modify policies listed in FMT_SMF.1(1)
- listed in FMT_SMF.1(2)
to Authorized Administrators.

5.1.4.2 Management of Enrollment Function (FMT_MOF.1(2))

FMT_MOF.1(2).1
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 (FMT_MOF.1(3))

FMT_MOF.1(3).1
Refinement: The MAS Server shall restrict the ability to configure user groups for user-access to specific applications to the administrator.

5.1.4.4 Management of Download Function in MAS Server (FMT_MOF.1(4))

FMT_MOF.1(4).1
Refinement: The MAS Server shall restrict the ability to download applications to enrolled mobile devices that are compliant with MDM policies and assigned to a user in the application access group.

5.1.4.5 Specification of Management Functions (Server configuration of Agent) (FMT_SMF.1(1))

FMT_SMF.1(1).1
Refinement: The MDM Server shall be capable of communicating the following commands to the MDM Agent:
1. transition to the locked state, (MDF Function 8)
2. full wipe of protected data, (MDF Function 9)
3. unenroll from management,
4. install policies,

5. query connectivity status,
6. query the current version of the MD firmware/software
7. query the current version of the hardware model of the device
8. query the current version of installed mobile applications
9. import X.509v3 certificates into the Trust Anchor Database, (MDF Function 13)
10. install applications, (MDF Function 18)
11. update system software, (MDF Function 17)
12. remove applications, (MDF Function 16)
13. remove Enterprise applications, (MDF Function 19)

and the following commands to the MDM Agent:

*23. **[no other management functions***]

and the following MD configuration policies:

24. password policy:
    a. minimum password length
    b. minimum password complexity
    c. maximum password lifetime (MDF Function 1)
25. session locking policy:
    a. screen-lock enabled/disabled
    b. screen lock timeout c. number of authentication failures (MDF Function 2)
26. wireless networks (SSIDs) to which the MD may connect (MDF Function 6)
27. security policy for each wireless network:
    a. [***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 additional WLAN management functions***]*** (MDF Function 7)
28. application installation policy by
    [***a. specifying authorized application repository(s),***
    c. denying application installation], (MDF Function 10)
29. enable/disable policy for ***[*camera, microphone***]*** across MD, [***no other method***], (MDF Function 5)

and the following MD configuration policies:

[***33. enable/disable policy for [protocols supporting remote access] (MDF Function 23),***
34. enable/disable policy for developer modes (MDF Function 24),
35. enable policy for data-at rest protection (MDF Function 25),
36. enable policy for removable media's data-at-rest protection (MDF Function 26),
46. the unlock banner policy (MDF Function 36),
47. configure the auditable items (MDF Function 37),
48. enable/disable
[a. USB mass storage mode] (MDF Function 39),
49. enable/disable backup to [remote system] (MDF Function 40),
50. enable/disable
[a. Hotspot functionality authenticated by [no authentication],
b. USB tethering authenticated by [no authentication]] (MDF Function 41),
51. enable/disable location services:
a. across device
[c. no other method] (MDF Function 44),
53. [no other policies].

5.1.4.6 Specification of Management Functions (Server Configuration of Server) (FMT_SMF.1(2))

FMT_SMF.1(2).1

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

a. configure X.509v3 certificates for MDM Server use
b. configure the [***a number of devices***] allowed for enrollment
[d. configure the TOE unlock banner,
e. configure periodicity of the following commands to the agent: [
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].

5.1.4.7 Specification of Management Functions (MAS Server) (FMT_SMF.1(3))

FMT_SMF.1(3).1

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***].

5.1.4.8 Specification of Management Functions (FMT_SMF_EXT.3)

FMT_SMF_EXT.3.1

The MDM **Android** Agent shall be capable of interacting with the platform to perform the following functions:
[***b. administrator-provided device management functions in MDM PP***]
c. Import the certificates to be used for authentication of MDM Agent communications
d. [***no additional functions***].

FMT_SMF_EXT.3.2

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

5.1.4.9 Security Management Roles (FMT_SMR.1(1)) - Server

FMT_SMR.1(1).1

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

FMT_SMR.1(1).2

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

5.1.4.10 Security Management Roles (FMT_SMR.1(2)) - Agent

FMT_SMR.1(2).1

Refinement: The MAS Server shall maintain the roles administrator, MD user, enrolled mobile devices, application access groups, and **[Server primary administrator, Security configuration administrator, Device user group administrator, Auditor]**.

FMT_SMR.1(2).2

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

5.1.4.11 User Unenrollment Prevention (FMT_UNR_EXT.1)

FMT_UNR_EXT.1.1

The MDM **Android** Agent shall provide a mechanism to prevent users from unenrolling the mobile device from management.

5.1.5 Protection of the TSF (FPT)

5.1.5.1 Basic Internal TSF Data Transfer Protection (MDM Server) (FPT_ITT.1(1))

FPT_ITT.1(1).1

Refinement: The TSF shall protect all data from disclosure and modification through use of [***TLS***] when it is transferred between the MDM **Android** Agent and MDM Server.

5.1.5.2 Basic Internal TSF Data Transfer Protection (Distributed TOE) (FPT_ITT.1(2))

FPT_ITT.1(2).1

Refinement: The MDM **Android** Agent and MDM Server shall protect all data from disclosure and modification through use of [***TLS***] when it is transferred between separate parts of the TOE.

5.1.5.3 Basic Internal TSF Data Transfer Protection (MAS Server) (FPT_ITT.1(3))

FPT_ITT.1(3).1

Refinement: The TSF shall protect all data from disclosure and modification through use of [***TLS***] when it is transferred between the MDM **Android** Agent and MAS Server.

5.1.5.4 TSF Testing (FPT_TST_EXT.1) - Server

FPT_TST_EXT.1.1

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

FPT_TST_EXT.1.2

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

5.1.5.5 TSF Testing (FPT_TST_EXT.1(1)) - Agent

FPT_TST_EXT.1(1).1

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

5.1.5.6 Trusted Update (FPT_TUD_EXT.1)

FPT_TUD_EXT.1.1

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

FPT_TUD_EXT.1.2

The [***MDM Server***] shall provide Authorized Administrators the ability to initiate updates to MDM Server software.

FPT_TUD_EXT.1.3

The [***MDM Server***] shall provide a means to verify software updates to the MDM Server using a digital signature mechanism prior to installing those updates.

5.1.6 TOE access (FTA)

5.1.6.1 Default TOE Access Banners (FTA_TAB.1)

FTA_TAB.1.1

Before establishing a user session, the [***MDM Server***] shall 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) (FTP_ITC.1(1)) - Server

FTP_ITC.1(1).1

Refinement: The [***MDM Server***] shall use [***TLS***] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [***[no other capabilities]***] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

FTP_ITC.1(1).2

The TSF shall permit the MDM Server or other authorized IT entities to initiate communication via the trusted channel.

FTP_ITC.1(1).3

The TSF shall initiate communication via the trusted channel for ***[exporting audit events]***.

5.1.7.2 Inter-TSF Trusted Channel (Authorized IT Entities) (FTP_ITC.1(2)) – iOS Communications

FTP_ITC.1(2).1

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

FTP_ITC.1(2).2

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

FTP_ITC.1(2).3

The TSF shall initiate communication via the trusted channel for all communication between the MDM Server and the MDM **iOS** Agent and [***no other communication***].

5.1.7.3 Inter-TSF Trusted Channel (Authorized IT Entities) (FTP_ITC.1(3)) – MAS Server

FTP_ITC.1(3).1

Refinement: The [***MAS Server***] shall use [***TLS***] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [***[no other capabilities]***] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

FTP_ITC.1(3).2

The TSF shall permit the MAS Server or other authorized IT entities to initiate communication via the trusted channel.

FTP_ITC.1(3).3

The TSF shall initiate communication via the trusted channel for ***[exporting audit events]***.

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

FTP_TRP.1(1).1

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

FTP_TRP.1(1).2

Refinement: The [***MDM Server***] shall permit remote administrators to initiate communication via the trusted path.

FTP_TRP.1(1).3

Refinement: The [***MDM Server***] shall require the use of the trusted path for all remote administration actions.

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

FTP_TRP.1(2).1

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

FTP_TRP.1(2).2

Refinement: The [***MDM Server***] shall permit MD users to initiate communication via the trusted path.

FTP_TRP.1(2).3

Refinement: The [***MDM Server***] shall 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 3 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 fulfil 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

The Security audit function is designed to satisfy the following security functional requirements:

- FAU_ALT_EXT.1: The EMM Agent will alert the EMM Server after applying policies (to notify the Server as to whether or not its application attempt succeeded or failed) and for event triggered policy notifications. For example, if the Server were to configure an Agent with a policy to disable the device's camera upon entering a "geo-fence" (geographically defined area), then upon the Agent detecting that the device has physically moved within the geo-fenced area, it would alert the Server that the policy to disable the device's camera has become active.

  The EMM Server provides information about all of its actively managed devices and allows the administrator to query a device to obtain its current status. If the specified device does not have network connectivity, the EMM Server queues the query and delivers it when the device next contacts the Server.

- FAU_ALT_EXT.2: The EMM Server alerts administrators by displaying a "notifications" tab containing alerts for the administrator. Currently, the EMM Server displays alerts for changes in enrollment status (i.e., successful un/enrollment of devices), a failure of an Agent to apply policies, and Agent denial of a user attempt to install a disallowed mobile application (whether the Server disallows an application based upon a whitelist or blacklist). The EMM Server enforces no limit on the maximum number of queued messages, and queues messages for the administrator in the notifications tab.

- FAU_CRP_EXT.1: The EMM Server provides the administrator the ability to export configuration data for the mobile devices the Server manages in a CSV (Comma Separated Values) format.

- FAU_GEN.1(1)/FAU_GEN.1(2)/FAU_GEN.1(3): The EMM Server automatically generates audit records for all required events specified in the SFR without any additional administrator configuration. A complete list of the audit records generated are listed in the table below along with the included information (which includes the minimum set specified by the SFR). Each event in the TOE's audit log includes Log Data and Time, an Admin ID and Mobile IDs (if applicable), a Client IP (indicating the subject), an Event Category (type), an Event (indicates success or failure), a Severity, and additional information for specific events (indicated in the third column of the below table).

Requirement	Auditable Event	Additional Content
FAU_ALT_EXT.1	Type of alert. 1. successful application of Policies to a mobile device 2. event triggered policy notifications	Identity of EMM Agent that sent alert.
FAU_ALT_EXT.2	Type of alert. 1. change in enrollment status 2. failure to apply Policies to a mobile	Identity of EMM Agent that sent alert.

Requirement	Auditable Event	Additional Content
	device 3. denial of mobile application installation	
FAU_GEN.1	Start-up and shutdown of the MDM Server software. All administrative actions. Commands issued from the MDM Server to an EMM Agent.	No additional information.
FCS_CKM.1	Failure of the key generation activity.	No additional information.
FCS_HTTPS_EXT.1	Failure of the certificate validity check.	No additional information.
FCS_RBG_EXT.1	Failure of the randomization process.	No additional information.
FCS_TLSC_EXT.1	Failure to establish a TLS session. Failure to verify presented identifier.	Reason for failure. Presented identifier and reference identifier.
FCS_TLSS_EXT.1	Failure to establish a TLS session.	Reason for failure.
FIA_ENR_EXT.1.1	Failure of MD user authentication.	Presented credentials.
FIA_X509_EXT.1	Failure of X.509 certificate validation.	Reason for failure of validation.
FIA_X509_EXT.2	Failure to establish connection to determine revocation status.	No additional information
FMT_MOF.1(1)	Issuance of command to perform function. Change of policy settings.	Command sent and identity of EMM Agent recipient. Policy changed and value or full policy.
FMT_MOF.1(2)	Enrollment by a user.	Identity of user.
FMT_SMF.1(2)	Success or failure of function.	No additional information.
FMT_SMR_EXT.3	Success of failure of function	No additional information.
FPT_TST_EXT.1	Execution of this set of TSF self-tests. Detected integrity violations.	For integrity violations, the TSF code file that caused the integrity violation.
FPT_TUD_EXT.1	Initiation of update. Success or failure of update.	Version of update.
FTP_TAB.1	Change in banner setting.	No additional information.
FTP_TRP.1(1)	Initiation of the trusted channel. Termination of the trusted channel. Failures of the trusted path functions.	Identification of the claimed user identity.
FTP_TRP.1(2)	Initiation of the trusted channel. Termination of the trusted channel. Failures of the trusted path functions.	Identification of the claimed user identity.

- FAU_NET_EXT.1: The EMM Server component of the TOE provides the ability for an administrator to determine the connectivity status of any EMM Agent. Device synch normally occurs periodically, where an administrator configured the period. Device synchs can also be initiated by an administrator using the EMM Server web interface to cause an immediate check-in to ensure or determine the current connectivity status.
- FAU_SAR.1: Once logged into the EMM Server, an administrator can review all of the Server's audit records. The administrator can display audit records and filter the records displays based upon any of the available criteria.
- FAU_STG_EXT.1/FAU_STG_EXT.1(2): The EMM servers always store audit records locally in flat files stored on the TOE platform file system. The EMM Server provides administrators the capability to securely export EMM Console audit data through the server's administrative interface (WebUI) that is protected by an HTTPS trusted channel. The EMM Server and other server components provide administrators with a RDP (Remote Desktop Protocol) secured interface to access Windows log files generated by the server component's underlying Microsoft Platform.

- FAU_STG_EXT.2: The EMM Server secures its audit records by storing them in flat files protected by file access permissions enforced by the TOE Platform (Windows operating system) that preclude the ability to modify, insert, or delete a record (notwithstanding users with administrative rights on the TOE platform). Furthermore, the EMM Server only allows authenticated administrators access to display audit records, but provides no capability for an administrator to change those records.

6.2 Cryptographic support

The Cryptographic support function is designed to satisfy the following security functional requirements:

- FCS_CKM.1: Each EMM Server component uses its RSA BSAFE Crypto-J cryptographic module to generate asymmetric RSA and ECDSA keys as part of key establishment. For authentication, the EMM Server allows the administrator to import RSA and ECDSA certificates into each EMM Server component, as the components only generate keys during TLS key exchange.
- FCS_CKM.1(1): The EMM Agent relies upon the EMM Server for generation of RSA and ECDSA keypairs. During the EMM Agent's enrollment process, the EMM Client relies upon the EMM server to generate either an RSA or ECDSA keypair on behalf of the Agent, to send a CSR request to the CA, and then returns (through an HTTPS protected session) the newly generated private key and issued certificate (as a PFX/PKCS12 file) to the Agent.
- FCS_CKM.2: The EMM Server components support asymmetric key generation for key establishment as part of TLS and HTTPS. The EMM Server components allow the administrator to import authentication certificates (the Server components support both RSA and ECDSA certificates). The following table details which components act as TLS clients and servers as well as which ones generate ECDH keys used with ECDHE_* TLS cipher suites.

Server Component	Client/Server/Both	ECDH key gen?	RSA key gen?
AT Relay	Server	Yes	No (imported keys only)
Push Proxy	Server	Yes	No (imported keys only)
EMM Server	Both	Yes	No (imported keys only)
AT Server	Client	Yes	No (client only)
Push Server	Both	Yes	No (imported keys only)

  Each EMM Server component includes the RSA BSAFE Crypto-J library, utilized for asymmetric key generation as part of the different TLS cipher suites the Server supports. These cipher suites include RSA and ECDHE based mechanisms. The guidance specifies the utilization of asymmetric keys with 112-bits of security strength (RSA keys of 2048-bit or larger and ECDHE keys for curves P-256 or P-384).
- FCS_CKM.2(1): The EMM Agent relies upon its MDFPP evaluated platform (mobile device) for all cryptography including asymmetric key generation for authentication and key establishment (again the EMM Agent uses TLS/HTTPS for trusted channel connections). The evaluated platform can generate the asymmetric keys needed to support the ECDHE_* TLS ciphersuites.
- FCS_CKM_EXT.4: The EMM Server components clear keys (TLS and HTTPS session keys) from memory after those keys are no longer needed. Furthermore, the EMM Server components store their certificates (the only persistently stored keying material) on an internal hard drive in encrypted format, and when an administrator configures new certificates, the EMM Sever will directly overwrite the old keys with the new.
- FCS_CKM_EXT.4(1): The EMM Agent relies upon its platform to securely clear keys (TLS and HTTPS session keys) from memory when no longer needed as the EMM Agent utilizes platform provided TLS and key storage.
- FCS_COP.1(*): The EMM Server components use the RSA BASFE Crypto-J Software library version 6.2, which provides the algorithms noted in the table below (along with the NIST standards to which they comply). While the below algorithm certificates list the specific operational environments upon which

testing was performed, JAVA's virtual machine and "write once run anywhere" nature enables an expansive set of compatible or equivalent platforms.

SFR	RSA Crypto-J 6.2 CAVP Cert #
FCS_CKM.1 – Key Generation *RSA and ECDSA key gen*	
RSA 186-4: Key(gen) – 2048 bit	1663
ECDSA 186-4: Key(gen) P256, P-384	619
SHA-256	2701
FCS_CKM.2 – Key Agreement *FFC & ECC schemes in CVL; KARole Initiator and responder*	
CVL KAS ECC: EphemeralUnified, KaRole(initiator, responder) RSADP: vendor-affirmation	1024
FCS_COP.1(*)	
AES 128/256 CBC	3263
128/256 GCM	3263
RSA SigGen(2048), SigVer(2048)	1663
ECDSA PKG/PKV/SigGen/SigVer	619
SHA-1/256/384/512	2701
HMAC SHA-1/256/384	2062
FCS_RBG_EXT.1	
DRBG HMAC_DRBG (SHA-256) Seeded by Windows Platform	772

Table 4 EMM Server Components' Cryptographic Algorithms

As described above, both the EMM Server (which uses its RSA Crypto-J library) and the EMM Agent (which relies upon its evaluated platform) generate and verify RSA and ECDSA signatures, perform HMAC-SHA hashing, perform AES encryption and decryption, perform SHA hashing, establish TLS/HTTPS connections, generate IVs, and generate random data.

Both the Agent and Server utilize these cryptographic algorithms primarily during establishment of TLS/HTTPS connections (which requires signature generation and verification as part of peer authentication, hashing as part of the signatures for peer authentication and for HMAC integrity, HMAC for integrity of the trusted channel, AES for the confidentiality of the trusted channel, and RBGs to generate nonces and IVs). The Server also uses signature verification to ensure the authenticity of EMM Server software updates.

When using HMAC as part of TLS, both the Agent and Server utilize HMAC keys equal to the block size of the underlying hash algorithm. Thus, when employing HMAC-SHA-1, the TOE uses a 20-byte key to generate a 20-byte hash. Likewise, when employing HMAC-SHA-256 or HMAC-SHA-384, the TOE uses a 32 or 48-byte key when performing hashing using a block size of 64 or 128 bytes to produce a 32 or 48-byte hash, respectively.

In addition to its RSA Crypto-J library, the EMM Server utilizes its platform (Microsoft Windows Server 2012 and Server 2012 R2, VID 10520-2015 and CCID 2016.1052 respectively) for RDP export of audit records and Trusted Updates of the EMM itself while the Agent exclusively calls the evaluated Android

APIs provided by the underlying phone platform (see Section 1.4 for the specific devices and NIAP VID references).

The Server seeds its DRBG using the Windows BCryptGenRandom() function. Because we cannot test Microsoft's entropy implementation, we make an assumption of entropy regarding it (as required for any untestable third-party source) and assume that the output of BCryptGenRandom() contains at least 0.666 bits of entropy per bit of output. With at least 0.666 bits of entropy per bit of output, the Server appropriately seeds its DRBG with at least 256-bits of entropy.

- FCS_HTTPS_EXT.1: The EMM Server supports HTTPS and TLS in compliance with the requirements of the MDMPP20. When accepting incoming HTTPS connections from remote administrators, the EMM Server follows RFC 2818 and presents its server certificate. However, the EMM Server does not request that the remote administrator present a certificate (in other words, the EMM Server does not require TLS mutual/client authentication for remote administrators). Instead, the remote administrator authenticates to the EMM Server using a username and password, transmitted to the EMM Server after they have established the TLS session.
- FCS_IV_EXT.1: The EMM Server generates IVs for AES CBC using unpredictable (random) IVs drawn from the SHA-256 HMAC_DRBG (which meets the "unpredictable" requirement of SP 800-38A), and the Server uses AES CBC encryption for protection of the Server's private keys and user credentials. The EMM Server derives AES CBC and GCM IVs as part of the TLS handshake (which also meets the "unpredictable" and "non-repeating" requirements of SP 800-38A and SP 800-38D respectively).
- FCS_RBG_EXT.1: The EMM Server's RSA BSAFE Crypto-J Cryptographic library provides a SHA-256 HMAC_DRBG seeded by the underlying platform (Microsoft Windows Server). Specifically, the Server's cryptographic module seeds its DRBG using the Windows BCryptGenRandom() function. Because we cannot test Microsoft's entropy implementation, we make an assumption of entropy regarding it (as required for any untestable third-party source) and assume that the output of BCryptGenRandom() contains at least 0.666 bits of entropy per bit of output With at least 0.666 bits of entropy per bit of output, the Server appropriately seeds its DRBG with at least 256-bits of entropy
- FCS_RBG_EXT.1(1): The EMM Agent makes indirect use of the AES-256 CTR_DRBG belonging to its underlying platform for all random bit generation (indirectly using it when calling platform provided cryptographic APIs).
- FCS_STG_EXT.1/FCS_STG_EXT.2/FCS_STG_EXT.4: The EMM Server components encrypt their persistent keys (which consist exclusively of TLS/HTTPS certificates) by storing them encrypted with an AES-256 CBC key derived from a server secret. At no time does the Server store any plaintext keys on its hard drive (the only persistent memory the Server has). The Server does not store any ephemeral keys (e.g., TLS/HTTPS session keys). Each of the Agent's APKs store their keys in the platform provided keystorage (Android KeyStore) and then utilize those keys securely through the platform provided key storage.
- FCS_TLSC_EXT.1/FCS_TLSS_EXT.1: Both the EMM Server and Agent support TLS versions 1.1 and 1.2 and support the ciphersuites listed in section 5.1.2.22 and 5.1.2.23. All versions of the SSL protocol and older versions of the TLS protocol are refused by the EMM Server. Both the EMM Server and EMM Agent perform certificate checking in conformance with FIA_X509_EXT.1 and additionally perform hostname checking to ensure that either the expected hostname matches the certificate Common Name (when the EMM Agent verifies the EMM Server's certificate) or that the Distinguished Name (DN) in the presented certificate matches a DN in a database of valid, known DNs (when the EMM Server verifies the EMM Agent's certificate). When performing revocation checking, the TOE checks the peer's certificate against a CRL to determine if the certificate remains valid. Finally, the TOE will accept a TLS/HTTPS certificate as valid in the event that the Server cannot contact the revocation server. Neither the EMM Server nor the EMM Agent utilize certificate pinning. The EMM Agent supports only NIST curves secp256r1, secp384r1, and secp521r1 when using elliptic curve ciphers.

6.3 Identification and authentication

The identification and authentication function is designed to satisfy the following security functional requirements:

- FIA_ENR_EXT.1: During the enrollment process, the user enters a username, mobile ID, and password as well as the EMM Server's FQDN or IP into the EMM Client application running on their mobile device. The EMM Client then attempts to establish an HTTPS connection with the EMM Server. The EMM Server, having authenticated to the Client through presentation of its certificate during the TLS handshake, checks that the Client/User's credentials verify correctly. An administrator can configure the EMM Server to limit users to only enrolling between one and five mobile devices, and, assuming that the Agent/Client presents a valid username, mobile ID, and password (if not valid, the TOE will log the failure and reject the Client's enrollment attempt) and assuming that the User is within their quota of enrolled devices, the Server will generate a Client keypair and send to the issuing CA a Certificate Signing Request containing the public key, and upon receiving the CA issued certificate, the Server will return the newly generated private key and newly issued certificate (as a PCKS12 file) to the Client. The Server also records the Client's Distinguished Name (DN) so that the Server can verify that connecting mobile devices have both a valid certificate as well as a DN matching a DN in the Server's database. Once the enrollment process has completed, all subsequent connections from the EMM Client/Agent to the EMM Server occur through a mutually authenticated TLS session (in which the Client/Agent presents its certificate to the server).
- FIA_ENR_EXT.2: During enrollment the EMM Agent records the unique URL (FQDN or IP address) of the EMM Server for future communication purposes. This value is initially configured by the mobile device user when attempting to enroll the mobile device.
- FIA_UAU.1: The EMM Server requires that any user connecting to the Server authenticate by providing a username and password before providing any access to the connecting user. Put another way, an administrator cannot perform any actions at all (other than logging in) until the administrator successfully authenticates. Furthermore, the Server only allows remote administrators to connect via HTTPS to ensure confidentiality.
- FIA_X509_EXT.1/FIA_X509_EXT.2/ FIA_X509_EXT.1(1)/FIA_X509_EXT.2(1): Both the EMM Server and Agent validate and handle X.509 certificates in compliance with the MDMPP20/MSMAEP20 requirements. The Server and Agent use X.509 certificates only during TLS/HTTPS trusted channel establishment (for server and client/mutual authentication). Both the Server and Agent adhere to the MDMPP20/MSMAEP20 stipulated rules governing v3 extensions. The EMM Server and Agent use X.509v3 certificates for TLS authentication and will not establish a TLS session if the certificate presented by the peer is determined to be invalid.

  The TOE validates authentication certificates (including the full path) and checks their revocation station using CRLs. The TOE processes certificates presented during the TLS handshake by first checking the received certificate's validity period and appropriate key usage property. The TOE checks that it can construct a certificate path from the server's certificate through any intermediary CAs to a trusted root CA. If the TOE can successfully build the certificate path, then the TOE will next check the validity of the CA certificates (e.g., checking its validity dates and that the CA flag is present in the basic constraints section for all CA certs) in the chain. Assuming the TOE determines that all CA certificates in the chain are valid, the TOE will finally check the revocation status of the server's certificate. The TOE will not accept any certificate for which it cannot determine the validity and will reject the connection attempt.

  Both the Server and Agent will accept as valid any certificate for which the Server or Agent cannot reach the revocation server to check its status. The Server uses its Crypto-J library for X.509 certificate validity checking while the Agent uses the underlying phone to perform certificate validity checking of the server's certificate and certificate chain, but performs revocation checking itself (as the TOE obtains CRLs in a network optimized fashion).

  The EMM Client components receive their certificates during the 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). When the EMM Client components subsequently contact the EMM Server components, they will utilize the keys in the keystore. Each of the three EMM Client

components stores a single key in the Android keystore and does not attempt to store multiple keys. When a Client's keys expire, the user must un-enroll (which destroys the Client component certificates in Android's keystore) and re-enroll their device to obtain new certificates (which the components load again into the keystore).

The EMM Server components receive a certificate during the installation process, and each component has a single certificate for each TLS port enabled, thus each component offering a TLS connection (whether acting as a TLS server or client) always uses its single, configured certificate.

6.4 Security management

The Security management function is designed to satisfy the following security functional requirements:

Management Function **Status Markers:** M – Mandatory O - Optional I – Implemented	PP Requirement	Android Agent	Apple Agent
1. transition to the locked state, (MDF Function 8)	M	I	I
2. full wipe of protected data, (MDF Function 9)	M	I	I
3. unenroll from management,	M	I	I
4. install policies,	M	I	I
5. query connectivity status,	M	I	I
6. query the current version of the MD firmware/software	M	I	I
7. query the current version of the hardware model of the device	M	I	I
8. query the current version of installed mobile applications	M	I	I
9. import X.509v3 certificates into the Trust Anchor Database, (MDF Function 13)	M	I	-
10. install applications, (MDF Function 18)	M	I	I
11. update system software, (MDF Function 17)	M	I	I
12. remove applications, (MDF Function 16)	M	I	I
13. remove Enterprise applications, (MDF Function 19)	M	I	I
14. wipe Enterprise data, (MDF Function 28)	O	-	-
15. remove imported X.509v3 certificates and [selection: no other X.509v3 certificates, [assignment: list of other categories of X.509v3 certificates]] in the Trust Anchor Database, (MDF Function 14)	O	-	-
16. alert the administrator,	O	-	-
17. import keys/secrets into the secure key storage, (MDF Function 11)	O	-	-
18. destroy imported keys/secrets and [selection: no other keys/secrets, [assignment: list of other categories of keys/secrets]] in the secure key storage, (MDF Function 12)	O	-	-
19. read audit logs kept by the MD, (MDF Function 32)	O	-	-
20. retrieve MD-software integrity verification values, (MDF Function 38)	O	-	-
21. approve exceptions for sharing data between [selection: application processes, group of application processes], (MDF Function 42)	O	-	-
22. place applications into application process groups based on [assignment: application characteristics], (MDF Function 43)	O	-	-
23. [assignment: list of other management functions to be provided by the MD], no other management functions]	O	-	-

Management Function **Status Markers:** M – Mandatory O - Optional I – Implemented	PP Requirement	Android Agent	Apple Agent
24. password policy: a. minimum password length b. minimum password complexity c. maximum password lifetime (MDF Function 1)	M	I	I
25. session locking policy: a. screen-lock enabled/disabled b. screen lock timeout c. number of authentication failures (MDF Function 2)	M	I	I
26. wireless networks (SSIDs) to which the MD may connect (MDF Function 6)	O	I	-
27. security policy for each wireless network: a. [selection: specify the CA(s) from which the MD will accept WLAN authentication server certificate(s), specify the FQDN(s) of acceptable WLAN authentication server certificate(s)] b. ability to specify security type c. ability to specify authentication protocol d. specify the client credentials to be used for authentication e. [assignment: any additional WLAN management functions] (MDF Function 7)	M	I	I
28. application installation policy by [selection: a. specifying authorized application repository(s), b. specifying a set of allowed applications and versions (an application whitelist) c. denying application installation], (MDF Function 10)	M	I	I
29. enable/disable policy for [assignment: list of audio or visual collection devices] across MD, [selection: on a per-app basis, no other method], (MDF Function 5)	M	I - Camera & Microphone	I - Camera Only
30. enable/disable policy for the VPN protection across MD, [selection: on a per-app basis, no other method], (MDF Function 3)	O	-	I
31. enable/disable policy for [assignment: list of radios], (MDF Function 4)	O	-	-
32. enable/disable policy for data signaling over [assignment: list of externally accessible hardware ports], (MDF Function 22)	O	-	-
33. enable/disable policy for [assignment: list of protocols where the device acts as a server], (MDF Function 23)	O	I	-
34. enable/disable policy for developer modes, (MDF Function 24)	O	I	-
35. enable policy for data-at rest protection, (MDF Function 25)	O	-	-
36. enable policy for removable media's data-at-rest protection, (MDF Function 26)	O	I	-
37. enable/disable policy for local authentication bypass, (MDF Function 27)	O	-	-
38. the Bluetooth trusted channel policy: a. enable/disable the Discoverable mode (for BR/EDR) b. change the Bluetooth device name [selection: c. allow/disallow additional wireless technologies to be used with Bluetooth	O	-	-

Management Function **Status Markers:** M – Mandatory O - Optional I – Implemented	PP Requirement	Android Agent	Apple Agent
d. disable/enable Advertising (for LE) e. disable/enable the Connection mode f. disable/enable the Bluetooth services and/or profiles available on the device g. specify minimum level of security for each pairing h. configure allowable methods of Out of Band pairing i. no other Bluetooth configuration] (MDF Function 20)			
39. enable/disable policy for display notification in the locked state of [selection: a. email notifications, b. calendar appointments, c. contact associated with phone call notification, d. text message notification, e. other application-based notifications, f. none] (MDF Function 21)	O	-	-
40. policy for establishing a trusted channel or disallowing establishment if the MD cannot establish a connection to determine the validity of a certificate, (MDF Function 30)	O	-	I
41. enable/disable policy for the cellular protocols used to connect to cellular network base stations, (MDF Function 31)	O	-	-
42. policy for import and removal by applications of X.509v3 certificates in the Trust Anchor Database, (MDF Function 29)	O	-	-
43. [selection: certificate, public-key] used to validate digital signature on applications, (MDF Function 33)	O	-	-
44. policy for exceptions for shared use of keys/secrets by multiple applications, (MDF Function 34)	O	-	-
45. policy for exceptions for destruction of keys/secrets by applications that did not import the key/secret, (MDF Function 35)	O	-	-
46. the unlock banner policy, (MDF Function 36)	O	I	I
47. configure the auditable items (MDF Function 37)	O	I	-
48. enable/disable [selection: a. USB mass storage mode, b. USB data transfer without user authentication, c. USB data transfer without authentication of the connection system] (MDF Function 39)	O	I	-
49. enable/disable backup to [selection: locally connected system, remote system] (MDF Function 40)	O	I	I
50. enable/disable [selection: a. Hotspot functionality authenticated by [selection: pre-shared key, passcode, no authentication], b. USB tethering authenticated by [selection: pre-shared key, passcode, no authentication]] (MDF Function 41)	O	I	-
51. enable/disable location services: a. across device [selection: b. on a per-app basis, c. no other method] (MDF Function 44)	O	I	-

Management Function **Status Markers:** M – Mandatory O - Optional I – Implemented	PP Requirement	Android Agent	Apple Agent
52. enable/disable policy for user unenrollment	O	-	-

- FMT_MOF.1: The EMM Server provides authorized administrators (i.e., an administrator remotely logged into the EMM Server) the ability to perform the required functions specified in the SFR and the ability to apply policies that the EMM Agents enforce. Before authenticating to the EMM Server, an operator has no ability to perform any functions or to alter policies. The EMM Server also requires that any user attempting to enroll a mobile device authenticate to the Server (by providing a valid username and password, which the EMM Agent transmits to the Server through an HTTPS trusted channel).
- FMT_MOF.1(1)/FMT_MOF.1(3)/FMT_MOF.1(4): The EMM 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 assigned to the available management screens and associated functions. The administrator can define device groups to allow for the grouping of mobile devices having the same device management profile and app management profile. The table above identifies the management functions that can be configured by a device management profile and app management profile.
- 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 provides the mobile device user with a username and a password that will allow them to enroll their devices.
- FMT_SMF.1(1): The EMM server allows administrators to send commands and configure all required policies (as identified in FMT_SMF.1(1), which the Server then transmits to the EMM Agents, which apply and enforce (in conjunction with the mobile device itself) those policies. The table above identifies the management functions implemented for the Android and Apple agents.
- FMT_SMF.1(2): In addition to managing mobile devices, the EMM 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, manage the device registration process (enrolling specific devices and limiting the number of devices a user can enroll), and 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.
- FMT_SMF.1(3): Furthermore, in support of application hosting, the MDM server supports the configuration of application groups assigned to individual apps and devices. It also supports the ability to download applications for deployment.
- FMT_SMF_EXT.3/FMT_UNR_EXT.1: The EMM Agent component of the TOE is configured with an X.509v3 certificate suitable to facilitate secure communication with the EMM Server. This certificate is provisioned during device enrollment. The EMM Server can be configured to use an external Microsoft CA to sign CSRs from the EMM Agent during enrollment. Once secure communication is enabled and the device is enrolled, the EMM Agent accepts commands and policies from the enrolled EMM Server and implements those commands and policies (identified above). The administrator can configure (using the EMM Console) the EMM Agent to prevent the mobile phone's user from removing the Agent's administrative privileges, thus preventing the user from unenrolling the Agent. If an administrator has not restricted the mobile phone user's ability to remove the EMM Agent's administrative privileges, then the user can remove the EMM Agent's administrative privileges (unenrolling it from the EMM Server).

Finally, the administrator can forcibly unenroll the EMM Agent from EMM Server (and if the Agent will receive the unenrollment request when it has network connectivity to the EMM Server).

- FMT_SMR.1(1)/FMT_SMR.1(2): The EMM Server provides several different roles: server primary administrators, security configuration administrators, device user administrators, auditor, and MD users. Server primary administrators are administrators that have an administrative account on the underlying Microsoft Windows Server platform (i.e., the Windows administrator), log into Windows locally or through RDP, and are responsible for installation, install configuration, and monitoring of Windows/platform level audit logs. Security configuration administrators log into the EMM Server's HTTPS WebUI and are responsible for configuring the EMM Server's settings. Device user administrators also login through the EMM Server's HTTPS WebUI and are responsible for setting up accounts for mobile device users, inspecting the status of a given mobile device, and revoking/unenrolling a mobile device. Finally, auditors (who also login through the EMM Server's HTTPS WebUI) have permissions only to access the EMM Console's audit log. All Administrators (other than the server primary administrator) connect remotely to the Server via HTTPS (using a standard web browser) and must be authentication (providing a username and password) before gaining any access to the Server. The Server requires that administrator accounts be created for each administrator, and separates such administrators from MD users (unless an administrator has explicitly created a separate administrative account for the user). The Server allows MD users to enroll their mobile devices and thus allows MD users to have the EMM Server manage their mobile devices to secure organization data and access.

6.5 Protection of the TSF

The Protection of the TSF function is designed to satisfy the following security functional requirements:

- FPT_ITT.1(1)/FPT_ITT.1(2)/FPT_ITT.1(3): The SDS EMM utilizes TLS (with mutual/client authentication using configured X509 certificates) as the trusted channel to protect all data transmitted among its distributed parts including the EMM Server, Agents, and proxy components from disclosure and modification.
- FPT_TST_EXT.1: The EMM Server performs power-up tests to ensure correct operation. The EMM Server's Crypto-J library performs power-up Known Answer Tests for each of its cryptographic algorithms (including AES, RSA, ECDSA, SHA, HMAC-SHA) to ensure correct operations, and the EMM Server as a whole performs an startup integrity check of its executable code to ensure its integrity. Should the startup integrity check fail, the server component will log the error and attempt to continue loading.
- FPT_TST_EXT.1(1): The EMM Agent relies upon its platform to perform a test of its cryptographic algorithms upon power-up.
- FPT_TUD_EXT.1: The EMM Server provides a System page that displays the version of the EMM Server's software. To update the EMM Server's software, the administrator can (following the Administrator Guidance) obtain a software update, if one is available, and install the update. During the installation process, the EMM Server's platform will check the Microsoft AuthentiCode signature embedded in the update file itself.

6.6 TOE access

The TOE access function is designed to satisfy the following security functional requirements:

- FTA_TAB.1: An Administrator can configure the EMM Server to display an Administrator-specified advisory notice and consent warning message (in the form of a custom logo and/or custom login notification message) regarding use of the EMM Server. The logo and login notification are shown on the EMM Server login page.

6.7 Trusted path/channels

The Trusted path/channels function is designed to satisfy the following security functional requirements:

- FTP_ITC.1(1)/FTP_ITC.1(2)/FTP_ITC.1(3): The EMM Server uses TLS to secure communication with an external administrator (either through their browser or through an RDP session). Since the EMM Server provides the MAS server functionality, there is no additional communication paths for the MAS audit communications. The EMM server communicates with the Apple agent using TLS to secure the communication pathway.
- FTP_TRP.1(1): The EMM Server uses TLS/HTTPS as its trusted communication path for communications and remote Administrators must connect to the EMM Server using HTTPS (though a normal web browser) to securely administer the Server. The EMM provides no other mechanism or method beyond HTTPS for a remote Administrator to configure or access the EMM Server.
- FTP_TRP.1(2): Likewise, the EMM Server uses TLS and TLS/HTTPS as the trusted communication channels for all communications with MD users. MD users initiate the communication channel by logging into the EMM Client software (part of the agent) and thereafter all communications between the Agent (on behalf of the MD user) and the Server travel across the secure channel.