Samsung SDS Co., LTD Samsung SDS CellWe EMM (MDMPP11) Security Target

Version 0.6
2015/05/08

Prepared for:

Samsung SDS

123, Olympic-ro 35-gil, Songpa-gu, Seoul, Korea 138-240

Prepared By:

www.gossamersec.com

LIST OF TABLES

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is SDS MDM 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 letter placed at the end of the component. For example FDP_ACC.1a and FDP_ACC.1b indicate that the ST includes two iterations of the FDP_ACC.1 requirement, a and b.
    - 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 …").
- The NDPP uses an additional convention – the 'case' – which defines parts of an SFR that apply only when corresponding selections are made or some other identified conditions exist. Only the applicable cases are identified in this ST and they are identified using **bold** text.
- 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 Samsung SDS CellWe EMM (MDMPP11) Security Target

ST Version – Version 0.6

ST Date – 2015/05/08

1.2 TOE Reference

TOE Identification – Samsung SDS Co., LTD Samsung SDS CellWe EMM version 1.1

TOE Developer – Samsung SDS Co., LTD

Evaluation Sponsor – Samsung SDS Co., LTD

1.3 TOE Overview

The Target of Evaluation (TOE) Samsung SDS Co., LTD's Samsung SDS CellWe EMM.

The EMM Suite 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 1.7 and Tomcat 7.0 running on the Microsoft Windows Server 2008 R2 operating system through Windows Server 2012 R2. 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 6.1 cryptographic module as part of its software, and the EMM Server's Microsoft Windows platform includes SQL server 2008-2012 and an EJBCA certificate authority.

Samsung SDS provides the EMM Agent software for evaluated Samsung mobile devices (including the Galaxy S4, Note 3, S5, Note 4, and Galaxy Note Edge), and 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 software.

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

1) The EMM Server (version 1.1.0) installed upon the Microsoft Windows 2012 R2 operating system with Oracle JRE 1.7, SQL Server 2012, and EJBCA 4.0.16.
2) The EMM Client version 1.1.0 APKs (EMM Agent, PushAgent, EMM Agent Resource, Samsung SDS EMM) installed upon a Samsung Galaxy S5 running Android KitKat 4.4.2

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 two types of profiles

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

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 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 EMM Suite 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 an EJBCA certificate authority.

1.4.1.2 Logical Boundaries

This section summarizes the security functions provided by EMM Suite:

- 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 be configured to export the audit records to an external SYSLOG server utilizing TLS for protection of the records on the network. 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 the EMM Server to indicate (i.e., respond) 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 he 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 FIPS 140-2 certified algorithms for a wide range of cryptographic functions including: asymmetric key generation and establishment, encryption/decryption, cryptographic hashing and keyed-hash message authentication. These functions are supported with suitable random bit generation, initialization vector generation, secure key storage, and key and protected data destruction.

The primitive cryptographic functions are used to implement security communication protocols: TLS and HTTPS used for communication between the 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 to 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 with 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 Enterprise Mobility Management Administrator's Guide, v1.1.0, March 2015

Samsung SDS Enterprise Mobility Management User's Guide, Version 1.1.0, March 2015.

Samsung SDS Enterprise Mobility Management Installation Guide, Version 1.1.0, March 2015

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 3, July 2009.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 3, July 2009.
    - Part 3 Conformant
- Protection Profile for Mobile Device Management, Version 1.1, 7 March 2014 (MDMPP11)
- Package Claims:
    - Assurance Level: EAL 1 conformant

2.1 Conformance Rationale

The ST conforms to the MDMPP11. As explained previously, 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 MDMPP11 and this section reproduces only the corresponding Security Objectives for operational environment for reader convenience. The MDMPP11 offers additional information about the identified security objectives, but that has not been reproduced here and the MDMPP11 should be consulted if there is interest in that material.

In general, the MDMPP11 has defined Security Objectives appropriate for mobile device management and as such are applicable to the SDS MDM 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.

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.

4. Extended Components Definition

All of the extended requirements in this ST have been drawn from the MDMPP11. The MDMPP11 defines the following extended requirements and since they are not redefined in this ST the MDMPP11 should be consulted for more information in regard to those CC extensions.

Extended SFRs:

- FAU_ALT_EXT.1: Extended: Agent Alerts
- FAU_ALT_EXT.2: Extended: Server Alerts
- FAU_CRP_EXT.1: Extended: Support for Compliance Reporting of Mobile Device Configuration
- FAU_STG_EXT.1: Extended: External Audit Trail Storage
- FAU_STG_EXT.2: Extended: Audit Event Storage
- FCS_CKM_EXT.2(1): Cryptographic Key Storage (MDM Server)
- FCS_CKM_EXT.2(2): Cryptographic Key Storage (MDM Agent)
- FCS_CKM_EXT.4(1): Cryptographic Key Destruction
- FCS_CKM_EXT.4(2): Cryptographic Key Destruction
- FCS_HTTPS_EXT.1(1): Extended: HTTPS Implementation
- FCS_IV_EXT.1: Extended: Initialization Vector Generation
- FCS_RBG_EXT.1(1): Extended: Random Bit Generation
- FCS_RBG_EXT.1(2): Extended: Random Bit Generation
- FCS_STG_EXT.1: Encrypted Cryptographic Key Storage (MDM Server)
- FCS_TLS_EXT.1(1): Extended: TLS Implementation
- FCS_TLS_EXT.1(2): Extended: TLS Implementation
- FIA_ENR_EXT.1: Extended: Enrollment of Mobile Device into Management
- FIA_X509_EXT.1(1): Extended: X509 Validation
- FIA_X509_EXT.1(2): Extended: X509 Validation
- FIA_X509_EXT.2(1): Extended: X509 Authentication
- FIA_X509_EXT.2(2): Extended: X509 Authentication
- FMT_POL_EXT.1: Extended: Trusted Policy Update (MDM Agent)
- FPT_TST_EXT.1(1): TSF Testing
- FPT_TST_EXT.1(2): TSF Testing
- FPT_TUD_EXT.1: Extended: Trusted Update (MDM Server)

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 MDMPP11. The refinements and operations already performed in the MDMPP11 are not identified (e.g., highlighted) here, rather the requirements have been copied from the MDMPP11 and any residual operations have been completed herein. Of particular note, the MDMPP11 made a number of refinements and completed some of the SFR operations defined in the Common Criteria (CC) and that PP should be consulted to identify those changes if necessary.

The SARs are also drawn from the MDMPP11 which includes all the SARs for EAL 1. However, the SARs are effectively refined since requirement-specific 'Assurance Activities' are defined in the MDMPP11 that serve to ensure corresponding evaluations will yield more practical and consistent assurance than the EAL 1 assurance requirements alone. The MDMPP11 should be consulted for the assurance activity definitions.

5.1 TOE Security Functional Requirements

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

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

	FCS_STG_EXT.1: Encrypted Cryptographic Key Storage (MDM Server)
	FCS_TLS_EXT.1(1): Extended: TLS Implementation
	FCS_TLS_EXT.1(2): Extended: TLS Implementation
FIA: Identification and authentication	FIA_ENR_EXT.1: Extended: Enrollment of Mobile Device into Management
	FIA_UAU.1: Timing of Authentication
	FIA_X509_EXT.1(1): Extended: X509 Validation
	FIA_X509_EXT.1(2): Extended: X509 Validation
	FIA_X509_EXT.2(1): Extended: X509 Authentication
	FIA_X509_EXT.2(2): Extended: X509 Authentication
FMT: Security management	FMT_MOF.1(1): Management of functions in MDM Server
	FMT_MOF.1(2): Management of Enrollment function
	FMT_POL_EXT.1: Extended: Trusted Policy Update (MDM Agent)
	FMT_SMF.1(1): Specification of management functions (Server configuration of Agent)
	FMT_SMF.1(2): Specification of management functions (Agent configuration of platform)
	FMT_SMF.1(3): Specification of management functions (Server Configuration of Server)
	FMT_SMR.1: Security management roles
FPT: Protection of the TSF	FPT_ITT.1: Basic internal TSF data transfer protection
	FPT_TST_EXT.1(1): TSF Testing
	FPT_TST_EXT.1(2): TSF Testing
	FPT_TUD_EXT.1: Extended: Trusted Update (MDM Server)
FTA: TOE access	FTA_TAB.1: Default TOE Access Banners
FTP: Trusted path/channels	FTP_TRP.1: Trusted path for Remote Administration
	FTP_TRP.2: Trusted path for Enrollment

Table 1 TOE Security Functional Components

5.1.1 Security audit (FAU)

5.1.1.1 Extended: Agent Alerts (FAU_ALT_EXT.1)

FAU_ALT_EXT.1.1

The MDM Agent shall provide a 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. [***no other events***].

FAU_ALT_EXT.1.2

The MDM Server shall provide the ability to query for Agent network connectivity status.

5.1.1.2 Extended: Server Alerts (FAU_ALT_EXT.2)

FAU_ALT_EXT.2.1

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***].

5.1.1.3 Extended: 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***].

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

FAU_GEN.1(1).1

Refinement: The MDM Server shall be able to generate an MDM Server 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 7 of MDMPP11[1]; 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 7 of MDMPP11,
- [**no other audit relevant information**].

5.1.1.5 Audit Review (MDM Server) (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.

5.1.1.6 Extended: 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.

5.1.1.7 Extended: 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(1))

FCS_CKM.1(1).1

Refinement: The [***MDM Server***] shall generate asymmetric cryptographic keys used for key

[1] Omitting the auditable events identified in NIAP Technical Decision #33 (https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?td_id=35)

establishment in accordance with [***- NIST Special Publication 800-56A, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography' for finite field-based key establishment schemes,***
- NIST Special Publication 800-56A, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography' for elliptic curve-based key establishment schemes and implementing 'NIST curves' P-256, P-384 and [no other curves],
- NIST Special Publication 800-56B, 'Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography' for RSA-based key establishment schemes] and specified cryptographic key sizes equivalent to, or greater than, a symmetric key strength of 112 bits.

5.1.2.2 Cryptographic Key Generation (FCS_CKM.1(2))

FCS_CKM.1(2).1

The [***MDM Server***] shall generate asymmetric cryptographic keys used for authentication in accordance with a specified cryptographic key generation algorithm [***- FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.3 for RSA schemes,***
- FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.4 for ECDSA schemes and implementing 'NIST curves' P-256, P-384 and [no other curves]] and specified cryptographic key sizes equivalent to, or greater than, a symmetric key strength of 112 bits.

5.1.2.3 Cryptographic Key Generation (FCS_CKM.1(3))

FCS_CKM.1(3).1

Refinement: The [***MDM Agent platform***] shall generate asymmetric cryptographic keys used for key establishment in accordance with [***- NIST Special Publication 800-56A, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography' for finite field-based key establishment schemes,***
- NIST Special Publication 800-56A, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography' for elliptic curve-based key establishment schemes and implementing 'NIST curves' P-256, P-384 and [no other curves] (as defined in FIPS PUB 186-4, 'Digital Signature Standard'),
- NIST Special Publication 800-56B, 'Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography' for RSA-based key establishment schemes] and specified cryptographic key sizes equivalent to, or greater than, a symmetric key strength of 112 bits.

5.1.2.4 Cryptographic Key Generation (FCS_CKM.1(4))

FCS_CKM.1(4).1

The [***MDM Agent platform***] shall generate asymmetric cryptographic keys used for authentication in accordance with a specified cryptographic key generation algorithm [
- FIPS PUB 186-4, Digital Signature Standard (DSS), Appendix B.4 for ECDSA schemes and implementing NIST curves P-256, P-384 and [no other curves];
- ANSI X9.31-1998, Appendix A.2.4 Using AES for RSA schemes]
and specified cryptographic key sizes [equivalent to, or greater than, a symmetric key strength of 112 bits].

5.1.2.5 Cryptographic Key Storage (MDM Server) (FCS_CKM_EXT.2(1))

FCS_CKM_EXT.2(1).1

The [***MDM Server***] shall store persistent secrets and private keys when not in use, in [***as specified in FCS_STG_EXT.1***].

5.1.2.6 Cryptographic Key Storage (MDM Agent) (FCS_CKM_EXT.2(2))

FCS_CKM_EXT.2(2).1

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

5.1.2.7 Cryptographic Key Destruction (FCS_CKM_EXT.4(1))

FCS_CKM_EXT.4(1).1

The [***MDM Server***] shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

5.1.2.8 Cryptographic Key Destruction (FCS_CKM_EXT.4(2))

FCS_CKM_EXT.4(2).1

The [***MDM Agent platform***] shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

5.1.2.9 Cryptographic operation (Digital Signatures) (FCS_COP.1(1))

FCS_COP.1(1).1

Refinement: The [***MDM Server***] shall perform cryptographic signature services in accordance with the following specified cryptographic algorithms [***- RSA Digital Signature Algorithm (RSA) with a key size (modulus) of 2048 bits or greater that meets FIPS PUB 186-2 or FIPS PUB 186-4, 'Digital Signature Standard',***
- Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of 256 bits or greater] that meets FIPS PUB 186-4, 'Digital Signature Standard' with 'NIST curves' P-256, P-384 and [no other curves] (as defined in FIPS PUB 186-4, 'Digital Signature].

5.1.2.10 Cryptographic operation (Keyed-Hash Message Authentication) (FCS_COP.1(2))

FCS_COP.1(2).1

The [***MDM Server***] shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm HMAC-[***SHA-1, SHA-256, SHA-384, SHA-512***], key sizes [**256, 384, 512**], and message digest sizes [***160, 256, 384, 512***] bits that meet the following: FIPS Pub 198-1, 'The Keyed-Hash Message Authentication Code', and FIPS Pub 180-4, 'Secure Hash Standard.'

5.1.2.11 Cryptographic operation (Encryption and Decryption) (FCS_COP.1(3))

FCS_COP.1(3).1

The [***MDM Server***] shall perform encryption/decryption in accordance with a specified cryptographic algorithm [***- AES-CBC (as defined in NIST SP 800-38A) mode,***
- AES-GCM (as defined in NIST SP 800-38D)] and cryptographic key sizes [***128-bit, 256-bit***] key sizes.

5.1.2.12 Cryptographic operation (Hashing) (FCS_COP.1(4))

FCS_COP.1(4).1

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

5.1.2.13 Cryptographic operation (Digital Signatures) (FCS_COP.1(5))

FCS_COP.1(5).1

Refinement: The [***MDM Agent platform***] shall perform cryptographic signature services in accordance with the following specified cryptographic algorithms [***- RSA Digital Signature***

Algorithm (RSA) with a key size (modulus) of 2048 bits or greater that meets FIPS PUB 186-2 or FIPS PUB 186-4, 'Digital Signature Standard',
- Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of 256 bits or greater that meets FIPS PUB 186-4, 'Digital Signature Standard' with 'NIST curves' P-256, P-384 and [no other curves] (as defined in FIPS PUB 186-4, 'Digital Signature Standard')].

5.1.2.14 Cryptographic operation (Keyed-Hash Message Authentication) (FCS_COP.1(6))

FCS_COP.1(6).1

The [***MDM Agent 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 Cryptographic operation (Encryption and Decryption) (FCS_COP.1(7))

FCS_COP.1(7).1

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

5.1.2.16 Cryptographic operation (Hashing) (FCS_COP.1(8))

FCS_COP.1(8).1

The [***MDM Agent 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.

5.1.2.17 Extended: HTTPS Implementation (FCS_HTTPS_EXT.1(1))

FCS_HTTPS_EXT.1(1).1

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

FCS_HTTPS_EXT.1(1).2

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

5.1.2.18 Extended: Initialization Vector Generation (FCS_IV_EXT.1)

FCS_IV_EXT.1.1

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

5.1.2.19 Extended: Random Bit Generation (FCS_RBG_EXT.1(1))

FCS_RBG_EXT.1(1).1

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

FCS_RBG_EXT.1(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.

5.1.2.20 Extended: Random Bit Generation (FCS_RBG_EXT.1(2))

FCS_RBG_EXT.1(2).1

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

FCS_RBG_EXT.1(2).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.

5.1.2.21 Encrypted Cryptographic Key Storage (MDM Server) (FCS_STG_EXT.1)

FCS_STG_EXT.1.1

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

5.1.2.22 Extended: TLS Implementation (FCS_TLS_EXT.1(1))

FCS_TLS_EXT.1(1).1

The [***MDM Server***] shall implement one or more of the following protocols [***TLS 1.0 (RFC 2246), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)***] supporting the following ciphersuites:
Mandatory Ciphersuites: TLS_RSA_WITH_AES_128_CBC_SHA and
Optional Ciphersuites: [***TLS_RSA_WITH_AES_256_CBC_SHA,***
TLS_RSA_WITH_AES_128_CBC_SHA256,
TLS_RSA_WITH_AES_256_CBC_ SHA256,
TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256,
TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256,
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,
TLS_DHE_RSA_WITH_AES_128_CBC_SHA,
TLS_DHE_RSA_WITH_AES_256_CBC_SHA].

FCS_TLS_EXT.1(1).2

The [***MDM Server***] shall not establish a trusted channel if the distinguished name (DN) contained in a certificate does not match the expected DN for the peer.

5.1.2.23 Extended: TLS Implementation (FCS_TLS_EXT.1(2))

FCS_TLS_EXT.1(2).1

The [***MDM Agent platform***] shall implement one or more of the following protocols [***TLS 1.0 (RFC 2246), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)***] supporting the following ciphersuites:
Mandatory Ciphersuites: TLS_RSA_WITH_AES_128_CBC_SHA and
Optional Ciphersuites: [***TLS_RSA_WITH_AES_256_CBC_SHA,***
TLS_RSA_WITH_AES_128_CBC_SHA256,
TLS_RSA_WITH_AES_256_CBC_ SHA256,
TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256,
TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256,
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,
TLS_DHE_RSA_WITH_AES_128_CBC_SHA,
TLS_DHE_RSA_WITH_AES_256_CBC_SHA].

FCS_TLS_EXT.1(2).2

The [***MDM Agent platform***] shall not establish a trusted channel if the distinguished name (DN) contained in a certificate does not match the expected DN for the peer.

5.1.3 Identification and authentication (FIA)

5.1.3.1 Extended: 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***].

FIA_ENR_EXT.1.3

The MDM Agent shall record the DN of the MDM Server during the enrollment process.

5.1.3.2 Timing of Authentication (FIA_UAU.1)

FIA_UAU.1.1

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

5.1.3.3 Extended: X509 Validation (FIA_X509_EXT.1(1))

FIA_X509_EXT.1(1).1

The [***MDM Server***] shall validate certificates in accordance with the following rules:
- RFC 5280 certificate validation and certificate path validation.
- 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).
o Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 1 with oID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.

FIA_X509_EXT.1(1).2

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

5.1.3.4 Extended: X509 Validation (FIA_X509_EXT.1(2))

FIA_X509_EXT.1(2).1

The [***MDM Agent platform***] shall validate certificates in accordance with the following rules:
- RFC 5280 certificate validation and certificate path validation.
- 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).
o Client 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.

FIA_X509_EXT.1(2).2

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

5.1.3.5 Extended: X509 Authentication (FIA_X509_EXT.2(1))

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

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

FIA_X509_EXT.2(1).3
The [***MDM Server***] shall not establish a trusted communication channel if the peer certificate is deemed invalid.

FIA_X509_EXT.2(1).5
The [***MDM Server***] shall generate a Certificate Request Message as specified in RFC 2986 and be able to provide the following information in the request: public key, Common Name, organization, organizational Unit, and Country.

5.1.3.6 Extended: X509 Authentication (FIA_X509_EXT.2(2))

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

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

FIA_X509_EXT.2(2).3
The [***MDM Agent, MDM Agent platform***] shall not establish a trusted communication channel if the peer certificate is deemed invalid.

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(3)
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 Extended: Trusted Policy Update (MDM Agent) (FMT_POL_EXT.1)

FMT_POL_EXT.1.1
The MDM Agent shall report the successful installation of each Policy update to the MDM Server.

5.1.4.4 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,
2. full wipe of protected data,

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,
10. remove administrator-imported X.509v3 certificates and [***no other X.509v3 certificates***] in the Trust Anchor Database,

and the following commands to the MDM Agent:
[***no other management functions***]
and the following MD configuration Policies:
21. password Policy:
a. minimum password length
b. minimum password complexity
c. maximum password lifetime
22. session locking Policy:
a. screen-lock enabled/disabled
b. screen lock timeout
c. number of authentication failures
23. wireless networks (SSIDs) to which the MD may connect
24. 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**]
25. application installation Policy by [***b. specifying a set of allowed applications and versions (an application whitelist),***
c. denying application installation],
26. enable/disable Policy for [**camera, microphone**],
and the following MD configuration Policies:
[***no other Policies***].

5.1.4.5 Specification of management functions (Agent configuration of platform) (FMT_SMF.1(2))

FMT_SMF.1(2).1

Refinement: The MDM Agent shall be capable of interacting with the platform to perform the following functions:
a. Perform the functions listed in FMT_SMF.1(1) and the MDM configuration Policies listed in FMT_SMF.1(1)
b. Configure the certificate to be used for authentication of MDM Agent communications
c. [**no additional functions**].

5.1.4.6 Specification of management functions (Server Configuration of Server) (FMT_SMF.1(3))

FMT_SMF.1(3).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
c) [**no additional functions required to support SFRs**],
d) [***no other management functions***].

5.1.4.7 Security management roles (FMT_SMR.1)

FMT_SMR.1.1

Refinement: The MDM Server shall maintain the roles administrator, MD user, and [**no additional authorized identified roles**].

FMT_SMR.1.2

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

5.1.5 Protection of the TSF (FPT)

5.1.5.1 Basic internal TSF data transfer protection (FPT_ITT.1)

FPT_ITT.1.1

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

5.1.5.2 TSF Testing (FPT_TST_EXT.1(1))

FPT_TST_EXT.1(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 MDM Server.

FPT_TST_EXT.1(1).2

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

5.1.5.3 TSF Testing (FPT_TST_EXT.1(2))

FPT_TST_EXT.1(2).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 MDM Agent.

FPT_TST_EXT.1(2).2

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

5.1.5.4 Extended: Trusted Update (MDM Server) (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 Trusted path for Remote Administration (FTP_TRP.1)

FTP_TRP.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.1.2

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

FTP_TRP.1.3

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

5.1.7.2 Trusted path for Enrollment (FTP_TRP.2)

FTP_TRP.2.1

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

FTP_TRP.2.2

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

FTP_TRP.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 EAL 1 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 2 EAL 1 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 categorisation 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)..

- 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 Variable) format.

- FAU_GEN.1(1): 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 a 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 MDM Agent that sent alert.
FAU_ALT_EXT.2	Type of alert. 1. change in enrollment status 2. failure to apply Policies to a mobile device 3. denial of mobile application installation	Identity of MDM Agent that sent alert.

Requirement	Auditable Event	Additional Content
FAU_GEN.1	Start-up and shutdown of the MDM Server software. All administrative actions. Commands issued from the MDM Server to an MDM Agent.	No additional information.
FCS_CKM.1	Failure of the key generation activity.	No additional information.
FCS_RBG_EXT.1(1)	Failure of the randomization process.	No additional information.
FCS_TLS_EXT.1	Failure to establish a TLS session. Establishment/termination of a TLS session.	Reason for failure. Non-TOE endpoint of connection (IP address).
FIA_ENR_EXT.1.1	Failure of MD user authentication.	Presented credentials.
FIA_ENR_EXT.1.2	Failure of enrollment.	Reason for failure.
FIA_X509_EXT.1	Failure of X.509 certificate validation.	Reason for failure of validation.
FIA_X509_EXT.2	Generation of a Certificate Request Message.	Content of Certificate Request Message.
FMT_MOF.1(1)	Issuance of command to perform function. Change of policy settings.	Command sent and identity of MDM Agent recipient. Policy changed and value or full policy.
FMT_MOF.1(2)	Enrollment by a user	Identity of user.
FMT_SMF.1(3)	Success or failure of function.	No additional information.
FPT_ITT.1	Initiation of the trusted channel. Termination of the trusted channel. Failures of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
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_TRP.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.2	Initiation of the trusted channel. Termination of the trusted channel. Failures of the trusted path functions.	Identification of the claimed user identity.

- 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: The EMM Server always stores audit records locally in flat files stored on the TOE Platform file system. The server additional provides the capability to securely transmit audit data to a remote syslog server. The Server tunnels the syslog data using TLS (using stunnel) as the secure channel to ensure confidentiality and integrity.
- 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(1):The EMM Server components support asymmetric key generation for key establishment as part of TLS and HTTPS. The following table details which components act as TLS clients and servers as well as which ones generate DH, ECDH, or RSA keys used during DHE_*, ECDHE_*, and TLS_RSA_* TLS cipher suites.

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

The following tables specifically identify the “should”, “should not”, and “shall not” conditions from the publication along with an indication of how the TOE conforms to those conditions.

NIST SP800-56A Section Reference	“should”, “should not”, or “shall not”	Implemented?	Rationale for deviation
5.4	should	Yes	Not applicable
5.5.1.1	should	Yes	Not applicable
5.5.2	should	Yes	Not applicable
5.6.2	should	Yes	Not applicable
5.6.2.1	should	Yes	Not applicable
5.6.2.2	should	Yes	Not applicable
5.6.2.3	should	Yes	Not applicable
5.6.3.1	should	Yes	Not applicable
5.6.3.2.1	should	Yes	Not applicable
5.6.4.1	shall not	No	Not applicable
5.6.4.2	shall not	No	Not applicable
5.6.4.2	should	Yes	Not applicable
5.6.4.3	should (first occurrence)	Yes	Not applicable
5.6.4.3	should (second occurrence)	Yes	Not applicable
5.8	shall not (first occurrence)	No	Not applicable
5.8	shall not (second occurrence)	No	Not applicable
6	should (first occurrence)	Yes	Not applicable
6	should (second occurrence)	Yes	Not applicable
7	shall not (first occurrence)	No	Not applicable
7	shall not (second occurrence)	No	Not applicable
9	shall not	No	Not applicable

Table 3 NIST SP800-56A Conformance

NIST SP800-56B Section Reference	“should”, “should not”, or “shall not”	Implemented?	Rationale for deviation
5.6	Should	Yes	Not applicable
5.8	shall not	No	Not applicable
5.9	shall not (first occurrence)	No	Not applicable
5.9	shall not (second occurrence)	No	Not applicable
6.1	should not	No	Not applicable
6.1	should (first occurrence)	Yes	Not applicable
6.1	should (second occurrence)	Yes	Not applicable
6.1	should (third occurrence)	Yes	Not applicable
6.1	should (fourth occurrence)	Yes	Not applicable
6.1	shall not (first occurrence)	No	Not applicable
6.1	shall not (second occurrence)	No	Not applicable

NIST SP800-56B Section Reference	"should", "should not", or "shall not"	Implemented?	Rationale for deviation
6.2.3	Should	Yes	Not applicable
6.5.1	Should	Yes	Not applicable
6.5.2	Should	Yes	Not applicable
6.5.2.1	Should	Yes	Not applicable
6.6	shall not	No	Not applicable
7.1.2	Should	Yes	Not applicable
7.2.1.3	Should	Yes	Not applicable
7.2.1.3	should not	No	Not applicable
7.2.2.3	should (first occurrence)	Yes	Not applicable
7.2.2.3	should (second occurrence)	Yes	Not applicable
7.2.2.3	should (third occurrence)	Yes	Not applicable
7.2.2.3	should (fourth occurrence)	Yes	Not applicable
7.2.2.3	should not	No	Not applicable
7.2.2.3	shall not	No	Not applicable
7.2.3.3	should (first occurrence)	Yes	Not applicable
7.2.3.3	should (second occurrence)	Yes	Not applicable
7.2.3.3	should (third occurrence)	Yes	Not applicable
7.2.3.3	should (fourth occurrence)	Yes	Not applicable
7.2.3.3	should (fifth occurrence)	Yes	Not applicable
7.2.3.3	should not	No	Not applicable
8	Should	Yes	Not applicable
8.3.2	should not	No	Not applicable

Table 4 NIST SP800-56B Conformance

- Each EMM Server component includes the FIPS 140-2 Approved RSA BSAFE Crypto-J cryptographic module which the component utilizes for asymmetric key generation as part of the different TLS cipher suites the Server supports. These cipher suites include RSA, DHE, and ECDHE based mechanisms. The Server only generates asymmetric keys with 112-bits of security strength (RSA/DHE keys of 2048-bit or larger and ECDHE keys for curves P-256 or P-384).FCS_CKM.1(2): Each EMM Server component also uses its FIP 140-2 Approved RSA BSAFE Crypto-J cryptographic module to generate its own asymmetric RSA and ECDSA keypairs, in order to submit a CSR for certificate issuance. The Server only generates RSA keys with a modulus of 2048-bits or great and ECDSA keys using curves of P-256 and P-384, and the Server (like the Agent) creates signatures using SHA-256 hashing
- FCS_CKM.1(3): The EMM Agent relies upon its MDFPP evaluated platform (mobile device) for all cryptography including asymmetric key generation for 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 DHE_*, and ECDHE_* TLS ciphersuites.
- FCS_CKM.1(4): While the EMM Agent can rely upon the platform (mobile device) for generation of RSA and ECDSA key pairs, the EMM Agent does not generate any RSA or ECDSA key pairs to be used for authentication. Instead, as part of the EMM Agent's enrollment process, the EMM Agent relies upon the EMM Server to generate a keypair and to return the newly generated keys along with a corresponding certificate.
- FCS_CKM_EXT.2(1): The EMM server stores its keys locally by encrypting them with AES-256 CBC.

- FCS_CKM_EXT.2(2): The EMM Agent relies upon its evaluated platform to securely store certificates (and the contained private keys).
- FCS_CKM_EXT.4(1): 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(2): The EMM Agent relies upon its platform to securely clear keys (TLS and HTTPS session keys) from memory when no longer needed.
- FCS_COP.1: The EMM Server components use a FIPS 140-2 Approved cryptographic module (the RSA BASFE Crypto-J JSAFE and JCE Software Module version 6.1, CMVP certificate #2057), which provides the following algorithms (along with the NIST standards to which they comply). While the below algorithm certificates list the specific operational environments upon which testing was performed, the module's 140-2 security policy[2] lists over fifty different vendor-affirmed combinations of operating system and Java Runtime Environment (JRE) including version 6.0/7.0 JREs from Apple, Android, HP, Sun/Oracle and nearly all major operating systems. JAVA's virtual machine and "write once run anywhere" nature enables this expansive set of compatible or equivalent platforms.

Algorithm	NIST Standard	SFR Referenece	Cert#
AES 128/256 CBC, CCM, GCM, KW	FIPS 197, SP 800-38A/C/D/F	FCS_COP.1(1)	2249
CVL TLS KDF	SP 800-135	FCS_CKM.1(1)	39
DRBG Hash/HMAC/CTR	SP 800-90A	FCS_RBG_EXT.1(1)	273
ECDSA PKG/PKV/SigGen/SigVer	FIPS 186-4, SP 800-56A	FCS_CKM.1(1) FCS_CKM.1(2) FCS_COP.1(3)	357
HMAC SHA-1/256/384/512	FIPS 198-1 & 180-4	FCS_COP.1(4)	1378
RSA SIG(gen)/SIG(ver)/Key(gen)	FIPS 186-4, SP 800-56B	FCS_CKM.1(1) FCS_CKM.1(2) FCS_COP.1(3)	1154
SHS SHA-1/256/384/512	FIPS 180-4	FCS_COP.1(2)	1938

Table 5 EMM Server Components' Cryptographic Algorithms

As described above, the both EMM Server (which uses its FIP 140-2 Approved cryptographic module) and the EMM Agent (which relies upon its evaluated platform) to generate and verify RSA and ECDSA signatures, to perform HMAC-SHA hashing, to perform AES encryption and encryption, to perform SHA hashing, to establish TLS/HTTPS connections, to generate IVs, and to 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 for 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, the 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 to performing hashing using a block size of 64 or 128 bytes to produce a 32 or 48-byte hash, respectively.

For all cryptographic algorithms the Server calls the methods of its approved 140-2 validated cryptographic module, while the Agent calls the evaluated Android APIs provided by the underlying phone.

[2] Found here: http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp2057.pdf

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(1): The EMM Server supports HTTPS and TLS in compliance with the requirements of the MDMPP. 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 order words, the EMM Server does not require TLS mutual/client authentication). 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(1): The EMM Server's RSA BSAFE Crypto-J Cryptographic module 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(2): The EMM Agent makes use of the AES-256 CTR_DRBG belonging to its underlying platform for all random bit generation.
- FCS_STG_EXT.1: 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 an 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).
- FCS_TLS_EXT.1: Both the EMM Server and Agent support TLS versions 1.0, 1.1, and 1.2 and support the cipher suites listed in section 5.1.2.22 and 5.1.2.23. Both the MDM Server and MDM Agent perform certificate checking in conformance with FIA_X509_EXT.1 and additionally perform hostname checking to ensure that the 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.

  The different EMM Server components utilize different default TLS cipher suites as described in the following table:

EMM Server (Administrator Console)	EMM Application Tunnel and Push components
TLS_RSA_WITH_AES_128_CBC_SHA	TLS_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_128_CBC_SHA256	
TLS_RSA_WITH_AES_256_CBC_SHA	TLS_RSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_AES_256_CBC_ SHA256	
	TLS_DHE_RSA_WITH_AES_128_CBC_SHA
	TLS_DHE_RSA_WITH_AES_256_CBC_SHA

EMM Server (Administrator Console)	EMM Application Tunnel and Push components
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256	
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256	
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

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 into the EMM Client application running on their mobile device. The EMM Client already contains EMM Server's PQDN or IP fixed into its software (thus the Agent software differs for each organization deploying the EMM Suite) an 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 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 request that the Client generate keypairs and send the newly generated public key to the Server. The Server forwards the public key as part of a Certificate Signing Request, and upon receiving the CA issued certificate, the Server will return the certificate to the Client. The Server also records the Client's Distinguished Name so that the Server can verify that connecting mobile devices have both a valid ceritifate 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 occurs through a mutually authenticated TLS session (in which the Client/Agent presents its certificate to the server)..
- 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: Both the EMM Server and Agent validate and handle X.509 certificates in compliance with the MDMPP 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 MDMPP stipulated rules governing v3 extensions.

  The TOE validates authentication certificates (including the full path) and checks their revocation station using CRLs. The TOE processes certificates presenting during the TLS handshake by first checking the received certificate for validity, 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 accept any certificate for which it cannot determine the validity and 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 FIPS 140-2 cryptographic module for X.509 certificate validity checking while the Agent utilizes 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 key store (which stores the keys with permissions to only allow the applications themselves to access the keys). When then 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 key store 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:

- FMT_MOF.1: The EMM Server provides authorizations administrators (i.e., an administrator remotely logged into the EMM Server) the ability to perform the required functions specified in the SFR, 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_POL_EXT.1: As described before, the EMM Agent provides messages back to the EMM Server to report the success or failure of policy installation.
- FMT_SMF.1(1): The EMM server allows administrators to configure all MDMPP11 required policies, which the Server then transmits to the EMM Agents, which apply and enforce (in conjunction with the mobile device itself) those policies.
- FMT_SMF.1(2): The EMM Agent called the appropriate APIs offered by the evaluated mobile platform in order to apply and enforce the policies dictated by the Server. The EMM Agent can install the X.509 certificate that the Agent uses for trusted channel (TLS/HTTPS) communication with the Server and also implements logic to allow additional policy enforcement (geo-fencing, time based restrictions, etc.)
- FMT_SMF.1(3): The EMM Server allows authenticated administrators to configure the X.509 certificates that the Server will use to secure its TLS/HTTPS trusted channels and to configure the number of mobile devices that authenticated users can enroll.
- FMT_SMR.1: The EMM Server provides only two roles, administrators and MD users. Administrators connect remotely to the Server via HTTPS (using a standard web browser) and must first authentication (providing a username and password) before gaining any access to the Server. The Server requires that administrator accounts be created for each administrator account, and separates such administrators from MD users, who (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: The EMM Suite utilizes TLS as the trusted channel to protect all data transmitted between the Server and Agent (and vice-versa) from disclosure and modification.
- FPT_TST_EXT.1(1): The EMM Server performs power-up tests to ensure correct operation. The EMM Server's FIPS 140-2 Approved cryptographic module 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 performing an startup integrity check of its executable code to ensure its integrity.

- FPT_TST_EXT.1(2): The MDM 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 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 signature of the update during installation.

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 regarding use of the EMM Server.

6.7 Trusted path/channels

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

- FTP_TRP.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.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.