# Samsung SDS Co. Ltd. EMM and EMM Agent for Android 2.2.5 Security Target

Version 0.92
26 May 2023

***Prepared for:***

**Samsung SDS Co. Ltd.**

Samsung SDS Tower, 125,
Olympic-ro 35-gil, Songpa-gu,
Seoul, Korea 05510

***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 Samsung SDS EMM and EMM Agent for Android version 2.2.5 provided by Samsung SDS Co. Ltd. The TOE is being evaluated as a MDM Server and associated MDM Android Agent.

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.

***Common Acronyms and Terms***

The following acronyms or terms are used throughout this document:

- EMM – Enterprise Mobility Management
- MAS – Mobile Application Store
- MDM – Mobile Device Management
- MDMA10 – MOD_MDM_AGENT_V1.0
- MDMPP40 – PP_MDM_V4.0
- PKGTLS11 – PKG_TLS_V1.1
- SDS – Samsung Data Services

## 1.1 Security Target Reference

**ST Title –** Samsung SDS Co. Ltd. EMM and EMM Agent for Android Security Target

**ST Version** – Version 0.92

**ST Date** – 26 May 2023

## 1.2 TOE Reference

**TOE Identification** – Samsung SDS EMM and EMM Agent for Android version 2.2.5

**TOE Developer** – Samsung SDS Co. Ltd.

**Evaluation Sponsor** – Samsung SDS Co. Ltd.

## 1.3 TOE Overview

The Target of Evaluation (TOE) is Samsung SDS EMM and EMM Agent for Android version 2.2.5.

The SDS EMM provides centralized management of mobile devices and the EMM Agent for Android (installed on each device) enforces the policies of the Server on each device.

## 1.4 TOE Description

Samsung SDS offers the EMM as a software installation for Java 1.8 running on the Microsoft Windows Server 2016 or Windows Server 2019 operating system. Note that testing was conducted on Windows Server 2016 with Java 1.8 operating on hardware with an Intel(R) Xeon(R) CPU E3-1230. Once installed, the EMM allows administrators to configure policies for devices and also serves as a Mobile Application Store (MAS) server to serve configured applications to enrolled devices. Administrators connect securely to the EMM using a web browser (whether local to the Server itself or remote) and through the EMM's web interface can enroll, audit, lock, unlock, manage, and set policies for enrolled mobile devices. The EMM includes the RSA Crypto-J 6.3 cryptographic module as part of its software, and the EMM's Microsoft Windows platform includes SQL server 2008-2016, 2019 and a Microsoft Certificate Authority.

Note that one can install multiple EMM systems in order to allow the overall solution to scale the supported number of mobile devices as a High Availability (HA) option. In this case, the multiple EMM systems can operate concurrently and with the same policies and other information by sharing the same SQL database.

Samsung SDS provides the EMM Agent for Android software for evaluated Samsung mobile devices. The EMM Agent software, once installed and enrolled with the EMM, will apply and enforce administrator configured policies communicated through the EMM to the EMM Agent's running on the mobile devices. The scope of supported EMM Agent for Android devices for the evaluation will be limited by the set of devices evaluated on the NIAP PCL (refer to the following evaluations).

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

- Android 13 - https://www.niap-ccevs.org/Product/Compliant.cfm?PID=11342,
- Android 12 - https://www.niap-ccevs.org/Product/Compliant.cfm?PID=11307,
- Android 12 - https://www.niap-ccevs.org/Product/Compliant.cfm?PID=11228, and
- Android 11 - https://www.niap-ccevs.org/Product/Compliant.cfm?PID=11211.

While Samsung SDS does not provide an iOS agent, the EMM is designed to work with the iOS agents developed and evaluated by Apple. Since the iOS agents are evaluated as part of the Apple iOS evaluations, the EMM was tested only to ensure it can manage those devices, but the agent's behavior on those devices was not otherwise tested. The support is limited by the set of devices evaluated on the NIAP PCL (ref.

- iOS 15 - https://www.niap-ccevs.org/Product/Compliant.cfm?PID=11237, and
- iOS 16 - https://www.niap-ccevs.org/Product/PINE.cfm (in progress - VID 11349)

### 1.4.1 TOE Architecture

The EMM actually consists of the following different servers (these components are referred to collectively as EMM throughout subsequent sections of this document):

1. EMM Server – This is the main server running to which remote administrators connect. The EMM Server bears responsibility for all logic needed to manage mobile devices and to serve configured applications as a MAS server. When multiple EMM Servers are configured in a single operational environment, they are not directly associated or communicating with each other, but rather can simply share the same SQL database. The EMM Server is installed as a single component, but includes an integrated Tomcat server to implement the administrator Web User Interface and also an integrated Push server implementation (PUSH_SA) used to communicate with the other EMM server components (Push and AppTunnel).
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. Each Push Server is installed as a single component, but includes multiple internal modules (DCM/PS/SCM/ECM/ICM) to implement its full range of communication channels among EMM components and agents.
    a. Push Proxy – This is an alternate optional deployment of the Push Server that serves to relay or proxy messages between the mobile devices and Push Server. This deployment would normally be used to accommodate network architectures with DMZs. Each Push Proxy is installed as a single alternate deployment of the Push Server and also includes multiple internal modules (DPP/PPP/EPP) to implement its full range of communication channels among EMM components and agents in its proxy role.
3. AppTunnel (AT) Server – this server accepts connections from the EMM Agent (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. Each AT Server is installed as a single component and includes only a single module to implement its role of managing secure channels for apps.
    a. AT Relay – This is an alternate optional deployment of the AppTunnel Server that serves to relay or proxy messages between the mobile devices and AT Server. This deployment would normally be used to accommodate network architectures with DMZs. Each AT Relay is installed as a single alternate deployment of the AT Server and also includes only a single module to implement its role of relaying secure channels for apps.

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

The minimum deployment for an EMM is an EMM Server, a Push Server, and an AT Server. While each EMM Server is paired with an AT Server, multiple Push Servers can optionally be configured to operate with a single EMM Server. Also optionally, a Push Proxy can be configured for each Push Server and an AT Relay can be configured for the AT Server. The majority of EMM security functions are implemented in the EMM Server while the other server components are primarily responsible for secure communications between the EMM Server and the enrolled device agents (see Appendix B. Requirement Allocation).

The EMM Agent for Android consists of two different components on evaluated Android platforms (these components are referred to collectively as EMM Agent throughout subsequent sections of this document):

1. The “EMM Agent” – 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. The EMM Agent presents the UI to allow users to start the enrollment process and, once enrolled, to log in and log out. This component also 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 enforces the policies of the Server.

2. 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 TOE includes a single component on evaluated iOS platforms:

1. The EMM Client – this iOS application provides a user interface to allow the user to enroll their phone with their organization's SDS EMM. The application relies upon the evaluated, embedded Apple agent for all agent functionality. As such, this component doesn't provide any security functions.

Figure 1 depicts a simple TOE architecture diagram where in practice many alternatives are possible given the available components – the TOE components are identified in green boxes. Examples of the various communication channels among components, depending on specific deployment, are identified in the installation and administrator guidance documents.

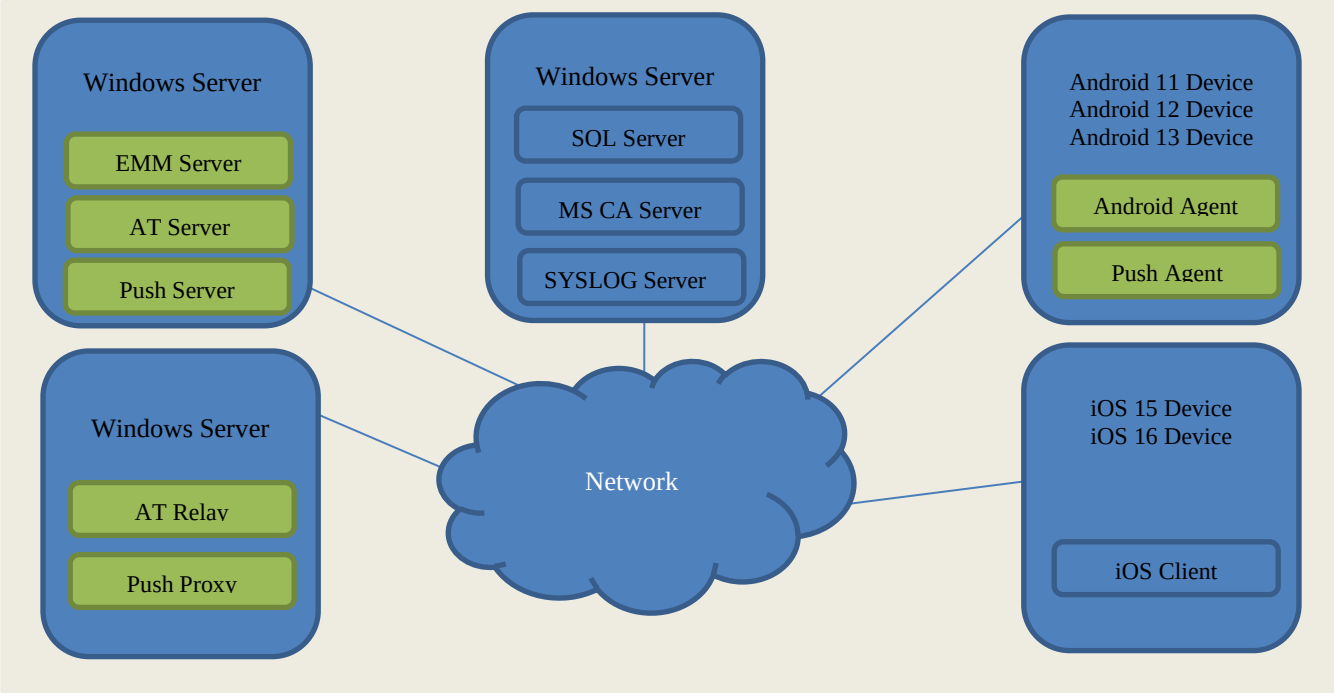


**Figure 1 Simple TOE Architecture**

#### 1.4.1.1 Physical Boundaries

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

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

#### 1.4.1.2 Logical Boundaries

This section summarizes the security functions provided by the Samsung SDS Co. Ltd. EMM and EMM Agent for Android:

- 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 can generate and store audit records for security-relevant events as they occur. These events are stored and protected by the EMM and can be reviewed by an authorized administrator. The EMM 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 protects the exported audit records using TLS (as part of HTTPS and RDP). The EMM 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 when it has been enrolled and when it applies policies successfully. The EMM 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 and EMM Agent both include or 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 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 using the EMM Agent as well as an administrator logging on to manage the EMM configuration, MDM policies for mobile devices, etc.

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

##### 1.4.1.2.4 Security management

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

The EMM 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 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.

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

#### 1.4.1.2.5 Protection of the TSF

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

Both the EMM 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 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 uses TLS/HTTPS to secure communication channels between its distributed components 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 and EMM Agent.

### 1.4.2 TOE Documentation

- Samsung EMM Administrator’s Guide, Solution version 2.2.5, January 2023
- Samsung SDS EMM Installation Guide, Solution version 2.2.5, January 2023
- Samsung SDS EMM Configuration Guide for IPsec settings in Microsoft Windows Server 2016/2019 for Common Criteria Evaluation, Solution version 2.2.5, 27th January 2023

# 2. Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, April 2017.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1, Revision 5, April 2017.
    - Part 3 Conformant
- PP-Configuration for Mobile Device Management (MDM) and MDM Agents, Version 1.0, January 27, 2020 (CFG_MDM-MDM_AGENT_V1.0)
    - Base PP: Protection Profile for Mobile Device Management, Version 4.0, 2019-04-25 (PP_MDM_V4.0 or MDMPP40) with the following technical decisions: TD0438, TD0461, TD0462, TD0479, TD0552, TD0594, TD0600, TD0616, TD0629, TD0641, and TD0650.
    - Module: PP-Module for MDM Agents, Version 1.0, 2019-04-25 (MOD_MDM_AGENT_V1.0 or MDMA10) TD0491, TD0497, TD0600, TD0650, TD0660, and TD0673.
    - Functional Package for Transport Layer Security (TLS), Version 1.1, 1 March 2019 (PKG_TLS_V1.1 or PKGTLS11) with the following technical decisions: TD0442, TD0469, TD0499, TD0513, TD0588, TD0726, and TD0739.

## 2.1 Conformance Rationale

The ST conforms to the combination of MDMPP40, MDMA10, and PKGTLS11 as identified above. The security problem definition, security objectives, and security requirements have been drawn from this combination of PP, Module, and Functional Package.

# 3. Security Objectives

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

In general, the MDMPP40/MDMA10/PKGTLS11 documents have defined Security Objectives appropriate for an MDM server and corresponding MDM agents and as such are applicable to the Samsung SDS EMM and EMM Agent for Android TOE.

## 3.1 Security Objectives for the Operational Environment

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

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

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

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

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

**OE.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 MDMPP40/MDMA10/PKGTLS11 documents (with minor typographical or formatting issues corrected). The MDMPP40/MDMA10/PKGTLS11 documents define the following extended requirements and since they are not redefined in this ST the MDMPP40/MDMA10/PKGTLS11 documents should be consulted for more information in regard to those CC extensions.

**Extended SFRs:**

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

# 5. Security Requirements

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

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

The SARs are also drawn from the MDMPP40/MDMA10/PKGTLS11 documents. However, the SARs are effectively refined since requirement-specific 'Assurance Activities' are defined in the MDMPP40/MDMA10/PKGTLS11 documents. The MDMPP40/MDMA10/PKGTLS11 documents should be consulted for the assurance activity definitions.

## 5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by Samsung SDS EMM and EMM Agent for Android TOE.

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

| Requirement Class | Requirement Component |
|---|---|
| | PKGTLS11:FCS_TLSS_EXT.2: TLS Server Support for Mutual Authentication |
| | PKGTLS11:FCS_TLSS_EXT.4: TLS Server Support for Renegotiation |
| **FIA: Identification and authentication** | MDMPP40:FIA_ENR_EXT.1: Enrollment of Mobile Device into Management |
| | MDMA10:FIA_ENR_EXT.2: Agent Enrollment of Mobile Device into Management |
| | MDMPP40:FIA_UAU.1: Timing of Authentication |
| | MDMPP40:FIA_X509_EXT.1(1): X.509 Certificate Validation |
| | MDMPP40:FIA_X509_EXT.1(2): X.509 Certificate Validation |
| | MDMPP40:FIA_X509_EXT.2: X.509 Certificate Authentication |
| | MDMPP40:FIA_X509_EXT.5: X.509 Unique Certificate |
| **FMT: Security management** | MDMPP40:FMT_MOF.1(1): Management of Functions Behavior |
| | MDMPP40:FMT_MOF.1(2): Management of Functions Behavior (Enrollment) |
| | MDMPP40:FMT_MOF.1(3): Management of Functions in (MAS Server Downloads) |
| | MDMPP40:FMT_POL_EXT.1: Trusted Policy Update |
| | MDMA10:FMT_POL_EXT.2: Agent Trusted Policy Update |
| | MDMPP40:FMT_SMF.1(1): Specification of Management Functions (Server configuration of Agent) |
| | MDMPP40:FMT_SMF.1(2): Specification of Management Functions (Server Configuration of Server) |
| | MDMPP40:FMT_SMF.1(3): Specification of Management Functions (MAS Server) |
| | MDMA10:FMT_SMF_EXT.4: Specification of Management Functions |
| | MDMPP40:FMT_SMR.1(1): Security Management Roles |
| | MDMPP40:FMT_SMR.1(2): Security Management Roles (MAS Server) |
| | MDMA10:FMT_UNR_EXT.1: User Unenrollment Prevention |
| **FPT: Protection of the TSF** | MDMPP40:FPT_API_EXT.1: Use of Supported Services and APIs |
| | MDMPP40:FPT_ITT.1(1): Internal TOE TSF Data Transfer |
| | MDMPP40:FPT_ITT.1(2): Internal TOE TSF Data Transfer (MDM Agent) |
| | MDMPP40:FPT_LIB_EXT.1: Use of Third Party Libraries |
| | MDMPP40:FPT_TST_EXT.1: Functionality Testing |
| | MDMPP40:FPT_TUD_EXT.1: Trusted Update |
| **FTA: TOE access** | MDMPP40:FTA_TAB.1: Default TOE Access Banners |
| **FTP: Trusted path/channels** | MDMPP40:FTP_ITC.1(1): Inter-TSF Trusted Channel (Authorized IT Entities) |
| | MDMPP40:FTP_ITC.1(2): Inter-TSF Trusted Channel (MDM Agent) |
| | MDMPP40:FTP_ITC_EXT.1: Trusted Channel |
| | MDMPP40:FTP_TRP.1(1): Trusted Path (for Remote Administration) |
| | MDMPP40:FTP_TRP.1(2): Trusted Path (for Enrollment) |

**Table 1 TOE Security Functional Components**

### 5.1.1 Security audit (FAU)

#### 5.1.1.1 Server Alerts (MDMPP40:FAU_ALT_EXT.1)

**MDMPP40:FAU_ALT_EXT.1.1**

The TSF shall alert the administrators in the event of any of the following: a. Change in enrollment status b. Failure to apply policies to a mobile device c. [***no other events***].

#### 5.1.1.2 Agent Alerts (MDMA10:FAU_ALT_EXT.2)

**MDMA10:FAU_ALT_EXT.2.1**

The MDM Agent shall provide an alert via the trusted channel to the MDM Server in the event of any of the following audit events:

successful application of policies to a mobile device,
[***receiving, generating***] periodic reachability events,
[***change in enrollment state,***
***failure to install an application from the MAS Server,***
***failure to update an application from the MAS Server***].

**MDMA10:FAU_ALT_EXT.2.2**

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

#### 5.1.1.3 Audit Data Generation (MDMPP40:FAU_GEN.1(1))

**MDMPP40:FAU_GEN.1.1(1)**

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

a. All administrative actions
b. [***Commands issued to the MDM Agent***]
c. Specifically defined auditable events listed in **Table 2**
d. [***start up and shut down of the MDM system***]. (TD0629 applied)

| Requirement | Auditable Events | Additional Content |
|---|---|---|
| **MDMPP40:FAU_ALT_EXT.1** | Type of alert. | Identity of Mobile Device that sent alert. |
| **MDMPP40:FAU_GEN.1(1)** | None. | |
| **MDMPP40:FAU_GEN.1(2)** | None. | |
| **MDMPP40:FAU_NET_EXT.1** | None. | |
| **MDMPP40:FAU_SAR.1** | None. | |
| **MDMPP40:FAU_STG_EXT.1** | None. | |
| **MDMPP40:FAU_STG_EXT.2** | None. | |
| **MDMPP40:FCS_CKM.1** | [***None***] | No additional information. |
| **MDMPP40:FCS_CKM.2** | None. | |
| **MDMPP40:FCS_CKM_EXT.4** | None. | |
| **MDMPP40:FCS_COP.1(1)** | None. | |
| **MDMPP40:FCS_COP.1(2)** | None. | |
| **MDMPP40:FCS_COP.1(3)** | None. | |
| **MDMPP40:FCS_COP.1(4)** | None. | |
| **MDMPP40:FCS_HTTPS_EXT.1** | Failure of the certificate validity check. | Issuer Name and Subject Name of certificate.<br>[***no additional information***] |
| **MDMPP40:FCS_IV_EXT.1** | None. | |
| **MDMPP40:FCS_RBG_EXT.1** | Failure of the randomization process. | No additional information. |
| **MDMPP40:FCS_STG_EXT.1** | None. | |
| **MDMPP40:FCS_STG_EXT.2** | None. | |
| **PKGTLS11:FCS_TLS_EXT.1** | None. | None |
| **PKGTLS11:FCS_TLSC_EXT.1** | Failure to establish a TLS session.<br>Failure to verify presented identifier. | Reason for failure.<br>Presented identifier and reference identifier. |
| **PKGTLS11:FCS_TLSC_EXT.2** | None. | None |
| **PKGTLS11:FCS_TLSC_EXT.5** | None. | None |
| **PKGTLS11:FCS_TLSS_EXT.1** | Failure to establish a TLS session. | Reason for failure. |
| **PKGTLS11:FCS_TLSS_EXT.2** | None. | None |
| **PKGTLS11:FCS_TLSS_EXT.4** | None. | None |
| **MDMPP40:FIA_ENR_EXT.1** | Failure of MD user authentication. | Presented username. |
| **MDMPP40:FIA_UAU.1** | None. | |

| Requirement | Auditable Events | Additional Content |
|---|---|---|
| **MDMPP40:FIA_X509_EXT.1(1)** | Failure to validate X.509 certificate. | Reason for failure. |
| **MDMPP40:FIA_X509_EXT.1(2)** | Failure to validate X.509 certificate. | Reason for failure. |
| **MDMPP40:FIA_X509_EXT.2** | Failure to establish connection to determine revocation status. | No additional information. |
| **MDMPP40:FIA_X509_EXT.5** | None. | |
| **MDMPP40:FMT_MOF.1(1)** | Issuance of command to perform function.<br>Change of policy settings. | Command sent and identity of MDM Agent recipient(s).<br>Policy changed and value or full policy. |
| **MDMPP40:FMT_MOF.1(2)** | Enrollment by a user. | Identity of user. |
| **MDMPP40:FMT_MOF.1(3)** | None. | |
| **MDMPP40:FMT_POL_EXT.1** | None. | |
| **MDMPP40:FMT_SMF.1(1)** | None. | |
| **MDMPP40:FMT_SMF.1(2)** | Success or failure of function. | No additional information. |
| **MDMPP40:FMT_SMF.1(3)** | None. | |
| **MDMPP40:FMT_SMR.1(1)** | None. | |
| **MDMPP40:FMT_SMR.1(2)** | None. | |
| **MDMPP40:FPT_API_EXT.1** | None. | |
| **MDMPP40:FPT_ITT.1(1)** | Initiation and termination of the trusted channel. | Trusted channel protocol. Identity of initiator and recipient. |
| **MDMPP40:FPT_ITT.1(2)** | Initiation and termination of the trusted channel. | Trusted channel protocol. Identity of initiator and recipient. |
| **MDMPP40:FPT_LIB_EXT.1** | None. | |
| **MDMPP40:FPT_TST_EXT.1** | Initiation of self-test.<br>Failure of self-test.<br>Detected integrity violation. | Algorithm that caused failure.<br>The TSF code file that caused the integrity violation. |
| **MDMPP40:FPT_TUD_EXT.1** | Success or failure of signature verification. | No additional information. |
| **MDMPP40:FTA_TAB.1** | Change in banner setting. | No additional information. |
| **MDMPP40:FTP_ITC.1(1)** | Initiation and termination of the trusted channel. | Trusted channel protocol. Non-TOE endpoint of connection |
| **MDMPP40:FTP_ITC.1(2)** | Initiation and termination of the trusted channel. | Trusted channel protocol. Non-TOE endpoint of connection. |
| **MDMPP40:FTP_ITC_EXT.1** | None. | |
| **MDMPP40:FTP_TRP.1(1)** | Initiation and termination of the trusted channel. | Trusted channel protocol. Identity of administrator. |
| **MDMPP40:FTP_TRP.1(2)** | Initiation and termination of the trusted channel. | Trusted channel protocol. |

**Table 2 MDM Server Auditable Events**

**MDMPP40:FAU_GEN.1.2(1)**

The TSF shall record within each TSF audit record at least the following information: date and time of the event; type of event; subject identity; (if relevant) the outcome (success or failure) of the event additional information in **Table 2**; [**no other audit relevant information**].

#### 5.1.1.4 Audit Data Generation (MDMA10:FAU_GEN.1(2))

**MDMA10:FAU_GEN.1.1(2)**

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

a. Startup and shutdown of the MDM Agent;

b. All auditable events for not specified level of audit; and
c. MDM policy updated, any modification commanded by the MDM Server, specifically defined auditable events listed in **Table 3**, and [***no other events***]. (TD0660 applied)

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

**Table 3 MDM Android Agent Auditable Events**

**MDMA10:FAU_GEN.1.2(2)**

Refinement: The [***TSF***] shall record within each MDM Agent audit record at least the following information:

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

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

**MDMPP40:FAU_GEN.1.1(2)**

Refinement: The MAS Server shall be able to generate an audit record of the following auditable events:

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

**MDMPP40:FAU_GEN.1.2(2)**

Refinement: The [***MAS Server***] shall record within each TSF audit record at least the following

information: date and time of the event, type of event, mobile device identity, [**no other audit relevant information**].

#### 5.1.1.6 Network Reachability Review (MDMPP40:FAU_NET_EXT.1)

**MDMPP40:FAU_NET_EXT.1.1**

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

#### 5.1.1.7 Audit Review (MDMPP40:FAU_SAR.1)

**MDMPP40:FAU_SAR.1.1**

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

**MDMPP40:FAU_SAR.1.2**

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

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

**MDMA10:FAU_SEL.1.1(2)**

Refinement: The TSF shall [***implement functionality***] to select the set of events to be audited from the set of all auditable events based on the following attributes: a. event type; b. success of auditable security events, failure of auditable security events, [**no other attributes**].

#### 5.1.1.9 External Trail Storage (MDMPP40:FAU_STG_EXT.1)

**MDMPP40:FAU_STG_EXT.1.1**

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

#### 5.1.1.10 Audit Event Storage (MDMPP40:FAU_STG_EXT.2)

**MDMPP40:FAU_STG_EXT.2.1**

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

### 5.1.2 Cryptographic support (FCS)

#### 5.1.2.1 Cryptographic Key Generation (MDMPP40:FCS_CKM.1)

**MDMPP40:FCS_CKM.1.1**

Refinement: The TSF shall [***invoke platform-provided functionality, implement functionality***] to 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 meets FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.3,***
***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,***
***FFC schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1***].

#### 5.1.2.2 Cryptographic Key Establishment (MDMPP40:FCS_CKM.2)

**MDMPP40:FCS_CKM.2.1**

Refinement: The TSF shall [***invoke platform-provided functionality, implement functionality***] to

perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [

***RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, 'Public-Key Cryptography Standards (PKCS) #1:RSA Cryptography Specifications Version 2.1',***
***Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography',***
***Finite field-based key establishment schemes that meets the following: NIST Special Publication 800-56A, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography'***].

#### 5.1.2.3 Cryptographic Key Destruction (MDMPP40:FCS_CKM_EXT.4)

**MDMPP40:FCS_CKM_EXT.4.1**

The TSF shall destroy plaintext keying material and critical security parameters by [
***invoking platform-provided functionality with the following rules:***

***- For volatile memory, the destruction shall be executed by [a single direct overwrite consisting of [zeroes] ,***
***- For non-volatile memory that consists of the invocation of an interface provided by the underlying platform that [logically addresses the storage location of the key and performs a [single] direct overwrite consisting of [a pseudo-random pattern using the TSF/Platform RBG (as specified inFCS_RBG_EXT.1]],***

***implementing key destruction in accordance with the following rules:***

***- For volatile memory, the destruction shall be executed by a single direct overwrite [consisting of zeroes] ,***
***- For non-volatile EEPROM, the destruction shall be executed by a single direct overwrite consisting of a pseudo-random pattern using the TSF/Platform RBG (as specified in FCS_RBG_EXT.1), followed by a read-verify,***
***- For non-volatile flash memory, that is not wear-leveled, the destruction shall be executed [by a single direct overwrite consisting of zeros followed by a read-verify] ,***
***- For non-volatile flash memory, that is wear-leveled, the destruction shall be executed [by a single direct overwrite consisting of zeros] ,***
***- For non-volatile memory other than EEPROM and flash, the destruction shall be executed by a single direct overwrite with a random pattern that is changed before each write***].

**MDMPP40:FCS_CKM_EXT.4.2**

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

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

**MDMPP40:FCS_COP.1.1(1)**

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

***AES-CBC (as defined in FIPS PUB 197 and NIST SP 800-38A) mode,***
***AES-GCM (as defined in NIST SP 800-38D),***
***(MDM agent platform) AES Key Wrap (KW) (as defined in NIST SP 800-38F***
] and cryptographic key sizes [***128-bit, 256-bit***].

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

**MDMPP40:FCS_COP.1.1(2)**

Refinement: The TSF shall [***invoke platform-provided functionality, implement functionality***] to

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

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

**MDMPP40:FCS_COP.1.1(3)**

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

***RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 4, ECDSA schemes using 'NIST curves' P-384 and [P-256] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 5***].

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

**MDMPP40:FCS_COP.1.1(4)**

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

#### 5.1.2.8 HTTPS Protocol (MDMPP40:FCS_HTTPS_EXT.1)

**MDMPP40:FCS_HTTPS_EXT.1.1**

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

**MDMPP40:FCS_HTTPS_EXT.1.2**

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

#### 5.1.2.9 Initialization Vector Generation (MDMPP40:FCS_IV_EXT.1)

**MDMPP40:FCS_IV_EXT.1.1**

The TSF shall [***invoke platform-provided functionality, implement functionality***] to generate IVs in accordance with Table 4.

| Cipher Mode | Reference | IV Requirement |
|---|---|---|
| Cipher Block Chaining (CBC) | SP800-38A | IVs shall be unpredictable. Repeating IVs leak information about whether the first one or more blocks are shared between two messages, so IVs should be non-repeating in such situations. |
| Galois Counter Mode (GCM) | SP800-38D | IV shall be non-repeating. The number of invocations of GCM shall not exceed 2^32 for a given secret key unless an implementation only uses 96-bit IVs (default length). |

**Table 4 References and IV Requirements for NIST-approved Cipher Modes**

#### 5.1.2.10 Extended: Random Bit Generation (MDMPP40:FCS_RBG_EXT.1)

**MDMPP40:FCS_RBG_EXT.1.1**

The TSF shall [***invoke platform-provided functionality, implement functionality***] to perform all deterministic random bit generation services in accordance with NIST Special Publication 800-90A using [***(MDM server) HMAC_DRBG (SHA-256), (MDM agent) Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES)***].

**MDMPP40:FCS_RBG_EXT.1.2**

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

#### 5.1.2.11 Cryptographic Key Storage (MDMPP40:FCS_STG_EXT.1)

**MDMPP40:FCS_STG_EXT.1.1**

The TSF shall utilize [***encryption as specified in FCS_STG_EXT.2***] for all persistent secrets and private keys.

#### 5.1.2.12 Cryptographic Key Storage (MDMA10:FCS_STG_EXT.1(2))

**MDMA10:FCS_STG_EXT.1.1(2)**

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

#### 5.1.2.13 Encrypted Cryptographic Key Storage (MDMPP40:FCS_STG_EXT.2)

**MDMPP40:FCS_STG_EXT.2.1**

The TSF shall [***implement functionality***] to encrypt all keys using AES in the [***CBC mode***].

#### 5.1.2.14 TLS Protocol (PKGTLS11:FCS_TLS_EXT.1)

**PKGTLS11:FCS_TLS_EXT.1.1**

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

#### 5.1.2.15 TLS Client Protocol (PKGTLS11:FCS_TLSC_EXT.1)

**PKGTLS11:FCS_TLSC_EXT.1.1**

The product shall implement TLS 1.2 (RFC 5246) and [***no earlier TLS versions***] as a client that supports the cipher suites [

***TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,***
***TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,***
***TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,***
***TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,***
***TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,***
***TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,***
***TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,***
***TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,***
***TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,***
***TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,***
***TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,***
***TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,***
***TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,***
***TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289***] and also supports functionality for [***mutual authentication***]. (TD0442 applied)

**PKGTLS11:FCS_TLSC_EXT.1.2**

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

**PKGTLS11:FCS_TLSC_EXT.1.3**

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

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

**PKGTLS11:FCS_TLSC_EXT.2.1**
The product shall support mutual authentication using X.509v3 certificates.

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

**PKGTLS11:FCS_TLSC_EXT.5.1**
The product shall present the Supported Groups Extension in the Client Hello with the supported groups [***secp256r1, secp384r1, ffdhe2048(256), ffdhe3072(257), ffdhe4096(258)***]

#### 5.1.2.18 TLS Server Protocol (PKGTLS11:FCS_TLSS_EXT.1)

**PKGTLS11:FCS_TLSS_EXT.1.1**
The product shall implement TLS 1.2 (RFC 5246) and [ ***no earlier TLS versions***] as a server that supports the cipher suites [
***TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,***
***TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,***
***TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,***
***TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,***
***TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,***
***TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,***
***TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,***
***TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,***
***TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,***
***TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,***
***TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,***
***TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,***
***TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,***
***TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289***] and no other cipher suites, and also supports functionality for [***mutual authentication, session renegotiation, no session resumption or session tickets***]. (TD0442 and TD0588 applied)

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

**PKGTLS11:FCS_TLSS_EXT.1.3**
The product shall perform key establishment for TLS using [
***RSA with size [2048 bits, 3072 bits, 4096 bits] and no other sizes,***
***Diffie-Hellman parameters with size [2048 bits, 3072 bits, 4096 bits] and no other sizes,***
***ECDHE parameters using elliptic curves [secp256r1, secp384r1] and no other curves***].
(TD0726 applied)

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

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

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

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

#### 5.1.2.20 TLS Server Support for Renegotiation (PKGTLS11:FCS_TLSS_EXT.4)

**PKGTLS11:FCS_TLSS_EXT.4.1**
The product shall support the 'renegotiation_info' TLS extension in accordance with RFC 5746.
**PKGTLS11:FCS_TLSS_EXT.4.2**
The product shall include the renegotiation_info extension in ServerHello messages.

### 5.1.3 Identification and authentication (FIA)

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

**MDMPP40:FIA_ENR_EXT.1.1**
The TSF shall authenticate the remote users over a trusted channel during the enrollment of a mobile device.
**MDMPP40:FIA_ENR_EXT.1.2**
The TSF shall limit the user's enrollment of devices to devices specified by [***IMEI***] and [***a number of devices***].

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

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

#### 5.1.3.3 Timing of Authentication (MDMPP40:FIA_UAU.1)

**MDMPP40:FIA_UAU.1.1**
Refinement: The TSF shall [***implement functionality***] to allow [recover user ID, **request a password reset and change display language**] on behalf of the user to be performed before the user is authenticated with the Server.
**MDMPP40:FIA_UAU.1.2**
Refinement: The TSF shall [***implement functionality***] that requires each user to be successfully authenticated with the Server before allowing any other TSF-mediated actions on behalf of that user.

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

**MDMPP40:FIA_X509_EXT.1.1(1)**
The TSF shall [***invoke platform-provided functionality, implement functionality***] to validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension, that the CA flag is set to TRUE for all CA certificates, and that any path constraints are met.
- The TSF shall validate that any CA certificate includes caSigning purpose in the key usage field.
- The TSF shall validate the revocation status of the certificate using [
  **(MDM server and MDM agent platform)** ***OCSP as specified in RFC 6960,*** **(MDM server and MDM agent platform)** ***a CRL as specified in RFC 5759 Section 5***].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  -- Certificates used for trusted updates and executable code integrity verification shall have the Code Signing Purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.

-- Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
-- Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the EKU field.
-- OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the EKU field.
-- Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the EKU field. (TD0641 applied)

**MDMPP40:FIA_X509_EXT.1.2(1)**

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

### 5.1.3.5 X.509 Certificate Validation (MDMPP40:FIA_X509_EXT.1(2))

**MDMPP40:FIA_X509_EXT.1.1(2)**

The TSF shall [***invoke platform-provided functionality, implement functionality***] to validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension, that the CA flag is set to TRUE for all CA certificates, and that any path constraints are met.
- The TSF shall validate that any CA certificate includes caSigning purpose in the key usage field.
- The TSF shall validate the revocation status of the certificate using [
  **(MDM server and MDM agent platform)** ***OCSP as specified in RFC 6960*****, (MDM server and MDM agent platform)** ***a CRL as specified in RFC 5759 Section 5*** ].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  -- Certificates used for trusted updates and executable code integrity verification shall have the Code Signing Purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
  -- Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
  -- Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the EKU field.
  -- OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the EKU field.
  -- Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the EKU field. (TD0641 applied)

**MDMPP40:FIA_X509_EXT.1.2(2)**

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

### 5.1.3.6 X.509 Certificate Authentication (MDMPP40:FIA_X509_EXT.2)

**MDMPP40:FIA_X509_EXT.2.1**

The TSF shall [

***- invoke platform-provided functionality to use X.509v3 certificates as defined by RFC 5280 to support authentication for [TLS] , and [no additional uses],***

***- implement functionality to use X.509v3 certificates as defined by RFC 5280 to support authentication for [***
***-- HTTPS in accordance with FCS_HTTPS_EXT.1,***
***-- TLS as defined in the Package for Transport Layer],***
***and [no additional uses]***].

**MDMPP40:FIA_X509_EXT.2.2**

When the [***TSF, TOE platform***] cannot establish a connection to determine the validity of a certificate, the TSF shall [***invoke platform-provided functionality, implement functionality***] to [***accept the certificate***].

#### 5.1.3.7 X.509 Unique Certificate (MDMPP40:FIA_X509_EXT.5)

**MDMPP40:FIA_X509_EXT.5.1**

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

### 5.1.4 Security management (FMT)

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

**MDMPP40:FMT_MOF.1.1(1)**

Refinement: The TSF shall restrict the ability to perform the functions
- listed in FMT_SMF.1(1),
- enable, disable, and modify policies listed in FMT_SMF.1(1),
- listed in FMT_SMF.1(2),
- [***enable, disable and modify policies listed in FMT_SMF.1(3)***] to authorized administrators.

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

**MDMPP40:FMT_MOF.1.1(2)**

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

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

**MDMPP40:FMT_MOF.1.1(3)**

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

#### 5.1.4.4 Trusted Policy Update (MDMPP40:FMT_POL_EXT.1)

**MDMPP40:FMT_POL_EXT.1.1**

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

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

**MDMA10:FMT_POL_EXT.2.1**

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

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

**MDMPP40:FMT_SMF.1.1(1)**

Refinement: The MDM Server shall be capable of communicating the following commands to the MDM Agent:
1. transition to the locked state (MDF Function 6)
2. full wipe of protected data (MDF Function 7)
3. unenroll from management
4. install policies
5. query connectivity status
6. query the current version of the MD firmware/software
7. query the current version of the hardware model of the device
8. query the current version of installed mobile applications
9. import X.509v3 certificates into the Trust Anchor Database (MDF Function 11)
10. install applications (MDF Function 16)
11. update system software (MDF Function 15)
12. remove applications (MDF Function 14)

and the following commands to the MDM Agent: [
***13. remove Enterprise applications (MDF Function 17),***
***14. wipe Enterprise data (MDF Function 28),***
***15. remove imported X.509v3 certificates and [ no other X.509v3 certificates] in the Trust Anchor Database (MDF Function 12),***
***17. import keys/secrets into the secure key storage (MDF Function 9),***
***18. destroy imported keys/secrets and [ no other keys/secrets] in the secure key storage (MDF Function 10),***
***19. read audit logs kept by the MD (MDF Function 32) (Samsung-only)***]
and the following MD configuration policies:
25. password policy:
    a. minimum password length
    b. minimum password complexity
    c. maximum password lifetime (MDF Function 1)
26. session locking policy:
    a. screen-lock enabled/disabled
    b. screen lock timeout
    c. number of authentication failures (MDF Function 2)
27. wireless networks (SSIDs) to which the MD may connect (MDF Function 2) ***(Samsung-only),***
28. security policy for each wireless network:
    a. [***specify the CA(s) from which the MD will accept WLAN authentication server certificate(s)***]
    b. ability to specify security type
    c. ability to specify authentication protocol
    d. specify the client credentials to be used for authentication
    e. [**no additional WLAN management functions**] (WLAN Client Function 1)
29. application installation policy by [
    ***- specifying authorized application repository(s) (Samsung-only),***
    ***- specifying a set of allowed applications and versions (an application whitelist) (Samsung-only),***
    ***- denying application installation***] (MDF Function 8)
30. enable/disable policy for [**camera and microphone**] across device and [***no other devices***] (MDF Function 5),

and the following MD configuration policies: [
***32. enable/disable policy for [NFC, Bluetooth, Wi-Fi, and cellular radios] (MDF Function 4) (Samsung-only),***

***34. enable/disable policy for [protocols supporting remote access] (MDF Function 25) (Samsung-only),***
***35. enable/disable policy for developer modes (MDF Function 26) (Samsung-only),***
***36. enable policy for data-at-rest protection (MDF Function 20),***
***37. enable policy for removable media's data-at-rest protection (MDF Function 21) (Samsung-only),***
***38. enable/disable policy for local authentication bypass (MDF Function 27) (Samsung-only),***
***47. the unlock banner policy (MDF Function 36) (Samsung-only),***
***48. configure the auditable items (MDF Function 37) (Samsung-only),***
***49. enable/disable [USB mass storage mode] (MDF Function 39) (Samsung-only),***
***51. enable/disable [***
***- Hotspot functionality authenticated by [pre-shared key],***
***- USB tethering authenticated by [no authentication]] (MDF Function 41) (Samsung-only),***
***52. enable/disable location services: [ across device] (MDF Function 22) (Samsung-only),***
***55. enable/disable policy for use of Biometric Authentication Factor (MDF Function 23)***].

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

**MDMPP40:FMT_SMF.1.1(2)**

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

a. choose X.509v3 certificates for MDM Server use
b. configure the [***a number of devices***] and [***no other features***] allowed for enrollment
c. [
***2. configure the TOE unlock banner,***
***3. 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***
***- query the current version of installed mobile applications***
***- read audit logs kept by the MD],***
***8. [Configure administrator login session timeout]***].

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

**MDMPP40:FMT_SMF.1.1(3)**

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

a. Configure application access groups
b. Download applications
c. [***no other functions***].

### 5.1.4.9 Specification of Management Functions (MDMA10:FMT_SMF_EXT.4)

**MDMA10:FMT_SMF_EXT.4.1**

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

Import the certificates to be used for authentication of MDM Agent communications;[***administrator provided device management functions in MDM PP***]; [***no additional functions***].

**MDMA10:FMT_SMF_EXT.4.2**

The MDM Agent shall be capable of performing the following functions:

Enroll in management;
Configure whether users can unenroll from management;

[***configure periodicity of reachability events***].

#### 5.1.4.10 Security Management Roles (MDMPP40:FMT_SMR.1(1))

**MDMPP40:FMT_SMR.1.1(1)**
Refinement: The TSF shall maintain the roles administrator, MD user, and [***Server primary administrator, Security configuration administrator, Device user group administrator, Auditor***].

**MDMPP40:FMT_SMR.1.2(1)**
The TSF shall be able to associate users with roles.

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

**MDMPP40:FMT_SMR.1.1(2)**
Refinement: The TSF shall additionally maintain the roles enrolled mobile devices, application access groups, and [**no additional authorized identified roles**].

**MDMPP40:FMT_SMR.1.2(2)**
Refinement: The MAS Server shall be able to associate users with roles.

#### 5.1.4.12 User Unenrollment Prevention (MDMA10:FMT_UNR_EXT.1)

**MDMA10:FMT_UNR_EXT.1.1**
The MDM Agent shall provide a mechanism to enforce the following behavior upon an attempt to unenroll the mobile device from management: [***prevent the unenrollment from occurring***].

### 5.1.5 Protection of the TSF (FPT)

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

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

#### 5.1.5.2 Internal TOE TSF Data Transfer (MDMPP40:FPT_ITT.1(1))

**MDMPP40:FPT_ITT.1.1(1)**
Refinement: The TSF shall [***implement functionality using [mutually authenticated TLS as defined in the Package for Transport Layer Security]***] to protect all data from disclosure and modification when it is transferred between separate parts of the TOE.

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

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

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

**MDMPP40:FPT_LIB_EXT.1.1**
The MDM software shall be packaged with only [**Eldos Coporation Solid File System, RSA BSAFE Cryto-J, Ext JS, Ext JS-org.webjars:extjs**].

#### 5.1.5.5 Functionality Testing (MDMPP40:FPT_TST_EXT.1)

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

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

#### 5.1.5.6 Trusted Update (MDMPP40:FPT_TUD_EXT.1)

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

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

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

### 5.1.6 TOE access (FTA)

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

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

### 5.1.7 Trusted path/channels (FTP)

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

**MDMPP40:FTP_ITC.1.1(1)**
Refinement: The TSF shall [***invoke platform-provided functionality to use [IPsec] (for the SYSLOG audit server, database, and Certificate Authority Server connections), implement functionality using [HTTPS in accordance with FCS_HTTPS_EXT.1] (for the administrator audit server channel)***] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [***[database server, Certificate Authority Server]***] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification and disclosure.

**MDMPP40:FTP_ITC.1.2(1)**
Refinement: The TSF shall [***invoke platform-provided functionality, implement functionality***] to permit the MDM Server or other authorized IT entities to initiate communication via the trusted channel.

**MDMPP40:FTP_ITC.1.3(1)**
Refinement: The TSF shall [***invoke platform-provided functionality (for the database and Certificate Authority Server connection), implement functionality (for the audit server channel)***] to initiate communication via the trusted channel for [**database server, Certificate Authority Server , audit server**].

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

**MDMPP40:FTP_ITC.1.1(2)**

Refinement: The TSF shall [***implement functionality using [mutually authenticated TLS as defined in the Package for Transport Layer Security]***] to provide a trusted communication channel between itself (as a server) and the MDM Agent that is logically distinct from other communication channels, provides assured identification of its end points, protects channel data from disclosure, and detects modification of the channel data.

**MDMPP40:FTP_ITC.1.2(2)**

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

**MDMPP40:FTP_ITC.1.3(2)**

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

#### 5.1.7.3 Trusted Channel (MDMPP40:FTP_ITC_EXT.1)

**MDMPP40:FTP_ITC_EXT.1.1**

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

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

**MDMPP40:FTP_TRP.1.1(1)**

Refinement: The TSF shall [***implement functionality using [HTTPS in accordance with FCS_HTTPS_EXT.1]***] to provide a trusted communication path between itself as a [***server***] and remote administrators that is logically distinct from other communication paths and provides assured identification of its endpoints and protection of the communicated data from modification and disclosure.

**MDMPP40:FTP_TRP.1.2(1)**

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

**MDMPP40:FTP_TRP.1.3(1)**

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

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

**MDMPP40:FTP_TRP.1.1(2)**

Refinement: The TSF shall [***invoke platform-provided functionality to use [TLS], implement functionality using [TLS as defined in the Package for Transport Layer Security]***] to provide a trusted communication path between itself (as a server) and MD users that is logically distinct from other communication paths and provides assured identification of its endpoints and protection of the communicated data from disclosure and detection of modification of the communicated data from modification and disclosure.

**MDMPP40:FTP_TRP.1.2(2)**

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

**MDMPP40:FTP_TRP.1.3(2)**

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

## 5.2 TOE Security Assurance Requirements

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

| Requirement Class | Requirement Component |
|---|---|
| **ADV: Development** | ADV_FSP.1: Basic Functional Specification |
| **AGD: Guidance documents** | AGD_OPE.1: Operational User Guidance |
| | AGD_PRE.1: Preparative Procedure |
| **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 5 Assurance Components**

### 5.2.1 Development (ADV)

#### 5.2.1.1 Basic Functional Specification (ADV_FSP.1)

**ADV_FSP.1.1d**
The developer shall provide a functional specification.

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

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

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

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

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

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

**ADV_FSP.1.2e**
The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

### 5.2.2 Guidance documents (AGD)

#### 5.2.2.1 Operational User Guidance (AGD_OPE.1)

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

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

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

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

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

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

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

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

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

#### 5.2.2.2 Preparative Procedures (AGD_PRE.1)

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

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

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

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

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

### 5.2.3 Life-cycle support (ALC)

#### 5.2.3.1 Labelling of the TOE (ALC_CMC.1)

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

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

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

#### 5.2.3.2 TOE CM Coverage (ALC_CMS.1)

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

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

**ALC_CMS.1.2c**
The configuration list shall uniquely identify the configuration items.

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

### 5.2.4 Tests (ATE)

#### 5.2.4.1 Independent Testing â€“ 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 satisfies the following security functional requirements:

- MDMPP40:FAU_ALT_EXT.1: The EMM Server alerts administrators by displaying a “notifications” tab containing alerts for the administrator. Currently, the EMM Server displays alerts received from the MDM agent for changes in enrollment status (i.e., successful un/enrollment of devices) and a failure of an Agent to apply policies. The EMM Server enforces no limit on the maximum number of queued messages, and queues messages for the administrator in the notifications tab.
- MDMA10:FAU_ALT_EXT.2: The EMM Agent can send a range of alerts to the EMM including: when policies are applied; reachability status (which could be triggered by the Server or Client); change in enrollment state; and failure to install or update an application from the MAS Server. These alerts are sent across the mutually authenticated TLS channel to the EMM available once enrolled and if that channel is not available any alerts are queued and forwarded when the channel is re-established.
  - Candidate Policies are received by the EMM Agent, sent by the enrolled EMM, via mutually authenticated TLS. When received the candidate policy is checked according to MDMA10:FMT_POL_EXT.2 where an alert is sent if the policy is not accepted (i.e., fails its signature check). If the check succeeds, the EMM Agent checks each policy setting and applies the settings that are valid for the given device using available management APIs. The EMM Agent sends an alert after applying policies including identifying any invalid settings.
  - Reachability events can be initiated by the EMM Agent when it sends any alerts or other messages to the EMM and alternately can be initiated by the EMM where it can make requests of the EMM Agent. The reachability status on the EMM Server is based on any secure communication with the EMM Agent.
  - If the communication channel used for alerts is not available (e.g., Airplane mode or other network discontinuities), the EMM Agent will cache alerts in its local file storage and will send those alerts once connectivity is restored. The local file storage is limited only by the space available on the mobile device.
- MDMPP40:FAU_GEN.1(1)/MDMPP40:FAU_GEN.1(2):: The EMM 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 **Table 2** along with the included information (which includes the minimum set specified by the SFR) along with
  - Startup and shutdown of the MDM Agent,
  - Commands issued to the MDM Agent,

- MDM policy updated,
- any modification commanded by the MDM Server,
- Failure to push a new application on a managed mobile device, and
- Failure to update an existing application on a managed mobile device.

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 **Table 2**).

- MDMA10:FAU_GEN.1(2)/MDMA10:FAU_SEL.1(2): The EMM Agent generates audit events (see **Table 3**) and immediately sends them to the EMM. As the audit events are generated, the EMM Agent will filter them so that only audit records matching the configured filter will be sent to the EMM and the rest discarded. Each event created by EMM Agent includes Log Data and Time, a device and/or Mobile IDs (if applicable), an Event Category (type), an Event (indicates success or failure) and additional information for specific events (indicated in the third column of the **Table 3**).
- MDMPP40: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.
- MDMPP40: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.
- MDMPP40:FAU_STG_EXT.1/MDMPP40:FAU_STG_EXT.2: The EMM always stores audit records locally in flat files stored on the TOE platform file system. The EMM provides administrators the capability to securely export EMM Console audit data through the EMM Server's administrative interface (WebUI) that is protected by an HTTPS trusted channel or alternately to a configured SYSLOG server that is protected using IPsec implemented by the underlying platform and configured according to available guidance documents. Windows log files generated by the server component's underlying Microsoft Platform can be locally access on the underlying platform or remotely using a RDP (Remote Desktop Protocol) or IPsec secured connection.

  The EMM 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 only allows authenticated administrators access to display audit records, but provides no capability for an administrator to change those records.

  The EMM Agent stores its audit logs within its local files and transmits those to the EMM via the MDMPP40:FPT_ITT.1(2) channel using TLS when requested by the enrolled EMM.

## 6.2 Cryptographic support

The Cryptographic support function satisfies the following security functional requirements:

- MDMPP40:FCS_CKM.1: Each EMM server component uses its RSA BSAFE Crypto-J cryptographic module to generate asymmetric RSA, DSA and ECDSA keys as part of key establishment. For authentication, the EMM allows the administrator to import RSA and ECDSA certificates into each EMM server component, as the components only generate keys during TLS key exchange.

  The EMM Agent relies upon the EMM for generation of RSA and ECDSA keypairs. During the EMM Agent's enrollment process, the EMM Agent relies upon the EMM 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.

- MDMPP40:FCS_CKM.2: The EMM supports asymmetric key generation for key establishment as part of TLS and HTTPS. The EMM allows 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 | Server | Yes | No (imported keys only) |
| AT Server | Both | Yes | No (imported keys 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).

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.

- MDMPP40:FCS_CKM_EXT.4: The EMM clears keys (TLS and HTTPS session keys) from memory after those keys are no longer needed. Furthermore, the EMM stores its certificates (the only persistently stored keying material) on an internal hard drive in encrypted format, and when an administrator configures new certificates, the EMM will directly overwrite the old keys with the new.

  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.

- MDMPP40:FCS_COP.1(*): The EMM uses the RSA BASFE Crypto-J Software library version 6.3, which provides the algorithms noted in the table below (along with the NIST standards to which they comply).

| SFR | NIST Standard | RSA Crypto-J 6.3 CAVP Cert # |
|---|---|---|
| **FCS_CKM.1 – Key Generation** *RSA, FFC and ECDSA key gen* | | |
| RSA<br>186-4: Key(gen) – 2048 bit | FIPS 186-4 | A3218 |
| ECDSA<br>186-4: Key(gen) P256, P-384 | FIPS 186-4 | A3218 |
| DSA FFC<br>186-4: Key(gen) – 2048 bit | FIPS 186-4 | A3218 |
| **FCS_CKM.2 – Key Establishment** | | |
| KAS ECC<br>KAS FFC | NIST SP 800-56A | A3218 |
| RSA-based key establishment schemes | RFC 8017 | Vendor Affirmed |

| | | |
|---|---|---|
| **FCS_COP.1(*)** | | |
| AES<br>128/256 CBC, GCM | FIPS 197, SP 800-38A/D | A3218 |
| RSA<br>SigGen(2048), SigVer(2048) | FIPS 186-4 | A3218 |
| ECDSA<br>SigGen/SigVer (P-256, P-384) | FIPS 186-4 | A3218 |
| SHA-256/384/512 | FIPS 180-4 | A3218 |
| HMAC SHA-256/384 | FIPS 198-1 & 180-4 | A3218 |
| **FCS_RBG_EXT.1** | | |
| DRBG<br>HMAC_DRBG (SHA-256) | | A3218 |

**Table 6 EMM Cryptographic Algorithms**

As described above, both the EMM (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 EMM also uses signature verification to ensure the authenticity of EMM 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. When employing HMAC-SHA-256 or HMAC-SHA-384, the TOE uses a 32 or 48-byte key and 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 utilizes its platform (Microsoft Windows Server 2016 or Windows Server 2019) 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 references in section 1.4.

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.

- MDMPP40:FCS_HTTPS_EXT.1: The EMM supports HTTPS and TLS (using an internal Tomcat server implementation) in compliance with the requirements of the MDMPP40. When accepting incoming HTTPS connections from remote administrators, the EMM follows RFC 2818 and presents its server certificate. However, the EMM does not request that the remote administrator present a certificate (in other words, the EMM does not require TLS mutual/client authentication for remote administrators). Instead, the remote administrator authenticates to the EMM using a username and password, transmitted to the EMM after they have established the TLS session.
- MDMPP40:FCS_IV_EXT.1: The EMM 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 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).

The EMM Agent is entirely dependent on the underlying platform to generate IVs as necessary in support of cryptographic functions used by the EMM Agent.

- MDMPP40:FCS_RBG_EXT.1: The EMM'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.

  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).

- MDMPP40:FCS_STG_EXT.1/MDMA10:FCS_STG_EXT.1(2)/MDMPP40:FCS_STG_EXT.2: The EMM encrypts its 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 (i.e., private keys associated with certificates) in the platform provided key storage (Android KeyStore) and then utilize those keys securely through the platform provided key storage.

- PKGTLS11:FCS_TLS_EXT.1/PKGTLS11:FCS_TLSC_EXT.1/PKGTLS11:FCS_TLSC_EXT.2/PKGTLS 11:FCS_TLSC_EXT.5/PKGTLS11:FCS_TLSS_EXT.1/PKGTLS11:FCS_TLSS_EXT.2/PKGTLS11:FCS_ TLSS_EXT.4: The EMM support TLS version 1.2 and the ciphersuites listed in section 5.1.2.22 and 5.1.2.23 – these can be configured by an administrator according to the available guidance documents. All versions of the SSL protocol and older versions of the TLS protocol are refused by the EMM. The EMM[1] performs certificate checking in conformance with FIA_X509_EXT.1 and additionally performs hostname checking to ensure that the expected hostname matches the Distinguished Name (DN) in the presented 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 EMM cannot contact the revocation server. The EMM does not utilize certificate pinning. The EMM supports only NIST curves secp256r1, and secp384r1 when using elliptic curve ciphers. The EMM supports RSA or DHE key exchange using certificates of either 2048, 3072 or 4096 bits. The EMM supports TLS renegotiation only for its web UI interface. When acting as a TLS client, the EMM server components support the use of wildcards in accordance with RFC 6125.

## 6.3 Identification and authentication

The Identification and authentication function satisfies the following security functional requirements:

- MDMPP40: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 Agent application running on their mobile device. The EMM Agent 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 a whitelist of IMEI values such that only devices with a configured IMEI value can successfully enroll. Otherwise, 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 and the device is allowed according to any

[1] The EMM web UI interface does not support mutual authentication for remote administrator access nor for initial device enrollment, so in those cases there is no certificate checking performed. After initial enrollment Agents communicate exclusively with the AT and PUSH server component interfaces requiring mutual authentication using X509 certificates.

IMEI restrictions, 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 Agent (or other agent) to the EMM occur through a mutually authenticated TLS session (in which the Client/Agent presents its certificate to the server).

- MDMA10: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.
- MDMPP40:FIA_UAU.1: The EMM Server's implementation requires that any user connecting to the Server authenticate by providing a username and password before providing any access to the EMM Server management functions. Prior to login, a user can request to recover their user ID, request a password reset, or change the display language; otherwise, a user cannot perform any actions (other than logging in). Furthermore, the Server only allows remote users to connect via HTTPS to ensure confidentiality.
- MDMPP40:FIA_X509_EXT.1(1)/MDMPP40:FIA_X509_EXT.1(2)/MDMPP40:FIA_X509_EXT.2/MDMPP40:FIA_X509_EXT.5: Both the EMM and EMM Agent validate and handle X.509v3 certificates in compliance with the requirements. The EMM and EMM Agent use X.509 certificates only during TLS/HTTPS trusted channel establishment (for server and client/mutual authentication). The EMM and EMM 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 and OCSP. 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 peer'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 revocation status and will accept the connection attempt.

  Both the EMM and EMM Agent will accept as valid any certificate for which the EMM or EMM Agent cannot reach the revocation server to check its status. The EMM 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.

  The EMM Agent components receive unique certificates during their enrollment process and they store the received certificates in the Android keystore (which stores the keys with permissions to only allow the applications themselves to access the keys). When the EMM Agent components subsequently contact the EMM, its will utilize the keys in the keystore. Each of the EMM Agent 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 receives 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 satisfies the following security functional requirements:

- MDMPP40:FMT_MOF.1(1)/MDMPP40:FMT_MOF.1(2)/MDMPP40:FMT_MOF.1(3): 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).

  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 below identifies the management functions that can be configured by a device management profile and app management profile.

  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.

- MDMPP40:FMT_POL_EXT.1/MDMA10:FMT_POL_EXT.2: The EMM Server signs each policy with a certificate configured for that purpose that the EMM Agent is configured to trust upon enrollment. When the EMM Agent receives any policy, it checks its signature and if the check fails the policy is discarded and an alert is sent to the EMM.

- MDMPP40: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 below identifies the management functions implemented for the Android and Apple iOS agents. Mandatory functions with green cells and functions supported by the Samsung or iOS phones are identified. The Samsung and iOS columns identify claimed functions in the respective STs and whether they can be managed by the EMM Server (Y=yes, N=no or not applicable).

| Management Commands and Policies | Samsung Android | iOS |
|---|---|---|
| 1. transition to the locked state (MDF Function 6) | Y | Y |
| 2. full wipe of protected data (MDF Function 7) | Y | Y |
| 3. unenroll from management | Y | Y |
| 4. install policies | Y | Y |
| 5. query connectivity status | Y | Y |
| 6. query the current version of the MD firmware/software | Y | Y |
| 7. query the current version of the hardware model of the device | Y | Y |
| 8. query the current version of installed mobile applications | Y | Y |
| 9. import Y.509v3 certificates into the Trust Anchor Database (MDF Function 11) | Y | Y |
| 10. install applications, (MDF Function 16) | Y | Y |
| 11. update system software, (MDF Function 15) | Y | Y |
| 12. remove applications, (MDF Function 14) | Y | Y |
| 13. remove Enterprise applications (MDF Function 17) | Y | |
| 14. wipe Enterprise data, (MDF Function 28) | Y | Y |
| 15. remove imported X.509v3 certificates and default X.509v3 certificates in the Trust Anchor Database (MDF Function 12) | Y | Y |
| 17. import keys/secrets into the secure key storage (MDF Function 9) | Y | Y |
| 18. destroy imported keys/secrets and [no other keys/secrets] in the secure key storage (MDF Function 10) | Y | Y |

| Management Commands and Policies | Samsung Android | iOS |
|---|---|---|
| 19. read audit logs kept by the MD (MDF Function 32) | Y | N |
| 25. password policy: a. minimum password length (MDF Function 1) | Y | Y |
| 25. password policy: b. minimum password complexity (MDF Function 1) | Y | Y |
| 25. password policy: c. maximum password lifetime (MDF Function 1) | Y | Y |
| 26. session locking policy: a. screen-lock enabled/disabled (MDF Function 2) | Y | Y |
| 26. session locking policy: b. screen lock timeout (MDF Function 2) | Y | Y |
| 26. session locking policy: c. number of authentication failures (MDF Function 2) | Y | Y |
| 27. wireless networks (SSIDs) to which the MD may connect (WLAN Client EP Function 2) | Y | N |
| 28. security policy for each wireless network: a. specify the CA(s) from which the MD will accept WLAN authentication server certificate(s) (WLAN Client EP Function 1) | Y | Y |
| 28. security policy for each wireless network: b. ability to specify security type (WLAN Client EP Function 1) | Y | Y |
| 28. security policy for each wireless network: c. ability to specify authentication protocol (WLAN Client EP Function 1) | Y | Y |
| 28. security policy for each wireless network: d. specify the client credentials to be used for authentication (WLAN Client EP Function 1) | Y | Y |
| 29. application installation policy by a. specifying authorized application repository(s) (MDF Function 8) – Samsung only | Y | N |
| 29. application installation policy by b. specifying a set of allowed applications based on [application name, developer signature] (an application whitelist) (MDF Function 8) – Samsung only | Y | N |
| 29. application installation policy by c. denying application installation (MDF Function 8) | Y | Y |
| 30. enable/disable policy for camera and microphone across device (MDF Function 5) | Y | Y |
| 32. enable/disable policy for NFC, Bluetooth, Wi-Fi, and cellular radios (MDF Function 4) - Samsung only | Y | N |
| 34. enable/disable policy for protocols supporting remote access (MDF Function 25) - Samsung only | Y | N |
| 35. enable/disable policy for developer modes (MDF Function 26) | Y | N |
| 36. enable policy for data-at rest protection (MDF Function 20) | Y | Y |
| 37. enable policy for removable media's data-at-rest protection (MDF Function 21) - Samsung only | Y | N |
| 38. enable/disable policy for local authentication bypass, (MDF Function 27) - Samsung only | Y | N |
| 47. the unlock banner policy (MDF Function 36) | Y | N |
| 48. configure the auditable items (MDF Function 37) | Y | N |
| 49. enable/disable a. USB mass storage mode (MDF Function 39) | Y | N |
| 51. enable/disable a. Hotspot functionality (MDF Function 41) - Samsung only | Y | N |
| 51. enable/disable b. USB tethering (MDF Function 41) - Samsung only | Y | N |
| 52. enable/disable location services: a. across device (MDF Function 22) | Y | N |
| 55. enable/disable policy for use of Biometric Authentication Factor (MDF Function 23) | Y | Y |

**Table 7 EMM Server Mobile Device Management Functions per Device**

- MDMPP40:FMT_SMF.1(2): In addition to managing mobile devices, the EMM Server 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.

- MDMPP40:FMT_SMF.1(3): The EMM Server provides the MAS server functionality. 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.
- MDMA10:FMT_SMF_EXT.4/MDMA10:FMT_UNR_EXT.1: Enrollment of a device occurs requires an MDM administrator to define a device, user and password on the EMM Server. The EMM Agent on that device is then used (by the mobile device user) to initiate an enrollment action which provides the user's ID and password, along with identification of the device. The EMM Agent component of the TOE is configured with an X.509v3 certificate suitable to facilitate secure communication with the EMM. 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 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). Finally, the administrator can forcibly unenroll the EMM Agent from EMM (and if the Agent will receive the unenrollment request when it has network connectivity to the EMM).
- MDMPP40:FMT_SMR.1(1)/MDMPP40: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'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 manage their mobile devices to secure organization data and access.

## 6.5 Protection of the TSF

The Protection of the TSF function satisfies the following security functional requirements:

- MDMPP40:FPT_API_EXT.1: A proprietary list of platform APIs used on server and agent is provided in Appendix A below.
- MDMPP40:FPT_ITT.1(1)/MDMPP40:FPT_ITT.1(2): The EMM (i.e., EMM Server, Push Server, AppTunnel Server, Push Proxy and AT-Relay) utilize 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 components and EMM Agents from disclosure and modification. The EMM Agent utilizes the TLS functions in its platform device for TLS and X509 to protect the

communication channels between itself and the EMM server components via TLS mutually authenticated using X509 certificates.

- MDMPP40:FPT_LIB_EXT.1: The EMM links with the following 3rd party libraries:
  - RSA BSAFE Cryto-J,
  - Ext JS / Ext JS-org.webjars:extjs,

  and the EMM Agent links with the following 3rd party library:
  - Eldos Coporation Solid File System.
- MDMPP40:FPT_TST_EXT.1: The EMM performs power-up tests to ensure correct operation. The EMM'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 as a whole performs a startup integrity check of its executable code to ensure that the calculated SHA-256 hash of the code during startup matches its expected value.. Should the startup integrity check fail, the server component will log the error and attempt to continue loading. This ensures that corrupted TOE code is detected.

  The EMM Agent relies upon its platform to perform a test of its cryptographic algorithms upon power-up.
- MDMPP40:FPT_TUD_EXT.1: The EMM Server provides a System page that displays the version of the EMM's software. To update the EMM's software, the administrator can (following the Administrator Guidance) obtain a software update, if one is available, and install the update through the platform user interface. During the installation process, the EMM's platform will check the Microsoft AuthentiCode signature embedded in the update file itself.

## 6.6 TOE access

The TOE access function satisfies the following security functional requirements:

- MDMPP40: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 copyright 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 satisfies the following security functional requirements:

- MDMPP40:FTP_ITC_EXT.1/MDMPP40:FTP_ITC.1(1)/MDMPP40:FTP_ITC.1(2): The EMM uses communication channels to an MDM Agent that is are internal to the TOE (i.e., the Android agent), external to the TOE (i.e., the Apple agent), as well as communication channels to other parts of the distributed TOE. The EMM uses TLS to secure communication with EMM Agents and HTTPS to secure communication with administrators (through their browser) where audit records can be exported. Since the EMM provides the MAS server functionality, there are no additional communication paths for the MAS audit communications. The EMM communicates with the Apple agent using TLS to secure the communication pathway with either initiating the communication. The EMM can be configured to use an external SYSLOG server for audit records and depends on a database server and Microsoft Certificate Authority and it utilizes IPsec provided by its host operating system to secure those communication channels when the SYSLOG server, database server and/or Certificate Authority are not operating on the same host.
- MDMPP40: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 (through a normal web browser) to securely administer the server. The EMM Server provides no other mechanism or method beyond HTTPS for a remote Administrator to configure or access the server.

- MDMPP40:FTP_TRP.1(2): Likewise, the EMM 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 Agent 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. Note that the EMM Agent utilizes the TLS functions in its platform device for this channel.

# Appendix A. Platform APIs Invoked by TOE

The following API are used by the EMM and EMM Agent to obtain services from the operating environment. All are published interfaces for the OE component being invoked.

**NOTE: Text in BLUE will be removed when a non-proprietary Security Target is provided for publication.**

**EMM's Invoked Platform API**

The EMM uses SQL and Java interfaces provided by its operating environment. The specific details are considered proprietary and are excluded in this public document.

- java.io.FileOutputStream.write()
- ISQLServerCallableStatement Interface
- ISQLServerConnection Interface
- SQLServerDataSource Class
- ISQLServerPreparedStatement
- ISQLServerResultSet
- ISQLServerStatement
- DateTimeOffset
- SQLServerBlob
- SQLServerCallableStatement
- SQLServerClob
- SQLServerConnection
- SQLServerConnectionPoolDataSource
- SQLServerDatabaseMetaData
- SQLServerDataSource
- SQLServerDataSourceObjectFactory
- SQLServerDriver
- SQLServerException
- SQLServerNClob Class
- SQLServerParameterMetaData
- SQLServerPooledConnection
- SQLServerPreparedStatement
- SQLServerResource
- SQLServerResultSet
- SQLServerResultSetMetaData
- SQLServerSavepoint
- SQLServerStatement
- SQLServerXAConnection
- SQLServerXADataSource
- SQLServerXAResource

**EMM Agent's Invoked Platform API**

The EMM Agent uses standard Android APIs, including available interfaces for TLS connections and other cryptographic services. The specific details are considered proprietary and are excluded in this public document.

- io.netty.handler.ssl.SslHandler.handshakeFuture().sync()
- java.security.keyStore.deleteEntry()
- javax.crypto.Cipher.doFinal()
- java.security.MessageDigest.digest()
- java.security.Signature.getInstance()
- java.security.Signature.initVerify()

- java.security.Signature.update()
- java.security.MessageDigest.digest()
- java.security.SecureRandom.setSeed()
- java.security.KeyStore.getCertificateChain()
- java.security.cert.X509Certificate.getExtendedKeyUsage()
- java.security.cert.X509Certificate.getBasicConstraints()
- java.security.cert.X509Certificate.getSubjectAlternativeNames()
- java.security.cert.X509Certificate.getSubjectDN()
- java.security.cert.X509Certificate.getExtensionValue()
- java.security.cert.X509CRL.isRevoked()
- java.security.cert.X509Certificate cert.verify()
- javax.net.ssl.SSLEngine.createSSLEngine()
- com.samsung.android.knox
- com.samsung.android.knox.accounts
- com.samsung.android.knox.application
- com.samsung.android.knox.bluetooth
- com.samsung.android.knox.browser
- com.samsung.android.knox.container
- com.samsung.android.knox.custom
- com.samsung.android.knox.datetime
- com.samsung.android.knox.ddar
- com.samsung.android.knox.deviceinfo
- com.samsung.android.knox.devicesecurity
- com.samsung.android.knox.dex
- com.samsung.android.knox.display
- com.samsung.android.knox.dlp
- com.samsung.android.knox.integrity
- com.samsung.android.knox.keystore
- com.samsung.android.knox.kiosk
- com.samsung.android.knox.kpcc
- com.samsung.android.knox.license
- com.samsung.android.knox.location
- com.samsung.android.knox.lockscreen
- com.samsung.android.knox.log
- com.samsung.android.knox.multiuser
- com.samsung.android.knox.net
- com.samsung.android.knox.net.apn
- com.samsung.android.knox.net.billing
- com.samsung.android.knox.net.firewall
- com.samsung.android.knox.net.nap
- com.samsung.android.knox.net.vpn
- com.samsung.android.knox.net.vpn.serviceprovider
- com.samsung.android.knox.net.wifi
- com.samsung.android.knox.nfc
- com.samsung.android.knox.remotecontrol
- com.samsung.android.knox.restriction
- com.samsung.android.knox.sdp
- com.samsung.android.knox.sdp.core
- com.samsung.android.knox.seams
- com.samsung.android.knox.ucm.configurator
- com.samsung.android.knox.ucm.core
- com.samsung.android.knox.ucm.plugin.agent
- com.samsung.android.knox.ucm.plugin.keystore
- ISQLServerCallableStatement Interface

- ISQLServerConnection Interface
- SQLServerDataSource Class
- ISQLServerPreparedStatement
- ISQLServerResultSet
- ISQLServerStatement
- DateTimeOffset
- SQLServerBlob
- SQLServerCallableStatement
- SQLServerClob
- SQLServerConnection
- SQLServerConnectionPoolDataSource
- SQLServerDatabaseMetaData
- SQLServerDataSource
- SQLServerDataSourceObjectFactory
- SQLServerDriver
- SQLServerException
- SQLServerNClob Class
- SQLServerParameterMetaData
- SQLServerPooledConnection
- SQLServerPreparedStatement
- SQLServerResource
- SQLServerResultSet
- SQLServerResultSetMetaData
- SQLServerSavepoint
- SQLServerStatement
- SQLServerXAConnection
- SQLServerXADataSource
- SQLServerXAResource
- MS CA Web Service for Certificate Enroll Service (CES) (i.e., https://IP:Port/…/service.svc/CES)

# Appendix B. Requirement Allocation

This section provides a mapping of the distributed TOE components to the SFRs in this ST. The following table presents the required mapping. Note that all MDMA10* SFRs apply only to the Android Agent and are excluded here.

| Requirement | Distributed TOE SFR Allocation |
|---|---|
| MDMPP40:FAU_ALT_EXT.1 | EMM Server |
| MDMPP40:FAU_GEN.1(1) | All[2] |
| MDMPP40:FAU_GEN.1(2) | EMM Server |
| MDMPP40:FAU_NET_EXT.1 | EMM Server |
| MDMPP40:FAU_SAR.1 | EMM Server |
| MDMPP40:FAU_STG_EXT.1 | All |
| MDMPP40:FAU_STG_EXT.2 | All |
| MDMPP40:FCS_CKM.1 | All |
| MDMPP40:FCS_CKM.2 | All |
| MDMPP40:FCS_CKM_EXT.4 | All |
| MDMPP40:FCS_COP.1(1) | All |
| MDMPP40:FCS_COP.1(2) | All |
| MDMPP40:FCS_COP.1(3) | All |
| MDMPP40:FCS_COP.1(4) | All |
| MDMPP40:FCS_HTTPS_EXT.1 | EMM/AT/Push Server |
| MDMPP40:FCS_IV_EXT.1 | All |
| MDMPP40:FCS_RBG_EXT.1 | All |
| MDMPP40:FCS_STG_EXT.1 | All |
| MDMPP40:FCS_STG_EXT.2 | EMM/AT/Push Server |
| PKGTLS11:FCS_TLS_EXT.1 | EMM/AT/Push Server |
| PKGTLS11:FCS_TLSC_EXT.1 | EMM/AT/Push Server |
| PKGTLS11:FCS_TLSC_EXT.2 | EMM/AT/Push Server |
| PKGTLS11:FCS_TLSC_EXT.5 | EMM/AT/Push Server |
| PKGTLS11:FCS_TLSS_EXT.1 | EMM/AT/Push Server |
| PKGTLS11:FCS_TLSS_EXT.2 | EMM/AT/Push Server |
| PKGTLS11:FCS_TLSS_EXT.4 | EMM/AT/Push Server |
| MDMPP40:FIA_ENR_EXT.1 | EMM Server |
| MDMPP40:FIA_UAU.1 | EMM Server |
| MDMPP40:FIA_X509_EXT.1(1) | EMM/AT/Push Server |
| MDMPP40:FIA_X509_EXT.1(2) | EMM/AT/Push Server |
| MDMPP40:FIA_X509_EXT.2 | EMM/AT/Push Server |
| MDMPP40:FIA_X509_EXT.5 | EMM Server |
| MDMPP40:FMT_MOF.1(1) | EMM Server |
| MDMPP40:FMT_MOF.1(2) | EMM Server |
| MDMPP40:FMT_MOF.1(3) | EMM Server |
| MDMPP40:FMT_POL_EXT.1 | EMM Server |
| MDMPP40:FMT_SMF.1(1) | EMM Server |
| MDMPP40:FMT_SMF.1(2) | EMM Server |
| MDMPP40:FMT_SMF.1(3) | EMM Server |
| MDMPP40:FMT_SMR.1(1) | EMM Server |

[2] Note that this ST identified the relevant agent events from the MDMPP40 in a table for MDMA10:FAU_GEN.1(2) to help distinguish.

| Requirement | Distributed TOE SFR Allocation |
|---|---|
| MDMPP40:FMT_SMR.1(2) | EMM Server |
| MDMPP40:FPT_API_EXT.1 | All |
| MDMPP40:FPT_ITT.1(1) | EMM/AT/Push Server |
| MDMPP40:FPT_ITT.1(2) | AT/Push Server and Agent |
| MDMPP40:FPT_LIB_EXT.1 | All |
| MDMPP40:FPT_TST_EXT.1 | EMM/AT/Push Server (see TD0438) |
| MDMPP40:FPT_TUD_EXT.1 | EMM/AT/Push Server (see TD0438) |
| MDMPP40:FTA_TAB.1 | EMM Server |
| MDMPP40:FTP_ITC.1(1) | EMM Server |
| MDMPP40:FTP_ITC.1(2) | AT/Push Server |
| MDMPP40:FTP_ITC_EXT.1 | EMM/AT/Push Server |
| MDMPP40:FTP_TRP.1(1) | EMM Server |
| MDMPP40:FTP_TRP.1(2) | EMM/AT/Push Server |