BlackBerry Unified Endpoint Management (UEM) Server and Android Client, Version 12 Security Target

Version 0.7
09 March 2022

(MDM PP v4.0/MDM Agent Module v1.0/TLS Package v1.1)

Prepared for:

BlackBerry

2240 University Ave. E
Waterloo, ON N2K 0A9
Canada

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 Blackberry UEM Server and Android client provided by Blackberry. The TOE is being evaluated as a Mobile Device Management (MDM) Server (that also acts as a Mobile Application Store (MAS) Server) and associated MDM 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 (SFRs) – 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 …").
- However, this Security Target only identifies operations performed while copying SFRs from the source Protection Profile and doesn't necessarily identify operations that were already performed in the Protection Profile. The reader should consult the source Protection Profiles to identify any operations already performed therein.
- Given SFRs are drawn from multiple Protection Profiles; each component name is prefixed with an acronym representing the source of the SFR. For example, MDMPP40:FCS_CKM.1 indicates that FCS_CKM.1 was drawn from the MDMPP40.
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

Acronyms

AA	Assurance Activity
CC	Common Criteria
CAVP	Cryptographic Algorithm Validation Program
CAVS	Cryptographic Algorithm Validation System
CEM	Common Evaluation Methodology
CCEVS	Common Criteria Evaluation and Validation Scheme
CCTL	Common Criteria Testing Laboratory

CCUF	Common Criteria Users Forum
CMVP	Cryptographic Module Validation Program
CSfC	Commercial Solutions for Classified
DEK	Data Encryption Key
DISA	Defense Information Systems Agency
DoD	Department of Defense
EAL	Evaluation Assurance Level
EAR	Entropy Analysis Report
FIPS	Federal Information Processing Standard
FSO	Field Security Office (DISA-FSO)
GUI	Graphical User Interface
HLD	High-level Design
IA	Initial Assessment
JRE	Java Runtime Environment
LDAP	Lightweight Directory Access Protocol
MAS	Mobile Application Store
MDM	Mobile Device Management
MDMAEP	Mobile Device Management Agent Extended Package
MDMPP	Mobile Device Management Protection Profile
NIAP	National Information Assurance Partnership
NIST	National Institute of Standards and Technology
NSA	National Security Agency
OTA	Over The Air
PCL	Product Compliant List
PKI	Public Key Infrastructure
PP	Protection Profile
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SOF	Strength of Function
ST	Security Target
STIG	Security Technical Implementation Guide
TOE	Target of Evaluation
TSF	TOE Security Functions
TSFI	TOE Security Function Interface
TSS	TOE Summary Specification
UCCO	Unified Capabilities Certification Office (DISA UCCO)
UEM	Unified Endpoint Management
U.S.	United States
VR	Validation Report

1.1 Security Target Reference

ST Title – BlackBerry UEM Server and Android Client, Version 12 Security Target

ST Version – Version 0.6

ST Date – 28 April 2020

1.2 TOE Reference

TOE Identification – BlackBerry UEM Server and Android Client, Version 12

TOE Developer – BlackBerry

Evaluation Sponsor – BlackBerry

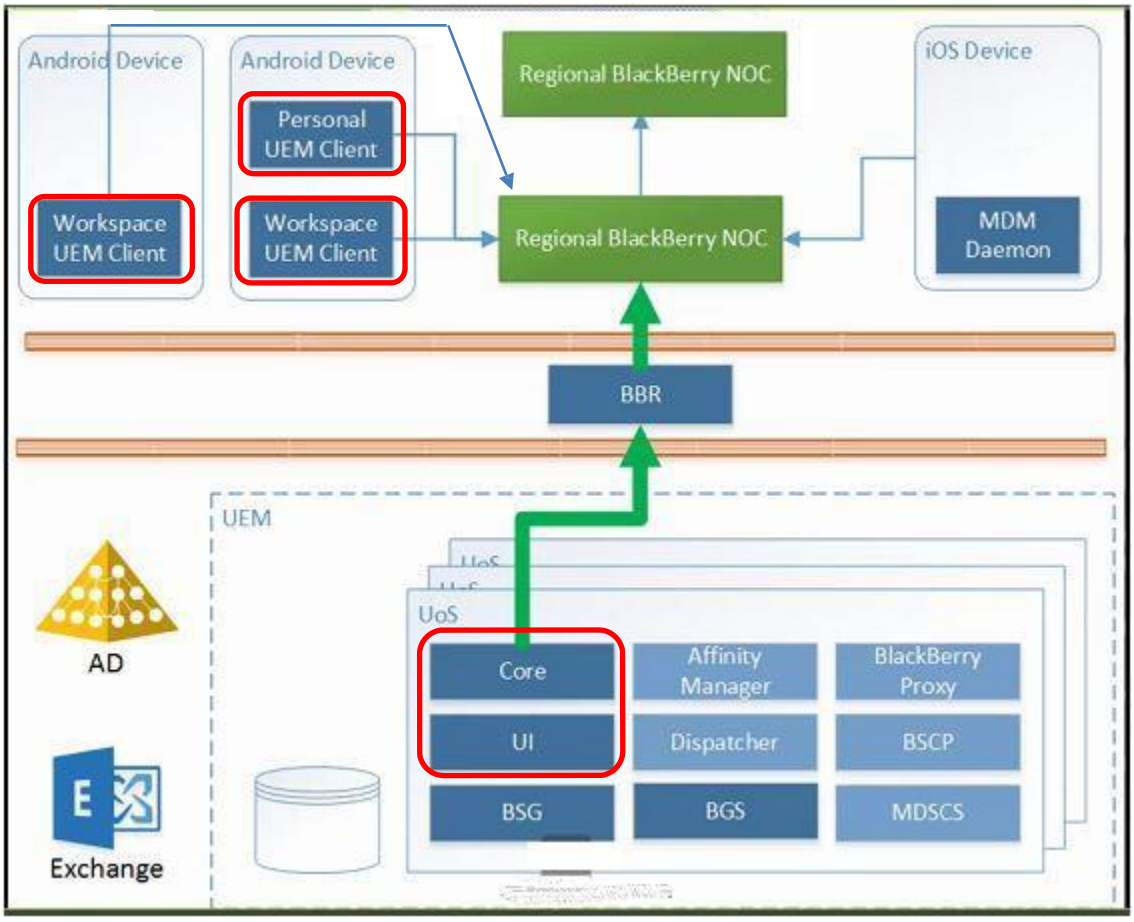
1.3 TOE Overview

The Target of Evaluation (TOE) is the BlackBerry Unified Endpoint Management (UEM) Server and Android Client version 12.

The UEM Server provides centralized management of mobile devices and the UEM Android Client Agent (installed on each android device) enforces the policies of the Server on each android device.

1.4 TOE Description

While the BlackBerry UEM product has a fairly complex overall architecture, the security enforcing components of the TOE are those circled in red in the figure below. The other components in the figure are either not related to the enforcement of any security requirements or are not part of the product, but rather are part of the operational environment.



The BlackBerry UEM server, including the Core and UI security enforcing components, is implemented with a combination of Java and native code running on Windows Server 2016 with Java JRE 8.0. Ideally, the scope of supported platforms for the evaluation would be Windows Server 2016 wherever it is deployable, however, it will be limited due to NIAP policy about CAVP algorithm certificates – the allowed environments would be expected to conform to the environments of the CAVP certificates (e.g., using the processors used for CAVP algorithm testing). In this case, the CAVP testing for Certicom was done on Windows Server 2016 running in a virtual environment (VMWare ESXi 6.5) on an Intel Xeon E5-2670**.**

The BlackBerry UEM Android Client has two main deployment methods– as a single Workspace client or alternate as a dual client with one managing the Personal (whole) device and another managing the Workspace. There is one BlackBerry

UEM client deployment per enrolled mobile device. The scope of supported managed client devices for the evaluation will be limited by the set of devices evaluated on the NIAP PCL (ref.

Android 11 https://www.niap-ccevs.org/Product/Compliant.cfm?PID=11211
Android 11 https://www.niap-ccevs.org/Product/Compliant.cfm?PID=11160
Android 10 https://www.niap-ccevs.org/Product/Compliant.cfm?PID=11109
Android 10 https://www.niap-ccevs.org/Product/Compliant.cfm?PID=11042

Since the iOS agents are evaluated as part of the Apple iOS evaluations, the UEM server will be tested to ensure it can manage those devices, but the agent's behavior on those devices will not otherwise be tested. The support will be limited by the set of devices evaluated on the NIAP PCL (ref.

- iOS 14 - https://www.niap-ccevs.org/Product/Compliant.cfm?PID=11146

1.4.1 TOE Architecture

As depicted above the UEM Server consists of a number of components. However, only the Core and UI components are included in the TOE for the purpose of evaluation. The other components are either disabled or play no role in any security enforcement.

The UEM Server requires a SQL database to operate and can optionally be configured to utilize an LDAP server for user authentication as well as a SYSLOG server to export audit records. Some other components such as Exchange are not included in the scope of evaluation or are not security relevant – the BlackBerry NOC is a network routing component through which UEM Server – client communication travels. They are not security relevant for the purpose of this evaluation since the server-client channels are secured end to end between the TOE components and through the other components. Those other components cannot decrypt or otherwise access information in those secure channels, although they can disrupt or redirect them, like any other components on the Internet.

The UEM Android Client is part of the TOE since Android does not have agents of its own. The UEM Server can manage mobile Android devices through interaction with an enrolled UEM Android Client and can alternately manage mobile iOS devices through interaction with the iOS agent developed and evaluated by Apple.

1.4.1.1 Physical Boundaries

The physical boundaries of the BlackBerry UEM Server and Android Client are the physical perimeter of the servers hosting the UEM Server and the physical perimeter of the mobile devices being managed by the UEM Server (put another way, the mobile devices running the Android Client).

The UEM Server also interacts with Microsoft SQL server and optionally LDAP and SYSLOG servers as described above.

1.4.1.2 Logical Boundaries

This section summarizes the security functions provided by the BlackBerry UEM Server and Android Client:

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

1.4.1.2.1 Security audit

The BlackBerry UEM server is designed to generate and export audit events listed in **Table 5-2**. The audit events are stored in the SQL database and sent to the configured syslog servers as events occur. The BlackBerry UEM server can also generate alerts for specific events – these alerts are sent to administrators as e-mails. The BlackBerry UEM server supports TLS tunneling of syslog messages to protect exported audit records.

The BlackBerry UEM Android client is also designed to generate and export audit events listed in **Table 5-3**. It stores audit events in the platform audit logs which it can retrieve and send to its enrolled BlackBerry UEM server. The BlackBerry UEM server will forward the events to a configured syslog server as the events are received. The BlackBerry UEM Android client can also send required alerts directly to the BlackBerry UEM server which are received, logged as audit events, and treated as administrator alerts.

1.4.1.2.2 Cryptographic support

The BlackBerry UEM server uses the Certicom Security Builder GSE-J Crypto Core Module (https://csrc.nist.gov/projects/cryptographic-module-validation-program/Certificate/3391) for its cryptographic operations. It includes the following algorithm certificates which are applicable as the platform for this evaluation:

- AES Cert. #5539
- DRBG Cert. #2194
- DSA Cert. #1421
- ECDSA Cert. #1490,
- HMAC Cert. #3690
- KAS Cert. #190
- RSA Cert. #2972
- SHS Cert. #4445

The BlackBerry UEM Android client uses the cryptographic functions provided by the evaluated mobile devices. As such, the Android client can reference the applicable certificates in the preceding evaluations of those devices.

The BlackBerry UEM server implements a X.509 key hierarchy summarized as follows:

1. The PKI is rooted in a self-signed certificate (RSA 4096 SHA512) created when the first server is installed.
2. The root is used to issue an intermediate CA certificate (RSA 3072 SHA512) also created when the first server is installed.
3. Additional certificates are issued using the intermediate CA certificate as follows:
    a. Console web server certificate (RSA 2048 SHA512)
    b. Server client certificate (RSA 2048 SHA512) – used for SYSLOG, LDAP, etc.
    c. Profile signing certificate (RSA 2048 SHA512) – used for Apple MDM
    d. Per-device BDMI payload signing key (RSA 3072 SHA512)
    e. Per-device enrolled device certificates - issued during enrollment (RSA 2048 SHA512)
4. All of the certificates above, except the per-device certificates, are stored in the SQL database and the key store is encrypted with a DEK (AES-CBC 256) also created during installation. The per-device BDMI keys are encrypted using the DEK separately from the rest of the key store. The DEK is encrypted using an EC secp512r1 key (stored in the Windows key store), that is unique to each unit of scale (created during installation), and stored on the local file system of each unit of scale.
5. Each individual certificate in the key store is also encrypted individually using a DEK created during installation using PBEWithHmacSHA256AndAES256 (AES-CBC mode).
6. The enrolled device certificate private keys are generated on the mobile device and signed by the intermediate CA on the applicable UEM server.
7. Additional trusted root CAs can be loaded to support accepting certificates from other devices (syslog, ldap, etc.).

1.4.1.2.3 Identification and authentication

The BlackBerry UEM server requires administrators to login prior to performing any security functions or accessing any services, such as creating an activation password. Similarly, mobile devices must authenticate with the server using an activation password prior to enrolling

Both the BlackBerry UEM server and Android client use X.509 certificates in conjunction with TLS to both authenticate and secure remote connections.

1.4.1.2.4 Security management

The BlackBerry UEM server facilitates granular administrative access to functions based on roles: server primary administrators, security configuration administrators, device user administrators, auditor, and mobile device users. Administrators access the BlackBerry UEM server via a web-based interface. The BlackBerry UEM server also supports the definition of mobile device users, and upon enrollment each mobile device generates an X.509 certificate used to identify that enrolled device.

The BlackBerry UEM server provides all the features necessary to manage its own security functions as well as to manage mobile device policies sent to enrolled mobile devices (via their clients).

The BlackBerry UEM Android client provides the features necessary to securely communicate and enroll with the BlackBerry UEM server, apply policies received from the BlackBerry UEM server, and report the results of applying policies.

1.4.1.2.5 Protection of the TSF

The BlackBerry UEM server and Android client work together to ensure that all security related communication between those components is protected from disclosure and modification.

The BlackBerry UEM server includes self-testing capabilities to ensure that they are functioning properly as well as to cryptographically verify that their executable images are not corrupted. The UEM server also includes secure update capabilities to ensure the integrity of any updates so that updates will not introduce malicious or other unexpected changes in the TOE.

1.4.1.2.6 TOE access

The BlackBerry UEM 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 BlackBerry UEM server uses TLS/HTTPS to secure communication channels between itself and remote administrators and mobile device users accessing the server via a web-based user interface. It also uses TLS to secure communication channels between itself, enrolled devices, its configured SQL database server, syslog servers, and optionally configured LDAP servers.

The following is a summary of applicable secure channels:

1. UEM server console used by administrators – TLS not subject to mutual X.509 authentication. Certicom implementation of TLS on server.
2. Mobile device UEM client to UEM server – TLS not subject to mutual X.509 authentication for initial enrollment, but always uses mutual X.509 authentication once enrolled. Certicom implementation of TLS on server – Mobile device implementation of TLS on the client end.
3. UEM server to SQL database, SYSLOG and LDAP – TLS optionally configured for mutual X.509 authentication. Certicom implementation of TLS on server. Communication with the SQL database is either local within the Windows platform on which the UEM server executes, or protected by IPsec provided by the Windows platform.

1.4.2 TOE Documentation

BlackBerry offers documentation that describes the use and administration of the applicable security features of the TOE. The following document was examined as part of the evaluation.

- . BlackBerry UEM Administrative Guidance Document, UEM Version 12.12, April 2020.

2. Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, April 2017.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1, Revision 5, April 2017.
    - Part 3 Conformant
- PP-Configuration for Mobile Device Management (MDM) and MDM Agents, Version 1.0, 2020-01-27
- The PP-Configuration is comprised of the following:
    - Protection Profile for Mobile Device Management, Version 4.0, 2019-04-25 (MDMPP40 - https://www.niap-ccevs.org/Profile/Info.cfm?PPID=428&id=428) with the following Technical Decisions applied: TD0438, TD0461, TD0467, TD0479 and the following packages:
        - Functional Package for Transport Layer Security (TLS), Version 1.1, 1 March 2019 (PKGTLS11 - https://www.niap-ccevs.org/Profile/Info.cfm?PPID=439&id=43) with the following Technical Decisions applied: TD0442, TD0469, TD0499
    - PP-Module for MDM Agents, Version 1.0, 2019-04-25 (MDMA10 - https://www.niap-ccevs.org/Profile/Info.cfm?PPID=441&id=441) with the following Technical Decisions applied: TD0491

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 BlackBerry UEM Server and Android Client 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.3: TLS Client Support for Signature Algorithms Extension
- PKGTLS11:FCS_TLSC_EXT.5: TLS Client Support for Supported Groups Extension
- PKGTLS11:FCS_TLSS_EXT.1: TLS Server Protocol
- PKGTLS11:FCS_TLSS_EXT.2: TLS Server Support for Mutual Authentication
- MDMPP40:FIA_ENR_EXT.1: Enrollment of Mobile Device into Management
- MDMA10:FIA_ENR_EXT.2: Agent Enrollment of Mobile Device into Management
- MDMPP40:FIA_X509_EXT.1(1): X.509 Certificate Validation
- MDMPP40:FIA_X509_EXT.2: X.509 Certificate Authentication
- MDMPP40:FIA_X509_EXT.5: X.509 Unique Certificate
- MDMPP40:FMT_POL_EXT.1: Trusted Policy Update
- MDMA10:FMT_POL_EXT.2: Agent Trusted Policy Update
- MDMPP40:FMT_SAE_EXT.1: Security Attribute Expiration
- MDMA10:FMT_SMF_EXT.4: Specification of Management Functions
- MDMA10:FMT_UNR_EXT.1: User Unenrollment Prevention
- MDMPP40:FPT_API_EXT.1: Use of Supported Services and APIs
- MDMPP40:FPT_LIB_EXT.1: Use of Third Party Libraries
- MDMPP40:FPT_TST_EXT.1: Functionality Testing
- MDMPP40:FPT_TUD_EXT.1: Trusted Update
- MDMPP40:FTP_ITC_EXT.1: Trusted Channel

5. Security Requirements

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

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

The SARs are also drawn from the MDMPP40/MDMA10/PKGTLS11 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 BlackBerry UEM Server and Android Client 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
	MDMPP40:FAU_SEL.1: Security Audit Event Selection
	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.3: TLS Client Support for Signature Algorithms Extension
	PKGTLS11:FCS_TLSC_EXT.5: TLS Client Support for Supported Groups Extension
	PKGTLS11:FCS_TLSS_EXT.1: TLS Server Protocol
	PKGTLS11:FCS_TLSS_EXT.2: TLS Server Support for Mutual Authentication

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

Table 5-1 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. Start up and shut down of the MDM System,
b. All administrative actions,
c. [***Commands issued to the MDM Agent***],
d. Specifically defined auditable events listed in **Table 5-2**,
e. [***no other events***].

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_SEL.1	All modifications to the audit configuration that occur while the audit collection functions are operating.	No additional information.
MDMPP40:FAU_STG_EXT.1	None.	
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. [***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. Failure to verify presented identifier.	Reason for failure. Presented identifier and reference identifier.
PKGTLS11:FCS_TLSC_EXT.2	None.	None
PKGTLS11:FCS_TLSC_EXT.3	None.	None
PKGTLS11:FCS_TLSC_EXT.5	None.	None

Requirement	Auditable Events	Additional Content
PKGTLS11:FCS_TLSS_EXT.1	Failure to establish a TLS session.	Reason for failure.
PKGTLS11:FCS_TLSS_EXT.2	None.	None.
MDMPP40:FIA_ENR_EXT.1	Failure of MD user authentication.	Presented username.
MDMPP40:FIA_UAU.1	None.	
MDMPP40:FIA_X509_EXT.1(1)	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. Change of policy settings.	Command sent and identity of MDM Agent recipient(s). Policy changed and value or full policy.
MDMPP40:FMT_MOF.1(2)	Enrollment by a user.	Identity of user.
MDMPP40:FMT_MOF.1(3)	None.	
MDMPP40:FMT_POL_EXT.1	None.	
MDMPP40:FMT_SAE_EXT.1	Enrollment attempted after expiration of authentication data.	Identity of user.
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(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. Failure of self-test. Detected integrity violation	Algorithm that caused failure. The TSF code file that caused the integrity violation.
MDMPP40:FPT_TUD_EXT.1	Success or failure of signature verification.	No additional information.
MDMPP40:FTA_TAB.1	Change in banner setting.	No additional information.
MDMPP40:FTP_ITC.1(1)	Initiation and termination of the trusted channel.	Trusted channel protocol. Non-TOE endpoint of connection
MDMPP40:FTP_ITC.1(2)	Initiation and termination of the trusted channel.	Trusted channel protocol. Non-TOE endpoint of connection.
MDMPP40:FTP_ITC_EXT.1	None.	
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 5-2 MDM Server Auditable Events[1]

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 5-2**; [**no other audit relevant information**].

[1] Note that the FAU_GEN.1(1) audit events that must also be generated by the MDM agent have been included in the FAU_GEN.1(2) audit table so that the MDM server and MDM agent audit events are more clearly distinguished.

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 5-3**, and [***no other events***].

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 5-3 MDM Android Client Auditable Events[2]

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 5-3**; 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**].

[2] Note that some of these events (with MDMPP40 prefix) are included here from FAU_GEN.1(1) in order to more clearly represent the full set of events the MDM agent is required to produce.

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 [***invoke platform-provided functionality, 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 [***invoke platform-provided functionality, 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 (MDMPP40:FAU_SEL.1)

MDMPP40:FAU_SEL.1.1

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 c. failure of auditable security events d. [**no other attributes**].

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

MDMA10:FAU_SEL.1.1(2)

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

5.1.1.10 External Trail Storage (MDMPP40:FAU_STG_EXT.1)

MDMPP40:FAU_STG_EXT.1.1

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

5.1.1.11 Audit Event Storage (MDMPP40:FAU_STG_EXT.2)

MDMPP40:FAU_STG_EXT.2.1

The TSF shall [***invoke platform-provided 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 [

(MDM server and MDM agent platform) ***RSA schemes using cryptographic key sizes of 2048-bit or greater that meets FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.3,***

(MDM server and MDM agent platform) ***ECC schemes using 'NIST curves' P-384 and [P-256, P-521] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.4***].

5.1.2.2 Cryptographic Key Establishment (MDMPP40:FCS_CKM.2)

MDMPP40:FCS_CKM.2.1

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

(MDM server and MDM agent platform) ***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',***

(MDM server and MDM agent platform) ***Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography'***].

5.1.2.3 Cryptographic Key Destruction (MDMPP40:FCS_CKM_EXT.4)

MDMPP40:FCS_CKM_EXT.4.1

The TSF shall destroy plaintext keying material and critical security parameters by [

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 [selection:

-- logically addresses the storage location of the key and performs a [single] direct overwrite consisting of [zeroes]],

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 block erase] ,

- 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: [

(MDM server and MDM agent platform) ***AES-CBC (as defined in FIPS PUB 197 and NIST SP 800-38A) mode,***
(MDM server and MDM agent platform) ***AES-GCM (as defined in NIST SP 800-38D),***
(MDM agent platform) ***AES Key Wrap (KW) (as defined in NIST SP 800-38F),***
(MDM agent platform) ***AES-CCM (as defined in NIST SP 800-38C)***]

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, P-521] 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, SHA-512***], key sizes [**256, 384, 512-bits**], and message digest sizes [***256, 384, 512***] bits that meet the following: FIPS Pub 198-1, 'The Keyed-Hash Message Authentication Code', and FIPS Pub 180-4, 'Secure Hash Standard.'

5.1.2.8 HTTPS Protocol (MDMPP40:FCS_HTTPS_EXT.1)

MDMPP40:FCS_HTTPS_EXT.1.1

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

MDMPP40:FCS_HTTPS_EXT.1.2

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

5.1.2.9 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 5-4**.

Cipher Mode	Reference	IV Requirement
Cipher Block Chaining (CBC) **(MDM server and MDM agent platform)**	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) **(MDM agent platform)**	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 5-4 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) ***Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES);***
(MDM agent platform) ***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 [***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 [***platform-provided key storage, 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) (TD0442 applied)

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_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,***
- ***TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289]*** and

also supports functionality for [***mutual authentication***]

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, secp521r1***]

5.1.2.18 TLS Server Protocol (PKGTLS11:FCS_TLSS_EXT.1) (TD0442 applied)

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 [

- ***(Server–to–Agent communication only) TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,***
- ***(Server–to–Agent communication only) TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,***
- ***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***]

PKGTLS11:FCS_TLSS_EXT.1.2

The product shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0 and [***none***]

PKGTLS11:FCS_TLSS_EXT.1.3

The product shall perform key establishment for TLS using [***RSA with size [2048 bits, 3072 bits, 4096 bits], ECDHE parameters using elliptic curves [secp256r1, secp384r1, secp521r1] and no other curves***]

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.

5.1.3 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 [***[Device Serial Number]***] and [***a number of devices, specific time period***].

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 [**password recover, choice of language and selection of authentication provider**] 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 and that the CA flag is set to TRUE for all CA certificates.
- The TSF shall validate the revocation status of the certificate using [
  (MDM server and MDM agent platform) ***the Online Certificate Status Protocol (OCSP) as specified in RFC 2560,***
  (MDM agent platform) ***a Certificate Revocation List (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 extendedKeyUsage field.
  -- CSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id- kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.
  -- Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the extendedKeyUsage field.

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 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 Security], 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 [***not accept the certificate***].

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

MDMPP40:FIA_X509_EXT.5.1

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

5.1.4 Security management (FMT)

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

MDMPP40:FMT_MOF.1.1(1)

Refinement: The TSF shall restrict the ability to perform the functions

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

MDMA10:FMT_POL_EXT.2.2

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

5.1.4.6 Security Attribute Expiration (MDMPP40:FMT_SAE_EXT.1)

MDMPP40:FMT_SAE_EXT.1.1

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

MDMPP40:FMT_SAE_EXT.1.2

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

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

MDMPP40:FMT_SMF.1.1(1)

Refinement: The MDM Server shall be capable of communicating the following commands to the MDM Agent:

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

and the following commands to the MDM Agent: [

13. remove Enterprise applications (MDF Function 17),

14. wipe Enterprise data (MDF Function 28),

15. remove imported X.509v3 certificates and [no other X.509v3 certificates] in the Trust Anchor Database (MDF Function 12)

17. import keys/secrets into the secure key storage (MDF Function 9),

18. destroy imported keys/secrets and [no other keys/secrets] in the secure key storage (MDF Function 10),

19. read audit logs kept by the MD (MDF Function 32) (Android 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) (Android 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) (Android Only),
- specifying a set of allowed applications and versions (an application whitelist) (Android Only),
- denying application installation] (MDF Function 8)

30. enable/disable policy for [***camera (iOS and Android) and microphone (Android 8.0 and Android 9 only)***] across device and [***no other method***] (MDF Function 5),

and the following MD configuration policies: [
31. enable/disable policy for the VPN protection across MD and [no other method] (MDF Function 3),
32. enable/disable policy for [NFC, Bluetooth, Wi-Fi, and cellular radios] (MDF Function 4) (Android Only),
34. enable/disable policy for [protocols supporting remote access] (MDF Function 25) (Android Only),
35. enable/disable policy for developer modes (MDF Function 26) (Android 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) (Android Only),
38. enable/disable policy for local authentication bypass (MDF Function 27) (Android Only),
40. enable/disable policy for display notification in the locked state of [all other application-based notifications] (MDF Function 19) (iOS Only),
47. the unlock banner policy (MDF Function 36),
48. configure the auditable items (MDF Function 37) (Android Only),
49. enable/disable [USB data transfer without authentication of the connection system] (MDF Function 39) (Android Only),
51. enable/disable [
- Hotspot functionality authenticated by [pre-shared key] (Android Only),
- USB tethering authenticated by [no authentication]] (MDF Function 41) (Android Only),
52. enable/disable location services: [across device] (MDF Function 22) (Android Workspace-only activation types only),
54. enable/disable policy for the Always-On VPN protection across device (MDF Function 45) (Android 8.1 and Android 9 only),
55. enable/disable policy for use of Biometric Authentication Factor (MDF Function 23)].

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

MDMPP40:FMT_SMF.1.1(2)

Refinement: The TSF shall be capable of performing the following management functions:
a. choose X.509v3 certificates for MDM Server use
b. configure the [
- device serial numbers
- a number of devices,
- specific time period] and [***no other features***] allowed for enrollment
c. [
2. configure the TOE unlock banner,
5. configure the length of time the enrollment authenticator is valid,
8. [Configure administrator login session timeout = 15 minutes]].

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

MDMPP40:FMT_SMF.1.1(3)

Refinement: The MAS Server shall be capable of performing the following management functions: a. Configure application access groups b. Download applications c. [***no other functions***].

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

MDMA10:FMT_SMF_EXT.4.1

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

MDMA10:FMT_SMF_EXT.4.2

The MDM Agent shall be capable of performing the following functions:
Enroll in management;
Configure whether users can unenroll from management;
[***no other functions***].

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

MDMPP40:FMT_SMR.1.1(1)

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

MDMPP40:FMT_SMR.1.2(1)

The TSF shall be able to associate users with roles.

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

MDMPP40:FMT_SMR.1.1(2)

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

MDMPP40:FMT_SMR.1.2(2)

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

5.1.4.13 User Unenrollment Prevention (MDMA10:FMT_UNR_EXT.1)

MDMA10:FMT_UNR_EXT.1.1

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

5.1.5 Protection of the TSF (FPT)

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

MDMPP40:FPT_API_EXT.1.1

The TSF shall use only documented platform API's.

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

MDMPP40:FPT_ITT.1.1(2)

Refinement: The TSF shall [***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.

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

MDMPP40:FPT_LIB_EXT.1.1

The MDM software shall be packaged with only [**Certicom Security Builder® GSE-J Crypto Core 2.9.2**].

5.1.5.4 Functionality Testing (MDMPP40:FPT_TST_EXT.1)

MDMPP40:FPT_TST_EXT.1.1

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

MDMPP40:FPT_TST_EXT.1.2

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

5.1.5.5 Trusted Update (MDMPP40:FPT_TUD_EXT.1)

MDMPP40:FPT_TUD_EXT.1.1

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

MDMPP40:FPT_TUD_EXT.1.2

The TSF shall [***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] (database server), implement functionality using [mutually authenticated TLS as defined in the Package for Transport Layer Security] (audit and authentication servers)***] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [***authentication server, [database 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 (database server), implement functionality (audit and authentication servers)***] 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 (database server), implement***

functionality (audit and authentication servers)] to initiate communication via the trusted channel for [**audit server authenticate server, and database server access**].

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(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 [***selection: server***] and remote administrators that is logically distinct from other communication paths and provides assured identification of its endpoints and protection of the communicated data from modification and disclosure.

MDMPP40:FTP_TRP.1.2(1)

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

MDMPP40:FTP_TRP.1.3(1)

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

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

MDMPP40:FTP_TRP.1.1(2)

Refinement: The TSF shall [***implement functionality using [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 Procedures
ALC: Life-cycle support	ALC_CMC.1: Labelling of the TOE
	ALC_CMS.1: TOE CM Coverage
ATE: Tests	ATE_IND.1: Independent Testing Conformance
AVA: Vulnerability assessment	AVA_VAN.1: Vulnerability Survey

Table 5-5 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 -- 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 UEM Server has configurable administrator notifications, including changes in device enrollment status and failure to apply policies to mobile devices. These administrator notifications may be forwarded by email to a configured administrator.
- MDMA10:FAU_ALT_EXT.2: The UEM Android Client can send a range of alerts to the UEM Server 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 UEM Server available once enrolled and if that channel is not available any alerts are queued and forwarded when the channel is re-established. The local file storage is limited only by the space available on the mobile device.

  When the UEM Android Client receives a candidate policy, it 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.

  An administrator can query a device to obtain its current status. If the specified device does not have network connectivity, the UEM Android Client queues the query and delivers it when the device next contacts the UEM Server. This section describes that reachability events can be initiated by the UEM Android Client when it sends any alerts or other messages to the UEM Server and alternately can be initiated by the UEM Server where it can make requests of the UEM Android Client. The reachability status on the UEM Server is based on any secure communication with the UEM Android Client.

- MDMPP40:FAU_GEN.1(1)/ MDMPP40:FAU_GEN.1(2): The UEM Server automatically generates audit records for all required events specified in the SFR without any additional administrator configuration. The minimum set of audit records that can be generated are listed in **Table 5-2** along with the additional information along with:
  - Startup and shutdown of the MDM Server,
  - Administrative commands issued on the MDM Server,
  - Commands issued from the MDM Server to and MDM agent (e.g., MDM Android Client),
  - 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 a date/time stamp, event type and category, user, host, success indicator, and additional information for specific events (indicated in the third column of the **Table 5-2**).

- MDMA10:FAU_GEN.1(2): The UEM Android Client uses the mobile device audit store, so the required audit events (see **Table 5-3**) are recorded directly in the mobile device security log and subject to any filtering configured in the mobile device. Each event in the audit log includes a date/time stamp, event type, subject identity, success indicator, and additional information for specific events (indicated in the third column of the **Table 5-3**).
- MDMPP40:FAU_NET_EXT.1: The UEM Server component of the TOE provides the ability for an administrator to determine the connectivity status of any enrolled agent. The UEM server can display the last check-in time of all of its enrolled devices and the administrator can initiate a check-in command to refresh that information if required.
- MDMPP40:FAU_SAR.1: The UEM Server collects, protects, and can display audit messages to authorized administrators. As the UEM Server generates audit events it stores them within its SQL database. Those events are available to be viewed via the UEM Server administrator portal. Periodically, the audit events stored in the SQL database are also forwarded to a configured SYSLOG server where they can be viewed in the operational environment.

  Note that audit events collected from enrolled mobile devices are received and immediately forwarded to a configured SYSLOG server where they can be examined. The UEM Server does not provide any interface to view these audit events and thus the functions from the environment are invoked to read audit data from the SYSLOG server.
- MDMPP40:FAU_SEL.1: The UEM Server provides a mechanism to allow an authorized administrator to configure the set of events which are actually audited from the set of auditable events. This feature prevents other (not configured) audit events from being generated and stored by the server. The UEM supports filtering based on event type and success/failure of event.
- MDMA10:FAU_SEL.1(2): The UEM Android Client uses the mobile device audit storage and leverages the audit selection capabilities of the mobile device which can filter events based on event type, severity level, user identifier, and success/failure.
- MDMPP40:FAU_STG_EXT.1/MDMPP40:FAU_STG_EXT.2: The UEM Server stores audit data in the same SQL database where most of its configuration data is stored. As such, the UEM Server depends on SQL access restrictions to protect audit data. This audit information can be periodically exported to a secure SYSLOG server using TLS that can optionally be configured for mutual authentication.

6.2 Cryptographic support

The Cryptographic support function satisfies the following security functional requirements:

- MDMPP40:FCS_CKM.1/MDMPP40:FCS_CKM.2: The UEM Server uses its Certicom Security Builder GSE-J Crypto Core (version 2.9.2) to generate asymmetric RSA keys for authentication and key establishment. The UEM Server issues its own RSA 2048-bit certificates and stores them into the UEM Server SQL database so that the can be shared by distributed UEM Server instances.

  The UEM Android Client relies upon the UEM Server for issuance of its certificate. During the UEM Android Client's enrollment process, the UEM Android Client uses its evaluated mobile device platform to generate a RSA 2048-bit key pair and it sends a CSR request to the UEM Server resulting in an issued certificate.

  Both UEM Server and UEM Android client provide ECDSA key generation using P-256, P-384 and P-521 curves in support of ECDHE key establishment.

  The TOE handles decryption errors in accordance with NIST Special Publication 800-56B. The TOE does not reveal the particular error that occurred, either through the contents of any outputted or logged error message or through timing variations.
- MDMPP40:FCS_CKM_EXT.4: The UEM Server clears keys (TLS and HTTPS session keys) from memory after those keys are no longer needed. Furthermore, the UEM Server stores certificates (the only persistently stored keying material) in its SQL database in encrypted format, and when any new certificates are generated or

imported, the UEM Sever will directly overwrite the old keys with the new in the encrypted key store in the SQL database.

The UEM Android Client relies upon its evaluated platform to securely clear keys (TLS and HTTPS session keys) from memory when no longer needed as the UEM Android Client utilizes platform provided TLS and key storage.

- MDMPP40:FCS_COP.1(*):The UEM Server uses Certicom Security Builder GSE-J Crypto Core version 2.9.2 (https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=9718), which provides the algorithms noted in the list below (along with the NIST standards to which they comply).

  The UEM Android client relies upon its evaluated platform to provide cryptographic services for TLS, HTTPS, and certificate-based operations (including hashing, encryption, signature services and keyed hashing).

SFR	Cert #
FCS_CKM.1 – Key Generation *RSA and ECDSA key gen*	
RSA 186-4: Key(gen) – 2048 bit	RSA 2972
ECDSA 186-4: Key(gen) P256, P-384, P-521	ECDSA 1490
FCS_CKM.2 – Key Agreement *KAS ECC schemes; KA Role Initiator and responder*	
KAS ECC: Ephemeral Unified, KaRole(initiator, responder) RSADP: vendor-affirmation	KAS 190
FCS_COP.1(*)	
AES 128/256 CBC	AES 5539
128/256 GCM	AES 5539
RSA SigGen(2048), SigVer(2048)	RSA 2972
ECDSA PKG/PKV/SigGen/SigVer P-256, P384, P-521	ECDSA 1490
SHA-256/384/512	SHS 4445
HMAC SHA-256/384	HMAC 3690
FCS_RBG_EXT.1	
DRBG Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES)	DRBG 2194

Both the UEM Server (which uses its Certicom library) and the UEM Android Client (which relies upon its evaluated platform) generate and verify RSA (using 2048 bits) and ECDSA (using P-256, P-384, and P-521 curves) signatures, perform HMAC-SHA hashing, perform AES encryption and decryption, perform SHA hashing, establish TLS/HTTPS connections, generate IVs, and generate random data. The UEM Android Client invokes platform APIs for TLS connections as well as signature verification where the applicable cryptographic operations are implemented.

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

When using HMAC as part of TLS, both the client and server utilize HMAC keys equal to the block size of the underlying hash algorithm. Thus, when employing HMAC-SHA-256, the TOE uses a 32-byte key to generate a 32-byte hash. Likewise, when employing HMAC-SHA-384, the TOE uses a 48-byte key when performing hashing to produce a 48-byte hash and when employing HMAC-SHA-512, the TOE uses a 48-byte key when performing hashing to produce a 64-byte hash.

In addition to its Certicom library, the UEM Server utilizes its platform (Microsoft Windows Server 2016) for RDP remote access (to the underlying OS) and Trusted Updates of the UEM itself while the client exclusively calls the evaluated Android APIs provided by the underlying phone platform – see references in section **Error! eference source not found.**.

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

- MDMPP40:FCS_IV_EXT.1: The UEM Server generates IVs for AES CBC and AES GCM using unpredictable (random) IVs drawn from the SHA-256 Hash_DRBG (any), HMAC_DRBG (any), and CTR_DRBG (AES) (which meets the "unpredictable" requirement of SP 800-38A and SP800-38D). The UEM Server derives AES CBC IVs as part of the TLS handshake (which also meets the "unpredictable" and "non-repeating" requirements of SP 800-38A and SP800-38D). The UEM Android Client invokes platform APIs for TLS connections where IVs are generated by its platform as required.

- MDMPP40:FCS_RBG_EXT.1 The UEM Server's Certicom library provides a Hash_DRBG (any), HMAC_DRBG (any), and CTR_DRBG (AES) seeded by the underlying platform (Microsoft Windows Server). The Server seeds its DRBG using the Windows BCryptGenRandom() function. Because Microsoft's entropy implementation cannot be tested, an assumption of entropy is made regarding it (as required for any untestable third-party source), specifically 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 UEM Android Client 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 UEM Server uses Certificates to provide an internal PKI which supports the issuance of certificates for its own use (i.e., to identify itself to other parties) and to be sent to mobile devices during enrollment. For the private keys of certificates used by the TOE the UEM Server encrypts these persistent keys by storing them encrypted in its SQL database with PBEWithHmacSHA256AndAES256 (AES-CBC mode) using a 256-bit DEK created during installation. This DEK is stored and protected by the Windows key store. At no time does the Server store any plaintext keys on its hard drive or its SQL database. The Server does not store any ephemeral keys (e.g., TLS/HTTPS session keys). The UEM Android Client stores keys for the certificates it receives from the UEM Server in the platform provided key storage (Android Key Store) and then utilizes those keys securely through the platform provided key storage API. This certificate is used to identify and authenticate the mobile device to the UEM server.

- PKGTLS11:FCS_TLS_EXT.1/PKGTLS11:FCS_TLSC_EXT.1/PKGTLS11:FCS_TLSC_EXT.2/ PKGTLS11:FCS_TLSS_EXT.1/PKGTLS11:FCS_TLSS_EXT.2: The UEM Server supports TLS version 1.2 and supports the ciphersuites listed in sections 5.1.2.15 and 5.1.2.18. All versions of the SSL protocol and older versions of the TLS protocol are refused by the UEM Server.

  The UEM Server performs certificate checking in conformance with FIA_X509_EXT.1 and accepts the use of wildcards in the SAN or CN for all certificates received during a TLS session negotiation. The UEM server performs hostname checking to ensure that the expected hostname matches the certificate Common Name or Subject Alternate Name (when the UEM Server validates the certificate from an LDAP or Syslog server). When

the UEM server is accepting TLS communications from an Agent, the UEM server verifies the UEM Android Client's certificate by ensuring that that the Distinguished Name (DN) in the presented certificate matches a DN in a database of valid, known DNs.

For key exchanges, the UEM server supports DHE key exchanges using 2048-bit RSA certificates and ECDHE key exchanges with EC curves secp256r1, secp384r1, and secp521r1. For communication between the UEM Server and Agents, the TOE also supports not only the DHE and ECDHE previously mentioned, but also RSA 2048-bit key exchanges. The UEM Android Client supports only NIST curves secp256r1, secp384r1, and secp521r1 when using elliptic curve ciphers.

When performing revocation checking, the TOE checks the peer's certificate against an OSCP server to determine if the certificate remains valid. Finally, the TOE will not accept a TLS/HTTPS certificate as valid in the event that the Server cannot contact the revocation server. Neither the UEM Server nor the UEM Android Client utilize certificate pinning.

The UEM Server can be configured with an X509v3 certificate to present during TLS negotiation with an audit (Syslog) server and an X509v3 certificate to present during TLS negotiation with an LDAP server. The certificates used by the UEM server for these communication channels are independent from one another and from the TOE's internal certificates used for communication with mobile devices. The administrator must load the certificate corresponding to each client communication channel, prior to installing the UEM server's client certificate for these channels.

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, a user (or administrator) can log into the UEM Server to issue an activation password. The activation password is in turn used to activate a mobile device over a secure TLS channel between the agent on the mobile device and the UEM Server. The UEM Server, having authenticated to the agent through presentation of its certificate during the TLS handshake, checks that the activation password is valid. An administrator can configure the UEM Server to limit enrollment based on device serial numbers, number of devices, and within a specific time period. Assuming any configured conditions are met, the UEM Server will issue an X509 certificate to the agent. Once the enrollment process has completed, all subsequent connections between the agent and the UEM Server occur through a mutually authenticated TLS session (in which the agent presents its certificate to the server).
- MDMA10:FIA_ENR_EXT.2: During enrollment the UEM Android Client records the unique URL (FQDN or IP address) of the UEM 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 UEM Server allows users to request password recover, choose a language and select the authentication provider before requiring that any user (or administrator) connecting to the server authenticate by providing a username and password. Put another way, a user cannot perform any actions at all (other than logging in, requesting password recover, choosing a language and selecting the authentication provider) until the user successfully authenticates. Furthermore, the UEM Server only allows remote administrators to connect via HTTPS to ensure confidentiality.
- MDMPP40:FIA_X509_EXT.1(1)/MDMPP40:FIA_X509_EXT.2/MDMPP40:FIA_X509_EXT.5: The UEM Server implements the validation and handling of X509 certificates in compliance with the MDMPP40 requirements. The UEM Android Client invokes platform services which validate and handle X.509 certificates also in compliance with the MDMPP40 requirements. The server and client use X.509 certificates for mutual authentication during the establishment of the trusted channel for UEM Server to Agent communication. Both the server and client adhere to the MDMPP40-stipulated rules governing v3 extensions. The server and client 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 UEM Server validates authentication certificates (including the full path) and check their revocation station using 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 server's certificate through any intermediary CAs to a trusted root CA. If the TOE can successfully build the certificate path, then the TOE will next check the validity of the CA certificates (e.g., checking its validity dates and that the CA flag is present in the basic constraints section for all CA certs) in the chain. Assuming the TOE determines that all CA certificates in the chain are valid, the TOE will finally check the revocation status of the server's certificate. The TOE will not accept any certificate for which it cannot determine the validity and will reject the connection attempt.

The UEM Android client invokes the evaluated TLS/HTTPS APIs provided by its platform to provide the TLS protocol and X509 validation whenever the client establishes a trusted channel to the UEM Server.

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

The UEM Android Client components are issued unique X509 certificates from the UEM Server during the enrollment process and they store the certificates in the Android key store (which stores the keys with permissions to only allow the applications themselves to access the keys). When a UEM Android Client component subsequently contacts the UEM Server components, it will utilize the keys in the key store. When a UEM Android Client's keys expire, the mobile device must be re-enrolled to get a new certificate.

The UEM Server components generate a certificate key hierarchy during the installation process and store the resulting certificates in the associated SQL database.

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 UEM Server provides authorized administrators (i.e., an administrator remotely logged into the UEM Server) the ability to perform the required functions specified in the SFR and the ability to apply policies that the UEM clients enforce. Before authenticating to the UEM Server, a user has no ability to perform any functions or to alter policies. The UEM Server also requires that any user attempting to enroll a mobile device authenticate to the server (through a TLS trusted channel).

  The UEM 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 table below identifies management functions that can be configured in device profiles.

  While most security management functions are restricted to an authorized administrator, the authorized administrator can enable mobile device users to enroll their mobile devices. An authorized administrator provides the mobile device user with a username and a password that will allow them to login in order to create an activation password to enroll their devices.

- MDMPP40:FMT_POL_EXT.1/MDMA10:FMT_POL_EXT.2: The UEM Server utilizes TLS (with mutual/client authentication using X509 certificates) as the trusted channel to protect all data transmitted between the UEM Server and the UEM Android Clients. The UEM Android client periodically contacts the UEM Server. Once a trusted channel is established, the UEM server transfers applicable device policies and commands to the UEM Android client. Each device policy is signed by the UEM server using an RSA certificate issued for that purpose. The UEM Android Client checks the signature of each policy is receives in order to ensure it is valid before application to the mobile device. A policy with an invalid signature is not installed.

- MDMPP40:FMT_SAE_EXT.1: The UEM Server can issue activation passwords used to enroll mobile devices. An administrator can configure the time limit after which an activate password will expire and no longer be usable to enroll a mobile device.

- MDMPP40:FMT_SMF.1(1): The UEM 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 UEM clients, which apply and enforce (in conjunction with the mobile device itself) those policies. The table below identifies the management functions implemented for the UEM clients. The scope of supported managed client devices for the evaluation is limited by the set of devices evaluated on the NIAP PCL. Refer to Section 1.4 of [ST] for additional information about the specific evaluated platforms).

Table 6-1 Supported Device Management Commands and Policies

Management Commands and Policies	Samsung Android	iOS
1. transition to the locked state (MDF Function 6)	X	X
2. full wipe of protected data (MDF Function 7)	X	X
3. unenroll from management	X	X
4. install policies	X	X
5. query connectivity status	X	X
6. query the current version of the MD firmware/software	X	X
7. query the current version of the hardware model of the device	X	X
8. query the current version of installed mobile applications	X	X
9. import X.509v3 certificates into the Trust Anchor Database (MDF Function 11)	X	X
10. install applications, (MDF Function 16)	X	X
11. update system software, (MDF Function 15)	X	X
12. remove applications, (MDF Function 14)	X	X
13. remove Enterprise applications (MDF Function 17)	X	X
14. wipe Enterprise data, (MDF Function 28)	X	X
15. remove imported X.509v3 certificates and [no other X.509v3 certificates] in the Trust Anchor Database (MDF Function 12)	X	X
17. import keys/secrets into the secure key storage (MDF Function 9)	X	X
18. destroy imported keys/secrets and [no other keys/secrets] in the secure key storage function (MDF Function 10)	X	X
19. read audit logs kept by the MD (MDF Function 32)	X	
25. password policy: a. minimum password length (MDF Function 1)	X	X
25. password policy: b. minimum password complexity (MDF Function 1)	X	X
25. password policy: c. maximum password lifetime (MDF Function 1)	X	X
26. session locking policy: a. screen-lock enabled/disabled (MDF Function 2)	X	X
26. session locking policy: b. screen lock timeout (MDF Function 2)	X	X

Management Commands and Policies	Samsung Android	iOS
26. session locking policy: c. number of authentication failures (MDF Function 2)	X	X
27. wireless networks (SSIDs) to which the MD may connect (WLAN Client EP Function 2)	X	X
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)	X	
28. security policy for each wireless network: b. ability to specify security type (WLAN Client EP Function 1)	X	X
28. security policy for each wireless network: c. ability to specify authentication protocol (WLAN Client EP Function 1)	X	X
28. security policy for each wireless network: d. specify the client credentials to be used for authentication (WLAN Client EP Function 1)	X	X
29. application installation policy by a. specifying authorized application repository(s) (MDF Function 8)	X	
29. application installation policy by b. specifying a set of allowed applications based on [application name, developer signature] (an application whitelist) (MDF Function 8)	X	
29. application installation policy by c. denying application installation (MDF Function 8)	X	X
30. enable/disable policy for camera across device (MDF Function 5)	X	X
30. enable/disable policy for microphone across device (MDF Function 5)	X	
31. enable/disable policy for the VPN protection a) across MD (MDF Function 3)	X	X
31. enable/disable policy for the VPN protection b) on a per-app basis (MDF Function 3)	X	X
32. enable/disable policy for NFC, Bluetooth, Wi-Fi, and cellular radios (MDF Function 4)	X	
34. enable/disable policy for protocols supporting remote access (MDF Function 25	X	
35. enable/disable policy for developer modes (MDF Function 26)	X	
36. enable policy for data-at rest protection (MDF Function 20)	X	X
37. enable policy for removable media's data-at-rest protection (MDF Function 21)	X	
38. enable/disable policy for local authentication bypass, (MDF Function 27)	X	
40. enable/disable policy for display notification in the locked state of [f. all notifications], (MDF Function 19)		X
47. the unlock banner policy (MDF Function 36)	X	X

Management Commands and Policies	Samsung Android	iOS
48. configure the auditable items (MDF Function 37)	X	
49. enable/disable a. USB mass storage mode (MDF Function 39)	X	
51. enable/disable a. Hotspot functionality with pre-shared key (MDF Function 41)	X	
51. enable/disable b. USB tethering (MDF Function 41)	X	
52. enable/disable location services: a. across device (MDF Function 22)	X	
54. enable/disable policy for the Always-On VPN protection across device (MDF Function 45)	X	
55. enable/disable policy for use of Biometric Authentication Factor (MDF Function 23)	X	X

- MDMPP40:FMT_SMF.1(2): In addition to managing mobile devices, the UEM Server component of the TOE supports the security management functions to configure and manage itself, including configuring a login banner. Among the available security management functions are the ability to configure X.509v3 certificates, manage the device registration process (enrolling specific devices by serial number, limiting the number of devices a user can enroll and controlling the time period available for enrollment). The UEM Server can also configure the administrator login session timeout value.
- MDMPP40:FMT_SMF.1(3): The EMM Server provides the MAS server functionality. Furthermore, in support of application hosting, the UEM Server supports the configuration of application groups assigned to individual applications and devices. Once an individual application is assigned to an app group, if it is marked as "Required", the application is pushed to and installed on the mobile devices assigned to that app group. When assigned to the app group but not required, the users of the assigned mobile devices can use the client to download and install the application.
- MDMA10:FMT_SMF_EXT.4/MDMA10:FMT_UNR_EXT.1: The UEM Android Client component of the TOE is configured with an X.509v3 certificate suitable to facilitate secure communication with the UEM Server. This certificate is provisioned (signed using a certificate issued by the UEM Server) during device enrollment. Once secure communication is enabled and the device is enrolled, the UEM Android Client accepts commands and policies from the enrolling UEM Server and implements those commands and policies (identified above). The administrator can configure (using the UEM Console) the UEM Android Client to prevent the mobile phone's user from removing the client's administrative privileges, thus preventing the user from un-enrolling the client. If an administrator has not restricted the mobile phone user's ability to remove the UEM Android Client's administrative privileges, then the user can remove the UEM Android Client's administrative privileges (un-enrolling it from the UEM Server). Finally, the administrator can forcibly un-enroll the UEM Android Client from the UEM Server.
- MDMPP40:FMT_SMR.1(1)/MDMPP40:FMT_SMR.1(2): The UEM 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 have access to the SQL server database associated with the UEM server. Security configuration administrators log into the UEM Server's HTTPS WebUI and are responsible for configuring the UEM Server's settings. Device user administrators also login through the UEM 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/un-enrolling a mobile device. Finally, auditors (who also login through the UEM Server's HTTPS WebUI) have permissions only to access the UEM Server'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 authenticated (providing a username and password) before gaining any access

to the server. The UEM Server requires that administrator accounts be created for each administrator (and associated with existing roles or custom roles that can be configured by the server primary administrators to have a wide variety of permission combinations), and separates such administrators from mobile device users (unless an administrator has explicitly created a separate administrative account for the user). The UEM Server allows mobile device users to create activation passwords to enroll their mobile devices and thus allows mobile device users to have the UEM Server 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 list of platform APIs used on UEM Server and UEM Android Client is provided in Appendix A below.
- MDMPP40:FPT_ITT.1(2): The UEM Server utilizes TLS (with mutual/client authentication using X509 certificates) as the trusted channel to protect all data transmitted between the UEM Server and the UEM Android Clients. The UEM Server implements TLS, while the UEM Android client invokes platform APIs to utilize the evaluated TLS and x509 functions provided by its platform.
- MDMPP40:FPT_LIB_EXT.1: The TOE utilizes the Certicom Security Builder® GSE-J Crypto Core 2.9.2 for secure TLS communication for each of its secure connections. The TOE also relies upon the MS SQL Driver to provide a communication pathway to the MS SQL server.
- MDMPP40:FPT_TST_EXT.1: The UEM Server performs power-up tests to ensure correct operation, particularly of its cryptographic functions. The UEM Server's Certicom library performs power-up Known Answer Tests for each of its cryptographic algorithms (including AES, RSA, Diffie-Hellman, SHA, and HMAC-SHA) to ensure correct operations. Also, the UEM Server performs a startup integrity check of its executable code verifying that the computed SHA-256 has for TOE software matches the expected value. This ensures TOE integrity (see FPT_TUD_EXT.1 below where the same checks are made for the manifest and other executable files during installation or update). Should any self-test fail the cryptographic module will enter an error state and if the startup integrity check fails the UEM Server will fail to start. The combination of cryptographic tests, integrity tests and blocking startups upon failure is sufficient to prevent the TOE from executing if its software is corrupted.
- MDMPP40:FPT_TUD_EXT.1: The UEM Server provides a page on its administrator portal that displays the version of the UEM Server software. To update the UEM Server, the administrator can (following the Administrator Guidance) obtain a software update, if one is available, and install the update. Updates for the UEM server are distributed in a zip file. Upon unzipping the administrator is supplied with an extractor exe file, data files, a manifest file, signature file, and 3rd party tools. The extractor is a signed executable (using a BlackBerry X.509 public certificate) and the signature is checked by the Windows Server platform. The extractor file is then responsible for checking the signature file, which validates the manifest file and the signatures of the reminder of included files. The checks performed after the extractor tool runs are identical to the integrity checks performed for self-tests on start up.

  Client updates occur like other Android application updates where the signatures are checked by the mobile device prior to installing or updating any installed application.

6.6 TOE access

The TOE access function satisfies the following security functional requirements:

- MDMPP40:FTA_TAB.1: An administrator may configure a login notice to display whenever an administrator accesses the management console (i.e., Web UI). The notice can be configured to inform the administrator or user about any terms and conditions involved with using the interface. When configured to display, the administrator or user must click 'OK' before being allowed to log in.

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 UEM Server implements TLS to secure communication with all enrolled clients or agents (including the Apple iOS agent and UEM Android Client) as well as external audit and LDAP authentication servers. Since the UEM Server provides the MAS server functionality, there is no additional communication path for the MAS audit communications. The TLS for each of these channels is capable of supporting mutual authentication, while mutual authentication is always required for enrolled clients and agents. The UEM Server uses an IPsec channel implemented in its host Windows Server 2016 operating system for all communication with its SQL database server. The UEM Server is a TLS client when communicating with the external audit and LDAP authentication servers.
- MDMPP40:FTP_TRP.1(1): The UEM Server implements HTTPS as its trusted communication path for communications and remote Administrators must connect to the UEM Server using HTTPS (through a normal web browser) to securely administer the UEM Server. The UEM Server provides no other mechanism or method beyond HTTPS for a remote Administrator to configure or access the UEM Server.
- MDMPP40:FTP_TRP.1(2): The UEM Server implements a TLS trusted communication channel for all communications with MD users including enrollment. MD users initiate the communication channel by login via the UEM Android Client (or other client) and thereafter all communications between the Client (on behalf of the MD user) and the UEM Server travel across the secure channel.

Appendix A. Platform APIs Invoked by TOE

The following APIs are used by the TOE to obtain services from the operating environment. All are published interfaces for the OE component being invoked.

UEM Server Platform APIs

- Windows Cryptography API: Next Generation-Bcrypt

UEM Android Client Platform APIs

- Knox APIs
- Java Cryptographic APIs
- Java SSL and X509 APIs

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	UEM Server
MDMPP40:FAU_GEN.1(1)	All[3]
MDMPP40:FAU_GEN.1(2)	UEM Server
MDMPP40:FAU_NET_EXT.1	UEM Server
MDMPP40:FAU_SAR.1	UEM Server
MDMPP40:FAU_SEL.1	All
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	UEM Server
MDMPP40:FCS_IV_EXT.1	All
MDMPP40:FCS_RBG_EXT.1	All
MDMPP40:FCS_STG_EXT.1	All
MDMPP40:FCS_STG_EXT.2	UEM Server
PKGTLS11:FCS_TLS_EXT.1	UEM Server
PKGTLS11:FCS_TLSC_EXT.1	UEM Server
PKGTLS11:FCS_TLSC_EXT.2	UEM Server
PKGTLS11:FCS_TLSC_EXT.3	UEM Server
PKGTLS11:FCS_TLSC_EXT.5	UEM Server
PKGTLS11:FCS_TLSS_EXT.1	UEM Server
PKGTLS11:FCS_TLSS_EXT.2	UEM Server
MDMPP40:FIA_ENR_EXT.1	UEM Server
MDMPP40:FIA_UAU.1	UEM Server
MDMPP40:FIA_X509_EXT.1(1)	UEM Server
MDMPP40:FIA_X509_EXT.2	UEM Server
MDMPP40:FIA_X509_EXT.5	UEM Server
MDMPP40:FMT_MOF.1(1)	UEM Server
MDMPP40:FMT_MOF.1(2)	UEM Server
MDMPP40:FMT_MOF.1(3)	UEM Server
MDMPP40:FMT_POL_EXT.1	UEM Server
MDMPP40:FMT_SAE_EXT.1	UEM Server
MDMPP40:FMT_SMF.1(1)	UEM Server
MDMPP40:FMT_SMF.1(2)	UEM Server
MDMPP40:FMT_SMF.1(3)	UEM Server
MDMPP40:FMT_SMR.1(1)	UEM Server

[3] 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)	UEM Server
MDMPP40:FPT_API_EXT.1	All
MDMPP40:FPT_ITT.1(2)	All
MDMPP40:FPT_LIB_EXT.1	All
MDMPP40:FPT_TST_EXT.1	UEM Server (see TD0438)
MDMPP40:FPT_TUD_EXT.1	UEM Server (see TD0438)
MDMPP40:FTA_TAB.1	UEM Server
MDMPP40:FTP_ITC.1(1)	UEM Server
MDMPP40:FTP_ITC.1(2)	UEM Server
MDMPP40:FTP_ITC_EXT.1	UEM Server
MDMPP40:FTP_TRP.1(1)	UEM Server
MDMPP40:FTP_TRP.1(2)	UEM Server