Requirements from the

Protection Profile for Mobile Device Management

Version: 4.0

2019-04-25

National Information Assurance Partnership

Revision History

Version	Date	Comment

Introduction

Purpose. This document presents the functional and assurance requirements found in the *Protection Profile for Mobile Device Management*. Common Criteria evaluation, facilitated in the U.S. by the National Information Assurance Partnership (NIAP), is required for IA and IA-enabled products in National Security Systems according to CNSS Policy #11.

Using this document. This representation of the Protection Profile includes:

- *Security Functional Requirements* for use in evaluation. These are featured without the formal Assurance Activities specified in the Protection Profile, to assist the reader who is interested only in the requirements.

  It also includes, in tables shown later, particular types of security functional requirements that are not strictly required in all cases. These are:

  - *Selection-based Security Functional Requirements* which become required when certain selections are made inside the regular Security Functionality Requirements (as indicated by the **[selection:]** construct).
  - *Objective Security Functional Requirements* which are highly desired but not yet widely-available in commercial technology.
  - *Optional Security Functional Requirements* which are available for evaluation and which some customers may insist upon.

- *Security Assurance Requirements* which relate to developer support for the product under evaluation, development processes, and other non-functionality security relevant requirements.

Security Functional Requirements

Server Alerts

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. [**selection**: *[**assignment**: Other events]*, *no other events*]

Application Note: An alert can be defined as any form of providing straightaway notice to the administrator. An alert is different from an audit record, however the fact that an alert was sent should be audited per FAU_GEN.1. Email, pop-up notifications, or other methods are acceptable forms of alerts.

The MDM Agent is required to report to the MDM Server on successful application of policies on a managed mobile device, and failures can be inferred from the absence of such alerts. This requirement is intended to ensure that the MDM Server notifies administrators when policies are not properly installed. Failure to properly install policy updates does not affect the enrollment status of the mobile device.

Audit Data Generation

FAU_GEN.1.1(1) **Refinement:** The TSF shall [**selection**: *invoke platform-provided functionality*, *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. [**selection**: *Commands issued to the MDM Agent*, *none*]
d. Specifically defined auditable events listed in
e. [**selection**: *[**assignment**: other events]*, *no other events*].

Application Note: This requirement outlines the events for which an audit record must be generated by either the MDM System or the MDM Server platform. Each of these audit records may be written by the MDM System or may be dispatched to the operating system on which it runs. It is acceptable to select both "invoke platform-provided functionality" and "implement functionality." It should be specified which auditable events are completed by the MDM System and which are completed by the MDM platform.

The ST author can include other auditable events in the assignment; they are not limited to the list presented. All audits must contain at least the information mentioned in FAU_GEN.1.2(1), but may contain more information which can be assigned.

For distributed TOEs each component must generate an audit record for each of the SFRs that it implements. If more than one TOE component is involved when an audit event is triggered, the event has to be audited on each component (e.g. rejection of a connection by one component while attempting to establish a secure communication channel between two components should result in an audit event being generated by both components). This is not limited to error cases but also includes events about successful actions like successful build up/tear down of a secure communication channel between TOE components.

Item a above requires the audtiting of the start-up and shutdown of the given component of the MDM System. If the TOE is distributed, this applies to all components. If the TOE is not distributed then MDM System is equivalent to MDM Server.

Item b above requires all administrative actions to be auditable. Administrative actions refer to any management functions specified by FMT_MOF.1(1). Thus no additional specification for the auditability of these actions is specified in aside from those that require additional record content. If the TOE is distributed and the given component does not deal with setting the policy applied to the MDM Agent, it is acceptable to not have any administrative actions to audit.

Item c includes those commands, which may be performed automatically based on triggers or on a schedule. If the TOE component, if distributed, interacts directly with the MDM Agent, then "Commands issued to an MDM Agent" must be selected. If the TOE component, if distributed, does not interact directly with the MDM Agent, then it is acceptable to select "none".

Depending on the specific requirements selected by the ST author from Security Functional Requirements, Optional Requirements, Selection-Based Requirements, and Objective Requirements, the ST author should include the appropriate auditable event from in the ST for the requirements selected.

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
- [**assignment**: *other audit relevant information*].

Application Note: This requirement outlines the information to be included in audit records. All audits must contain at least the information mentioned in FAU_GEN.1.2(1), but may contain more information which can be assigned. The ST author must identify in the TSS which information of the audit record that is performed by the TSF and that which is performed by the TOE platform.

Network Reachability Review

FAU_NET_EXT.1.1 The TSF shall provide authorized administrators with the capability to read the network connectivity status of an enrolled agent.

Application Note: The MDM Server establishes the network connectivity status of enrolled agents using periodic reachability event alerts from the agents according to FAU_ALT_EXT.2.1 in the MDM Agent PP-Module. This status may be determined by sending an update request from the MDM Server which the Agent is required to respond to or by using scheduled periodic notifications of connectivity initiated by the MDM Agent.

External Trail Storage

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 [**selection**: *store audit data locally*, *no other method*].

Application Note: The TOE must be capable of transmitting audit data to an external entity using a trusted channel as specified in FTP_ITC.1(1) and optionally can store audit data locally. If "store audit data locally" is selected, then FAU_STG_EXT.2.1 must be included in the ST.

This requirement only applies to audit data maintained by the TSF, not audit data that is maintained by the platform. Audit data may include the audit records received from the Agent, in addition to the audit records generated by the MDM Server.

The TOE may rely on a non-TOE audit server for storage and review of audit records. Although the TOE generates audit records and receives audit records from managed mobile devices, the storage of these audit records and the ability to allow the administrator to review these audit records is provided by the operational environment. The TSF may rely on the underlying operating system for this functionality.

Although the audit server is outside of the TOE, the MDM Server should still be able to support mutual authentication. There are no requirements levied on the audit server, but the client (MDM Server) should be able to support TLS client certificate authentication. This way if the non-TOE audit server does support verifying client certs, the MDM Server is in a position to make use of that.

For distributed TOEs each component must be able to export audit data across a protected channel external (FTP_ITC.1) or intercomponent (FPT_ITT.1(1)/FPT_ITT.1(2) or FTP_ITC.1) as appropriate. At least one component of the TOE must be able to export audit records via FTP_ITC.1(1) such that all TOE audit records can be exported to an external IT entity.

Cryptographic Key Generation

FCS_CKM.1.1 **Refinement:** The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to generate **asymmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm [**selection**:

- *RSA schemes using cryptographic key sizes of 2048-bit or greater that meets FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3*,
- *ECC schemes using "NIST curves" P-384 and [**selection**: P-256, P-521, no other curves] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4* ,
- *FFC schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1* ,
- *FFC schemes using Diffie-Hellman group 14 that meet the following: RFC3526, Section 3*,
- *FFC schemes using "safe-prime" groups that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"', and [**selection**: RFC 3526, RFC 7919]*

].

Application Note: The ST author must select all key generation schemes used for key establishment and MDM authentication. When key generation is used for key establishment, the schemes in FCS_CKM.2.1 and selected cryptographic protocols must match the selection. When key generation is used for MDM authentication, the public key is expected to be associated with an X.509v3 certificate.

If the TOE only acts as a receiver in the RSA key establishment scheme, the TOE does not need to implement RSA key generation.

In a distributed TOE, if the TOE component acts as a receiver in the key establishment scheme, the TOE does not need to implement key generation.

Cryptographic Key Establishment

FCS_CKM.2.1 **Refinement:** The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] **to perform** cryptographic **key establishment** in accordance with a specified cryptographic key **establishment** method: [**selection**:

- *RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1:RSA Cryptography Specifications Version 2.1"*,
- *Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"*,
- *Finite field-based key establishment schemes that meets the following: NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"*,
- *Key establishment schemes using Diffie-Hellman group 14 that meets the following: RFC 3526, Section 3*,
- *FFC schemes using "safe-prime" groups that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"', and [**selection**: RFC 3526, RFC 7919]*

].

Application Note: The ST author must select all key establishment schemes used for the selected cryptographic protocols.

The elliptic curves used for the key establishment scheme must correlate with the curves specified in FCS_CKM.1.1.

The domain parameters used for the finite field-based key establishment scheme are specified by the key generation according to FCS_CKM.1.1.

Cryptographic Key Destruction

FCS_CKM_EXT.4.1 The TSF shall destroy plaintext keying material and critical security parameters by [**selection**:

- *invoking platform-provided functionality with the following rules:*
  - *For volatile memory, the destruction shall be executed by [**selection**:*
    - *a single direct overwrite consisting of [**selection**: a pseudo-random pattern using the TSF/Platform RBG (as specified in FCS_RBG_EXT.1), zeroes, ones, a new value of a key, [**assignment**: some value that does not contain any CSP]] ,*
    - *removal of power to the memory,*
    - *destruction of reference to the key directly followed by a request for garbage collection*

    ]
  - *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 [**selection**: single, [**assignment**: ST author defined multi-pass]] direct overwrite consisting of [**selection**: a pseudo-random pattern using the TSF/Platform RBG (as specified in FCS_RBG_EXT.1), zeroes, ones, a new value of a key, [**assignment**: some value that does not contain any CSP]] ,*
    - *instructs the underlying platform to destroy the abstraction that represents the key*

    ]

,

- *implementing key destruction in accordance with the following rules:*
  - *For volatile memory, the destruction shall be executed by a single direct overwrite [**selection**: consisting of a pseudo-random pattern using the TSF/Platform RBG (as specified in FCS_RBG_EXT.1), 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 [**selection**: by a single direct overwrite consisting of zeros followed by a read-verify, by a block erase that erases the reference to memory that stores data as well as the data itself]*
- *For non-volatile flash memory, that is wear-leveled, the destruction shall be executed [**selection**: by a single direct overwrite consisting of zeros , 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*

].

Application Note: The ST author should select "invoking platform-provided functionality" if the MDM Server performs no operations using plaintext secret, private cryptographic keys, and CSPs.

Any security related information (such as keys, authentication data, and passwords) must be zeroized when no longer in use to prevent the disclosure or modification of security critical data.

The zeroization indicated above applies to each intermediate storage area for plaintext key and Cryptographic Service Provider (CSP) (i.e., any storage, such as memory buffers, that is included in the path of such data) upon the transfer of the key/CSP to another location.

Since the TOE does not include the host IT environment, the extent of this capability is necessarily somewhat limited. For the purposes of this requirement, it is sufficient for the TOE to invoke the correct underlying functions of the host to perform the zeroization--it does not imply that the TOE has to include a kernel-mode memory driver to ensure the data are zeroized. It is assumed that the host platform appropriately performs zeroization of key material in its internal processes.

Several selections allow assignment of a 'value that does not contain any CSP'. This means that the TOE uses some other specified data not drawn from a source that may contain key material or reveal information about key material, and not being any of the particular values listed as other selection options. The point of the phrase "does not contain any CSP" is to ensure that the overwritten data is carefully selected, and not taken from a general 'pool' that might contain current or residual data that itself requires confidentiality protection.

FCS_CKM_EXT.4.2 The TSF shall destroy all plaintext keying material and critical security parameters (CSPs) when no longer needed.

Application Note: Key destruction procedures are performed in accordance with FCS_CKM_EXT.4.1. Even if "invoking platform-provided functionality" is selected in FCS_CKM_EXT.4.1, the TSF must determine when the plaintext keying material and CSP are no longer needed and thus should be destroyed. The TSF must "release" the key material and CSP when no longer needed, regardless if the TSF or TOE platform destroys the key material and CSPs.

For the purposes of this requirement, plaintext keying material refers to authentication data, authorization data, secret/private symmetric keys, data used to derive keys, etc.

Cryptographic Operation (Confidentiality Algorithms)

FCS_COP.1.1(1) **Refinement:** The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to perform encryption/decryption in accordance with a specified cryptographic algorithm: [**selection**:

- *AES-CBC (as defined in FIPS PUB 197, and NIST SP 800-38A) mode* ,
- *AES-GCM (as defined in NIST SP 800-38D)* ,
- *AES Key Wrap (KW) (as defined in NIST SP 800-38F)* ,
- *AES Key Wrap with Padding (KWP) (as defined in NIST SP 800-38F)* ,
- *AES-CCM (as defined in NIST SP 800-38C)*

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

Application Note: For the second selection of FCS_COP.1.1(1) , the ST author should choose the mode or modes in which AES operates in the trusted channel protocols. For the third selection, the ST author should choose the key sizes that are supported by this functionality.

Cryptographic Operation (Hashing Algorithms)

FCS_COP.1.1(2) **Refinement:** The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to perform cryptographic hashing in accordance with a specified cryptographic algorithm [**selection**: *SHA-256*, *SHA-384*, *SHA-512*] and message digest sizes [**selection**: *256*, *384*, *512*] bits that meet the following: FIPS Pub 180-4.

Application Note: The intent of this requirement is to specify the hashing function for trusted

channel protocols. The hash selection must support the message digest size selection. The hash selection should be consistent with the overall strength of the algorithm used (for example, SHA-256 for 128-bit keys).

Cryptographic Operation (Signature Algorithms)

FCS_COP.1.1(3) **Refinement:** The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to perform cryptographic signature services (generation and verification) in accordance with a specified cryptographic algorithm [**selection**:

- *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 [**selection**: P-256, P-521, no other curves] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5*

]

Application Note: The ST Author should choose the algorithm implemented to perform digital signatures. The MDM Server must perform digital signatures in accordance with the trusted channel protocols. The MDM Server is required to validate any signed policies and policy updates sent by the MDM Server.

Cryptographic Operation (Keyed-Hash Message Authentication)

FCS_COP.1.1(4) **Refinement:** The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to perform keyed-hash message authentication in accordance with a specified cryptographic algorithm HMAC -[**selection**: *SHA-256*, *SHA-384*, *SHA-512*], key sizes [**assignment**: *key size (in bits) used in HMAC*], and message digest sizes [**selection**: *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."

Application Note: The intent of this requirement is to specify the keyed-hash message authentication function used when used for key establishment purposes for the various cryptographic protocols used by the TOE (e.g., trusted channel). The hash selection must support the message digest size selection. The hash selection should be consistent with the overall strength of the algorithm used for FCS_COP.1(3).

Extended: Random Bit Generation

FCS_RBG_EXT.1.1 The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to perform all deterministic random bit generation services in accordance with NIST Special Publication 800-90A using [**selection**: *Hash_DRBG (any)*, *HMAC_DRBG (any)*, *CTR_DRBG (AES)*].

Application Note: The ST author should select whether the server provides its own DRBG or uses the platforms. SP 800-90A contains three different methods of generating random numbers; each of these, in turn, depends on underlying cryptographic primitives (hash functions/ciphers). The ST author will select the function used, and include the specific underlying cryptographic primitives used in the requirement or in the TSS. While any of the identified hash functions (SHA-224, SHA-256, SHA-384, SHA-512) are allowed for Hash_DRBG or HMAC_DRBG, only AES-based implementations for CTR_DRBG are allowed.

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by an entropy source that accumulates entropy from [**selection**: *a software-based noise source*, *a platform-based RBG*, *a hardware-based noise source*, *no other sources*] with a minimum of [**selection**: *128 bits*, *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.

Application Note: For the first selection in this requirement, the ST author selects 'software-based noise source' if any additional noise sources are used as input to the application's DRBG. Note that the application must use the platform's DRBG to seed its DRBG.

In the second selection in this requirement, the ST author selects the appropriate number of bits of entropy that corresponds to the greatest security strength of the algorithms included in the ST. Security strength is defined in Tables 2 and 3 of NIST SP 800-57A. For example, if the implementation includes 2048-bit RSA (security strength of 112 bits), AES 128 (security strength 128 bits), and HMAC-SHA-256 (security strength 256 bits), then the ST author would select 256 bits.

Cryptographic Key Storage

FCS_STG_EXT.1.1 The TSF shall utilize [**selection**: *platform-provided key storage*, *encryption as specified in FCS_STG_EXT.2*] for all persistent secrets and private keys.

Application Note: This requirement ensures that persistent secrets (credentials, secret keys) and private keys are stored securely when not in use. If some secrets/keys are manipulated by the TOE and others are manipulated by the platform, then both of the selections can be specified by the ST author and the ST author must identify in the TSS those keys which are manipulated by the TOE and those by the platform.

If "encryption as specified in FCS_STG_EXT.2" is selected then FCS_STG_EXT.2 and FCS_IV_EXT.1 must be included in the ST.

If the TSF is an application, and not a dedicated server, then it should store its private keys in the platform-provided key storage.

The ST author is responsible for selecting the manner in which the keys are stored and where they are stored in the selections above.

Enrollment of Mobile Device into Management

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

Application Note: The MDM Server may use its own directory or a directory server to perform the authentication decision for users performing the remote enrollment of a mobile device.

FIA_ENR_EXT.1.2 The TSF shall limit the user's enrollment of devices to devices specified by [**selection**: *IMEI*, *[**assignment**: a unique device ID]*] and [**selection**: *specific device models*, *a number of devices*, *specific time period*, *[**assignment**: other features]*, *no other features*].

Application Note: This requirement is designed to permit the enterprise to restrict users' enrollment of devices. A unique device ID is required to limit the user's enrollment. The unique device ID can be the IMEI or an ID specific to a particular platform.

Timing of Authentication

FIA_UAU.1.1 **Refinement:** The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to allow [**assignment**: *list of TSF mediated actions*] on behalf of the user to be performed before the user is authenticated with the Server.

FIA_UAU.1.2 **Refinement:** The TSF shall [**selection**: *invoke platform-provided functionality*, *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.

Application Note: This requirement ensures that any user attempting to access the TSF must be authenticated. These users may be administrators attempting to administer the TOE or ordinary users attempting to enroll for management by the MDM system. The ST author is responsible for assigning the list of actions that can take place before this authentication. The TSF or TOE platform may utilize enterprise authentication to meet this requirement.

For distributed TOEs at least one TOE component has to support the authentication of administrators but not necessarily all TOE components. In case not all TOE components support authentication for administrators the TSS must describe how administrators are authenticated and identified.

X.509 Certificate Validation

FIA_X509_EXT.1.1(1) The TSF shall [**selection**: *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 [**selection**: *the Online Certificate Status Protocol (OCSP) as specified in RFC 2560*, *a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3*, *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.

Application Note: FIA_X509_EXT.1.1(1) lists the rules for validating certificates. The ST author must select whether revocation status is verified using OCSP or CRLs. FIA_X509_EXT.2 requires that certificates are used for trusted channels; this use requires that the extendedKeyUsage rules are verified. Certificates may optionally be used for code signing and policy signing and, if implemented, must be validated to contain the corresponding extendedKeyUsage.

Regardless of the selection of implement functionality or invoke platform-provided functionality, the validation is expected to end in a trusted root CA certificate in a root store managed by the platform.

FIA_X509_EXT.1.2(1) The TSF shall [**selection**: *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.

Application Note: This requirement applies to certificates that are used and processed by the TOE or platform and restricts the certificates that may be added as trusted CA certificates.

X.509 Certificate Authentication

FIA_X509_EXT.2.1 The TSF shall [**selection**:

- *invoke platform-provided functionality to use X.509v3 certificates as defined by RFC 5280 to support authentication for [**selection**: IPsec, HTTPS, TLS, DTLS, SSH, no protocols], and [**selection**:*
  - *code signing for system software updates,*
  - *code signing for integrity verification,*
  - *policy signing,*
  - *[**assignment**: other uses],*
  - *no additional uses*

  ],
- *implement functionality to use X.509v3 certificates as defined by RFC 5280 to support authentication for [**selection**:*
  - *IPsec as defined in the PP-Module for VPN Client,*
  - *HTTPS in accordance with FCS_HTTPS_EXT.1,*
  - *TLS as defined in the Package for Transport Layer Security,*
  - *DTLS as defined in the Package for Transport Layer Security,*
  - *SSH as defined in the Extended Package for Secure Shell,*
  - *no protocols*

  *], and [**selection**:*
  - *code signing for system software updates,*
  - *code signing for integrity verification,*
  - *policy signing,*
  - *[**assignment**: other uses],*
  - *no additional uses*

  ].

]

Application Note: The ST author's selection(s) must match the selection of FTP_TRP.1(2), FTP_ITC.1(1), FTP_ITC.1(2), FPT_ITT.1(1), and FPT_ITT.1(2). Certificates may optionally be used for trusted updates of system software (FPT_TUD_EXT.1.3) and software integrity verification (FPT_TST_EXT.1.2). If some authentication services are provided by the TOE and others by the platform, the ST author must clearly identify which services are provided by the TOE and which by the platform.

If code signing for integrity verification is selected, the MDM vendor is not expected to digitally sign DLL's from other vendors that have been incorporated into their product.

FIA_X509_EXT.2.2 When the [**selection**: *TSF*, *TOE platform*] cannot establish a connection to determine the validity of a certificate, the TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to [**selection**: *allow the administrator to choose whether to accept the certificate in these cases*, *accept the certificate*, *not accept the certificate*].

Application Note: Often a connection must be established to perform a verification of the revocation status of a certificate - either to download a CRL or to perform OCSP. The selection is used to describe the behavior in the event that such a connection cannot be established (for example, due to a network error). If the TOE has determined the certificate is valid according to all other rules in FIA_X509_EXT.1(1), the behavior indicated in the second selection must determine the validity. The TOE must not accept the certificate if it fails any of the other validation rules in FIA_X509_EXT.1(1). If the administrator-configured option is selected by the ST Author, the ST Author must also select function d in FMT_SMF.1(2).

If the TOE is distributed and FIA_X509_EXT.1(2) is selected, then certificate revocation checking is optional. This is due to additional authorization actions being performed in the enabling and disabling of the intra-TOE trusted channel as defined in FCO_CPC_EXT.1. In this case, a connection is not required to determine certificate validity and this SFR is trivially satisfied.

X.509 Unique Certificate

FIA_X509_EXT.5.1 The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to require a unique certificate for each client device.

Application Note: Each client device will have a unique X.509v3 certificate for use by the MDM Agent; the certificate is not to be reused among clients. This requirement is to ensure that the MDM Server either provides a unique certificate or verifies that each client certificate is unique.

Management of Functions Behavior

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)
- [**selection**: *enable, disable and modify policies listed in FMT_SMF.1(3)*, *no other functions*]

to [authorized administrators].

Application Note: This requirement outlines the functions that administrators will have the power to enable, disable, modify, and monitor functions and policies listed in FMT_SMF.1(1). It also includes functions necessary to maintain and configure the MDM Server itself.

"Enable, disable and modify policies listed in FMT_SMF.1(3)" must be selected if the TOE includes MAS functionality and FMT_SMF.1(3), FAU_GEN.1(2), FMT_MOF.1(3), FMT_SMR.1(2) must be included in the ST.

Management of Functions Behavior (Enrollment)

FMT_MOF.1.1(2) **Refinement:** The **MDM Server** shall restrict the ability to *[initiate the enrollment process]* to *[authorized administrators and MD users]*.

Application Note: This requirement outlines the enrollment functions that both administrators and MD users may perform. The enrollment actions are identified in the TSS as a part of FIA_ENR_EXT.1.

The authorized administrator does not remotely initiate enrollment of the mobile devices that are in the possession of users but may enroll mobile devices when they are in the possession of the administrator, for example, before distributing the mobile devices to the users.

Trusted Policy Update

FMT_POL_EXT.1.1 The TSF shall provide digitally signed policies and policy updates to the MDM Agent.

Application Note: The intent of this requirement is to cryptographically tie the policies to the enterprise that mandated the policy, not to protect the policies in transit (as they are already protected by FPT_ITT.1(1)/FPT_ITT.1(2) or FTP_ITC.1(2)). This is especially critical for users who connect to multiple enterprises.

Specification of Management Functions (Server configuration of Agent)

FMT_SMF.1.1(1) **Refinement:** The **MDM Server** shall be capable of **communicating the following commands to**

the MDM Agent:

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

and the following commands to the MDM Agent:
[**selection**:

- *. remove Enterprise applications (MDF Function 17)*,
- *. wipe Enterprise data (MDF Function 28)*,
- *. remove imported X.509v3 certificates and [**selection**:*
  - *no other X.509v3 certificates,*
  - *[**assignment**: list of other categories of X.509v3 certificates]*

  ] in the Trust Anchor Database (MDF Function 12),
- *. alert the user*,
- *. import keys/secrets into the secure key storage (MDF Function 9)*,
- *. destroy imported keys/secrets and [**selection**:*
  - *no other keys/secrets,*
  - *[**assignment**: list of other categories of keys/secrets]*

  ] in the secure key storage (MDF Function 10),
- *. read audit logs kept by the MD (MDF Function 32)*,
- *. retrieve MD-software integrity verification values (MDF Function 38)*,
- *. approve exceptions for sharing data between [**selection**:*
  - *application processes,*
  - *group of application processes*

  ] (MDF Function 42),
- *. place applications into application process groups based on [**assignment**: application characteristics] (MDF Function 43)*,
- *. revoke Biometric template*,
- *. [**assignment**: list of other management functions to be provided by the MD]* ,
- *no other management functions*

] **and the following MD configuration policies**:

. password policy:
  a. minimum password length
  b. minimum password complexity
  c. maximum password lifetime (MDF Function 1)
. session locking policy:
  a. screen-lock enabled/disabled
  b. screen lock timeout
  c. number of authentication failures (MDF Function 2)
. wireless networks (SSIDs) to which the MD may connect (MDF Function 2)
. security policy for each wireless network:
  a. [**selection**:
     - *specify the CA(s) from which the MD will accept WLAN authentication server certificate(s)*,
     - *specify the FQDN(s) of acceptable WLAN authentication server certificate(s)*

     ]
  b. ability to specify security type
  c. ability to specify authentication protocol
  d. specify the client credentials to be used for authentication
  e. [**assignment**: *any additional WLAN management functions*] (WLAN Client PP-Module Function 1)
. application installation policy by [**selection**:
  - *specifying authorized application repository(s)*,
  - *specifying a set of allowed applications and versions (an application whitelist)*,
  - *denying application installation*

], (MDF Function 8)
. enable/disable policy for [**assignment**: *list of audio or visual collection devices*] across device, [**selection**:
  - *on a per-app basis*,
  - *on a per-group of applications processes basis* ,
  - *no other method*

], (MDF Function 5)

and the following MD configuration policies:

[**selection**:
  - *. enable/disable policy for the VPN protection across MD, [**selection**:*

- *on a per-app basis*,
- *on a per-group of application processes basis* ,
- *no other method*

] (MDF Function 3),

- *. enable/disable policy for [**assignment**: list of radios], (MDF Function 4)*,
- *. enable/disable policy for data signaling over [**assignment**: list of externally accessible hardware ports], (MDF Function 24)*,
- *. enable/disable policy for [**assignment**: list of protocols where the device acts as a server], (MDF Function 25)*,
- *. enable/disable policy for developer modes, (MDF Function 26)*,
- *. enable policy for data-at-rest protection, (MDF Function 20)*,
- *. enable policy for removable media's data-at-rest protection, (MDF Function 21)*,
- *. enable/disable policy for local authentication bypass, (MDF Function 27)*,
- *. the Bluetooth trusted channel policy:*
  a. *enable/disable the Discoverable mode (for BR/EDR)*
  b. *change the Bluetooth device name, [**selection**:*
     - *allow/disallow additional wireless technologies to be used with Bluetooth* ,
     - *disable/enable Advertising (for LE),*
     - *disable/enable the Connection mode,*
     - *disable/enable the Bluetooth services and/or profiles available on the device* ,
     - *specify minimum level of security for each pairing*,
     - *configure allowable methods of Out of Band pairing* ,
     - *no other Bluetooth configuration*

  ](MDF Function 18)

,

- *. enable/disable policy for display notification in the locked state of*
  *[**selection**:*
  - *email notifications,*
  - *calendar appointments,*
  - *contact associated with phone call notification*,
  - *text message notification*,
  - *other application-based notifications*,
  - *none*

  ](MDF Function 19)

,

- *. policy for establishing a trusted channel or disallowing establishment if the MD cannot establish a connection to determine the validity of a certificate, (MDF Function 30)*,
- *. enable/disable policy for the cellular protocols used to connect to cellular network base stations, (MDF Function 31)*,
- *. policy for import and removal by applications of X.509v3 certificates in the Trust Anchor Database, (MDF Function 29)*,
- *. [**selection**:*
  - *certificate,*
  - *public-key*

  ] used to validate digital signature on applications, (MDF Function 33) ,
- *. policy for exceptions for shared use of keys/secrets by multiple applications, (MDF Function 34)*,
- *. policy for exceptions for destruction of keys/secrets by applications that did not import the key/secret, (MDF Function 35)*,
- *. the unlock banner policy, (MDF Function 36)*,
- *. configure the auditable items (MDF Function 37)*,
- *. enable/disable [**selection**:*
  - *USB mass storage mode,*
  - *USB data transfer without user authentication,*
  - *USB data transfer without authentication of the connection system*

  ] (MDF Function 39) ,
- *. enable/disable backup of [**selection**:*
  - *all applications,*
  - *selected applications,*
  - *selected groups of applications,*
  - *configuration data*

  *] to [**selection**: locally connected system, remote system] (MDF Function 40)*,
- *. enable/disable [**selection**:*
  - *Hotspot functionality authenticated by [**selection**: pre-shared key, passcode, no authentication]* ,
  - *USB tethering authenticated by [**selection**: pre-shared key, passcode, no authentication]*

  ](MDF Function 41) ,
- *. enable/disable location services:*
  *[**selection**:*
  - *across device,*
  - *on a per-app basis*,
  - *on a per-group of application processes basis* ,
  - *no other method*

  ](MDF Function 22) ,
- *. enable/disable policy for user unenrollment*,
- *. enable/disable policy for the Always-On VPN protection across device (MDF Function 45)*,
- *. enable/disable policy for use of Biometric Authentication Factor (MDF Function 23)*,
- *. Connectivity timeout policy: [**selection**:*

        - *allowed [**selection**: number of missed reachability events, length of time without server connectivity] ,*
        - *when server connectivity timeout is exceeded agent shall [**selection**: disable user password, wipe device] and [**selection**: [**assignment**: other action], none]]* ,
    - *. enable/disable multi-user modes*,
    - *. enable/disable automatic updates of system software*,
    - *. enable/disable removable media*,
    - *. [**assignment**: list of other policies to be provided by the MD], no other policies].*

]

Application Note: This requirement captures all the configuration functionality the TSF provides the administrator to configure the MDM Agent. This requirement is broken into two configurable areas: MDM Agent commands and MDM Agent policies. The ST author can add more commands and configuration policies by completing the appropriate assignment statements.

The ST author must not claim any functionality not provided by the Mobile Device. All selections and assignments performed by the ST author in this requirement should match the selections and assignments of the validated Mobile Device ST.

Function-specific Application Notes:

Function-specific application notes reference Mobile Device Fundamentals (MDF) PP v3.1.

Function may be satisfied for the BYOD use case by application blacklisting and/or disabling. In the case of BYOD, an enterprise may not want to remove "personal" applications, thus for that use case disabling the application rather than removing it would allow the user to not lose any information they might have in the application.

Function provides the MDM server to display an alert to the user of the mobile device.

The audit records read according to Function are to be transmitted to an external audit server according to FAU_STG_EXT.1. The MDM Server is not expected to retain those logs.

Function provides the ability to enable/disable policy for the list of protocols where the device acts as a server, such as a mobile hotspot.

Function corresponds to FPT_NET_EXT.1.1 in Agent. If the MDM Agent has not had a successful reachability event with the MDM Server in the amount of time specified in 'a', then the agent must perform the action selected in 'b'. It is feasible, but not required, that if multiple actions are selected in 'b', each action could be associated with a different timeout in 'a'. If function is included in the ST, than FPT_NET_EXT.1.1 must be included in the Agent ST.

Specification of Management Functions (Server Configuration of Server)

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 [**selection**:
    - *devices specified by [**selection**: IMEI, [**assignment**: a unique device ID]]*,
    - *specific device models*,
    - *a number of devices*,
    - *specific time period*

   ] and [**selection**: *[**assignment**: other features]*, *no other features*] allowed for enrollment
c. [**selection**:
   1. allow the administrator to choose whether to accept the certificate when connection cannot be made to establish validity,
   2. configure the TOE unlock banner,
   *3. configure periodicity of the following commands to the agent: [**assignment**: list of commands]*,
   4. configure the privacy-sensitive information that will and will not be collected from particular mobile devices,
   5. configure the length of time the enrollment authenticator is valid,
   6. configure the interaction between TOE components,
   7. configure the cryptographic functionality,
   *8. [**assignment**: other management functions]*,
   9. no other management functions]

Application Note: This requirement captures all the configuration functionality in the MDM Server to configure the underlying MDM Server. The ST author can add more commands and configuration policies by completing the assignment statement.

Function a can be met by relying on the platform to configure the certificates used by the MDM server, however, the MDM Server must allow the administrator to choose which certificate is used for a specific functionality. The selection in b corresponds to the selection in FIA_ENR_EXT.1.2.

The selection in c.1 includes a function that corresponds to the selection in FIA_X509_EXT.2.2. Function c.3 allows the administrator to configure periodicity of assigned commands, for example "read audit logs kept by the Mobile Device". In this way the administrator can configure the MDM system to retrieve audit logs from the Mobile Device on a periodic, such as daily, basis in order to ensure freshness of log data and to minimize loss of audit logs. Function c.4 allows the administrator to configure the privacy-sensitive information that will and will not be collected from particular mobile devices to handle BYOD environments where some information may not be appropriate to collect. Privacy sensitive information may include items such as device physical location and lists of installed personal applications, and would vary depending on the particular capabilities of the TOE and MDM agent. The TOE should provide the capability to group enrolled devices into categories such as enterprise-owned and personal-owned and define the information that will be collected from devices in each category. Function c.5 corresponds to configuring the length of time the user authenticator for enrollment is valid in FMT_SAE_EXT.1. This function must be included in the ST if and only if FMT_SAE_EXT.1 is included in the ST.

For distributed TOEs the interaction between TOE components will be configurable (see FCO_CPC_EXT.1). Therefore, the ST author includes the selection "Ability to configure the interaction between TOE components" for distributed TOEs. A simple example would be the change of communication protocol according to FPT_ITT.1(1). Another example would be changing the management of a TOE component from direct remote administration to remote administration through another TOE component. A more complex use case would be if the realization of an SFR is achieved through two or more TOE components and the responsibilities between the two or more components could be modified.

For distributed TOEs that implement a registration channel (as described in FCO_CPC_EXT.1.2), the ST author uses the selection "configure the cryptographic functionality" in this SFR, and its corresponding mapping in the TSS, to describe the configuration of any cryptographic aspects of the registration channel that can be modified by the operational environment in order to improve the channel security.

Security Management Roles

FMT_SMR.1.1(1) **Refinement:** The TSF shall maintain the roles administrator, MD user, and [**selection**: *[**assignment**: additional authorized identified roles]*, *no additional roles*].

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

Application Note: It is envisioned that the MDM Server will be configured and maintained by different user roles. The assignment is used by the ST author to list the roles that are supported. At a minimum, one administrative role must be supported. If no additional roles are supported, then "no additional roles" is selected. The MD user role is used for enrollment of mobile devices to the MDM according to FIA_ENR_EXT.1.

For distributed TOEs, not every TOE component is required to implement its own user management to fulfill this SFR. At least one component has to support authentication and identification of users according to FIA_UAU.1. For the other TOE components authentication can be realized through the use of a trusted channel (either according to FTP_ITC.1 or FPT_ITT.1(1)/FPT_ITT.1(2)) from a component that supports the authentication of users according to FIA_UAU.1. The identification of users according to FIA_UAU.1.2 and the association of users with roles according to FMT_SMR.1.2(1) is done through the components that support the authentication of users according to FIA_UAU.1.

Use of Supported Services and APIs

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

Application Note: This requirement applies to the APIs used when "invoke platform provided functionality" is selected in an SFR. The definition of *documented* may vary depending upon whether the MDM software is provided by a third party (who relies upon documented platform APIs) or by a platform vendor who may be able to guarantee support for platform API's.

Use of Third Party Libraries

FPT_LIB_EXT.1.1 The MDM software shall be packaged with only [**assignment**: *list of third-party libraries*].

Application Note: This requirement applies to libraries used when "implement functionality" is selected in an SFR. The intention of this requirement is for the evaluator to document which software libraries the MDM software is including in case vulnerabilities are later discovered with those libraries.

Functionality Testing

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.

FPT_TST_EXT.1.2 The TSF shall [**selection**: *invoke platform-provided functionality*, *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 [**selection**: *TSF*, *TOE platform*]-provided cryptographic services.

Application Note: While the TOE is typically a software package running in the IT Environment, it is still capable of performing the self-test activities required above. It should be understood, however, that there is a significant dependency on the host environment in assessing the assurance provided by the tests mentioned above (meaning that if the host environment is compromised, the self-tests will not be meaningful).

For distributed TOEs all TOE components (except the MDM Agent components) have to perform self-tests. This does not necessarily mean that each TOE component has to carry out the same self-tests: the ST describes the applicability of the selection (i.e. when self-tests are run) and the final assignment (i.e. which self-tests are carried out) to each TOE component.

Trusted Update

FPT_TUD_EXT.1.1 The TSF shall provide Authorized Administrators the ability to query the current version of the software.

Application Note: For a distributed TOE, the method of determining the installed versions on each component of the TOE is described in the operational guidance. In the requirement, "software" refers to the component of the distributed TOE to which the requirement is being applied.

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

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

Application Note: The software on the TSF will occasionally need to be updated. This requirement is intended to ensure that the TSF only installs updates provided by the vendor, as updates provided by another source may contain malicious code. If the server is not an appliance, the update will be verified by the platform on which the server software runs. If the server is an appliance, the update must be verified by the TSF software or hardware.

For distributed TOEs all TOE components must support Trusted Update. The verification of the signature or hash on the update must either be done by each TOE component itself (signature verification) or for each TOE component (hash verification).

Updating a distributed TOE might lead to the situation where different TOE components are running different software versions. Depending on the differences between the different software versions the impact of a mixture of different software versions might be no problem at all or critical to the proper functioning of the TOE. The TSS must detail the mechanisms that support the continuous proper functioning of the TOE during trusted update of distributed TOEs.

Trusted Channel

FTP_ITC_EXT.1.1 The TSF shall provide a communication channel between itself and [**selection**:

- *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 [**selection**: *FPT_ITT.1(1)*, *FPT_ITT.1(2)*, *FTP_ITC.1(2)*].

Inter-TSF Trusted Channel (Authorized IT Entities)

FTP_ITC.1.1(1) **Refinement:** The TSF shall [**selection**:

- *invoke platform-provided functionality to use [**selection**:*
  - *IPsec,*

    - *SSH,*
    - *mutually authenticated TLS,*
    - *mutually authenticated DTLS,*
    - *HTTPS*

  ],
- *implement functionality using [**selection**:*
    - *IPsec as defined in the PP-Module for VPN Client,*
    - *SSH as defined in the Extended Package for Secure Shell,*
    - *mutually authenticated TLS as defined in the Package for Transport Layer Security,*
    - *mutually authenticated DTLS as defined in the Package for Transport Layer Security,*
    - *HTTPS in accordance with FCS_HTTPS_EXT.1*

  ]

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

Application Note: The intent of the mandatory portion of the above requirement is to use the cryptographic protocols identified in the requirement to establish and maintain a trusted channel with authorized IT entities that the TOE interacts with to perform its functions.

Protection (by one of the listed protocols) is required at least for communications with the server that collects the audit information. If it communicates with an authentication server (e.g., RADIUS), then the ST author chooses "authentication server" in FTP_ITC.1.1(1) and this connection must be protected by one of the listed protocols. If other authorized IT entities (e.g., NTP server) are protected, the ST author makes the appropriate assignments (for those entities) and selections (for the protocols that are used to protect those connections).

To summarize, the connection to an external audit collection server is required to be protected by one of the listed protocols. If an external authentication server is supported, then it is required to protect that connection with one of the listed protocols. For any other external server, external communications are not required to be protected, but if protection is claimed, then it must be protected with one of the identified protocols.

For communications with any authorized IT entities outside of the TOE, the MDM Server should still be able to support mutual authentication. There are no requirements levied on the external entities, but the MDM Server should be able to support mutual authentication. This way if the non-TOE authorized entity does support mutual authentication, the MDM Server is in a position to make use of that.

The trusted channel uses IPsec, TLS, DTLS, or HTTPS as the protocol that preserves the confidentiality and integrity of MDM communications. The ST author chooses the mechanism or mechanisms supported by the TOE.

If "IPsec as defined in the PP-Module for VPN Client" is selected, the TSF must claim conformance to a PP-Configuration that includes the VPN Client PP-Module.

If the ST author selects "SSH as defined in the Extended Package for Secure Shell", the TSF must be validated against the EP for Secure Shell with the MDM PP. It should be noted that due to constraints imposed by this PP that sha1 cannot be used.

If the ST author selects "mutually authenticated TLS as defined in the Package for Transport Layer Security" or "mutually authenticated DTLS as defined in the Package for Transport Layer Security", the TSF must be validated against requirements from the Package for Transport Layer Security, with the following selections made:

- FCS_TLS_EXT.1:
    - either TLS or DTLS is selected depending on the selection made in FTP_ITC.1.1(1)
    - either client or server is selected as appropriate
- FCS_TLSC_EXT.1.1 or FCS_TLSS_EXT.1.1 (as appropriate):
    - The cipher suites selected must correspond with the algorithms and hash functions allowed in FCS_COP.1.
    - mutual authentication must be selected
- FCS_DTLSC_EXT.1.1 or FCS_DTLSS_EXT.1.1 (as appropriate):
    - The cipher suites selected must correspond with the algorithms and hash functions allowed in FCS_COP.1.
    - mutual authentication must be selected

Protocol, RBG, Certificate validation, algorithm, and similar services may be met with platform provided services.

The requirement implies that not only are communications protected when they are initially established, but also on resumption after an outage. It may be the case that some part of the TOE setup involves manually setting up tunnels to protect other communication, and if after an outage the TOE attempts to re-establish the communication automatically with (the necessary) manual intervention, there may be a window created where an attacker might be able to gain critical information or compromise a connection.

FTP_ITC.1.2(1)	**Refinement:** The TSF shall [**selection**: *invoke platform-provided functionality*, *implement*

functionality] to permit the MDM Server or other authorized IT entities to initiate communication via the trusted channel.

FTP_ITC.1.3(1) **Refinement:** The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to initiate communication via the trusted channel for [**assignment**: *list of services for which the TSF is able to initiate communications*].

Application Note: While there are no requirements on the party initiating the communication, the ST author lists in the assignment for FTP_ITC.1.3(1) the services for which the TOE can initiate the communication with the authorized IT entity.

Trusted Path (for Remote Administration)

FTP_TRP.1.1(1) **Refinement:** The TSF shall [**selection**:

- *invoke platform-provided functionality to use [**selection**:*
  - *IPsec,*
  - *TLS,*
  - *HTTPS,*
  - *SSH*

  ],
- *implement functionality using [**selection**:*
  - *IPsec as defined in the PP-Module for VPN Client,*
  - *TLS as defined in the Package for Transport Layer Security,*
  - *HTTPS in accordance with FCS_HTTPS_EXT.1,*
  - *SSH as defined in the Extended Package for Secure Shell*

  ]

] to provide a trusted communication path between itself as a [**selection**: *server*, *peer*] 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, disclosure].

FTP_TRP.1.2(1) **Refinement:** The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to permit remote administrators to initiate communication via the trusted path.

FTP_TRP.1.3(1) **Refinement:** The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to require the use of the trusted path for [all remote administration actions].

Application Note: This requirement ensures that authorized remote administrators initiate all communication with the TOE via a trusted path, and that all communications with the TOE by remote administrators is performed over this path. The data passed in this trusted communication channel are encrypted as defined in the protocol chosen in the first selection. The ST author chooses the mechanism or mechanisms supported by the TOE.

If "IPsec as defined in the PP-Module for VPN Client" is selected, the TSF must claim conformance to a PP-Configuration that includes the VPN Client PP-Module.

If the ST author selects "SSH as defined in the Extended Package for Secure Shell", the TSF must be validated against the EP for Secure Shell with the MDM PP. It should be noted that due to constraints imposed by this PP that sha1 cannot be used.

If the ST author selects "TLS as defined in the Package for Transport Layer Security" the TSF must be validated against requirements from the Package for Transport Layer Security, with the following selections made:

- FCS_TLS_EXT.1:
  - TLS shall be selected
  - server shall be selected
- FCS_TLSS_EXT.1.1:
  - The cipher suites selected must correspond with the algorithms and hash functions allowed in FCS_COP.1.

Protocol, RBG, Certificate validation, algorithm, and similar services may be met with platform provided services.

Trusted Path (for Enrollment)

FTP_TRP.1.1(2) **Refinement:** The TSF shall [**selection**:

- *invoke platform-provided functionality to use [**selection**:*
  - *TLS,*
  - *HTTPS*

  ],
- *implement functionality using [**selection**:*

- *TLS as defined in the Package for Transport Layer Security,*
- *HTTPS in accordance with FCS_HTTPS_EXT.1*

]

] to provide a trusted communication path between itself (as a server) and MD users that is logically distinct from other communication paths and provides assured identification of its endpoints and protection of the communicated data from disclosure and detection of modification of the communicated data from [modification, disclosure].

FTP_TRP.1.2(2)	**Refinement:** The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to permit MD users to initiate communication via the trusted path.

FTP_TRP.1.3(2)	**Refinement:** The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to require the use of the trusted path for [all MD user actions].
	
	Application Note: This requirement ensures that authorized MD users initiate all communication with the TOE via a trusted path, and that all communications with the TOE by MD users is performed over this path. The purpose of this connection is for enrollment by the MD user. The data passed in this trusted communication channel are encrypted as defined in the protocol chosen in the first selection. The ST author chooses the mechanism or mechanisms supported by the TOE.
	
	If the ST author selects "TLS as defined in the Package for Transport Layer Security" the TSF must be validated against requirements from the Package for Transport Layer Security, with the following selections made:
	
	• FCS_TLS_EXT.1:
	  ◦ TLS must be selected
	  ◦ server must be selected
	• FCS_TLSS_EXT.1.1:
	  ◦ The cipher suites selected must correspond with the algorithms and hash functions allowed in FCS_COP.1.
	
	Protocol, RBG, Certificate validation, algorithm, and similar services may be met with platform provided services.

Security Assurance Requirements

ASE_TSS.1.1C	The TOE summary specification shall describe how the TOE meets each SFR.
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. **Application Note:** As indicated in the introduction to this section, the functional specification is comprised of the information contained in the AGD_OPE and AGD_PRE documentation. The developer may reference a website accessible to application developers and the evaluator. The evaluation activities in the functional requirements point to evidence that should exist in the documentation and TSS section; since these are directly associated with the SFRs, the tracing in element ADV_FSP.1.2D is implicitly already done and no additional documentation is necessary.
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.
AGD_OPE.1.1D	The developer shall provide operational user guidance. **Application Note:** The operational user guidance does not have to be contained in a single

	document. Guidance to users, administrators and application developers can be spread among documents or web pages. Where appropriate, the guidance documentation is expressed in the eXtensible Configuration Checklist Description Format (XCCDF) to support security automation. Rather than repeat information here, the developer should review the evaluation activities for this component to ascertain the specifics of the guidance that the evaluator will be checking for. This will provide the necessary information for the preparation of acceptable 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. **Application Note:** User and administrator are to be considered in the definition of user role.
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.
AGD_PRE.1.1D	The developer shall provide the TOE, including its preparative procedures. **Application Note:** As with the operational guidance, the developer should look to the evaluation activities to determine the required content with respect to 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. **Application Note:** It is recognised that the application of these requirements will vary depending on aspects such as whether the TOE is delivered in an operational state, or whether it has to be installed at the TOE owner's site, etc. It might also be the case that no installation is necessary, for example as a Software as a Service implementation in a Cloud environment. In this case it may be inappropriate to require and analyse installation procedures and thus this requirement is implicitly satisfied.
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.
ALC_CMC.1.1D	The developer shall provide the TOE and a reference for the TOE.
ALC_CMC.1.1C	The TOE shall be labeled 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.
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. **Application Note:** In cases where the MDM software is Software as a Service, running in a cloud environment where they have little to no control of the operating system and underlying hardware, the evaluated configuration is considered a snapshot of the MDM software with the OS and/or VM versions used at the time of testing.
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.
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.

Selection-Based Security Functional Requirements

Audit Generation (MAS Server)

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. **Application Note:** The MDM Agent is required to report to the MAS Server on successful receipt of an application or update on a managed mobile device, and failures can be inferred from the absence of such alerts.
FAU_GEN.1.2(2)	**Refinement:** The [**selection**: *MAS Server*, *MAS Server platform*] shall record within each TSF audit record at least the following information: • date and time of the event • type of event • mobile device identity • [**assignment**: *other audit relevant information*] **Application Note:** All audits must contain at least the information mentioned in FAU_GEN.1.2(2), but may contain more information which can be assigned. The ST author must identify in the TSS which information of the audit record that is performed by the TSF and that which is performed by the TOE platform.

Audit Event Storage

FAU_STG_EXT.2.1 The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to protect the stored audit records in the audit trail from unauthorized modification.

Application Note: If "store audit data locally" is selected in FAU_STG_EXT.1.1, this SFR shall be included in the ST.

The purpose of this requirement is to ensure that audit records are stored securely. The ST author is responsible for selecting whether audit records are maintained when audit storage or failure occurs. The ST author must choose a means by which audit records are saved and select the events during which the records will be saved. The TSF may rely on the underlying operating system for this functionality, and the first selection should be made appropriately.

HTTPS Protocol

FCS_HTTPS_EXT.1.1 The TSF shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS in accordance with the Package for Transport Layer Security .

Application Note: The TLS Functional Package must be included in the ST, with the following selections made:

- FCS_TLS_EXT.1:
  - TLS must be selected
  - either client or server is selected as appropriate
- FCS_TLSC_EXT.1.1 or FCS_TLSS_EXT.1.1 (as appropriate):
  - The cipher suites selected must correspond with the algorithms and hash functions allowed in FCS_COP.1.

Validity is determined by the certificate path, the expiration date, and the revocation status in accordance with RFC 5280.

Initialization Vector Generation

FCS_IV_EXT.1.1 The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to generate IVs in accordance with .

Application Note: This requirement must be included in the ST if the selection in FCS_STG_EXT.1 indicates that the TSF is protecting private keys and persistent secrets with encryption rather than the platform-provided key storage.

lists the requirements for composition of IVs according to the corresponding NIST Special Publications for each cipher mode. The composition of IVs generated for encryption according to a cryptographic protocol is addressed by the protocol. Thus, this requirement addresses only IVs generated for key storage encryption.

: References and IV Requirements for NIST-approved Cipher Modes

Cipher Mode	Reference	IV Requirement
Electronic Codebook (ECB)	SP800-38A	No IV
Counter (CTR)	SP800-38A	"Initial Counter" shall be non-repeating. No counter value shall be repeated across multiple messages with the same secret key.
Cipher Block Chaining (CBC)	SP800-38A	IVs shall be unpredictable. Repeating IVs leak information about whether the first one or more blocks are shared between two messages, so IVs should be non-repeating in such situations.
Output Feedback (OFB)	SP800-38A	IVs shall be non-repeating and shall not be generated by invoking the cipher on another IV.
Cipher Feedback (CFB)	SP800-38A	IVs should be non-repeating as repeating IVs leak information about the first plaintext block and about common shared prefixes in messages.
XEX (XOR Encrypt XOR) Tweakable Block Cipher with Ciphertext Stealing	SP800-38E	No IV. Tweak values shall be non-negative integers, assigned consecutively, and starting at an arbitrary non-negative integer.

(XTS)		
Cipher-based Message Authentication Code (CMAC)	SP800-38B	No IV
Key Wrap and Key Wrap with Padding	SP800-38F	No IV
Counter with CBC-Message Authentication Code (CCM)	SP800-38C	No IV. Nonces shall be non-repeating.
Galois Counter Mode (GCM)	SP800-38D	IV shall be non-repeating. The number of invocations of GCM shall not exceed 2^32 for a given secret key unless an implementation only uses 96-bit IVs (default length).

Encrypted Cryptographic Key Storage

FCS_STG_EXT.2.1 The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to encrypt all keys using AES in the [**selection**: *Key Wrap (KW) mode*, *Key Wrap with Padding (KWP) mode*, *GCM*, *CCM*, *CBC mode*] .

Application Note: This requirement states that keys used by the TSF shall not be kept in plaintext. The intent of this requirement is to ensure that the private keys, credentials, and persistent secrets cannot be accessed in the TOE in an unencrypted state, allowing an attacker to access keys without having to exhaust the AES key space.

This requirement must be including in the ST if the selection in FCS_STG_EXT.1 indicates that the TSF is protecting private keys and persistent secrets with encryption rather than the platform-provided key storage.

X.509 Certificate Validation

FIA_X509_EXT.1.1(2) The TSF shall [**selection**: *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 [**selection**: *the Online Certificate Status Protocol (OCSP) as specified in RFC 2560*, *a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3*, *a Certificate Revocation List (CRL) as specified in RFC 5759 Section 5*, *no revocation method*].
- 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.

Application Note: FIA_X509_EXT.1.1(2) should be chosen if the TOE is distributed and the protocol(s) selected in FPT_ITT.1(1) utilize X.509 certificates for peer authentication. In this case, the use of revocation list checking is optional as there are additional requirements surrounding the enabling and disabling of the ITT channel as defined in FCO_CPC_EXT.1. If revocation checking is not supported, the ST author should select no revocation method. However, if certificate revocation checking is supported, the ST author selects whether this is performed using OCSP or CRLs.

The TSS must describe when revocation checking is performed. It is expected that revocation checking is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device.

This SFR lists the rules for validating certificates. The ST author must select whether revocation status is verified using OCSP or CRLs. FIA_X509_EXT.2 requires that certificates are used for trusted channels; this use requires that the extendedKeyUsage rules are verified. Certificates may optionally be used for code signing and policy signing and, if implemented, must be validated to contain the corresponding extendedKeyUsage.

Regardless of the selection of implement functionality or invoke platform-provided functionality, the validation is expected to end in a trusted root CA certificate in a root store managed by the platform.

FIA_X509_EXT.1.2(2) The TSF shall [**selection**: *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.

Application Note: This requirement applies to certificates that are used and processed by the TOE or platform and restricts the certificates that may be added as trusted CA certificates.

Management of Functions in (MAS Server Downloads)

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.

Specification of Management Functions (MAS Server)

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. [**selection**: *[**assignment**: other MAS management functions]*, *no other functions*]

Application Note: This requirement captures all the configuration functionality in the MAS Server to configure the underlying MAS Server. The ST author can add more commands and configuration policies by completing the assignment statement.

The MAS Server must be able to create groups to configure which applications a user can access based on which group they are in. If the MAS Server uses the groups defined by the MDM, then it must communicate with the MDM Server (if separate server) to determine which applications the user can access.

Security Management Roles (MAS Server)

FMT_SMR.1.1(2) **Refinement:** The TSF shall additionally maintain the roles enrolled mobile devices, application access groups, and [**assignment**: *additional authorized identified roles*].

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

Application Note: It is envisioned that the MAS Server will be configured and maintained by different user roles. The assignment is used by the ST author to list the roles that are supported. At a minimum, one administrative role must be supported. If no additional roles are supported, then "no additional roles" is stated. The MD user role is used for enrollment of mobile devices to the MAS according to FIA_ENR_EXT.1.

Internal TOE TSF Data Transfer

FPT_ITT.1.1(1) **Refinement:** The TSF shall [**selection**:

- *invoke platform-provided functionality to use [**selection**:*
  - *IPsec,*
  - *mutually authenticated TLS,*
  - *mutually authenticated DTLS,*
  - *HTTPS,*
  - *SSH*

  ],
- *implement functionality using [**selection**:*
  - *IPsec as defined in the PP-Module for VPN Client,*

- *mutually authenticated TLS as defined in the Package for Transport Layer Security,*
- *mutually authenticated DTLS as defined in the Package for Transport Layer Security,*
- *HTTPS in accordance with FCS_HTTPS_EXT.1,*
- *SSH as defined in the Extended Package for Secure Shell*

]

] to protect all data from [disclosure and modification] when it is transferred between separate parts of the TOE.

Application Note: This requirement ensures all communications between components of a distributed TOE are protected through the use of an encrypted communications channel. The data passed in this trusted communication channel are encrypted as defined in the protocol chosen in the second selection.

The trusted channel uses secure protocols that preserve the confidentiality and integrity of MDM communications. The ST author chooses the mechanism or mechanisms supported by the TOE. To support mutual authentication FIA_X509_EXT.1(2) should be included in the ST. This channel may also be used as the registration channel for the registration process, as described in section 3.1 and FCO_CPC_EXT.1.2.

If "IPsec as defined in the PP-Module for VPN Client" is selected, the TSF must claim conformance to a PP-Configuration that includes the VPN Client PP-Module.

If the ST author selects "SSH as defined in the Extended Package for Secure Shell", the TSF must be validated against the EP for Secure Shell with the MDM PP. It should be noted that due to constraints imposed by this PP that sha1 cannot be used.

If the ST author selects "mutually authenticated TLS as defined in the Package for Transport Layer Security" or "mutually authenticated DTLS as defined in the Package for Transport Layer Security", the TSF must be validated against requirements from the Package for Transport Layer Security, with the following selections made:

- FCS_TLS_EXT.1:
  - either TLS or DTLS is selected depending on the selection made in FPT_ITT.1.1(1)
  - either client or server is selected as appropriate
- FCS_TLSC_EXT.1.1 or FCS_TLSS_EXT.1.1 (as appropriate):
  - The cipher suites selected must correspond with the algorithms and hash functions allowed in FCS_COP.1.
  - mutual authentication must be selected

Protocol, RBG, Certificate validation, algorithm, and similar services may be met with platform provided services.

Internal TOE TSF Data Transfer (MDM Agent)

FPT_ITT.1.1(2) **Refinement:** The TSF shall [**selection**:

- *invoke platform-provided functionality to use [**selection**:*
  - *mutually authenticated TLS,*
  - *mutually authenticated DTLS,*
  - *HTTPS*

  ],
- *implement functionality using [**selection**:*
  - *mutually authenticated TLS as defined in the Package for Transport Layer Security,*
  - *mutually authenticated DTLS as defined in the Package for Transport Layer Security,*
  - *HTTPS in accordance with FCS_HTTPS_EXT.1*

  ]

] to protect all data from [disclosure and modification] when it is transferred between the TSF and MDM Agent.

Application Note: This requirement ensures all communications between the TSF and MDM Agent are protected through the use of an encrypted communications channel. The data passed in this trusted communication channel are encrypted as defined in the protocol chosen in the second selection.

The trusted channel uses secure protocols that preserve the confidentiality and integrity of MDM communications. The ST author chooses the mechanism or mechanisms supported by the TOE. To support mutual authentication FIA_X509_EXT.1(1) should be included in the ST. This channel may also be used as the registration channel for the registration process, as described in section 3.1 and FCO_CPC_EXT.1.2.

If the ST author selects "mutually authenticated TLS as defined in the Package for Transport Layer Security" or "mutually authenticated DTLS as defined in the Package for Transport Layer Security", the TSF must be validated against requirements from the Package for Transport Layer Security, with the following selections made:

- FCS_TLS_EXT.1:
  - either TLS or DTLS is selected depending on the selection made in FPT_ITT.1.1(2)

    - either client or server is selected as appropriate
- FCS_TLSC_EXT.1.1 or FCS_TLSS_EXT.1.1 (as appropriate):
    - The cipher suites selected must correspond with the algorithms and hash functions allowed in FCS_COP.1.
    - mutual authentication must be selected

Protocol, RBG, Certificate validation, algorithm, and similar services may be met with platform provided services.

Inter-TSF Trusted Channel (MDM Agent)

FTP_ITC.1.1(2) **Refinement:** The TSF shall [**selection**:

- *invoke platform-provided functionality to use [**selection**:*
    - *mutually authenticated TLS,*
    - *mutually authenticated DTLS,*
    - *HTTPS*

  ],
- *implement functionality using [**selection**:*
    - *mutually authenticated TLS as defined in the Package for Transport Layer Security,*
    - *mutually authenticated DTLS as defined in the Package for Transport Layer Security,*
    - *HTTPS in accordance with FCS_HTTPS_EXT.1*

  ]

] to provide a trusted communication channel between itself (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.

Application Note: The intent of the mandatory portion of the above requirement is to use the cryptographic protocols identified in the requirement to establish and maintain a trusted channel between the TOE and the MDM Agent. If the TOE includes a separate MAS Server, this requirement also addresses the communication between the MAS Server and the MDM Agent. Only TLS, DTLS, or HTTPS are used in this trusted channel.

This requirement is to ensure that the transmission of any audit logs, mobile device information data (software version, hardware model, and application versions), and configuration data collected by the MDM Agent and sent from the MDM Agent to the MDM Server, when commanded, or at configurable intervals, is properly protected. This trusted channel also protects any commands and policies sent by the MDM Server to the MDM Agent. Either the MDM Agent or the MDM Server is able to initiate the connection.

For TLS connections between the MDM Server and Agent, the MDM Server is the Server side of the TLS connection, therefore it is appropriate to include the selection-based FCS_TLSS SFRs in the ST, not FCS_TLSC SFRs. With respect to mutual authentication, in cases where the Agent is outside of the TOE, it should be verified that the server can support mutual authentication, meaning that the server includes support for client-side certificates for TLS mutual authentication post-enrollment. However, the client side is not evaluated since the agent is not in the TOE.

This trusted channel protects the connection between an enrolled MDM Agent and the MDM Server. FTP_TRP.1(2) provides a trusted channel to protect the connection between an unenrolled MDM Agent and the MDM Server during the enrollment operation.

The trusted channel uses TLS, DTLS, or HTTPS as the protocol that preserves the confidentiality and integrity of MDM communications. The ST author chooses the mechanism or mechanisms supported by the TOE.

If the ST author selects "mutually authenticated TLS as defined in the Package for Transport Layer Security" or "mutually authenticated DTLS as defined in the Package for Transport Layer Security", the TSF must be validated against requirements from the Package for Transport Layer Security, with the following selections made:

- FCS_TLS_EXT.1:
    - either TLS or DTLS is selected depending on the selection made in FTP_ITC.1.1(2)
    - server must be selected
- FCS_TLSS_EXT.1.1:
    - The cipher suites selected must correspond with the algorithms and hash functions allowed in FCS_COP.1.
    - mutual authentication must be selected

Protocol, RBG, Certificate validation, algorithm, and similar services may be met with platform provided services.

The requirement implies that not only are communications protected when they are initially established, but also on resumption after an outage. It may be the case that some part of the TOE setup involves manually setting up tunnels to protect other communication, and if after an outage the TOE attempts to re-establish the communication automatically with (the necessary) manual intervention, there may be a window created where an attacker might be able to gain critical information or compromise a connection.

FTP_ITC.1.2(2)	**Refinement:** The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to permit the TSF and MDM Agent to initiate communication via the trusted channel.
FTP_ITC.1.3(2)	**Refinement:** The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to initiate communication via the trusted channel for all communication between the TSF and the MDM Agent

Objective Security Functional Requirements

Support for Compliance Reporting of Mobile Device Configuration

FAU_CRP_EXT.1.1 The TSF shall provide [**selection**: *an interface that provides responses to queries about the configuration of enrolled devices*, *an interface that permits the export of data about the configuration of enrolled devices*] to authorized entities over a channel that meets the secure channel requirements in FTP_ITC.1(1). The provided information for each enrolled mobile device includes:

a. The current version of the MD firmware/software
b. The current version of the hardware model of the device
c. The current version of installed mobile applications
d. List of MD configuration policies that are in place on the device (as defined in FMT_SMF.1.1(1))
e. [**selection**: *[**assignment**: list of other available information about enrolled devices]*, *no other information*]

Application Note: The intent of this requirement is that the MDM Server be able to provide compliance information about enrolled mobile devices for use by other enterprise security infrastructure systems. There are active standards efforts underway by the Internet Engineering Task Force (IETF) Security Automation and Continuous Monitoring (SACM) Working Group and others to define protocols and standards to assess and report upon endpoint device posture. We expect that this requirement will evolve in future versions of this Protection Profile as standards efforts mature.

Component Registration Channel Definition

FCO_CPC_EXT.1.1 The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to require an Administrator to enable communications between any pair of TOE components before such communication can take place.

FCO_CPC_EXT.1.2 The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to implement a registration process in which components establish and use a communications channel that uses [**selection**:

- *A channel that meets the secure channel requirements in [**selection**: FTP_ITC.1, FPT_ITT.1(1), FPT_ITT.1(2)]* ,
- *A channel that meets the secure registration channel requirements in FTP_TRP.1(3)*,
- *No channel*

] for at least TSF data.

FCO_CPC_EXT.1.3 The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to enable an administrator to disable communications between any pair of TOE components.

Application Note: This SFR is only applicable if the TOE is distributed and therefore has multiple components that need to communicate via an internal TSF channel. When creating the TSF from the initial pair of components, either of these components may be identified as the TSF for the purposes of satisfying the meaning of "TSF" in this SFR.

The intention of this requirement is to ensure that there is a registration process that includes a positive enablement step by an administrator before components joining a distributed TOE can communicate with the other components of the TOE and before the new component can act as part of the TSF. The registration process may itself involve communication with the joining component: many implementations use a bespoke process for this, and the security requirements for the "registration communication" are then defined in FCO_CPC_EXT.1.2. Use of this "registration communication" channel is not deemed inconsistent with the requirement of FCO_CPC_EXT.1.1 (i.e. the registration channel can be used before the enablement step, but

only in order to complete the registration process).

The channel selection (for the registration channel) in FCO_CPC_EXT.1.2 is essentially a choice between the use of a normal secure channel that is equivalent to a channel used to communicate with external IT entities (FTP_ITC.1) or existing TOE components (FPT_ITT.1(1)/FPT_ITT.1(2)), or else a separate type of channel that is specific to registration (FTP_TRP.1(3)). If the TOE does not require a communications channel for registration (e.g. because the registration is achieved entirely by configuration actions by an administrator at each of the components) then the main selection in FCO_CPC_EXT.1.2 is completed with the "No channel" option.

If the ST author selects the FTP_ITC.1 or FPT_ITT.1(1)/FPT_ITT.1(2) channel type in the main selection in FCO_CPC_EXT.1.2 then the TSS identifies the relevant SFR iteration that specifies the channel used. If the ST author selects the FTP_TRP.1(3) channel type, then the TSS (possibly with support from the operational guidance) describes details of the channel and the mechanisms that it uses (and describes how the registration process ensures that the channel can only be used by the intended joiner and gatekeeper). Note that the FTP_TRP.1(3) channel type may require support from security measures in the operational environment (see the definition of FTP_TRP.1(3) for details).

If the ST author selects the FTP_ITC.1 or FPT_ITT.1(1)/FPT_ITT.1(2) channel type in the main selection in FCO_CPC_EXT.1.2 then the ST identifies the registration channel as a separate iteration of FTP_ITC.1 or FPT_ITT.1(1)/FPT_ITT.1(2) and gives the iteration identifier (e.g. "FPT_ITT.1/Join") in an ST Application Note for FCO_CPC_EXT.1.

Note that the channel that is set up and used for registration may be adopted as a continuing internal communication channel (i.e. between different TOE components) provided that the channel meets the requirements of FTP_ITC.1 or FPT_ITT.1(1)/FPT_ITT.1(2). Otherwise the registration channel is closed after use and a separate channel is used for the internal communications.

Specific requirements for Preparative Procedures relating to FCO_CPC_EXT.1 are defined in the Evaluation Activities.

User Authentication (Re-Use Prevention)

FIA_UAU_EXT.4.1(1) The TSF shall prevent reuse of enrollment authentication data related to [**assignment**: *identified authentication mechanism(s)*].

Application Note: This requirement references the authentication mechanism(s) used to authenticate the user for enrollment in FIA_ENR_EXT.1.1. If a username and password is used to authenticate the user for enrollment, the password must not be reused. Thus if the user has two devices enrolled in management or needs to re-enroll the same device (i.e., after a device wipe), the password must be different for each enrollment. Additionally, if two different users are enrolling the password must be different for each user.

User Authentication (Re-Use Prevention for Device Enrollment)

FIA_UAU_EXT.4.1(2) The TSF shall prevent reuse of [**selection**: *IMEI*, ***[assignment**: a unique device ID]*] related to limiting the user's enrollment of devices.

Application Note: The MDM server must not allow two devices to be enrolled using the same unique identifier. The unique identifier is specified in FIA_ENR_EXT.1.2.

FIA_UAU_EXT.4.1(2) can only be included in the ST if "devices specified by IMEI" or "device specified by [**assignment**: *a unique device ID*] is selected in FIA_ENR_EXT.1.2. The same selection must be completed for this requirement.

X.509 Enrollment

FIA_X509_EXT.3.1 The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to generate a Certificate Request Message as specified by RFC 2986 and be able to provide the following information in the request: public key and [**selection**: *device-specific information*, *Common Name*, *Organization*, *Organizational Unit*, *Country*].

Application Note: The public key is the public key portion of the public-private key pair generated by the TOE as specified in FCS_CKM.1.1.

As Enrollment over Secure Transport (EST) is a new standard that has not yet been widely adopted, this requirement is included as an interim objective requirement in order to allow developers to distinguish those products which have to have the ability to generate Certificate Request Messages but do not yet implement EST.

FIA_X509_EXT.3.2	The TSF shall[**selection**: *invoke platform-provided functionality*, *implement functionality*] to validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

Alternate X.509 Enrollment

FIA_X509_EXT.4.1	The TSF shall use the Enrollment over Secure Transport (EST) protocol as specified in RFC 7030 to request certificate enrollment using the simple enrollment method described in RFC 7030 Section 4.2.
FIA_X509_EXT.4.2	The TSF shall be capable of authenticating EST requests using an existing certificate and corresponding private key as specified by RFC 7030 Section 3.3.2.
FIA_X509_EXT.4.3	The TSF shall be capable of authenticating EST requests using HTTP Basic Authentication with a username and password as specified by RFC 7030 Section 3.2.3.
FIA_X509_EXT.4.4	The TSF shall perform authentication of the EST server using an Explicit Trust Anchor following the rules described in RFC 7030, section 3.6.1. **Application Note:** EST also uses HTTPS as specified in FCS_HTTPS_EXT.1 to establish a secure connection to an EST server, and thus, the ST author must also include FCS_HTTPS_EXT.1 in the main body of the ST. The separate Trust Anchor Database dedicated to EST operations is described as Explicit Trust Anchors in RFC 7030.
FIA_X509_EXT.4.5	The TSF shall be capable of requesting server-provided private keys as specified in RFC 7030 Section 4.4.
FIA_X509_EXT.4.6	The TSF shall be capable of updating its EST-specific Trust Anchor Database using the "Root CA Key Update" process described in RFC 7030 Section 4.1.3.
FIA_X509_EXT.4.7	The TSF shall generate a Certificate Request Message for EST as specified in RFC 2986 and be able to provide the following information in the request: public key and [**selection**: • *device-specific information*, • *Common Name, Organization, Organizational Unit, and Country*].
FIA_X509_EXT.4.8	The TSF shall validate the chain of certificates from the Root CA certificate in the Trust Anchor Database to the EST Server CA certificate upon receiving a CA Certificates Response. **Application Note:** The public key referenced in FIA_X509_EXT.4.7 is the public key portion of the public-private key pair generated by the TOE as specified in FCS_CKM.1.

Security Attribute Expiration

FMT_SAE_EXT.1.1	The TSF shall be capable to specify a configurable expiration time for enrollment authentication data.
FMT_SAE_EXT.1.2	The TSF shall be able to deny enrollment after the expiration time for the enrollment authentication data has passed. **Application Note:** This requirement references the user authenticator used for device enrollment in management in FIA_ENR_EXT.1.1. The user authenticator must only be valid for a configurable time limit. If the authenticator is expired, even if entered correctly, enrollment must not occur. The length of the time the authenticator is valid for is configured per function c.5 in FMT_SMF.1(2). If FMT_SAE_EXT.1 is included in the ST, then function g must be selected in FMT_SMF.1(2).

Trusted Path (for Joining)

FTP_TRP.1.1(3)	**Refinement:** The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to provide a communication path between itself and a joining component that is logically distinct from other communication paths and provides assured identification of [**selection**: *the TSF endpoint*, *both joining component and TSF endpoint*] and protection of the communicated data from modification and [**selection**: *disclosure*, *none*].
FTP_TRP.1.2(3)	**Refinement:** The TSF shall [**selection**: *invoke platform-provided functionality*, *implement*

functionality] to permit [**selection**: *the TSF*, *the joining component*] to initiate communication via the trusted path.

FTP_TRP.1.3(3) **Refinement:** The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to require the use of the trusted path for [joining components to the TSF under environmental constraints identified in [**assignment**: *reference to operational guidance*]].

Application Note: This SFR implements one of the types of channel identified in the main selection for FCO_CPC_EXT.1.2. The "joining component" in FTP_TRP.1(3) is the IT entity that is attempting to join the distributed TOE by using the registration process.

The effect of this SFR is to require the ability for components to communicate in a secure manner while the distributed TSF is being created (or when adding components to an existing distributed TSF). When creating the TSF from the initial pair of components, either of these components may be identified as the TSF for the purposes of satisfying the meaning of "TSF" in this SFR.

The selection at the end of FTP_TRP.1.1(3) recognises that in some cases confidentiality (i.e. protection of the data from disclosure) may not be provided by the channel. The ST author distinguishes in the TSS whether in this case the TOE relies on the environment to provide confidentiality (as part of the constraints referenced in FTP_TRP.1.3(3)) or whether the registration data exchanged does not require confidentiality (in which case this assertion must be justified). If "none" is selected, then this word may be omitted in the ST to improve readability.

The assignment in FTP_TRP.1.3(3) ensures that the ST highlights any specific details needed to protect the registration environment. Note that when the ST uses FTP_TRP.1(3) for the registration channel then this channel cannot be reused as the normal inter-component communication channel (the latter channel must meet FTP_ITC.1 or FPT_ITT.1(1)/FPT_ITT.1(2)). Specific requirements for Preparative Procedures relating to FTP_TRP.1(3) are defined in the Evaluation Activities.

Optional Security Functional Requirements

Audit Review

FAU_SAR.1.1 **Refinement:** The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to provide [Authorized Administrators] with the capability to read [all audit data] from the audit records.

FAU_SAR.1.2 **Refinement:** The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to provide the audit records in a manner suitable for the Authorized Administrators to interpret the information.

Application Note: The intent of this requirement is to ensure that the administrator can view and interpret the audit records and to prevent unauthorized users from accessing the logs.

Security Audit Event Selection

FAU_SEL.1.1 **Refinement:** The TSF shall [**selection**: *invoke platform-provided functionality*, *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. [**assignment**: *other attributes*]

Application Note: The intent of this requirement is to identify all criteria that can be selected to trigger an audit event. The ST author must select whether the TSF or the platform maintains the audit record. For the ST author, the assignment is used to list any additional criteria or "none".

Default TOE Access Banners

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

Application Note: This requirement is to ensure that an advisory notice and/or consent banner is presented to the user on start-up or unlock of the TSF.