Tabular Presentation of the *Protection Profile for Mobile Device Management*

Version: 4.0

2019-04-25

National Information Assurance Partnership

Revision History

Version	Date	Comment

Introduction

This document presents the Security Functional Requirements and Security Assurance Requirements from the *Protection Profile for Mobile Device Management*. This tabular representation is provided for those audiences whose interest primarily lies in those portions of that document. The Protection Profile itself remains the only complete and authoritative representation, and includes discussion of assumptions, threats, and objectives.

Security Functional Requirements

ID	Requirement	Assurance Activity
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.	The evaluator shall examine the TSS and verify that it describes how the alert system is implemented. The evaluator shall also verify that a description of each assigned event is provided in the TSS. The evaluator shall examine the guidance document and verify that it describes how the alerts can be configured, if configurable. For each MDM Agent/platform listed as supported in the ST: • **Test 1:** The evaluator shall enroll a device and ensure that the MDM server alerts the administrator of the change in enrollment status. The evaluator shall unenroll (retire) a device and ensure that the MDM server alerts the administrator of the change in enrollment status. • **Test 2:** The evaluator shall configure policies, which the MDM agent should not be able to apply. These policies shall include: ◦ a setting which is configurable on the MDM Server interface but not supported by the platform on which the MDM Agent runs, if any such settings exist ◦ a valid configuration setting with an invalid parameter, which may require manual modification of the policy prior to transmission to the device The evaluator shall deploy such policies and verify that the MDM server alerts the administrator about the failed application of the policy. • **Test 3:** (Conditional) The evaluator shall trigger each of the events listed and ensure that the MDM Server alerts the administrator.
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 evaluator shall check to ensure that the operational guidance contains instructions on how to access the MDM Server's compliance

ID	Requirement	Assurance Activity
	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*] ***This is currently an objective requirement.*** **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.	reporting interface. • **Test 1:** Using the operational guidance, the evaluator shall demonstrate the ability to access the compliance reporting interface from an authorized entity and successfully obtain information about enrolled devices. • **Test 2:** The evaluator shall attempt to access the compliance reporting interface from an unauthorized entity and demonstrate that the attempt is denied.
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.	The evaluator shall check the TSS and ensure that it lists all of the auditable events. The evaluator shall check to make sure that every audit event type mandated by the PP is described in the TSS. The evaluator shall verify that for every audit event described in the TSS, the description indicates where the audit event is generated (TSF, TOE platform). If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). The evaluator shall check the administrative guide and ensure that it lists all of the auditable events. The evaluator shall check to make sure that every audit event type mandated by the PP is described. The evaluator shall also make a determination of the administrative actions that are relevant in the context of this PP including those listed in the Management section. The evaluator shall examine the administrative guide and make a determination of which administrative commands are related to the configuration (including enabling or disabling) of the mechanisms implemented in the TOE that are necessary to enforce the requirements specified in the PP. The evaluator shall document the methodology or approach taken while determining which actions in the administrative guide are security relevant with respect to this PP. The evaluator may perform this activity as part of the activities associated with ensuring the AGD_OPE guidance satisfies the requirements. The evaluator shall test the TOEs ability to correctly generate audit records by having the TOE generate audit records for the events listed in the provided table and administrative actions. This should include all instances of an event. The evaluator shall test that audit records are generated for the establishment and termination of a channel for each of the cryptographic protocols contained in the ST. For administrative actions, the evaluator shall test that each action determined by the evaluator above to be security relevant in the context of this PP is auditable. Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly. For example, testing performed to ensure that the administrative guidance provided is correct verifies that AGD_OPE.1 is satisfied and should address the invocation of the administrative actions that are needed to verify the audit records are generated as expected.
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	The evaluator shall check the TSS and ensure that it provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field. The evaluator shall check the administrative guide and ensure that it provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field. The evaluator shall check to make sure that the description of the fields contains the information required in FAU_GEN.1.2(1). When verifying the test results from FAU_GEN.1.1(1), the evaluator shall ensure the audit records generated during testing match the format specified in the administrative guide, and that the fields in each audit record have the proper entries.

ID	Requirement	Assurance Activity
	that is performed by the TSF and that which is performed by the TOE platform.	Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly. For example, testing performed to ensure that the administrative guidance provided is correct verifies that AGD_OPE.1 is satisfied and should address the invocation of the administrative actions that are needed to verify the audit records are generated as expected. The Auditable Events table includes optional, selection-based and objective requirements. The auditing of these requirements are only required if the requirement is included in the ST. • (man) - mandatory requirement • (sel) - selection-based requirement • (obj) - objective requirement • (opt) - optional requirement : Auditable Events **Requirement** | **Auditable Events** | **Additional Audit Record Contents** FAU_ALT_EXT.1 (man) | Type of alert. | Identity of Mobile Device that sent alert. FAU_CRP_EXT.1 (obj) | None. | FAU_GEN.1(1) (man) | None. | FAU_GEN.1(2) (sel) | None. | FAU_NET_EXT.1 (man) | None. | FAU_SAR.1 (opt) | None. | FAU_SEL.1 (opt) | All modifications to the audit configuration that occur while the audit collection functions are operating. | No additional information. FAU_STG_EXT.1 (man) | None. | FAU_STG_EXT.2 (sel) | None. | FCO_CPC_EXT.1 (obj) | Enabling or disabling communications between a pair of components. | Identities of the endpoints pairs enabled or disabled. FCS_CKM_EXT.4 (man) | None. | FCS_CKM.1 (man) | [**selection**: *Failure of key generation activity for authentication keys*, *None*] | No additional information. FCS_CKM.2 (man) | None. | FCS_COP.1(1) (man) | None. | FCS_COP.1(2) (man) | None. | FCS_COP.1(3) (man) | None. | FCS_COP.1(4) (man) | None. | FCS_DTLSC_EXT.1 (TLS Package) | Failure of the certificate validity check. | Issuer Name and Subject Name of certificate FCS_DTLSC_EXT.2 (TLS Package) | None. | FCS_DTLSS_EXT.1 (TLS Package) | Failure of the certificate validity check. | Issuer Name and Subject Name of certificate FCS_DTLSS_EXT.2 (TLS Package) | None. | FCS_HTTPS_EXT.1 (sel) | Failure of the certificate validity check. | Issuer Name and Subject Name of certificate. [**selection**: *User's authorization decision*, *no additional information*] FCS_IV_EXT.1 (sel) | None. |

ID	Requirement	Assurance Activity		
		FCS_RBG_EXT.1 (man)	Failure of the randomization process.	No additional information.
		FCS_STG_EXT.1 (man)	None.	
		FCS_STG_EXT.2 (sel)	None.	
		FCS_TLS_EXT.1 (TLS Package)	None.	
		FCS_TLSC_EXT.1 (TLS Package)	Failure to establish a TLS session.	Reason for failure.
			Failure to verify presented identifier.	Presented identifier and reference identifier.
		FCS_TLSC_EXT.2 (TLS Package)	None.	
		FCS_TLSC_EXT.3 (TLS Package)	None.	
		FCS_TLSC_EXT.4 (TLS Package)	None.	
		FCS_TLSC_EXT.5 (TLS Package)	None.	
		FCS_TLSS_EXT.1 (TLS Package)	Failure to establish a TLS session.	Reason for failure.
		FCS_TLSS_EXT.2 (TLS Package)	None.	
		FCS_TLSS_EXT.3 (TLS Package)	None.	
		FCS_TLSS_EXT.4 (TLS Package)	None.	
		FIA_ENR_EXT.1 (man)	Failure of MD user authentication.	Presented username.
		FIA_UAU_EXT.4(1) (obj)	Attempt to reuse enrollment data.	Enrollment data.
		FIA_UAU_EXT.4(2) (obj)	Attempt to reuse enrollment data.	Enrollment data.
		FIA_UAU.1 (man)	None.	
		FIA_X509_EXT.1(1) (man)	Failure to validate X.509 certificate	Reason for failure.
		FIA_X509_EXT.1(2) (sel)	Failure to validate X.509 certificate	Reason for failure.
		FIA_X509_EXT.2 (man)	Failure to establish connection to determine revocation status.	No additional information.
		FIA_X509_EXT.3 (obj)	Generation of Certificate Request Message.	Content of Certificate Request Message.
			Success or failure of verification.	Issuer and Subject name of added certificate or reason for failure.
		FIA_X509_EXT.4 (obj)	Generation of Certificate Enrollment Request.	Issuer and Subject name of EST Server. Method of authentication. Issuer and Subject name of certificate used to authenticate. Content of Certificate Request Message.
			Success or failure of enrollment.	Issuer and Subject name of added certificate or reason for failure.
			Update of EST Trust Anchor Database.	Subject name of added Root CA.

ID	Requirement	Assurance Activity		
		FIA_X509_EXT.5 (man)	None.	
		FMT_MOF.1(1) (man)	Issuance of command to perform function.	Command sent and identity of MDM Agent recipient(s).
			Change of policy settings.	Policy changed and value or full policy.
		FMT_MOF.1(2) (man)	Enrollment by a user.	Identity of user.
		FMT_MOF.1(3) (sel)	None.	
		FMT_POL_EXT.1 (man)	None.	
		FMT_SAE_EXT.1 (obj)	Enrollment attempted after expiration of authentication data.	Identity of user.
		FMT_SMF.1(1) (man)	None.	
		FMT_SMF.1(2) (man)	Success or failure of function.	No additional information.
		FMT_SMF.1(3) (sel)	None.	
		FMT_SMR.1(1) (man)	None.	
		FMT_SMR.1(2) (sel)	None.	
		FPT_API_EXT.1 (man)	None.	
		FPT_ITT.1(1) (sel)	Initiation and termination of the trusted channel.	Trusted channel protocol. Identity of initiator and recipient.
		FPT_ITT.1(2) (sel)	Initiation and termination of the trusted channel.	Trusted channel protocol. Identity of initiator and recipient.
		FPT_LIB_EXT.1 (man)	None.	
		FPT_TST_EXT.1 (man)	Initiation of self-test. Failure of self-test. Detected integrity violation.	Algorithm that caused failure. The TSF code file that caused the integrity violation.
		FPT_TUD_EXT.1 (man)	Success or failure of signature verification.	No additional information.
		FTA_TAB.1 (opt)	Change in banner setting.	No additional information.
		FTP_ITC.1(1) (man)	Initiation and termination of the trusted channel.	Trusted channel protocol. Non-TOE endpoint of connection.
		FTP_ITC.1(2) (sel)	Initiation and termination of the trusted channel.	Trusted channel protocol. Non-TOE endpoint of connection.
		FTP_ITC_EXT.1 (man)	None.	
		FTP_TRP.1(1) (man)	Initiation and termination of the trusted channel.	Trusted channel protocol. Identity of administrator.
		FTP_TRP.1(2) (man)	Initiation and termination of the trusted channel.	Trusted channel protocol.
		FTP_TRP.1(3) (obj)	Initiation and termination of the trusted channel.	Trusted channel protocol.
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	The evaluator shall check the TSS and ensure that it provides a format for audit records. The evaluator shall check the administrative guide and ensure that it		

ID	Requirement	Assurance Activity
	b. Failure to update an existing application on a managed mobile device. ***This is a selection-based requirement. Its inclusion depends upon selection in .*** **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.	provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field. The evaluator shall verify that when an application or update push fails, that the audit records generated match the format specified in the guidance and that the fields in each audit record have the proper entries.
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*] ***This is a selection-based requirement. Its inclusion depends upon selection in .*** **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.	The evaluator shall check the TSS and ensure that it provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field. The evaluator shall check the administrative guide and ensure that it provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field. The evaluator shall check to make sure that the description of the fields contains the information required in FAU_GEN.1.2(2). When verifying the test results from FAU_GEN.1.1(2), the evaluator shall ensure the audit records generated during testing match the format specified in the administrative guide, and that the fields in each audit record have the proper entries. Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly. For example, testing performed to ensure that the administrative guidance provided is correct verifies that AGD_OPE.1 is satisfied and should address the invocation of the administrative actions that are needed to verify the audit records are generated as expected.
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.	The evaluator ensures that the TSS describes how reachability events are implemented, for each supported mobile platform. The evaluator verifies that this description clearly indicates who (MDM Agent or MDM Server) initiates reachability events. The evaluator shall verify that the guidance instructs administrators on the method of determining the network connectivity status of an enrolled agent. For each MDM Agent/platform listed as supported in the ST: The evaluator shall configure the MDM Agent/platform to perform a network reachability test, both with and without such connectivity and shall ensure that by following the guidance, the evaluator can determine results that reflect both.
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. ***This is an optional requirement. It may be required by Extended Packages of this Protection Profile.***	If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).
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. ***This is an optional requirement. It may be required by Extended Packages of this Protection Profile.*** **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.	If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). The evaluator shall check the AGD guidance and ensure that it describes how the administrator accesses the audit data and describes the format of the audit record. The evaluator shall attempt to view the audit record as the authorized administrator and verify that the action succeeds. The evaluator shall ensure the audit records generated during testing match the format specified in the administrative guide.
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*] ***This is an optional requirement. It may be required by Extended Packages of this Protection Profile.*** **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".	If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). The evaluator shall review the administrative guidance to ensure that the guidance itemizes all event types, as well as describes all attributes that are to be selectable in accordance with the requirement, to include those attributes listed in the assignment. The administrative guidance shall also contain instructions on how to set the pre-selection as well as explain the syntax (if present) for multi-value pre-selection. The administrative guidance shall also identify those audit records that are always recorded, regardless of the selection criteria currently being enforced. The evaluator shall also perform the following tests: • **Test 1:** For each attribute listed in the requirement, the evaluator shall devise a test to show that selecting the attribute causes only audit events with that attribute (or those that are always recorded, as identified in the administrative guidance) to be recorded. • **Test 2:** [conditional] If the TSF supports specification of more complex audit pre-selection criteria (e.g., multiple attributes, logical expressions using attributes) then the evaluator shall devise tests showing that this capability is correctly implemented. The evaluator shall also, in the test plan, provide a short narrative justifying the set of tests as representative and sufficient to exercise the capability.

ID	Requirement	Assurance Activity
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.	The evaluator shall examine the TSS to ensure it describes the means by which the audit data are transferred to the external audit server, and how the trusted channel is provided. The evaluator shall also examine the operational guidance to determine that it describes the relationship between the local audit data and the audit data that are sent to the audit log server. For example, when an audit event is generated, is it simultaneously sent to the external server and the local store, or is the local store used as a buffer and "cleared" periodically by sending the data to the audit server. The evaluator shall also examine the operational guidance to ensure it describes how to establish the trusted channel to the audit server, as well as describe any requirements on the audit server (particular audit server protocol, version of the protocol required, etc.), as well as configuration of the TOE needed to communicate with the audit server. Testing of the trusted channel mechanism will be performed as specified in the associated evaluation activities for the particular trusted channel mechanism. The evaluator shall perform the following test for this requirement: • **Test 1:** The evaluator shall establish a session between the TOE and the audit server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the audit server and the TOE during several activities of the evaluator's choice designed to generate audit data to be transferred to the audit server. The evaluator shall observe that these data are not able to be viewed in the clear during this transfer, and that they are successfully received by the audit server. The evaluator shall record the particular software (name, version) used on the audit server during testing.
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. ***This is a selection-based requirement. Its inclusion depends upon selection in .*** **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.	If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how the audit record protection functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). If "implement functionality" is selected, the evaluator shall ensure that the TSS describes how the audit records are protected from unauthorized modification or deletion. The evaluator shall ensure that the TOE uses audit trail specific protection mechanisms. The evaluator shall perform the following tests: • **Test 1:** The evaluator shall access the audit trail as an unauthorized user and attempt to modify and delete the audit records. The evaluator shall verify that these attempts fail. • **Test 2:** The evaluator shall access the audit trail as an authorized user and attempt to modify and delete the audit records. The evaluator shall verify that these attempts succeed. The evaluator shall verify that only the records intended for modification and deletion are modified and deleted.
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. ***This is currently an objective requirement.***	If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how the audit record protection functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).
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. ***This is currently an objective requirement.***	If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how the audit record protection functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).
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. ***This is currently an objective requirement.*** **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,	If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how the audit record protection functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). The evaluator shall examine the guidance documentation to confirm that it contains instructions for enabling and disabling communications with any individual component of a distributed TOE. The evaluator shall confirm that the method of disabling is such that all other components can be prevented from communicating with the component that is being removed from the TOE (preventing the remaining components from either attempting to initiate communications to the disabled component, or from responding to communications from the disabled component).

ID	Requirement	Assurance Activity
	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.	• **Test 1:** The evaluator shall confirm that an IT entity that is not currently a member of the distributed TOE cannot communicate with any component of the TOE until the non-member entity is enabled by an administrator for each of the non-equivalent TOE components that it is required to communicate with (non-equivalent TOE components are as defined in the minimum configuration for the distributed TOE) • **Test 2:** The evaluator shall confirm that after enablement, an IT entity can communicate only with the components that it has been enabled for. This includes testing that the enabled communication is successful for the enabled component pair, and that communication remains unsuccessful with any other component for which communication has not been explicitly enabled. Some TOEs may set up the registration channel before the enablement step is carried out, but in such a case the channel must not allow communications until after the enablement step has been completed. • **Test 3:** The evaluator shall separately disable each TOE component in turn and ensure that the other TOE components cannot then communicate with the disabled component, whether by attempting to initiate communications with the disabled component or by responding to communication attempts from the disabled component.
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.	**If "invoke platform-provided functionality" is selected:** The evaluator shall examine the TSS to verify that it describes (for each supported platform) how the key generation functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). **If "implement functionality" is selected:** The evaluator shall ensure that the TSS identifies the key sizes supported by the TOE. If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme. The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key generation schemes and key sizes for all uses defined in this PP. **Key Generation for FIPS PUB 186-4 RSA Schemes** The evaluator shall verify the implementation of RSA Key Generation by the TOE using the Key Generation test. This test verifies the ability of the TSF to correctly produce values for the key components including the public verification exponent e, the private prime factors p and q, the public modulus n and the calculation of the private signature exponent d. Key Pair generation specifies 5 ways (or methods) to generate the primes p and q. These include: 1. Random Primes: a. Provable primes b. Probable primes 2. Primes with Conditions: a. Primes p1, p2, q1,q2, p and q shall all be provable primes b. Primes p1, p2, q1, and q2 shall be provable primes and p and q shall be probable primes c. Primes p1, p2, q1,q2, p and q shall all be probable primes To test the key generation method for the Random Provable primes method and for all the Primes with Conditions methods, the evaluator must seed the TSF key generation routine with sufficient data to deterministically generate the RSA key pair. This includes the random seed(s), the public exponent of the RSA key, and the desired key length. For each key length supported, the evaluator shall have the TSF generate 25 key pairs. The evaluator shall verify the correctness of the TSF's implementation by comparing values generated by the TSF with those generated from a known good implementation. If possible, the Random Probable primes method should also be verified against a known good implementation as described above. Otherwise, the evaluator shall have the TSF generate 10 keys pairs

ID	Requirement	Assurance Activity
		for each supported key length nlen and verify: • n = p*q, • p and q are probably prime according to Miller-Rabin tests, • GCD(p-1,e) = 1, • GCD(q-1,e) = 1, • 2^16 <= e <= 2^256 and e is an odd integer, • |p-q| > 2^(nlen/2 - 100), • p >= squareroot(2)*(2^(nlen/2 -1)), • q >= squareroot(2)*(2^(nlen/2 -1)), • 2^(nlen/2) < d < LCM(p-1,q-1), • e*d = 1 mod LCM(p-1,q-1). **Key Generation for Elliptic Curve Cryptography (ECC)** **FIPS 186-4 ECC Key Generation Test** For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator shall require the implementation under test (IUT) to generate 10 private/public key pairs. The private key shall be generated using an approved random bit generator (RBG). To determine correctness, the evaluator shall submit the generated key pairs to the public key verification (PKV) function of a known good implementation. **FIPS 186-4 Public Key Verification (PKV) Test** For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator shall generate 10 private/public key pairs using the key generation function of a known good implementation and modify five of the public key values so that they are incorrect, leaving five values unchanged (i.e., correct). The evaluator shall obtain in response a set of 10 PASS/FAIL values. **Key Generation for Finite-Field Cryptography (FFC)** The evaluator shall verify the implementation of the Parameters Generation and the Key Generation for FFC by the TOE using the Parameter Generation and Key Generation test. This test verifies the ability of the TSF to correctly produce values for the field prime p, the cryptographic prime q (dividing p-1), the cryptographic group generator g, and the calculation of the private key x and public key y. The Parameter generation specifies two ways (or methods) to generate the cryptographic prime q and the field prime p: Cryptographic and Field Primes: • Primes q and p shall both be provable primes • Primes q and field prime p shall both be probable primes and two ways to generate the cryptographic group generator g: Cryptographic Group Generator: • Generator g constructed through a verifiable process • Generator g constructed through an unverifiable process. The Key generation specifies two ways to generate the private key x: Private Key: • len(q) bit output of RBG where 1 <=x <= q-1 • len(q) + 64 bit output of RBG, followed by a mod q-1 operation where 1<= x<=q-1. The security strength of the RBG must be at least that of the security offered by the FFC parameter set. To test the cryptographic and field prime generation method for the provable primes method and/or the group generator g for a verifiable process, the evaluator must seed the TSF parameter generation routine with sufficient data to deterministically generate the parameter set. For each key length supported, the evaluator shall have the TSF generate 25 parameter sets and key pairs. The evaluator shall verify the correctness of the TSF's implementation by comparing values generated by the TSF with those generated from a known good implementation. Verification must also confirm • g != 0,1 • q divides p-1 • g^q mod p = 1 • g^x mod p = y for each FFC parameter set and key pair. **Diffie-Hellman Group 14 and FFC Schemes using "safe-prime" groups** Testing for FFC Schemes using Diffie-Hellman group 14 and/or "safe-prime" groups is done as part of testing in FCS_CKM.2.1.
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:	**If "invoke platform-provided functionality" is selected:**

ID	Requirement	Assurance Activity
	[**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.	The evaluator shall examine the TSS to verify that it describes (for each supported platform) how the key establishment functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). **If "implement functionality" is selected:** The evaluator shall ensure that the supported key establishment schemes correspond to the key generation schemes identified in FCS_CKM.1.1. If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme. The evaluator shall ensure that the TSS describes how the TOE handles decryption errors. In accordance with NIST Special Publication 800-56B, the TOE must not reveal the particular error that occurred, either through the contents of any outputted or logged error message or through timing variations. If KTS-OAEP is supported, the evaluator shall create separate contrived ciphertext values that trigger each of the three decryption error checks described in NIST Special Publication 800-56B section 7.2.2.3, ensure that each decryption attempt results in an error, and ensure that any outputted or logged error message is identical for each. If KTS-KEM-KWS is supported, the evaluator shall create separate contrived ciphertext values that trigger each of the three decryption error checks described in NIST Special Publication 800-56B section 7.2.3.3, ensure that each decryption attempt results in an error, and ensure that any outputted or logged error message is identical for each. If Diffie-Hellman group 14 is selected from FCS_CKM.2.1, the TSS shall describe how the implementation meets RFC 3526 Section 3. The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key establishment scheme(s). The evaluator shall verify the implementation of the key establishment schemes supported by the TOE using the applicable tests below. **SP800-56A Key Establishment Schemes** The evaluator shall verify a TOEs implementation of SP800-56A key agreement schemes using the following Function and Validity tests. These validation tests for each key agreement scheme verify that a TOE has implemented the components of the key agreement scheme according to the specifications in the recommendation. These components include the calculation of the primitives (the shared secret value Z) and the calculation of the derived keying material (DKM) via the Key Derivation Function (KDF). If key confirmation is supported, the evaluator shall also verify that the components of key confirmation have been implemented correctly, using the test procedures described below. This includes the parsing of the DKM, the generation of MACdata and the calculation of MACtag. Function Test The Function test verifies the ability of the TOE to implement the key agreement schemes correctly. To conduct this test the evaluator shall generate or obtain test vectors from a known good implementation of the TOE supported schemes. For each supported key agreement scheme-key agreement role combination, KDF type, and, if supported, key confirmation role- key confirmation type combination, the tester shall generate 10 sets of test vectors. The data set consists of one set of domain parameter values (FFC) or the NIST approved curve (ECC) per 10 sets of public keys. These keys are static, ephemeral or both depending on the scheme being tested. The evaluator shall obtain the DKM, the corresponding TOEs public keys (static and/or ephemeral), the MAC tag(s), and any inputs used in the KDF, such as the Other Information field OI and TOE ID fields. If the TOE does not use a KDF defined in SP 800-56A, the evaluator shall obtain only the public keys and the hashed value of the shared secret. The evaluator shall verify the correctness of the TSF's implementation of a given scheme by using a known good implementation to calculate the shared secret value, derive the keying material DKM, and compare hashes or MAC tags generated from these values. If key confirmation is supported, the TSF shall perform the above for each implemented approved MAC algorithm. Validity Test The Validity test verifies the ability of the TOE to recognize another party's valid and invalid key agreement results with or without key confirmation. To conduct this test, the evaluator shall obtain a list of the supporting cryptographic functions included in the SP800-56A key agreement implementation to determine which errors the TOE should be able to recognize. The evaluator generates a set of 24 (FFC) or 30 (ECC) test vectors consisting of data sets including domain parameter values or NIST approved curves, the evaluator's public keys, the TOEs public/private key pairs, MACTag, and any inputs used in the KDF, such as the other info and TOE ID fields.

ID	Requirement	Assurance Activity
		The evaluator shall inject an error in some of the test vectors to test that the TOE recognizes invalid key agreement results caused by the following fields being incorrect: the shared secret value Z, the DKM, the other information field OI, the data to be MACed, or the generated MACTag. If the TOE contains the full or partial (only ECC) public key validation, the evaluator will also individually inject errors in both parties' static public keys, both parties' ephemeral public keys and the TOEs static private key to assure the TOE detects errors in the public key validation function and/or the partial key validation function (in ECC only). At least two of the test vectors shall remain unmodified and therefore should result in valid key agreement results (they should pass). The TOE shall use these modified test vectors to emulate the key agreement scheme using the corresponding parameters. The evaluator shall compare the TOEs results with the results using a known good implementation verifying that the TOE detects these errors. **RSA-based key establishment** The evaluator shall verify the correctness of the TSF's implementation of RSAES-PKCS1-v1_5 by using a known good implementation for each protocol selected in FTP_TRP.1(1), FTP_TRP.1(2), FTP_TRP.1(3), FTP_ITC.1(1), FTP_ITC.1(2), FPT_ITT.1(1), and FPT_ITT.1(2) that uses RSAES-PKCS1-v1_5. **Diffie-Hellman Group 14** The evaluator shall verify the correctness of the TSF's implementation of Diffie-Hellman group 14 by using a known good implementation for each protocol selected in FTP_TRP.1(1), FTP_TRP.1(2), FTP_TRP.1(3), FTP_ITC.1(1), FTP_ITC.1(2), FPT_ITT.1(1), and FPT_ITT.1(2) that uses Diffie-Hellman Group 14. **FFC Schemes using "safe-prime" groups** The evaluator shall verify the correctness of the TSF's implementation of "safe-prime" groups by using a known good implementation for each protocol selected in FTP_TRP.1(1), FTP_TRP.1(2), FTP_TRP.1(3), FTP_ITC.1(1), FTP_ITC.1(2), FPT_ITT.1(1), and FPT_ITT.1(2) that uses "safe-prime" groups. This test must be performed for each "safe-prime" group that each protocol uses.
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.	If "invoking platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how the key destruction functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).

ID	Requirement	Assurance Activity
	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.	Evaluation Activity Note: The evaluation activity used is dependent on the selection made in FCS_CKM_EXT.4.1. The evaluator shall check to ensure the TSS lists each type of plaintext key material and CSP (authentication data, authorization data, secret/private symmetric keys, data used to derive keys, etc.) and its origin and storage location. The evaluator shall verify that the TSS describes when each type of key material and CSP is no longer needed. **If "invoke platform-provided functionality" is selected:** The evaluator shall examine the TSS to verify that it describes (for each supported platform) how the key releasing functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). **If "implement functionality" is selected:** The evaluator shall also verify that, for each type, the type of clearing procedure that is performed is listed. If different types of memory are used to store the materials to be protected, the evaluator shall check to ensure that the TSS describes the clearing procedure in terms of the memory in which the data are stored (for example, "secret keys stored on flash are cleared by overwriting once with zeros, while secret keys stored on the internal persistent storage device are cleared by overwriting one time with a random pattern that is changed before each write"). For block erases, the evaluator shall also ensure that the block erase command used is listed and shall verify that the command used also addresses any copies of the plaintext key material that may be created in order to optimize the use of flash memory. For each software and firmware key clearing situation the evaluator shall repeat the following tests. Note that at this time hardware-bound keys are explicitly excluded from testing. • **Test 1:** The evaluator shall utilize appropriate combinations of specialized operational environment and development tools (debuggers, simulators, etc.) for the TOE and instrumented TOE builds to test that keys are cleared correctly, including all intermediate copies of the key that may have been created internally by the TOE during normal cryptographic processing with that key. Cryptographic TOE implementations in software shall be loaded and exercised under a debugger to perform such tests. The evaluator shall perform the following test for each key subject to clearing, including intermediate copies of keys that are persisted encrypted by the TOE: 1. Load the instrumented TOE build in a debugger. 2. Record the value of the key in the TOE subject to clearing. 3. Cause the TOE to perform a normal cryptographic processing with the key from #1. 4. Cause the TOE to clear the key. 5. Cause the TOE to stop the execution but not exit. 6. Cause the TOE to dump the entire memory footprint of the TOE into a binary file. 7. Search the content of the binary file created in #4 for instances of the known key value from #1. The test succeeds if no copies of the key from #1 are found in step #7 above and fails otherwise. The evaluator shall perform this test on all keys, including those persisted in encrypted form, to ensure intermediate copies are cleared. • **Test 2:** In cases where the TOE is implemented in firmware and operates in a limited operating environment that does not allow the use of debuggers, the evaluator shall utilize a

ID	Requirement	Assurance Activity
		simulator for the TOE on a general purpose operating system. The evaluator shall provide a rationale explaining the instrumentation of the simulated test environment and justifying the obtained test results.
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.	**If "invoke platform-provided functionality" is selected:** The evaluator shall examine the TSS to verify that it describes (for each supported platform) how the encryption/decryption functionality is invoked for each mode and key size selected in the MDM Server's ST (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). **If "implement functionality" is selected:** **AES-CBC Tests** • **Test 1:** AES-CBC Known Answer Tests There are four Known Answer Tests (KATs), described below. In all KATs, the plaintext, ciphertext, and IV values shall be 128-bit blocks. The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation. ◦ **Test 1.1:** KAT-1. To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 10 plaintext values and obtain the ciphertext value that results from AES-CBC encryption of the given plaintext using a key value of all zeros and an IV of all zeros. Five plaintext values shall be encrypted with a 128-bit all-zeros key, and the other five shall be encrypted with a 256-bit all-zeros key. To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using 10 ciphertext values as input and AES-CBC decryption. ◦ **Test 1.2:** KAT-2. To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 10 key values and obtain the ciphertext value that results from AES-CBC encryption of an all-zeros plaintext using the given key value and an IV of all zeros. Five of the keys shall be 128-bit keys, and the other five shall be 256-bit keys. To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using an all-zero ciphertext value as input and AES-CBC decryption. ◦ **Test 1.3:** KAT-3. To test the encrypt functionality of AES-CBC, the evaluator shall supply the two sets of key values described below and obtain the ciphertext value that results from AES encryption of an all-zeros plaintext using the given key value and an IV of all zeros. The first set of keys shall have 128 128-bit keys, and the second set shall have 256 256-bit keys. Key i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1,N]. To test the decrypt functionality of AES-CBC, the evaluator shall supply the two sets of key and ciphertext value pairs described below and obtain the plaintext value that results from AES-CBC decryption of the given ciphertext using the given key and an IV of all zeros. The first set of key/ciphertext pairs shall have 128 128-bit key/ciphertext pairs, and the second set of key/ciphertext pairs shall have 256 256-bit key/ciphertext pairs. Key i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1,N]. The ciphertext value in each pair shall be the value that results in an all-zeros plaintext when decrypted with its corresponding key. ◦ **Test 1.4:** KAT-4. To test the encrypt functionality of AES-CBC, the evaluator shall supply the set of 128 plaintext values described below and obtain the two ciphertext values that result from AES-CBC encryption of the given plaintext using a 128-bit key value of all zeros with an IV of all zeros and using a 256-bit key value of all zeros with an IV of all zeros, respectively. Plaintext value i in each set shall have the leftmost i bits be ones and the rightmost 128-i bits be zeros, for i in [1,128]. To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt,

ID	Requirement	Assurance Activity

		using ciphertext values of the same form as the plaintext in the encrypt test as input and AES-CBC decryption.
		
		• **Test 2:** AES-CBC Multi-Block Message Test
		
		The evaluator shall test the encrypt functionality by encrypting an i-block message where 1 < i <=10. The evaluator shall choose a key, an IV and plaintext message of length i blocks and encrypt the message, using the mode to be tested, with the chosen key and IV. The ciphertext shall be compared to the result of encrypting the same plaintext message with the same key and IV using a known good implementation.
		
		The evaluator shall also test the decrypt functionality for each mode by decrypting an i-block message where 1 < i <=10. The evaluator shall choose a key, an IV and a ciphertext message of length i blocks and decrypt the message, using the mode to be tested, with the chosen key and IV. The plaintext shall be compared to the result of decrypting the same ciphertext message with the same key and IV using a known good implementation.
		
		• **Test 3:** AES-CBC Monte Carlo Tests
		
		The evaluator shall test the encrypt functionality using a set of 200 plaintext, IV, and key 3-tuples. 100 of these shall use 128 bit keys, and 100 shall use 256 bit keys. The plaintext and IV values shall be 128-bit blocks. For each 3-tuple, 1000 iterations shall be run as follows:
		
		`# Input: PT, IV, Key`
		`for i = 1 to 1000:`
		`  if i == 1:`
		`  CT[1] = AES-CBC-Encrypt(Key, IV, PT)`
		`  PT = IV`
		`  else:`
		`  CT[i] = AES-CBC-Encrypt(Key, PT)`
		`  PT = CT[i-1]`
		
		The ciphertext computed in the 1000th iteration (i.e., CT[1000]) is the result for that trial. This result shall be compared to the result of running 1000 iterations with the same values using a known good implementation.
		
		The evaluator shall test the decrypt functionality using the same test as for encrypt, exchanging CT and PT and replacing AES-CBC-Encrypt with AES-CBC-Decrypt.
		
		AES-GCM Test
		• **Test 1:** The evaluator shall test the authenticated encrypt functionality of AES-GCM for each combination of the following input parameter lengths:
		
		◦ 128 bit and 256 bit keys
		◦ Two plaintext lengths. One of the plaintext lengths shall be a non-zero integer multiple of 128 bits, if supported. The other plaintext length shall not be an integer multiple of 128 bits, if supported.
		◦ Three AAD lengths. One AAD length shall be 0, if supported. One AAD length shall be a non-zero integer multiple of 128 bits, if supported. One AAD length shall not be an integer multiple of 128 bits, if supported.
		◦ Two IV lengths. If 96 bit IV is supported, 96 bits shall be one of the two IV lengths tested.
		The evaluator shall test the encrypt functionality using a set of 10 key, plaintext, AAD, and IV tuples for each combination of parameter lengths above and obtain the ciphertext value and tag that results from AES-GCM authenticated encrypt. Each supported tag length shall be tested at least once per set of 10. The IV value may be supplied by the evaluator or the implementation being tested, as long as it is known.
		
		The evaluator shall test the decrypt functionality using a set of 10 key, ciphertext, tag, AAD, and IV 5- tuples for each combination of parameter lengths above and obtain a Pass/Fail result on authentication and the decrypted plaintext if Pass. The set shall include five tuples that Pass and five that Fail.
		
		The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.
		
		AES-CCM Tests
		• **Test 1:** The evaluator shall test the generation-encryption and decryption-verification functionality of AES-CCM for the following input parameter and tag lengths:

ID	Requirement	Assurance Activity
		◦ 128 bit and 256 bit keys ◦ Two payload lengths. One payload length shall be the shortest supported payload length, greater than or equal to zero bytes. The other payload length shall be the longest supported payload length, less than or equal to 32 bytes (256 bits). ◦ Two or three associated data lengths. One associated data length shall be 0, if supported. One associated data length shall be the shortest supported payload length, greater than or equal to zero bytes. One associated data length shall be the longest supported payload length, less than or equal to 32 bytes (256 bits). If the implementation supports an associated data length of 216 bytes, an associated data length of 216 bytes shall be tested. ◦ Nonce lengths. All supported nonce lengths between 7 and 13 bytes, inclusive, shall be tested. ◦ Tag lengths. All supported tag lengths of 4, 6, 8, 10, 12, 14 and 16 bytes shall be tested. To test the generation-encryption functionality of AES-CCM, the evaluator shall perform the following four tests: ◦ **Test 1.1:** For EACH supported key and associated data length and ANY supported payload, nonce and tag length, the evaluator shall supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext. ◦ **Test 1.2:** For EACH supported key and payload length and ANY supported associated data, nonce and tag length, the evaluator shall supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext. ◦ **Test 1.3:** For EACH supported key and nonce length and ANY supported associated data, payload and tag length, the evaluator shall supply one key value and 10 associated data, payload and nonce value 3-tuples and obtain the resulting ciphertext. ◦ **Test 1.4:** . For EACH supported key and tag length and ANY supported associated data, payload and nonce length, the evaluator shall supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext. To determine correctness in each of the above tests, the evaluator shall compare the ciphertext with the result of generation-encryption of the same inputs with a known good implementation. To test the decryption-verification functionality of AES-CCM, for EACH combination of supported associated data length, payload length, nonce length and tag length, the evaluator shall supply a key value and 15 nonce, associated data and ciphertext 3-tuples and obtain either a FAIL result or a PASS result with the decrypted payload. The evaluator shall supply 10 tuples that should FAIL and 5 that should PASS per set of 15. **AES Key Wrap (AES-KW) and Key Wrap with Padding (AES-KWP) Test** • **Test 1:** The evaluator shall test the authenticated encryption functionality of AES-KW for EACH combination of the following input parameter lengths: ◦ 128 and 256 bit key encryption keys (KEKs) ◦ Three plaintext lengths. One of the plaintext lengths shall be two semi-blocks (128 bits). One of the plaintext lengths shall be three semi-blocks (192 bits). The third data unit length shall be the longest supported plaintext length less than or equal to 64 semi-blocks (4096 bits). using a set of 100 key and plaintext pairs and obtain the ciphertext that results from AES-KW authenticated-encryption. To determine correctness, the evaluator shall use the AES-KW authenticated-encryption function of a known good implementation. The evaluator shall test the authenticated-decryption functionality of AES-KW using the same test as for authenticated-encryption, replacing plaintext values with ciphertext values and AES-KW authenticated-encryption with AES-KW authenticated-decryption. The evaluator shall test the authenticated-encryption functionality of AES-KWP using the same test as for AES-KW authenticated-encryption with the following change in the three plaintext lengths: ◦ One plaintext length shall be one octet. ◦ One plaintext length shall be 20 octets (160-bits). ◦ One plaintext length shall be the longest supported plaintext length less than or equal to 512 octets (4096-bits). The evaluator shall test the authenticated-decryption functionality of AES-KWP using the same test as for AES-KWP authenticated-encryption, replacing plaintext values with ciphertext values and AES-KWP authenticated-encryption with AES-KWP authenticated-decryption.
FCS_COP.1.1(2)	**Refinement:** The TSF shall [**selection**: *invoke platform-provided functionality*,	

ID	Requirement	Assurance Activity
	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).	**If "invoke platform-provided functionality" is selected:** The evaluator shall examine the TSS to verify that it describes (for each supported platform) how the hash functionality is invoked for each digest size selected in the MDM Server's ST (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). **If "implement functionality" is selected:** The evaluator shall check that the association of the hash function with other TSF cryptographic functions (for example, the digital signature verification function) is documented in the TSS. The evaluator checks the AGD documents to determine that any configuration that is required to be done to configure the functionality for the required hash sizes is present. The TSF hashing functions can be implemented in one of two modes. The first mode is the byte-oriented mode. In this mode the TSF only hashes messages that are an integral number of bytes in length; i.e., the length (in bits) of the message to be hashed is divisible by 8. The second mode is the bit-oriented mode. In this mode the TSF hashes messages of arbitrary length. As there are different tests for each mode, an indication is given in the following sections for the bit-oriented vs. the byte-oriented testmacs. The evaluator shall perform all of the following tests for each hash algorithm implemented by the TSF and used to satisfy the requirements of this PP. **Short Messages Test Bit-oriented Mode** • **Test 1:** The evaluators devise an input set consisting of m+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m bits. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF. **Short Messages Test Byte-oriented Mode** • **Test 1:** The evaluators devise an input set consisting of m/8+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m/8 bytes, with each message being an integral number of bytes. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF. **Selected Long Messages Test Bit-oriented Mode** • **Test 1:** The evaluators devise an input set consisting of m messages, where m is the block length of the hash algorithm. The length of the ith message is 512 + 99*i, where 1 ≤ i ≤ m. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF. **Selected Long Messages Test Byte-oriented Mode** • **Test 1:** The evaluators devise an input set consisting of m/8 messages, where m is the block length of the hash algorithm. The length of the ith message is 512 + 8*99*i, where 1 ≤ i ≤ m/8. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF. **Pseudorandomly Generated Messages Test** • **Test 1:** This test is for byte-oriented implementations only. The evaluators randomly generate a seed that is n bits long, where n is the length of the message digest produced by the hash function to be tested. The evaluators then formulate a set of 100 messages and associated digests by following the algorithm provided in Figure 1 of [SHAVS]. The evaluators then ensure that the correct result is produced when the messages are provided to the TSF.
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*]	**If "invoke platform-provided functionality" is selected:** The evaluator shall examine the TSS to verify that it describes (for each supported platform) how the digital signature functionality is invoked for each operation they are used for in the MDM Server (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). **If "implement functionality" is selected:**

ID	Requirement	Assurance Activity
	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.	**ECDSA Algorithm Tests** • **Test 1:** ECDSA FIPS 186-4 Signature Generation Test For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate 10 1024-bit long messages and obtain for each message a public key and the resulting signature values R and S. To determine correctness, the evaluator shall use the signature verification function of a known good implementation. • **Test 2:** ECDSA FIPS 186-4 Signature Verification Test For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate a set of 10 1024-bit message, public key and signature tuples and modify one of the values (message, public key or signature) in five of the 10 tuples. The evaluator shall obtain in response a set of 10 PASS/FAIL values. **RSA Signature Algorithm Tests** • **Test 1:** Signature Generation Test The evaluator shall verify the implementation of RSA Signature Generation by the TOE using the Signature Generation Test. To conduct this test the evaluator must generate or obtain 10 messages from a trusted reference implementation for each modulus size/SHA combination supported by the TSF. The evaluator shall have the TOE use their private key and modulus value to sign these messages. The evaluator shall verify the correctness of the TSF's signature using a known good implementation and the associated public keys to verify the signatures. • **Test 2:** Signature Verification Test The evaluator shall perform the Signature Verification test to verify the ability of the TOE to recognize another party's valid and invalid signatures. The evaluator shall inject errors into the test vectors produced during the Signature Verification Test by introducing errors in some of the public keys e, messages, IR format, and/or signatures. The TOE attempts to verify the signatures and returns success or failure. The evaluator shall use these test vectors to emulate the signature verification test using the corresponding parameters and verify that the TOE detects these errors.
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).	**If "invoke platform-provided functionality" is selected:** The evaluator shall examine the TSS to verify that it describes (for each supported platform) how the keyed-hash functionality is invoked for each mode and key size selected in the MDM Server's ST (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). **If "implement functionality" is selected:** The evaluator shall examine the TSS to ensure that it specifies the following values used by the HMAC function: key length, hash function used, block size, and output MAC length used. For each of the supported parameter sets, the evaluator shall compose 15 sets of test data. Each set shall consist of a key and message data. The evaluator shall have the TSF generate HMAC tags for these sets of test data. The resulting MAC tags shall be compared to the result of generating HMAC tags with the same key and IV using a known good implementation.
FCS_HTTPS_EXT.1.1	The TSF shall implement the HTTPS protocol that complies with RFC 2818. ***This is a selection-based requirement. Its inclusion depends upon selection in .***	
FCS_HTTPS_EXT.1.2	The TSF shall implement HTTPS using TLS in accordance with the Package for Transport Layer Security . ***This is a selection-based requirement. Its inclusion depends upon selection in .*** **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.	• **Test 1:** The evaluator shall attempt to establish an HTTPS connection with a web server, observe the traffic with a packet analyzer, and verify that the connection succeeds and that the traffic is identified as TLS or HTTPS. Other tests are performed in conjunction with the TLS evaluation activities.

ID	Requirement	Assurance Activity
FCS_IV_EXT.1.1	The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to generate IVs in accordance with . ***This is a selection-based requirement. Its inclusion depends upon selection in .*** **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 (XTS) / SP800-38E / No IV. Tweak values shall be non-negative integers, assigned consecutively, and starting at an arbitrary non-negative integer. 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).	**If "invoke platform-provided functionality" is selected:** The evaluator shall examine the TSS to verify that it describes (for each supported platform) how the IV generation is invoked for each mode selected in the MDM Server's ST (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). **If "implement functionality" is selected:** The evaluator shall examine the TSS to ensure that it details the encryption of user credentials, persistent secrets, and private keys and the generation of the IVs used for that encryption. The evaluator shall ensure that the generation of IVs for each key encrypted by the same KEK meets .
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.	**If "invoke platform-provided functionality" is selected:** The evaluator shall examine the TSS to verify that it describes (for each supported platform) how the RBG functionality is invoked for each operation they are used for in the MDM Server (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). **If "implement functionality" is selected:** The evaluator shall perform the following tests. The evaluator shall perform 15 trials for the RNG implementation. If the RNG is configurable, the evaluator shall perform 15 trials for each configuration. The evaluator shall also confirm that the operational guidance contains appropriate instructions for configuring the RNG functionality. If the RNG has prediction resistance enabled, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits, (3) generate a second block of random bits, and (4) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 - 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The

ID	Requirement	Assurance Activity
		final two are additional input and entropy input for the second call to generate. These values are randomly generated. "generate one block of random bits" means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP 800-90A). If the RNG does not have prediction resistance, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) reseed, (4) generate a second block of random bits, and (5) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 - 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The fifth value is additional input to the first call to generate. The sixth and seventh are additional input and entropy input to the call to reseed. The final value is additional input to the second generate call. The following paragraphs contain more information on some of the input values to be generated/selected by the evaluator. Entropy input: the length of the entropy input value must equal the seed length. Nonce: If a nonce is supported (CTR_DRBG with no Derivation Function does not use a nonce), the nonce bit length is one-half the seed length. Personalization string: The length of the personalization string must be less than or equal to seed length. If the implementation only supports one personalization string length, then the same length can be used for both values. If more than one string length is support, the evaluator shall use personalization strings of two different lengths. If the implementation does not use a personalization string, no value needs to be supplied. Additional input: the additional input bit lengths have the same defaults and restrictions as the personalization string lengths.
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.	Documentation shall be produced-and the evaluator shall perform the activities-in accordance with Appendix D: Entropy Documentation and Assessment and the "Clarification to the Entropy Documentation and Assessment Annex." In the future, specific statistical testing (in line with NIST SP 800-90B) will be required to verify the entropy estimates.
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.	Regardless of whether this requirement is met by the TSF or the TOE platform, the evaluator will check the TSS to ensure that it lists each persistent secret (credential, secret key) and private key needed to meet the requirements in the ST. For each of these items, the evaluator will confirm that the TSS lists for what purpose it is used, and how it is stored. The evaluator then performs the following actions. Persistent secrets and private keys manipulated by the TOE platform: The evaluator shall examine the TSS to verify that it describes (for each supported platform) how the key storage functionality is invoked for each persistent secret and private key described in the TSS (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). Persistent secrets and private keys manipulated by the TSF: The evaluator reviews the TSS to determine that it makes a case that, for each item listed as being manipulated by the TOE, it is not written unencrypted to persistent memory, and that the item is stored by the platform.
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*] . ***This is a selection-based requirement. Its inclusion depends upon selection in .*** **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.	The evaluator shall examine the TSS to ensure it describes in detail how user credentials, persistent secret and private keys are stored and encrypted. The evaluator shall review the TSS to determine that it makes a case that key material is not written unencrypted to persistent memory and that it identifies the mode of encryption. If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how the key encryption functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).

ID	Requirement	Assurance Activity
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.	The evaluator shall examine the TSS and verify that it describes the process of enrollment for each MDM Agent/platform listed as supported in the ST. This description shall include the trusted path used for enrollment (FTP_TRP.1(2)), the method of user authentication (username/password, token, etc.), the method of authentication decision (local or remote authentication services), and the actions performed on the MDM Server upon successful authentication. • **Test 1:** The evaluator shall attempt to enroll a device without providing correct credentials. The evaluator shall verify that the device is not enrolled and that the described enrollment actions are not taken. • **Test 2:** The evaluator shall attempt to enroll the device providing correct credentials. The evaluator shall verify that the device is enrolled and that the described enrollment actions are taken.
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.	The evaluator shall examine the TSS and verify that it implements a policy to limit the user's enrollment of devices. The evaluator shall ensure that the administrative guidance describes the method(s) of restricting user enrollment and that it instructs the administrator how to configure the restrictions. For each type of policy selected, the evaluator shall perform the following: • **Test 1:** The evaluator shall attempt to configure the MDM Server according to the administrative guidance in order to prevent enrollment. The evaluator shall verify that the user cannot enroll a device outside of the configured limitation. (For example, the evaluator may try to enroll a disallowed device, or may try to enroll additional devices beyond the number allowed.)
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.	If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).
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.	If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). The evaluator shall perform the following tests: • **Test 1:** The evaluator shall attempt to perform the prohibited actions before authentication. The evaluator shall verify the actions cannot be performed. • **Test 2:** The evaluator shall attempt to perform the prohibited actions after authentication. The evaluator shall verify the actions can be performed.
FIA_UAU_EXT.4.1(1)	The TSF shall prevent reuse of enrollment authentication data related to [**assignment**: *identified authentication mechanism(s)*]. ***This is currently an objective requirement.*** **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.	The evaluator shall verify that the TSS contains a description of the process of enrollment for each MDM Agent/platform listed as supported in the ST. This description shall include the method of user authentication (username/password, token, etc.) and how reuse of the authentication data is prevented. • **Test 1:** The evaluator shall enroll a device providing correct credentials. The evaluator shall attempt to enroll a second device using the same credentials used to enroll the first device. The evaluator shall verify that the second device could not enroll.
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. ***This is currently an objective requirement.*** **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.	The evaluator shall verify that the TSS contains a description of the policy to limit the user's enrollment of devices. The evaluator shall ensure that the administrative guidance describes the method(s) of restricting user enrollment and that it instructs the administrator how to configure the restrictions. • **Test 1:** The evaluator shall configure the MDM Server to restrict user's enrollment to a specific unique device ID and enroll a device with that device ID. The evaluator shall attempt to enroll a second device with the same unique device ID. (The evaluator may need to wipe the device without network connectivity, so the device is un-enrolled but the MDM server considers the device still enrolled.) The evaluator shall verify that the second enrollment using the same unique device ID fails.
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	**If "invoke platform-provided functionality" is selected:** The evaluator shall examine the TSS to verify that it describes (for each supported platform) how the certificate validation functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that

ID	Requirement	Assurance Activity
	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.	mechanism will be identified in the TSS as part of this evaluation activity). **If "implement functionality" is selected:** The evaluator shall ensure the TSS describes where the check of validity of the certificates takes place. The evaluator ensures the TSS also provides a description of the certificate path validation algorithm. The tests described must be performed in conjunction with the other certificate services evaluation activities, including each of the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. The evaluator shall create a chain of at least three certificates: the node certificate to be tested, an Intermediate CA, and the self-signed Root CA. • **Test 1:** The evaluator shall load a certificate or certificates as trusted CAs needed to validate the certificate to be used in the function, and demonstrate that the function succeeds. The evaluator shall then delete one of the certificates, and show that the function fails. • **Test 2:** The evaluator shall demonstrate that validating an expired certificate results in the function failing. • **Test 3:** The evaluator shall test that the TOE can properly handle revoked certificates--conditional on whether CRL or OCSP is selected; if both are selected, then a test shall be performed for each method. The evaluator shall test revocation of the node certificate and revocation of the intermediate CA certificate (i.e. the intermediate CA certificate should be revoked by the root CA). The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails. • **Test 4:** [Conditional] If OCSP is selected, the evaluator shall configure the OCSP server or use a man-in-the-middle tool to present a certificate that does not have the OCSP signing purpose and verify that validation of the OCSP response fails. If CRL is selected and the CA contains a Key Usage extension, the evaluator shall configure the CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set, and verify that validation of the CRL fails. If the CA is a root CA with no Key Usage extension, this test is not performed. • **Test 5:** The evaluator shall modify any byte in the first eight bytes of the certificate and demonstrate that the certificate fails to validate. (The certificate will fail to parse correctly.) • **Test 6:** The evaluator shall modify any byte in the last byte of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.) • **Test 7:** The evaluator shall modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)
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.	**If "invoke platform-provided functionality" is selected:** The evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). **If "implement functionality" is selected:** The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. • **Test 1:** The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOEs certificate does not contain the basicConstraints extension. The validation of the certificate path fails. • **Test 2:** The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOEs certificate has the cA flag in the basicConstraints extension not set. The validation of the certificate path fails. • **Test 3:** The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOEs certificate has the cA flag in the basicConstraints extension set to TRUE. The validation of the certificate path succeeds.
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*].	**If "invoke platform-provided functionality" is selected:** The evaluator shall examine the TSS to verify that it describes (for each supported platform) how the certificate validation functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). **If "implement functionality" is selected:**

ID	Requirement	Assurance Activity
	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. ***This is a selection-based requirement. Its inclusion depends upon selection in .*** **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.	The evaluator shall ensure the TSS describes where the check of validity of the certificates takes place. The evaluator ensures the TSS also provides a description of the certificate path validation algorithm. The tests described must be performed in conjunction with the other certificate services evaluation activities, including each of the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. The evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. • **Test 1:** The evaluator shall load a certificate or certificates as trusted CAs needed to validate the certificate to be used in the function, and demonstrate that the function succeeds. The evaluator shall then delete one of the certificates, and show that the function fails. • **Test 2:** The evaluator shall demonstrate that validating an expired certificate results in the function failing. • **Test 3:** The evaluator shall test that the TOE can properly handle revoked certificates--conditional on whether CRL or OCSP is selected; if both are selected, then a test shall be performed for each method. The evaluator shall test revocation of the node certificate and revocation of the intermediate CA certificate (i.e. the intermediate CA certificate should be revoked by the root CA). The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails. • **Test 4:** If OCSP is selected, the evaluator shall configure the OCSP server or use a man-in-the-middle tool to present a certificate that does not have the OCSP signing purpose and verify that validation of the OCSP response fails. If CRL is selected and the CA contains a Key Usage extension, the evaluator shall configure the CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set, and verify that validation of the CRL fails. If the CA is a root CA with no Key Usage extension, this test is not performed. • **Test 5:** The evaluator shall modify any byte in the first eight bytes of the certificate and demonstrate that the certificate fails to validate. (The certificate will fail to parse correctly.) • **Test 6:** The evaluator shall modify any byte in the last byte of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.) • **Test 7:** The evaluator shall modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)
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. ***This is a selection-based requirement. Its inclusion depends upon selection in .*** **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.	**If "invoke platform-provided functionality" is selected:** The evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). **If "implement functionality" is selected:** The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. • **Test 1:** The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOEs certificate does not contain the basicConstraints extension. The validation of the certificate path fails. • **Test 2:** The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOEs certificate has the cA flag in the basicConstraints extension not set. The validation of the certificate path fails. • **Test 3:** The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOEs certificate has the cA flag in the basicConstraints extension set to TRUE. The validation of the certificate path succeeds.
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,*	If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).

ID	Requirement	Assurance Activity
	◦ *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.	The evaluator shall check the TSS to ensure that it describes how the TOE chooses which certificates to use, and any necessary instructions in the administrative guidance for configuring the operating environment so that the TOE can use the certificates. If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). If "implement functionality" is selected, the evaluator shall examine the TSS to confirm that it describes the behavior of theTOE when a connection cannot be established during the validity check of a certificate used in establishing a trusted channel. The evaluator shall verify that any distinctions between trusted channels are described. If the requirement that the administrator is able to specify the default action is selected, then the evaluator shall ensure that the operational guidance contains instructions on how this configuration action is performed. The evaluator shall perform the following test for each trusted channel: • **Test 1:** The evaluator shall demonstrate use of a valid certificate requiring certificate validation checking to be performed in at least some part by communicating with a non-TOE IT entity. The evaluator shall then manipulate the environment so that the TOE is unable to verify the validity of the certificate, and observe that the action selected in FIA_X509_EXT.2.2 is performed. If the selected action is administrator-configurable, then the evaluator shall follow the operational guidance to determine that all supported administrator-configurable options behave in their documented manner.
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*]. ***This is currently an objective requirement.*** **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.	If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).
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. ***This is currently an objective requirement.***	If the ST author selects "device-specific information", the evaluator shall verify that the TSS contains a description of the device-specific fields used in certificate requests. If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). The evaluator shall check to ensure that the operational guidance contains instructions on requesting certificates from a CA, including generation of a Certificate Request Message. If the ST author selects "Common Name", "Organization", "Organizational Unit", or "Country", the evaluator shall ensure that this guidance includes instructions for establishing these fields before creating the certificate request message. • **Test 1:** The evaluator shall use the operational guidance to cause the TOE to generate a certificate request message.

ID	Requirement	Assurance Activity
		The evaluator shall capture the generated message and ensure that it conforms to the format specified. The evaluator shall confirm that the certificate request provides the public key and other required information, including any necessary user-input information. • **Test 2:** The evaluator shall demonstrate that validating a certificate response message without a valid certification path results in the function failing. The evaluator shall then load a certificate or certificates as trusted CAs needed to validate the certificate response message, and demonstrate that the function succeeds. The evaluator shall then delete one of the certificates, and show that the function fails.
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. ***This is currently an objective requirement.***	
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. ***This is currently an objective requirement.***	
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. ***This is currently an objective requirement.***	
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. ***This is currently an objective requirement.*** **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. ***This is currently an objective requirement.***	
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. ***This is currently an objective requirement.***	
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*]. ***This is currently an objective requirement.***	
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. ***This is currently an objective requirement.*** **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.	The evaluator shall check to ensure that the operational guidance contains instructions on requesting certificates from an EST server, including generating a Certificate Request Message. The evaluator shall also perform the following tests. Other tests are performed in conjunction with the TLS evaluation activities. • **Test 1:** The evaluator shall use the operational guidance to cause the TOE to request certificate enrollment from an EST server using the simple enrollment method described in RFC 7030 Section 4.2, authenticating the certificate request to the server using an existing certificate and private key as described by RFC 7030 Section 3.3.2. The evaluator shall confirm that the resulting certificate is successfully obtained and installed in the TOE key store. • **Test 2:** The evaluator shall use the operational guidance to cause the TOE to request certificate enrollment from an EST server using the simple enrollment method described in RFC 7030 Section 4.2, authenticating the certificate request to the server using a username and password as described by RFC 7030 Section 3.2.3. The evaluator shall confirm that the resulting certificate is successfully obtained and installed in the TOE key store. • **Test 3:** The evaluator shall modify the EST server to return a certificate containing a different public key than the key included in the TOEs certificate request. The evaluator shall use the operational guidance to cause the TOE to request certificate enrollment from an EST server. The evaluator shall confirm that the TOE does not accept the resulting certificate since the public key in the issued certificate does not match the public key in the certificate request. • **Test 4:** The evaluator shall configure the EST server or use a man-in-the-middle tool to present a server certificate to the TOE that is present in the TOE general Trust Anchor Database but not its EST-specific Trust Anchor Database. The evaluator shall cause the TOE to request certificate enrollment from the EST server. The evaluator shall verify that the request is not successful. • **Test 5:** The evaluator shall configure the EST server or use a man-in-the-middle tool to present an invalid certificate. The evaluator shall cause the TOE to request certificate

ID	Requirement	Assurance Activity
		enrollment from the EST server. The evaluator shall verify that the request is not successful. The evaluator shall configure the EST server or use a man-in-the-middle tool to present a certificate that does not have the CMC RA purpose and verify that requests to the EST server fail. The tester shall repeat the test using a valid certificate and a certificate that contains the CMC RA purpose and verify that the certificate enrollment requests succeed. • **Test 6:** The evaluator shall use a packet sniffing tool between the TOE and an EST server. The evaluator shall turn on the sniffing tool and cause the TOE to request certificate enrollment from an EST server. The evaluator shall verify that the EST protocol interaction occurs over a Transport Layer Security (TLS) protected connection. The evaluator is not expected to decrypt the connection but rather observe that the packets conform to the TLS protocol format. • **Test 7:** The evaluator shall use the operational guidance to cause the TOE to request a server-provided private key and certificate from an EST server. The evaluator shall confirm that the resulting private key and certificate are successfully obtained and installed in the TOE key store. • **Test 8:** The evaluator shall modify the EST server to, in response to a server-provided private key and certificate request, return a private key that does not correspond with the public key in the returned certificate. The evaluator shall use the operational guidance to cause the TOE to request a server-provided private key and certificate. The evaluator shall confirm that the TOE does not accept the resulting private key and certificate since the private key and public key do not correspond. • **Test 9:** The evaluator shall configure the EST server to provide a "Root CA Key Update" as described in RFC 7030 Section 4.1.3. The evaluator shall cause the TOE to request CA certificates from the EST server and shall confirm that the EST-specific Trust Anchor Database is updated with the new trust anchor. • **Test 10:** The evaluator shall configure the EST server to provide a "Root CA Key Update" as described in RFC 7030 Section 4.1.3, but shall modify part of the NewWithOld certificate's generated signature. The evaluator shall cause the TOE to request CA certificates from the EST server and shall confirm that the EST-specific Trust Anchor Database is not updated with the new trust anchor since the signature did not verify. • **Test 11:** The evaluator shall use the operational guidance to cause the TOE to generate a certificate request message. The evaluator shall capture the generated message and ensure that it conforms with the format specified by RFC 2986. The evaluator shall confirm that the certificate request provides the public key and other required information, including any necessary user-input information.
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.	If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). If "implement functionality" is selected then the evaluator shall examine the TSS to verify that it describes the methods to ensure that each client utilizes a unique certificate. For each MDM Agent/platform listed as supported in the ST: The evaluator shall utilize appropriate combinations of specialized operational environment and development tools (debuggers, simulators, etc.) for the TOE and instrumented TOE builds as needed to perform this test. One of the following tests must be performed depending on whether the MDM agent allows for the loading of certificates. • **Test 1:** [conditional] If the MDM agent allows for the loading of certificates: The evaluator shall initiate communications between the MDM Server and a client device over a trusted channel established using the device's unique certificate, verifying that a successful communication channel was established. The evaluator shall then attempt to initiate communications between the MDM Server and a second client device over a trusted channel established using the unique certificate from the first device, verifying that the MDM Server rejects this attempt at communication. • **Test 2:** [conditional] If the MDM agent does not allow for the loading of certificates: The evaluator shall concurrently enroll 10 devices and ensure that the client certificate for each is unique, per the methods described in the TSS.
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].	The evaluator shall examine the TSS and user documents to ensure that they describe what security management functions are restricted to the administrator and what actions can be taken for each management function. The evaluator shall verify that the security management functions are restricted to authorized administrators and the administrator is only able to take the actions as described in the user documents. • **Test 1:** The evaluator shall attempt to access the functions

ID	Requirement	Assurance Activity
	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.	and policies in FMT_SMF.1(1) as an unauthorized user and verify that the attempt fails. • **Test 2:** [conditional] The evaluator shall attempt to access the functions and policies in FMT_SMF.1(3) as an unauthorized user and verify that the attempt fails.
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.	The evaluator shall examine the TSS and verify that it describes how unauthorized users are prevented from enrolling in the MDM services. The test of this function is performed in conjunction with FIA_ENR_EXT.1.
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. ***This is a selection-based requirement. Its inclusion depends upon selection in .***	The evaluator shall examine the TSS to determine that all methods of initiating an application download or update push are specified. The evaluator shall confirm that the operational guidance contains how to initiate an application download or update push. The evaluator shall ensure that the MAS Server verifies that the mobile device is enrolled in the MDM Server and is in a compliant state. The evaluator shall verify that an application cannot be downloaded from the MAS Server prior to enrolling the device with the MDM. The evaluator shall partially enroll the mobile device, so the device is connected to the MDM Server, but is not compliant and verify that applications cannot be downloaded.
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.	Policies must be digitally signed by the enterprise using the algorithms in FCS_COP.1(3). The evaluator shall ensure that the TSS describes how policies are digitally signed by the TSF. If applicable, the evaluator shall verify that the AGD guidance instructs administrators on configuring the Enterprise certificate to be used for signing policies or signing the policies before applying them. The evaluator shall perform a policy update in accordance with FMT_SMF.1(1). The evaluator shall examine the policy either at the MDM Server, in transmission, or at the MDM agent, and verify the TSF signs the update and provides it to the MDM Agent.
FMT_SAE_EXT.1.1	The TSF shall be capable to specify a configurable expiration time for enrollment authentication data. ***This is currently an objective requirement.***	
FMT_SAE_EXT.1.2	The TSF shall be able to deny enrollment after the expiration time for the enrollment authentication data has passed. ***This is currently an objective requirement.*** **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).	The evaluator shall verify that the TSS contains a description of the process of enrollment for each MDM Agent/platform listed as supported in the ST. This description shall the method of user authentication (username/password, token, etc.). The evaluator shall check to ensure that the operational guidance contains instructions to configure the expiration time for each method of user authentication listed in the TSS. • **Test 1:** The evaluator shall configure the MDM Server according to the administrative guidance to set an expiration time for the enrollment authentication data. For each method of user authentication listed in the TSS, the evaluator shall attempt to enroll using authentication data that has expired. The evaluator shall verify that enrollment was unsuccessful.
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)*,	The evaluator shall examine the TSS to ensure that it describes each management function listed. The evaluator shall examine the TSS to verify that any differences between management functions and policies for each supported MDM Agent/platform are listed. The evaluator shall also examine the ST of the claimed Mobile Device to verify that the selections and assignments in the functions and policies in the TSS do not exceed the capabilities of the supported MD. The evaluator shall examine the TSS to ensure that it identifies the management functions implemented for each supported MDM Agent/platform, which are likely to be subsets of all of the management functions available to the administrator on the MDM Server. For each MDM Agent/platform listed as supported in the ST: • **Test 1:** The evaluator shall verify the ability to command each MDM Agent functional capability and configure each MDM Agent policy listed above.

ID	Requirement	Assurance Activity
	*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)* ,	

ID	Requirement	Assurance Activity
	. *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	

ID	Requirement	Assurance Activity
	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.	
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.	The evaluator shall examine the TSS to ensure that it describes each management function listed. For function c.4, the evaluator shall examine the TSS to ensure that it describes the privacy-sensitive information that the TOE has the capability to collect from enrolled mobile devices. The evaluator shall verify the AGD guidance includes detailed instructions of what options are available and how to configure each management functional capability listed. The tests of functions b, c.1, c.2, and c.5 are performed in conjunction with the use of the function. Test 3 also covers function c.4. The evaluator shall perform the following test: • **Test 1:** The evaluator shall configure the TSF authentication certificate(s) and verify that the correct certificate is used in established trusted connections (FPT_ITT.1(1), FPT_ITT.1(2), FTP_ITC.1(1), and FTP_TRP.1(2)). • **Test 2:** (conditional) The evaluator shall configure the periodicity for the assigned list of commands to the agent for several configured time periods and shall verify that the MDM Server performs the commands schedule. • **Test 3:** (conditional) The evaluator shall design and perform tests to demonstrate that the assigned function may be configured and that the intended behavior of the function is enacted by the MDM 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*] ***This is a selection-based requirement. Its inclusion depends upon selection in .*** **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.	The evaluator shall examine the TSS to ensure that it describes each management function listed. The evaluator shall examine the TSS to determine if the MAS Server creates its own groups or relies on the groups specified by the MDM Server. The evaluator shall confirm that the operational guidance contains how to create and define user groups and how to specify which applications are accessible by which group. The evaluator shall verify the AGD guidance includes detailed instructions of what options are available and how to configure each management functional capability listed. The evaluator shall ensure that the MAS client can only access the applications specified for the group they are enrolled in. The evaluator shall create a user group, making sure that the MAS client user is excluded from the group. Verify that an application accessible to that group cannot be accessed. The evaluator shall include the MAS client user in the group and assure that the application can be accessed.
FMT_SMR.1.1(1)	**Refinement:** The TSF shall maintain the roles administrator, MD user, and	

ID	Requirement	Assurance Activity
	[**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.	The evaluator shall examine the TSS to verify that it describes the administrator role and the powers granted to and limitations of the role. The evaluator shall review the operational guidance to ensure that it contains instructions for administering the TOE and which interfaces are supported. In the course of performing the testing activities for the evaluation, the evaluator shall use all supported interfaces, although it is not necessary to repeat each test involving an administrative action with each interface. The evaluator shall ensure, however, that each supported method of administering the TOE that conforms to the requirements of this PP be tested; for instance, if the TOE can be administered through a local hardware interface or HTTPS then both methods of administration must be exercised during the evaluation team's test activities.
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*]. ***This is a selection-based requirement. Its inclusion depends upon selection in .***	
FMT_SMR.1.2(2)	**Refinement:** The MAS Server shall be able to associate users with roles. ***This is a selection-based requirement. Its inclusion depends upon selection in .*** **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.	The evaluator shall examine the TSS to verify that it describes the administrator role and the powers granted to and limitations of the role. The evaluator shall review the operational guidance to ensure that it contains instructions for administering the TOE and which interfaces are supported. In the course of performing the testing activities for the evaluation, the evaluator shall use all supported interfaces, although it is not necessary to repeat each test involving an administrative action with each interface. The evaluator shall ensure, however, that each supported method of administering the TOE that conforms to the requirements of this PP be tested; for instance, if the TOE can be administered through a local hardware interface or HTTPS then both methods of administration must be exercised during the evaluation team's test activities.
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.	The evaluator shall verify that the TSS lists the platform APIs used by the MDM software. The evaluator shall then compare the list with the supported APIs (available through e.g. developer accounts, platform developer groups) and ensure that all APIs listed in the TSS are supported.
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.	The evaluator shall verify that the TSS lists the libraries used by the MDM software. The evaluator shall verify that libraries found to be packaged with or employed by the MDM software are limited to those in the assignment.
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. ***This is a selection-based requirement. Its inclusion depends upon selection in .*** **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	The evaluator shall examine the TSS to determine that the methods and protocols used to protect distributed TOE components are described. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST. If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). The evaluator shall confirm that the operational guidance contains instructions for establishing the communication paths for each supported method. • **Test 1:** The evaluator shall ensure that communications using each specified (in the operational guidance) communication method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful. • **Test 2:** The evaluator shall ensure, for each method of communication, the channel data is not sent in plaintext. Further evaluation activities are associated with the specific protocols.

ID	PP Module Requirement	Assurance Activity
	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.	
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. ***This is a selection-based requirement. Its inclusion depends upon selection in .*** **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.	The evaluator shall examine the TSS to determine that the methods and protocols used to protect distributed TOE components are described. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST. If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). The evaluator shall confirm that the operational guidance contains instructions for establishing the communication paths for each supported method. • **Test 1:** The evaluator shall ensure that communications using each specified (in the operational guidance) communication method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful. • **Test 2:** The evaluator shall ensure, for each method of communication, the channel data is not sent in plaintext. Further evaluation activities are associated with the specific protocols.
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.	If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). If "implement functionality" is selected, the evaluator shall examine the TSS to ensure that it details the self-tests that are run by the TSF on start-up; this description should include an outline of what the tests are actually doing (e.g., rather than saying "memory is tested", a description similar to "memory is tested by writing a value to each memory location and reading it back to ensure it is identical to what was written" shall be used). The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly. The evaluator shall examine the TSS to ensure that it describes how to verify the integrity of stored TSF executable code when it is loaded for execution. The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the integrity of stored TSF executable code has not been compromised. The evaluator also ensures that the TSS (or the operational guidance)

ID	Requirement	Assurance Activity
		describes the actions that take place for successful (e.g. hash verified) and unsuccessful e.g., hash not verified) cases. The evaluator shall perform the following tests: • **Test 1:** The evaluator performs the integrity check on a known good TSF executable and verifies that the check is successful. • **Test 2:** The evaluator modifies the TSF executable, performs the integrity check on the modified TSF executable and verifies that the check fails.
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.	The evaluator shall ensure that the administrator guidance includes instructions for determining the current version of the TOE. The evaluator shall query the TSF for the current version of the software according to the AGD guidance and shall verify that the current version matches that of the documented and installed version.
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.	If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).
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.	If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). If "implement functionality" is selected, the evaluator shall examine the TSS and verify that it describes the standards by which the updates are digitally signed and how the signature verification process is implemented. The evaluator shall examine the AGD guidance to verify that it describes how to query the current version of the TSF software, how to initiate updates and how to check the integrity of updates prior to installation. The evaluator shall perform the following tests: • **Test 1:** The evaluator shall attempt to initiate an update digitally signed by the vendor and verify that the update is successfully installed. • **Test 2:** The evaluator shall attempt to install an update not digitally signed by the vendor and verify that either the signature can be checked (allowing the update to be aborted) or the update is not installed.
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. ***This is an optional requirement. It may be required by Extended Packages of this Protection Profile.*** **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.	If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). If "implement functionality" is selected, the TSS shall describe when the banner is displayed. The evaluator follows the operational guidance to configure a notice and consent warning message. The evaluator shall also perform the following test: The evaluator shall start up or unlock the TSF. The evaluator shall verify that the notice and consent warning message is displayed in each instance described in the TSS.
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)*].	The evaluator shall ensure that the TSS contains whether the MDM Server communication channel is internal or external to the TOE. This testing can be completed in conjunction with the testing for FPT_ITT.1(1)/FPT_ITT.1(2), FTP_ITC.1(2) or FTP_ITC.1(3).
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* *]*	If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).

ID	Requirement	Assurance Activity
	]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.	If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).
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.	The evaluator shall examine the TSS to determine that the methods of communication with authorized IT entities are indicated, along with how those communications are protected. If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). The evaluator shall confirm that the operational guidance contains instructions for configuring the communication channel between the MDM Server and authorized IT entities for each supported method.

ID	Requirement	Assurance Activity
		• **Test 1:** The evaluators shall ensure that communications using each specified (in the operational guidance) communication method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful. • **Test 2:** The evaluator shall ensure, for each method of communication, the channel data is not sent in plaintext. • **Test 3:** The evaluator shall ensure, for each communication channel with the MDM Server, that a protocol analyzer identifies the traffic as the protocol under testing. Further evaluation activities are associated with the specific protocols.
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. ***This is a selection-based requirement. Its inclusion depends upon selection in .*** **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.	If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).
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.	If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM

ID	Requirement	Assurance Activity
	This is a selection-based requirement. Its inclusion depends upon selection in .	Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).
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 ***This is a selection-based requirement. Its inclusion depends upon selection in .***	The evaluator shall examine the TSS to determine that the methods of Agent-Server communication are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of remote TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST. If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). The evaluator shall confirm that the operational guidance contains instructions for configuring the communication channel between the MDM Agent and the MDM Server for each supported method. • **Test 1:** The evaluators shall ensure that communications using each specified (in the operational guidance) Agent-Server communication method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful. • **Test 2:** The evaluator shall ensure, for each method of Agent-Server communication, the channel data is not sent in plaintext. • **Test 3:** The evaluator shall ensure, for each communication channel with the MDM Server, that a protocol analyzer identifies the traffic as the protocol under testing. Further evaluation activities are associated with the specific protocols.
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].	If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).
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.	If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).
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.	The evaluator shall examine the TSS to determine that the methods of remote TOE administration are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST. If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). The evaluator shall confirm that the operational guidance contains instructions for establishing the remote administrative sessions for each supported method. The evaluator shall also perform the following tests: • **Test 1:** The evaluators shall ensure that communications using each specified (in the operational guidance) remote administration method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful. • **Test 2:** For each method of remote administration supported, the evaluator shall follow the operational guidance to ensure that there is no available interface that can be used by a remote user to establish remote administrative sessions without invoking the trusted path. • **Test 3:** The evaluator shall ensure, for each method of remote administration, the channel data is not sent in plaintext. Further evaluation activities are associated with the specific

ID	Requirement	Assurance Activity
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].	protocols. If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).
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.	If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).
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.	The evaluator shall examine the TSS to determine that the methods of remote enrollment are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of enrollment are consistent with those specified in the requirement, and are included in the requirements in the ST. If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). The evaluator shall confirm that the operational guidance contains instructions for establishing the enrollment sessions for each supported method. For each MDM Agent/platform listed as supported in the ST: • **Test 1:** The evaluators shall ensure that communications using each specified (in the operational guidance) enrollment method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful. • **Test 2:** For each method of enrollment supported, the evaluator shall follow the operational guidance to ensure that there is no available interface that can be used by a remote user to establish enrollment sessions without invoking the trusted path. • **Test 3:** The evaluator shall ensure, for each method enrollment, the channel data is not sent in plaintext. Further evaluation activities are associated with the specific protocols.
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*]. ***This is currently an objective requirement.***	If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).
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. ***This is currently an objective requirement.***	If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity).
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*]]. ***This is currently an objective requirement.*** **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	The evaluator shall examine the TSS to determine that the methods of joining TOE components are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of joining are consistent with those specified in the requirement, and are included in the requirements in the ST. If "invoke platform-provided functionality" is selected, the evaluator shall examine the TSS to verify that it describes (for each supported platform) how this functionality is invoked (it should be noted that this may be through a mechanism that is not implemented by the MDM Server; nonetheless, that mechanism will be identified in the TSS as part of this evaluation activity). The evaluator shall confirm that the operational guidance contains instructions for joining TOE components for each supported method. The evaluator shall also perform the following tests: • **Test 1:** The evaluator shall ensure that the communications path for joining components to the TSF is tested for each distinct (nonequivalent) component type, setting up the connections as described in the guidance documentation and ensuring that communication is successful. In particular the evaluator shall confirm that requirements on environment

ID	Requirement	Assurance Activity
	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.	protection for the registration process are consistent with observations made on the test configuration (for example, a requirement to isolate the components from the Internet during registration might be inconsistent with the need for a component to contact a license server). If no requirements on the registration environment are identified as necessary to protect confidentiality, then the evaluator shall confirm that the key used for registration can be configured (following the instructions in the guidance documentation) to be at least the same length as the key used for the internal TSF channel that is being enabled. The evaluator shall confirm that the key used for the channel is unique to the pair of components (this is done by identifying the relevant key during the registration test: it is not necessary to examine the key value). • **Test 2:** The evaluator shall follow the guidance documentation to ensure that in fact the communication channel can be enabled by an administrator for all the TOE components identified in the guidance documentation as capable of initiation. • **Test 3:** The evaluator shall ensure that if the guidance documentation states that the channel data is encrypted then the data observed on the channel is not plaintext. • **Test 4:** The evaluator shall ensure that, for each different pair of nonequivalent component types that can use the registration channel, the connection is physically interrupted during a joining attempt. The evaluator shall ensure that when physical connectivity is restored, communications are appropriately protected. Further evaluation activities are associated with the specific protocols.

Security Assurance Requirements

ID	Requirement	Assurance Activity
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.	There are no specific evaluation activities associated with these SARs, except ensuring the information is provided. The functional specification documentation is provided to support the evaluation activities described in Section 5 and the relevant appendices, and other activities described for AGD, ATE, and AVA SARs. The requirements on the content of the functional specification information is implicitly assessed by virtue of the other evaluation activities being performed; if the evaluator is unable to perform an activity because there is insufficient interface information, then an adequate functional specification has not been provided.
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	

ID	Requirement	Assurance Activity
	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.	Some of the contents of the operational guidance will be verified by the evaluation activities in Sections 4.2, 4.3, and 4.4and evaluation of the TOE according to the CEM. The following additional information is also required. If cryptographic functions are provided by the TOE, the operational guidance shall contain instructions for configuring the cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE. The documentation must describe the process for verifying updates to the TOE by verifying a digital signature - this may be done by the TOE or the underlying platform. The evaluator shall verify that this process includes the following steps: Instructions for obtaining the update itself. This should include instructions for making the update accessible to the TOE (e.g., placement in a specific directory). Instructions for initiating the update process, as well as discerning whether the process was successful or unsuccessful. This includes generation of the hash/digital signature. The TOE will likely contain security functionality that does not fall in the scope of evaluation under this PP. The operational guidance shall make it clear to an administrator which security functionality is covered by the evaluation activities.
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.	As indicated in the introduction above, there are significant expectations with respect to the documentation, especially when configuring the operational environment to support TOE functional requirements. The evaluator shall check to ensure that the guidance provided for the TOE adequately addresses all platforms claimed for the TOE in the ST.

ID	Requirement	Assurance Activity
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.	The evaluator shall check the ST to ensure that it contains an identifier (such as a product name/version number) that specifically identifies the version that meets the requirements of the ST. Further, the evaluator shall check the AGD guidance and TOE samples received for testing to ensure that the version number is consistent with that in the ST. If the vendor maintains a website advertising the TOE, the evaluator shall examine the information on the website to ensure that the information in the ST is sufficient to distinguish the product.
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.	The "evaluation evidence required by the SARs" in this PP is limited to the information in the ST coupled with the guidance provided to administrators and users under the AGD requirements. By ensuring that the TOE is specifically identified and that this identification is consistent in the ST and in the AGD guidance (as done in the evaluation activity for ALC_CMC.1), the evaluator implicitly confirms the information required by this component. Life-cycle support is targeted aspects of the developer's life-cycle and instructions to providers of applications for the developer's devices, rather than an in-depth examination of the TSF manufacturer's development and configuration management process. This is not meant to diminish the critical role that a developer's practices play in contributing to the overall trustworthiness of a product; rather, it's a reflection on the information to be made available for evaluation. The evaluator shall ensure that the TSF is uniquely identified (with respect to other products from the TSF vendor), and that documentation provided by the developer in association with the requirements in the ST is associated with the TSF using this unique identification.
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.	The evaluator shall prepare a test plan and report documenting the testing aspects of the system. The test plan covers all of the testing actions contained in the CEM and the body of this PP's Evaluation Activities. While it is not necessary to have one test case per test listed in an evaluation activity, the evaluator must document in the test plan that each applicable testing requirement in the ST is covered. The test plan identifies the platforms to be tested, and for those platforms not included in the test plan but included in the ST, the test plan provides a justification for not testing the platforms. This justification must address the differences between the tested platforms and the untested platforms, and make an argument that the differences do not affect the testing to be performed. It is not sufficient to merely assert that the differences have no affect; rationale must be provided. If all platforms claimed in the ST are tested, then no rationale is necessary. The test plan describes the composition of each platform to be tested, and any setup that is necessary beyond what is contained in the part of a test or as a standard pre-test condition. This may include special test drivers or tools. For each driver or tool, an argument (not just an assertion) should be provided that the driver or tool will not adversely affect the performance of the functionality by the TOE and its platform. This also includes the configuration of the cryptographic engine to be used. The cryptographic algorithms implemented by this engine are those specified by this PP and used by the cryptographic protocols being evaluated (IPsec, TLS/HTTPS, SSH). The test plan identifies high-level test objectives as well as the test procedures to be followed to achieve those objectives. These procedures include expected results. The test report (which could just be an annotated version of the test plan) details the activities that took place when the test procedures were executed, and includes the actual results of the tests. This shall be a cumulative account, so if there was a test run that resulted in a failure; a fix installed; and then a successful re-run of the test, the report would show a "fail" and "pass" result (and the supporting details), and not just the "pass" result.
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.	As with ATE_IND, the evaluator shall generate a report to document their findings with respect to this requirement. This report could physically be part of the overall test report mentioned in ATE_IND, or a separate document. The evaluator performs a search of public information to determine the vulnerabilities that have been found in network infrastructure devices and the implemented communication protocols in general, as well as those that pertain to the particular TOE. The evaluator documents the sources consulted and the vulnerabilities found in the report. For each vulnerability found, the evaluator either provides a rationale with respect to its non-applicability, or the evaluator formulates a test (using the guidelines provided in ATE_IND) to confirm the vulnerability, if suitable. Suitability is determined by assessing the attack vector needed to take advantage of the vulnerability. If exploiting the vulnerability requires expert skills and an electron microscope, for instance, then a test would not be suitable and an appropriate justification would be formulated.

Glossary

Common Criteria (CC)	Common Criteria for Information Technology Security Evaluation (International Standard ISO/IEC 15408).
Common Criteria Testing Laboratory	Within the context of the Common Criteria Evaluation and Validation Scheme (CCEVS), an IT security evaluation facility, accredited by the National Voluntary Laboratory Accreditation Program (NVLAP) and approved by the NIAP Validation Body to conduct Common Criteria-based evaluations.
Common Evaluation Methodology (CEM)	Common Evaluation Methodology for Information Technology Security Evaluation.
Extended Package (EP)	An implementation-independent set of security requirements for a category of products, which extends those in a Protection Profile.
Protection Profile (PP)	An implementation-independent set of security requirements for a category of products.
Protection Profile Module (PP-Module)	An implementation-independent statement of security needs for a TOE type complementary to one or more Base Protection Profiles..
Security Assurance Requirement (SAR)	A requirement for how the TOEs proper implementation of the SFRs is verified by an evaluator.
Security Functional Requirement (SFR)	A requirement for security enforcement by the TOE.
Security Target (ST)	A set of implementation-dependent security requirements for a specific product.
Target of Evaluation (TOE)	The product under evaluation.
TOE Security Functionality (TSF)	The security functionality of the product under evaluation.
TOE Summary Specification (TSS)	A description of how a TOE satisfies the SFRs in an ST.
TSF Data	Data for the operation of the TSF that is used to enforce its security requirements.

Administrator	The person who is responsible for management activities, including setting the policy that is applied by the enterprise on the mobile device.
Application Programming Interface (API)	A specification of routines, data structures, object classes, and variables that allows an application to make use of services provided by another software component, such as a library. API's are often provided for a set of libraries included with the platform.
Critical Security Parameter (CSP)	Security-related information whose disclosure or modification can compromise the security of a cryptographic module and/or authentication system.
Data	Program or application or data files that are stored or transmitted by a server or MD.
Data Encryption Key (DEK)	A key used to encrypt data-at-rest.
Developer Modes	States in which additional services are available to a user in order to provide enhanced system access for debugging of software.
Enrolled State	The state in which a mobile device is managed by a policy from an MDM.
Enterprise Applications	Applications that are provided and managed by the enterprise as opposed to a public application store.
Enterprise Data	Any data residing in enterprise servers or temporarily stored on mobile devices to which the mobile device user is allowed access according to the security policy defined by the enterprise and implemented by the administrator.
Enrollment over Secure Transport (EST)	Cryptographic protocol that describes an X.509 certificate management protocol targeting public key infrastructure (PKI) clients that need to acquire client certificates and associated certificate authority (CA) certificates.
Key Encryption Key (KEK)	A key that is used to encrypt other keys, such as (DEKs) or storage repositories that contain keys.
Locked State	Mobile device state where the device is powered on but most functionality is unavailable for use without authentication.
Mobile Device (MD)	A device which is composed of a hardware platform and its system software. The device typically provides wireless connectivity and may include software for functions like secure messaging, email, web, VPN connection, and VoIP (Voice over IP), for access to the protected enterprise network, enterprise data and applications, and for communicating to other MDs.
Mobile Device Management (MDM)	Products that allow enterprises to apply security policies to MDs. This system consists of two primary components: the MDM Server and the MDM Agent.
Mobile Device User	The person who uses and is held responsible for a MD.
Operating System (OS)	Software which runs at the highest privilege level and can directly control hardware resources. Modern mobile devices typically have at least two primary operating systems: one which runs on the cellular baseband processor and one which runs on the application processor. The platform of the application processor handles most user interaction and provides the execution environment for apps. The platform of the cellular baseband processor handles communications with the cellular network and may control other peripherals. The term OS, without context, may be assumed to refer to the platform of the application processor.
Powered-Off State	Mobile device shutdown state.
Protected Data	All non-TSF data on the mobile device, including user or enterprise data. Protected data is encrypted while the mobile device is in the powered-off state. This includes keys in software-based storage. May overlap with sensitive data.
Root Encryption Key (REK)	A key tied to a particular device that is used to encrypt all other keys for that device.
Sensitive Data	Data that is encrypted by the mobile device. May include all user or enterprise data or may be data for specific applications such as emails, messaging, documents, calendar items, or contacts. May be protected while the mobile device is in the locked state. Must include at minimum some keys in software-based key storage.
Trust Anchor Database	A list of trusted root Certificate Authority certificates.
Unenrolled	Mobile device state when it is not managed by an MDM.

State	
Unlocked State	Mobile device state where it is powered on and its functionality is available for use.