BlackBerry Smartphones with OS 10.3.3

Security Target

Doc No: 1958-001-D102
Version: 1.10
9 January 2017

BlackBerry
2200 University Ave. E
Waterloo, Ontario, Canada
N2K 0A7

Prepared by:
EWA-Canada
1223 Michael Street, Suite 200
Ottawa, Ontario, Canada
K1J7T2

CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 SECURITY TARGET INTRODUCTION

This Security Target (ST) defines the scope of the evaluation in terms of the assumptions made, the intended environment for the TOE, the Information Technology (IT) security functional and assurance requirements to be met, and the level of confidence (evaluation assurance level) to which it is asserted that the TOE satisfies its IT security requirements. This document forms the baseline for the Common Criteria (CC) evaluation.

1.1 DOCUMENT ORGANIZATION

Section 1, ST Introduction, provides the Security Target (ST) reference, the Target of Evaluation (TOE) reference, the TOE overview and the TOE description.

Section 2, Conformance Claims, describes how the ST conforms to the Common Criteria and Packages. The ST does not conform to a Protection Profile.

Section 3, Security Problem Definition, describes the expected environment in which the TOE is to be used. This section defines the set of threats that are relevant to the secure operation of the TOE, organizational security policies with which the TOE must comply, and secure usage assumptions applicable to this analysis.

Section 4, Security Objectives, defines the set of security objectives to be satisfied by the TOE and by the TOE operating environment in response to the problem defined by the security problem definition.

Section 5, Extended Components Definition, defines the extended components which are then detailed in Section 6.

Section 6, Security Requirements, specifies the security functional and assurance requirements that must be satisfied by the TOE and the Information Technology (IT) environment.

Section 7, TOE Summary Specification, describes the security functions that are included in the TOE to enable it to meet the IT security functional requirements.

Section 8 Terminology and Acronyms, defines the acronyms and terminology used in this ST.

1.2 SECURITY TARGET REFERENCE

ST Title:	BlackBerry Smartphones with OS 10.3.3 Security Target
ST Version:	1.10
ST Date:	9 January 2017

1.3 TOE REFERENCE

TOE Identification:	BlackBerry Smartphones with OS 10.3.3.1668
TOE Developer:	BlackBerry
TOE Type:	Mobile Device

1.4 TOE OVERVIEW

In the context of the Protection Profile for Mobile Device Fundamentals (MDF PP), a mobile device is 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 Mobile Devices.

The BlackBerry devices running Operating System (OS) 10.3.3 are intended to satisfy the Use Case for an enterprise-owned device for specialized, high-security use. This use case describes an enterprise-owned device with intentionally-limited network connectivity, tightly-controlled configuration, and limited software inventory appropriate for specialized, high-security use cases. For example, the device may not be permitted connectivity to any external peripherals. It may only be able to communicate via its Wi-Fi or cellular radios with the enterprise-run network, which may not even permit connectivity to the Internet. Use of the device may entail compliance with policies that are more restrictive than those in any general-purpose use case, yet may mitigate risks to highly sensitive information. The enterprise will look for additional applications providing enterprise connectivity and services to have a similar level of assurance as the platform.

The BlackBerry smartphones running OS 10.3.3 together fulfill the requirements of both the Mobile Device and the Agent as described in the claimed Protection Profiles.

BlackBerry smartphones running OS 10.3.3, used with BlackBerry Enterprise Service (BES), allow mobile workers secure access to mail, application and content servers in the organization's network, in accordance with the organization’s policies. Evaluation of the BES 12.5 against the Protection Profile for Mobile Device Management Version 2.0 will be performed in parallel with this evaluation.

It should be noted that although there are more security policy enforcement options supported by BlackBerry smartphone mobile devices than described in this document, only those features described by the Security Functional Requirements (SFRs) have been tested as part of this evaluation.

The TOE is a combined hardware and software TOE.

1.5 TOE DESCRIPTION

1.5.1 Physical Scope

The mobile device is the BlackBerry 10.3.3 software running on one of the following BlackBerry smartphone devices:

- Passport
- Classic
- Leap
- Z30
- Z10
- Q10
- P'9982 (Porsche Design)
- P'9983 (Porsche Design)

Device Name	Model Number	WiFi Support	Cellular Support
Classic	SQC100-1	802.11/a/b/g/n	GSM/GPRS/EDGE 2, 3, 5, 8 UMTS/HSPA+/DC-HSPA 1, 2, 5/6, 8 FDD-LTE 1, 2, 3, 5, 7, 8
	SQC100-2	802.11/a/b/g/n	GSM/GPRS/EDGE 2, 3, 5, 8 UMTS/HSPA+ 1, 2, 5/6 FDD-LTE 1, 2, 3, 4, 5, 7, 17, 20
	SQC100-3	802.11/a/b/g/n	CDMA 0, 1 GSM/GPRS/EDGE 2, 3, 5, 8 UMTS/HSPA+ 1, 2, 5/6, 8 FDD-LTE 3, 4, 7, 13
	SQC100-4	802.11/a/b/g/n	GSM/GPRS/EDGE 2, 3, 5, 8 UMTS/HSPA+/DC-HSPA 1, 2, 4, 5/6 FDD-LTE 1, 2/25, 4, 5, 7, 13, 17
	SQC100-5	802.11/a/b/g/n	CDMA 0, 1 GSM/GPRS/EDGE 2, 3, 5, 8 UMTS/HSPA+ 1, 2, 5/6, 8 FDD-LTE 3, 4, 7, 13
Passport	SQW100-1	802.11/a/b/g/n/ac	GSM/GPRS/EDGE 2, 3, 5, 8 UMTS/HSPA+ 1, 2, 3, 4, 5/6, 8 FDD-LTE 1, 2, 3, 4, 5, 7, 8, 13, 17, 20
	SQW100-3	802.11/a/b/g/n/ac	GSM/GPRS/EDGE 2, 3, 5, 8 UMTS/HSPA+ 1, 2, 4, 5,/6, 8 FDD-LTE 1, 2, 3, 4, 5, 7, 8, 17, 20, 29

Device Name	Model Number	WiFi Support	Cellular Support
	SQW100-4	802.11/a/b/g/n/ac	GSM/GPRS/EDGE 2, 3, 5, 8 UMTS/HSPA+ 1, 2, 3, 4, 5/6, 8 FDD-LTE 1, 2, 3, 4, 5, 7, 8, 13, 17, 20
Leap	STR100-1	802.11/b/g/n	GSM/GPRS/EDGE 2, 3, 5, 8 UMTS/HSPA+/DC-HSPA 1, 2, 5/6, 8 FDD-LTE 1, 3, 7, 8, 20
	STR100-2	802.11/b/g/n	GSM/GPRS/EDGE 2, 3, 5, 8 UMTS/HSPA+/DC-HSPA 1, 2, 4, 5/6 FDD-LTE 1, 2/25, 4, 5, 7, 13, 17
Z30	STA100-2	802.11/a/b/g/n	GSM/GPRS/EDGE 2, 3, 5, 8 UMTS/HSPA+ 1, 2, 5/6, 8 FDD-LTE 3, 7, 8, 20
	STA100-3	802.11/a/b/g/n	CDMA 0, 1 GSM/GPRS/EDGE 2, 3, 5, 8 UMTS/HSPA+ 1, 2, 5/6, 8 FDD-LTE 4, 13
	STA100-5	802.11/a/b/g/n	GSM/GPRS/EDGE 2, 3, 5, 8 UMTS/HSPA+ 1, 2, 4, 5/6 FDD-LTE 4, 5, 7
	STA100-6	802.11/a/b/g/n	GSM/GPRS/EDGE 2, 3, 5, 8 UMTS/HSPA+ 1, 2, 5/6, 8 FDD-LTE 1, 3, 7, 8, 20
Q10 - Porche	SQK100-1	802.11/a/b/g/n	GSM/GPRS/EDGE 2,3,5,8 UMTS/HSPA+: 1, 2, 5/6, 8 FDD-LTE 3, 7, 8, 20
	SQK100-2	802.11/a/b/g/n	GSM/GPRS/EDGE 2, 3, 5, 8 UMTS/HSPA+: 1, 2, 5/6 FDD-LTE: 2, 4, 5, 17
Q10	SQN100-1	802.11/a/b/g/n	GSM/GPRS/EDGE 2, 3, 5, 8 UMTS/HSPA+: 1, 2, 5/6 FDD-LTE: 2, 4, 5, 17
	SQN100-2	802.11/a/b/g/n	CDMA 0, 1 GSM/GPRS/EDGE 2, 3, 5, 8 UMTS/HSPA+: 1, 2, 5/6, 8 FDD-LTE 4, 13
	SQN100-3	802.11/a/b/g/n	GSM/GPRS/EDGE 2, 3, 5, 8 HSPA+: 1, 2, 5/6, 8 FDD-LTE 3, 7, 8, 20

Device Name	Model Number	WiFi Support	Cellular Support
	SQN100-4	802.11/a/b/g/n	CDMA 0, 1 GSM/GPRS/EDGE 2,3,5,8 UMTS/HSPA+ 1, 2, 5/6, 8 FDD-LTE 25
	SQN100-5	802.11/a/b/g/n	GSM/GPRS/EDGE 2, 3, 5, 8 UMTS/HSPA+ 1, 2, 4, 5/6 LTE 2, 4, 5, 17
Z10 - Porche	STK100-1	802.11/a/b/g/n	GSM/GPRS/EDGE 2, 3, 5, 8 UMTS/HSPA+ 1, 5/6, 8 LTE 3, 7, 8, 20
	STK100-2	802.11/a/b/g/n	GSM/GPRS/EDGE 2, 3, 5, 8 UMTS/HSPA 1, 2, 4, 5/6 FDD-LTE 2, 4, 5, 17
Z10	STL100-2	802.11/a/b/g/n	GSM/GPRS/EDGE 2,3,5,8 UMTS/HSPA+ 1, 5/6, 8 FDD-LTE 3, 7, 8, 20
	STL100-3	802.11/a/b/g/n	GSM/GPRS/EDGE 2, 3, 5, 8 UMTS/HSPA 1, 2, 4, 5 FDD-LTE 2, 4, 5, 17
	STL100-4	802.11/a/b/g/n	CDMA 0, 1 GSM/GPRS/EDGE 2, 3, 5, 8 UMTS 1, 8 FDD-LTE 13

Table 1 – BlackBerry Smartphone Devices

Bluetooth 4.0 is supported for all listed devices.

The BlackBerry smartphone devices were tested with BES 12.5 with Policy Pack Version 1.9.3.97.

1.5.2 TOE Guidance

The TOE includes the following guidance documentation:

- BlackBerry Passport Smartphone Version: 10.3.3 User Guide
  SWD- 20160427115319481
  Published: 2016-04-27
- BlackBerry Classic Smartphone Version: 10.3.3 User Guide
  SWD-20160426135955284
  Published: 2016-04-26
- BlackBerry Leap Smartphone Version: 10.3.3 User Guide
  SWD-20160427113258316
  Published: 2016-04-27

- BlackBerry Z30 Smartphone Version: 10.3.3 User Guide
SWD-20160427134318308
Published: 2016-04-27
- BlackBerry Z10 Smartphone Version: 10.3.3 User Guide
SWD-20160427140021771
Published: 2016-04-27
- BlackBerry Q10 Smartphone Version: 10.3.3 User Guide
SWD-20160427135115364
Published: 2016-04-27
- BlackBerry P'9982 Smartphone User Guide Version: 10.3.3
SWD-20160427133504750
Published: 2016-04-27
- BlackBerry P'9983 Smartphone User Guide Version: 10.3.3
SWD-20160427132628821
Published: 2016-04-27
- BlackBerry Smartphones with OS 10.3.3 Common Criteria Guidance Supplement Version 1.1

1.5.3 Logical Scope

The logical boundary of the TOE includes all interfaces and functions within the physical boundary. The logical boundary of the TOE may be broken down by the security function classes described in Section 6. The following breakdown also provides the description of the security features of the TOE, and loosely follows the security functional classes described in Section 6. Table 2 summarizes the logical scope of the TOE.

Functional Classes	Description
Security Audit	The MDM Agent sends alerts to the MDM Server when certain conditions are met.
Cryptographic Support	Cryptographic functionality and key management services are provided to address requirements for the protection of data at rest and transmitted data.
User Data Protection	The TOE provides access control to keys and private data, ensuring access is granted to approved application processes only.
Identification and Authentication	Users must be authenticated prior to accessing controlled functions of the TOE. Passwords must be sufficiently complex, and the TOE must respond to multiple unsuccessful authentication attempts. The TOE must support certificate based authentication and validation of X.509 certificates.

Functional Classes	Description
Security Management	The TOE provides security management capabilities, including trusted policy update, to configure the options required to support the claimed security functionality. Users are prevented from unenrolling themselves from management.
Protection of the TSF	The design of the TOE supports anti-exploitation services, application processor mediation and meets key storage and transmission requirements. The TOE provides self-test functionality and supports trusted updates. The TOE provides reliable timestamps.
TOE Access	A banner is presented on user login. The TOE supports user and administrator initiated lockout, and may control access to various wireless networks.
Trusted Path/Channel	The communications links between the TOE and other entities are protected using standard protocols.

Table 2 – Logical Scope of the TOE

2 CONFORMANCE CLAIMS

2.1 COMMON CRITERIA CONFORMANCE CLAIM

This Security Target claims to be conformant to Version 3.1 of Common Criteria for Information Technology Security Evaluation according to:

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; CCMB-2012-09-001, Version 3.1, Revision 4, September 2012
- Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components; CCMB-2012-09-002, Version 3.1, Revision 4, September 2012
- Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components CCMB-2012-09-003, Version 3.1, Revision 4, September 2012

As follows:

- CC Part 2 extended
- CC Part 3 extended

The Common Methodology for Information Technology Security Evaluation, Version 3.1, Revision 4, September 2012 has to be taken into account.

2.2 ASSURANCE PACKAGE CLAIM

This Security Target does not claim conformance to an Assurance Package, but conforms to the Security Assurance Requirements described in Section 6 of the Protection Profile for Mobile Device Fundamentals.

2.3 PROTECTION PROFILE CONFORMANCE CLAIM

The Security Target claims exact conformance with the National Information Assurance Partnership (NIAP) Protection Profile for Mobile Device Fundamentals Version 2.0 dated 17 September 2014. The claimed mobile devices are intended to address the security problems associated with the Enterprise-owned device for specialized, high-security use case. Compliance considers the following Technical Decisions: TD0028, TD0030, TD0034, TD0038, TD0057, TD0058, TD0059, TD0060, TD0064, TD079, and TD0091.

The Security Target also claims conformance to the NIAP Extended Package for Mobile Device Management Agents Version 2.0 dated 31 December 2014. Compliance considers the following Technical Decisions: TD0034, TD0057, and TD080.

3 SECURITY PROBLEM DEFINITION

3.1 THREATS

Table 3 lists the threats addressed by the TOE. Potential threat agents are authorized TOE users, and unauthorized persons. The level of expertise of both types of attacker is assumed to be unsophisticated. TOE users are assumed to have access to the TOE, extensive knowledge of TOE operations, and to possess a high level of skill. They have moderate resources to alter TOE parameters, but are assumed not to be wilfully hostile. Unauthorized persons have little knowledge of TOE operations, a low level of skill, limited resources to alter TOE parameters and no physical access to the TOE.

Mitigation to the threats is through the objectives identified in Section 4.1, Security Objectives for the TOE.

Threat Name	Threat Definition
T.EAVESDROP	If positioned on a wireless communications channel or elsewhere on the network, attackers may monitor and gain access to data exchanged between the Mobile Device and other endpoints.
T.FLAWAPP	Malicious or exploitable code could be used knowingly or unknowingly by a developer, possibly resulting in the capability of attacks against the platform's system software.
T.MALICIOUS _APPS	An administrator of the MDM or mobile device user may inadvertently import malicious code, or an attacker may insert malicious code into the TOE or OE, resulting in the compromise of TOE or TOE data.
T.NETWORK	An attacker may initiate communications with the Mobile Device or alter communications between the Mobile Device and other endpoints.
T.NETWORK _ATTACK	An attacker may masquerade as MDM Server and attempt to compromise the integrity of the mobile device by sending malicious management commands.
T.NETWORK _EAVESDROP	Unauthorized entities may intercept communications between the MDM and mobile devices to monitor, gain access to, disclose, or alter remote management commands. Unauthorized entities may intercept unprotected wireless communications between the mobile device and the Enterprise to monitor, gain access to, disclose, or alter TOE data.
T.PERSISTENT	An attacker gains and continues to have access the device, resulting it loss of integrity and possible control by both an adversary and legitimate owner.

Threat Name	Threat Definition
T.PHYSICAL	Loss of confidentiality of user data and credentials may be a result of an attacker gaining physical access to a Mobile Device.
T.PHYSICAL _ACCESS	The mobile device may be lost or stolen, and an unauthorized individual may attempt to access OE data.

Table 3 – Threats

3.2 ORGANIZATIONAL SECURITY POLICIES

Table 4 identifies the Organizational Security Policies that must be enforced by the TOE or its operational environment.

Policy Name	Policy Definition
P.ADMIN	The configuration of the mobile device security functions must adhere to the Enterprise security policy.
P.DEVICE_ENROLL	A mobile device must be enrolled for a specific user by the administrator of the MDM prior to being used in the Enterprise network by the user.
P.NOTIFY	The mobile user must immediately notify the administrator if a mobile device is lost or stolen so that the administrator may apply remediation actions via the MDM system.
P.ACCOUNTABILITY	Personnel operating the TOE shall be accountable for their actions within the TOE.

Table 4 – Organizational Security Policies

3.3 ASSUMPTIONS

The assumptions required to ensure the security of the TOE are listed in Table 5.

Assumption Name	Assumption Definition
A.CONFIG	It is assumed that the TOE's security functions are configured correctly in a manner to ensure that the TOE security policies will be enforced on all applicable network traffic flowing among the attached networks.
A.CONNECTIVITY	The TOE relies on network connectivity to carry out its management activities. The TOE will robustly handle instances when connectivity is unavailable or unreliable.

Assumption Name	Assumption Definition
A.MOBILE_ DEVICE_PLATFORM	The MDM Agent relies upon Mobile platform and hardware evaluated against the MDF PP and assured to provide policy enforcement as well as cryptographic services and data protection. The Mobile platform provides trusted updates and software integrity verification of the MDM Agent.
A.NOTIFY	It is assumed that the mobile user will immediately notify the administrator if the Mobile Device is lost or stolen.
A.PRECAUTION	It is assumed that the mobile user exercises precautions to reduce the risk of loss or theft of the Mobile Device.
A.PROPER_ADMIN	One or more competent, trusted personnel who are not careless, willfully negligent, or hostile, are assigned and authorized as the TOE Administrators, and do so using and abiding by guidance documentation.
A.PROPER_USER	Mobile device users are not willfully negligent or hostile, and use the device within compliance of a reasonable Enterprise security policy.

Table 5 – Assumptions

4 SECURITY OBJECTIVES

The purpose of the security objectives is to address the security concerns and to show which security concerns are addressed by the TOE, and which are addressed by the environment. Threats may be addressed by the TOE or the security environment or both. Therefore, the CC identifies two categories of security objectives:

- Security objectives for the TOE
- Security objectives for the environment

4.1 SECURITY OBJECTIVES FOR THE TOE

This section identifies and describes the security objectives that are to be addressed by the TOE.

Security Objective	Description
O.ACCOUNTABILITY	The TOE must provide logging facilities which record management actions undertaken by its administrators.
O.APPLY_POLICY	The TOE must facilitate configuration and enforcement of enterprise security policies on mobile devices via interaction with the mobile OS and the MDM Server. This will include the initial enrollment of the device into management, through its lifecycle including policy updates and through its possible unenrollment from management services.
O.AUTH	The TOE will provide the capability to authenticate the user and endpoints of a trusted path to ensure they are communicating with an authorized entity with appropriate privileges.
O.COMMS	The TOE will provide the capability to communicate using one (or more) standard protocols as a means to maintain the confidentiality of data that are transmitted outside of the TOE.
O.CONFIG	The TOE will provide the capability to configure and apply security policies. This ensures the Mobile Device can protect user and enterprise data that it may store or process.
O.DATA _PROTECTION _TRANSIT	Data exchanged between the MDM Server and the MDM Agent must be protected from being monitored, accessed and altered.
O.INTEGRITY	The TOE will provide the capability to perform self-tests to ensure the integrity of critical functionality, software/firmware and data has been maintained. The TOE will also provide a means to verify the integrity of downloaded updates.

Security Objective	Description
O.STORAGE	The TOE will provide the capability to encrypt all user and enterprise data and authentication keys to ensure the confidentiality of data that it stores.

Table 6 – Security Objectives for the TOE

4.2 SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT

This section identifies and describes the security objectives that are to be addressed by the IT environment or by non-technical or procedural means.

Security Objective	Description
OE.CONFIG	TOE administrators will configure the Mobile Device security functions correctly to create the intended security policy.
OE.IT _ENTERPRISE	The Enterprise IT infrastructure provides security for a network that is available to the TOE and mobile devices that prevents unauthorized access.
OE.MOBILE _DEVICE _PLATFORM	The MDM Agent relies upon the trustworthy Mobile platform and hardware to provide policy enforcement as well as cryptographic services and data protection. The Mobile platform provides trusted updates and software integrity verification of the MDM Agent.
OE.NOTIFY	The Mobile User will immediately notify the administrator if the Mobile Device is lost or stolen.
OE.PRECAUTION	The Mobile User exercises precautions to reduce the risk of loss or theft of the Mobile Device.
OE.PROPER _ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.
OE.PROPER _USER	Users of the mobile device are trained to securely use the mobile device and apply all guidance in a trusted manner.
OE.WIRELESS _NETWORK	A wireless network will be available to the mobile devices.

Table 7 – Security Objectives for the Operational Environment

4.3 SECURITY OBJECTIVES RATIONALE

The following table maps the security objectives to the assumptions, threats, and organizational policies identified for the TOE.

	T.EAVESDROP	T.FLAWAPP	T.MALICIOUS_APPS	T.NETWORK	T.NETWORK_ATTACK	T.NETWORK_EAVESDROP	T.PERSISTENT	T.PHYSICAL	T.PHYSICAL_ACCESS	P.ADMIN	P. DEVICE_ENROLL	P.NOTIFY	P.ACCOUNTABILITY	A.CONFIG	A.CONNECTIVITY	A.MOBILE_DEVICE_PLATFORM	A.NOTIFY	A.PRECAUTION	A.PROPER_ADMIN	A.PROPER_USER
O.ACCOUNTABILITY													X							
O.APPLY_POLICY			X						X											
O.AUTH	X	X		X				X												
O.COMMS	X	X		X																
O.CONFIG	X	X		X																
O.DATA _PROTECTION _TRANSIT					X	X														
O.INTEGRITY		X					X													
O.STORAGE								X												
OE.CONFIG														X						
OE.IT_ENTERPRISE											X									
OE.MOBILE _DEVICE_PLATFORM																X				
OE.NOTIFY																	X			
OE.PRECAUTION																		X		
OE.PROPER_ADMIN										X									X	
OE.PROPER_USER												X								X
OE.WIRELESS _NETWORK															X					

Table 8 – Mapping Between Objectives, Threats, OSPs, and Assumptions

4.3.1 Security Objectives Rationale Related to Threats

The security objectives rationale related to threats traces the security objectives for the TOE back to the threats addressed by the TOE.

Threat: **T.EAVESDROP**	If positioned on a wireless communications channel or elsewhere on the network, attackers may monitor and gain access to data exchanged between the Mobile Device and other endpoints.	
Objectives:	O.AUTH	The TOE will provide the capability to authenticate the user and endpoints of a trusted path to ensure they are communicating with an authorized entity with appropriate privileges.
	O.COMMS	The TOE will provide the capability to communicate using one (or more) standard protocols as a means to maintain the confidentiality of data that are transmitted outside of the TOE.
	O.CONFIG	The TOE will provide the capability to configure and apply security policies. This ensures the Mobile Device can protect user and enterprise data that it may store or process.
Rationale:	O.AUTH helps to mitigate the threat by ensuring that only identified, authenticated users have access to the TOE. O.COMMS mitigates this threat by ensuring that standard protocols that provide confidentiality are used for transmissions outside of the TOE. O.CONFIG mitigates this threat by ensuring that security policies may be applied to data storage and processing.	

Threat: **T.FLAWAPP**	Malicious or exploitable code could be used knowingly or unknowingly by a developer, possibly resulting in the capability of attacks against the platform's system software.	
Objectives:	O.AUTH	The TOE will provide the capability to authenticate the user and endpoints of a trusted path to ensure they are communicating with an authorized entity with appropriate privileges.
	O.COMMS	The TOE will provide the capability to communicate using one (or more) standard protocols as a means to maintain the confidentiality of data that are transmitted outside of the TOE.
	O.CONFIG	The TOE will provide the capability to configure and apply security policies. This ensures the Mobile Device can protect user and enterprise data that it may store or process.

	O.INTEGRITY	The TOE will provide the capability to perform self-tests to ensure the integrity of critical functionality, software/firmware and data has been maintained. The TOE will also provide a means to verify the integrity of downloaded updates.
Rationale:	O.AUTH mitigates this threat by ensuring that only authorized, authenticated users have the ability to execute code, reducing the risk that malicious code will be run. O.COMMS ensures that standard protocols are used to maintain confidentiality, reducing the risk that malicious code could be included in that code. O.CONFIG mitigates the threat by ensuring that security policies are enforced, reducing the risk that users will be able to unwittingly run malicious code. O.INTEGRITY mitigates this threat by ensuring that all software is self-tested before being allowed to run.	

Threat: **T.MALICIOUS_APPS**	An administrator of the MDM or mobile device user may inadvertently import malicious code, or an attacker may insert malicious code into the TOE or OE, resulting in the compromise of TOE or TOE data.	
Objectives:	O.APPLY_POLICY	The TOE must facilitate configuration and enforcement of enterprise security policies on mobile devices via interaction with the mobile OS and the MDM Server. This will include the initial enrollment of the device into management, through its lifecycle including policy updates and through its possible unenrollment from management services.
Rationale:	O.APPLY_POLICY mitigates this threat by ensuring that policy enforcement addresses the risk of importation of malicious code.	

Threat: **T.NETWORK**	An attacker may initiate communications with the Mobile Device or alter communications between the Mobile Device and other endpoints.	
Objectives:	O.AUTH	The TOE will provide the capability to authenticate the user and endpoints of a trusted path to ensure they are communicating with an authorized entity with appropriate privileges.
	O.COMMS	The TOE will provide the capability to communicate using one (or more) standard

		protocols as a means to maintain the confidentiality of data that are transmitted outside of the TOE.
	O.CONFIG	The TOE will provide the capability to configure and apply security policies. This ensures the Mobile Device can protect user and enterprise data that it may store or process.
Rationale:	O.AUTH mitigates this threat by ensuring that only credentialed users may initiate communications. O.COMMS mitigates the threat by ensuring that only standard protocols are supported. O.CONFIG mitigates the threat by ensuring that all communications are subject to the implemented security policies.	

Threat: **T.NETWORK _ATTACK**	An attacker may masquerade as MDM Server and attempt to compromise the integrity of the mobile device by sending malicious management commands.	
Objectives:	O.DATA _PROTECTION _TRANSIT	Data exchanged between the MDM Server and the MDM Agent must be protected from being monitored, accessed and altered.
Rationale:	O.DATA_PROTECTION_TRANSIT mitigates this threat by ensuring that a malicious individual cannot alter the data transferred between the MDM Server and the MDM Agent.	

Threat: **T.NETWORK_ EAVESDROP**	Unauthorized entities may intercept communications between the MDM and mobile devices to monitor, gain access to, disclose, or alter remote management commands. Unauthorized entities may intercept unprotected wireless communications between the mobile device and the Enterprise to monitor, gain access to, disclose, or alter TOE data.	
Objectives:	O.DATA _PROTECTION _TRANSIT	Data exchanged between the MDM Server and the MDM Agent must be protected from being monitored, accessed and altered.
Rationale:	O.DATA_PROTECTION_TRANSIT mitigates this threat by ensuring that a malicious individual cannot read the data transferred between the MDM Server and the MDM Agent.	

Threat: **T.PERSISTENT**	An attacker gains and continues to have access the device, resulting it loss of integrity and possible control by both an adversary and legitimate owner.	
Objectives:	O.INTEGRITY	The TOE will provide the capability to perform self-tests to ensure the integrity of critical functionality, software/firmware and data has been maintained. The TOE will also provide a means to verify the integrity of downloaded updates.
Rationale:	O.INTEGRITY mitigates the threat by ensuring that the software is tested prior to execution, reducing the risk that integrity of the software could be compromised without detection.	

Threat: **T.PHYSICAL**	Loss of confidentiality of user data and credentials may be a result of an attacker gaining physical access to a Mobile Device.	
Objectives:	O.AUTH	The TOE will provide the capability to authenticate the user and endpoints of a trusted path to ensure they are communicating with an authorized entity with appropriate privileges.
	O.STORAGE	The TOE will provide the capability to encrypt all user and enterprise data and authentication keys to ensure the confidentiality of data that it stores.
Rationale:	O.AUTH mitigates the threat by ensuring that only authorized users may access the Mobile Device. O.STORAGE mitigates the threat by ensuring that data may be encrypted, reducing the risk that an attacker with physical access to the Mobile Device could access the data.	

Threat: **T.PHYSICAL _ACCESS**	The mobile device may be lost or stolen, and an unauthorized individual may attempt to access OE data.	
Objectives:	O.APPLY_POLICY	The TOE must facilitate configuration and enforcement of enterprise security policies on mobile devices via interaction with the mobile OS and the MDM Server. This will include the initial enrollment of the device into management, through its lifecycle including policy updates and through its possible unenrollment from management services.

Rationale:	O.APPLY_POLICY mitigates this threat by ensuring that policy enforcement addresses the risk of access by an unauthorized individual.

4.3.2 Security Objectives Rationale Related to Organizational Security Policies

The security objectives rationale related to Organizational Security Policies (OSPs) traces the security objectives for the TOE and the Operational Environment back to the OSPs applicable to the TOE.

OSP: **P.ADMIN**	The configuration of the mobile device security functions must adhere to the Enterprise security policy.	
Objectives:	OE.PROPER _ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.
Rationale:	OE.PROPER_ADMIN addresses the policy by ensuring that administrators may be trusted to properly configure policies.	

OSP: **P.DEVICE _ENROLL**	A mobile device must be enrolled for a specific user by the administrator of the MDM prior to being used in the Enterprise network by the user.	
Objectives:	OE.IT _ENTERPRISE	The Enterprise IT infrastructure provides security for a network that is available to the TOE and mobile devices that prevents unauthorized access.
Rationale:	OE.IT_ENTERPRISE addresses the policy by ensuring that the IT infrastructure provides an appropriate Enterprise network.	

OSP: **P.NOTIFY**	The mobile user must immediately notify the administrator if a mobile device is lost or stolen so that the administrator may apply remediation actions via the MDM system.	
Objectives:	OE.PROPER_USER	Users of the mobile device are trained to securely use the mobile device and apply all guidance in a trusted manner.
Rationale:	OE.PROPER_USER addresses the policy by ensuring that users are appropriately trained and follow guidance.	

OSP: **P.ACCOUNTABILITY**	Personnel operating the TOE shall be accountable for their actions within the TOE.	
Objectives:	O.ACCOUNTABILITY	The TOE must provide logging facilities which record management actions undertaken by its administrators.
Rationale:	O.ACCOUNTABILITY addresses the policy by ensuring that logs are recorded to track administrative actions.	

4.3.3 Security Objectives Rationale Related to Assumptions

The security objectives rationale related to assumptions traces the security objectives for the operational environment back to the assumptions for the TOE's operational environment.

Assumption: **A.CONFIG**	It is assumed that the TOE's security functions are configured correctly in a manner to ensure that the TOE security policies will be enforced on all applicable network traffic flowing among the attached networks.	
Objectives:	OE.CONFIG	TOE administrators will configure the Mobile Device security functions correctly to create the intended security policy.
Rationale:	OE.CONFIG supports this assumption by ensuring that the Mobile Device is correctly configured.	

Assumption: **A.CONNECTIVITY**	The TOE relies on network connectivity to carry out its management activities. The TOE will robustly handle instances when connectivity is unavailable or unreliable.	
Objectives:	OE.WIRELESS_NETWORK	A wireless network will be available to the mobile devices.
Rationale:	OE.WIRELESS_NETWORK supports this assumption by ensuring that the required network connectivity is in place.	

Assumption: **A.MOBILE_DEVICE_PLATFORM**	The MDM Agent relies upon Mobile platform and hardware evaluated against the MDFPP and assured to provide policy enforcement as well as cryptographic services and data protection. The Mobile platform provides trusted updates and software integrity verification of the MDM Agent.

Objectives:	OE.MOBILE _DEVICE _PLATFORM	The MDM Agent relies upon the trustworthy Mobile platform and hardware to provide policy enforcement as well as cryptographic services and data protection. The Mobile platform provides trusted updates and software integrity verification of the MDM Agent.
Rationale:	OE.MOBILE_DEVICE_PLATFORM supports this assumption by ensuring that the appropriate mobile device platform is available.	

Assumption: **A.NOTIFY**	It is assumed that the mobile user will immediately notify the administrator if the Mobile Device is lost or stolen.	
Objectives:	OE.NOTIFY	The Mobile User will immediately notify the administrator if the Mobile Device is lost or stolen.
Rationale:	OE.NOTIFY supports this assumption by ensuring that the Mobile User notifies the administrator if the Mobile Device is lost or stolen.	

Assumption: **A.PRECAUTION**	It is assumed that the mobile user exercises precautions to reduce the risk of loss or theft of the Mobile Device.	
Objectives:	OE.PRECAUTION	The Mobile User exercises precautions to reduce the risk of loss or theft of the Mobile Device.
Rationale:	OE.PRECAUTION supports this assumption by ensuring that the Mobile User protects the Mobile Device from loss or theft.	

Assumption: **A.PROPER _ADMIN**	One or more competent, trusted personnel who are not careless, willfully negligent, or hostile, are assigned and authorized as the TOE Administrators, and do so using and abiding by guidance documentation.	
Objectives:	OE.PROPER _ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.
Rationale:	OE.PROPER_ADMIN supports this assumption by ensuring that trusted administrators are available to manage the TOE.	

Assumption: **A.PROPER_USER**	Mobile device users are not willfully negligent or hostile, and use the device within compliance of a reasonable Enterprise security policy.	
Objectives:	OE.PROPER_USER	Users of the mobile device are trained to securely use the mobile device and apply all guidance in a trusted manner.
Rationale:	OE.PROPER_USER supports this assumption by ensuring that the users are appropriately trained and are trusted to follow guidance.	

5 EXTENDED COMPONENTS DEFINITION

5.1 EXTENDED SECURITY FUNCTIONAL REQUIREMENTS

This section specifies the extended Security Functional Requirements (SFRs) used in this ST. The extended SFRs included in this ST originate from the Protection Profile for Mobile Device Fundamentals. The extended components are:

FAU_ALT_EXT.2
FCS_CKM_EXT.1
FCS_CKM_EXT.2
FCS_CKM_EXT.3
FCS_CKM_EXT.4
FCS_CKM_EXT.5
FCS_CKM_EXT.6
FCS_HTTPS_EXT.1
FCS_IV_EXT.1
FCS_RBG_EXT.1
FCS_SRV_EXT.1
FCS_STG_EXT.1(1)
FCS_STG_EXT.1(2)
FCS_STG_EXT.2
FCS_STG_EXT.3
FCS_STG_EXT.4
FCS_TLSC_EXT.1
FCS_TLSC_EXT.2
FDP_ACF_EXT.1
FDP_DAR_EXT.1
FDP_IFC_EXT.1
FDP_STG_EXT.1
FDP_UPC_EXT.1
FIA_AFL_EXT.1
FIA_BLT_EXT.1
FIA_BLT_EXT.2
FIA_ENR_EXT.2
FIA_PAE_EXT.1
FIA_PMG_EXT.1
FIA_TRT_EXT.1
FIA_UAU_EXT.1
FIA_UAU_EXT.2
FIA_UAU_EXT.3
FIA_X509_EXT.1
FIA_X509_EXT.2
FIA_X509_EXT.3
FMT_MOF_EXT.1
FMT_POL_EXT.2
FMT_SMF_EXT.1
FMT_SMF_EXT.2
FMT_SMF_EXT.3
FMT_UNR_EXT.1
FPT_AEX_EXT.1
FPT_AEX_EXT.2
FPT_AEX_EXT.3
FPT_AEX_EXT.4
FPT_BBD_EXT.1
FPT_KST_EXT.1
FPT_KST_EXT.2
FPT_KST_EXT.3
FPT_NOT_EXT.1
FPT_TST_EXT.1
FPT_TST_EXT.2
FPT_TUD_EXT.1
FPT_TUD_EXT.2
FTA_SSL_EXT.1
FTA_WSE_EXT.1
FTP_ITC_EXT.1

CLASS FAU: SECURITY AUDIT

One family has been added to the Security audit class. FAU_ALT_EXT deals with alerts and is modelled after the FAU_ARP Security audit automatic response family. FAU_ALT_EXT.2 is modelled after FAU_ARP.1 Security alarms.

5.1.1 FAU_ALT_EXT

Family Behaviour

This family defines the requirements for providing alerts from the agent to the server.

Component Levelling

FAU_ALT_EXT: Alerts
2

Figure 1 – FAU_ALT_EXT: Alerts Component Levelling

Management

There are no management activities foreseen.

Audit

The type of alert must be audited.

5.1.1.1 FAU_ALT_EXT.2 Agent Alerts

Hierarchical to: No other components.

Dependencies: No dependencies.

FAU_ALT_EXT.2.1 The MDM Agent shall provide an alert via the trusted channel to the MDM Server in the event of any of the following:

a. successful application of policies to a mobile device;

b. [selection: receiving, generating] periodic reachability events;

[selection:

c. change in enrollment state,

d. failure to install an application from the MAS Server,

e. failure to update an application from the MAS Server,

f. [assignment: *other events*], no other events].

FAU_ALT_EXT.2.2 The MDM Agent shall queue alerts if the trusted channel is not available.

5.2 CLASS FCS: CRYPTOGRAPHIC SUPPORT

Several SFRs and six new families have been added to the Cryptographic support class.

Several SFRs have been added to the Cryptographic key management family. FCS_CKM_EXT.1 to FCS_CKM_EXT.6 are part of the Cryptographic key management family and are modelled after FCS_CKM.1, FCS_CKM.2 and FCS_CKM.4.

FCS_HTTPS_EXT is a new family, HTTPS implementation, and is modelled after FCS_COP Cryptographic operation. FCS_HTTPS_EXT.1 is modelled after FCS_COP.1 Cryptographic operation.

FCS_IV_EXT is a new family, Initialization vector requirements, and is modelled after FCS_CKM Cryptographic key management. FCS_IV_EXT.1 is modelled after FCS_CKM.1 Cryptographic key generation.

FCS_RBG_EXT is a new family, Random bit generation, and is modelled after FCS_CKM Cryptographic key management. FCS_RBG_EXT.1 is modelled after FCS_CKM.1 Cryptographic key generation.

FCS_SRV_EXT is a new family, Cryptographic key services. FCS_SRV_EXT is modelled after FCS_CKM Cryptographic key management, and FCS_SRV_EXT.1 is modelled after FCS_CKM.1 Cryptographic key generation.

FCS_STG_EXT is a new family, Cryptographic key storage. FCS_STG_EXT is modelled after FCS_CKM Cryptographic key management and FAU_STG, Security audit event storage. FCS_STG_EXT.1, FCS_STG_EXT.2, FCS_STG_EXT.3 and FCS_STG_EXT.4 are all modelled after FCS_CKM.1 Cryptographic key generation.

FCS_TLSC_EXT is a new family, TLS protocol. FCS_TLSC_EXT is modelled after FCS_CKM Cryptographic key management. FCS_TLSC_EXT.1 and FCS_ TLSC _EXT.2 are modelled after FCS_CKM.1 Cryptographic key generation.

5.2.1 FCS_CKM

Family Behaviour

This family defines the requirements for key management.

Component Levelling

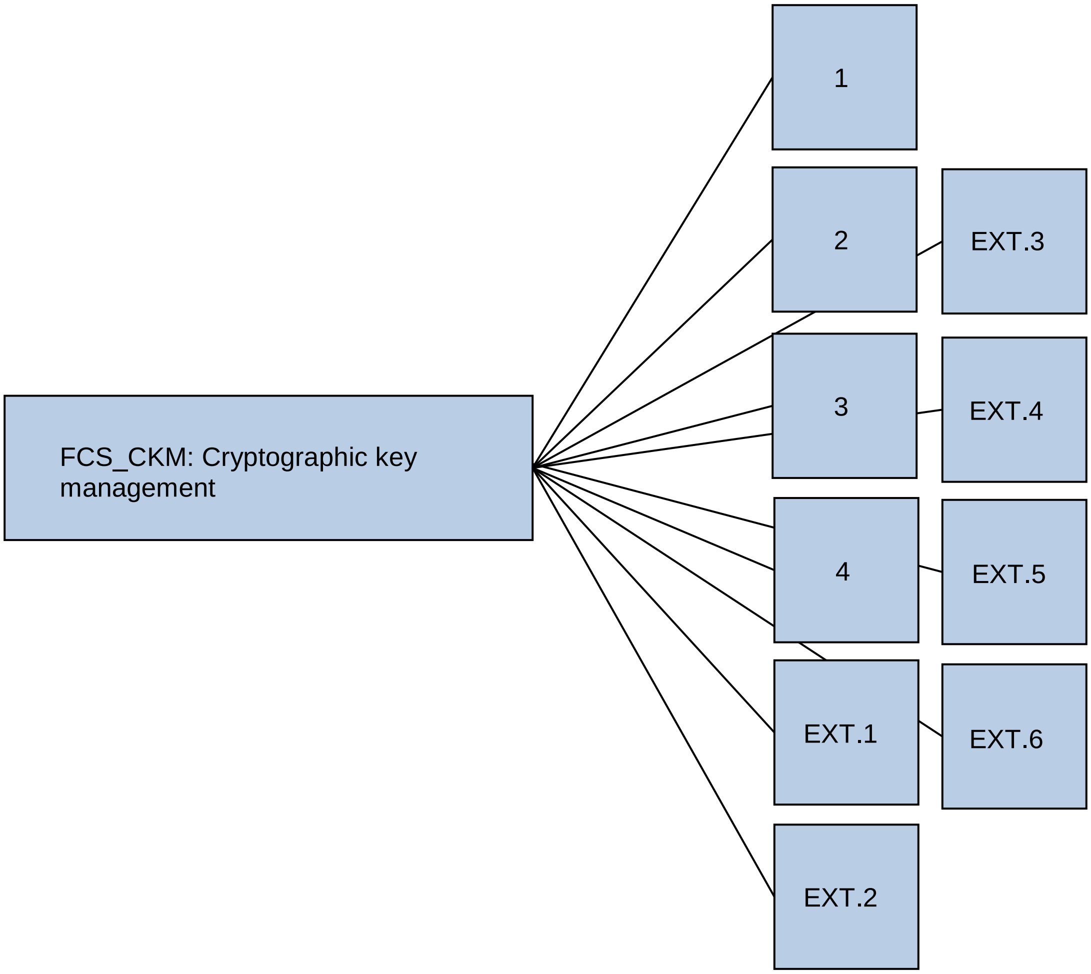

Figure 2 – FCS_CKM: Cryptographic Key Management Component Levelling

Management

There are no management activities foreseen.

Audit FCS_CKM_EXT.1

The following actions may be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Basic: Generation of a REK.

Audit FCS_CKM_EXT.5

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: Success or failure of the wipe.

5.2.1.1 FCS_CKM_EXT.1 Cryptographic key support

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_CKM_EXT.1.1 The TSF shall support a [selection: hardware-isolated, hardware-protected] REK with a [selection: symmetric/asymmetric] key of strength [selection: 112 bits, 128 bits, 192 bits, 256 bits].

FCS_CKM_EXT.1.2 System software on the TSF shall be able only to request [selection: encryption/decryption, NIST SP 800-108 key derivation] by the key and shall not be able to read, import, or export a REK.

FCS_CKM_EXT.1.3 A REK shall be generated by a RBG in accordance with FCS_RBG_EXT.1.

FCS_CKM_EXT.1.4 A REK shall not be able to be read from or exported from the hardware.

5.2.1.2 FCS_CKM_EXT.2 Cryptographic key random generation

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_CKM_EXT.2.1 All DEKs shall be randomly generated with entropy corresponding to the security strength of AES key sizes of [selection: 128, 256] bits.

5.2.1.3 FCS_CKM_EXT.3 Cryptographic key generation

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_CKM_EXT.3.1 The TSF shall use [selection: asymmetric KEKs of [assignment: *security strength greater than or equal to 112 bits*] security strength,[selection: 128-bit, 256-bit] symmetric KEKs] corresponding to at least the security strength of the keys encrypted by the KEK.

FCS_CKM_EXT.3.2 The TSF shall generate all KEKs using one of the following methods:

a) derive the KEK from a Password Authentication Factor using PBKDF and

[selection:

b) generate the KEK using an RBG that meets this profile (as specified in FCS_RBG_EXT.1)

c) generate the KEK using a key generation scheme that meets this profile (as specified in FCS_RBG_EXT.1).

d) Combine the KEK from other KEKs in a way that preserves the effective entropy of each factor by [selection: using an XOR operation, concatenating the keys and use a KDF (as described in SP 800-108), encrypting one key with another]].

5.2.1.4 FCS_CKM_EXT.4 Cryptographic key destruction

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_CKM_EXT.4.1 The TSF shall destroy cryptographic keys in accordance with the specified cryptographic key destruction methods:

- by clearing the KEK encrypting the target key,
- in accordance with the following rules:
    - o For volatile memory, the destruction shall be executed by a single direct overwrite [selection: consisting of a pseudo-random pattern using the TSF's RBG, consisting of zeroes].
    - o For non-volatile EEPROM, the destruction shall be executed by a single direct overwrite consisting of a pseudo random pattern using the TSF's RBG (as specified in FCS_RBG_EXT.1), followed by a read-verify.
    - o 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].
    - o 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].
    - o For non-volatile memory other than EEPROM and flash, the destruction shall be executed by overwriting three or more times with a random pattern that is changed before each write.

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

5.2.1.5 FCS_CKM_EXT.5 TSF wipe

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_CKM_EXT.5.1 The TSF shall wipe all protected data by [selection:

- Cryptographically erasing the encrypted DEKs and/or the KEKs in non-volatile memory by following the requirements in FCS_CKM_EXT.4.1;
- Overwriting all protected data according to the following rules:
    - o For EEPROM, the destruction shall be executed by a single direct overwrite consisting of a pseudo random pattern using the TSF's RBG (as specified in FCS_RBG_EXT.1, followed a read-verify.
    - o For flash memory the destruction shall be executed [selection: by a single direct overwrite consisting of zeros followed by a read-verify, by a block erase followed by a read-verify].
    - o For non-volatile memory other than EEPROM and flash, the destruction shall be executed by overwriting three or more times with a random pattern that is changed before each write.]

FCS_CKM_EXT.5.2 The TSF shall perform a power cycle on conclusion of the wipe procedure.

5.2.1.6 FCS_CKM_EXT.6 Salt generation

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_CKM_EXT.6.1 The TSF shall generate all salts using a RBG that meets FCS_RBG_EXT.1.

5.2.2 FCS_HTTPS_EXT

Family Behaviour

This family defines the requirements for HTTPS Implementation.

Component Levelling

FCS_HTTPS_EXT: HTTPS Implementation
1

Figure 3 – FCS_HTTPS_EXT: HTTPS Implementation Component Levelling

Management

There are no management activities foreseen.

Audit

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: Failure of the certificate validity check.

5.2.2.1 FCS_HTTPS_EXT.1 Cryptographic operation

Hierarchical to: No other components.

Dependencies: No dependencies.

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

FCS_ HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS (FCS_TLSC_EXT.2).

FCS_ HTTPS_EXT.1.3 The TSF shall notify the application and [selection: not establish the connection, request application authorization to establish the connection, no other action] if the peer certificate is deemed invalid.

5.2.3 FCS_IV_EXT

Family Behaviour

This family defines the requirements for Initialization vector requirements.

Component Levelling

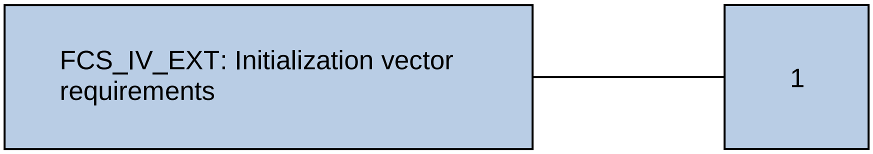

Figure 4 – FCS_IV_EXT: Initialization Vector Requirements Component Levelling

Management

There are no management activities foreseen.

Audit

There are no auditable events foreseen.

5.2.3.1 FCS_IV_EXT.1 Initialization vector generation

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_IV_EXT.1.1 The TSF shall generate IVs in accordance with Table 9: References and IV Requirements for NIST-approved Cipher Modes.

Cipher Mode	Reference	IV Requirements
Electronic Codebook (ECB)	SP 800-38A	No IV
Counter (CTR)	SP 800-38A	"Initial Counter" shall be non-repeating. No counter value shall be repeated across multiple messages with the same secret key.
Cipher Block Chaining (CBC)	SP 800-38A	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)	SP 800-38A	IVs shall be non-repeating and shall not be generated by invoking the cipher on another IV.
Cipher Feedback (CFB)	SP 800-38A	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)	SP 800-38E	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)	SP 800-38B	No IV
Key Wrap and Key Wrap	SP 800-38F	No IV

Cipher Mode	Reference	IV Requirements
with Padding		
Counter with CBC-Message Authentication Code (CCM)	SP 800-38C	No IV. Nonces shall be non-repeating.
Galois Counter Mode (GCM)	SP 800-38D	IV shall be non-repeating. The number of invocations of GCM shall not exceed 2^32 for a given secret key unless an implementation only uses 96-bit IVs (default length).

Table 9 – References and IV Requirements for NIST-approved Cipher Modes

5.2.4 FCS_RBG_EXT

Family Behaviour

This family defines the requirements for random bit generation.

Component Levelling

Figure 5 – FCS_RBG_EXT: Random Bit Generation Component Levelling

Management

There are no management activities foreseen.

Audit

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: Failure of the randomization process.

5.2.4.1 FCS_RBG_EXT.1 Cryptographic operation (Random bit generation)

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RBG_EXT.1.1 The TSF shall perform all deterministic random bit generation services in accordance with [selection, choose one of: NIST Special Publication 800-90A using [selection: Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES); FIPS Pub 140-2 Annex C: X9.31 Appendix 2.4 using AES].

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, TSF-hardware-based noise source] 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.

FCS_RBG_EXT.1.3 The TSF shall be capable of providing output of the RBG to applications running on the TSF that request random bits.

5.2.5 FCS_SRV_EXT

Family Behaviour

This family defines the requirements for cryptographic algorithm services.

Component Levelling

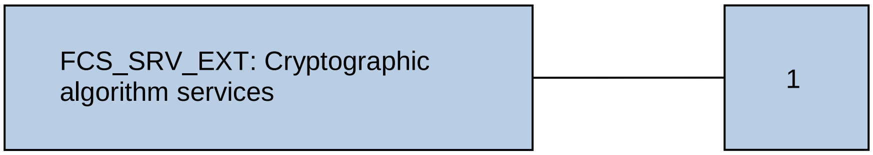

Figure 6 – FCS_SRV_EXT: Cryptographic Algorithm Services Component Levelling

Management

There are no management activities foreseen.

Audit

There are no auditable events foreseen.

5.2.5.1 FCS_SRV_EXT.1 Cryptographic algorithm services

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_SRV_EXT.1.1 The TSF shall provide a mechanism for applications to request the TSF to perform the following cryptographic operations:

- All mandatory and [selection: selected algorithms, selected algorithms with the exception of ECC over curve 25519-based algorithms] in FCS_CKM.2(1)
- The following algorithms in FCS_COP.1(1): AES-CBC, [selection: AES Key Wrap, AES Key Wrap with Padding, AES-GCM, AES-CCM, no other modes]
- All mandatory and selected algorithms in FCS_COP.1(3)
- All mandatory and selected algorithms in FCS_COP.1(2)
- All mandatory and selected algorithms in FCS_COP.1(4)

[selection:

- All mandatory and [selection: selected algorithms, selected algorithms with the exception of ECC over curve 25519-based algorithms] in FCS_CKM.1(1),
- The selected algorithms in FCS_COP.1(5),
- No other cryptographic operations].

5.2.6 FCS_STG_EXT

Family Behaviour

This family defines the requirements for cryptographic key storage.

Component Levelling

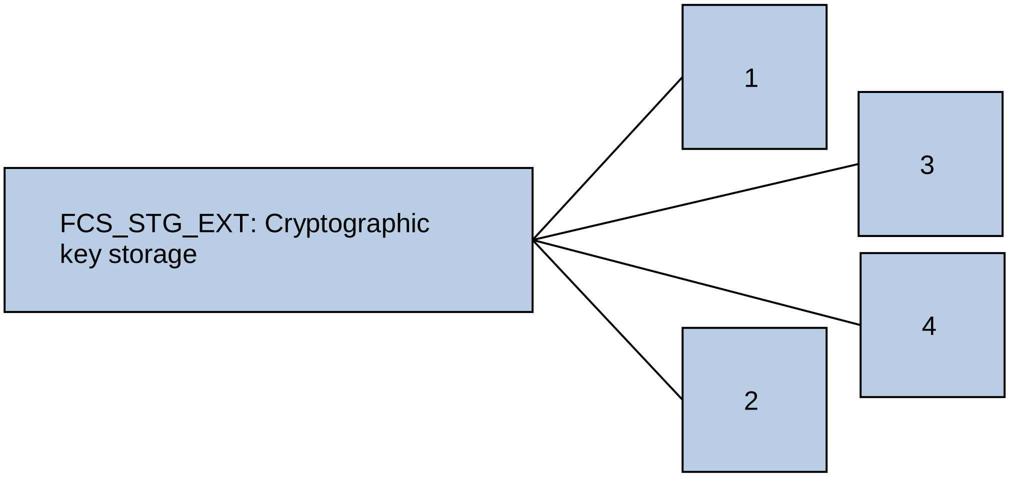

Figure 7 – FCS_STG_EXT: Cryptographic Key Storage Component Levelling

Management

The following actions could be considered for the management functions in FMT:

a. In FCS_STG_EXT.1.2, if the ST Author selects only user, the ST Author shall select function 11 in FMT_MOF_EXT.1.1;

b. In FCS_STG_EXT.1.3, if the ST Author selects only user, the ST Author shall select function 12 in FMT_MOF_EXT.1.1;

c. In FCS_STG_EXT.1.4, if the ST Author selects user or administrator, the ST Author must also select function 34 in FMT_SMF_EXT.1.1. If the ST Author selects only user, the ST Author shall select function 34 in FMT_MOF_EXT.1.1; and

d. In FCS_STG_EXT.1.5, if the ST Author selects user or administrator, the ST Author must also select function 35 in FMT_SMF_EXT.1.1. If the ST Author selects only user, the ST Author shall select function 35 in FMT_MOF_EXT.1.1.

Audit FCS_STG_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: Import or destruction of key.

Audit FCS_STG_EXT.3

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: Failure to verify integrity of stored key.

5.2.6.1 FCS_STG_EXT.1 Cryptographic key storage

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_STG_EXT.1.1 The TSF shall provide [selection: hardware, hardware-isolated, software-based] secure key storage for asymmetric private keys and [selection: symmetric keys, persistent secrets, no other keys].

FCS_STG_EXT.1.2 The TSF shall be capable of importing keys/secrets into the secure key storage upon request of [selection: the user, the administrator] and [selection: applications running on the TSF, no other subjects].

FCS_STG_EXT.1.3 The TSF shall be capable of destroying keys/secrets in the secure key storage upon request of [selection: the user, the administrator].

FCS_STG_EXT.1.4 The TSF shall have the capability to allow only the application that imported the key/secret the use of the key/secret. Exceptions may only be explicitly authorized by [selection: the user, the administrator, a common application developer].

FCS_STG_EXT.1.5 The TSF shall allow only the application that imported the key/secret to request that the key/secret be destroyed. Exceptions may only be explicitly authorized by [selection: the user, the administrator, a common application developer].

5.2.6.2 FCS_STG_EXT.2 Encrypted cryptographic key storage

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_STG_EXT.2.1 The TSF shall encrypt all DEKs and KEKs and [selection: long-term trusted channel key material, all software-based key storage, no other keys] by KEKs that are [selection:

1) Protected by the REK with [selection:

   a. encryption by a REK,

   b. encryption by a KEK chaining to a REK],

2) Protected by the REK and the password with [selection:

   a. encryption by a REK and the password-derived KEK,

   b. encryption by a KEK chaining to a REK and the password-derived KEK]

].

FCS_STG_EXT.2.2 DEKs and KEKs and [selection: long-term trusted channel key material, all software-based key storage, no other keys] shall be encrypted using one of the following methods [selection: using a SP800-56B key establishment scheme, using AES in the [selection: Key Wrap (KW) mode, Key Wrap with Padding (KWP) mode, GCM, CCM, CBC mode]].

5.2.6.3 FCS_STG_EXT.3 Integrity of encrypted key storage

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_STG_EXT.3.1 The TSF shall protect the integrity of any encrypted DEKs and KEKs and [selection: long-term trusted channel key material, all software-based key storage, no other keys] by [selection:

- [selection: GCM, CCM, Key Wrap, Key Wrap with Padding] cipher mode for encryption according to FCS_STG_EXT.2;
- a hash (FCS_COP.1(2)) of the stored key that is encrypted by a key protected by FCS_STG_EXT.2;
- a keyed hash (FCS_COP.1(4)) using a key protected by a key protected by FCS_STG_EXT.2;
- a digital signature of the stored key using an asymmetric key protected according to FCS_STG_EXT.2].

FCS_STG_EXT.3.2 The TSF shall verify the integrity of the [selection: hash, digital signature, MAC] of the stored key prior to use of the key.

5.2.6.4 FCS_STG_EXT.4 Cryptographic key storage

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_STG_EXT.4.1 The MDM Agent shall store persistent secrets and private keys when not in use in platform-provided key storage.

5.2.7 FCS_TLSC_EXT

Family Behaviour

This family defines the requirements for use of the TLS protocol.

Component Levelling

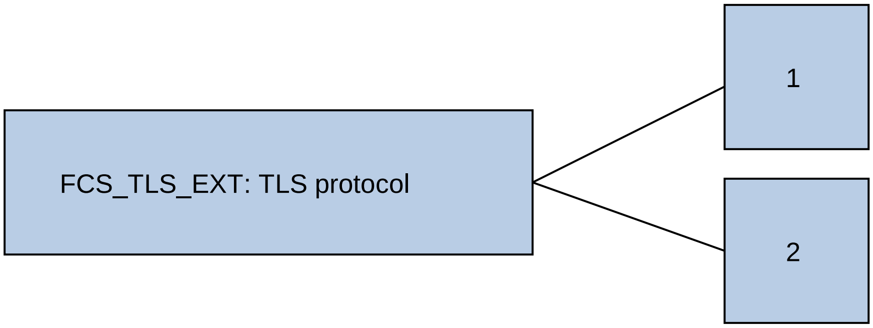

Figure 8 – FCS_TLS_EXT: TLS Implementation Component Levelling

Management

The following actions could be considered for the management functions in FMT:

a. In FCS_TLSC_EXT.1.2, the CA or FQDN is specified according to FMT_SMF_EXT.1 function 7.a.

Audit FCS_TLSC_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: Failure to establish an EAP-TLS session.

b. Minimal: Establishment/termination of an EAP TLS session.

Audit FCS_TLSC_EXT.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: Failure to establish a TLS session.

b. Minimal: Failure to verify presented identifier.

c. Minimal: Establishment/termination of a TLS session.

5.2.7.1 FCS_TLSC_EXT.1 EAP TLS protocol

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FCS_TLSC_EXT.1.1 The TSF shall implement TLS 1.0 and [selection: TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246), no other TLS version] supporting the following ciphersuites: [

- Mandatory Ciphersuites:
    - TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246
- [selection: Optional Ciphersuites:
    - TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246
    - TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246
    - TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246
    - TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
    - TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
    - TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
    - TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
    - TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
    - TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246
    - TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256 as defined in RFC 5246
    - TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246

- ○ TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- ○ TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- ○ TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- ○ TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- ○ no other ciphersuite]].

FCS_TLSC_EXT.1.2 The TSF shall verify that the server certificate presented for EAP-TLS [selection: chains to one of the specified CAs, contains the specified FQDN of the acceptable authentication server certificate.].

FCS_TLSC_EXT.1.3 The TSF shall not establish a trusted channel if the peer certificate is invalid.

FCS_TLSC_EXT.1.4 The TSF shall support mutual authentication using X.509v3 certificates.

FCS_TLSC_EXT.1.5 The TSF shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [selection: secp256r1, secp384r1, secp521r1] and no other curves.

FCS_TLSC_EXT.1.6 The TSF shall present the signature_algorithms extension in the Client Hello with the supported_signature_algorithms value containing the following hash algorithms: [selection: SHA256, SHA384, SHA512] and no other hash algorithms.

5.2.7.2 FCS_TLSC_EXT.2 TLS protocol

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_TLSC_EXT.2.1 The TSF shall implement TLS 1.2 (RFC 5246) supporting the following ciphersuites: [

- Mandatory Ciphersuites:
  - ○ TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246
- [Optional Ciphersuites:
  - ○ TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246
  - ○ TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246
  - ○ TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246
  - ○ TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
  - ○ TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
  - ○ TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492

- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- no other ciphersuite]].

FCS_TLSC_EXT.2.2 The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.2.3 The TSF shall not establish a trusted channel if the peer certificate is invalid.

FCS_TLSC_EXT.2.4 The TSF shall support mutual authentication using X.509v3 certificates.

FCS_TLSC_EXT.2.5 The TSF shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [selection: secp256r1, secp384r1, secp521r1] and no other curves.

5.3 CLASS FDP: USER DATA PROTECTION

Several SFRs and three new families have been added to the User data protection class.

FDP_ACF_EXT.1 is part of the Access control functions family and is modelled after FDP_ACF.1 Security attribute based access control and FDP_ACC.1 Subset access control.

FDP_DAR_EXT is a new family, Data-at-rest protection, and is modelled after FDP_ACF Access control functions. FDP_DAR_EXT.1 is modelled after FDP_ACF.1 Security attribute based access control.

FDP_IFC_EXT.1 is part of the Information flow control policy family and is modelled after FDP_IFC.1 Subset information flow control and FDP_IFF.1 Simple security attributes.

FDP_STG_EXT is a new family, User data storage, and is modelled after FAU_STG Security audit event storage. FDP_STG_EXT.1 is modelled after FAU_STG.1 Protected audit trail storage.

FDP_UPC_EXT is a new family, Inter-TSF user data transfer protection, and is modelled after FDP_ITT Internal TOE transfer. FDP_UPC_EXT.1 is modelled after FDP_ITT.1 Basic internal transfer protection.

5.3.1 FDP_ACF

Family Behaviour

This family defines the requirements for Access control functions.

Component Levelling

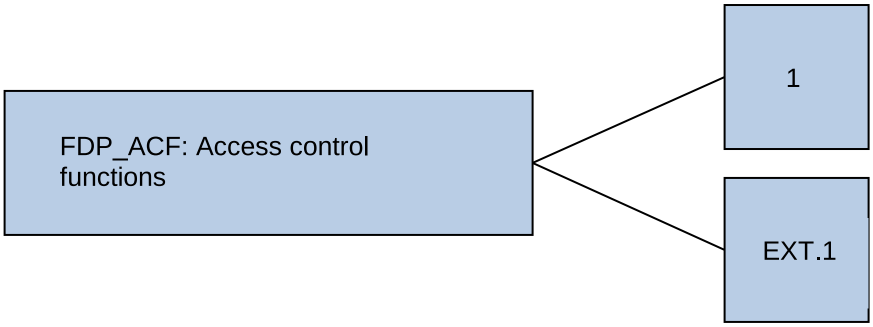

Figure 9 – FDP_ACF: Access Control Functions Component Levelling

Management

There are no management activities foreseen.

Audit

There are no auditable events foreseen.

5.3.1.1 FDP_ACF_EXT.1 Security access control

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_ACF_EXT.1.1 The TSF shall provide a mechanism to restrict the system services that are accessible to an application.

FDP_ACF_EXT.1.2 The TSF shall provide an access control policy that prevents [selection: application processes, groups of application processes] from accessing [selection: all, private] data stored by other [selection: application processes, groups of application processes]. Exceptions may only be explicitly authorized for such sharing by [selection: the user, the administrator, a common application developer].

5.3.2 FDP_DAR_EXT

Family Behaviour

This family defines the requirements for data-at-rest protection.

Component Levelling

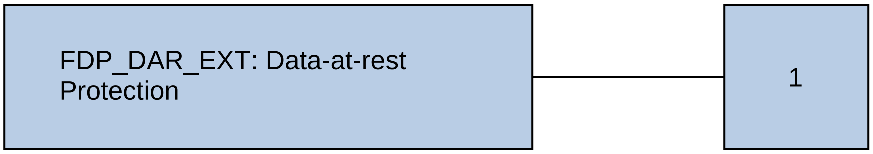

Figure 10 – FDP_DAR_EXT: Data-at-rest Protection Component Levelling

Management

There are no management activities foreseen.

Audit FDP_DAR_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: Failure to encrypt/decrypt data.

5.3.2.1 FDP_DAR_EXT.1 Data-at-rest protection

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_DAR_EXT.1.1 Encryption shall cover all protected data.

FDP_DAR_EXT.1.2 Encryption shall be performed using DEKs with AES in the [selection: XTS, CBC, GCM] mode with key size [selection: 128,256] bits.

5.3.3 FDP_IFC

Family Behaviour

This family defines the requirements for Information flow control policy.

Component Levelling

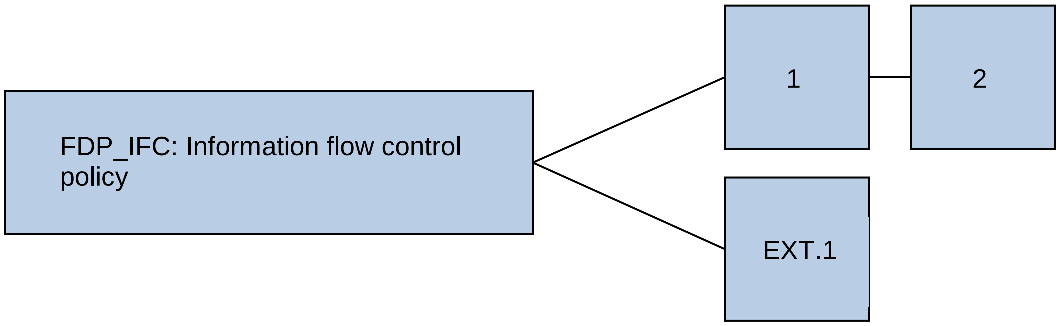

Figure 11 – FDP_IFC: Information Flow Control Policy Component Levelling

Management

There are no management activities foreseen.

Audit

There are no auditable events foreseen.

5.3.3.1 FDP_IFC_EXT.1 Subset information flow control

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_IFC_EXT.1.1 The TSF shall [selection: provide an interface to VPN clients to enable all IP traffic (other than IP traffic required to establish the VPN connection) to flow through the IPsec VPN client; enable all IP traffic (other than IP traffic required to establish the VPN connection) to flow through the IPsec VPN client].

5.3.4 FDP_STG_EXT

Family Behaviour

This family defines the requirements for User data storage.

Component Levelling

FDP_STG_EXT: User data storage
1

Figure 12 – FDP_STG_EXT: User Data Storage Component Levelling

Management

The following actions could be considered for the management functions in FMT:

a. The permissions related to management functionality dictating how certificates are loaded into the store, and how the store is protected from unauthorized access (for example, unix permissions) must be established in FMT_SMF_EXT.1 and FMT_MOF_EXT.1.

Audit

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: Addition or removal of certificate from Trust Anchor Database.

5.3.4.1 FDP_STG_EXT.1 User data storage

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_STG_EXT.1.1 The TSF shall provide protected storage for the Trust Anchor Database.

5.3.5 FDP_UPC_EXT

Family Behaviour

This family defines the requirements for Inter-TSF user data transfer protection.

Component Levelling

Figure 13 – FDP_UPC_EXT: Inter-TSF User Data Transfer Protection Component Levelling

Management

There are no management activities foreseen.

Audit

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: Application initiation of trusted channel.

5.3.5.1 FDP_UPC_EXT.1 Inter-TSF user data transfer protection

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_UPC_EXT.1.1 The TSF provide a means for non-TSF applications executing on the TOE to use TLS, HTTPS, Bluetooth BR/EDR, and [selection: DTLS, Bluetooth LE, no other protocol] to provide a protected communication channel between the non-TSF application and another IT product 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.

FDP_UPC_EXT.2.1 The TSF shall permit the non-TSF applications to initiate communication via the trusted channel.

5.4 CLASS FIA: IDENTIFICATION AND AUTHENTICATION

Several SFRs and five new families have been added to the Identification and authentication class.

FIA_AFL_EXT.1 is part of the Authentication failures family and is modelled after FIA_AFL.1 Authentication failure handling.

FIA_BLT_EXT is a new family, Bluetooth authentication, and is modelled after FIA_UAU User authentication. FIA_BLT_EXT.1 is modelled after FIA_UAU.2 User authentication before any action, and FIA_BLT_EXT.2 is modelled after FIA_UAU.1 Timing of authentication.

FIA_ENR_EXT is a new family, Enrollment of Mobile Device into Management, and is modelled after FIA_UAU User authentication. FIA_ENR_EXT.2 is modelled after FIA_UAU.2 User authentication before any action.

FIA_PAE_EXT is a new family, Port Access Entity (PAE) authentication, and is modelled after FIA_UAU User authentication. FIA_PAE_EXT.1 is modelled after FIA_UAU.2 User authentication before any action.

FIA_PMG_EXT is a new family, Password management, and is modelled after FIA_SOS Specification of secrets. FIA_PMG_EXT.1 is modelled after FIA_SOS.1 Verification of secrets.

FIA_TRT_EXT is a new family, Authentication throttling, and is modelled after FIA_AFL Authentication failures. FIA_TRT_EXT.1 is modelled after FIA_AFL.1 Authentication failure handling.

FIA_UAU_EXT.1 Authentication for cryptographic function, FIA_UAU_EXT.2 and FIA_UAU_EXT.3 Re-authentication are part of the User authentication family. FIA_UAU_EXT.1 is modelled after FIA_UAU.2 User authentication before any action. FIA_UAU_EXT.2 Timing of authentication is modelled after FIA_UAU.1 Timing of authentication. FIA_UAU_EXT.3 is modelled after FIA_UAU.6 Re-authenticating.

FIA_X509_EXT is a new family, Certificate authentication, and is modelled after FIA_UAU User authentication. FIA_X509_EXT.1 Validation of certificates, FIA_X509_EXT.2 X509 certificate authentication and FIA_X509_EXT.3 Request validation of certificates are all modelled after FIA_UAU.1 User authentication before any action.

5.4.1 FIA_AFL

Family Behaviour

This family defines the requirements for Authentication failure handling.

Component Levelling

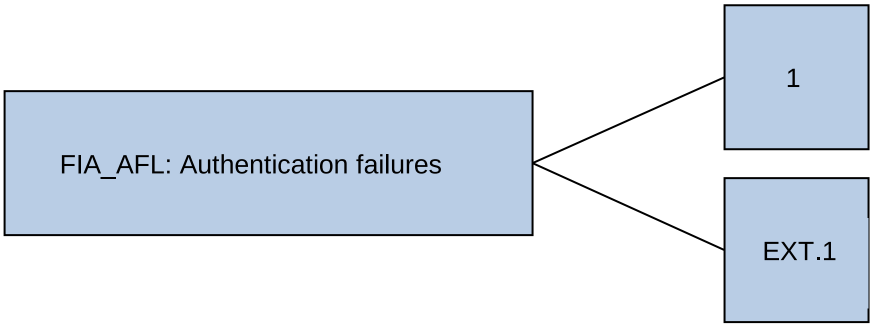

Figure 14 – FIA_AFL: Authentication Failures Component Levelling

Management

The following actions could be considered for the management functions in FMT:

a. The positive integer is configured according to FMT_SMF_EXT.1.1 function 2.c.

Audit

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: Excess of authentication failure limit.

5.4.1.1 FIA_AFL_EXT.1 Authentication failure handling

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_AFL_EXT.1.1 The TSF shall detect when a configurable positive integer within [assignment: range of acceptable values] of unsuccessful authentication attempts occur related to last successful authentication by that user.

FIA_AFL_EXT.1.2 When the defined number of unsuccessful authentication attempts has been surpassed, the TSF shall perform wipe of all protected data.

FIA_AFL_EXT.1.3 The TSF shall maintain the number of unsuccessful authentication attempts that have occurred upon power off.

5.4.2 FIA_BLT_EXT

Family Behaviour

This family defines the requirements for Bluetooth authentication.

Component Levelling

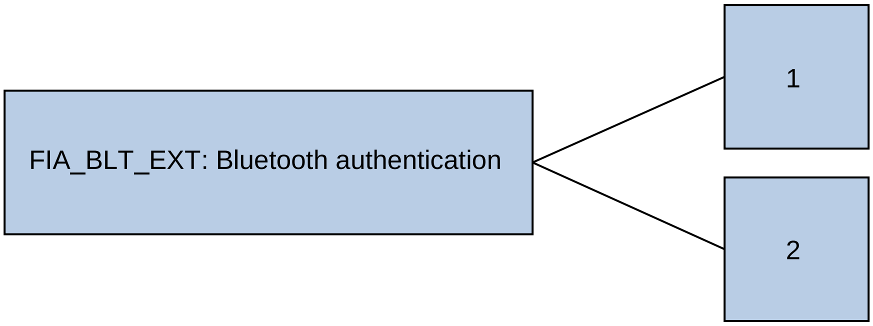

Figure 15 – FIA_BLT_EXT: Bluetooth Authentication Component Levelling

Management

There are no management activities foreseen.

Audit FIA_BLT_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: User authorization of Bluetooth device.

b. Minimal: User authorization for local Bluetooth service.

Audit FIA_BLT_EXT.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: Initiation of Bluetooth connection.

b. Minimal: Failure of Bluetooth connection.

5.4.2.1 FIA_BLT_EXT.1 Bluetooth user authorization

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FIA_BLT_EXT.1.1 The TSF shall require explicit user authorization before pairing with a remote Bluetooth device.

FIA_BLT_EXT.1.2 The TSF shall require explicit user authorization before granting trusted remote devices access to services associated with the following Bluetooth profiles: [assignment: list of Bluetooth profiles], and shall require explicit user authorization before granting untrusted remote devices access to services associated with the following Bluetooth profiles: [assignment: list of Bluetooth profiles].

5.4.2.2 FIA_BLT_EXT.2 Bluetooth mutual authentication

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FIA_BLT_EXT.2.1 The TSF shall require Bluetooth mutual authentication between devices prior to any data transfer over the Bluetooth link.

5.4.3 FIA_ENR_EXT

Family Behaviour

This family defines the requirements for the enrollment of mobile devices into management.

Component Levelling

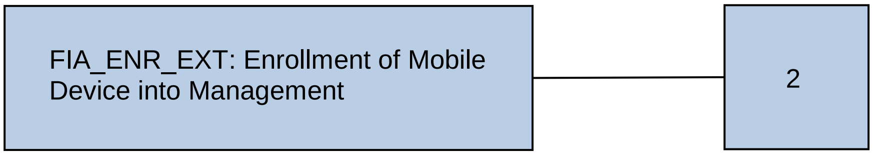

Figure 16 – FIA_ENR_EXT: Enrollment of Mobile Device into Management Component Levelling

Management

There are no management activities foreseen.

Audit

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: Enrollment in management.

5.4.3.1 FIA_ENR_EXT.2 Enrollment of Mobile Device into Management

Hierarchical to: No other components.

Dependencies: No dependencies.**FIA_ENR_EXT.2.1** The MDM Agent shall record the reference identifier of the MDM Server during the enrollment process.

5.4.4 FIA_PAE_EXT

Family Behaviour

This family defines the requirements for Port Access Entity (PAE) authentication.

Component Levelling

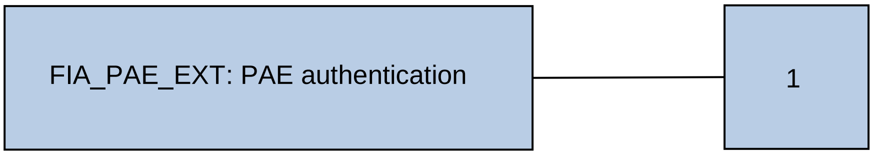

Figure 17 – FIA_PAE_EXT: PAE Authentication Component Levelling

Management

There are no management activities foreseen.

Audit

There are no auditable events foreseen.

5.4.4.1 FIA_PAE_EXT.1 PAE authentication

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_PAE_EXT.1.1 The TSF shall conform to IEEE Standard 802.1X for a Port Access Entity (PAE) in the "Supplicant" role.

5.4.5 FIA_PMG_EXT

Family Behaviour

This family defines the requirements for Password management.

Component Levelling

FIA_PMG_EXT: Password management
1

Figure 18 – FIA_PMG_EXT: Password Management Component Levelling

Management

There are no management activities foreseen.

Audit

There are no auditable events foreseen.

5.4.5.1 FIA_PMG_EXT.1 Password management

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_PMG_EXT.1.1 The TSF shall support the following for the Password Authentication Factor:

1. Passwords shall be able to be composed of any combination of [selection: upper and lower case letters, [assignment: a character set of at least 52 characters]], numbers, and special characters: [selection: "!", "@", "#", "$", "%", "^", "&", "*", "(", ")", assignment: other characters];

2. Password length up to [assignment: an integer greater than or equal to 14] characters shall be supported.

5.4.6 FIA_TRT_EXT

Family Behaviour

This family defines the requirements for Authentication throttling.

Component Levelling

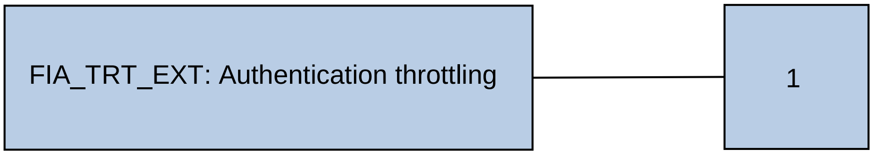

Figure 19 – FIA_TRT_EXT: Authentication Throttling Component Levelling

Management

There are no management activities foreseen.

Audit

There are no auditable events foreseen.

5.4.6.1 FIA_TRT_EXT.1 Authentication throttling

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_TRT_EXT.1.1 The TSF shall limit automated user authentication attempts by [selection: preventing authentication via an external port, enforcing a delay between incorrect authentication attempts]. The minimum delay shall be such that no more than 10 attempts can be attempted per 500 milliseconds.

5.4.7 FIA_UAU

Family Behaviour

This family defines the requirements for User authentication.

Component Levelling

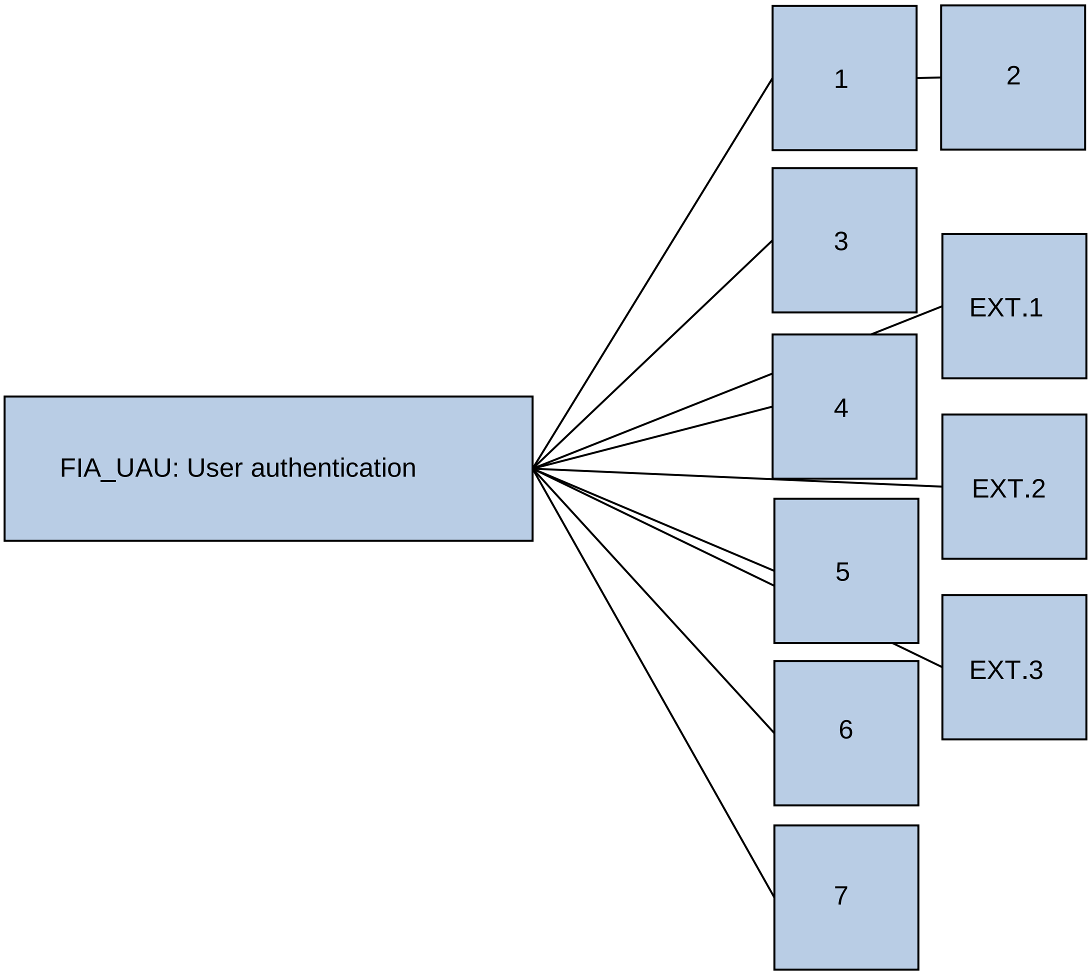

Figure 20 – FIA_UAU: User Authentication Component Levelling

Management

There are no management activities foreseen.

Audit FIA_UAU_EXT.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: Action performed before authentication.

Audit FIA_UAU_EXT.3

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: User changes Password Authentication Factor.

5.4.7.1 FIA_UAU_EXT.1 Authentication for cryptographic operation

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU_EXT.1.1 The TSF shall require the user to present the Password Authentication Factor prior to decryption of protected data and encrypted DEKs, KEKs and [selection: long- term trusted channel key material, all software-based key storage, no other keys] at startup.

5.4.7.2 FIA_UAU_EXT.2 Timing of authentication

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU_EXT.2.1 The TSF shall allow [selection: [assignment: *list of actions*], no actions] on behalf of the user to be performed before the user is authenticated.

FIA_UAU_EXT.2.2 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

5.4.7.3 FIA_UAU_EXT.3 Re-authentication

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU_EXT.3.1 The TSF shall require the user to enter the correct Password Authentication Factor when the user changes the Password Authentication Factor, and following TSF- and user-initiated locking in order to transition to the unlocked state, and [selection:[assignment: *other conditions*], no other conditions].

5.4.8 FIA_X509_EXT

Family Behaviour

This family defines the requirements for Certificate authentication.

Component Levelling

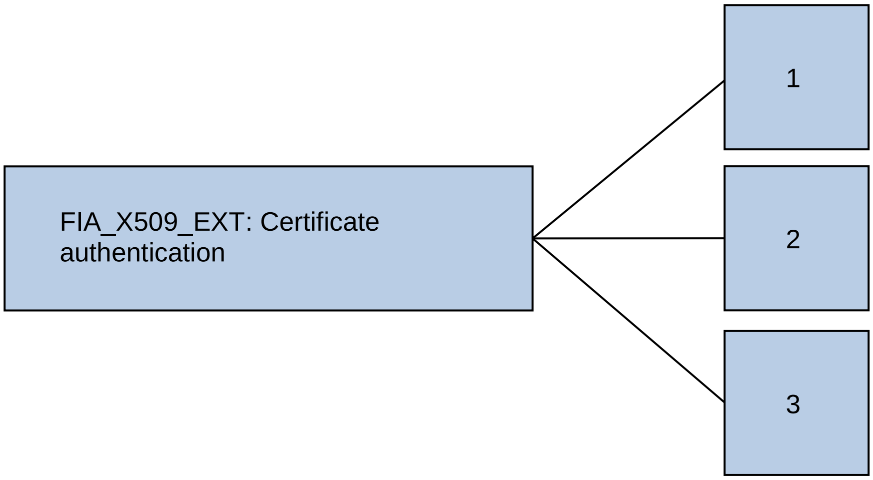

Figure 21 – FIA_X509_EXT: Certificate Authentication Component Levelling

Management FIA_X509_EXT.2

The following actions could be considered for the management functions in FMT:

a. If the administrator-configured or user-configured option is selected by the ST Author, the ST Author must also select function 30 in FMT_SMF_EXT.1.

Audit FIA_X509_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: Failure to validate X.509v3 certificate.

Audit FIA_X509_EXT.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: Failure to establish connection to determine revocation status.

5.4.8.1 FIA_X509_EXT.1 Validation of certificates

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FIA_X509_EXT.1.1 The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a certificate in the Trust Anchor Database.
- 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 5759].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - o Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
  - o 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.
  - o (Conditional) 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.

FIA_X509_EXT.1.2 The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

5.4.8.2 FIA_X509_EXT.2 X509 certificate authentication

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_X509_EXT.2.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for EAP-TLS exchanges, and [selection: IPsec, TLS, HTTPS, DTLS]], and [selection: code signing for system software updates, code signing for mobile applications, code signing for integrity verification, [assignment: *other uses*], no additional uses].

FIA_X509_EXT.2.2 When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [selection: allow the administrator to choose whether to accept the certificate in these cases, allow the user to choose whether to accept the certificate in these cases, accept the certificate, not accept the certificate].

5.4.8.3 FIA_X509_EXT.3 Request validation of certificates

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_X509_EXT.3.1 The TSF shall provide a certificate validation service to applications.

FIA_X509_EXT.3.2 The TSF shall respond to the requesting application with the success or failure of the validation.

5.5 CLASS FMT: SECURITY MANAGEMENT

Two new families and several SFRs and have been added to the Security management class.

FMT_MOF_EXT.1 Management of security functions behavior is modelled after FMT_MOF.1 Management of security functions behaviour. FMT_SMF_EXT.1 Specification of management functions, FMT_SMF_EXT.2 Specification of remediation actions, and FMT_SMF_EXT.3 Specification of management functions, are modelled after FMT_SMF.1 Specification of management functions. FMT_UNR_EXT.1 User Unenrollment Prevention is loosely modelled after FMT_MOF.1 Management of security functions behaviour.

FMT_POL_EXT is a new family, Trusted policy update, and is modelled after FMT_MSA Management of security attributes. FMT_POL_EXT.2 is modelled after FMT_MSA.2 Secure security attributes.

FMT_UNR_EXT is a new family, User unenrollment prevention, and is modelled after FMT_MSA Management of security attributes. FMT_UNR_EXT.1 is modelled after FPT_FLS.1 Failure with preservation of secure state.

5.5.1 FMT_MOF

Family Behaviour

This family defines the requirements for Management functions in the TSF.

Component Levelling

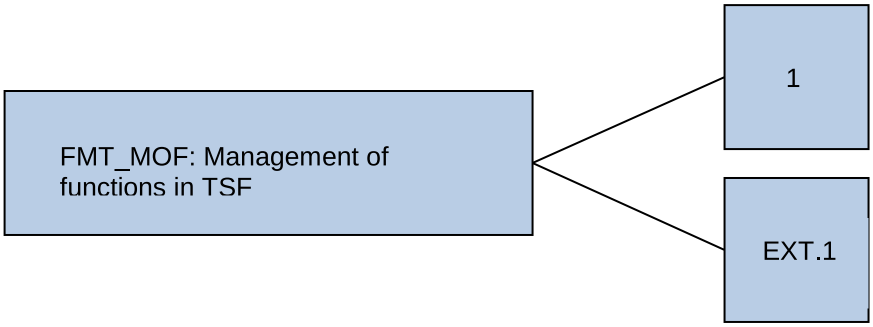

Figure 22 – FMT_MOF: Management of Functions in TSF Component Levelling

Management

There are no management activities foreseen.

Audit

There are no auditable events foreseen.

5.5.1.1 FMT_MOF_EXT.1 Management of security functions behavior

Hierarchical to: No other components.

Dependencies: No dependencies.

FMT_MOF_EXT.1.1 The TSF shall restrict the ability to perform the functions in column 3 of Table 10 to the user.

FMT_MOF_EXT.1.2 The TSF shall restrict the ability to perform the functions in column 5 of Table 10 to the administrator when the device is enrolled and according to the administrator-configured policy.

Note: Table 10 may be found in Section 5.5.3.

5.5.2 FMT_POL_EXT

Family Behaviour

This family defines the requirements for trusted policy update.

Component Levelling

FMT_POL_EXT: Trusted Policy Update
2

Figure 23 – FMT_POL_EXT: Trusted Policy Update Component Levelling

Management

There are no management activities foreseen.

Audit

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: Failure of policy validation.

5.5.2.1 FMT_POL_EXT.2 Trusted Policy Update

Hierarchical to: No other components.

Dependencies: No dependencies.

FMT_POL_EXT.2.1 The MDM Agent shall only accept policies and policy updates digitally signed by the Enterprise.

FMT_POL_EXT.2.2 The MDM Agent shall not install policies if the policy signing certificate is deemed invalid.

5.5.3 FMT_SMF

Family Behaviour

This family defines the requirements for Specification of management functions.

Component Levelling

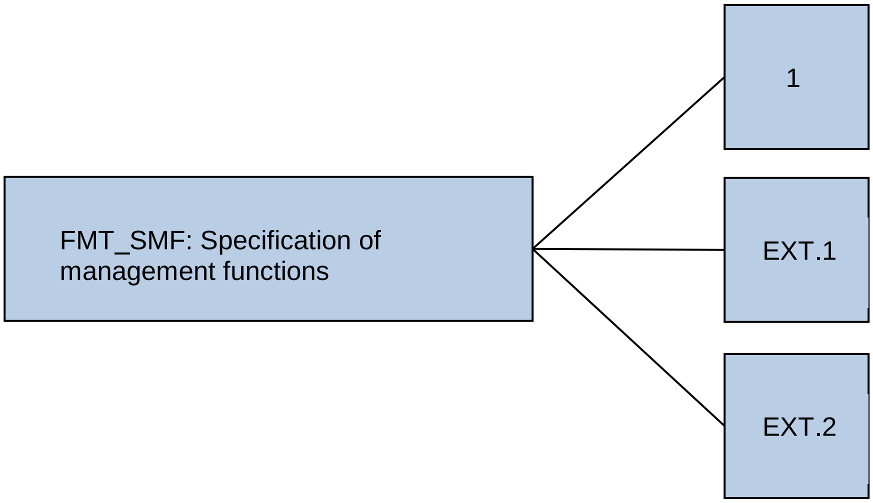

Figure 24 – FMT_SMF: Specification of Management Functions Component Levelling

Management

There are no management activities foreseen.

Audit FMT_SMF_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: Change of settings;

b. Minimal: Success or failure of function;

c. Minimal: Initiation of software update;

d. Minimal: Initiation of application installation or update.

Audit FMT_SMF_EXT.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: Unenrollment.

Audit FMT_SMF_EXT.3

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: Success or failure of function.

5.5.3.1 FMT_SMF_EXT.1 Specification of management functions

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FMT_SMF_EXT.1.1 The TSF shall be capable of performing the following management functions:

Management Function **Status Markers:** M-Mandatory O-Optional/Objective	**FMT_SMF_EXT.1**	**FMT_MOF_EXT.1.1**	**Administrator**	**FMT_MOF_EXT.1.2**
1. configure password policy: a. minimum password length b. minimum password complexity c. maximum password lifetime	M	-	M	M
2. configure session locking policy: a. screen-lock enabled/disabled b. screen lock timeout c. number of authentication failures	M	-	M	M
3. enable/disable the VPN protection: a. across device [selection: b. on a per-app basis c. no other method]	M	O	O	O
4. enable/disable [assignment: *list of radios*]	M	O	O	O
5. enable/disable [assignment: *list of audio or visual collection devices*]: a. across device [selection: b. on a per-pp basis c. no other method]	M	O	O	O
6. specify wireless networks (SSIDs) to which the TSF may connect	O	-	O	O
7. configure security policy for each wireless network: a. [selection: specify the CA(s) from which the TSF will accept WLAN authentication server certificate(s), specify the FQDN(s) of acceptable WLAN authentication server certificate(s)] b. security type c. authentication protocol d. client credentials to be used for authentication	M	-	M	O
8. transition to the locked state	M	-	M	-

Management Function **Status Markers:** M-Mandatory O-Optional/Objective	**FMT_SMF_EXT.1**	**FMT_MOF_EXT.1.1**	**Administrator**	**FMT_MOF_EXT.1.2**
9. TSF wipe of protected data	M	-	M	-
10. configure application installation policy by [selection: a. restricting the sources of applications, b. specifying a set of allowed applications based on [assignment: *application characteristics*] (an application whitelist), c. denying installation of applications]	M	-	M	M
11. import keys/secrets into the secure key storage	M	O	O	-
12. destroy imported keys/secrets and [selection: no other keys/secrets, [assignment: *list of other categories of keys/secrets*]] in the secure key storage	M	O	O	-
13. import X.509v3 certificates into the Trust Anchor Database	M	-	M	O
14. 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	M	O	O	-
15. enroll the TOE in management	M	M	-	-
16. remove applications	M	-	M	O
17. update system software	M	-	M	O
18. install applications	M	-	M	O
19. remove Enterprise applications	M	-	M	-

Management Function **Status Markers:** M-Mandatory O-Optional/Objective	**FMT_SMF_EXT.1**	**FMT_MOF_EXT.1.1**	**Administrator**	**FMT_MOF_EXT.1.2**
20. configure the Bluetooth trusted channel: a. disable/enable the Discoverable mode (for BR/EDR) b. change the Bluetooth device name [selection: c. allow/disallow additional wireless technologies to be used with Bluetooth, d. disable/enable Advertising (for LE), e. disable/enable the Connectable mode f. disable/enable the Bluetooth services and/or profiles available on the device, g. specify minimum level of security for each pairing , h. configure allowable methods of Out of Band pairing i. no other Bluetooth configuration]	M	O	O	O
21. enable/disable display notification in the locked state of: [selection: a. email notifications, b. calendar appointments, c. contact associated with phone call notification, d. text message notification, e. other application-based notifications, f. all notifications]	M	O	O	O
22. enable/disable all data signaling over [assignment: *list of externally accessible hardware ports*]	O	O	O	O
23. enable/disable [assignment: *list of protocols where the device acts as a server*]	O	O	O	O
24. enable/disable developer modes	O	O	O	O
25. enable data-at rest protection	O	O	O	O
26. enable removable media’s data-at-rest protection	O	O	O	O
27. enable/disable bypass of local user authentication	O	O	O	O
28. wipe Enterprise data	O	O	O	-
29. approve [selection: import, removal] by applications of X.509v3 certificates in the Trust Anchor Database	O	O	O	O

Management Function **Status Markers:** M-Mandatory O-Optional/Objective	**FMT_SMF_EXT.1**	**FMT_MOF_EXT.1.1**	**Administrator**	**FMT_MOF_EXT.1.2**
30. configure whether to establish a trusted channel or disallow establishment if the TSF cannot establish a connection to determine the validity of a certificate	O	O	O	O
31. enable/disable the cellular protocols used to connect to cellular network base stations	O	O	O	O
32. read audit logs kept by the TSF	O	O	O	-
33. configure [selection: certificate, public-key] used to validate digital signature on applications	O	O	O	O
34. approve exceptions for shared use of keys/secrets by multiple applications	O	O	O	O
35. approve exceptions for destruction of keys/secrets by applications that did not import the key/secret	O	O	O	O
36. configure the unlock banner	O	-	O	O
37. configure the auditable items	O	-	O	O
38. retrieve TSF-software integrity verification values	O	O	O	O
39. enable/disable [selection: a. USB mass storage mode, b. USB data transfer without user authentication, c. USB data transfer without authentication of the connecting system]	O	O	O	O
40. enable/disable backup to [selection: locally connected system, remote system]	O	O	O	O
41. enable/disable [selection: a. Hotspot functionality authenticated by [selection: pre-shared key, passcode, no authentication], b. USB tethering authenticated by [selection: pre-shared key, passcode, no authentication]]	O	O	O	O
42. approve exceptions for sharing data between [selection: application processes, groups of application processes]	O	O	O	O
43. place applications into application process groups based on [assignment: *application characteristics*]	O	O	O	O

Management Function **Status Markers:** M-Mandatory O-Optional/Objective	**FMT_SMF_EXT.1**	**FMT_MOF_EXT.1.1**	**Administrator**	**FMT_MOF_EXT.1.2**
44. enable/disable location services: a. across device [selection: b. on a per-app basis c. no other method]	M	O	O	O
45. [assignment: *list of other management functions to be provided by the TSF*]	O	O	O	O

Table 10 – Management Functions

5.5.3.2 FMT_SMF_EXT.2 Specification of remediation actions

Hierarchical to: No other components.

Dependencies: No dependencies.

FMT_SMF_EXT.2.1 The TSF shall offer [selection: wipe of protected data, wipe of sensitive data, alert the administrator, remove Enterprise applications, [assignment: *list other available remediation actions*]] upon unenrollment and [selection: [assignment: *other administrator-configured triggers*], no other triggers].

5.5.3.3 FMT_SMF_EXT.3 Specification of Management Functions

Hierarchical to: No other components.

Dependencies: No dependencies.

FMT_SMF_EXT.3.1 The MDM Agent shall be capable of interacting with the platform to perform the following functions:

[selection:

a. administrator-provided management functions in MDF PP;

b. administrator-provided device management functions in MDM PP]

c. Import the certificates to be used for authentication of MDM Agent communications

d. [selection: [assignment: *additional functions*], no additional functions].

FMT_SMF_EXT.3.2 The MDM Agent shall be capable of performing the following functions:

a. Enroll in management;

b. Configure whether users can unenroll the agent from management

c. [selection: configure periodicity of reachability events, [assignment: *other management functions*], no other functions].

5.5.4 FMT_UNR_EXT

Family Behaviour

This family defines the requirements for the prevention of user self-unenrollment.

Component Levelling

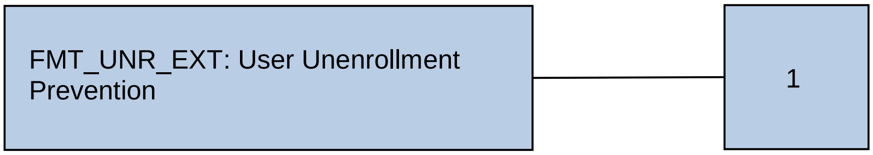

Figure 25 – FMT_UNR_EXT: User Unenrollment Prevention Component Levelling

Management

There are no management activities foreseen.

Audit

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: Attempt to unenroll.

5.5.4.1 FMT_UNR_EXT.1 User Unenrollment Prevention

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FMT_UNR_EXT.1.1 The MDM Agent shall provide a mechanism to prevent users from unenrolling the mobile device from management.

5.6 CLASS FPT: PROTECTION OF THE TSF

Several SFRs and five new families have been added to the Protection of the TSF class.

FPT_AEX_EXT is a new family, Anti-exploitation services, and is modelled after FPT_FLS Fail secure. FPT_AEX_EXT.1 Anti-exploitation services (ALSR), FPT_AEX_EXT.2 Anti-exploitation services (Memory page permissions) and FPT_AEX_EXT.1 Anti-exploitation services (Overflow protection) are all modelled after FPT_FLS.1 Failure with preservation of secure state.

FPT_BBD_EXT is a new family, Application processor mediation, and is modelled after FRU_PRS Priority of service. FPT_BBD_EXT.1 is modelled after FRU_PRS.2 Full priority of service.

FPT_KST_EXT is a new family, Key storage and transmission, and is modelled after FCS_CKM Key Management. FPT_KST_EXT.1 Key Storage, FPT_KST_EXT.2

No key transmission and FPT_KST_EXT.1 No plaintext key export are all modelled after FCS_CKM.2 Cryptographic key distribution.

FPT_NOT_EXT is a new family, Self-test notification, and is modelled after FPT_TST TSF self test. FPT_NOT_EXT.1 is modelled after FPT_TST.1 TSF testing.

FPT_TST_EXT.1 TSF cryptographic functionality testing and FPT_TST_EXT.2 Integrity testing are part of the FPT_TST TSF self test family. FPT_TST_EXT.1 and FPT_TST_EXT.2 are modelled after FPT_TST.1 TSF testing.

FPT_TUD_EXT is a new family, Trusted update, and is modelled after FPT_SSP State synchrony protocol. FPT_TUD_EXT.1 Trusted update: TSF version query and FPT_TUD_EXT.1 Trusted update verification are modelled after FPT_SSP.1 Simple trusted acknowledgement.

5.6.1 FPT_AEX_EXT

Family Behaviour

This family defines the requirements for Anti-exploitation services.

Component Levelling

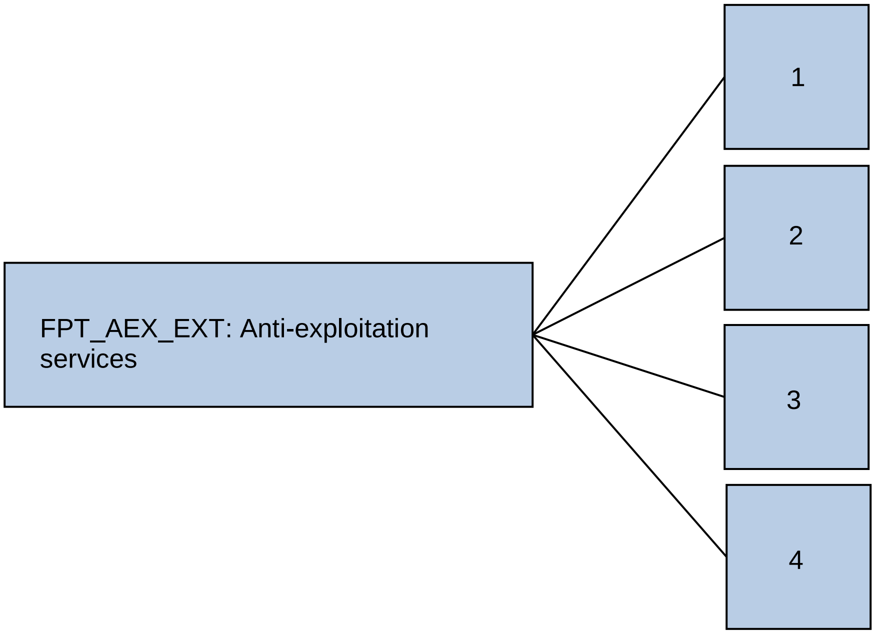

Figure 26 – FPT_AEX_EXT: Anti-Exploitation Services Component Levelling

Management

There are no management activities foreseen.

Audit FPT_AEX_EXT.4

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: Blocked attempt to modify TSF data.

5.6.1.1 FPT_AEX_EXT.1 Anti-exploitation services (ASLR)

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_AEX_EXT.1.1 The TSF shall provide address space layout randomization (ASLR) to applications.

FPT_AEX_EXT.1.2 The base address of any user-space memory mapping will consist of at least 8 unpredictable bits.

5.6.1.2 FPT_AEX_EXT.2 Anti-exploitation services (Memory page permissions)

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_AEX_EXT.2.1 The TSF shall be able to enforce read, write, and execute permissions on every page of physical memory.

FPT_AEX_EXT.2.2 The TSF shall prevent write and execute permissions from being simultaneously granted to any page of physical memory [selection: with no exceptions, assignment: [*specific exceptions*]].

5.6.1.3 FPT_AEX_EXT.3 Anti-exploitation services (Overflow protection)

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_AEX_EXT.3.1 TSF processes that execute in a non-privileged execution domain on the application processor shall implement stack-based buffer overflow protection.

5.6.1.4 FPT_AEX_EXT.4 Anti-exploitation services (Domain isolation)

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_AEX_EXT.4.1 The TSF shall protect itself from modification by untrusted subjects.

FPT_AEX_EXT.4.2 The TSF shall enforce isolation of address space between applications.

5.6.2 FPT_BBD_EXT

Family Behaviour

This family defines the requirements for Application processor mediation.

Component Levelling

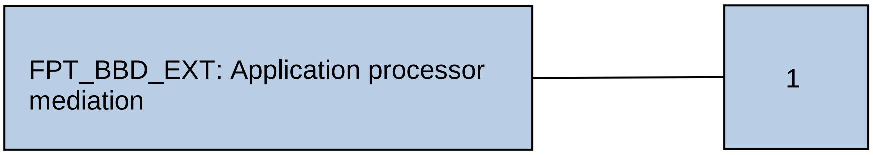

Figure 27 – FPT_BBD_EXT: Application Processor Mediation Component Levelling

Management

There are no management activities foreseen.

Audit

There are no auditable events foreseen.

5.6.2.1 FPT_BBD_EXT.1 Application processor mediation

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_BBD_EXT.1.1 The TSF shall prevent code executing on any baseband processor (BP) from accessing application processor (AP) resources except when mediated by the AP.

5.6.3 FPT_KST_EXT

Family Behaviour

This family defines the requirements for Key storage and transmission.

Component Levelling

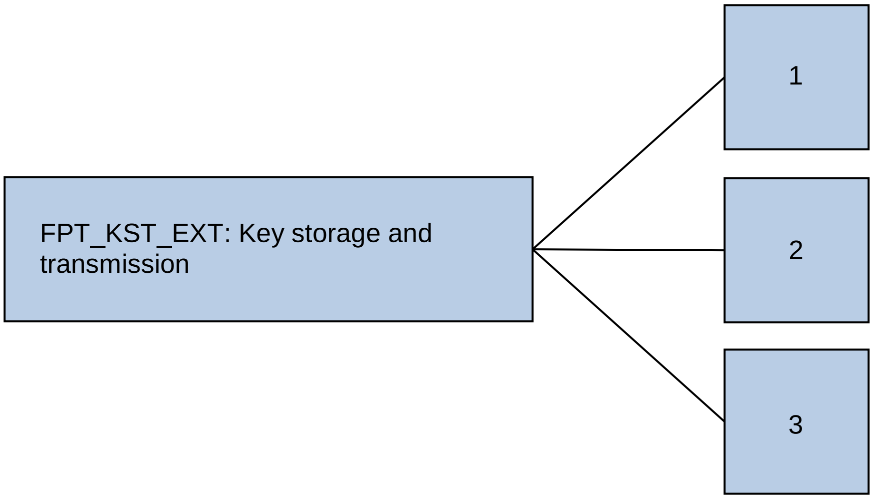

Figure 28 – FPT_KST_EXT: Key Storage and Transmission Component Levelling

Management

There are no management activities foreseen.

Audit

There are no auditable events foreseen.

5.6.3.1 FPT_KST_EXT.1 Key storage

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_KST_EXT.1.1 The TSF shall not store any plaintext key material in readable non-volatile memory.

5.6.3.2 FPT_KST_EXT.2 No key transmission

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_KST_EXT.2.1 The TSF shall not transmit any plaintext key material outside the security boundary of the TOE.

5.6.3.3 FPT_KST_EXT.3 No plaintext key export

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_KST_EXT.3.1 The TSF shall ensure it is not possible for the TOE user(s) to export plaintext keys.

5.6.4 FPT_NOT_EXT

Family Behaviour

This family defines the requirements for Self-test notification.

Component Levelling

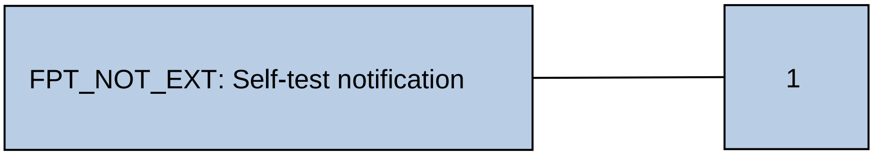

Figure 29 – FPT_NOT_EXT: Self-Test Notification Component Levelling

Management

There are no management activities foreseen.

Audit

The following actions may be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Detailed: Measurement of TSF software.

5.6.4.1 FPT_NOT_EXT.1 Self-test notification

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_NOT_EXT.1.1 The TSF shall transition to non-operational mode and [selection: log failures in the audit record, notify the administrator, [assignment: *other actions*], no other actions] when the following types of failures occur:

- failures of the self-test(s)
- TSF software integrity verification failures
- [selection: no other failures, [assignment: *other failures*]].

5.6.5 FPT_TST

Family Behaviour

This family defines the requirements for TSF self testing.

Component Levelling

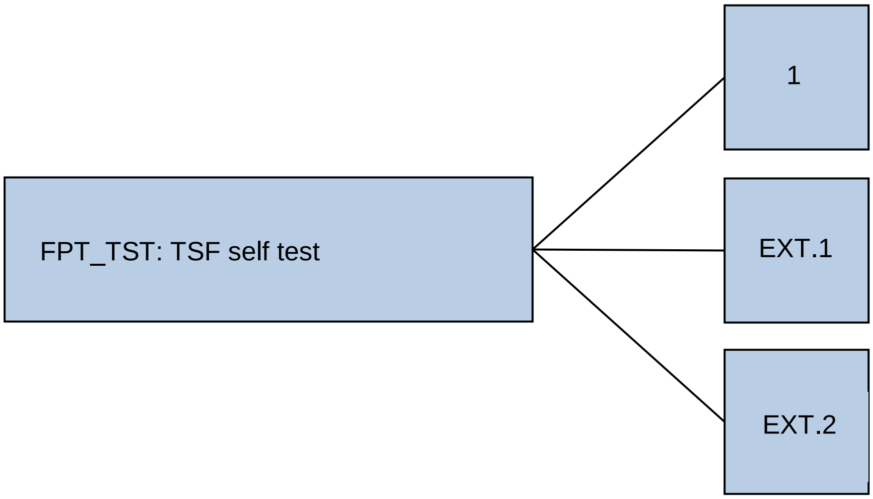

Figure 30 – FPT_TST: TSF Self Test Component Levelling

Management

There are no management activities foreseen.

Audit FPT_TST_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: Initiation of self-test;

b. Minimal: Failure of self-test.

Audit FPT_TST_EXT.2

The following actions may be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: Start-up of TOE.

b. Detailed: Detected integrity violation.

5.6.5.1 FPT_TST_EXT.1 TSF cryptographic functionality testing

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_TST.1.1 The TSF shall run a suite of self-tests during initial start-up (on power on) to demonstrate the correct operation of all cryptographic functionality.

5.6.5.2 FPT_TST_EXT.2 Integrity testing

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_TST_EXT.2.1 The TSF shall verify the integrity of the bootchain up through the Application Processor OS kernel, and [selection: all executable code stored in mutable media, [assignment: *list of other executable code*], no other executable code], stored in mutable media prior to its execution through the use of [selection: a digital signature using a hardware-protected asymmetric key, a hardware-protected hash].

FPT_TST_EXT.2.2 The TSF shall not execute code if the code signing certificate is deemed invalid.

5.6.6 FPT_TUD_EXT

Family Behaviour

This family defines the requirements for Trusted update.

Component Levelling

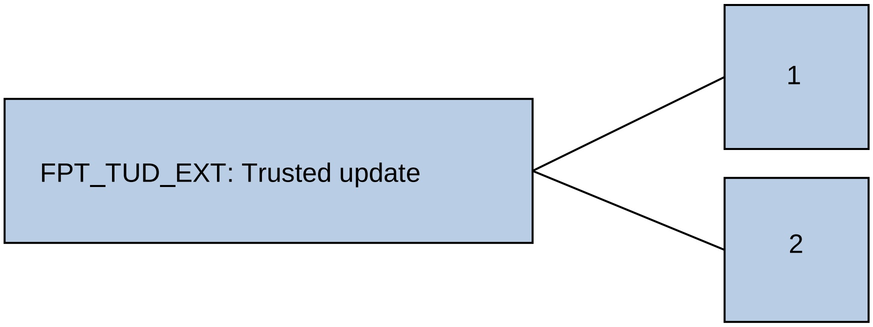

Figure 31 – FPT_TUD_EXT: Trusted Update Component Levelling

Management FPT_TUD_EXT.2

The following actions could be considered for the management functions in FMT:

a. The configured certificate used to verify the signature is set according to FMT_SMF_EXT.1 function 33.

Audit FPT_TUD_EXT.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: Success or failure of signature verification for software updates;

b. Minimal: Success or failure of signature verification for applications.

5.6.6.1 FPT_TUD_EXT.1 Trusted update: TSF version query

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_TUD_EXT.1.1 The TSF shall provide authorized users the ability to query the current version of the TOE firmware/software.

FPT_TUD_EXT.1.2 The TSF shall provide authorized users the ability to query the current version of the hardware model of the device.

FPT_TUD_EXT.1.3 The TSF shall provide authorized users the ability to query the current version of installed mobile applications.

5.6.6.2 FPT_TUD_EXT.2 Trusted update verification

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_TUD_EXT.2.1 The TSF shall verify software updates to the Application Processor system software and [selection: [assignment: *other processor system software*], no other processor system software] using a digital signature by the manufacturer prior to installing those updates.

FPT_TUD_EXT.2.2 The TSF shall [selection: never update, update only by verified software] the TSF boot integrity [selection: key, hash].

FPT_TUD_EXT.2.3 The TSF shall verify that the digital signature verification key used for TSF updates [selection: is validated to a public key in the Trust Anchor Database, matches a hardware-protected public key].

FPT_TUD_EXT.2.4 The TSF shall verify mobile application software using a digital signature mechanism prior to installation.

5.7 CLASS FTA: TOE ACCESS

Two SFRs and one new family have been added to the TOE access class.

FTA_SSL_EXT.1 TSF- and user-initiated locked state is part of the FTA_SSL Session locking and termination family. FTA_SSL_EXT.1 is modelled after FTA_SSL.2 User-initiated locking and FTA_SSL.3 TSF-initiated termination.

FTA_WSE_EXT is a new family, Wireless network access, and is modelled after FTA_TSE TOE session establishment. FTA_WSE_EXT.1 Wireless network access is modelled after FTA_TSE.1 TOE session establishment.

5.7.1 FTA_SSL

Family Behaviour

This family defines the requirements for the TSF to provide the capability for TSF-initiated and user-initiated locking, unlocking, and termination of interactive sessions.

Component Levelling

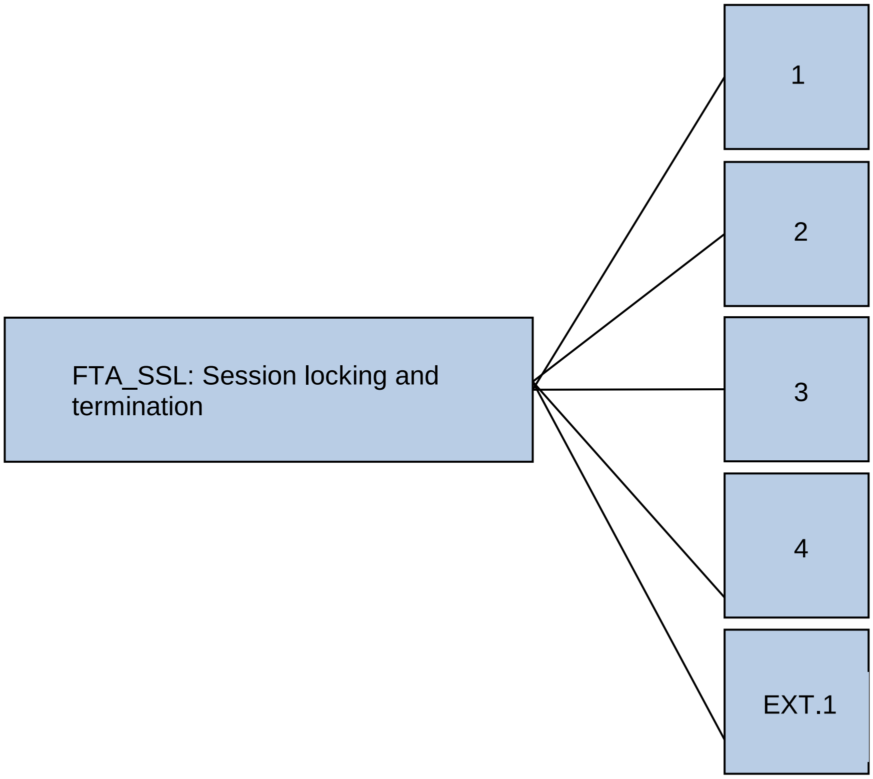

Figure 32 – FTA_SSL: Session Locking and Termination Component Levelling

Management

The following actions could be considered for the management functions in FMT:

a. The time interval of inactivity is configured using FMT_SMF_EXT.1 function 2.b.

b. The user-/administrator-initiated lock is specified in FMT_SMF_EXT.1 function 8.

Audit

There are no auditable events foreseen.

5.7.1.1 FTA_SSL_EXT.1 TSF- and user-initiated locked state

Hierarchical to: No other components.

Dependencies: No dependencies.

FTA_SSL_EXT.1.1 The TSF shall transition to a locked state after a time interval of inactivity.

FTA_SSL_EXT.1.2 The TSF shall transition to a locked state after initiation by either the user or the administrator.

FTA_SSL_EXT.1.3 The TSF shall, upon transitioning to the locked state, perform the following operations:

a) clearing or overwriting display devices, obscuring the previous contents;

b) [assignment: *other actions performed upon transitioning to the locked state*].

5.7.2 FTA_WSE_EXT

Family Behaviour

This family defines the requirements for Wireless network access.

Component Levelling

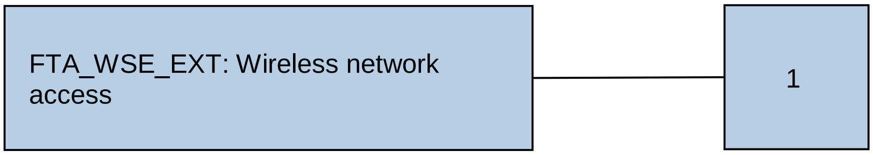

Figure 33 – FTA_WSE_EXT: Wireless Network Access Component Levelling

Management

The following actions could be considered for the management functions in FMT:

a. administrator configuration of the access points to which the TSF may connect.

Audit

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: All attempts to connect to access points.

5.7.2.1 FTA_WSE_EXT.1 Wireless network access

Hierarchical to: No other components.

Dependencies: No dependencies.

FTA_WSE_EXT.1.1 The TSF shall be able to attempt connections to wireless networks specified as acceptable networks as configured by the administrator in FMT_SMF_EXT.1.

5.8 CLASS FTP: TRUSTED PATH/CHANNEL

One new SFR has been added to the Trusted path/channel class.

FTP_ITC_EXT.1 Trusted channel communication is part of the FTP_ITC Trusted path/channel family. FTP_ITC_EXT.1 is modelled after FTP_ITC.1 Inter-TSF trusted channel.

5.8.1 FTP_ITC

Family Behaviour

This family defines the requirements for Trusted channel communications.

Component Levelling

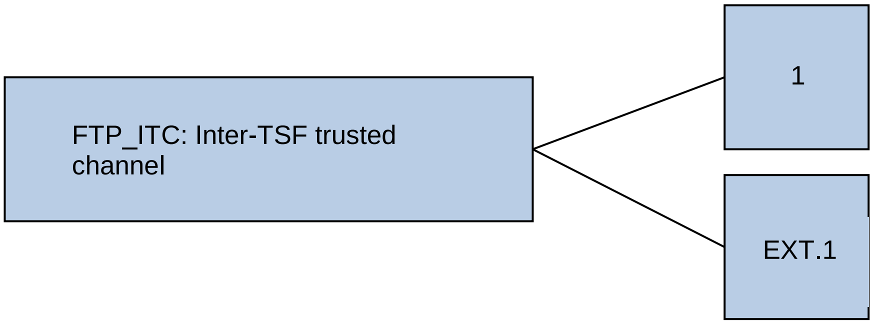

Figure 34 – FTP_ITC: Inter-TSF Trusted Channel Component Levelling

Management

There are no management activities foreseen.

Audit

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a. Minimal: Initiation and termination of trusted channel.

5.8.1.1 FTP_ITC_EXT.1 Trusted channel communication

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FTP_ITC_EXT.1.1 The TSF shall use 802.11-2012, 802.1X, and EAP-TLS and [selection, at least one of: ~~IPsec,~~ TLS, DTLS, HTTPS protocol] to provide a communication channel between itself and another trusted IT product 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.

FTP_ITC_EXT.1.2 The TSF shall permit the TSF **and the MDM Server and [MAS Server, no other IT entities]** to initiate communication via the trusted channel.

FTP_ITC_EXT.1.3 The TSF shall initiate communication via the trusted channel for wireless access point connections, administrative communication, configured enterprise connections, and [selection: OTA updates, no other connections].

5.9 EXTENDED SECURITY ASSURANCE REQUIREMENTS

This section specifies the extended Security Assurance Requirements (SARs) used in this ST. The extended SARs included in this ST originate from the Protection Profile for Mobile Device Fundamentals. The extended components are:

ALC_TSU_EXT Timely security updates

This component requires the TOE developer, in conjunction with any other necessary parties, to provide information as to how the end-user devices are updated to address security issues in a timely manner. Application notes and Assurance Activities are detailed in the Protection Profile for Mobile Device Fundamentals.

5.9.1 ALC_TSU_EXT Timely Updates (ALC_TSU_EXT)

The objective of this family is to mandate security updates for the TOE following evaluation.

5.9.1.1 ALC_TSU_EXT.1 Timely security updates

Dependencies: No dependencies.

Developer action elements:

ALC_TSU_EXT.1.1D The developer shall provide a description in the TSS of how timely security updates are made to the TOE.

Content and presentation elements:

ALC_TSU_EXT.1.1C The description shall include the process for creating and deploying security updates for the TOE software/firmware.

ALC_TSU_EXT.1.2C The description shall express the time window as the length of time, in days, between public disclosure of a vulnerability and the public availability of security updates to the TOE.

ALC_TSU_EXT.1.3C The description shall include the mechanisms publicly available for reporting security issues pertaining to the TOE.

Evaluator action elements:

ALC_TSU_EXT.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6 SECURITY REQUIREMENTS

Section 6 provides security functional and assurance requirements that must be satisfied by a compliant TOE. These requirements consist of functional components from Part 2 of the CC, extended requirements from the MDF PP, and the security assurance components specified in the MDF PP.

6.1 CONVENTIONS

The CC permits four types of operations to be performed on functional requirements: selection, assignment, refinement, and iteration (where permitted). These operations, when performed on requirements that derive from CC Part 2, are identified in this ST in the following manner:

- Selection: Indicated by surrounding brackets, e.g., [selected item].
- Assignment: Indicated by surrounding brackets and italics, e.g., [*assigned item*].
- Refinement: Refined components are identified by using **bold** for additional information, or ~~strikeout~~ for deleted text.
- Iteration: Indicated by assigning a number in parenthesis to the end of the functional component identifier as well as by modifying the functional component title to distinguish between iterations, e.g., ‘FDP_ACC.1(1), Subset access control (administrators)’ and ‘FDP_ACC.1(2) Subset access control (devices)’.

Where operations are shown as assignments in CC Part 2 and selections in the MDF PP, the operations will be shown primarily as assignments, with selections indicated within the assignments where practical.

6.2 TOE SECURITY FUNCTIONAL REQUIREMENTS

The security functional requirements for this ST consist of the following components from Part 2 of the CC and extended components defined in Section 5, summarized in Table 11. The relevant Technical Decisions are listed in Section 2.3.

Class	Identifier	Name
Security Audit (FAU)	FAU_ALT_EXT.2	Agent Alerts
Cryptographic Support (FCS)	FCS_CKM.1(1)	Cryptographic key generation
	FCS_CKM.1(2)	Cryptographic key generation
	FCS_CKM.2(1)	Cryptographic key establishment
	FCS_CKM.2(2)	Cryptographic key distribution

Class	Identifier	Name
	FCS_CKM_EXT.1	Cryptographic key support
	FCS_CKM_EXT.2	Cryptographic key random generation
	FCS_CKM_EXT.3	Cryptographic key generation
	FCS_CKM_EXT.4	Cryptographic key destruction
	FCS_CKM_EXT.5	TSF wipe
	FCS_CKM_EXT.6	Salt generation
	FCS_COP.1(1)	Cryptographic operation
	FCS_COP.1(2)	Cryptographic operation
	FCS_COP.1(3)	Cryptographic operation
	FCS_COP.1(4)	Cryptographic operation
	FCS_COP.1(5)	Cryptographic operation
	FCS_HTTPS_EXT.1	HTTPS protocol
	FCS_IV_EXT.1	Initialization vector generation
	FCS_RBG_EXT.1	Cryptographic operation (Random bit generation)
	FCS_SRV_EXT.1	Cryptographic algorithm services
	FCS_STG_EXT.1	Cryptographic key storage
	FCS_STG_EXT.2	Encrypted cryptographic key storage
	FCS_STG_EXT.3	Integrity of encrypted key storage
	FCS_STG_EXT.4	Cryptographic key storage
	FCS_TLSC_EXT.1	EAP-TLS protocol
	FCS_TLSC_EXT.2	TLS protocol
User Data Protection (FDP)	FDP_ACF_EXT.1	Security access control
	FDP_DAR_EXT.1	Data-at-rest protection
	FDP_IFC_EXT.1	Subset information flow control

Class	Identifier	Name
	FDP_STG_EXT.1	User data storage
	FDP_UPC_EXT.1	Inter-TSF user data transfer protection
Identification and Authentication (FIA)	FIA_AFL_EXT.1	Authentication failure handling
	FIA_BLT_EXT.1	Bluetooth user authorization
	FIA_BLT_EXT.2	Bluetooth authentication
	FIA_ENR_EXT.1	Enrollment of Mobile Device into Management
	FIA_PAE_EXT.1	PAE authentication
	FIA_PMG_EXT.1	Password management
	FIA_TRT_EXT.1	Authentication throttling
	FIA_UAU.7	Protected authentication feedback
	FIA_UAU_EXT.1	Authentication for cryptographic operation
	FIA_UAU_EXT.2	Timing of authentication
	FIA_UAU_EXT.3	Re-authentication
	FIA_X509_EXT.1	Validation of certificates
	FIA_X509_EXT.2	X509 certificate authentication
	FIA_X509_EXT.3	Request validation of certificates
Security Management (FMT)	FMT_MOF_EXT.1	Management of security functions behavior
	FMT_POL_EXT.2	Trusted policy update
	FMT_SMF_EXT.1	Specification of Management Functions
	FMT_SMF_EXT.2	Specification of remediation actions
	FMT_SMF_EXT.3	Specification of Management Functions
	FMT_UNR_EXT.1	User Unenrollment Prevention
Protection of the TSF (FPT)	FPT_AEX_EXT.1	Anti-exploitation services (ASLR)
	FPT_AEX_EXT.2	Anti-exploitation services (Memory page permissions)

Class	Identifier	Name
	FPT_AEX_EXT.3	Anti-exploitation services (Overflow protection)
	FPT_AEX_EXT.4	Anti-exploitation services (Domain isolation)
	FPT_BBD_EXT.1	Application processor mediation
	FPT_KST_EXT.1	Key storage
	FPT_KST_EXT.2	No key transmission
	FPT_KST_EXT.3	No plaintext key export
	FPT_NOT_EXT.1	Self-test notification
	FPT_STM.1	Reliable time stamps
	FPT_TST_EXT.1	TSF cryptographic functionality testing
	FPT_TST_EXT.2	Integrity testing
	FPT_TUD_EXT.1	Trusted update: TSF version query
	FPT_TUD _EXT.2	Trusted update verification
TOE Access (FTA)	FTA_SSL_EXT.1	TSF- and user-initiated locked state
	FTA_TAB.1	Default TOE access banners
	FTA_WSE_EXT.1	Wireless network access
Trusted path/channels (FTP)	FTP_ITC_EXT.1	Trusted channel communication

Table 11 – Summary of Security Functional Requirements

6.2.1 Security Audit (FAU)

6.2.1.1 FAU_ALT_EXT.2 Agent Alerts

Hierarchical to: No other components.

Dependencies: No dependencies.

FAU_ALT_EXT.2.1 The MDM Agent shall provide an alert via the trusted channel to the MDM Server in the event of any of the following:

a. successful application of policies to a mobile device;

b. [receiving] periodic reachability events;

[selection:

c. change in enrollment state,

f. [no other events].

FAU_ALT_EXT.2.2 The MDM Agent shall queue alerts if the trusted channel is not available.

6.2.2 Cryptographic Support (FCS)

6.2.2.1 FCS_CKM.1(1) Cryptographic key generation

Hierarchical to: No other components.

Dependencies: [FCS_CKM.2 Cryptographic key distribution,
or FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1(1) The TSF shall generate **asymmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm [

- [*ECC schemes*] ~~and specified~~ **using** [*"NIST curves" P-256, P-384 and [P-521]*] that meet the following: [*FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4*];

6.2.2.2 FCS_CKM.1(2) Cryptographic key generation

Hierarchical to: No other components.

Dependencies: [FCS_CKM.2 Cryptographic key distribution,
or FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1(2) The TSF shall generate **symmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm [*PRF-384*] and specified cryptographic key sizes [*128 bits*] **using a Random Bit Generator as specified in FCS_RBG_EXT.1** that meet the following: [IEEE 802.11-2012].

6.2.2.3 FCS_CKM.2(1) Cryptographic key establishment

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or

FDP_ITC.2 Import of user data with security attributes, or

FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction

FCS_CKM.2.1(1) The TSF shall ~~distribute cryptographic keys~~ **perform cryptographic key establishment** in accordance with a specified cryptographic key ~~distribution~~ **establishment** method:

- [*RSA-based key establishment schemes*] that meets the following: [*NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography"*];

- [*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"*]

6.2.2.4 FCS_CKM.2(2) Cryptographic key distribution

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or

FDP_ITC.2 Import of user data with security attributes, or

FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction

FCS_CKM.2.1(2) The TSF shall ~~distribute cryptographic keys~~ **decrypt Group Temporal Key (GTK)** in accordance with a specified cryptographic key distribution method [*AES Key Wrap in an EAPOL-Key frame*] that meets the following: [*NIST SP 800-38F, IEEE 802.11-2012 for the packet format and timing considerations*] **and does not expose the cryptographic keys**.

6.2.2.5 FCS_CKM_EXT.1 Cryptographic key support

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_CKM_EXT.1.1 The TSF shall support a [hardware-isolated] REK with a [symmetric] key of strength [256 bits].

FCS_CKM_EXT.1.2 System software on the TSF shall be able only to request [NIST SP 800-108 key derivation] by the key and shall not be able to read, import, or export a REK.

FCS_CKM_EXT.1.3 A REK shall be generated by a RBG in accordance with FCS_RBG_EXT.1.

FCS_CKM_EXT.1.4 A REK shall not be able to be read from or exported from the hardware.

6.2.2.6 FCS_CKM_EXT.2 Cryptographic key random generation

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_CKM_EXT.2.1 All DEKs shall be randomly generated with entropy corresponding to the security strength of AES key sizes of [256] bits.

6.2.2.7 FCS_CKM_EXT.3 Cryptographic key generation

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_CKM_EXT.3.1 The TSF shall use [security strength,[256-bit] symmetric KEKs] corresponding to at least the security strength of the keys encrypted by the KEK.

FCS_CKM_EXT.3.2 The TSF shall generate all KEKs using one of the following methods:

a) derive the KEK from a Password Authentication Factor using PBKDF and [

b) generate the KEK using an RBG that meets this profile (as specified in FCS_RBG_EXT.1)

d) Combine the KEK from other KEKs in a way that preserves the effective entropy of each factor by [concatenating the keys and use a KDF (as described in SP 800-108)]

6.2.2.8 FCS_CKM_EXT.4 Cryptographic key destruction

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_CKM_EXT.4.1 The TSF shall destroy cryptographic keys in accordance with the specified cryptographic key destruction methods:

- by clearing the KEK encrypting the target key,
- in accordance with the following rules:
  - o For volatile memory, the destruction shall be executed by a single direct overwrite [consisting of zeroes].
  - o For non-volatile EEPROM, the destruction shall be executed by a single direct overwrite consisting of a pseudo random pattern using the TSF's RBG (as specified in FCS_RBG_EXT.1), followed a read-verify.
  - o For non-volatile flash memory that is not wear-leveled, the destruction shall be executed [by a block erase that erases the reference to memory that stores data as well as the data itself].
  - o For non-volatile flash memory that is wear-leveled, the destruction shall be executed [a single direct overwrite consisting of zeros].
  - o For non-volatile memory other than EEPROM and flash, the destruction shall be executed by overwriting three or more times with a random pattern that is changed before each write.

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

6.2.2.9 FCS_CKM_EXT.5 TSF wipe

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_CKM_EXT.5.1 The TSF shall wipe all protected data by [

- Cryptographically erasing the encrypted DEKs and/or the KEKs in non-volatile memory by following the requirements in FCS_CKM_EXT.4.1.]

FCS_CKM_EXT.5.2 The TSF shall perform a power cycle on conclusion of the wipe procedure.

6.2.2.10 FCS_CKM_EXT.6 Salt generation

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_CKM_EXT.6.1 The TSF shall generate all salts using a RBG that meets FCS_RBG_EXT.1.

6.2.2.11 FCS_COP.1(1) Cryptographic operation

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or

FDP_ITC.2 Import of user data with security attributes, or

FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1(1) The TSF shall perform [*encryption/decryption*] in accordance with a specified cryptographic algorithm [

- *AES-CBC (as defined in FIPS PUB 197, and NIST SP 800-38A) mode,*
- *AES-CCMP (as defined in FIPS PUB 197, NIST SP 800-38C and IEEE 802.11-2012), and*

[

- *AES Key Wrap (KW) (as defined in NIST SP 800-38F)*
- *AES-GCM (as defined in NIST SP 800-38D),*
- *AES-CCM (as defined in NIST SP 800-38C),*
- *AES-XTS (as defined in NIST SP 800-38E) mode*]]

and cryptographic key sizes [*128-bit key sizes and* [256-bit key sizes]] ~~that meet the following: [assignment: *list of standards*]~~.

6.2.2.12 FCS_COP.1(2) Cryptographic operation

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or

FDP_ITC.2 Import of user data with security attributes, or

FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1(2) The TSF shall perform [*cryptographic hashing*] in accordance with a specified cryptographic algorithm [*SHA-1 and* [SHA-256, SHA-384, SHA-512] and ~~cryptographic key~~ **message digest** sizes [*160 and* [256, 384, 512 bits]] that meet the following: [*FIPS Pub 180-4*].

6.2.2.13 FCS_COP.1(3) Cryptographic operation

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or

FDP_ITC.2 Import of user data with security attributes, or

FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1(3) The TSF shall perform [*cryptographic signature services (generation and verification)*] in accordance with a specified cryptographic algorithm [

- [*RSA schemes*] ~~and~~ **using** cryptographic key sizes [*of 2048-bit or greater*] that meet the following: [*FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 4*]

and [

- [*ECDSA schemes*] ~~and cryptographic key sizes~~ **using** [*"NIST curves" P-256, P-384 and* [P-521]] that meet the following: [*FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5*]

]].

6.2.2.14 FCS_COP.1(4) Cryptographic operation

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or

FDP_ITC.2 Import of user data with security attributes, or

FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1(4) The TSF shall perform [*keyed-hash message authentication*] in accordance with a specified cryptographic algorithm [*HMAC-SHA-1 and* [HMAC-SHA-256, HMAC-SHA-384] and cryptographic key sizes [160, 256, 384 bits] **and message digest sizes 160 and [256, 384] bits** that meet the following: [*FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code, and FIPS Pub 180-4, "Secure Hash Standard*].

6.2.2.15 FCS_COP.1(5) Cryptographic operation

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or

FDP_ITC.2 Import of user data with security attributes, or

FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1(5) The TSF shall perform [*Password-based Key Derivation Functions*] in accordance with a specified cryptographic algorithm [*HMAC-[SHA-256]*], **with [25,000] iterations, and output** ~~and~~ cryptographic key sizes [*256*] that meet the following: [*NIST SP 800-132*].

6.2.2.16 FCS_HTTPS_EXT.1 HTTPS protocol

Hierarchical to: No other components.

Dependencies: No dependencies.

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

FCS_ HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS (FCS_TLSC_EXT.2).

FCS_ HTTPS_EXT.1.3 The TSF shall notify the application and [request application authorization to establish the connection] if the peer certificate is deemed invalid.

6.2.2.17 FCS_IV_EXT.1 Initialization vector generation

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_IV_EXT.1.1 The TSF shall generate IVs in accordance with Table 12: References and IV Requirements for NIST-approved Cipher Modes.

Cipher Mode	Reference	IV Requirements
Counter (CTR)	SP 800-38A	"Initial Counter" shall be non-repeating. No counter value shall be repeated across multiple messages with the same secret key.
Cipher Block Chaining (CBC)	SP 800-38A	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.
XEX (XOR Encrypt XOR) Tweakable Block Cipher with Ciphertext Stealing (XTS)	SP 800-38E	No IV. Tweak values shall be non-negative integers, assigned consecutively, and starting at an arbitrary non-negative integer.
Key Wrap and Key Wrap with Padding	SP 800-38F	No IV
Counter with CBC-Message Authentication Code (CCM)	SP 800-38C	No IV. Nonces shall be non-repeating.
Galois Counter Mode (GCM)	SP 800-38D	IV shall be non-repeating. The number of invocations of GCM shall not exceed 2^32 for a given secret key unless an implementation only uses 96-bit IVs (default length).

Table 12 – References and IV Requirements for NIST-approved Cipher Modes

6.2.2.18 FCS_RBG_EXT.1 Cryptographic operation (Random bit generation)

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RBG_EXT.1.1 The TSF shall perform all deterministic random bit generation services in accordance with [NIST Special Publication 800-90A using [HMAC_DRBG (any)]].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by an entropy source that accumulates entropy from [a software-based noise source and TSF-hardware-based noise source] with a minimum of [256 bits] of entropy at least equal to the greatest security strength (according to NIST SP 800-57) of the keys and hashes that it will generate.

FCS_RBG_EXT.1.3 The TSF shall be capable of providing output of the RBG to applications running on the TSF that request random bits.

6.2.2.19 FCS_SRV_EXT.1 Cryptographic algorithm services

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_SRV_EXT.1.1 The TSF shall provide a mechanism for applications to request the TSF to perform the following cryptographic operations:

- All mandatory and [selected algorithms] in FCS_CKM.2(1)
- The following algorithms in FCS_COP.1(1): AES-CBC, [AES Key Wrap, AES-GCM, AES-CCM]
- All mandatory and selected algorithms in FCS_COP.1(3)
- All mandatory and selected algorithms in FCS_COP.1(2)
- All mandatory and selected algorithms in FCS_COP.1(4) [
- No other cryptographic operations].

6.2.2.20 FCS_STG_EXT.1 Cryptographic key storage

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_STG_EXT.1.1 The TSF shall provide [software-based] secure key storage for asymmetric private keys and [symmetric keys, persistent secrets].

FCS_STG_EXT.1.2 The TSF shall be capable of importing keys/secrets into the secure key storage upon request of [the user, the administrator] and [applications running on the TSF].

FCS_STG_EXT.1.3 The TSF shall be capable of destroying keys/secrets in the secure key storage upon request of [the user, the administrator].

FCS_STG_EXT.1.4 The TSF shall have the capability to allow only the application that imported the key/secret the use of the key/secret. Exceptions may only be explicitly authorized by [the administrator].

FCS_STG_EXT.1.5 The TSF shall allow only the application that imported the key/secret to request that the key/secret be destroyed. Exceptions may only be explicitly authorized by [the user, the administrator].

6.2.2.21 FCS_STG_EXT.2 Encrypted cryptographic key storage

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_STG_EXT.2.1 The TSF shall encrypt all DEKs and KEKs and [long-term trusted channel key material, all software-based key storage] by KEKs that are [

1) Protected by the REK with [

   b. encryption by a KEK chaining to a REK],

2) Protected by the REK and the password with [

   b. encryption by a KEK chaining to a REK and the password-derived KEK]

].

FCS_STG_EXT.2.2 DEKs and KEKs and [all software-based key storage] shall be encrypted using one of the following methods [using AES in the [GCM, CBC mode]].

6.2.2.22 FCS_STG_EXT.3 Integrity of encrypted key storage

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_STG_EXT.3.1 The TSF shall protect the integrity of any encrypted DEKs and KEKs and [no other keys] by [

- a hash (FCS_COP.1(2)) of the stored key that is encrypted by a key protected by FCS_STG_EXT.2;
- a keyed hash (FCS_COP.1(4)) using a key protected by a key protected by FCS_STG_EXT.2.

FCS_STG_EXT.3.2 The TSF shall verify the integrity of the [hash, MAC] of the stored key prior to use of the key.

6.2.2.23 FCS_STG_EXT.4 Cryptographic key storage

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_STG_EXT.4.1 The MDM Agent shall store persistent secrets and private keys when not in use in platform-provided key storage.

6.2.2.24 FCS_TLSC_EXT.1 EAP-TLS protocol

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_TLSC_EXT.1.1 The TSF shall implement TLS 1.0 and [TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)] supporting the following ciphersuites: [

- Mandatory Ciphersuites:
  - TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246
- [Optional Ciphersuites:
  - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

  ]].

FCS_TLSC_EXT.1.2 The TSF shall verify that the server certificate presented for EAP-TLS [chains to one of the specified CAs].

FCS_TLSC_EXT.1.3 The TSF shall not establish a trusted channel if the peer certificate is invalid.

FCS_TLSC_EXT.1.4 The TSF shall support mutual authentication using X.509v3 certificates.

FCS_TLSC_EXT.1.5 The TSF shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [secp256r1, secp384r1, secp521r1] and no other curves.

6.2.2.25 FCS_TLSC_EXT.2 TLS protocol

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_TLSC_EXT.2.1 The TSF shall implement TLS 1.2 (RFC 5246) supporting the following ciphersuites: [

- Mandatory Ciphersuites:
    - TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246
- [Optional Ciphersuites:
    - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

]].

FCS_TLSC_EXT.2.2 The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.2.3 The TSF shall not establish a trusted channel if the peer certificate is invalid.

FCS_TLSC_EXT.2.4 The TSF shall support mutual authentication using X.509v3 certificates.

FCS_TLSC_EXT.2.5 The TSF shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [secp256r1, secp384r1 secp521r1] and no other curves.

6.2.3 User Data Protection (FDP)

6.2.3.1 FDP_ACF_EXT.1 Security access control

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_ACF_EXT.1.1 The TSF shall provide a mechanism to restrict the system services that are accessible to an application.

FDP_ACF_EXT.1.2 The TSF shall provide an access control policy that prevents [application processes] from accessing [all] data stored by other [application processes, groups of application processes]. Exceptions may only be explicitly authorized for such sharing by [a common application developer].

6.2.3.2 FDP_DAR_EXT.1 Data-at-rest protection

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_DAR_EXT.1.1 Encryption shall cover all protected data.

FDP_DAR_EXT.1.2 Encryption shall be performed using DEKs with AES in the [CBC] mode with key size [256] bits.

6.2.3.3 FDP_IFC_EXT.1 Subset information flow control

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_IFC_EXT.1.1 The TSF shall [enable all IP traffic (other than IP traffic required to establish the VPN connection) to flow through the IPsec VPN client].

6.2.3.4 FDP_STG_EXT.1 User data storage

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_STG_EXT.1.1 The TSF shall provide protected storage for the Trust Anchor Database.

6.2.3.5 FDP_UPC_EXT.1 Inter-TSF user data transfer protection

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_UPC_EXT.1.1 The TSF provide a means for non-TSF applications executing on the TOE to use TLS, HTTPS, Bluetooth BR/EDR, and [no other protocol] to provide a protected communication channel between the non-TSF application and another IT product 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.

FDP_UPC_EXT.1.2 The TSF shall permit the non-TSF applications to initiate communication via the trusted channel.

6.2.4 Identification and Authentication (FIA)

6.2.4.1 FIA_AFL_EXT.1 Authentication failure handling

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_AFL_EXT.1.1 The TSF shall detect when a configurable positive integer within [*3 to 10*] of unsuccessful authentication attempts occur related to last successful authentication by that user.

FIA_AFL_EXT.1.2 When the defined number of unsuccessful authentication attempts has been surpassed, the TSF shall perform wipe of all protected data.

FIA_AFL_EXT.1.3 The TSF shall maintain the number of unsuccessful authentication attempts that have occurred upon power off.

6.2.4.2 FIA_BLT_EXT.1 Bluetooth user authorization

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_BLT_EXT.1.1 The TSF shall require explicit user authorization before pairing with a remote Bluetooth device.

FIA_BLT_EXT.1.2 The TSF shall require explicit user authorization before granting trusted remote devices access to services associated with the following Bluetooth profiles: [*all available Bluetooth profiles*], and shall require explicit user authorization before granting untrusted remote devices access to services associated with the following Bluetooth profiles: [*all available Bluetooth profiles*].

6.2.4.3 FIA_BLT_EXT.2 Bluetooth Mutual Authentication

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_BLT_EXT.2.1 The TSF shall require Bluetooth mutual authentication between devices prior to any data transfer over the Bluetooth link.

6.2.4.4 FIA_ENR_EXT.2 Enrollment of Mobile Device into Management

Hierarchical to: No other components.

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

6.2.4.5 FIA_PAE_EXT.1 PAE authentication

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_PAE_EXT.1.1 The TSF shall conform to IEEE Standard 802.1X for a Port Access Entity (PAE) in the “Supplicant” role.

6.2.4.6 FIA_PMG_EXT.1 Password management

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_PMG_EXT.1.1 The TSF shall support the following for the Password Authentication Factor:

1. Passwords shall be able to be composed of any combination of [upper and lower case letters, [*Basic Latin character set*]], numbers, and special characters: [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”];
2. Password length up to [32] characters shall be supported.

6.2.4.7 FIA_TRT_EXT.1 Authentication throttling

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_TRT_EXT.1.1 The TSF shall limit automated user authentication attempts by [preventing authentication via an external port]. The minimum delay shall be such that no more than 10 attempts can be attempted per 500 milliseconds.

6.2.4.8 FIA_UAU.7 Protected authentication feedback

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

FIA_UAU.7.1 The TSF shall provide only [*obscured feedback to the device's display*] to the user while the authentication is in progress.

6.2.4.9 FIA_UAU_EXT.1 Authentication for cryptographic operation

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU_EXT.1.1 The TSF shall require the user to present the Password Authentication Factor prior to decryption of protected data and encrypted DEKs, KEKs and [no other keys] at startup.

6.2.4.10 FIA_UAU_EXT.2 Timing of authentication

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU_EXT.2.1 The TSF shall allow [*viewing of notifications, viewing of mobile network and Wi-Fi strength, viewing of battery life indicator, use of the camera application, turning on/off device, emergency calling*] on behalf of the user to be performed before the user is authenticated.

FIA_UAU_EXT.2.2 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

6.2.4.11 FIA_UAU_EXT.3 Re-authentication

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU_EXT.3.1 The TSF shall require the user to enter the correct Password Authentication Factor when the user changes the Password Authentication Factor, and following TSF- and user-initiated locking in order to transition to the unlocked state, and [no other conditions].

6.2.4.12 FIA_X509_EXT.1 Validation of certificates

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_X509_EXT.1.1 The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a certificate in the Trust Anchor Database.
- 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 [the Online Certificate Status Protocol (OCSP) as specified in RFC 2560].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - o Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
  - o 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.
  - o (Conditional) 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.

FIA_X509_EXT.1.2 The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

6.2.4.13 FIA_X509_EXT.2 X509 certificate authentication

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_X509_EXT.2.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for EAP-TLS exchanges, and [IPsec, TLS, HTTPS], and [[*MDM Communication*]].

FIA_X509_EXT.2.2 When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [allow the administrator to choose whether to accept the certificate in these cases].

6.2.4.14 FIA_X509_EXT.3 Request validation of certificates

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_X509_EXT.3.1 The TSF shall provide a certificate validation service to applications.

FIA_X509_EXT.3.2 The TSF shall respond to the requesting application with the success or failure of the validation.

6.2.5 Security Management (FMT)

6.2.5.1 FMT_MOF_EXT.1 Management of security functions behavior

Hierarchical to: No other components.

Dependencies: No dependencies.

FMT_MOF_EXT.1.1 The TSF shall restrict the ability to perform the functions in column 3 of Table 13 to the user.

FMT_MOF_EXT.1.2 The TSF shall restrict the ability to perform the functions in column 5 of Table 13 to the administrator when the device is enrolled and according to the administrator-configured policy.

6.2.5.2 FMT_POL_EXT.2 Trusted Policy Update

Hierarchical to: No other components.

Dependencies: No dependencies.

FMT_POL_EXT.2.1 The MDM Agent shall only accept policies and policy updates digitally signed by the Enterprise.

FMT_POL_EXT.2.2 The MDM Agent shall not install policies if the policy signing certificate is deemed invalid.

6.2.5.3 FMT_SMF_EXT.1 Specification of management functions

Dependencies: No dependencies.

FMT_SMF_EXT.1.1 The TSF shall be capable of performing the following management functions:

Management Function **Status Markers:** Y-Claimed functionality	**FMT_SMF_EXT.1**	**FMT_MOF_EXT.1.1**	**Administrator**	**FMT_MOF_EXT.1.2**
1. configure password policy: a. minimum password length b. minimum password complexity c. maximum password lifetime	Y		Y	Y
2. configure session locking policy: a. screen-lock enabled/disabled b. screen lock timeout c. number of authentication failures	Y		Y	Y
3. enable/disable the VPN protection: a. across device [b. on a per-app basis]	Y		Y	Y
4. enable/disable [*Mobile network*]	Y	Y		
4. enable/disable [*Wi-Fi, GPS, FM radio, NFC and Bluetooth*]	Y		Y	Y
5. enable/disable [*camera*, *microphone*]: a. across device [*c. no other method*]	Y		Y	Y
6. specify wireless networks (SSIDs) to which the TSF may connect	Y		Y	Y
7. configure security policy for each wireless network: a. [specify the CA(s) from which the TSF will accept WLAN authentication server certificate(s)]	Y		Y	Y

Management Function **Status Markers:** Y-Claimed functionality	**FMT_SMF_EXT.1**	**FMT_MOF_EXT.1.1**	**Administrator**	**FMT_MOF_EXT.1.2**
b. security type c. authentication protocol d. client credentials to be used for authentication				
8. transition to the locked state	Y		Y	
9. TSF wipe of protected data	Y		Y	
10. configure application installation policy by [a. restricting the sources of applications, c. denying installation of applications]	Y		Y	Y
11. import keys/secrets into the secure key storage	Y			
12. destroy imported keys/secrets and [no other keys/secrets] in the secure key storage	Y			
13. import X.509v3 certificates into the Trust Anchor Database	Y		Y	Y
14. remove imported X.509v3 certificates and [[*All other X.509v3 certificates*]] in the Trust Anchor Database	Y		Y	
15. enroll the TOE in management	Y	Y		
16. remove applications	Y		Y	
17. update system software	Y		Y	
18. install applications	Y		Y	Y
19. remove Enterprise applications	Y		Y	
20. configure the Bluetooth trusted channel: a. disable/enable the Discoverable mode (for BR/EDR) b. change the Bluetooth device name f. disable/enable the Bluetooth services and/or profiles available on the device.	Y			
21. enable/disable display notification in the locked state of: [a. email notifications, b. calendar appointments, c. contact associated with phone call notification, d. text message notification,]	Y			Y
22. enable/disable all data signaling over [*USB, SD Card, HDMI*]	Y	Y	Y	Y

Management Function **Status Markers:** Y-Claimed functionality	**FMT_SMF_EXT.1**	**FMT_MOF_EXT.1.1**	**Administrator**	**FMT_MOF_EXT.1.2**
23. enable/disable [Media sharing, Miracase, BlackBerry Bridge, Wi-Fi hotspot, Bluetooth]	Y		Y	
24. enable/disable developer modes	Y		Y	Y
25. enable data-at rest protection	Y		Y	Y
26. enable removable media's data-at-rest protection	Y		Y	Y
27. enable/disable bypass of local user authentication	Y		Y	Y
28. wipe Enterprise data	Y		Y	
29. approve [selection: import, removal] by applications of X.509v3 certificates in the Trust Anchor Database				
30. configure whether to establish a trusted channel or disallow establishment if the TSF cannot establish a connection to determine the validity of a certificate	Y			
31. enable/disable the cellular protocols used to connect to cellular network base stations	Y			
32. read audit logs kept by the TSF				
33. configure [certificate] used to validate digital signature on applications	Y		Y	Y
34. approve exceptions for shared use of keys/secrets by multiple applications			Y	Y
35. approve exceptions for destruction of keys/secrets by applications that did not import the key/secret				
36. configure the unlock banner	Y		Y	Y
37. configure the auditable items	Y		Y	Y
38. retrieve TSF-software integrity verification values				
39. enable/disable [a. USB mass storage mode]	Y	Y	Y	Y
40. enable/disable backup to [locally connected system, remote system]	Y			Y
41. enable/disable [a. Hotspot functionality authenticated by [pre-shared key], b. USB tethering authenticated by [passcode]]	Y		Y	
42. approve exceptions for sharing data between [selection:				

Management Function **Status Markers:** Y-Claimed functionality	**FMT_SMF_EXT.1**	**FMT_MOF_EXT.1.1**	**Administrator**	**FMT_MOF_EXT.1.2**
application processes, groups of application processes]				
43. place applications into application process groups based on [assignment: *application characteristics*]				
44. enable/disable location services: a. across device [c. no other method]	Y		Y	Y
45. [a. *enable/disable automatic transfer of diagnostic data to an external device or service other than an MDM service with which the device has enrolled;* b. *disenroll the TOE in management;* c. *enable/disable multi-user modes;* d. *enable/disable automatic updates of system software (see function 17);* e. *wipe non-enterprise data*].				Y

Table 13 – Management Functions

6.2.5.4 FMT_SMF_EXT.2 Specification of remediation actions

Hierarchical to: No other components.

Dependencies: No dependencies.

FMT_SMF_EXT.2.1 The TSF shall offer [[*delete all device data*]] upon unenrollment and [no other triggers].

6.2.5.5 FMT_SMF_EXT.3 Specification of Management Functions

Hierarchical to: No other components.

Dependencies: No dependencies.

FMT_SMF_EXT.3.1 The MDM Agent shall be capable of interacting with the platform to perform the following functions:

[a. administrator-provided management functions in MDF PP];

c. Import the certificates to be used for authentication of MDM Agent communications;

d. [no additional functions].

FMT_SMF_EXT.3.2 The MDM Agent shall be capable of performing the following functions:

a. Enroll in management;

b. Configure whether users can unenroll the agent from management;

c. [configure periodicity of reachability events].

6.2.5.6 FMT_UNR_EXT.1 User Unenrollment Prevention

Hierarchical to: No other components.

Dependencies: No dependencies.

FMT_UNR_EXT.1.1 The MDM Agent shall provide a mechanism to prevent users from unenrolling the mobile device from management.

6.2.6 Protection of the TSF (FPT)

6.2.6.1 FPT_AEX_EXT.1 Anti-exploitation services (ASLR)

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_AEX_EXT.1.1 The TSF shall provide address space layout randomization (ASLR) to applications.

FPT_AEX_EXT.1.2 The base address of any user-space memory mapping will consist of at least 8 unpredictable bits.

6.2.6.2 FPT_AEX_EXT.2 Anti-exploitation services (Memory page permissions)

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_AEX_EXT.2.1 The TSF shall be able to enforce read, write, and execute permissions on every page of physical memory.

FPT_AEX_EXT.2.2 The TSF shall prevent write and execute permissions from being simultaneously granted to any page of physical memory [[*unless a special permission has been granted to the process*]].

6.2.6.3 FPT_AEX_EXT.3 Anti-exploitation services (Overflow protection)

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_AEX_EXT.3.1 TSF processes that execute in a non-privileged execution domain on the application processor shall implement stack-based buffer overflow protection.

6.2.6.4 FPT_AEX_EXT.4 Anti-exploitation services (Domain isolation)

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_AEX_EXT.4.1 The TSF shall protect itself from modification by untrusted subjects.

FPT_AEX_EXT.4.2 The TSF shall enforce isolation of address space between applications.

6.2.6.5 FPT_BBD_EXT.1 Application processor mediation

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_BBD_EXT.1.1 The TSF shall prevent code executing on any baseband processor (BP) from accessing application processor (AP) resources except when mediated by the AP.

6.2.6.6 FPT_KST_EXT.1 Key storage

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_KST_EXT.1.1 The TSF shall not store any plaintext key material in readable non-volatile memory.

6.2.6.7 FPT_KST_EXT.2 No key transmission

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_KST_EXT.2.1 The TSF shall not transmit any plaintext key material outside the security boundary of the TOE.

6.2.6.8 FPT_KST_EXT.3 No plaintext key export

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_KST_EXT.3.1 The TSF shall ensure it is not possible for the TOE user(s) to export plaintext keys.

6.2.6.9 FPT_NOT_EXT.1 Self-test notification

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_NOT_EXT.1.1 The TSF shall transition to non-operational mode and [no other actions] when the following types of failures occur:

- failures of the self-test(s)
- TSF software integrity verification failures
- [no other failures].

6.2.6.10 FPT_STM.1 Reliable time stamps

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps **for its own use**.

6.2.6.11 FPT_TST_EXT.1 TSF cryptographic functionality testing

Hierarchical to: No other components.

Dependencies: No dependencies.

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

6.2.6.12 FPT_TST_EXT.2 Integrity testing

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_TST_EXT.2.1 The TSF shall verify the integrity of the bootchain up through the Application Processor OS kernel, and [no other executable code], stored in mutable media prior to its execution through the use of [a hardware-protected hash].

FPT_TST_EXT.2.2 The TSF shall not execute code if the code signing certificate is deemed invalid.

6.2.6.13 FPT_TUD_EXT.1 Trusted update: TSF version query

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_TUD_EXT.1.1 The TSF shall provide authorized users the ability to query the current version of the TOE firmware/software.

FPT_TUD_EXT.1.2 The TSF shall provide authorized users the ability to query the current version of the hardware model of the device.

FPT_TUD_EXT.1.3 The TSF shall provide authorized users the ability to query the current version of installed mobile applications.

6.2.6.14 FPT_TUD_EXT.2 Trusted update verification

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_TUD_EXT.2.1 The TSF shall verify software updates to the Application Processor system software and [no other processor system software] using a digital signature by the manufacturer prior to installing those updates.

FPT_TUD_EXT.2.2 The TSF shall [update only by verified software] the TSF boot integrity [hash].

FPT_TUD_EXT.2.3 The TSF shall verify that the digital signature verification key used for TSF updates [matches a hardware-protected public key].

FPT_TUD_EXT.2.4 The TSF shall verify mobile application software using a digital signature mechanism prior to installation.

6.2.7 TOE Access (FTA)

6.2.7.1 FTA_SSL_EXT.1 TSF- and user-initiated locked state

Hierarchical to: No other components.

Dependencies: No dependencies.

FTA_SSL_EXT.1.1 The TSF shall transition to a locked state after a time interval of inactivity.

FTA_SSL_EXT.1.2 The TSF shall transition to a locked state after initiation by either the user or the administrator.

FTA_SSL_EXT.1.3 The TSF shall, upon transitioning to the locked state, perform the following operations:

a) clearing or overwriting display devices, obscuring the previous contents;

b) [*clearing all plaintext KEKs and DEKs from volatile memory*].

6.2.7.2 FTA_TAB.1 Default TOE access banners

Hierarchical to: No other components.

Dependencies: No dependencies.

FTA_TAB.1.1 Before establishing a user session, the TSF shall display an advisory warning message regarding unauthorised use of the TOE.

6.2.7.3 FTA_WSE_EXT.1 Wireless network access

Hierarchical to: No other components.

Dependencies: No dependencies.

FTA_WSE_EXT.1.1 The TSF shall be able to attempt connections to wireless networks specified as acceptable networks as configured by the administrator in FMT_SMF_EXT.1.

6.2.8 Trusted Path/Channels (FTP)

6.2.8.1 FTP_ITC_EXT.1 Trusted channel communication[1]

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_ITC_EXT.1.1 The TSF shall use 802.11-2012, 802.1X, and EAP-TLS and [IPsec, TLS] to provide a communication channel between itself and another trusted IT product 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.

FTP_ITC_EXT.1.2 The TSF shall permit the TSF and the MDM Server and [no other IT entities] to initiate communication via the trusted channel.

FTP_ITC_EXT.1.3 The TSF shall initiate communication via the trusted channel for wireless access point connections, administrative communication, configured enterprise connections, and [OTA updates].

[1] In accordance with the Extended Package for Mobile Device Management Agents, this SFR replaces FTP_ITC_EXT.1 from the Protection Profile for Mobile Device Fundamentals v2.0.

6.3 SECURITY FUNCTIONAL REQUIREMENTS RATIONALE

The following table provides a mapping between the SFRs and Security Objectives:

	O.AUTH	O.COMMS	O.CONFIG	O.INTEGRITY	O.STORAGE	O.DATA_PROTECTION_TRANSIT	O.ACCOUNTABILITY	O.APPLY_POLICY
FAU_ALT_EXT.2							X	
FCS_CKM.1(1)		X						
FCS_CKM.1(2)		X						
FCS_CKM.2(1)	X	X						
FCS_CKM.2(2)		X						
FCS_CKM_EXT.1					X			
FCS_CKM_EXT.2					X			
FCS_CKM_EXT.3					X			
FCS_CKM_EXT.4					X			
FCS_CKM_EXT.5					X			
FCS_CKM_EXT.6					X			
FCS_COP.1(1)		X			X			
FCS_COP.1(2)		X		X	X			
FCS_COP.1(3)		X		X				
FCS_COP.1(4)		X						

	O.AUTH	O.COMMS	O.CONFIG	O.INTEGRITY	O.STORAGE	O.DATA_PROTECTION_TRANSIT	O.ACCOUNTABILITY	O.APPLY_POLICY
FCS_COP.1(5)		X						
FCS_HTTPS_EXT.1		X						
FCS_IV_EXT.1					X			
FCS_RBG_EXT.1		X			X			
FCS_SRV_EXT.1		X						
FCS_STG_EXT.1					X			
FCS_STG_EXT.2					X			
FCS_STG_EXT.3					X			
FCS_STG_EXT.4						X		
FCS_TLSC_EXT.1		X				X		
FCS_TLSC_EXT.2		X						
FDP_ACF_EXT.1				X				
FDP_DAR_EXT.1					X			
FDP_IFC_EXT.1		X						
FDP_STG_EXT.1		X						
FDP_UPC_EXT.1		X						
FIA_AFL_EXT.1	X							
FIA_BLT_EXT.1	X	X						

	O.AUTH	O.COMMS	O.CONFIG	O.INTEGRITY	O.STORAGE	O.DATA_PROTECTION_TRANSIT	O.ACCOUNTABILITY	O.APPLY_POLICY
FIA_BLT_EXT.2	X	X						
FIA_ENR_EXT.1						X		
FIA_PAE_EXT.1		X						
FIA_PMG_EXT.1	X							
FIA_TRT_EXT.1	X							
FIA_UAU.7	X							
FIA_UAU_EXT.1	X				X			
FIA_UAU_EXT.2	X							
FIA_UAU_EXT.3	X							
FIA_X509_EXT.1		X						
FIA_X509_EXT.2	X	X						
FIA_X509_EXT.3		X						
FMT_MOF_EXT.1			X					
FMT_POL_EXT.2								X
FMT_SMF_EXT.1			X					
FMT_SMF_EXT.2			X					
FMT_SMF_EXT.3								X
FMT_UNR_EXT.1								X

	O.AUTH	O.COMMS	O.CONFIG	O.INTEGRITY	O.STORAGE	O.DATA_PROTECTION_TRANSIT	O.ACCOUNTABILITY	O.APPLY_POLICY
FPT_AEX_EXT.1				X				
FPT_AEX_EXT.2				X				
FPT_AEX_EXT.3				X				
FPT_AEX_EXT.4				X				
FPT_BBD_EXT.1				X				
FPT_KST_EXT.1					X			
FPT_KST_EXT.2					X			
FPT_KST_EXT.3					X			
FPT_NOT_EXT.1				X				
FPT_STM.1				X				
FPT_TST_EXT.1				X				
FPT_TST_EXT.2				X				
FPT_TUD_EXT.1				X				
FPT_TUD _EXT.2				X				
FTA_SSL_EXT.1	X							
FTA_TAB.1			X					
FTA_WSE_EXT.1		X						
FTP_ITC_EXT.1		X						

Table 14 – Mapping of SFRs to Security Objectives

6.4 DEPENDENCY RATIONALE

Table 15 identifies the Security Functional Requirements from Part 2 of the CC and their associated dependencies. It also indicates whether the ST explicitly addresses each dependency.

SFR	Dependency	Dependency Satisfied	Rationale
FAU_ALT_EXT.2	None	N/A	
FCS_CKM.1(1)	FCS_CKM.2 or FCS_COP.1	✓	Satisfied by FCS_COP.1(3)
	FCS_CKM.4	✓	Satisfied by FCS_CKM_EXT.4
FCS_CKM.1(2)	FCS_CKM.2 or FCS_COP.1	✓	Satisfied by FCS_COP.1(1)
	FCS_CKM.4		Satisfied by FCS_CKM_EXT.4
FCS_CKM.2(1)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	✓	Satisfied by FCS_CKM.1(1)
	FCS_CKM.4	✓	Satisfied by FCS_CKM_EXT.4
FCS_CKM.2(2)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	✓	Satisfied by FCS_CKM.1(2)
	FCS_CKM.4	✓	Satisfied by FCS_CKM_EXT.4
FCS_CKM_EXT.1	None	N/A	
FCS_CKM_EXT.2	None	N/A	
FCS_CKM_EXT.3	None	N/A	
FCS_CKM_EXT.4	None	N/A	
FCS_CKM_EXT.5	None	N/A	
FCS_CKM_EXT.6	None	N/A	
FCS_COP.1(1)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	✓	Satisfied by FCS_CKM.1(2)
	FCS_CKM.4	✓	Satisfied by FCS_CKM_EXT.4
FCS_COP.1(2)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	x	The dependency is no longer applicable due to the refinements made to the SFR.
	FCS_CKM.4	✓	Satisfied by FCS_CKM_EXT.4

SFR	Dependency	Dependency Satisfied	Rationale
FCS_COP.1(3)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	✓	Satisfied by FCS_CKM.1(1)
	FCS_CKM.4	✓	Satisfied by FCS_CKM_EXT.4
FCS_COP.1(4)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	✓	Satisfied by FCS_CKM.1(2)
	FCS_CKM.4	✓	Satisfied by FCS_CKM_EXT.4
FCS_COP.1(5)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	✓	Satisfied by FCS_CKM.1(2)
	FCS_CKM.4	✓	Satisfied by FCS_CKM_EXT.4
FCS_HTTPS_EXT.1	None	N/A	
FCS_IV_EXT.1	None	N/A	
FCS_RBG_EXT.1	None	N/A	
FCS_SRV_EXT.1	None	N/A	
FCS_STG_EXT.1	None	N/A	
FCS_STG_EXT.2	None	N/A	
FCS_STG_EXT.3	None	N/A	
FCS_STG_EXT.4	None	N/A	
FCS_TLSC_EXT.1	None	N/A	
FCS_TLSC_EXT.2	None	N/A	
FDP_ACF_EXT.1	None	N/A	
FDP_DAR_EXT.1	None	N/A	
FDP_IFC_EXT.1	None	N/A	
FDP_STG_EXT.1	None	N/A	
FDP_UPC_EXT.1	None	N/A	
FIA_BLT_EXT.1	None	N/A	
FIA_BLT_EXT.2	None	N/A	
FIA_ENR_EXT.1	None	N/A	

SFR	Dependency	Dependency Satisfied	Rationale
FIA_PAE_EXT.1	None	N/A	
FIA_PMG_EXT.1	None	N/A	
FIA_TRT_EXT.1	None	N/A	
FIA_UAU.7	FIA_UAU.1	✓	Satisfied by FIA_UAU_EXT.2
FIA_UAU_EXT.1	None	N/A	
FIA_UAU_EXT.2	None	N/A	
FIA_UAU_EXT.3	None	N/A	
FIA_X509_EXT.1	None	N/A	
FIA_X509_EXT.2	None	N/A	
FIA_X509_EXT.3	None	N/A	
FMT_MOF_EXT.1	None	N/A	
FMT_POL_EXT.2	None	N/A	
FMT_SMF_EXT.1	None	N/A	
FMT_SMF_EXT.2	None	N/A	
FMT_SMF_EXT.3	None	N/A	
FMT_UNR_EXT.1	None	N/A	
FPT_AEX_EXT.1	None	N/A	
FPT_AEX_EXT.2	None	N/A	
FPT_AEX_EXT.3	None	N/A	
FPT_AEX_EXT.4	None	N/A	
FPT_BBD_EXT.1	None	N/A	
FPT_KST_EXT.1	None	N/A	
FPT_KST_EXT.2	None	N/A	
FPT_KST_EXT.3	None	N/A	
FPT_NOT_EXT.1	None	N/A	
FPT_STM.1	None	N/A	
FPT_TST_EXT.1	None	N/A	
FPT_TST_EXT.2	None	N/A	

SFR	Dependency	Dependency Satisfied	Rationale
FPT_TUD_EXT.1	None	N/A	
FPT_TUD_EXT.2	None	N/A	
FTA_SSL_EXT.1	None	N/A	
FTA_TAB.1	None	N/A	
FTA_WSE_EXT.1	None	N/A	
FTP_ITC_EXT.1	None	N/A	

Table 15 – Functional Requirement Dependencies

6.5 TOE SECURITY ASSURANCE REQUIREMENTS

The TOE assurance requirements for this ST consist of the requirements specified in the Protection Profile for Mobile Device Fundamentals and the Extended Package for Mobile Device Management Agents.

The assurance requirements are summarized in Table 16.

Assurance Class	Assurance Components	
	Identifier	Name
Development (ADV)	ADV_FSP.1	Basic functional specification
Guidance Documents (AGD)	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
Life-Cycle Support (ALC)	ALC_CMC.1	Labelling of the TOE
	ALC_CMS.1	TOE CM coverage
	ALC_TSU_EXT.1	Timely security updates
Security Target Evaluation (ASE)	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST introduction
	ASE_OBJ.1	Security objectives for the operational environment
	ASE_REQ.1	Stated security requirements

Assurance Class	Assurance Components	
	Identifier	Name
	ASE_TSS.1	TOE summary specification
Tests (ATE)	ATE_IND.1	Independent testing - conformance
Vulnerability Assessment (AVA)	AVA_VAN.1	Vulnerability survey

Table 16 – Security Assurance Requirements

7 TOE SUMMARY SPECIFICATION

This section provides a description of the security functions and assurance measures of the TOE that meet the TOE security requirements.

7.1 SECURITY AUDIT

7.1.1 Agent alerts

Policy updates are received from the MDM Server as described in the BlackBerry Enterprise Service 12.5 Security Target.

The MDM Agent sends a notice to the MDM Server when there is a change in enrollment status, or there is a failure to apply policies to a mobile device. When the Agent does not respond to a reachability request, the MDM Server considers the Agent to be unreachable.

Reachability is determined by the MDM Server by polling the device. Devices are polled every minute, every 15 minutes or every eight hours depending upon the connection conditions.

When the device has lost connectivity, data lock queue files are used to store the audit events. An alert is generated when storage is 85% and 95% full. The agent allows only emergency calls when the storage is exhausted.

TOE Security Functional Requirements addressed: FAU_ALT_EXT.2.

7.2 CRYPTOGRAPHIC SUPPORT

BlackBerry 10 devices use the BlackBerry Cryptographic Library Version 5.6.2, CMVP certificate number 1578.

7.2.1 Cryptographic Key Generation

The key generation functionality for asymmetric keys used for key establishment is invoked through use of the TLS protocol.

ECC keys are generated in accordance with FIPS 186-4, Appendix B.4. The Master DLQ keypair and Data Anchor keypair are used together in ECDH key derivation to generate initial keying material (shared secret) for a given anchor point. The supported key sizes are 256, 384 and 521 bits.

Key generation may also be invoked as part of the SCEP enrollment process.

The TOE cryptographic module (CMVP certificate number 1578) supports the following algorithm:

Requirement	Algorithm	CAVP Certificate
FCS_CKM.1(1)	ECC key generation	199

Table 17 – Key Generation CAVP Certificate Number

TOE Security Functional Requirements addressed: FCS_CKM.1(1).

7.2.2 Cryptographic Key Generation for WLAN

Keys for Wireless LAN access are generated in accordance with IEEE 802.11-2012.

The device derives the PTK from the 4-way handshake as described in Clause 11.6.6.4 of IEEE 802.11-2012. The device makes use of the CTR with CBC-MAC Protocol (CCMP) to protect the link with the Access Point. This protocol is defined in clause 11.4.3 of IEEE 802.11-2012. The device also supports Protected Management frames, if enabled on the AP infrastructure. Protected management frames are described in clause 11.5.17 of IEEE 802.11-2012.

The TOE is Wi-Fi Certified from Wi-Fi® Alliance and is tested for interoperability against the 10 leading market APs, and is beta tested against BlackBerry's WPA2-Enterprise network infrastructure. See Table 18 for certification details.

Device	Model	WiFi Alliance Certificate	Certification Details
Classic	SQC100-1	WFA57100	Connectivity: Wi-Fi CERTIFIED™ a,b,g,n Wi-Fi Direct® WPA™ – Enterprise, Personal WPA2™ – Enterprise, Personal Optimization:WMM®, WMM®-Power Save Access: Wi-Fi Protected Setup™ Application and Service: Miracast® - Source, Voice-Personal
	SQC100-2	WFA59936	
	SQC100-3	WFA56510	
	SQC100-4	WFA59939	
	SQC100-5	WFA56510	
Passport	SQW100-1	WFA55078	Connectivity: Wi-Fi CERTIFIED™ a,b,g,n,ac Wi-Fi Direct® WPA™ – Enterprise, Personal Optimization:WMM®, WMM®-Power Save Access: Wi-Fi Protected Setup™ Application and Service: Miracast® - Source, Voice-Personal
	SQW100-3	WFA59935	
	SQW100-4	WFA60544	
Leap	STR100-1	WFA59167	Connectivity: Wi-Fi CERTIFIED™ b,g,n Wi-Fi Direct® Optimization:WMM®, WMM®-Power Save
	STR100-2	WFA59168	

Device	Model	WiFi Alliance Certificate	Certification Details
			Access: Wi-Fi Protected Setup™ Application and Service: Miracast® - Source, Voice-Personal
Z30	STA100-2	WFA59941	Connectivity: Wi-Fi CERTIFIED™ a,b,g,n Wi-Fi Direct® WPA™ – Enterprise, Personal WPA2™ – Enterprise, Personal Optimization:WMM®, WMM®-Power Save Access: Wi-Fi Protected Setup™ Application and Service: Miracast® - Source, Voice-Personal
	STA100-3	WFA20438	
	STA100-5	WFA59943	
	STA100-6	WFA20417	
Q10 - Porche	SQK100-1	WFA60165	Connectivity: Wi-Fi CERTIFIED™ a,b,g,n Wi-Fi Direct® WPA™ – Enterprise, Personal WPA2™ – Enterprise, Personal Optimization:WMM®, WMM®-Power Save Access: Wi-Fi Protected Setup™ Application and Service: Miracast® - Source, Voice-Personal
	SQK100-2	WFA60166	
Q10	SQN100-1	WFA60143	Connectivity: Wi-Fi CERTIFIED™ a,b,g,n Wi-Fi Direct® WPA™ – Enterprise, Personal WPA2™ – Enterprise, Personal Optimization:WMM®, WMM®-Power Save Access: Wi-Fi Protected Setup™ Application and Service: Miracast® - Source, Voice-Personal
	SQN100-2	WFA60161	
	SQN100-3	WFA60162	
	SQN100-4	WFA60164	
	SQN100-5	WFA60163	
Z10 - Porche	STK100-1	WFA20658	Connectivity: Wi-Fi CERTIFIED™ a,b,g,n WPA™ – Enterprise, Personal
	STK100-2	WFA20660	

Device	Model	WiFi Alliance Certificate	Certification Details
			WPA2™ – Enterprise, Personal Optimization:WMM®, WMM®-Power Save Access: Wi-Fi Protected Setup™ Application and Service: Miracast® - Source, Voice-Personal
Z10	STL100-2	WFA19315	Connectivity: Wi-Fi CERTIFIED™ a,b,g,n WPA™ – Enterprise, Personal WPA2™ – Enterprise, Personal Optimization:WMM®, WMM®-Power Save Access: Wi-Fi Protected Setup™ Application and Service: Miracast® - Source, Voice-Personal
	STL100-3	WFA15346	
	STL100-4	WFA16634	

Table 18 – WiFi Certification Details

TOE Security Functional Requirements addressed: FCS_CKM.1(2).

7.2.3 Cryptographic Key Establishment

The key establishment functionality for asymmetric keys used for key establishment is invoked through use of the TLS protocol. The RSA OAEP decryption function in the Security Builder GSE-C returns only one error code (SB_FAIL_PKCS1_DECRYPT) on decryption failure. There are no timing variations.

RSA key agreement is supported in accordance with NIST Special Publication 800-56B for use with the TLS protocol. The supported key sizes are 2048 bits. The TOE acts as a client for RSA-based key establishment schemes.

ECC keys are generated in accordance NIST Special Publication 800-56A for use with the TLS protocol. The supported key sizes are 521 bits. The TOE acts as both a sender and a recipient for RSA-based key establishment schemes.

The TOE cryptographic module (CMVP certificate number 1578) supports the following algorithms:

Requirement	Algorithm	CAVP Certificate

Requirement	Algorithm	CAVP Certificate
FCS_CKM.2(1)	RSA-based key establishment	Vendor affirmed
	Elliptic curve-based key establishment	13

Table 19 – Key Establishment CAVP Certificate Numbers

TOE Security Functional Requirements addressed: FCS_CKM.2(1).

7.2.4 Cryptographic Key Distribution (WLAN)

The TOE decrypts the Group Temporal Key (GTK) in accordance with the AES Key Wrap in an EAPOL-frame that meets NIST SP 800-38F, and IEEE 802.11-2012 for packet format and timing considerations.

The TOE adheres to RFC 3394, SP 800-38F, and 802.11-2012 standards and unwraps the GTK, when sent encrypted with the WPA2 KEK using AES Key Wrap in an EAPOL-Key frame. The TOE, upon receiving an EAPOL frame, will subject the frame to a number of checks (i.e. frame length, EAPOL version, frame payload size, EAPOL-Key type, key data length, EAPOL-Key CCMP descriptor version, and replay counter) to ensure that it is a proper EAPOL message, and then decrypts the GTK using the KEK, thus ensuring that it does not expose the GTK.

TOE Security Functional Requirements addressed: FCS_CKM.2(2).

7.2.5 Cryptographic Key Support (REK)

The Root Encryption Key (REK) of the TOE is a 256-bit symmetric key that is generated using a DRBG and embedded in the application processor on device bootup. The REK is protected such that it cannot be accessed by the BlackBerry OS or any application. The one time programmable fuses where the REK is stored may only be accessed via the TrustZone. The REK is used only for key derivation.

The RBG used to generate the REK uses a DRBG and is seeded with a minimum of 256 bits of entropy.

TOE Security Functional Requirements addressed: FCS_CKM_EXT.1.

7.2.6 Cryptographic Key Random Generation

The Data Encryption Keys (DEKs) are AES keys used to encrypt/decrypt files in the TOE. They are generated using the DRBG implementation in the BlackBerry OS Cryptographic Library with entropy to provide a security strength of 256 bits.

TOE Security Functional Requirements addressed: FCS_CKM_EXT.2.

7.2.7 Cryptographic Key Generation

All Key Encryption Keys (KEKs) in the TOE provide a security strength of 256 bits. KEKs that are encrypted by another KEK and stored in the TOE are generated with the DRBG implementation in the BlackBerry OS Cryptographic Kernel. KEKs that are not stored in the TOE are derived using a Password Authentication Factor and combined with another KEK using a Key Derivation Function (KDF).

The PBKDF used to derive KEKs is PBKDF2(HMAC-SHA-256) defined in PKCS5v2. A 128 bit salt is generated by a DRBG and stored in the fsec.

TOE Security Functional Requirements addressed: FCS_CKM_EXT.3.

7.2.8 Cryptographic Key Destruction

Cryptographic key materials in the TOE are destroyed when they are no longer needed by the TSF. The plaintext keys in the RAM are deleted with an overwrite with zeros. Persistent secrets are destroyed using a block erase on wear-leveled flash.

TOE Security Functional Requirements addressed: FCS_CKM_EXT.4.

7.2.9 TSF Wipe

To affect a data wipe of the TSF, the perimeterMgr triggers the File System to erase the domain key associated with each encryption domain in the work space. Zeroizing the Domain Key (DK) of each encryption domain renders all data within the encryption domain inaccessible and marks the blocks storing each file as unused. All unused blocks in the workspace are then block erased.

Flash memory is wiped using a block erase.

TOE Security Functional Requirements addressed: FCS_CKM_EXT.5.

7.2.10 Salt Generation

All salts used in the TOE are generated using the DRBG function implemented in the BlackBerry OS Cryptographic Kernel. Salts are used only in the derivation of AES keys.

Salts are required for symmetric key derivation to create the Derived Domain System Keys (DDSK), the Domain Master Key (DMK), and for the derivation of the Workspace Password Hash from the Workspace Password.

The following table describes the use of salts in the TOE:

Key	Salt Generation
Domain Master Key (DMK)	28 bit salt for PBKDF is generated from DRBG using SB function hu_RngGetBytes
Workspace Password	64 bit salt for hashing the password is generated from DRBG

Key	Salt Generation
hash	using SB function hu_RngGetBytes

Table 20 – Use of Salts

TOE Security Functional Requirements addressed: FCS_CKM_EXT.6.

7.2.11 Cryptographic Operation

The TOE cryptographic module (CMVP certificate number 1578) supports the following algorithms:

Requirement	Algorithm	CAVP Certificate
FCS_COP.1(1)	AES (AES-CBC, AES-CCMP, AES-GCM, AES-CCM, AES-KW and AES-XTS)	1608
FCS_COP.1(2)	SHA-1, SHA-256, SHA-384, SHA-512	1421 3404
FCS_COP.1(3)	RSA	790
FCS_COP.1(3)	ECDSA	199
FCS_COP.1(4)	HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384	944 2707

Table 21 – Cryptographic Operations

7.2.12 Encryption / Decryption

The TOE supports AES-CBC, AES-CCMP, AES-GCM, AES-CCM and AES-XTS. CCMP encryption is performed in hardware.

TOE Security Functional Requirements addressed: FCS_COP.1(1).

7.2.13 Cryptographic Hashing

SHA-1 may be used with the TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite. SHA-256 is supported and may be used with an application that requires it. SHA-384 is used with the TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 ciphersuite. This ciphersuite is used to protect communications between the TOE device and the BES MDM Server. Cryptographic hashing is used in signature generation and verification, HMAC calculation and key derivation. SHA1 is not used in the generation of digital signatures.

There is a second secure hash algorithm implementation specifically for generation of the REK using CAVP certificate number 3404.

TOE Security Functional Requirements addressed: FCS_COP.1(2).

7.2.14 Cryptographic Operation

The TOE implements RSA and ECDSA for signature generation and verification.

TOE Security Functional Requirements addressed: FCS_COP.1(3).

7.2.15 Keyed Hash Algorithms

HMAC-SHA-1, HMAC-SHA-256 and HMAC-SHA-384 are supported with key sizes of 160, 256 and 384 bits and message digest sizes of 160, 256 and 384 bits. HMAC-SHA-256 is used for the PBKDF. SHA512 is used to calculate the password hash.

There is a second keyed hash algorithm implementation specifically for generation of the REK using CAVP certificate number 2707.

TOE Security Functional Requirements addressed: FCS_ COP.1(4).

7.2.16 HTTPS Protocol

The TSF implements the HTTPS Protocol in accordance with FCS_TLSC_EXT.2.

TOE Security Functional Requirements addressed: FCS_HTTPS_EXT.1.

7.2.17 Initialization Vector Generation

The TSF generates initialization vectors for the encryption of data at rest which conform to SP 800-38A for AES-CBC encryption. The IVs for FEKs are generated by encrypting the physical block address of the record with DK.

The TSF generates initialization vectors for key encryption which conform to NISP SP 800-38A for AES-CBC encryption. The IVs for KEKs are generated by encrypting the physical block address of the record with the higher level KEK.

The TSF generates initialization vectors for TLS which conform to SP 800-38D for AES-GCM.

TOE Security Functional Requirements addressed: FCS_IV_EXT.1.

7.2.18 Random Bit Generation

The BlackBerry OS Cryptographic Kernel implements HMAC_DRBG in accordance with SP 800-90 and DRBG certificate #81. The DRBG implementations are seeded by entropy with sources from software and hardware based noise sources with a minimum of 256 bits of entropy. Applications may request and be provided with random bits from these DRBG implementations.

Additionally, there is a HMAC_DRBG implementation in TrustZone which adheres to the guidance in SP 800-90. This DRBG implementation is seeded by entropy with a hardware based TRNG with a minimum of 256 bits of entropy. This is only used for generation of the REK and uses CAVP certificate number 1252.

Secure key storage is implemented in software storage located in encryption domains.

- All keying material stored in an encryption domain is protected by a KEK that chains back to the REK.
- All keying material stored in the operation domain and the lock domain is protected by a password and a KEK that chains back to the REK.

The File System manages all encryption for data written to encryption domains in the workspace. DEKs are 256 bits and AES encryption is in CBC mode.

TOE Security Functional Requirements addressed: FCS_RBG_EXT.1.

7.2.19 Cryptographic Algorithm Services

All of the cryptographic operations noted in FCS_SRV_EXT.1 and implemented in the BlackBerry OS Cryptographic Library are made accessible to applications that request them.

TOE Security Functional Requirements addressed: FCS_SRV_EXT.1.

7.2.20 Cryptographic Key Storage

Secure key storage is implemented in software storage located in encryption domains that provide the required protection specified in FCS_STG_EXT.2. All keying material stored in an encryption domain is protected by a KEK that chains back to the REK. All keying material stored in the operation domain and the lock domain is protected by a password and a KEK that chains back to the REK. These protection relationships are described in the key hierarchy. The File System manages encryption operations for data written to encryption domains in the workspace.

TOE Security Functional Requirements addressed: FCS_STG_EXT.1.

7.2.21 Encrypted Cryptographic Key Storage

Data Encryption Keys (DEKs), Key Encrypting Keys (KEKs) and all software based keys are encrypted by KEKs that chain to a password derived KEK and the REK. All DEKs and KEKs are encrypted using AES-256 encryption using the CBC mode.

TOE Security Functional Requirements addressed: FCS_STG_EXT.2.

7.2.22 Integrity of Encrypted Key Storage

The integrity of all encrypted KEKs is protected using HMAC-SHA-256 keyed hash algorithm. The integrity of a KEK is verified before a key is used.

TOE Security Functional Requirements addressed: FCS_STG_EXT.3.

7.2.23 EAP-TLS Protocol

The TOE implements TLS versions 1.0, 1.1 and 1.2 that support the following ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

The configuration used for EAP-TLS is specified when creating the Wi-Fi profile. The signature_algorithm extension is automatically configured when the TOE is operated with the CC-required security policies.

TOE Security Functional Requirements addressed: FCS_TLSC_EXT.1.

7.2.24 TLS Protocol

The TOE implements TLS version 1.2 that supports the following ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

When an administrator selects the policies required to enforce the Common Criteria requirements, including advanced data at rest protection, the BES MDM Server will communicate with the BlackBerry device using only TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384.

TLS_RSA_WITH_AES_128_CBC_SHA to communicate to their platforms. In this case, it is up to the application developer to call the appropriate API. This ciphersuite may also be used to configure a Wi-Fi profile.

The TSF supports mutual authentication using X.509v3 certificates and does not establish a trusted channel if the peer certificate is invalid. The TSF supports identifier verification as per RFC 6125 using DNS-ID and CN-ID and presents Elliptic Curves extensions in the Client Hello. Wildcards are supported for Wi-Fi, PIM and EMA. For VPN, reference identifiers containing wildcards are not supported, but certificates containing wildcards may be validated against a reference identifier that does not contain a wildcard. For Wi-Fi, domain name/alternate domain name information is not passed to the Wi-Fi component. IP addresses and certificate pinning options are not supported.

The TSF supports mutual authentication using X.509v3 certificates and does not establish a trusted channel if the peer certificate is invalid.

TOE Security Functional Requirements addressed: FCS_TLSC_EXT.2.

7.3 USER DATA PROTECTION

7.3.1 Security Access Control

The TOE controls access to limit the system services available for use by applications. This applies to the following services:

- camera
- microphone
- GPS service
- credential service
- contact list
- stored photographs

- stored text messages
- stored emails
- device identifier information
- network access

An application is identified uniquely using:

a. Package-Name (e.g. com.blackberry.xyz); and
b. Package-Id, a fixed length string derived from the Package-Name value and signing key of the application

Together, these are known as the App Key. Some App Key domains, such as sys.*, are reserved for internal device services and may not be used by external third party applications. It should be noted that each internal system service also has a unique Package-Name, but no Package-Id value. The service App Key is the Package-Name which is under the reserved sys.* namespace.

A permission is enforced with a set of rules that include deny, allow and similar access clauses. These rules can be found on any BB10 device under the following files:

/etc/authman/sys.acl: Privileges granted if the rule is allowed

/etc/authman/sys.res: Which App Key can get this permission

Carrier specific application rules are found under:

/etc/authman/subsystem/NNN.acl

where NNN is the carrier unique ID known at boot time. These extra rules are loaded on top of the system configuration for a carrier configured device.

Each allow or deny clause uses the App Key as the trigger. Wildcards (*) can be used in the names to target a subset of apps. For example, sys.* may be used for all system privileged applications. All clauses under a rule are evaluated and the most restrictive match prevails.

The rule files are embedded into the BBOS filesystem partition and cannot be altered. Any updates to these files require an over-the-air (OTA) update.

When an application is launched, a cryptographic hash verification of the application contents is checked to verify that the application has not been altered through tampering. The permissions requested by the application are evaluated against the permission configuration.

- A permission without rules (e.g. access_internet) is wide open.
- A permission with an explicit allow clause matching the App Key is permitted without further approval from the end user.
- A permission without an explicit allow clause matching the App Key is denied.

- A permission with a prompt clause will cause a dialogue popup to appear when the application is launched to allow the user to determine if the permissions will be granted to the application.
    - After the initial selection and agreement by the end user, this information is cached and the prompts do not appear in subsequent application launches.
    - The end user can access application permissions under 'Settings' and change these choices at any time after the first launch approval of permissions.

Additional other filtering criteria may be used in the clauses to account for a variety of configuration options, such as those for personal and enterprise use. For third party privileged applications and privileged carrier applications that require restricted permissions, the App Key is included in the configuration rule file.

Applications are not permitted to share data amongst themselves; their sandboxes are completely isolated from one another. Applications may only share data with some of the system services configured for this functionality: bbm, bluetooth, media, mmsynclite (media indexer), pim (messaging), search and wifidirect.

All services have a master folder with permissions "apps:<service GID> rwxr-s--x", and any files or folders created under this folder inherits the '<service GID>' group. This allows the service to be able to read everything that is found under its folder umbrella. For example:

/accounts/1000/sharewith/<service_name>

When an application is launched, its sandbox folder structure is created. This adds a folder under each of the services' sharewith folders with the permissions 'apps:<service GID> rwxr-s---'. An ACL of 'rwx' is added to this folder based on the GID of the application. The ACL is re-applied at each first launch of the application after a reboot. For example:

/accounts/1000/sharewith/<service_name>/<app key>

The first time that an application is launched, or following the first launch after a reboot, a symlink is created in the application's sandbox. This symlink points to the path that allows it to create and manage files in the services sharewith app folder, thereby allowing the application to share information with the service. For example:

/accounts/1000/appdata/<app key>/sharewith/<service_name>

Any file created under this folder will have the UID of the application and the GID of the service. No sharing is permitted unless all of these conditions are met, thereby preventing any disallowed sharing to take place.

TOE Security Functional Requirements addressed: FDP_ACF_EXT.1.

7.3.2 Data-at-rest Protection

The TOE implements a data-at-rest (DAR) protection system that encrypts all protected and sensitive data using Data Encryption Keys (DEKs) with AES-256 encryption using CBC mode.

All enterprise centric data is stored in the work space. All data in the workspace is persisted in encryption domains and therefore, by definition, is encrypted at rest. All data in the workspace is stored in encryption domains. This is enabled by default. All DEKs are 256 bits. The encryption performed by the File system in the workspace is AES encryption in CBC mode.

TOE Security Functional Requirements addressed: FDP_DAR_EXT.1.

7.3.3 Subset Information Flow Control

The TSF ensures that when an IPsec VPN connection is established, all IP traffic flows through the VPN client.

In the evaluated configuration, when native VPN is active, the TOE routes all IP traffic through the kernel's IPsec interface, with no split tunnel configured. The TOE VPN service also uses per packet filter rules to block IP traffic on the binding interface of the VPN tunnel. This ensures that all traffic (with the exception of Control Plane traffic) flows through the IPsec client. Control Plane traffic includes the IKE traffic required to establish the tunnel. This is the same for all baseband protocols.

TOE Security Functional Requirements addressed: FDP_IFC_EXT.1.

7.3.4 User Data Storage

The TOE implements a Trust Anchor Database that stores and protects certificates imported by the TOE user and administrator. The certificates in the Trust Anchor Database are assigned with POSIX ACL permissions that allow only applications with sufficient permissions to access them.

The Trust Anchor Database is a union of the certificate databases in the following locations:

- Factory Seeded: /var/certdb/global/certdb/nto/rim_trusted/certstore.db This database may be disabled via the ‘Allow use of pre-loaded trusted root certificates’ IT Policy.
- BES Managed databases:
    - EMA: /accounts/1000-enterprise/sysdata/certdb/_startup_data/nto/shared_ema_trusted/certstore.db
    - Browser&openssl: /accounts/1000-enterprise/sysdata/certdb/_startup_data/certmgr/web_ema_trusted/certstore.db

- VPN: /accounts/1000-enterprise/sysdata/certdb/_startup_data/ema/vpn_ema_trusted/certstore.db
- WIFI: /accounts/1000-enterprise/sysdata/certdb/_startup_data/ema/wifi_ema_trusted/certstore.db

All certificate stores are in encryption domains, which chain back to the REK. All areas of the workspace are also in encryption domains, which chain back to the REK. This meets the requirements for protected storage. The loading of certificates into the workspace Trust Anchor Database is controlled by the BES Administrator through the IT policies. The administrator may select which Trust Anchor Database into which a certificate may be loaded.

TOE Security Functional Requirements addressed: FDP_STG_EXT.1.

7.3.5 Inter-TSF User Data Transfer Protection

The TSF provides means for third-party applications to use TLS, HTTPS, and Bluetooth BR/EDR to establish a protected communications channel to another product in order to protect the user data.

TOE Security Functional Requirements addressed: FDP_UPC.1.

7.4 IDENTIFICATION AND AUTHENTICATION

7.4.1 Authentication Failure Handling

Both the TOE user and administrator are able to configure the number of unsuccessful attempts a user is allowed before device wipe to an integer value between 3 and 10. When this value is reached, a full wipe of the device is performed. The number of unsuccessful authentication attempts since the last successful authentication is maintained for each user for each authentication type. This record is maintained when the TOE is powered off, stored at /pps/system/perimeter/settings/1000-enterprise/settings. This is persisted across reboots.

Example: perimeter_curr_attempts::2

TOE Security Functional Requirements addressed: FIA_AFL_EXT.1.

7.4.2 Bluetooth User Authorization

At the commencement of the Bluetooth pairing process between the TOE and another device, the TSF enforces user authorization by manual input in order for the pairing process to complete. This is enforced for all Bluetooth pairing operations.

TOE Security Functional Requirements addressed: FIA_BLT_EXT.1.

7.4.3 Bluetooth Authentication

Service Discovery Protocol (SDP) is the only data transferred before the channel is authenticated. All other data is not transferred until the devices are mutually authenticated. This includes the cases where data is transferred using RFCOMM and L2CAP.

A secondary dialogue is presented following the completion of pairing. The user is required to indicate which profile the remote device must use for the connection. This dialogue also appears if the device attempts to connect to a profile that was not seen during pairing. This could occur if the device does not advertise all services at pairing. The profiles may be viewed and changed by the user through the Settings=>Bluetooth screen.

TOE Security Functional Requirements addressed: FIA_BLT_EXT.2.

7.4.4 Enrollment of Mobile Device into Management

During the enrollment process, a URL including the domain name or IP address of the MDM server, and the enterprise root management certificate are securely sent from the MDM Server to the MDM Agent. This data provides the reference identifier of the MDM Server.

TOE Security Functional Requirements addressed: FIA_ENR_EXT.2.

7.4.5 PAE Authentication

The TOE conforms to the IEEE 802.1X standard for Port Access Entity (PAE). The TOE can join 802.1X networks in the supplicant role.

TOE Security Functional Requirements addressed: FIA_PAE_EXT.1.

7.4.6 Password Management

The TOE supports passwords up to 32 characters long, comprising of uppercase letters, lowercase letters, numbers and special characters, including : "!", "@!", "#", "$", "%", "^", "&", "*", "(", ")". Rules may be added to enforce a minimum password length and complexity.

TOE Security Functional Requirements addressed: FIA_PMG_EXT.1.

7.4.7 Authentication Throttling

The TOE may be configured to prevent authentication using external hardware ports.

Authentication via external hardware ports is prevented using the USB IT policy options as follows:

- Using the 'Allow USB OTG mass storage' option, specify whether a user can use the USB OTG feature and connect USB mass storage devices (such as USB sticks) to a BlackBerry device. If this rule is not selected, the user cannot connect USB mass storage devices to the TOE.

- Using the ‘Allow computer to access device’ policy option, specify whether a computer can access content on a TOE device using a USB connection or the file-sharing option with a Wi-Fi connection. If this rule is not selected, the computer cannot access content on the device using a USB or Wi-Fi connection and the device can't share media content with DLNA Certified devices.

Additionally, the TOE performs a full wipe of the device after the maximum of ten unsuccessful authentication attempts. Therefore no more than 10 attempts may be performed in any time limit.

TOE Security Functional Requirements addressed: FIA_TRT_EXT.1.

7.4.8 Protected Authentication Feedback

User input during authentication attempts is obscured on the screen by dots. The characters are shown on the screen for a moment (less than a second) or until another character is input, after which it is obscured by a dot. This is the default behaviour.

TOE Security Functional Requirements addressed: FIA_UAU.7.

7.4.9 Enrollment of Mobile Device into Management

The user requires a username and activation password to enroll the device. The BES server is identified by the Server Routing Protocol (SRP) identifier which is held by the BlackBerry Infrastructure.

TOE Security Functional Requirements addressed: FIA_ENR_EXT.2.

7.4.10 Authentication for Cryptographic Operation

When the TOE boots up, all protected and sensitive data in the operation and lock domains are encrypted. When the user successfully authenticates with the TOE for the first time, the password is used to derive a KEK using PBKDF, which is then used to decrypt other KEKs and DEKs for decrypting data and keys in the operation and lock domain.

TOE Security Functional Requirements addressed: FIA_UAU_EXT.1.

7.4.11 Timing of Authentication

The TOE displays a number of unread messages and notifications, mobile network, Wi-Fi strength, and battery life indicator, and allows the user to make an emergency call, turn off the TOE, and access the camera before the user is authenticated. Note that the dialer is not available to the user. The user cannot dial the emergency number directly, but may select the ‘Emergency Call’ icon, and the call will be made for the user. All other actions require the user to be successfully authenticated to be performed.

TOE Security Functional Requirements addressed: FIA_UAU_EXT.2.

7.4.12 Re-authentication

The TOE requires the user to authenticate and unlock the device before the user can modify the password. In addition, during the password change, the user is prompted to confirm the current password before the new password is configured.

TOE Security Functional Requirements addressed: FIA_UAU_EXT.3.

7.4.13 Validation of Certificates

The TOE validates a certificate using the certificate path validation. The TOE only treats certificates as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE. The TOE can also be configured to validate the revocation status of the certificate by using the Online Certificate Status Protocol (OCSP).

The BB10 Certificate Manager performs the certificate path validation in the Trust Anchor Database using the certificate path validation algorithm, which performs the following checks:

1. Ensure the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates;
2. Use Online Certificate Status Protocol (OCSP) as specified in RFC 2560 to verify revocation status;
3. Validates 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, and
    - 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.

TOE Security Functional Requirements addressed: FIA_X509_EXT.1.

7.4.14 X509 Certificate Authentication

The TOE uses X.509v3 certificates for EAP-TLS, IPsec, and TLS authentication, including communications between the MDM Server and the MDM Agent. The TOE does not establish a secure connection if it deems the peer certificate to be invalid. If it cannot determine the certificate’s validity, the administrator may choose to accept or reject the certificate.

The IT Policy determines the certificate store into which a certificate is loaded. The application then selects the certificate to be used from its assigned certificate store. The actions taken when certificate validity cannot be determined are controlled by policy for VPN, determined by the user when the certificate is presented to a browser, and is user determined for S/MIME.

TOE Security Functional Requirements addressed: FIA_X509_EXT.2.

7.4.15 Request Validation of Certificates

The certificate validation service is available to applications running on the TOE. The validation service responds to the application with the result of the validation.

TOE Security Functional Requirements addressed: FIA_X509_EXT.3.

7.5 SECURITY MANAGEMENT

7.5.1 Management of Security Functions Behavior

The TSF provides the management functionality described in FMT_MOF.1 and Table 13. The following table provides any additional information required by the PP. Note that users in the role of Security Administrator and Enterprise Administrator have the ability to create, edit and delete policies. No other role has these permissions. Users in the Security Administrators, Enterprise Administrators and Senior HelpDesk roles may assign policies to users and groups.

Communication between the MDM Agent and the MDM Server are protected using TLS and mutual authentication. Certificates used for authentication of the MDM Agent are imported via the Certificate Manager.

The TOE does not include an administrator-accessible API for functions that may only be managed by users. Functions limited to administrators are not available through the user interface, or are restricted by IT Policy.

Management Function	Description
1. configure password policy: a. minimum password length b. minimum password complexity c. maximum password lifetime	The minimum password length may be set to 4 to 14 characters. (The maximum password length is 32 characters.) The possible options for BlackBerry password pattern checks are: • No restriction • At least 1 alpha and 1 numeric character • At least 1 alpha, 1 numeric, and 1 special character • At least 1 upper-case alpha, 1 lower-case alpha, 1 numeric, and 1 special character The maximum password age is set to ‘0’

Management Function	Description
	to cause passwords to never expire. The maximum value is 65535 days.
2. configure session locking policy: a. screen-lock enabled/disabled b. screen lock timeout c. number of authentication failures	The timeout period may be set to be between 10 minutes and 480 minutes. The allowable number of authentication failures may be set to be between 3 and 10.
3. enable/disable the VPN protection: a. across device [b. on a per-app basis]	VPN functionality may be enabled or disabled across the device, or on a per-app basis.
4. enable/disable [*Mobile network*]	The user may enable or disable access to a mobile network. The following bands are supported: 2G 4G&3G 4G&3G&2G LTE&4G/3G LTE&4G/3G&2G The available frequency ranges for various bands and carriers are publicly available.
4. enable/disable [*Wi-Fi, GPS, FM radio, NFC and Bluetooth*]	Administrators may enable/disable Wi-Fi (wireless local area network), Global Positioning System (GPS), Frequency Modulation (FM) radio, Near Field Communication (NFC) and Bluetooth functionality. The frequency range for each is: Wi-Fi: 2.4 Ghz: 2400 – 2500 Mhz 5 GHz: 5150 – 5250 Mhz 5250 – 5350 Mhz 5470 – 5725 Mhz 5725 – 5850 Mhz GPS: only L1C which has a centre frequency of 1575.42 MHz (P[Y] code spread over a 20.46 MHz bandwidth) FM Radio: 88.1 – 108.1 MHz NFC: 13.56 MHz Bluetooth: 2.4 – 2.5 GHz

Management Function	Description
5. enable/disable [*camera*, *microphone*]: a. across device [*c. no other method*]	The camera and microphone may be enabled or disabled only by the Administrator.
6. specify wireless networks (SSIDs) to which the TSF may connect	The Administrator may specify which wireless networks (SSIDs) to which the TSF may connect.
7. configure security policy for each wireless network: a. [specify the CA(s) from which the TSF will accept WLAN authentication server certificate(s)] b. security type c. authentication protocol d. client credentials to be used for authentication	The policy configuration options for a wireless network allow the Administrator to specify the CAs from which certificates used for WLAN may be accepted, the security type, protocol and client credentials. CA certificates are authorized by creating a CA certificate profile. This profile specifies a CA certificate that devices can use to trust the identity associated with any client or server certificate that has been signed by that CA. The Wi-Fi security type, authentication protocol and client credentials are specified in a Wi-Fi profile. The profile is then assigned to a user or group, causing the configuration settings in that profile to be sent to the TOE device.
8. transition to the locked state	Both users and Administrators may lock a device.
9. TSF wipe of protected data	Both users and Administrators may initiate a device wipe of protected data.
10. configure application installation policy by [a. restricting the sources of applications, b. specifying a set of allowed applications based on [application name and version] (an application whitelist), c. denying installation of applications]	Administrators may restrict the source of applications that may be installed. When the appropriate policies are in place, the Administrator may determine which applications are allowed by users. The applications are identified in the whitelist by application name and version. Administrators may also identify applications that may not be installed.
11. import keys/secrets into the secure key storage	Keys may be imported by either users of the device using the certmgr function or administrators of the management

Management Function	Description
	platform through policy configuration. See the key hierarchy.
12. destroy imported keys/secrets and [no other keys/secrets] in the secure key storage	Only the entity that imported the key may destroy the key.
13. import X.509v3 certificates into the Trust Anchor Database	Only Administrators may import X.509 certificates into the Trust Anchor Database[2].
14. remove imported X.509v3 certificates and [[*All other X.509v3 certificates*]] in the Trust Anchor Database	Only Administrators may remove X.509 certificates from the Trust Anchor Database. This includes Client Identity, Wi-Fi, VPN Identity, CredMgr Client, and ActiveSync Client certificates.
15. enroll the TOE in management	The user is provided credentials and initiates the enrollment. The policies identified for that user by the Administrator are enforced once enrollment is complete.
16. remove applications	User-installed, optional, public applications may be removed by users. However, only Administrators may remove Administrator-installed and Enterprise applications.
17. update system software	Users may initiate system software updates. Administrators may enable or disable the ability of the user to initiate the update, as well as restrict the software versions allowed to be used for an update.
18. install applications	Both users and Administrators may install applications. However, the TOE may be configured to allow only Administrators to install applications. Only applications made available by an administrator may be installed in the workspace. These applications may be designated ‘required’, in which case

[2] The Trust Anchor Database is a special instance of the certificate store accessed by the certificate manger API. The secure key store, or CertStore, is also the certificate store; however, each application may create a new instance of the certificate store in which to place its keys.

Management Function	Description
	they would be automatically installed, or 'optional', in which case they may or may not be installed by the user.
19. remove Enterprise applications	Only the administrator may remove Enterprise applications using an IT Policy setting. All Enterprise applications may be removed.
20. configure the Bluetooth trusted channel: a. disable/enable the Discoverable mode (for BR/EDR) b. change the Bluetooth device name f. disable/enable the Bluetooth services and/or profiles available on the device, [i. *no other Bluetooth configuration*]	The following Bluetooth Profiles and Services may be enabled/disabled on the device: Profiles: A2DP 1.2 AVRCP 1.4 HFP 1.6 OPP 1.1 PAN 1.0 (PAN-U + NAP) PBAP 1.1.1 SAP 1.1 MAP 1.1 HID 1.0 (host and client) HOGP 1.0 Device ID 1.3 GAVDP 1.2 SPP 1.1 Services: IAS 1.0 DIS 1.1 LLS 1.0 TPS 1.0 BAS 1.0 CTS 1.0 NDCS 1.0 ANS 1.0 Encryption Modes 2 and 4 are supported. Every Bluetooth Profile must be trusted. Additionally, the MAP, PBAP and HFP Profiles require a second level of trust.

Management Function	Description
21. enable/disable display notification in the locked state of: enable/disable display notification in the locked state of: [selection: a. email notifications, b. calendar appointments, c. contact associated with phone call notification, d. text message notification,]	Notification information that is displayed when the device is in the locked state may be enabled/disabled by an Administrator. This is enforced for all notifications that would appear in the work space. Text message notifications do not appear in the work space and are therefore always turned off.
22. enable/disable all data signaling over [*USB, SD Card, HDMI*]	The use of USB, SD Card and HDMI interfaces may be enabled/disabled by an Administrator through the implementation of policies.
23. enable/disable [*Media sharing, Miracase, BlackBerry Bridge, Wi-Fi hotspot, Bluetooth*]	Media Sharing: This is used to specify whether a BlackBerry device can share music, pictures, and videos over a Wi-Fi connection with DLNA Certified devices. The "Allow computer to access device" rule must also be selected for the device to share media content with DLNA Certified devices. Miracast: This is used to specify whether a BlackBerry device can send streaming video over a Wi-Fi Direct connection to other Wi-Fi CERTIFIED Miracast devices. The "Allow HDMI" rule must also be selected for the device to send streaming video using Miracast. To ensure a user cannot connect to Miracast devices using any type of connection, do not select this rule. Blackberry bridge: This is used to specify whether a BlackBerry 10 device user can use a BlackBerry PlayBook tablet to access work data on a device using the BlackBerry Bridge app. Wi-Fi hotspot: This is used to specify whether to allow Mobile Hotspot mode, tethering using Bluetooth technology, and tethering using a USB cable on a BlackBerry device. If this rule is

Management Function	Description
	selected, all of these features are available in the settings on the device. If this rule is not selected, none of these features are available on the device. Bluetooth: There are several Bluetooth policies controlling the types of access and types of sharing that may be enabled/disabled. All the above can be enabled/disabled through the use of IT policies.
24. enable/disable developer modes	Only the Administrator can enable/disable developer mode. Users may choose to enable/disable developer mode; however, an administrator may issue a policy restricting this functionality to only administrators.
25. enable data-at rest protection	All data in the Work Space is encrypted by default. The administrator can enable advanced data-at-rest protections which force the data-at-rest protections to be applied immediately (or after a specified period of time) after the work space is locked.
26. enable removable media's data-at-rest protection	The data-at-rest protection for removable media may be engaged by the user, or enforced through a policy created by the administrator. In a BlackBerry Balance configuration the SD card is available to only the personal perimeter. In a Workspace Only configuration it is available to the work perimeter.
27. enable/disable bypass of local user authentication	Only an administrator can reset a Workspace Password. There is a web based feature that may be enabled to allow users to perform limited self-service password reset. If enabled, users can use the web tool to reset device passwords only. To use this feature, the user must first authenticate through the enterprise authentication mechanisms.
28. wipe Enterprise data	The Administrator may initiate a wipe of Enterprise data.

Management Function	Description
29. approve [selection: *import, removal*] by applications of X.509v3 certificates in the Trust Anchor Database	
30. configure whether to establish a trusted channel or disallow establishment if the TSF cannot establish a connection to determine the validity of a certificate	The security policy may be set to permit or deny the establishment of a trusted channel if the certificate validity cannot be verified.
31. enable/disable the cellular protocols used to connect to cellular network base stations	The user may select which cellular protocols to use from the following options: 2G 4G&3G 4G&3G&2G LTE&4G/3G LTE&4G/3G&2G
32. read audit logs kept by the TSF	
33. configure [certificate] used to validate digital signature on applications	The Administrator is able to configure the certificate used to validate digital signatures associated with mobile applications.
34. approve exceptions for shared use of keys/secrets by multiple applications	If an administrator pushes the certificate to the device, the administrator may restrict the applications that may use the certificate. Additionally, the administrator may implement a policy that prevents a user from importing certificates.
35. approve exceptions for destruction of keys/secrets by applications that did not import the key/secret	
36. configure the unlock banner	An Administrator may configure the unlock banner. The banner is restricted to 2500 characters.
37. configure the auditable items	An Administrator may configure which items are to be audited. • Event logging. The administrator may specify that the device collects and sends logs back to BES for audit and regulatory requirements. • Info event logging. The administrator may specify that the device collects

Management Function	Description
	and sends info event logs back to BES for audit and regulatory requirements. • Warning event logging. The administrator may specify that the device collects and sends warning event logs back to BES for audit and regulatory requirements. • Error event logging. The administrator may specify that the device collects and sends error event logs back to BES for audit and regulatory requirements. • Successful event logging. The administrator may specify that the device collects and sends successful events logs back to BES for audit and regulatory requirements. • Failure event logging. The administrator may specify that the device collects and sends failure events logs back to BES for audit and regulatory requirements.
38. retrieve TSF-software integrity verification values	
39. enable/disable [a. *USB mass storage mode*]	The use of USB mass storage may be disabled through use of an IT Policy, as follows: Allow USB On-The-Go (OTG) mass storage. This policy may be used to specify whether a user can use the USB OTG feature and connect USB mass storage devices (such as USB sticks) to a BlackBerry device. If this rule is not selected, the user cannot connect USB mass storage devices to it. Allow computer to access device. This policy may be used to specify whether a computer can access content on a BlackBerry device using a USB connection or the file-sharing option with a Wi-Fi connection. If this rule is not selected, the computer cannot access content on the device using a USB or Wi-Fi connection and the device can't share media content with Digital Living Network Alliance (DLNA) Certified

Management Function	Description
	devices.
40. enable/disable backup to [locally connected system, remote system]	Data may be backed up locally to a USB device. IT policy may be implemented to enable or disable this functionality. Remote backup is not supported, and is therefore always disabled.
41. enable/disable [a. Hotspot functionality authenticated by [pre-shared key], b. USB tethering authenticated by [passcode]]	An IT Policy may be used to specify whether to allow Mobile Hotspot mode, tethering using Bluetooth technology, and tethering using a USB cable on a BlackBerry device. If this rule is selected, all of these features are available in the settings on the device. If this rule is not selected, none of these features are available on the device. Use of a pre-shared key to access the hotspot is supported. A user cannot enable/disable USB tethering without first entering the workspace passcode.
42. approve exceptions for sharing data between [selection: *application processes, groups of application processes*]	
43. place applications into application process groups based on [assignment: *application characteristics*]	
44. enable/disable location services: a. across device [c. no other method]	Location services may be enabled/disabled for the device.

Management Function	Description
45. [a. *enable/disable automatic transfer of diagnostic data to an external device or service other than an MDM service with which the device has enrolled;* b. *disenroll the TOE in management;* c. *enable/disable multi-user modes;* d. *enable/disable automatic updates of system software (see function 17);* e. *wipe non-enterprise data*].	a. Diagnostic data originating at the Mobile Device may be sent to wireless service providers. This functionality may be enabled or disabled. b. A device may be deactivated by an Administrator, essentially disenrolling the device in management. c. Multi-user modes are disabled. d. The allow wireless software updates policy may be set to allow all updates, allow security updates only or to disallow updates. e. The Administrator may choose to wipe only the work space, or the entire device.

Table 22 – Management Functions

This is a description of how the TOE meets the security objectives.

When using the ‘Workspace only’ configuration, all protected data is automatically wiped in the case of unenrollment.

TOE Security Functional Requirements addressed: FMT_MOF_EXT.1, FMT_SMF_EXT.1, FMT_SMF_EXT.2, FMT_SMF_EXT.3.

7.5.2 Trusted Policy Update

Policies are pushed from the MDM Server. When existing policies are modified, or new policies are added, the policy will be pushed to the device immediately or in accordance with a predetermined schedule. Whenever a device contacts the MDM Server, the MDM Server confirms that the device has the most recent version of all required policies and pushes any required policies/updates.

Before a policy is sent, the MDM Server constructs a SHA 512 hash of the message body, encodes it using base64, adds salt content to it, and then calculates a signature based on the current ECDSA encryption/decryption key. It then puts both the data with the hash, and its signature, in the message header.

The Enterprise Management Agent (EMA) (also called the MDM Agent) independently calculates a SHA 512 hash of the message body, and rejects the message if the hashes do not match. This ensures that the hash sent in the header corresponds to the message body, and that the entire string that was signed on the server corresponds to the message body.

The MDM Agent then verifies the signature against the message body, using the public portion of the same private/public ECDSA key pair that the MDM Server used to create the signature.

If this signature check fails, the message is rejected. The message is also rejected if any of the noted parts are missing from the header.

Upon rejection of the message, the event is logged both in the standard EMA log and in the NIAP CCL log. An exception is raised, ensuring that the data is not processed. The MDM Agent's internal statistics related to the network throughput (number of errors, etc.) is updated as well.

TOE Security Functional Requirements addressed: FMT_POL_EXT.2.

7.5.3 User Unenrollment Prevention

Prevention of user unenrollment is provided through the configuration of the 'Allow users to deactivate devices' policy. This policy may be used to specify if the user is permitted to deactivate the device and wipe all work data. If this rule is not selected, users cannot delete the work space. If the user exceeds the Maximum password attempts allowed, work data is not wiped from the device and the device can only be unlocked by an administration command.

TOE Security Functional Requirements addressed: FMT_UNR_EXT.1.

7.6 PROTECTION OF THE TSF

7.6.1 Anti-Exploitation Services

The BlackBerry 10 device prevents exploitation of memory corruption in a number of different ways, including the six security mechanisms listed in Table 23.

Security Mechanism	Description
Non-executable stack and heap	The stack and heap areas of memory are marked as non-executable. This means that a process cannot execute machine code in these areas of the memory, which makes it more difficult for an attacker to exploit potential buffer overflows.
Stack cookies	Stack cookies are a form of buffer overflow protection that helps prevent attackers from executing arbitrary code.
Robust heap implementations	The heap implementation includes a defence mechanism against the deliberate corruption of the heap area of memory. The mechanism is designed to detect or mitigate the overwriting of in-band heap data structures so that a program can fail in a secure manner. The mechanism helps prevent attackers from executing arbitrary code via heap corruption.

Security Mechanism	Description
Address space layout randomization (ASLR)	By default, the memory positions of all areas of a program are randomly arranged in the address space of a process. This mechanism makes it more difficult for an attacker to perform an attack that involves predicting target addresses to execute arbitrary code.
Compiler-level source fortification	The compiler GCC uses the FORTIFY_SOURCE option to replace insecure code constructs where possible. For example, it might replace an unbounded memory copy with its bounded equivalent.
Guard pages	If a process attempts to access a memory page, the guard page raises a onetime exception and causes the process to fail. These guard pages are placed strategically between memory used for different purposes, such as the standard program heap and the object heap. This mechanism helps prevent an attacker from causing a heap buffer overflow and changing the behaviour of a process or executing arbitrary code with the permissions of the compromised process.

Table 23 – Anti-Exploitation Services

7.6.1.1 Anti-Exploitation Services (ASLR)

The kernel employs a pseudo-random number generator, based on a few seed sources, which are fed to a SHA256 hash function. Whenever a non-fixed user-mapping is created, the virtual address is adjusted by taking a value from the PRNG and using it to change bits 12-23 in the resulting address. This ensures the availability of more than 8 unpredictable bits.

TOE Security Functional Requirements addressed: FPT_AEX_EXT.1.

7.6.1.2 Anti-Exploitation Services (Memory Page Permissions)

The QNX Neutrino microkernel integrates with the MMU, controlling access to the virtual memory. Every memory access is governed by the MMU. On Advanced RISC Machine (ARM) chips, every entry in the page table is associated with read, write and execute permissions, which govern all memory accesses.

User processes cannot change page tables directly since they are able to address and contact the physical memory holding the tables. All changes to page table permissions go through an API (mmap/mprotect) governed by the kernel. Unless a special permission has been granted to a process, attempting to set both write and execute permissions on a page will be rejected.

The ability to map pages with simultaneous execute and write permissions is controlled by a procmgr ability and enforced by the QNX microkernel. (Note that QNX uses the term 'ability' to describe processes with privileges.) It is an additional restriction, and is identified by the PROCMGR_AID_WRITE_AND_EXEC procmgr ability. If the ability is enabled in a process, requests to map pages with both the execute and write bits set will succeed. If the ability is not enabled, then any attempt to map memory with both the execute and write permissions enabled will fail.

TOE Security Functional Requirements addressed: FPT_AEX_EXT.2.

7.6.1.3 Anti-Exploitation Services (Overflow Protection)

The stack and heap areas of memory are marked as non-executable. This means that a process cannot execute machine code in these areas of the memory, which makes it more difficult for an attacker to exploit potential buffer overflows. Stack cookies are a form of buffer overflow protection that helps prevent attackers from executing arbitrary code.

Stack based overflow prevention is implemented in accordance with GCC, version 4.8.3 (http://wiki.osdev.org/Stack_Smashing_Protector). Although fstack-protector-all is used for most components, fstack-protector is also used. For those components that do not implement stack overflow, it is because they do not include functions that work on stack based buffers.

The following table lists the TSF binaries and libraries.

Function	Path	Name
WPA Supplicant	/base/usr/sbin/	P2P_supplicant_ti_08
Certificate Manager	/base/usr/sbin/	certmgr_pps
Ceriticate Utility	/base/usr/sbin/	certutil
Authentication	/base/usr/lib/	/libcredential.so.1
Authentication	/base/usr/lib/	libcredmgr_client.so.1
Cryptography	/base/usr/lib/	libcrypto.so.2
Wireless	/base/services/	wireless_manager
Wi-Fi	/base/services/	wifi_p2p_send
Device Wipe	/base/sbin/	/base/sbin/wipe
Authentication	./apps/	libbbauthplugin.so.1
Audit Logging	/base/usr/lib/	syslog.so
TLS	/base/usr/lib/	_ssl.so
Bluetooth	/base/usr/lib/	libbluetooth.so.1

Function	Path	Name
Authentication	./apps/	libPasswordKeeper.so.1.0.0
Cryptography	/base/usr/lib/	_certicom.so
Policy	/base/bin/	ssr_policy_mgr
Enrollment	/base/usr/lib	libemasec.so.1
Trust Management	/base/usr/lib/	libtrustmgrp.so.1
Time Services	/base/usr/lib/	time.so
Time Services	/base/usr/lib/	_datetime.so
Cryptography	/base/usr/lib/	garbage_zeroizer.so
Authentication	/base/bin/	login

Table 24 – TSF Binaries

TOE Security Functional Requirements addressed: FPT_AEX_EXT.3.

7.6.1.4 Anti-Exploitation Services (Domain Isolation)

The TOE validates the boot ROM, operating system, radio code, and applications before they are loaded. The operating system and the radio code are stored in the read-only media and applications require kernel level permissions to be modified.

Address spaces are kept apart through the use of virtual memory and per-address-space page tables that translate virtual addresses to physical ones. This ensures that applications are not able to access memory allocated to other applications.

Non-TSF software is prevented from modifying the TSF software or data that governs the behaviour of the TSF. Only applications which are approved by the Administrator may have access to TSF software and data. The application whitelist provides the Administrator the ability to have complete control over applications and software residing within the boundary.

The TOE does not include the provision of auxiliary boot modes. The phone dialer is not available when the device is in the locked state.

TOE Security Functional Requirements addressed: FPT_AEX_EXT.4.

7.6.2 Application Processor Mediation

The broadband processor (BP) does not have direct access to the application processor (AP) resources. BP access to AP resources is mediated by the Qualcomm component, xPU, which works with TrustZone.

The AP, baseband and other cores, communicate over shared memory via the Qualcomm Messaging Interface protocol (QMI). Shared memory regions for inter-processor communication are configured for mutual access in the XPU.

The XPU controls which of the Mobile Station Modem (MSM) resources may be accessed by the BP. Sensors external to the MSM are accessed through Inter-integrated circuit (I2C) interfaces. The BP is not on those I2C buses; however, BP access to those buses is determined by the xPU configuration. The MSM controllers for these are called Bus Access Manager/Module Low Speed Peripheral (BLSP) interface ports or BLSP ports. Microphones are connected via an external CODEC which is connected via a Slimbus interface. The BP cannot directly configure the CODEC or this bus. Access to memory containing audio data and control related information is determined by the xPU.

The design prevents modification of executable memory of the AP by the BP. Access from the BP to the main memory is restricted by the xPU such that only the main memory designated for message passing may be accessed. This shared memory block does not contain executables.

TOE Security Functional Requirements addressed: FPT_BBD_EXT.1.

7.6.3 Key Storage

The TOE ensures that all keys held in non-volatile storage are wrapped with a KEK. All data in the workspace is persisted in encryption domains. Any plaintext keying material is therefore encrypted by default when persisted. Additionally, individual components may choose to provide additional levels of encryption on specific keys. Protection of keying material is outlined in the various key hierarchy descriptions.

All keying material is stored within encryption domains within the file system. Availability of keying material on power-up depends on the storage location of the data. Data in the Startup domain becomes available on power-up. Data in the Operational and Lock domains is not available until the user has authenticated to the workspace. Additionally, data in the Lock domain becomes unavailable when the TOE transitions to the locked state. Regenerated keys reside within the workspace, and are therefore also subject to this protection.

The TOE Security Boundary consists of the application processor, TrustZone, BlackBerry Trusted Application, BlackBerry cryptographic module, and the following keystores:

- RPMB and rpmb resmgr
- fsecd
- Perimeter manager
- filesystem
- data_queue_man
- certificate manager
- credential manager

- Wi-Fi
- wpa_supplicant

as well as an implementation of OpenSSL, VPN, personal information management, Enterprise Management Agent, SCEP Client, and Bluetooth.

The REK is locked in hardware and is only accessible for use by the BlackBerry trusted application. Keying materials derived from the REK are only available for use within the BlackBerry trusted application. Access to the BlackBerry trusted application is limited to select privileged components with the required permissions. Keying material in RPMB is only accessible by privileged system processes that are able to correctly authenticate to RPMB.

Access and use of keying material is controlled by each owning component within the Security boundary. Use is typically only within the owning component. For example, DEKs owned by the filesystem component only exist in plaintext in the filesystem component when a file is in use. Those DEKs are destroyed in memory when the file is closed or the TOE is locked. Passing of keying material between components within the Security Boundary occurs according to access rules that are defined at design time, and which are hard coded into the TOE. For example, the PerimeterMgr passes the Domain Master Key to the Filesystem during unlock. In this case, both the PerimeterMgr and the File system destroy the key in memory when no longer in use. If keying material is passed outside of the TOE Security boundary, as in passing a password to a service for authentication, it is only passed using encrypted protocols.

Third party applications storing their keying material within the security boundary of the TOE are responsible for protection of their keying material outside of the Security Boundary of the TOE.

The same protections are provided in each power loss scenario.

TOE Security Functional Requirements addressed: FPT_KST_EXT.1.

7.6.4 No Key Transmission

The TOE does not transmit any plaintext key material outside the security boundary of the TOE.

The TOE Security Boundary consists of the application processor, the TrustZone, the BlackBerry Trusted Application, the BlackBerry cryptographic module, RPMB and rpmb resmgr, fsecd, Perimeter manager, filesystem, data_queue_man, certificate manager, Wi-Fi, wpa_supplicant, an implementation of OpenSSL, VPN, personal information management, Enterprise Management Agent, SCEP Client, and Bluetooth.

The REK is stored in hardware and is only accessible for use by the BlackBerry trusted application. Keying materials derived from the REK are only available for use within the BlackBerry trusted application. Access to the BlackBerry trusted application is limited to select privileged components with required permissions. Keying material in RPMB is only accessible by privileged system processes that are able to correctly authenticate to RPMB.

All keying material in the file system is stored within encryption domains. Availability of keying material on power-up depends on the storage location of the data. Data in the Startup domain becomes available on power-up. Data in the Operational or Lock domain is not available until the user has authenticated to the workspace. Additionally, data in the Lock domain becomes unavailable when the TOE transitions to the locked state.

Access and use of keying material is controlled by each owning component within the Security boundary. Use is typically only within the owning component. For example, DEKs owned by the filesystem component only exist in plaintext in the filesystem component when a file is in use. Those DEKs are destroyed in memory when the file is closed or the TOE is locked. Passing of keying material between components within the Security Boundary occurs according to access rules defined at design time that are hard coded into the TOE. For example, the PerimeterMgr passes the Domain Master Key to the Filesystem during unlock. In this case both the PerimeterMgr and the File system destroy the key in memory when it is no longer in use. If keying material is passed outside of the TOE Security boundary, as in passing a password to a service for authentication, it is only passed using encrypted protocols.

Any third party application that stores its keying material within the security boundary of the TOE is responsible for protection of that keying material when transferred outside of the Security Boundary of the TOE.

There are no APIs for transmission of plaintext keys beyond the security boundary.

TOE Security Functional Requirements addressed: FPT_KST_EXT.2.

7.6.5 No Plaintext Key Export

The TSF does not allow users to export plaintext DEKs, KEKs, or keys stored in the secure key storage. Backup of key material may be disabled through a policy.

TOE Security Functional Requirements addressed: FPT_KST_EXT.3.

7.6.6 Self-test Notification

The TOE performs self-tests on its cryptographic functionality and integrity validation on TSF software during the boot process. If one of these tests fails to complete, the TOE does not boot and thus transitions to a non-operational state.

Known Answer Tests (KATs) are performed on the cryptographic functions implemented in the cryptographic module, including those that are not implemented in the TSF. This includes TDES, AES, AES GCM, SHS (using HMAC-SHS), HMAC-SHS, DRBG, ANSI X9.62 RNG, ANSI X9.31 RNG, RSA Signature Algorithm, and KDF. For DSA and ECDSA, a Pair-wise Consistency Test is used. For DH, ECDH, ECMQV, the underlying arithmetic implementations are tested using DSA and ECDSA tests. Any failure of a KAT is a critical failure.

The software integrity test deploys ECDSA signature validation to verify the integrity of the cryptographic module. Failure of the integrity test is a critical failure.

Self-test failure places the cryptographic module in the Error state, wherein no cryptographic operations can be performed. If any self-test fails, the cryptographic module produces an error code and enters the Error state.

TOE Security Functional Requirements addressed: FPT_NOT_EXT.1.

7.6.7 Reliable Time Stamps

The TOE uses the carrier provided time by default, but may be configured to synchronize its system time with an authoritative central server to provide reliable time stamps.

The following functions use the provided time:

1. Certificate Management Functions - Certificate Manager (CertMgr)

   The following Certificate Management Functions require time:

   - Timeout on receiving response to a OCSP request
   - Timestamps inside OCSP response
   - OCSP response cache validity
   - Timeout on monitoring IT policies in PPS objects
   - Timeout on CertMgr functions (if set by certmgr_SetTimeoutMsec())
   - Timeout on Smart Card operations
   - Timeout on PPS communication channel between CertMgr client library and CertMgr server
   - Certificate validation: certification timestamp validation, and validation cache

   Time is obtained using the following functions: time(), gettimeofday() and clock_gettime(). The Certificate Manager does not change system time.

2. Key Store Functions - Credential Manager (CredMgr)

   The following Credential Manager Functions require time:

   - Timeout during Kerberos operation
   - Credential record timestamps and expiry

   Time is obtained using the following functions: time(), clock() and clock_gettime(). The Credential Manager does not change system time.

3. Work Space Lock Functions - Perimeter Manager (PMgr)

   The following Perimeter Manager Functions require time:

   - Password validity
   - IT policy publishing update time

- Time is added to policy group names
- Timing transition between UX Locked device state and Data Locked (ADARP) device state
- Timestamp in encryption/decryption transaction file, published for file system
- Unlocked device timer after initial CorpLiable perimeter setup
- Timeout when passing password derived key to EMA

Time is obtained using the following functions: time(), and clock_gettime(). The Perimeter Manager does not change system time.

4. Smart Card Functions - Smart Card Subsystem (SCS)

The following Smart Card Functions require time:

- NFC connect timeout, NFC transmit timeout
- Smart card powerdown timing
- Bluetooth smart card reader connection timeout
- Media Card Reader timeout
- Smart card communication timeout
- PIN cache validity
- Timestamp when publishing smart card reader status to PPS
- Timeout on PPS communication channel between SC client library and SC service

Time is obtained using the following functions: time(), clock_gettime(), and gettimeofday(). The Smart Card Functions do not change system time.

5. SSL Functions - OpenSSL Stack

The following SSL Functions require time:

- timestamp on message - put time stamp on protocol message as per TLS protocol

Time is obtained using the following functions: time(). SSL Functions do not change system time.

6. VPN Functions – VPN Stack

The following VPN Functions require time:

- state timeouts - process internal state machines timeout
- protocol timeouts - session time out as per implemented security function RFCs (e.g. ikev2, ipsec)
- certificate validation - certification timestamp validation

Time is obtained using the following functions: clock_gettime(). The VPN Functions do not change system time.

7. RNG Functions - Random

The following RNG Functions require time:

- Uses clock time to determine the minimum number of seconds before seed, drbg, or nvram updates will occur.
- Yarrow uses the current time when initializing the entropy pool upon startup.

Time is obtained using the following functions: srandom drbg uses clock_gettime(). The RNG Functions do not change system time.

TOE Security Functional Requirements addressed: FPT_STM.1.

7.6.8 TSF Cryptographic Functionality Testing

The BlackBerry OS Cryptographic Library performs a power-on self test (POST) to ensure the cryptographic library is not modified and all the cryptographic functions perform correctly.

TOE Security Functional Requirements addressed: FPT_TST_EXT.1.

7.6.9 Integrity Testing

Secure boot refers to startup of the processor software located in the internal ROM, which authenticates the boot ROM before execution. This processor software authentication protects against replacement of the flash chip.

The process to bind a processor to the Code Signing key is performed once and is irreversible since it involves burning fuses.

After a processor is bound, only a boot ROM that has been signed by supply chain management can be loaded onto the device with the same signing key, and only if the boot ROM is not write protected. If the boot ROM on the device fails authentication, the device will fail to enumerate as a BlackBerry device and the LED will never turn on.

The secure boot-up process for a mobile device is as follows:

1. A reset on the Application Processor core 1 is released.
2. The Primary Boot Loader is executed.
3. The Primary Boot Loader executes the first Secondary Boot Loader.
4. The first Secondary Boot Loader executes the second Secondary Boot Loader.
5. The second Secondary Boot Loader loads and begins initialization of TrustZone.
6. TrustZone returns execution back to the second Secondary Boot Loader.
7. The second Secondary Boot Loader starts the mobile device OS.

These steps execute within the internal static random access memory (SRAM). The OS is loaded into and executes in dynamic random access memory (DRAM).

The first Secondary Boot Loader is loaded from flash by the Primary Boot Loader. The first Secondary Boot Loader is not decrypted because it is stored unencrypted on flash. The OEM Public Key HASH is used to verify a BlackBerry Certificate stored on flash, and this certificate is then used to verify that the first Secondary Boot Loader was signed by BlackBerry.

The TOE does not support auxiliary boot modes.

TOE Security Functional Requirements addressed: FPT_TST_EXT.2.

7.6.10 Trusted Update: TSF Version Query

The TOE user may query the current version of the TOE firmware/software, current version of the hardware model of the device, and the current version of installed mobile applications.

TOE Security Functional Requirements addressed: FPT_TUD_EXT.1.

7.6.11 Trusted Update Verification

The TOE verifies software updates to the Application Processor using digital signatures. A hardware-protected public key held in the Trust Anchor Database is used to verify the key used for signatures on software updates. The TOE only allows the boot integrity hash to be updated during a software update process.

The digital signature verification of the software occurs before installation and the installation fails if the verification fails. The application or service being installed is delivered to the device via a BAR file. System software updates can only be delivered by OTA update servers. This includes the OS image, the radio image (which boots on the baseband processor and provides all radio functionality), and system applications.

When an update is received, the checksum of the OS image within the BAR is verified. The OS image is in a proprietary file format which includes headers and trailers. A Software Change Management Cyclical Redundancy Check (SCM CRC) signature is obtained from the trailer in one of the sections of the image. If this does not match the calculated check sum, the OS image will not be installed. The checksum of the assets within the BAR file are still calculated and compared those stored in the MANIFEST.MF file.

Verification is performed by the installer prior to installation by executing a hash calculation on the contents of the BAR file and comparing that calculation to the contents of the manifest file (MANIFEST.MF), which is held in the META-INF sub-directory of the BAR file. In the manifest is a list of all of the assets within the BAR file, and their hash values.

The manifest file is generated by the BAR packager. Before signing occurs, the BAR manifest is modified in the following ways:

1. A Package-Author-Certificate-Hash attribute is inserted, containing a hash of the developer certificate;

2. For all files in the BAR archive, with the exception of a small number of special files in the META-INF directory, the hashes of their contents are added to the BAR manifest;
3. Files called META-INF/RDK.SF and META-INF/AUTHOR.SF are created. These files contain the hash of the manifest as well as hashes of individual sections of the manifest;
4. During signing, the file RDK.SF file is signed using the private key corresponding to the hardware-protected public key used on the device for verifying BAR files. The signature is created as a PKCS#7 structure and added to the META-INF/RDK.EC file;
5. The Developer certificate is inserted into this PKCS#7 structure;
6. The AUTHOR.SF file is digitally signed with the private key corresponding to the developer certificate. That signature is also a PKCS#7 structure and is added to the META-INF/AUTHOR.EC file; and
7. The Developer certificate is added to this PKCS#7 structure as well.

The hash of the BAR file contents is stored in the fingerprint file (FINGERPRINT.DAT). The FINGERPRINT.DAT file is in the META-INF subdirectory. This file exists for OS BAR files and RDK.EC and AUTHOR.SF/AUTHOR.EC files.

Hash values are calculated using SHA-512. The signature on the developer certificate is verified to ensure the validity of the AUTHOR.EC PKCS#7 structure. Verification of the RDK.EC signature is performed using the hardware-protected public key. A Certicom X.509 certificate decoder is used to perform the signature verification function. The Package-Author-Certificate-Hash is held in the MANIFEST.MF file.

BlackBerry maintains the chain of trust by controlling the package signing, and by using hash algorithms (as part of the digital signature) to confirm that the BAR file has not been tampered with. If the checks fail at any point, the BAR file cannot be installed.

TOE Security Functional Requirements addressed: FPT_TUD_EXT.2.

7.7 TOE ACCESS

7.7.1 TSF- and User-initiated Locked State

The TOE can transition to a locked state after a time interval of inactivity, a user-initiated lock, or receiving the change password and lock device command from the BlackBerry Enterprise Service administration. When the TOE is locked, the display is cleared to obscure any previous contents. From a locked screen, the user is only able to view the date, time, battery level, signal strength, and the number of unread notifications available.

TOE Security Functional Requirements addressed: FTA_SSL_EXT.1.

7.7.2 Default TOE Access Banners

The administrator can configure the TOE to display an access banner on the lock screen.

TOE Security Functional Requirements addressed: FTA_TAB.1.

7.7.3 Wireless Network Access

The TOE user and administrator may configure a list of wireless access points (SSIDs) to which the TOE can connect. The TOE automatically attempts to connect to a configured access point when the TOE is within the range of the network.

TOE Security Functional Requirements addressed: FTA_WSE_EXT.1.

7.8 TRUSTED PATH / CHANNELS

7.8.1 Trusted Channel Communication

The TSF supports the use of 802.1X, EAP-TLS, IPsec, and TLS protocols for trusted channel communications, as follows:

- Wireless access point access is supported through the use of 802.1X, EAP-TLS
- TLS is used to secure the administrative channel between the TOE and the MDM Server.
- Configured Enterprise connections are protected using TLS
- OTA updates are protected using TLS
- VPN tunnels are protected using IPsec

The TOE allows the TSF to initiate communications through the trusted channel and uses a trusted channel to connect to wireless access point connections, administrative communication, configured enterprise connections, OTA updates, and a VPN Gateway.

TOE Security Functional Requirements addressed: FTP_ITC_EXT.1.

7.9 SECURITY UPDATES

BlackBerry 10 OS updates include updates for the BlackBerry 10 Operating System firmware and bundled applications.

Updates to the BlackBerry 10 OS are subject to carrier testing and acceptance. BlackBerry provides ‘carrier evaluation’ builds to the carrier partners, which enables early testing of BlackBerry 10 OS updates. When the Gold Candidate build is provided to the carrier, the early testing minimizes the testing and turnaround time for the build to be approved for release to customers. Ultimately, the approval and release timelines for BlackBerry 10 OS updates are controlled by the carrier partners.

All security updates are released in the form of a complete BlackBerry 10 OS update.

BlackBerry utilizes industry best practices to ensure their devices are patched to mitigate security flaws. Updates and patches to resolve reported issues are created as quickly as possible, at which point the update is provided to the wireless carriers. The delivery time for resolving an issue depends on the severity, and can be as rapid as a few days before the carrier handoff for high priority cases. The wireless carriers perform additional tests to ensure the updates will not adversely impact their networks and then plan device rollouts once that testing is complete. Carrier updates usually take at least two weeks to as much as three months to be rolled out to customers, depending on the type and severity of the update. BlackBerry communicates with the reporting parties to inform them of the status of the reported issues. Further information about updates is handled through the carrier release notes and security advisory notices, both of which are available on blackberry.com.

Security issues related to the TOE may be reported using the BlackBerry Security Incident Response website at www.blackberry.com/bbsirt/. Communications to the vulnerability reporting website use TLS 1.2 and AES 256 to secure the session.

8 TERMINOLOGY AND ACRONYMS

8.1 TERMINOLOGY

The following terminology is used in this ST:

Term	Description
MAS	A Mobile Application Store Server is an application on a general-purpose platform or on a network device, executing in a trusted network environment.
MDM Agent	The MDM Agent is installed on the mobile device as an application or as part of the mobile device's operating system. It is responsible, with the MDM Agent platform, for enforcing the SFRs on the mobile device.
MDM Server	The MDM Server is an application on a general-purpose platform or on a network device that executes in a trusted network environment. It is responsible for managing device enrollment, configuring and sending policies to the MDM Agents, collecting reports on device status and sending commands to the MDM Agents.
Wi-Fi	Wi-Fi refers to any technology that complies with the IEEE 802.11x standard.

Table 25 – Terminology

8.2 ACRONYMS

The following acronyms are used in this ST:

Acronym	Definition
ACL	Access Control List
AES	Advanced Encryption Standard
ARM	Advanced Reduced Instruction Set Computing Machine
ANSI	American National Standards Institute
AP	Application Processor
AP	Access Point (when discussing Wi-Fi)
API	Application Programming Interface
ASLR	Address Space Layout Randomization
BP	Baseband Processor
BES	BlackBerry Enterprise Service

Acronym	Definition
Bluetooth LE	Bluetooth Low Energy
BR/EDR	Basic Rate/Enhanced Data Rate
BSS	Basic Service Set
CA	Certification Authority
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CC	Common Criteria
CCM	Counter with CBC-Message Authentication Code
CCMP	Counter Mode CBC-MAC Protocol
CFB	Cipher Feedback
CMAC	Cipher-based Message Authentication Code
CMC	Certificate Management over Cryptographic Message Syntax
CMVP	Cryptographic Module Validation Program
CN	Common Name
CRL	Certificate Revocation List
CTR	Counter
DAR	Data-at-rest
DDSK	Derived Domain System Key
DEK	Data Encryption Key
DK	Domain Key
DLNA	Digital Living Network Alliance
DLQ	Data Lock Queue
DMK	Domain Master Key
DNS	Domain Name System
DRBG	Deterministic Random Bit Generator
DSK	Domain System Key
DSL	Digital Subscriber Line
DSS	Digital Signature Standard

Acronym	Definition
DTLS	Datagram Transport Layer Security
EAP	Extensible Authentication Protocol
EAPOL	Extensible Authentication Protocol over Local Area Network
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
ECIES	Elliptic Curve Integrated Encryption Scheme
ECMQV	Elliptic Curve Menezes-Qu-Vanstone
EEPROM	Electrically Erasable Programmable Read-Only Memory
EMA	Enterprise Management Agent
EMSK	Extended Master Session Key
EST	Enrollment over Secure Transport
FEK	File Encryption Key
FIPS	Federal Information Processing Standards
FM	Frequency Modulation
FS	File System
FQDN	Fully Qualified Domain Name
GCC	GNU Compiler Collection
GCM	Galois Counter Mode
GID	Group ID
GPS	Global Positioning System
GTK	Group Temporal Key
HDMI	High Definition Multimedia Interface
HFP	Hands Free Profile
HLOS	High Level Output Specification
HMAC	Hash Message Authentication Code
HSM	Hardware Security Module

Acronym	Definition
HTTPS	Hypertext Transfer Protocol Secure
I2C	Inter-integrated circuit
IEEE	Institute of Electrical and Electronics Engineers
IKE	Internet Key Exchange
IP	Internet Protocol
IPsec	Internet Protocol Security
IT	Information Technology
IV	Initialization Vector
KAT	Known Answer Test
KDF	Key Derivation Function
KEK	Key Encryption Key
KW	Key Wrap
KWP	Key Wrap with Padding
LTE	Long Term Evolution
MAC	Message Authentication Code
MAP	Message Access Profile
MAS	Mobile Application Store
MD	Mobile Device
MDF	Mobile Device Fundamentals
MDF PP	Protection Profile for Mobile Device Fundamentals
MDM	Mobile Device Management
MK	Master Key
MMU	Memory Management Unit
MSM	Mobile Station Modem
NFC	Near Field Communication
NIAP	National Information Assurance Partnership
NIST	National Institute of Standards and Technology
OAEP	Optimal Asymmetric Encryption Padding

Acronym	Definition
OCSP	Online Certificate Status Protocol
OE	Operational Environment
OEM	Original Equipment Manufacturer
OFB	Output Feedback
OID	Object Identifier
OS	Operating System
OSAT	Outsourced Semiconductor Assembly and Test
OSP	Organizational Security Policy
OTA	Over the Air
OTG	On-The-Go
PAE	Port Access Entity
PBAP	Phone Book Access Profile
PBKDF	Password Based Key Derivation Function
PIM	Personal Information Manager
PKCS	Public-Key Cryptography Standards
PMK	Pairwise Master Key
POSIX	Portable Operating System Interface
POST	Power-on self test
PP	Protection Profile
PRF	Pseudorandom Function
PRNG	Pseudorandom Number Generator
PTK	Pairwise Transient Key
QMI	Qualcomm Messaging Interface
RA	Registration Authority
RAM	Random Access Memory
RBG	Random Bit Generator
REK	Root Encryption Key
RFC	Request for Comment

Acronym	Definition
ROM	Read Only Memory
RPMB	Read Protected Memory Block
RSA	Rivest, Shamir and Adleman
SAR	Security Assurance Requirement
SB GSE-C	Security Builder Government Security Edition
SCEP	Simple Certificate Enrollment Protocol
SD	Secure Digital
SDP	Service Discovery Protocol
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SOC	System on a chip
SNMP	Simple Network Management Protocol
SP	Special Publication
SRP	Server Routing Protocol
SS	Shared Secret
SSID	Service Set Identifier
ST	Security Target
TA	Trusted Application
TD	Technical Decision
TDES	Triple Data Encryption Standard
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSS	TOE Summary Specification
TZ	Trust Zone
USB	Universal Serial Bus
USB OTG	USB On-The-Go
VoIP	Voice over IP

Acronym	Definition
VPN	Virtual Private Network
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access
XOR	Exclusive Or
xPU	Qualcomm eXternal Protection Unit
XEX	XOR Encrypt XOR
XTS	Tweakable Block Cipher with Ciphertext Stealing
2G/3G/4G	Second Generation/Third Generation/Fourth Generation

Table 26 – Acronyms