

Samsung Android 5 on Galaxy Devices
Guidance documentation

Version 1.16
April 14, 2014

Document management

Document identification

Document ID	Samsung Guidance documentation 1.16
Document title	Samsung Android 5 on Galaxy Devices Guidance documentation
Release authority	

Document history

Version	Date	Description	Author
0.1	16-October-2013	Initial draft	
0.2	20-December-2013	Initial draft for Samsung review.	
0.5	January 31, 2014	Update for Android 4.4	Brian Wood
0.6	February 3, 2014	Updated CC Mode API	Brian Wood
0.7	February 10, 2014	Updated based on feedback from CC evaluator	Brian Wood
0.8	February 11, 2014	Added info about determining versions of device, OS & apps	Brian Wood
0.9	February 12, 2014	Added info about obtaining API SDK	Brian Wood
0.10	February 13, 2014	Updates CC Mode app settings	Brian Wood
0.11	February 20, 2014	Added versioning information	Ed Morris
1.1	March 31, 2014	Updated for Galaxy S5/Note 10.1	Brian Wood
1.2	April 3, 2014	Added CRL Checking to the list of required settings	Brian Wood
1.3	April 23, 2014	Corrected Max Password value range	Brian Wood
1.4	April 29, 2014	Updated to show VPN release number	Brian Wood
1.5	April 30, 2014	Removed device locking on password failure	Brian Wood

1.5a	May 2, 2014	Updated device list	Brian Wood
1.5b	June 6, 2014	Modified device list table	Brian Wood
1.6	August 1, 2014	Updated for new devices and options	Brian Wood
1.7	September 7, 2014	Updated for KNOX configurations & devices	Brian Wood Sung Whan Moon
1.8	September 15, 2014	Edited container disable list	Brian Wood
1.9	September 18, 2014	Edits based on KNOX eval feedback	Brian Wood
1.10	September 19, 2014	Updated device list	Brian Wood
1.11	October 7, 2014	Edited versions and CC Mode access	Brian Wood
1.12	October 20, 2014	Edited CC mode access	Brian Wood
1.13	October 28, 2014	Updated device list	Brian Wood
1.14	October 30, 2014	Edits based on Validator feedback	Brian Wood
1.15	December 5, 2014	Edits based on Validator feedback	Brian Wood
1.16	April 14, 2015	Updated download site. Updated for OS update & devices	Brian Wood

Table of Contents

1	Document Introduction	6
1.1	Evaluated Devices	6
1.2	Terminology/Glossary	8
2	Guidance Overview	9
3	Introduction	10
3.1	Overview	10
3.2	Evaluated Capabilities	10
3.3	SAFE/KNOX Management API.....	12
4	Deployment process	13
4.1	Enterprise architecture	13
4.2	Secure preparation of the Enterprise Environment	17
4.3	Secure installation of Samsung Android user devices	17
4.4	Secure Delivery	36
4.5	Secure Updates	39
5	Operational security	40
5.1	Modes of operation	40
5.2	Wiping data	41
5.3	Additional notes on operational security	42

List of Figures

Figure 1 – Enterprise Environment 16

Figure 2 - Tracking label 37

Figure 3 - Security Seal (Black) 37

Figure 4 - Security Seal (White)..... 37

1 Document Introduction

This document contains enterprise guidance for the deployment of Samsung devices in accordance with the Common Criteria configuration.

1.1 Evaluated Devices

The Common Criteria evaluation was performed on devices with specific processors. The following list is divided based on the processors used in the devices that were evaluated:

- Qualcomm Snapdragon
 - Samsung Galaxy S4
 - Samsung Galaxy Note 3
 - Samsung Galaxy Note 4
 - Samsung Galaxy NotePRO
 - Samsung Galaxy Tab S (8.4 & 10.5) LTE
 - Samsung Galaxy Tab Active
- Qualcomm Snapdragon or System LSI Exynos
 - Samsung Galaxy S5
 - Samsung Galaxy Note Edge
 - Samsung Galaxy Alpha

All device models are evaluated with Samsung Android 5 (Lollipop). Other Samsung devices have the same processors and OS version as an evaluated device (i.e. a derivative device) and may be able to be placed into a configuration matching the evaluated configuration of these devices, but only the devices listed above have been evaluated for compliance to the Mobile Device Fundamentals Protection Profile.

Note: The Samsung Galaxy Note 4 with Qualcomm processors has been updated to be compliant to the MDFPPv2 and so is not included in this document. The Note 10.1 2014 Edition tablet will not be receiving Android 5 and so is removed from this document.

The model numbers and evaluated versions of the mobile devices are as follows:

Device Name	Base Model Number	Android Version	Kernel Version	Build Number
Galaxy Note 3	SM-N900	5.0.2	3.4.0	LRX22G
Galaxy NotePRO	SM-N905	5.0.2	3.4.0	LRX22G
Galaxy S5	SM-G900	5.0.2	3.4.0	LRX22G
Galaxy Note 4 (System LSI)	SM-N910	5.0.2	3.10.9	LRX22G
Galaxy Note Edge (Qualcomm)	SM-N915	5.0.2	3.10.0	LRX22G
Galaxy Note Edge (System LSI)			3.10.9	
Galaxy Alpha (Qualcomm)	SM-G850	5.0.2	3.4.0	LRX22G
Galaxy Alpha (System LSI)			3.10.9	
Galaxy Tab S 8.4	SM-T707	5.0.2	3.4.0	LRX22G
Galaxy Tab S 10.5	SM-T807	5.0.2	3.4.0	LRX22G
Galaxy Tab Active	SM-T360	5.0.2	3.4.0	LRX22G

These devices may include an additional letter or number at the end of the name (such as SM-N900V) that denotes the device is for a specific carrier (V = Verizon Wireless).

The Galaxy S4 had unique model numbers for each US carrier as listed below:

Carrier	Galaxy S4
Verizon	SCH-I545
AT&T	SGH-I337
Sprint	SPH-L720
T-Mobile	SGH-M919
US Cellular	SCH-R970
International	GT-I9505

The Galaxy S4 has the following build information:

- Android version: 5.0.2
- Kernel version: 3.4.0
- Build number: LRX22G

The following table shows the Security software versions for each device.

Device Name	MDF Version	MDF Release	VPN v1.4 Release
Galaxy S4	1.1	4	4.1
Galaxy Note 3	1.1	4	4.1
Galaxy NotePRO	1.1	4	4.1
Galaxy S5	1.1	4	4.1
Galaxy Note 4 (System LSI)	1.1	6	4.1
Galaxy Note Edge (Qualcomm)	1.1	6	4.1

Device Name	MDF Version	MDF Release	VPN v1.4 Release
Galaxy Note Edge (System LSI)	1.1	6	4.1
Galaxy Alpha (Qualcomm)	1.1	6	4.1
Galaxy Alpha (System LSI)	1.1	6	4.1
Galaxy Tab S 8.4	1.1	6	4.1
Galaxy Tab S 10.5	1.1	6	4.1
Galaxy Tab Active	1.1	6	4.1

The MDF version number is broken into two parts as the claimed MDFPP has been updated in the latest devices. For example, the Galaxy Alpha would show “MDF v1.1 Release 6”.

1.2 Terminology/Glossary

ADB	Android Debug Tool
ADT	Android Development Tools
API	Application programming interface
BYOD	Bring-Your-Own-Device
CA	Certification Authority
MDM	Mobile Device Management
ODE	On-Device Encryption
SDK	Samsung Enterprise Software Development Kit
SSL	Secure Socket Layer
VPN	Virtual Private Network

2 Guidance Overview

The Samsung model to maintain a secure mobile device environment involves a number of parties. These include:

- Approved Mobile Device Management (MDM) software developers;
- Samsung Approved Carriers;
- Enterprise and Mobile Device Administrators; and
- Enterprise Users.

As a result, a number of elements of maintaining a secure mobile environment are reliant on parties outside of Samsung and are not detailed in this documentation.

This document has been designed for Enterprise and Mobile Device Administrators and therefore provides guidance on the configuration and deployment of a Mobile Enterprise solution using Samsung devices. Guidance for device users is provided in a separate document.

3 Introduction

3.1 Overview

The TOE is a mobile operating system based on Android 5 with modifications made to increase the level of security provided to end users and enterprises. The TOE is intended to be used as part of an enterprise messaging solution providing mobile staff with enterprise connectivity.

The TOE combines with a Mobile Device Management (MDM) solution that enables the enterprise to watch, control and administer all deployed mobile devices, across multiple mobile service providers as well as facilitate secure communications through a VPN. This partnership provides a secure mobile environment that can be managed and controlled by the environment and reduce the risks that can be introduced through a Bring-Your-Own-Device (BYOD) model.

The Samsung Enterprise Software Development Kit (SDK) builds on top of the existing Android security model by expanding the current set of security configuration of options to more than 390 configurable policies and including additional security functionality such as application blacklisting. The ability to set these policies is based on the capabilities of the MDM.

3.2 Evaluated Capabilities

The product provides a significant amount of security capabilities with the core capabilities being included within the common criteria evaluation including:

Security feature	Description
Device data protection. The TOE provides security functionality to protect data at rest.	On Device Encryption (ODE). The TOE has the ability to encrypt data on the device using AES 256.
	Removable storage encryption. The TOE can encrypt all file placed onto, or already reside on, removable storage attached to the device.
Application Management. The device provides a number of security functions to manage device software.	Application resource restrictions. All applications are run within a controlled environment that limits applications to only accessing only authorized data and resources.

Security feature	Description
<p>Access Control. The device can implement access control that reduces mobile user permissions and assists in reducing unauthorized access.</p>	<p>Device lock. The TOE can be configured to automatically lock after a defined period of inactivity (1 to 60 minutes) limiting access to device functions except those that are explicitly authorized such as emergency calls.</p>
	<p>Local wipe. The TOE has the ability to wipe encryption keys/data on a device after an administratively defined amount of authentication attempts are surpassed.</p>
	<p>Credential complexity. The TOE can enforce enterprise password policies forcing users to use a defined level of complexity in device passwords.</p>
	<p>Privileged access. The TOE can be configured to restrict mobile user’s access to privileged functions such as device configurations.</p>
	<p>Hotspot Control. The TOE can be configured to act as a hotspot for sharing Internet access to other devices.</p>
	<p>Wireless network settings. The wireless network configuration of the TOE can be specified, providing requirements or pre-loaded networks.</p>
	<p>APN Settings. The TOE can be configured to connect to different cellular networks through the APN configuration.</p>
<p>Enterprise device management. Enterprise administrators can control mobile endpoint configurations and wipe device if needed.</p>	<p>Remote wipe. An enterprise administrator can send a message to the TOE to wipe all local storage and SD card.</p>
	<p>Security policy. The TOE can be configured by a Mobile Device Management solution that supports the Samsung Enterprise SDK.</p>

3.3 SAFE/KNOX Management API

Samsung provides an extensive set of management APIs to fully control a Samsung device within your environment. To obtain more information about specific APIs and capabilities provided by Samsung, sign up for an account at <http://www.samsungmobileb2b.com> and request access to the MDM API.

4 Deployment process

The specific deployment model is dependent on a number of factors including:

- Chosen MDM solutions supported architecture;
- Preferred mobile operating methods (often as a result of business culture);
- Financial considerations;
- Enterprise technical capability
- Risk appetite of the business; and
- Existing technological capital.

4.1 Enterprise architecture

The first step in deploying Samsung devices is to decide on both a Mobile Device Management solution and an appropriate architecture. These two selections may be done in either order depending on the preferences of the organization. In some organizations there may be a preferred architecture, and as a result an MDM solution is based on its compatibility with that architecture, in others, the architecture will be chosen to match the already chosen MDM.

There are three core architectures:

- Enterprise based deployment;
- Cloud based deployment; and
- Hybrid approach.

However, only the 'enterprise based deployment' architecture will be described in detail. The 'cloud based deployment' and the 'hybrid approach' are not covered by this evaluation, though they are certainly options which can be employed. Ideally any MDM solution will have been evaluated to the requirements of the MDMPP (Mobile Device Management Protection Profile).

4.1.1 Enterprise based deployment

In this architecture the enterprise environment must provide all of the services required to operate and manage devices. The basic components of this model include:

- **Mobile Device Management Solution**

The Mobile Device Management (MDM) Solution secures monitors, manages and supports mobile devices deployed across companies. By controlling and protecting the data and configuration settings for all Android devices in the corporate network business security risks are reduced. Samsung offers an extensive range of different solutions. Every Mobile Device Management solution supports the Samsung Enterprise SDK.

Android devices combine with a Mobile Device Management solution. This partnership provides a secure mobile environment that can be managed and controlled by the environment and reduce the risks that can be introduced through a Bring-Your-Own-Device (BYOD) model.

- **Secure tunnel termination**

A secure VPN tunnel should be initialized between the managed Android devices and the Enterprise Environment to prevent unauthorized access to enterprise resources. The connection should be based on certificates deployed on the Android user devices. Ideally mutual authentication is deployed, meaning that both the Android user devices authenticate themselves with a certificate but also the gateway to the enterprise environment. Mutual authentication serves to prevent Android user devices to login into an unauthorized enterprise network and on the other hand prevents the unauthorized login of untrusted devices into the enterprise environment.

The tunnel establishment should be terminated in case of invalid certificates. Further, an idle VPN session should be terminated after a certain time span.

- **Directory services**

The directory services should be set up to store, organize and provide access to information in a directory.

- **Business applications**

Business applications allow enterprise users to fulfill or access certain business tasks pertinent to requirements. This may include management tools, accounting utilities and contact management software/solutions.

- **Certificate services**

A certificate service must be implemented that manages all certificate needs throughout the enterprise environment. This includes issuing new Android device user certificates that are needed to facilitate a secure communications through a VPN.

The advantages of this solution are that there will be no issue with data sovereignty plus the enterprise increases its control of the over the managed devices as well as the deployed environment. The downside is the increased costs for managing this enterprise environment.

Figure 1 shows an example of a high level design of an enterprise based environment.

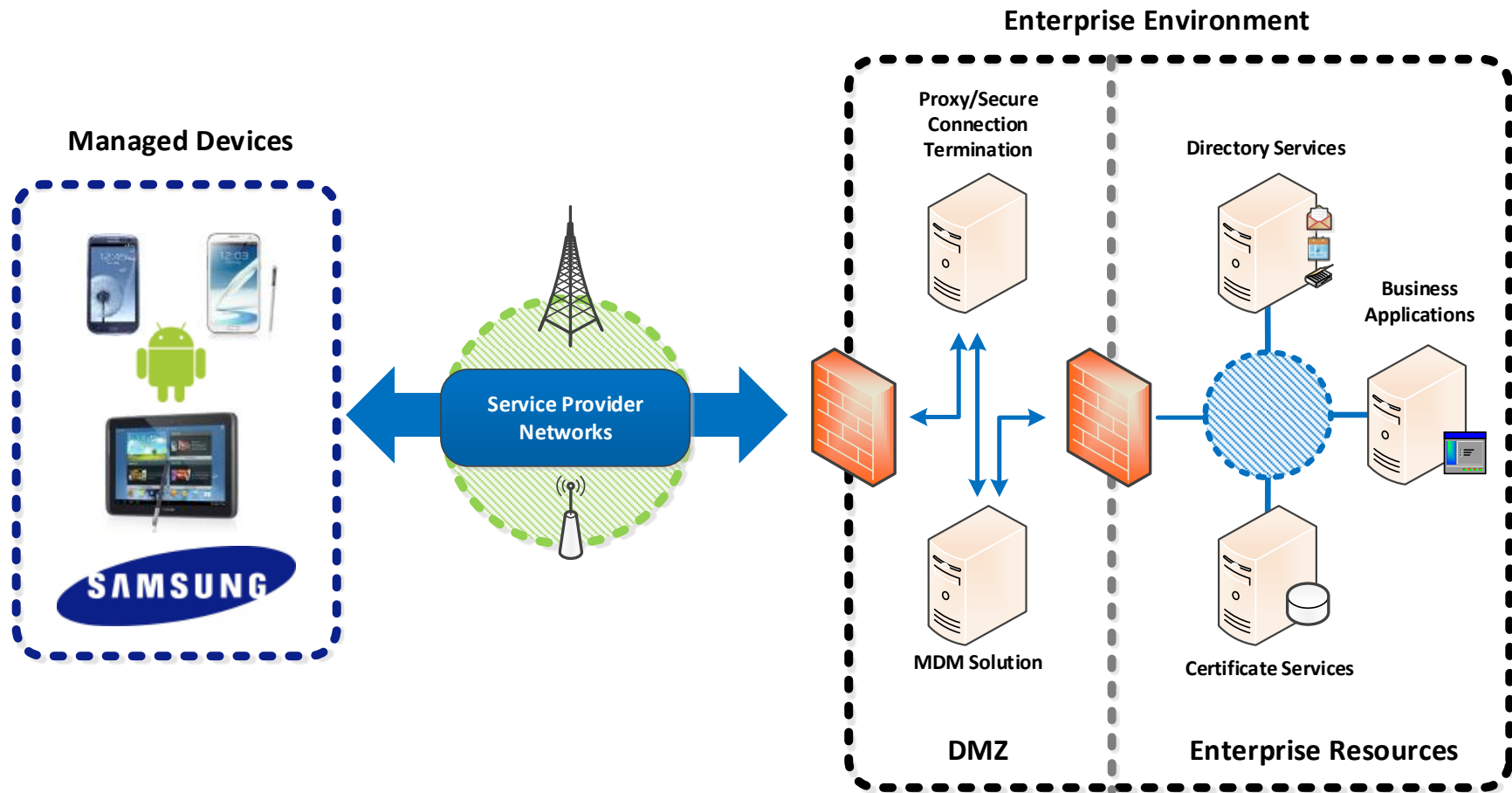


Figure 1 – Enterprise Environment

4.1.2 Compatible Mobile Device Management (MDM) solutions

The security configuration must be set through the MDM for the evaluated configuration. All other configuration items can be changed without changing the evaluated configuration. The evaluated configuration is provided in Section 4.3.2.

4.2 Secure preparation of the Enterprise Environment

Prior to the configuration of a Samsung Android user device, the enterprise environment must be securely prepared.

In particular, the guidance for the Mobile Device Management Solution should be followed. This documentation provides information about the capability to remotely manage devices and perform functions such as sending remote wipe messages. Further, it includes, or provides directions to implement, infrastructure to support secure transmissions with devices.

For an enterprise deployment of Samsung Android devices that is suitable for organizations working with official data, administrators should:

- Deploy and configure the requisite network components as described above
- Procure and set up an MDM server with a client that implements the SAFE APIs and is able to enforce all the settings given in the Common Criteria Configuration section below.

Section 4.3.2 provides more detailed information about the options the MDM must support in order to configure the devices in the evaluated configuration.

4.3 Secure installation of Samsung Android user devices

This section follows up on Section 4.2 and provides information on how an Enterprise and Mobile Device Administrator securely installs a Samsung Android user device.

For an enterprise deployment of Samsung Android devices that is suitable for organizations working with official data, administrators should:

- Perform the device deployment process described in on Section 4.3.1; and
- Create MDM security profiles for the devices in line with the guidance given in the Common Criteria Configuration (Section 4.3.2) and associate these profiles with the devices.

4.3.1 Device deployment process

The following steps should be followed to provision each end user device onto the enterprise network to prepare it for distribution to end users.

1. Install the MDM agent application, and enroll the device into the MDM.
2. Provision client certificates by either:
 - a. Provisioning the client certificates using a locally-enrolled MDM server;
 - b. Deploying the Android Development Tools (ADT) bundle and device-specific USB drivers onto a dedicated provisioning terminal. This will allow the client certificates to be manually deployed onto the device via the Android Debug Tool (ADB). Note that USB debugging should be disabled once provisioning is complete.

The certificates required for an MDM deployment are:

- i. Enterprise CA certificate (used to validate the server certificates presented by the VPN endpoint and reverse proxy),
 - ii. VPN client certificate (for authentication to the enterprise VPN endpoint),
 - iii. SSL client certificate (for authentication to the reverse proxy for intranet services).
3. Install applications required for enterprise productivity.
4. Ensure that only trusted applications are installed and enabled on the device (disable unnecessary applications including Google Play).
5. Configure on-device security settings (please refer also to Section 4.3.2).
6. Configure the VPN client to connect to the enterprise VPN endpoint, using the device-specific client certificate that has been loaded onto the device. Enable 'Always-On' VPN
7. Configure the email client to connect to the enterprise server using client certificate authentication.

4.3.2 Common Criteria Configuration

The following table shows settings which must be enabled to a specific value (or range of values) to meet the specification of the evaluation. The evaluated security configuration consists of both Samsung

specific (Samsung Enterprise SDK) as well as Android specific settings. Please also follow the guidance provided in [MDMG] to set the options listed below. The Classes or Methods used to configure these settings are provided for reference and can be used to verify whether the MDM will support your needs.

The following sections specify the required settings that must be enabled/configured to place a device into the evaluated configuration.

Note: Methods that can meet the requirement that are provided by Android natively are listed in *italics*. In most cases there is a corresponding Samsung SAFE API as well. When this is the case, the two Methods are **highlighted** to show the correspondence between the options. In these cases the MDM may use either call to achieve the same result.

4.3.2.1 Configurations with and without KNOX 2 Containers

Samsung devices include an integrated capability to create separate containers within the device. These are enabled by the KNOX 2 components included in Samsung Android. When a KNOX 2 container is configured it provides a separated area of the device which can have its own apps and data which is not accessible from the “normal” area. The KNOX 2 containers can be used to separate different apps, such as in BYOD scenarios where an enterprise could place their data into a separate container on the user’s device.

A Samsung device can be placed into an evaluated configuration both with and without a KNOX 2 container. For organizations that do not need to segment the device, a configuration can be used without creating any KNOX 2 containers. For organizations that have a need for data separation, KNOX 2 containers can be created and still be in an evaluated configuration.

The setting listed below show the APIs which are used to place a device into an evaluated configuration for either case. When configuring a device to use a KNOX 2 container all settings marked (All) and those marked (KNOX) are used. When using a device without a KNOX 2 container only the settings marked (All) need to be used.

Note: KNOX 2 containers implement many of the same APIs as are available to non-containers (such as hardware state configurations). Policies in KNOX containers are tied specifically to those containers as part of the KNOX Premium API configuration. All KNOX 2 APIs specified are part of the KNOX Premium set of APIs and require a KNOX 2 license to be used.

4.3.2.2 CC Mode Settings (All)

To place a device into the evaluated configuration the CC Mode must be enabled.

Setting	Value	Description	Class or Method
CC Mode	Enable	This setting enables FIPS-validated crypto, disables USB connectivity in recovery mode & only allows FOTA updates to the system	setCCMode()

CC Mode is a new function that is not yet widely supported by MDM vendors. To facilitate customers in enabling CC Mode, Samsung has provided a stand-alone app that can enable this setting locally on the device.

The CCMODE.apk can be downloaded from Samsung [here](#). You will need to register for an account. Click the Register link and follow the prompts to register for your account (it will have multiple steps including email verification). Each level can access the CC tools; contact your account manager for more information if you are unsure which level to register at.

Once you have completed the account registration login using your credentials.

After logging in, click the **Support** tab at the top. Then click the **View your available tools** link on the left side. Look for **Common Criteria Mode APK** and click the **Launch** button below it.

From this page you can see the list of applications provided with each validated device as well as the CC Mode application at the bottom.

Before installing the CC Mode app, you must enable Unknown Sources for applications as the app will not be installed from the Google Play Store. This can be achieved by going to **Settings/Security/Unknown sources**. Checking this box will prompt to confirm the enabling of Unknown sources due to the possibility of vulnerabilities in being able to install apps from outside of the Play Store. Download the APK to your device and install the app by opening the APK.

New to Samsung KNOX?

Create a Samsung KNOX web account to try a Samsung Solution including:

KNOX Express

Your mobile management solution made easy. [Learn more](#)

Register

KNOX Premium

Your end-to-end secure mobile platform solution. [Learn more](#)

Register

KNOX Workspace

Enjoy the freedom of one device for work and play. KNOX Workspace delivers the security needed for enterprise mobility. [Learn more](#)

Register

Note: Once installed, Unknown Sources can be disabled.

To enable CC Mode, find the app (named CC Mode). Launch the app and choose Activate to enable the application to make changes to the device settings. Once activated, select Turn on CCMODE. Once CC Mode is enabled, the device will be configured such that 5 unsuccessful login attempts will force a factory reset on the device wiping all data. This setting can be edited by the MDM once CC Mode has been enabled.

Note: Once a device has been placed into CC Mode, the only way to disable it is to perform a factory reset or to connect to an MDM which can disable it.

Once CC Mode has been enabled, the app can be removed from the device. To remove the app from the device, you must first disable it as a Device Administrator. This can be done through **Settings/Security/Device Administrators**. Unselect the CC Mode app and choose Deactivate. The app can now be removed through the Application Manager or through the MDM.

4.3.2.2.1 CC Mode and Approved Cryptography

Part of the Common Criteria-evaluated configuration is the availability of approved cryptographic engines for use by the system and applications. Samsung has chosen to utilize FIPS 140-2-validated cryptographic modules on its devices for the Common Criteria configuration.

Samsung provides three cryptographic modules on the evaluated devices:

- Samsung Kernel Cryptographic Module (FIPS certificate #1915)
- OpenSSL FIPS Object Module (FIPS certificate #1747)
- OpenSSL Object Module (not FIPS-validated)

By default on a device (i.e. out of the box), the Samsung Kernel Cryptographic Module and the OpenSSL Object Module are in use. To place the device into the evaluated configuration, CC Mode must be enabled. When CC Mode is enabled, the OpenSSL FIPS Object Module replaces the OpenSSL Object Module in use. At this point only approved cryptographic functions are used on the device.

Note: Only the Samsung Kernel Cryptographic Module and the OpenSSL FIPS Object Module have been evaluated in the configuration. While it is possible to use all other settings without enabling CC mode, doing so will not utilize the evaluated cryptographic modules and therefore will not be the evaluated configuration.

It is also possible that some applications may implement their own cryptography. Only the two cryptographic modules provided with the device are validated, any other cryptography must be evaluated on its own.

4.3.2.2.2 CC Mode Status

CC Mode has the following statuses:

Status	Description
Disabled	CC Mode has not been turned on
Enforced	CC Mode has been turned on but some of the required settings or configurations have not been set
Enabled	CC Mode has been turned on and all required settings and configurations have been set

The CC Mode status can be seen by going to **Settings/About phone/Software Security Version**. Clicking on the item will show the current status.

4.3.2.2.3 CC Mode Requirements/Configurations

When CC Mode is first turned on, it changes the status from Disabled to Enforced. To change the status to Enabled, the following settings must be configured:

1. Enable the Maximum Password Failure Policy
2. Enable On Device Encryption (ODE)
3. Enable SD Card Encryption
4. Enable CRL Checking

Note: To be Enabled, not only must the encryption settings be set, the storage must have been encrypted by the user.

4.3.2.3 Encryption Settings (All)

There are two sets of encryption settings, one for internal storage and one for external (SD Card) storage. Both must be enabled, even if no SD Card will be used in the device.

Setting	Value	Description	Class or Method
On Device Encryption (ODE)	Enable	This encrypts all internal storage media	setInternalStorageEncryption() setStorageEncryption() setRequireDeviceEncryption()

Setting	Value	Description	Class or Method
SD Card Encryption	Enable	This encrypts all external (SD Card) storage media	setExternalStorageEncryption() setRequireStorageCardEncryption()

4.3.2.4 Authentication Settings (All)

Setting	Value	Description	Class or Method
Max Password Failures (Wipe)	100 or less	The maximum number of times a password can be entered before the device is wiped	<i>setMaximumFailedPasswordsForWipe()</i>

4.3.2.5 Certificate Revocation Settings (All)

Setting	Value	Description	Class or Method
Certificate Revocation Checking	Enable for All apps	Specifies that CRL checking is enabled for all apps on the device	isRevocationCheckEnabled()

4.3.2.6 KNOX Container Policy (KNOX)

Setting	Value	Description	Class or Method
Create Container	Policy of container	Specifies the policy to be used when creating the container	Class: KnoxContainerManager createContainer()

Setting	Value	Description	Class or Method
Configure Container Policy	Container Type and policy settings 1-99 for failed password settings	Creates a policy template for a container. This is the default settings for a new container	Class: KnoxConfigurationType KnoxConfigurationType() setMaximumTimeToLock() setPasswordMinimumLength() setPasswordQuality() setPasswordMinimumSymbols() setMaximumFailedPasswordsForWipe() setMaximumFailedPasswordsForDeviceDisable() addConfigurationType()

Note: When set in a KnoxConfigurationType(), the setMaximumFailedPasswordsForDeviceDisable() or the setMaximumFailedPasswordsForWipe() settings will disable or wipe the container, not the whole device.

4.3.3 Other Common Criteria Configurations

The settings in this section have been evaluated, but no specific configuration is required to place the device into the evaluated configuration. They are part of the management functions that are included and can be configured as needed for your specific environment.

4.3.3.1 Authentication Settings (All)

The settings here deal with passwords and other authentication-related settings.

Setting	Value	Description	Class or Method
Password Length	6-16	Minimum password length	<i>setPasswordMinimumLength()</i>

Setting	Value	Description	Class or Method
Password Complexity	Set min # of characters or max sequences	Settings to require different types of characters in a password	setMaximumCharacterOccurrences() setMaximumCharacterSequenceLength() setMaximumNumericSequenceLength() setMinPasswordComplexChars() setMinimumCharacterChangeLength() <i>setPasswordMinimumLetters()</i> <i>setPasswordMinimumLowerCase()</i> <i>setPasswordMinimumNonLetter()</i> <i>setPasswordMinimumNumeric()</i> <i>setPasswordMinimumSymbols()</i> <i>setPasswordMinimumUpperCase()</i>
Password Expiration		Specify the maximum age of a password before it must be changed	setPasswordExpires() setPasswordExpirationTimeout()
Password Entry Visible	Disable	This prevents entered passwords from being displayed on the screen	setPasswordVisibilityEnabled() setScreenLockPatternVisibilityEnabled()

Note: Once a password is required there is no way to bypass the user authentication.

4.3.3.2 Admin Settings (All)

Setting	Value	Description	Class or Method
Allow new admin	Enable / Disable	An MDM can prevent new admins from getting installed / activated on the device	Class: AdvancedRestrictionPolicy preventNewAdminActivation() preventNewAdminInstallation()

4.3.3.3 Lock screen Settings (All)

Setting	Value	Description	Class or Method
Inactivity Timeout Lock Period	1 to 60 minutes	This specifies how long the device will remain unlocked after usage has stopped	setPasswordLockDelay() setMaximumTimeToLock()
Remote Lock	Enable	Will remotely lock the device immediately	lockNow()
Unlock Banner	Up to 256 characters	Text to display on the lock screen before login. Text will scroll at XX characters	changeLockScreenString()

4.3.3.4 Radio Control (including APN) Settings (All)

Setting	Value	Description	Class or Method
Control Bluetooth	Enable/Disable	Enable or Disable access to Bluetooth	allowBluetooth()
Control Wi-Fi	Enable/Disable	Enable or Disable access to Wi-Fi	allowWiFi()
Control NFC	Enable/Disable	Enable or Disable access to NFC	setEnabledNFC()
Control Cellular Data Access	Enable/Disable	Enable or Disable access to Cellular Data (not Voice)	setCellularData()
Control Location Provider	Enable/Disable	Enable or Disable access to Location services on the device	setLocationProviderState()
Access Point Name Settings		Configure specific APN settings for the device	Class: ApnSettingsPolicy Class: ApnSettings

4.3.3.5 Wi-Fi Settings (All)

Setting	Value	Description	Class or Method
Specify SSIDs for Wi-Fi	SSID values	Enable/Disable Wi-Fi restrictions based on acceptable SSID values. Both white and black listing of networks is supported.	activateWifiSsidRestriction() addBlockedNetwork() addWifiSsidToBlackList() addWifiSsidToWhiteList()
Set WLAN CA Certificate	CA Cert	Specify trusted CAs for accepting WLAN server certificates	setNetworkCaCertificate()
Set Wi-Fi security type	WLAN security	Specify the type of security required on a WLAN connection (i.e. open, WEP, WPA, etc)	setMinimumRequiredSecurity()
Wi-Fi authentication protocols	WLAN security	Specify the values required to connect to EAP-TLS connections	setNetworkAnonymousIdValue() setNetworkClientCertificate() setNetworkIdentityValue() setNetworkPhase2() setTlsCertificateSecurityLevel()
Wi-Fi client credentials	WLAN client credentials	Specify the client credentials to access a specified WLAN	setNetworkPSK() setNetworkPassword() setNetworkClientCertificate() setNetworkPrivateKey() setNetworkWEPKey1-4() setNetworkWEPKeyId()

Note: In CC Mode, LEAP, PEAP and FAST modes are disabled due to their use of non-FIPS algorithms.

4.3.3.6 Hotspot/Tethering Settings (All)

Setting	Value	Description	Class or Method
Specify if Wi-Fi Hotspot can be modified by user	Enable/Disable	Enable/Disable whether the user can edit the Hotspot settings	isWifiApSettingUserModificationAllowed()
Specify Hotspot settings	SSID, Security Type, Password	Specify the settings for the Hotspot	setWifiApSetting()
Tethering (Wi-Fi, USB and Bluetooth)	Enable/Disable	Controls ability to use the device as a Wi-Fi hotspot to share its Internet connection. setTethering() controls access to all other tethering options (if that is disabled no others are allowed).	setTethering() setBluetoothTethering() setUsbTethering()

4.3.3.7 Services Control Settings (All)

Setting	Value	Description	Class or Method
Camera control	Enable/Disable	Enable or Disable access to Camera	setCameraState() setCameraDisabled()
Microphone control	Enable/Disable	Enable or Disable access to microphone	setMicrophoneState()

Setting	Value	Description	Class or Method
Voice control	Enable/ Disable	Enable or Disable access to S-Voice or the Voice Dialer controls. This does not prevent access to other voice-controlled apps, only the Samsung-provided ones.	allowSVoice() disableVoiceDialer() allowVoiceDialer()
Allow FOTA Updates	Enable/ Disable	Specifies whether the device can check and receive OTA updates. This can be used to block auto-updates until they have been approved.	allowOTAUpgrade()
Allow Mounting over USB	Enable/ Disable	Enable or Disable the mounting of device storage over USB. When disabled USB can only be used for charging.	setUsbMediaPlayerAvailability()
Allow SD Card to be mounted	Enable/ Disable	Enable or Disable the ability to mount (and use) an SD Card. Also possible to mount SD Card as read-only.	setSdCardState() allowSDCardWrite()
Developer Mode	Enable/ Disable	Enable or Disable USB Debugging for developer access	allowDeveloperMode() setUsbDebuggingEnabled()

Setting	Value	Description	Class or Method
Setting Automatic Time	Enable/Disable	Enable or Disable the use of Carrier Time on the device. If this is disabled, then the time is handled solely on the device with no external checks.	setAutomaticTime()

4.3.3.8 Notification Settings (All)

Setting	Value	Description	Class or Method
Blacklist Application Notification Mode	Block All, Block Text, Block Text & Sound	Specify if notifications are blocked, what level of blocking should be done. Block all notifications, Text (status bar) notifications only, or only Text & Sound.	setApplicationNotificationMode()
App Notification Lists	App Names	Whitelist and Blacklist of apps that can override default notifications. Blacklist apps follow that notification mode setting	addPackagesToNotificationBlackList() removePackagesFromNotificationBlackList() addPackagesToNotificationWhiteList() removePackagesToNotificationWhiteList()

4.3.3.9 Messaging (SMS) Settings (All)

Setting	Value	Description	Class or Method
Allow incoming messages	Enable/Disable	Allow user to receive incoming SMS/MMS messages	allowIncomingMms() allowIncomingSms()

Setting	Value	Description	Class or Method
Allow outgoing messages	Enable/ Disable	Allow user to send SMS/MMS messages	allowOutgoingMms() allowOutgoingSms()

4.3.3.10 Certificate/Key Management Settings (All)

Setting	Value	Description	Class or Method
Import Certificates	Certs	Import CA Certificates into the Trust Anchor Database or the credential storage. The choice of storage is dependent on the type of certificate being imported.	installCertificate() installCertificatesFromSdCard() installCertificateWithType() installClientCertificate() (for VPN)
Remove Individual Certificates	Cert names	Remove Individual certificates from the database or credential store	removeCertificate()
Remove All Certificates		This will clear all imported Certificates (except the built-in TAD)	clearInstalledCertificates()

4.3.3.11 Application Management Settings (All)

Setting	Value	Description	Class or Method
Install Apps	App name	This allows an application to be installed on the device	installApplication()
Uninstall Apps	App name	This allows applications to be uninstalled from the device.	uninstallApplication() uninstallApplications() (bulk list of apps at one time)

Setting	Value	Description	Class or Method
Control app uninstall	App name	Enables / disables user uninstall of specified application	Class: ApplicationPolicy setApplicationUninstallationDisabled() setApplicationUninstallationEnabled()
Control Google Play	Enable / Disable	Allows installation of applications from Google Play	public class ApplicationPolicy enableAndroidMarket() disableAndroidMarket ()
Control Unknown Sources	Enable / Disable	Allows installation of application from unknown sources	public class RestrictionPolicy setAllowNonMarketApps

Black listing is preventing specific applications from being able to be installed on the device. This is done using the full name of the application (such as com.android.testingapp).

The method for configuring black lists is highly dependent on the MDM solution chosen. Please refer to the MDM specific guidance [MDMG] on exactly how to set these policies.

4.3.3.12 Remote Wipe Settings (All)

Setting	Value	Description	Class or Method
Remotely wipe the device	True	Remotely wipe the data stored on the device. This will perform a factory reset.	wipeDevice() wipeData()

4.3.3.13 Lock Screen Settings (KNOX)

Setting	Value	Description	Class or Method
Password Length	6-16	Minimum container password length	setPasswordMinimumLength()

Setting	Value	Description	Class or Method
Password Complexity	Set min # of characters or max sequences	Settings to require different types of characters in the container password	setMaximumCharacterOccurrences() setMaximumCharacterSequenceLength() setMaximumNumericSequenceLength() setMinPasswordComplexChars() setMinimumCharacterChangeLength()
Password Expiration	(in seconds)	Specify the maximum age of the container password before it must be changed	setPasswordExpires()
Password Entry Visible	Disable	This prevents entered passwords from being displayed on the container lockscreen	setPasswordVisibilityEnabled() setScreenLockPatternVisibilityEnabled()

4.3.3.14 Container Authentication Settings (KNOX)

Setting	Value	Description	Class or Method
Inactivity Timeout Lock Period	1 to 60 minutes	This specifies how long the container will remain unlocked after container usage has stopped	setPasswordLockDelay()
Remote Lock	Enable	Will remotely lock the container immediately	lock()

4.3.3.15 Services Control Settings (KNOX)

Setting	Value	Description	Class or Method
Camera control	Enable/Disable	Enable or Disable access to Camera inside the container	setCameraState()

Setting	Value	Description	Class or Method
Microphone control	Enable/ Disable	Enable or Disable access to microphone inside the container	setMicrophoneState()

4.3.3.16 Application Management Settings (KNOX)

Setting	Value	Description	Class or Method
Install Apps	App name	This allows an application to be installed to the container	installApplication()
Uninstall Apps	App name	This allows applications to be uninstalled from the container.	uninstallApplication() uninstallApplications() (bulk list of apps at one time)
Control app uninstall	App name	Enables / disables user uninstall of specified application from the container	Class:.ApplicationPolicy setApplicationUninstallationDisabled() setApplicationUninstallationEnabled()

4.3.3.17 Notification Settings (KNOX)

Setting	Value	Description	Class or Method
Blacklist Application Notification Mode	Block All, Block Text, Block Text & Sound	Specify if container notifications are blocked, what level of blocking should be done. Block all notifications, Text (status bar) notifications only, or only Text & Sound.	setApplicationNotificationMode()

Setting	Value	Description	Class or Method
App Notification Lists	App Names	Whitelist and Blacklist of apps that can override default container notifications. Blacklist apps follow that notification mode setting	addPackagesToNotificationBlackList() removePackagesFromNotificationBlackList() addPackagesToNotificationWhiteList() removePackagesToNotificationWhiteList()
Email notifications	Enable/Disable	Sets whether the email app notifications are displayed for the container	setEmailNotificationsState()

4.3.3.18 Container Sharing Settings (KNOX)

Setting	Value	Description	Class or Method
Application & File movement	Enable/Disable	Define whether apps can be moved into or out of the Container	allowMoveAppsToContainer() allowMoveFilesToContainer() allowMoveFilesToOwner()
Application Data Sync	Enable/Disable for apps	Define whether specific apps can share data between the container & outside	setAllowChangeDataSyncPolicy() <ul style="list-style-type: none"> ● Contacts ● Calendar

4.3.3.19 Container Browser Settings (KNOX)

Setting	Value	Description	Class or Method
Browser Web Proxy	IP / Port	Web proxy configuration for default container browser application	class BrowserPolicy setHttpProxy()

4.3.4 Additional notes

Samsung Android devices are usually configured by default to send anonymous usage data (including location, device ID etc.) to Google and Samsung servers. This can be disabled through device settings and will need to be enforced through procedural controls.

Samsung Android devices do not need to be associated with a Google account to operate as required within the enterprise. For example, it is still possible to receive push notifications through Google Cloud Messaging. SAFE MDM APIs can be used to prevent users from signing in to these services (see [MDMG]).

4.4 Secure Delivery

While a Samsung device requires initial configuration before it can be added to the enterprise environment, it is also critical to ensure that the device is received prior to configuration in a secure manner, free from tampering or modification.

It is very important that the devices to be deployed into the enterprise are obtained from reputable carriers to reduce the likelihood that tampering of devices may occur.

Upon receipt, the boxes containing the device should have both a tracking label and two labels placed at either end of the box to indicate whether the box has been opened prior to delivery. If these seals are broken, do not accept the device and return it to your supplier.

The tracking label should look similar to Figure 2 - Tracking label, while the two tamper labels should appear similar to Figure 3 - Security Seal (Black) or Figure 4 - Security Seal (White).

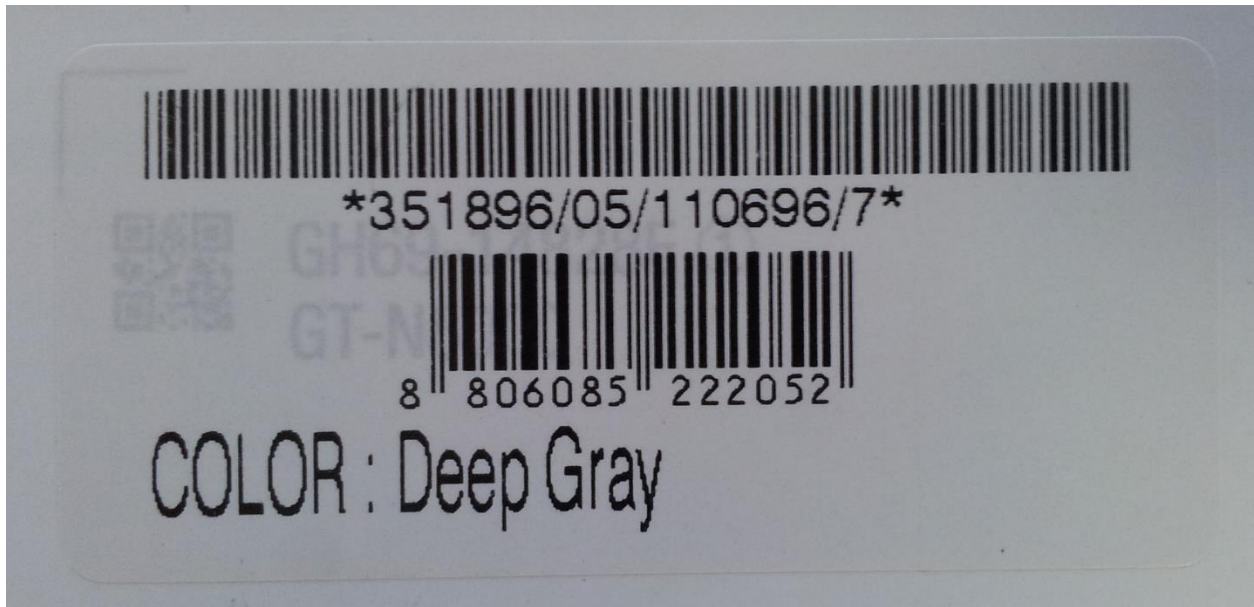


Figure 2 - Tracking label



Figure 3 - Security Seal (Black)

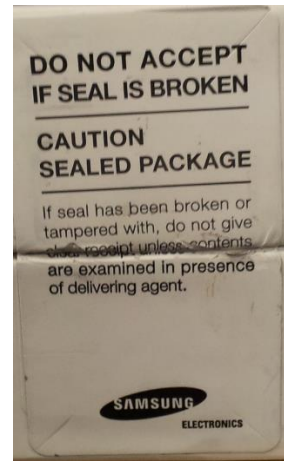


Figure 4 - Security Seal (White)

4.4.1 Evaluation version

There are a number of components to determining the device that is being used and the components on that device (such as the operating system version, the build version, etc.). These are all contained under **Settings/About device**. The following are version information that can be found:

- **Model number** – this is the hardware model (this is carrier specific, so for example a Samsung Galaxy S4 on Verizon Wireless has a different model number than on AT&T)
- **Android version** – this is the Android OS version
- **Build number** – this is the specific binary image version for the device
- **Security Software Version** – this shows the Common Criteria evaluations and the version of the software components related to those evaluations on the device

For the Common Criteria evaluation for the mobile device, this will show:

MDF v1.1 Release **XYZ**

where **XYZ** is the version number of the software that has been validated. As other validations are performed on the device they will also be listed here.

4.4.2 Pre-packaged Software Versions

Samsung Android devices come with large amounts of software apps to provide the full breadth of functionality expected by the customer. Some of the apps come from Google, some from Samsung, and others from the cellular carrier. For a list of the apps and their versions contained on a specific device, visit this [website](#) and select the device you are using. This will provide a complete list of the software installed on the evaluated device.

4.4.2.1 Software Versions on Device

To verify the versions of any software on the device (compared to the list from the website), open **Settings/Application manager**. Under the heading **All**, you will see every application on the device (both those that are pre-installed and any you have installed). Selecting an application will display its properties. The version number is shown at the top under the name.

Note: Using adb (USB debugging must be enabled to use adb) it is possible to extract all package version information at once.

4.5 Secure Updates

Once a device has been deployed, it may be desirable to accept updates to the software on the device to take advantage of the latest and greatest features of Samsung Android. Updates are provided for devices as determined by Samsung and the carriers based on many factors.

When updates are made available, they are signed by Samsung with a private key that is unique to the device/carrier combination (i.e. a Galaxy S4 on Verizon will not have an update signed with the same key as a Galaxy S4 on AT&T). The public key is embedded in the bootloader image, and is used to verify the integrity and validity of the update package.

When updates are made available for a specific device (they are generally rolled out in phases across a carrier network), the user will be prompted to download and install the update (see the User Guide for more information about checking for, downloading and installing the update). The update package is checked automatically for integrity and validity by the software on the device. If the check fails the user is informed that there were errors in the update and the update will not be installed.

4.5.1 Allowed Update Methods

When CC Mode is enabled, only Firmware Over the Air (FOTA) updates are allowed to be installed on the device. Other methods for installing updates (such as ODIN or Samsung KIES) are blocked and cannot be used to update the firmware. This provides insurance against local, physical attacks that could change the software unknowingly.

4.5.2 Blocking Updates

It is possible to block FOTA updates on a device by setting **allowOTAUpgrade()** to be false via the MDM. This can be used to either freeze the software installed or to allow an organization time to test the update before letting it roll out to the user community.

5 Operational security

5.1 Modes of operation

The TOE can be operated in four different modes, depending on the role of the user accessing the device:

- Administrator mode;
- User mode;
- Error mode; and
- Recovery mode

A device is considered to be in **Administrator mode** before it is delivered to the user. The device is prepared and configured for deployment in the enterprise environment via the Samsung Enterprise SDK. The TOE administrators are trusted to follow and apply all administrator guidance in a trusted manner. An unprivileged user will not have access to this mode of operation.

If an error or operational failure occurs during the transition from Administrator mode (causing the device to momentarily enter the Error mode of operation) to User mode, the administrator should follow the guidance for the Mobile Device Management Solution to rectify the failure and restore the device to normal operational abilities. If it is not possible to adequately eliminate the error or operational failure, the device is not to be delivered to an end user and should be returned to the supplier.

After the device is configured in accordance with the Common Criteria evaluated settings, the device is ready for deployment to a user. When the user receives the device, only the TouchWiz user interface will be visible and no further changes to the security configuration are possible. Once deployed to a user, the device will be operating in **User Mode**. Within User Mode, the only security relevant functions accessible for the user are 'lock screen password protection', 'change of password' and 'local device wipe'. Typically, an administrator will not access the device in this mode of operation.

The TOE may also be placed into Recovery mode, bypassing the standard boot process and allowing for configuration changes to be made to the installation of Android. However, this requires the boot loader for the device to be unlocked and is therefore considered out of scope for this environment.

5.2 Wiping data

The evaluated security configurations provide the ability to both locally or remotely wipe data on the device. Based on the device configuration, it is possible to wipe data at the device level, the KNOX container level or both.

An enterprise initiated remote wipe command (either for the device or just the KNOX container, depending on the configuration) occurs under the following conditions:

- The enterprise sends a remote wipe command to the device:
 - when the device has been lost or stolen;
 - in response to a reported incident;
 - in an effort to resolve current mobile issues; and
 - for other procedural reasons such as when an Android device end user leaves the organization.

5.2.1 Wiping the device

The evaluated security configuration provides for a local and a remote wiping process of Android user devices. This type of wipe works at the storage level and will wipe all data on the device. In a KNOX container configuration this will wipe all data including the KNOX container (as well as everything not in the container). This type of wipe is available in all configurations.

The local wipe is manually initiated by the Android device user or after an exceeded number of incorrect login attempts. The remote wipe process is in general remotely initiated by the Enterprise and Mobile Device Administrator via a remote wipe command.

5.2.2 Wiping the KNOX Container

When a KNOX container has been enabled it is also possible to wipe only the data stored in the KNOX container. A wipe of the container data will remove the container, including apps and data, but it will not remove anything outside the KNOX container. This process must be initiated remotely by the Enterprise and Mobile Device Administrator via a remote wipe container command.

The only way for a user to locally wipe the KNOX container is to unenroll the device from the control of the MDM. When this is done the KNOX container, all data and apps as well as the MDM Agent will all be removed from the device.

5.3 Additional notes on operational security

Common Criteria Part 3 does require operational user guidance for the following:

- User-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.
- Secure usage of available interfaces.
- Security parameters of interfaces and functions under the control of the user and their secure values.
- Each type of security-relevant event relative to the user-accessible functions.

Administrators and users are considered to use a Samsung Enterprise device. As described in previous sections of this document, the administrator is responsible for configuration and installation of the device. The end user receives the device in an operational state where no further security configuration is possible. The only user accessible user functions are 'lock screen password protection', 'change of password' and 'local device wipe'.

The user is responsible to obey the provided user guidance and to not actively working against the protection of the device data.

The TOE Administrators are trusted to follow and apply all administrator guidance, resp. [MDMG] in a trusted manner. [MDMG] provides further operational user guidance.