Samsung VPN Client on Galaxy Devices

Guidance documentation

Version 1.2

April 15, 2015

Document management

Document identification

Document ID	Samsung Guidance documentation 1.2
Document title	Samsung VPN Client on Galaxy Devices Guidance documentation
Release authority	

Document history

Version	Date	Description	Author
0.1	April 3, 2014	Initial draft	Brian Wood
0.2	April 4, 2014	Updated based on info about server cert field	Brian Wood
0.3	April 7, 2014	Updated supported devices list	Brian Wood
0.4	April 14, 2014	Updated to add use information	Brian Wood
0.5	April 21, 2014	Updated based on evaluation feedback	Brian Wood
0.6	May 13, 2014	Updated device list	Brian Wood
0.7	June 6, 2014	Modified device list table	Brian Wood
0.8	September 15, 2014	Updated device list	Brian Wood
0.9	September 19, 2014	Updated device list	Brian Wood
1.0	October 7, 2014	Edited versions and CC Mode access	Brian Wood
1.1	October 28, 2014	Updated device list	Brian Wood
1.2	April 15, 2015	Updated download site. Updated for OS update & devices	Brian Wood

Table of Contents

List of Figures

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 separately evaluated on Android 5 and so is removed from 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 VPN running 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 mobility solution providing mobile staff with enterprise connectivity. With a focus on enterprise security, the TOE supports IKEv2 VPN tunnels using both Pre-shared Keys as well as certificates, providing flexibility based on the environment.

The TOE combines with a Mobile Device Management (MDM) solution that enables the enterprise to manage VPN tunnels for mobile devices to facilitate secure communications back to the enterprise network. 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 whitelisting and 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
Secure Channel. Enterprise devices can securely connect to the enterprise network.	**VPN.** The TOE provides a secure communications channel to the VPN Gateway.
Enterprise device management. Enterprise administrators can control mobile endpoint configurations.	**Security policy.** The TOE can be configured by a Mobile Device Management solution that supports the Samsung Enterprise SDK.

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 (such as the TOE) 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.

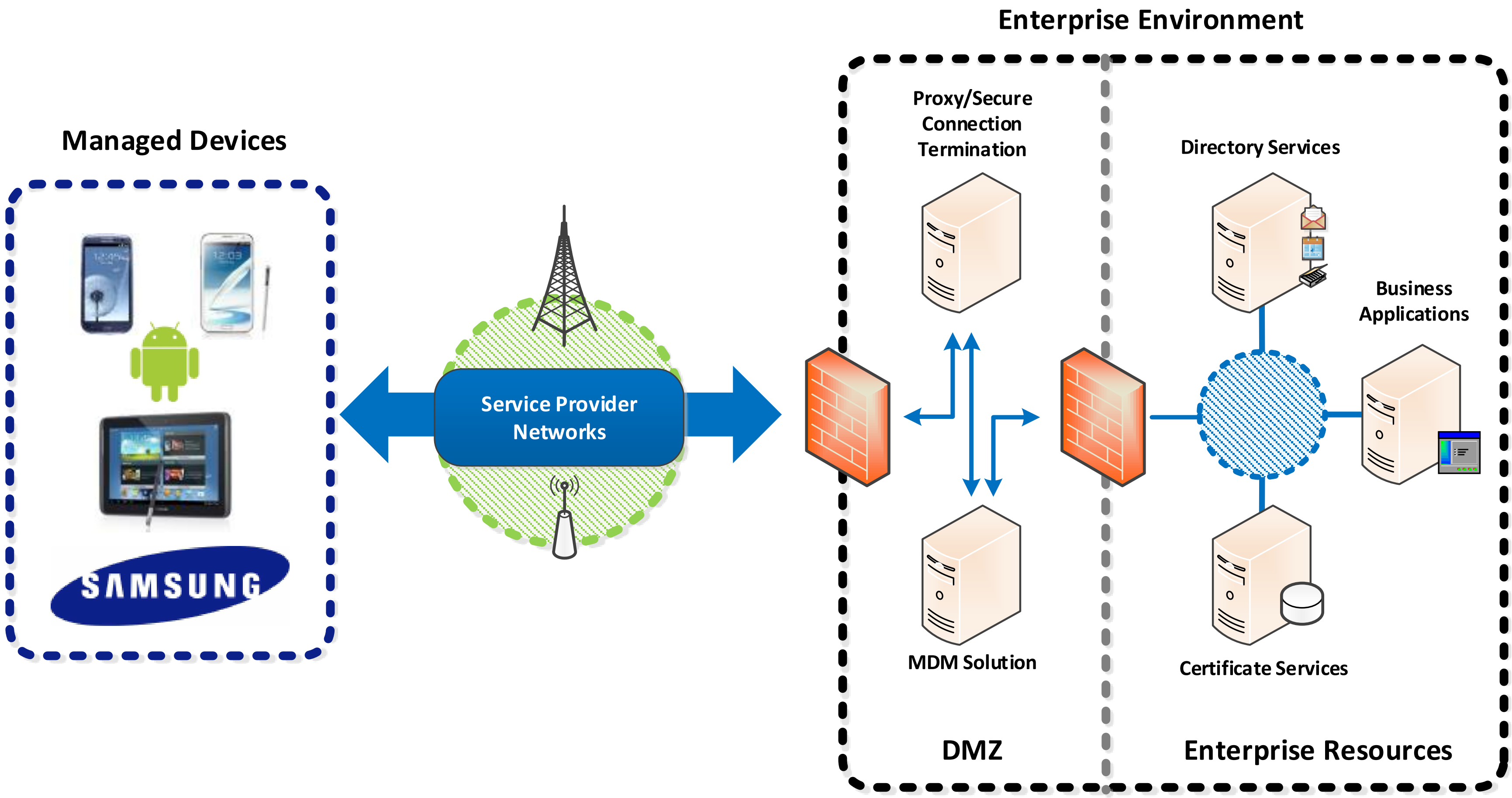


Figure 1 – Enterprise Environment

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 4.4.

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 4.4) 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 4.4).

6. Configure the VPN client (the TOE) 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.

4.4 Configuration of the VPN Client

Once the device has been deployed and connected to the MDM, the VPN client must be configured. In a managed environment this would be through the MDM, and the following sections specify the options the MDM must support to configure the VPN correctly.

As with most VPN clients, the client options are kept very simple and the primary configurations (such as key timeouts and cryptographic algorithms) are specified on the VPN gateway.

4.4.1 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.

4.4.1.1 CC Mode Settings

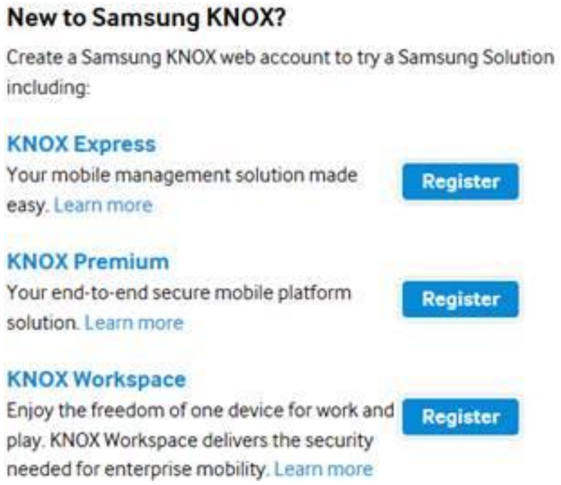
To place a device into the evaluated configuration the CC Mode must be enabled.

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.



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

KNOX Premium

Your end-to-end secure mobile platform solution. Learn more

KNOX Workspace

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

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.

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.

4.4.1.1.1 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.

4.4.1.1.2 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.

For the VPN configuration, CC Mode only needs to be Enforced as that will set the required FIPS encryption. Having CC Mode fully enabled is recommended (and required for the Mobile Device Fundamentals configuration), but those extra settings are not required for the VPN configuration.

4.4.1.2 VPN Settings

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 or Value

Profile Management	Profile name	Create, rename and delete VPN profiles	createProfile()
			setProfileName()
			deleteProfile()

VPN Type Setting		The types of VPN connections which can be set. Those listed here are the only validated types.	IPSEC_IKEV2_PSK IPSEC_IKEV2_RSA (see below)

VPN Settings (PSK)		The settings needed to configure the VPN tunnel when using a Pre-Shared Key.	setServerName()
			setIpSecIdentifier()
			setIPSecPreSharedKey()

VPN Settings (certificate)		The settings needed to configure the VPN tunnel when using a certificate.	setServerName()
			setIPSecCaCertificate()
			setIPSecUserCertificate()
			setOcspServerUrl()

Setting	Value	Description	Class or Method or Value

VPN Optional Network Settings		These settings provide additional network configuration and routing options for the tunnel.	setDnsDomains()
			setDnsServers()
			setForwardRoutes()

Always-on VPN	Enable/ Disable	Specifies whether all traffic must go through the specified VPN tunnel. If no connection can be made no traffic will flow.	setAlwaysOnProfile()

User Control	Enable/ Disable	Whether the user is allowed to create new VPN profiles, change profiles or modify the Always-on VPN setting.	allowUserAddProfiles()
			allowUserChangeProfiles()
			allowUserSetAlwaysOn()

4.4.1.2.1 Valid Certificate Types

While the menu selection for the type of tunnel states IPSec IKEv2 RSA it is possible to utilize both RSA and ECC certificates for the tunnel. As long as the certificates are valid (not expired, properly formatted, etc) they can be used for the VPN configuration.

4.4.1.2.2 Specifying a Strong Pre-Shared Key

A PSK (Pre-shared key) is like a password, a fixed string used to authenticate the VPN client to the VPN gateway. Since the PSK does not change (or at least does not change often), a strong string should be selected to protect against unauthorized access to the VPN by unknown clients.

The PSK can be entered in two forms: ASCII or HEX. All ASCII characters are supported. HEX keys must start with “0x” as the first two characters entered. If those are the first two characters, the remaining entry will be read as a HEX key. The maximum key size is 64 characters entered.

The PSK should be chosen according to your organization’s security policy regarding VPN configurations. Based on this policy, and acceptably complex PSK will consist of the following:

- The PSK should be at least 20 characters long
- Minimum letters required in password (a-z, A-Z);
- Minimum lowercase letters required in password (a-z);
- Minimum non-letter characters required in password (0-9 and special characters +=%_@#$/^&*()‘-”:!;?,`.~\|<>{}[]);
- Minimum numerical digits required in password (0-9);
- Minimum symbols required in password (+=%_@#$/^&*()‘-”:!;?,`.~\|<>{}[]); and
- Minimum uppercase letters required in password (A-Z).

When setting the PSK, you should be careful **not** to:

- Use known information about yourself or the company (e.g. company name, address, your name or any information recoverable from the public domain);
- Set a password which is similar to previous passwords (adding a ‘1’ or “!” to the end of the password is not sufficient); or
- Use simple dictionary words (Welcome1!).

4.4.1.2.3 Server Certificate for the Gateway

It is possible to specify a Server Certificate for the Gateway in the configuration of a VPN tunnel. This certificate will override any certificate provided by the Gateway during the negotiation of the tunnel.

This certificate can only be loaded through the UI and does not have an API. See the User Guidance for more information about loading this certificate.

4.4.1.3 Certificate/Key Management Settings

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)

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

4.4.2 VPN Gateway Configuration Control

As noted above, there are many configuration options for a VPN tunnel which cannot be configured from the client and must be configured from the gateway. The VPN client will utilize these settings from the gateway configuration to construct the secure tunnel. The following is a list of the settings that must be configured through the gateway:

- Encryption settings – the VPN client will use FIPS validated encryption, the gateway will specify which algorithms should be used.
- IKE Protocols & Authentication – the gateway specifies which IKE protocols are allowed and which authentication techniques are required for establishing the connection.
- IPsec Session Key cryptoperiod – the gateway specifies the session key cryptoperiod and can be used to configure periods under 1 hour in duration.

4.5 Using the VPN

4.5.1 Always-on Tunnel

When the device has a tunnel configured for Always-on VPN, all traffic will automatically go through this tunnel, and if for some reason a connection for the tunnel cannot be made, no traffic will be allowed to communicate off the device.

4.5.2 “Normal” VPN Tunnels

When VPN tunnels are configured and no tunnel is specified as Always-on, then the user must select the tunnel to be used. The user will select the tunnel from those available at ***Settings|More networks|VPN***.

4.6 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).

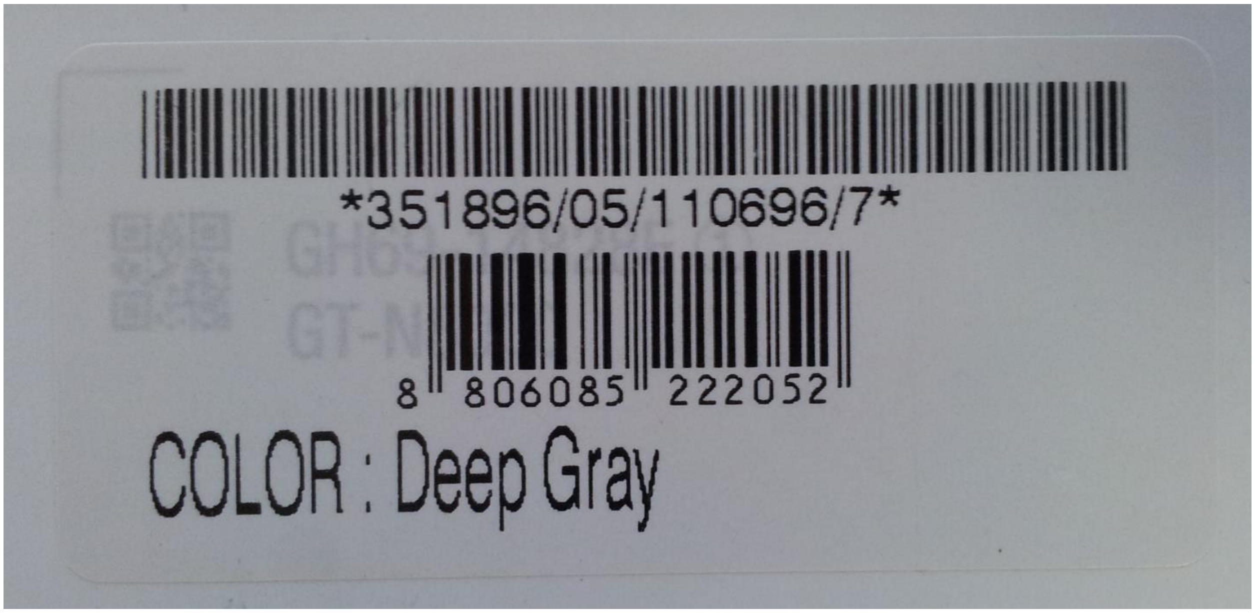


Figure 2 - Tracking label

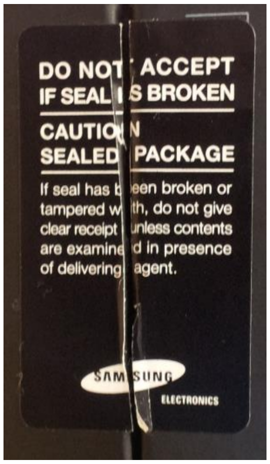

DO NOT ACCEPT
IF SEAL IS BROKEN
CAUTION
SEALED PACKAGE
If seal has been broken or tampered with, do not give clear receipt unless contents are examined in presence of delivering agent.
SAMSUNG
ELECTRONICS

Figure 3 - Security Seal (Black)

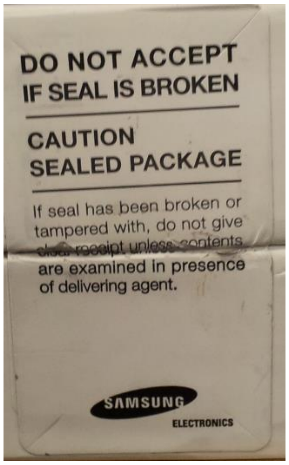

DO NOT ACCEPT
IF SEAL IS BROKEN
CAUTION
SEALED PACKAGE
If seal has been broken or tampered with, do not give clear receipt unless contents are examined in presence of delivering agent.
SAMSUNG
ELECTRONICS

Figure 4 - Security Seal (White)

4.6.1 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 VPN, this will show:

VPN v1.4 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.

4.7 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.

4.7.1 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.

4.7.2 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 two 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.