Samsung Enterprise Mobility Solutions

Samsung KNOX™ Technical Note:

IKnoxVpnService AIDL File Description

October 11th 2013

Copyright Notice

Document Information

This document was created on August 7, 2013 by the San Jose B2B Team
This document was last modified on September 10, 2013.

Contact Information

Samsung Enterprise Mobility Solutions – Santa Clara
Samsung Telecommunications America, Ltd
3920 Freedom Circle, Ste 101
Santa Clara, CA 95054
United States of America

Contents

About Samsung KNOX™ Technical Notes

This technical note contains important company confidential information regarding Samsung KNOX™.

Purpose

The purpose of this document is to describe the Samsung KNOX IKnoxVpnService AIDL file description.

Audience

The audience for this guide comprises VPN partners. The content presentation is based on the assumption that you are knowledgeable in the Android platform environment.

About this document

This document describes the Samsung KNOX IKnoxVpnService AIDL file description. Use the following links to jump to a specific location of your interest in this document:

Notation Conventions

This manual uses the following notation conventions.

- **Boldface** emphasizes words in text such as screen or window names or commands that you enter.
- *Italics* identifies new words or emphasizes phrases.
- `Monospace` represents information as it appears on a display or in command syntax.

1 IKnoxVpnService AIDL File Description

This section describes the APIs which make up the Samsung KNOX IKnoxVpnService AIDL file description.

About IKnoxVpnService AIDL File Description

All the APIs are synchronous and should return immediately. If certain APIs involve time-consuming network operations such as start, stop, and remove connection, the API can return once the time-consuming network operation is initiated.

VPN vendors should have the following intent action declared in the manifest file for the framework to bind. In addition, make sure that the call is coming from the framework (that is, the calling uid has the value Process.SYSTEM_UID).

```
<intent-filter>
        <action android:name="${PACKAGENAME}.BIND_SERVICE " />
</intent-filter>
The intent action is package name appended by ".BIND_SERVICE".
```

The SRG requirement details are available in SRG Implementation Requirements for the following API:

```
setServerCertValidationUserAcceptanceCriteria(String profileName, boolean
enableValidation, in List<String> condition, int frequency);
```

IKnoxVpnService APIs

This section describes the Samsung KNOX IKnoxVpnService APIs.

1. int createConnection(in String jsonProfile);

Description

This API is used to create a VPN connection.

Parameters

- jsonProfile: JSON File which contains the profileInfo in String format.

Returns

- 0—If the profile has been created successfully
- 1—If the profile was not created successfully

2. int removeConnection(String profileName);

Description

API used to delete the VPN profile.

Note: Refer to About IKnoxVpnService AIDL File Description.

Parameters

- profileName: The name of the VPN Connection which needs to be removed

Returns

- 0—If the profile has been removed successfully
- 1—If removeConnection API is called, before creating a Connection (that is, removing a non-existing profile)
- -1—If some error occurred while removing the profile

3. String getConnection(String profileName);

Description

Retrieves the VPN Connection details belonging to a particular profile.

Parameters

- profileName: Name of the connection to be retrieved

Returns

JSON object in String format which contains the connection information.

4. List<String> getAllConnections();

Description:

Retrieve all the vpn connection object belonging to a particular profile.

Returns:

list of JSON object in String format which contains the connection Information.

5. boolean setUserCertificate(String profileName, in byte[ ] pkcs12Blob, String password);

Description

This API allows app to configure the User certificate for a VPN profile.

Parameters

- profileName: Name of the profile
- pkcs12Blob: Byte array of User certificate in pkcs12 format
- Password: Password to decrypt the content of pkcs12 blob

Returns

True, if user certificate is configured with given profile successfully. If API fails to read PKCS12 blob or fails to store the user certificate for the specified profile, it returns false.

6. boolean setCACertificate(String profileName, in byte[ ] blob);

Description

This API allows app to configure the CA certificate for a VPN profile.

Parameters

- profileName: Name of the profile
- Blob: Byte array of User certificate in DER/PEM format

Returns

True, if CA certificate is configured with given profile successfully.

If API fails to read blob or fails to store the CA certificate for the specified profile, it returns false.

7. CertificateInfo getUserCertificate(String profileName);

Description

The API returns a User Certificate for the specified profile.

Parameters

- profileName: Name of the profile

Returns

CertificateInfo object.

Return value is null, if profile with name profileName is not found, or User certificate is not found for the specified profile.

8. CertificateInfo getCACertificate(String profileName);

Description

The API returns a CA Certificate for the specified profile.

Parameters

- profileName: Name of the profile

Returns

CertificateInfo object. Return value is null, if profile with name profileName is not found, or CA certificate is not found for the specified profile.

9. int startConnection(String profileName);

Description

The API is used to start a VPN connection.

Note: Refer to About IKnoxVpnService AIDL File Description.

Parameters

- profileName; Name of the profile for which the VPN connection has to be started

Returns

- 0—If the profile has been started successfully
- 1—If startConnection API is called, before creating a Connection (that is, starting a non-existing profile)
- -1—If some error occurred while starting the profile

10. int stopConnection (String profileName);

Description

The API is used to stop a VPN connection.

Note: Refer to About IKnoxVpnService AIDL File Description.

Parameters

- profileName: Name of the profile for which the VPN connection has to be stopped

Returns

- 0—If the profile has been stopped successfully
- 1—If stopConnection API is called, before creating a Connection (that is, stopping a non-existing profile)
- -1—If some error occurred while stopping the profile

11. int getState(String profileName);

Description

The API used to list the available states for the profile.

Parameters

- profileName: Name of the profile

Returns

- IDLE = 1;
- CONNECTING = 2;
- CONNECTED = 4;
- DISCONNECTING = 3;
- FAILED = 5;
- DELETED = 6;

12. String getErrorString(String profileName);

Description

This API gets the error description or error code for the connection profile.

Parameters

- profileName: Name of the profile

Returns

The current error code associated with the given profile.

13. int getVpnModeOfOperation(String profileName);

Description

The API specifies whether the VPN service is operating in a FIPS or a non-FIPS mode.

Parameters

- profileName VPN profile name of which FIPS mode has to be queried.

NB: If profile based FIPS is not supported, retrieve vendor VPN solution's current mode of operation.

Returns

- 1—FIPS. The VPN service is operating in FIPS mode
- 0—Non-FIPS mode. The VPN service is not operating in FIPS mode
- -1—Error

14. int setVpnModeOfOperation (String profileName, int vpnMode)

Description

API to set VPN mode of operation in either FIPS or non-FIPS mode.

Parameters

- profileName VPN profile to which FIPS mode has to be applied.
- vpnMode: 1 to set device in FIPS mode and 0 to set device in non-FIPS mode.

NB: If profile based FIPS is not supported, support FIPS for the all VPN connections. In that case, profileName parameter can be ignored.

Returns

- 0: If the requested Mode of operation has been successfully set.
- 1: If the requested Mode of operation is not successfully executed.
- -1: Error occurred when performing the above operation.

15. boolean setServerCertValidationUserAcceptanceCriteria(String profileName, boolean enableValidation, in List<String> condition, int frequency);

Description

API to enable list of SRG requirements for a given profile.

Note: Refer to About IKnoxVpnService AIDL File Description.

Parameters

- profileName: The profile name for which the SRG requirements has to be set or not
- enableValidation: To enable SRG requirements for the given profile or not
- Condition: The list of rules added. The details will be updated later
- Frequency: The frequency by which to notify the user

Returns

true: If the requested Mode of operation has been successfully set.
false: If the requested Mode of operation is not successfully executed.

16. boolean setAutoRetryOnConnectionError(String profileName, boolean enable);

Description

API to set whether to enable auto-reconnect feature or not for the given profile.

Parameters

- profileName: The profile name for which the auto-reconnect feature has to be set or not
- enable: True to enable auto-reconnect feature, else otherwise

Returns

true: If the requested Mode of operation has been successfully set.
false: If the requested Mode of operation is not successfully executed.

2 SRG Implementation Requirements

This section describes the SRG implementation requirements necessary to comply with VPN SRG rules.

Modifications to comply VPN SRG Rules

The following modifications are required to comply with VPN SRG rules:

1. When any of the following violations found during IKE, a notification will be displayed in the status bar (triangle notification, as shown in Figure 1) and the VPN connection will be suspended.

    a) Subject name mismatch violation
    b) Key usage violation
    c) Revocation verification failed

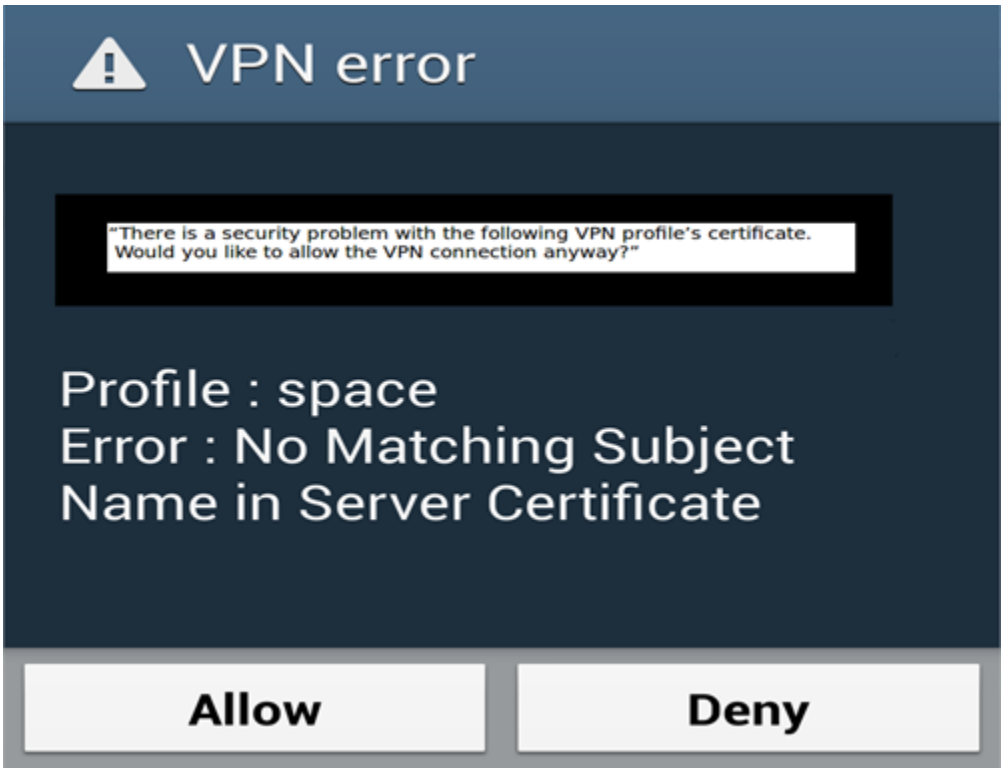

Figure 1. VPN error

2. When user drags status notification and taps certificate validation violation notification, a message box is displayed with the reason of the failure and the user is prompted with 'Allow' or 'Deny' VPN connection options.

3. Itemize all violation reasons (for example, a-c, as listed in modification #1 above) in a single message box if a certificate has more than one violation reason.

4. If user selects 'Allow' option, this time VPN service will skip validating the last violations found.

5. If user selects 'Deny' option (in step 2), then VPN connection fails and traffic from any packages added to VPN profile is blocked.

6. The user selection (Allow/Deny) is applicable to only VPN connection in which violation is identified. If the violation is identified in a different VPN connection, the user is notified separately (that is, the user is granted permission for individual VPN connection in case of violations found in multiple VPN connections).

7. Failure message can contain only the violation reason.

8. To handle certificate validation failure prompts, a new MDM API can be provided which will enable/disable certificate validation based on the condition set.

MDM API

Following API enables/disables certificate validation for the conditions set.

```
public boolean setServerCertValidationUserAcceptanceCriteria(String profile, boolean
enableValidation, List <int> condition, int frequency);
```

Parameters Description

Parameter 'profile' is used to pass name of profile of which server certificate validation condition is going to set for IKE negotiation.

Parameter 'enableValidation' is used to enable / disable certificate validation for given condition.

```
Private static boolean ENABLE_SERVER_CERT_VALIDATION true
Private static boolean DISABLE_SERVER_CERT_VALIDATION false
```

Parameter 'condition' can be one (or a list) of following:

```
Private static int VPN_ERROR_SERVER_CERT_SUBJECT_NAME_MISMATCH  0
Private static int VPN_ERROR_SERVER_CERT_KEY_USAGE_VIOLATION 1
Private static int VPN_ERROR_SERVER_CERT_REVOCATION_VERIFICATOIN_FAILED 2
```

Parameter 'frequency' can be one of the following values:

```
private static int VPN_SERVER_CERT_VALIDATION_IGNORE_ALWAYS 0
private static int VPN_SERVER_CERT_VALIDATION_IGNORE_CURRENT_SESSION 1
private static int VPN_SERVER_CERT_VALIDATION_IGNORE_ONCE 2
```

Framework will default to ' VPN_SERVER_CERT_VALIDATION_IGNORE_ALWAYS' when user selects 'Allow'.

MDM can enforce more stringent certificate validation condition based on the described API.

About Samsung Electronics Co., Ltd.

Samsung Electronics Co., Ltd. is a global leader in technology, opening new possibilities for people everywhere. Through relentless innovation and discovery, we are transforming the worlds of televisions, smartphones, personal computers, printers, cameras, home appliances, LTE systems, medical devices, semiconductors and LED solutions. We employ 236,000 people across 79 countries with annual sales exceeding KRW 201 trillion. To discover more, please visit www.samsung.com

For more information about Samsung KNOX, visit www.samsung.com/knox

Samsung Electronics Co., Ltd.
416, Maetan 3-dong, Yeongtong-gu
Suwon-si, Gyeonggi-do 443-772, Korea