CDM for Android - Vendor Extensions

version 1.6

Introduction

This document lists the vendor unique extensions to the Android MediaDrm API as implemented by the Widevine Modular DRM plugin.

UUID - isCryptoSchemeSupported(UUID uuid)

    {0xED,0xEF,0x8B,0xA9,0x79,0xD6,0x4A,0xCE,0xA3,0xC8,0x27,0xDC,0xD5,0x1D,0x21,0xED}

For historical/backward compatibility reasons, the plugin also supports the following UUID:

    {0x29,0x70,0x1F,0xE4,0x3C,0xC7,0x4A,0x34, 0x8C,0x5B,0xAE,0x90,0xC7,0x43,0x9A,0x47}

QueryKeyStatus

HashMap<String, String> queryKeyStatus(byte[] sessionId)

Request an informative description of the key status for the session. The status is in the form of {name, value} pairs. The specific status field key names and example values defined by Widevine are listed in the table below. They have been supported since android version 4.3 (API 18)

Key String	Example Values	Description
LicenseType	"Streaming", "Offline".	Streaming licenses only permit access to content from a server while online. Offline licenses are stored on the device and may be used to view content while not connected to a network.
PersistAllowed	"True", "False".	Indicates whether the license may be saved, typically True for Offline license types.
RenewalServerUrl	"https://<server-url>"	The server URL to use when renewing the license
LicenseDurationRemaining	"300"	Integer representing remaining license duration in seconds
RenewAllowed	"True", "False"	Boolean indicating whether the license is permitted to be renewed
PlaybackDurationRemaining	"3600"	Integer representing the remaining playback duration in seconds
PlayAllowed	"True", "False"	Boolean indicating whether playback is allowed

String Properties

String getPropertyString(String propertyName)
void setPropertyString(String propertyName, String value)

Read a DRM engine plugin String property value, given the property name string. Standard fields names are: PROPERTY_VENDOR, PROPERTY_VERSION, PROPERTY_DESCRIPTION, PROPERTY_ALGORITHMS. The standard and vendor-specific string property values defined for Widevine are:

Key String	Operations	Description	Added in Android version
PROPERTY_VENDOR	get	value="Google"	4.3
PROPERTY_VERSION	get	value="1.0"	4.3
PROPERTY_DESCRIPTION	get	value="Widevine CDM"	4.3
PROPERTY_ALGORITHMS	get	A comma-separated list of cipher and mac algorithms supported by CryptoSession. e.g. "AES/CBC/NoPadding,HmacSHA256"	4.3
"securityLevel"	get, set	The Widevine-defined security level. May be one of "L1", "L2,or "L3".	4.3
"systemId"	get	Widevine System ID value as string, e.g. "1234"	4.3
"privacyMode"	get, set	When privacyMode is enabled, the first call to getKeyRequest will return a service certificate request, which will be used to obscure the device identifying information in the subsequent license request. Values are "enable" or "disable".	4.4
"sessionSharing"	get, set	When sessionSharing is enabled, the keys for any sessions opened from a given MediaDrm object will be accessed by key_id from any other sessions opened from that MediaDrm object. Values are "enable" or "disable".	4.4

"usageReportingSupport"	get	Indicates whether the device supports usage reporting. Values are "True", "False".	5.0
"appId"	get, set	"com.example.appname" app ID establishes isolation for secure stop storage, to avoid conflicts between apps.	6.0
"origin"	get, set	"google.com" Used to keep provisioning storage separate for different origins, as outlined in the W3C Encrypted Media Extensions specification. If unset, the device-wide provisioning is used.	6.0
"hdcpLevel"	get	The current HDCP connection status. If multiple HDCP-protected ports are present, indicates the minimum level. "Disconnected" - No device is connected "Unprotected" - HDCP negotiation failed "HDCP-1.x" - HDCP level 1.x is active "HDCP-2.0" - HDCP level 2.0 is active "HDCP-2.1" - HDCP level 2.1 is active "HDCP-2.2" - HDCP level 2.2 is active	5.0
"maxHdcpLevel"	get	The maximum HDCP version that can be supported by the device. If multiple HDCP-capable ports are present, indicates the lowest maximum level for all ports. "Disconnected" - No device is connected "Unprotected" - HDCP negotiation failed "HDCP-1.x" - HDCP level 1.x is active "HDCP-2.0" - HDCP level 2.0 is active "HDCP-2.1" - HDCP level 2.1 is active "HDCP-2.2" - HDCP level 2.2 is active	5.0
"maxNumberOfSessions"	get	The maximum number of sessions supported by the DRM plugin	6.0
"numberOfOpenSessions"	get	The number of currently open sessions	6.0
"oemCryptoApiVersion"	get	The version of OEMCrypto for the current security level, e.g. "11". See "securityLevel" above to query/set the	7.0

		security level	
"CurrentSRMVersion"	get	The current HDCP System Renewability Message version e.g. "1"	8.0
"SRMUpdateSupport"	get	Indicates whether the device supports HDCP System Renewability Message updates. Values are "True", "False".	8.0

ByteArray Properties

setPropertyByteArray(String propertyName, byte[] value)
byte[] getPropertyByteArray(String propertyName)

Read a DRM engine plugin byte array property value, given the property name string. Standard fields names are: PROPERTY_DEVICE_UNIQUE_ID. The standard and vendor-specific byte array property values defined for Widevine are:

Key String	Operations	Added in android version	Description
PROPERTY_DEVICE_UNIQUE_ID	get	4.3	A device unique ID specified by the OEM. For devices originally shipping with android O or later, this ID is unique per device, app and origin (domain). It persists across factory resets.
"provisioningUniqueId"	get	4.3	A randomly generated UUID assigned to the device during provisioning. For devices originally shipping with android O or later, this UUID is unique per device, app and origin (domain). It persists across factory resets.
"serviceCertificate"	get, set	4.4	Allows app to get/set the privacy mode service certificate use in license requests.
"provisioningServiceCertificate"	get, set	9.0	Allows app to get/set the privacy mode service certificate use in provisioning requests.

Unprovisioning

provideProvisionResponse(byte[] response)

It is not possible to remove the device-wide provisioning from the device. However, it is possible to revoke the provisioning of a specific origin. First, set the "origin" property as outlined above, then pass the special value "unprovision" to provideProvisionResponse() instead of a provisioning response.

```
byte[] response = "unprovision".getBytes();
mediaDrm.provideProvisionResponse(response);
```

This will unprovision the provided origin. If the origin is not set, this will fail.