LAS16-302: LHG Reference Security Solutions

Zoltan Kuscsik, PhD
Secure video playback - what can your device play?

Widevine DRM (Android)

- Level 3 security - no real protection of video buffers.
- Level 1 security - buffers secure all the way to the display.

Applications can test the security level using the API call

DrmManagerClient.acquireDrmInfo()
Secure video playback - what can your device play?

Playready (Linux/Android)

- SL2000 - HD resolution, buffers exposed to non-secure world.
- SL3000 - UHD resolution, buffers fully protected.

Playready is integrated using OCDM (Open Content Decryption Module) on Linux/Chromium and using a MediaDRM plugin developed by LHG.
## LHG reference implementations on OP TEE

<table>
<thead>
<tr>
<th></th>
<th>Linaro Clear Key OCDM with OP TEE</th>
<th>Linaro OCDM with TEE</th>
<th>Linaro OCDM with software Playready</th>
<th>Linaro OCDM with HW Playready</th>
<th>Android Playready</th>
<th>Android Widevine</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPAPI CDM</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>OpenCDM</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>NA</td>
<td>A</td>
</tr>
<tr>
<td>OP TEE and TrustZone®</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Compatibility</td>
<td>ARMv7, ARMv8</td>
<td>ARMv7, ARMv8</td>
<td>ARMv7, ARMV8</td>
<td>ARMv7, ARMV8</td>
<td>32bit TEE only</td>
<td>ARMv7, ARMV8</td>
</tr>
<tr>
<td>HiKey</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes (need device keys)</td>
</tr>
<tr>
<td>Dragonboard</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
DRM - what parts are open source? Can we do more?
Android framework integration

Application Layer

ExoPlayer

MediaDRM (UUID, Key System query)

Android Framework

CDMs

Widevine CDM

W3C EME Clear Key

REE

OPTEE - Widevine Integration

TEE

PlayReady CDM

PlayReady

Lib PRiTEE

PlayReady TA
Writing or testing an Android DRM plugin?

Step 1.) Use ExoPlayer for testing

https://github.com/google/ExoPlayer

It supports HLS, DASH, SmoothStreaming and it comes with valid samples for various DRM formats.
Writing or testing an Android DRM plugin?

Step 2.) **Create MediaDRM native plugin.** It is good start is to base your plugin structure on the ClearKey implementation inside AOSP:

```
frameworks/av/drm/mediadrmm/plugins/clearkey/
```

ClearKey is useful to stress test the TEE, since it is easy to integrate with any TEE. You just need to implement a AES 128 call.

MediaDRM plugins are shared libraries dlopen-ed by Android framework on demand. Different DRMs are identified by self assigned UUIDs:

http://dashif.org/identifiers/protection/
Writing or testing an Android DRM plugin?

Step 3.) Integrate DRM with TEE. In our case OP TEE
OP TEE development on AOSP

- OP TEE Kernel patches are now upstream in Hikey Kernel.
- Needs to be upstreamed: OP TEE build scripts, ATF and UEFI patches.
- Built using SWG manifest:
  https://github.com/linaro-swg/optee_android_manifest

Outstanding issues:
- Coexistence of OPTEE with other TEEs like Trusty.
Outstanding issues of our reference implementations

- Secure Buffer Allocation - needs more work to support OP TEE.
- Secure Data Path - interfaces to decoder/GPU. Using secure EGL extensions?
- HDCP support - we don’t have access to a reference implementation.
Thank You

#LAS16
For further information: www.linaro.org
LAS16 keynotes and videos on: connect.linaro.org