NATIVE PLAYERS ARE GOOD ENOUGH: FACTS AND FALLACIES

White paper

inside secure
DRIVING TRUST

TABLE OF CONTENTS

INTRODUCTION

Content providers who stream VoD (Video on Demand) content to mobile devices sometimes choose a do-it-yourself solution, which almost invariably on Android means a combination of the open-source ExoPlayer and the built-in Widevine DRM (available on all modern Android devices) or sometimes the built-in PlayReady or Verimatrix DRM (available on a small subset of Android devices).

This whitepaper aims to shed some light on some of the tradeoffs between choosing to adopt ExoPlayer as-is and using a third-party solution that includes ExoPlayer within it.

MYTHS AND FACTS

Myth #1:

You can have secure content playback with just off-the-shelf open-source components

The thinking goes, ExoPlayer is open-source and is maintained by Google, which is a guarantee of quality. The baked-in DRM implementation is sufficient because it was implemented by the device's chipset vendor and the OEM, and it sometimes takes advantage of a Trusted Execution Environment (TEE).

Facts

Hardware-based DRM implementations are often hasty and partial, because SoC vendors need to provide a "minimum viable" implementation in order to keep their costs as low as possible. For example, at some point in time, some leading device vendors' TEE-based DRM implementations were lacking implementation of several compulsory APIs, and therefore workarounds were needed to exploit the native DRM stack properly.

Additionally, in order for a DRM implementation to be truly robust, it needs **to fulfil Compliance and Robustness rules** (related to, for example, enforcing HDCP 1.x or 2.x, or to having a tamper-free clock), which is typically a multi-man-year effort that OEMs or SoC vendors don't always have the bandwidth or resources for. Inside Secure, on the other hand, has spent and is spending considerable effort in ensuring adherence to the C&RR, on an ongoing basis[1].

An additional issue is renewability. Should a hardware-based DRM implementation become compromised, then the only possible solution is for the OEM to ship a firmware update. However, the extremely slow pace of firmware updates for general security vulnerabilities is very well-known in the industry[2]. For example, as of this writing, nearly 80% of devices run Android versions that are over two years old or older. These delays are further exacerbated in those situations where a device is further customized by a mobile network operator. On the other hand, all it takes to renew a software DRM implementation is an app update[3].

[1] It is worth mentioning that the Inside Secure ExoPlayer implementation can, optionally, piggyback on the native hardware DRM implementation if present ("pass-thru mode")
[2] There is a lot of evidence of the glacially slow pace of Android updates to devices. For an example, see this story.
[3] To date, there are no reported vulnerabilities on Inside Secure Content Protection Client.

Myth #2:

If I license a third-party implementation of ExoPlayer, it might be customized in a way I don't want

Service providers sometimes have concerns over an ExoPlayer variant that has been customized by a third party. The concerns are that this variant will have many unnecessary customizations that will affect performance, the quality of video playback, the user experience, or other desirable aspects of content playback.

Facts

The ExoPlayer customizations done by Inside Secure are only in the area of security and robustness, with the addition of some desirable features, such as Download & Play. Customers who are used to dealing with the original GitHub version of ExoPlayer will benefit from the same performance and from the same feature set (plus a few value-added ones) as they are accustomed to.

Myth #3:

It is easy to keep up with ExoPlayer releases and with Android OS versions

The team developing ExoPlayer releases very often. They often provide a host of new features with each release. If I am a content distributor who has an app to ship, I am only too happy to pass on these new features "for free" to my users, with little impact on my R&D team's development cycle.

Additionally, Google releases new versions of Android only a few times per year. Most of the changes will not affect content playback, thus I can rest assured that I will only need a limited amount of testing every time a new Android OS version is released.

Facts

It is true that there are very frequent ExoPlayer released on GitHub – for example, in 2017 there was a release every 3 weeks on average. These releases often do provide crucial features; they also fix a large number of bugs, some of which are regressions; and new bugs are – as is often the case – introduced. **Inside Secure has an extremely comprehensive QA cycle, refined over the years, with nearly 3000 test cases (manual and automated).** Any bugs that might have been introduced in any ExoPlayer release will be caught and addressed before the release.

The same could be said about Android releases. Every time Google releases a new OS version, there are many things that could go wrong: maybe a new corner case in H.264 decoding, that had lain dormant until the previous Android version, now surfaces. Or maybe a certain Java API has been deprecated or even obsoleted. Inside Secure obtains early beta images of Android and tests them extensively on "vanilla Android" devices (such as Nexus and Pixel devices), in such a way that when the General Availability version is released, Inside Secure's Content Protection Client will be ready.

Table 1 below shows how Inside Secure Content Protection Client has tracked Android releases over the last few years. ("Inside Secure Release date" refers to the first Content Protection Client version officially compatible with the corresponding Android version).

Table 1 - Android releases and Inside Secure releases

Android version	Android release date	Inside Secure release date	After this many business days
Oreo (**v8**)	21-Aug-17	26-Sep-17	27
Nougat (**v7**)	22-Aug-16	08-Sep-16	14
Marshmallow (**v6**)	05-Oct-15	02-Nov-15	21
Lollipop (**v5**)	12-Nov-14	13-Nov-14	2

Myth #4:

With a single code base it is possible to support Amazon FireOS, too

FireOS is, per Wikipedia[4],

“ An Android-based mobile operating system produced by Amazon for its Fire Phone and Kindle Fire range of tablets, Echo and Echo Dot, and other content delivery devices like Fire TV; the tablet versions of the Kindle e-readers are the Fire range. It is forked from Android. “

Given its common heritage with Android, and given the fact that ExoPlayer only uses a minimal subset of the Android APIs, it stands to reason that the GitHub version of ExoPlayer will play back video contents with minimal differences from the way it does on plain Android.

Facts

FireOS has substantial differences from Android. So substantial, in fact, that Amazon had to fork ExoPlayer in order to make it work on FireOS. FireOS only tracks Android loosely, and the latest (as of this whitepaper) FireOS version, 6.2.1.0, is based on Android 7.1, which at the time was already one year old. Additionally, the OS has been customized enough to render the usage of vanilla ExoPlayer extremely challenging.

Inside Secure, on the other hand, tracks both the regular ExoPlayer GitHub stream and the Amazon-forked one, and is in close contact with Amazon engineering to ensure the most recent fixes are applied to the Content Protection Client ExoPlayer codebase. In fact, sometimes Amazon will perform FireOS OS-level fixes upon suggestions by Inside Secure.

By using the Inside Secure version, you ensure that playback will happen smoothly regardless of whether you use Android or FireOS.

[4] https://en.wikipedia.org/wiki/Fire_OS, retrieved on 7-Dec-2017

REASONS TO CHOOSE INSIDE SECURE WITH EXOPLAYER

Value-added features

Inside Secure Content Protection Client adds a number of desirable features on top of vanilla ExoPlayer. See Table 2 below for a side-by-side comparison.

Table 2 - Inside Secure with ExoPlayer: some of the value-added features

Feature	ExoPlayer	Inside Secure with ExoPlayer
Playback of MPEG-DASH and Smooth Streaming streams	✓	✓
Playback of HLS streams	✓ (cleartext only)	✓ (cleartext and encrypted)
Download & Play offline	✓ (with some effort)	✓ (easy)
DRM renewability	✓ (slow, dependent of OEM)	✓ (instant, with app update)
Rooted device detection	✗	✓
Enforcement of Compliance & Robustness Rules	✓ (partial, slow to renew)	✓ (complete, continuous renewall)
FireOS Support	✗	✓
Multi-DRM	✓ (depends on device)	✓ (always PlayReady & Widevine)
Forensic watermarking	✗	✓ (optional)

Hardware pass-through to the TEE when necessary

If you as a content provider have a good reason to believe that a specific device's built-in DRM implementation is robust and feature-rich enough, Inside Secure exposes an API allowing you to bypass Content Protection Client's software DRM stack and go straight to the underlying hardware one. On the other hand, in all other cases you can continue relying on Inside Secure's software-hardened implementation.

Bank-grade robustness

One aspect that is often overlooked is the fact that DRM would be "easy", were it not for the fact that it needs extremely good resilience against attack. In fact, one might argue that this is the most important characteristic of a well-made DRM client.

Inside Secure Content Protection Client employs the same obfuscation and anti-tampering toolkit that is used in the Mobile Payments Client. Inside Secure Mobile Payments Client is used by tier-1 banks worldwide to enable secure NFC mobile payments, and it comes with extremely strict security guidelines and certifications mandated by Visa and MasterCard[5].

Keeping up with Android and with ExoPlayer releases

Inside Secure keeps constant tabs on new Android releases and new ExoPlayer releases. Should there be a regression in ExoPlayer on, say, a new Android beta version, Inside Secure's QA team will catch it early by means of a regression test suite, and the engineering team will fix it before it reaches customers.

One interface, multiple implementations

Did you know that Inside Secure Content Protection Client exposes two interfaces?

- One related to all DRM operations (e.g. provision certificates, process licenses, join domain, ...).

- One which is media player-specific, for media playback.

The DRM interface is common to all Inside Secure content protection products.

[5] Inside Secure is one of the very few companies worldwide which is both security- and functionally-certified by both Visa and MasterCard (press release)

In other words, if you are using the DRM interface with the ExoPlayer solution on Android today, you can use the same interface in the FireOS-based solution, and – if you are also an iOS licensee – a very similar one on the iOS solution. No need for a learning curve across devices.

Rooted device detection

With Content Protection Client, you don't "just" get ExoPlayer with DRM. There is much more to it than that. One of the most useful features is rooted device detection: with a simple API, you kick into action a set of heuristics aimed at detecting, sometimes from subtle clues, whether or not a device has been "rooted" (i.e. allowing one to attain privileged control over various Android subsystems which would not normally be accessible). And you don't just get a yes-or-no-answer: the API returns a complex data structure that allows you to zero in on *why* the device appears to be rooted.

Multi-DRM

By using the built-in hardware DRM implementation on an Android device, you are most of the time constrained to a single DRM system, which is almost always Widevine.

With Inside Secure Content Protection Client, you get the choice between a software-hardened PlayReady or Widevine implementation, or a pass-through to the corresponding hardware one. This brings you an unprecedented level of flexibility to the table.

Watermarking and traitor tracing

Inside Secure Content Protection Client is growing the ability to perform on-the-fly forensic (invisible) watermarking on streaming contents, allowing a service provider to identify the subscriber account which caused the content to be leaked. This is often a studios requirement for licensing Premium VoD content.

Alongside with it, as a licensee you will be able to benefit from a "web patrol" service that allows you to identify the offending devices and take corrective action. Corrective actions can vary between banning specific user accounts and blacklisting entire categories of devices.

Last but Not Least: Help is right at hand – by humans!

If you adopt ExoPlayer from GitHub, your engineering team is on its own. There are, of course, user forums and other collaborative resources, but when you have a show-stopping problem and you are going live on the App Store the next day, you want to have some certainties on the timing.

As an Inside Secure licensee, you get access to a support portal, manned by human beings (no automated or canned replies) with a first line of customer support professionals backed very closely by a second line consisting of the engineers who actually implement the software. All this with a service-level agreement that allows a fast turnaround to keep your service on the air and responsive, and consequentially your app's user ratings high.

Inside Secure (Euronext Paris – INSD) is at the heart of security solutions for mobile and connected devices, providing software, silicon IP, tools, services, and know-how needed to protect customers' transactions, ID, content, applications, and communications. With its deep security expertise and experience, the company delivers products having advanced and differentiated technical capabilities that span the entire range of security requirement levels to serve the demanding markets of network security, IoT and System-on-Chip security, video content and entertainment, mobile payment and banking, enterprise and telecom. Inside Secure's technology protects solutions for a broad range of customers including service providers, operators, content distributors, security system integrators, device makers and semiconductor manufacturers.

For more information, visit www.insidesecure.com.