Multi-DRM Strategies for Video Service Providers

Contents

Managing Multi-DRM Successfully

In delivering a compelling video experience anytime, anywhere, to any device, it is clearly the "any device" part that is the most challenging for video service providers. According to ScientiaMobile, the number of mobile device profiles in the market increased threefold in the past five years, reaching an estimated 45,000 distinct device profiles in April 2016.

As a constantly growing variety of consumer viewing devices floods the market at an increasing pace, how can video service providers support and maintain a service that runs on multiple operating environments and versions? All while dealing with numerous device configurations?

To address these challenges, many service providers are committing themselves to a multi-DRM strategy, using preinstalled, natively available DRMs ("native DRMs") instead of installing their own.

When adopting a multi-DRM approach, there are several factors you need to consider in order to manage it successfully:

- **Supporting a large variety of viewing platforms:** Different viewing platforms employ different kinds of content formats, client-server communication protocols, application development languages, and content protection capabilities. Developing services on each platform requires a specific knowledge base and development resources.
- **Orchestrating varied DRM functionality:** Different DRMs support different functionality and offer varied levels of security. It can therefore take a significant development and maintenance effort to create a rich and uniform experience across devices and DRMs.
- **Managing multiple DRM license services:** With multiple DRMs on client platforms, you need to integrate and operate multiple license services, making sure that they all speak the same entitlement language and maintaining them over time.
- **Securing the DRM system and the video service:** A native DRM system can address basic content protection requirements. But to meet the requirements for protecting premium content and to make sure of the integrity of the service, you need to build many more security controls into the video distribution system on top of the basic DRM capabilities.
- **Breach recovery:** When there is a security issue with a native DRM, recovery is not always simple, because platform vendors might take time to release a patch. Even when they do release a patch, getting it deployed on user devices can involve a complex and lengthy process.

In the next sections we expand on these points and recommend how to address them.

Supporting a Large Variety of Viewing Platforms

As a service provider looking to achieve service reach, you need to contend with several challenges. Not least are the large variety of media-enabled platforms and the constant flow of new ones. A platform, in this context, could be a media playback device such as Chromecast or Roku, a software client such as Android OS, or an Internet browser.

How do these platforms vary?

Each platform can have its own application development language and methodology, as well as proprietary content protection capabilities. As a result, you might need to develop security features multiple times in multiple languages, such as Native C/C++, ObjectiveC, Swift, Java, HTML/JS, EME/JS (one for each DRM), SilverLight .NET, Metro .NET, Brightscript, and so on.

Different platforms support different content formats and different client-server communication protocols.

Some platforms have more than one DRM. Consider, for example, the DRM setup for Internet browsers. Each browser supports a different DRM: Internet Explorer/Edge supports PlayReady, Chrome supports Widevine, and Safari supports FairPlay. When you have multiple DRMs on the host device, it is almost impossible to provide a single device identifier, because each DRM will provide its own. This will effectively fool the video service headend because the host device will appear to it as three or more separate devices. This, in turn, might lead to complex device management issues and cause features such as concurrency control to misbehave.

Other platforms might have no content protection capabilities at all or not allow any DRM technology. That might be because of limited device capabilities or because of licensing restrictions imposed by the device manufacturer.

Furthermore, new software versions for existing platforms might change device behaviors radically without any backward compatibility and on very short notice.

All these drive development costs even higher and have a knock-on effect with increased ongoing maintenance costs.

Opting for a vendor-supported multi-DRM solution that is preintegrated and backed by a roadmap can help dramatically reduce initial integration and ongoing system support costs and offer broad service reach from day one.

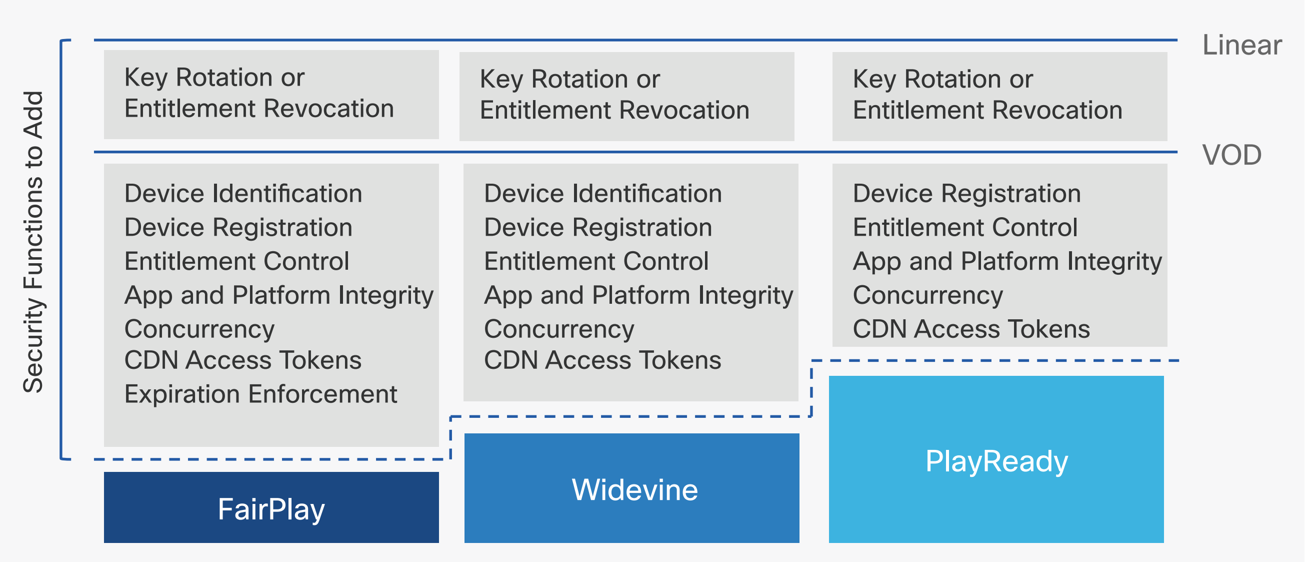
Orchestrating Varied DRM Functionality

Building a uniform user experience with aligned security capabilities that work on multiple viewing platforms requires the integration and orchestration of different DRM technologies and capabilities.

Most native DRMs are not designed to support functionality beyond basic video on demand (VOD) and linear streaming. They either prevent it or make it very difficult to implement features such as content download, home gateways (for example, home networking), multicast, DLNA import/export, and so on. In today's competitive environment, where the user experience is a primary differentiator, many service providers consider these features mandatory. Native DRMs therefore need to be augmented with additional security functionality that is well beyond their design capabilities.

Although native DRMs all provide a basic mechanism for license retrieval, content decryption, and encryption key protection, they also require significant security-related functionality enhancements in order to build even the simplest end-to-end VOD or linear (live) streaming system for a typical service provider. (See Table 1.)

Figure 1. Security Functions Needed above Native DRM for Linear and VOD Services



To illustrate the challenge, consider the following real-life examples:

- PlayReady DRM provides **device identification capability,** while others do not.
- None of the native DRMs provide **jailbreak** or **rooting detection** or effective enforcement of related **playback prevention business rules.** FairPlay does not support any business rules at all.
- In order to support a pay-per-view service, you need to implement **event-based business rules.** The content encryptor on the headend must have the interface to receive a schedule and change encryption keys on the event boundaries. The DRM client needs to be able to handle the different keys to enable playback and effectively enforce entitlements.
- Another requirement not currently supported by native DRMs is **secure forensic watermarking of content.** This is becoming a hard requirement for some premium content and an important prerequisite for effectively fighting service piracy.

Understanding the specific capabilities of each native DRM and skillfully orchestrating them into a working solution require expertise. Vendors that offer a multi-DRM system can package this expertise into their solution, simplifying the integration and deployment process for the video service provider.

Managing Multiple DRM License Services

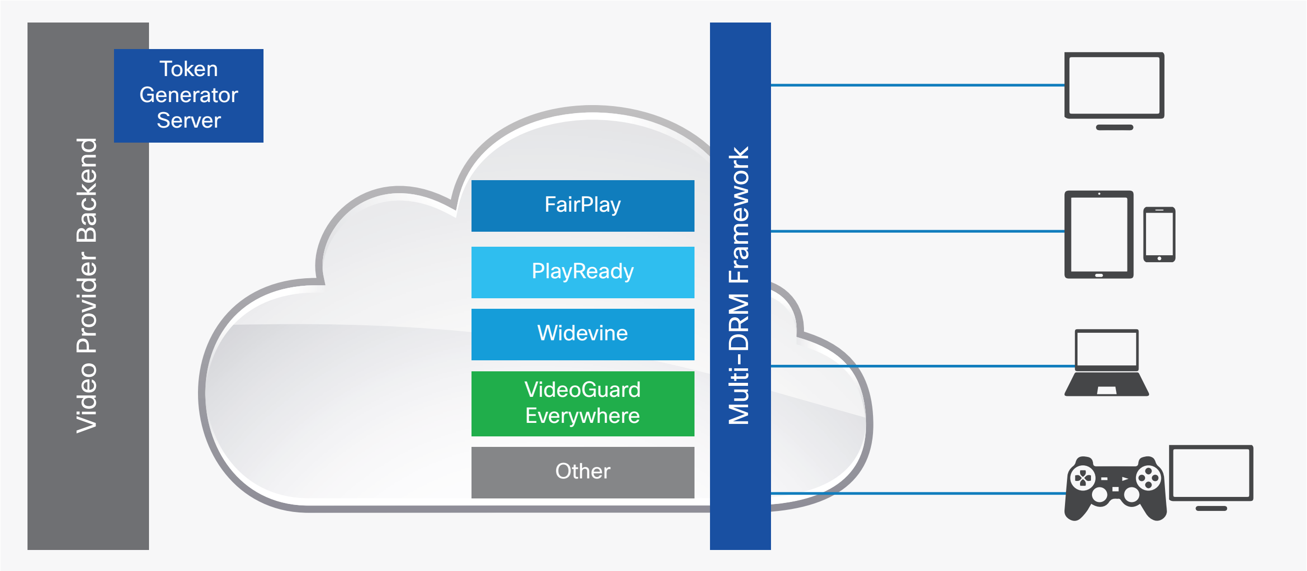
With multiple DRMs on client platforms, you need to run multiple license services. Managing multiple license services—one for each DRM—introduces additional complexity.

An effective multi-DRM headend service needs to provide a common, cross-DRM infrastructure to support capabilities such as device authentication and registration, service usage metering (for example, concurrency), and a common entitlement definition and validation language. Building such a multi-DRM headend service requires a deep understanding of what each DRM system will support and how to map entitlements from one scheme to another.

It is also advantageous to decouple authorizations coming from the control plane from the DRM headend. This decoupling lets the multi-DRM headend service become databaseless and stateless, allowing it to offer better performance, higher availability, and simpler management.

A well-designed multi-DRM headend service can offer a unified interface that abstracts away the complexities and nuances of each DRM scheme, provide a common set of infrastructure services across DRMs, and enable the DRM license service to be decoupled from the control place (See Figure 1.)

Figure 2. Simplifying Multi-DRM Implementation for Multiscreen Video Services



Securing the DRM System and the Video Service

Although securing content distribution rights is a huge undertaking on its own, service providers need to also think about **securing** their **service and revenue.** Whether deploying multi-DRM or not, protecting your service from piracy and other forms of unauthorized use requires robust protection for:

- Video playback devices
- The DRM system itself
- The client application and player
- The end-to-end system, including the service headend and distribution and communication channels

Securing the Devices

In a multiplatform service implementation, it is important to effectively detect and analyze the security posture for each device so that you can fully understand the potential risks it poses to your video service and content.

Effective jailbreak and root detection, for example, helps to determine whether the device can be trusted to play back premium content. Trustworthy attestation for the boot status of the device is important for determining whether the device is in secure boot mode and running only trusted applications. Device cloning detection is another important security control.

Securing the DRM System

Although some platforms provide security controls to make sure of the integrity of the DRM system itself, most do not. This means additional security controls need to be developed and deployed to protect the DRM system. The majority of open platforms (PCs, some mobiles, and so on), as well as jailbroken/rooted devices, do not provide any protection for the DRM system or the video application.

In order to provide adequate protection, techniques such as code obfuscation, white-box cryptography, runtime integrity monitoring, and remote serialization need to be deployed “on top” of the DRM client.

Securing the Application

In the end-to-end video service, the video application is often the weakest link. No more so than when running on open platforms. If the application is not hardened properly, hackers will be able to modify control functionality such as concurrency, authorization, and user management, rendering even the most robust DRM technology irrelevant.

Securing the integrity of the application and protecting it against hackers to prevent malicious code injections and hostile modifications to the application code and data are therefore additional critical considerations for protecting your service and content.

Securing the End-to-End System

Any video service requires added layers of protection above core DRM to secure content and service distribution across multiple consumer-owned devices and to obtain studio approval to deliver premium content on those devices.

Required technologies include basic device identification and authentication, concurrency controls, content delivery network (CDN) access control, distributed denial of service (DDoS) prevention, and other controls that are not typically provided by the DRM system.

The basic content protection capabilities offered by DRM systems fall short of securing the video service and in some cases are not enough to meet content protection requirements for premium content. An expert video security partner can help service providers make sure they achieve compliance with content protection requirements and make sure their service is protected.

Breach Recovery

It might seem implausible, but native DRMs can fail. At any given moment, you can find a newly published exploit or hack for a DRM system, trusted execution environment (TEE) used to protect the DRM system, applications for playing back DRM-protected content, and so on. This is a reality that spans across all platforms.

When an exploit or hack is published, updating the system is not always easy or straightforward. Although the platform owner might publish a patch, deploying the patch to user devices can be a slow process, with limited results.

If, for example, a hack is reported for a native DRM running in the TEE of a popular Android mobile device, then the respective chip and OS vendors might work together to quickly patch the problem. However, getting the patch installed on user devices is not always in their control, which means that only a few users will get the patch, and most others will stay vulnerable.

This problem is so concerning that the U.S. Federal Communications Commission and the U.S. Federal Trade Commission launched a probe in May 2016 to discover how manufacturers decide whether to patch a vulnerability on a particular device and how carriers review security updates, according to ZDNet. Apple, BlackBerry, Google, HTC, LG, Microsoft, Motorola, and Samsung have received letters.

With this being the reality, service providers using native DRMs to protect their content are advised to have a contingency plan.

Delivering DRM-Protected Content to the STB

This paper focuses on multi-DRM in the context of delivering content to unmanaged devices. Another important aspect, however, is the need for service providers to deploy multi-DRM so that their subscribers can view DRM-protected content on the service provider's set-top box.

Today, more and more service providers are making third-party over-the-top (OTT) services, such Netflix and YouTubeTV, a part of their service. Getting these OTT services onto their set-top box (STB) requires changes at the application and middleware levels as well as support for the relevant DRMs.

This introduces additional challenges. Here are some examples:

- Netflix requires a modified version of PlayReady, with additional Netflix-specific functions.
- YouTubeTV, in contrast, requires Widevine with EME/CDM/MSE extensions in the HTML5 client. (Note: Until recently, YouTube also allowed PlayReady, but this changed with the transition to UHD/4K).
- Both PlayReady and Widevine are licensable technologies that come with their own compliance and robustness specifications. It is left to the service provider to implement these specifications and requires significant security expertise to make sure it is done properly.
- In many cases, legacy STB devices in the field cannot support the requirements from Netflix and others. This usually requires a firmware upgrade and remote serialization with new DRM certificates and keys.

Furthermore, different DRMs running on the STB need to be orchestrated to support multiple OTT services that might run on the box. They also need to work in conjunction with each other to support features such as picture in picture.

UHD content protection requirements complicate things even further. They require that certain parts of these DRMs are placed into the SOC's TEE and that stringent compliance and robustness rules are met.

In Conclusion

A multi-DRM approach to content protection comes with the obvious benefits that the client DRM is already installed. However, in order to really benefit from this approach, you need to acknowledge what each DRM system can and cannot do and that DRM systems are not all the same.

Different DRMs support different functionality, and the supported functionality is often rudimentary relative to the business needs of a video service provider. Creating a uniform experience across devices and DRMs requires a significant development and maintenance effort.

With multiple DRMs on client platforms, you need multiple license services. This introduces operational complexity and drives up operational costs.

The DRM system can address basic content protection requirements. However, to actually meet the requirements for protecting premium content and to make sure of the integrity of the service, many more security controls need to be built into the video distribution system on top of the basic DRM capabilities.

And when the native DRM fails, recovery is not always simple, because platform vendors might take time to release a patch. Even when they do, getting it deployed on user devices can take a very long time.

To run a world-class video service, you cannot rely on the capabilities of native DRMs alone. You need a video security solution, built on a multi-DRM framework, that:

- Extends your security capabilities with the required functionality to meet content protection requirements.
- Enables service features that are demanded by users (for example, download) and required by the business (for example, pay per view).
- Secures your service against piracy and other forms of unauthorized use.
- Allows prompt recovery in case of platform risks or breaches.

About Cisco VideoGuard Everywhere

Cisco® VideoGuard® Everywhere (VGE) is an end-to-end video service protection and monetization solution for securely distributing premium video content—live, on demand, and on the go—over any network to any device.

Designed to address the business needs of pay-TV service providers, Cisco VideoGuard Everywhere supports a range of pay-TV services, including:

- Basic OTT (live, linear, and VOD)
- Advanced OTT (event-based entitlements on live channels, multiple download options, concurrency controls, watermarking, and more)
- Software-based STB services (DVR, proximity controls)
- Advanced STB services (in-home streaming, sideloading, watermarking, and more)

To make sure of service reach, Cisco VideoGuard Everywhere takes a multi-DRM approach and complements it with additional capabilities to support the business needs of service providers. These include service protection, CDN access tokens, download queuing, and fast channel change time.

For more information

For more information about Cisco VideoGuard Everywhere, visit http://www.cisco.com/go/videosecurity.