WHITEPAPER

The Ultimate Guide to DRM

BuyDRM

Table of Contents

DIGITAL RIGHTS MANAGEMENT (DRM)

Digital Rights Management (DRM) refers to technology, tools, and systems used to protect and control access to proprietary content – be it documents, data, video, or audio.

DRM gives you the power and flexibility to choose who gets to consume your content, from where, and when via custom business rules and highly secure encryption and communication protocols.

With DRM, you can decide:

- Who gets to consume your content?
- Restrict users based on geographical regions.
- Express which rights you would like to extend to those users.
- Decide who gets to watch HD, UHD, 4K content and on what devices.
- And so much more!

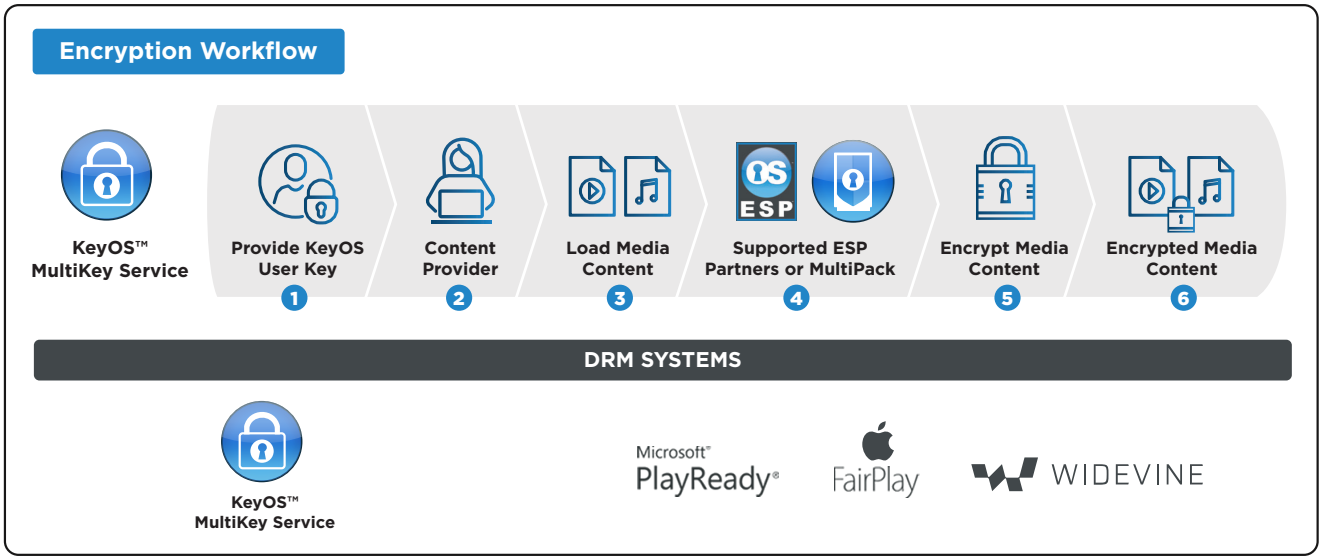
In this comprehensive whitepaper, we look at the most popular DRM technologies in-use in the movie industry today – Google Widevine, Apple FairPlay, and Microsoft PlayReady.

Also, we look at the standard building blocks of DRM and why a multi-DRM strategy is critical for your company.

THE ESSENTIAL BUILDING BLOCKS OF DRM

DRM is a combination of encryption, secure communication protocols, and business rules to control access and consumption of digital content.

In this section of the whitepaper, we will look at some of the fundamental building blocks of Digital Rights Management.



VIDEO ENCRYPTION

Encryption uses a “Key” to convert input data (plaintext) into an alternate form called ciphertext.

One of the most popular encryption techniques used in DRM is the “Advanced Encryption Standard” or “AES” for short. It is also called Rijndael (after its inventor) and was established by the U.S. National Institute of Standards and Technology (NIST) in 2001 to encrypt electronic data.

KEY, KEYID, AND THE LICENSE SERVER

Each video is encrypted using an encryption key that can be generated by the content provider manually or using software/tools provided by DRM vendors. Additionally, an association between the key and the content is generated which is called the "KeyID". It is a unique string of characters generated at the time of creating an encryption key for a particular streaming asset, or group of assets. The Encryption Key and the KeyID are stored in a secure server (Key Store) that works alongside a DRM License Server. When a player attempts to playback an encrypted asset, once the user has been authenticated, it requests the DRM license decryption key by providing that video's KeyID (sent to the player as part of the manifest/playlist). If the DRM license server is satisfied with the authenticity of the request, it will deliver the key from the Key Store to the client.

DECRYPTING VIDEO AT THE PLAYER AND THE KEY SERVER

The steps at the player are straightforward at this stage:

- The client application uses the KeyID to ask the License Server for the decryption key.
- The license server uses predefined mechanisms to recognize if the request is authentic or not.
- After the license server is satisfied with the request's authenticity, it responds with the license & decryption key.

CONTENT DECRYPTION MODULE (CDM)

Every DRM vendor such as Google, Microsoft, or Apple has its own set of rules and mechanisms for creating license requests, transferring the decryption key from the License Server, rules regarding storing the license locally on the client, license renewal, expiry, etc.

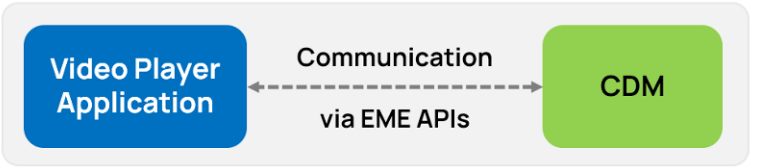
To adhere and enforce these rules, special software called the CDMs (Content Decryption Modules) are built and rigorously controlled to ensure that:

- the license requests are formed correctly as per specifications.
- decryption keys, decrypted content, or decoded videos are not leaked
- decryption keys are stored safely on-device if required.
- Licensing rules are strictly enforced on the devices.

EME OR ENCRYPTED MEDIA EXTENSIONS

EME or Encrypted Media Extensions is a layer between the player and the CDM and provides a standardized set of APIs for players (applications) to communicate with the CDMs.

Having a standardized set of APIs enables CDM vendors and player vendors to develop their software independently knowing that there is a standardized communication layer in between.



VIDEO DECODING AND DISPLAY – HOW TO KEEP IT SECURE?

After a video is decrypted, it must be decoded and displayed to the user without leaking the decrypted data. The CDM (Content Decryption Module) plays a vital role in preventing data leaks and it can:

- decrypt the movie and hand over the bitstream to the application (not very secure)
- decrypt, decode, and pass on the decoded frames of video to the platform's display engine.
- decrypt, decode, and display the video by itself (most secure)

Essentially, depending on the level of security enforced in the DRM license, the process of decrypting, decoding, and displaying the video can take place in software or the device's hardware (more secure).

GOOGLE WIDEVINE

Widevine DRM is a DRM technology from Google (which acquired Widevine in 2010). Widevine is an extremely popular DRM technology with support for the Android operating system, Smart TVs, browsers, Gaming Consoles, etc. It also provides support for MPEG-DASH, HLS, MSS streaming protocols and supports CENC and CMAF.

HISTORY & VERSIONS OF WIDEVINE

Let us look at the history of Widevine and its support. There are two versions of Widevine – Classic & Modular.

WIDEVINE CLASSIC:
Widevine Classic is supported only in legacy devices that require the media to be packaged into a proprietary .WVM format. Widevine Classic is not used anymore and had support in old Android (3.1 ~ 5.1) versions, legacy Smart TVs, Google TV, etc.

WIDEVINE MODULAR:
Widevine Modular is the current version of Widevine and has support for MPEG-DASH, HLS, MSS streaming protocols. It also has support for CMAF, CENC, and HTML5 standards such as EME & MSE.

BUILDING BLOCKS OF WIDEVINE DRM

The main building blocks of Widevine are:

WIDEVINE LICENSE SERVER:
Widevine provides a License Server to hold information needed for encrypting and decrypting media securely. It has two main jobs –

- After the media is packaged and encrypted, information is sent to the license server that helps it uniquely identify and associate a license key with the media.
- During video playback, the license server authenticates the request from the player for the license and encryption keys, fetches the decryption key from the key store (database), and responds to the player (or client) with the license and the decryption keys. The communication between the packager, License Server, and the player are all encrypted and sent over HTTPS.

SHAKA PACKAGER:
Widevine provides a complete open-source MPEG-DASH packaging software called the Shaka Packager that

- Converts all your video files to the fMP4 container format and packages it to the MPEG-DASH protocol.
- Encrypts each file with CENC using license information obtained from the Widevine License Server.
- Creates an mpd file or a Manifest file with all the information describing the DASH-packaged media.

OEMCRYPTO MODULE:
The OEMCrypto Module decrypts the content using information from the Player (and the License Server). The OEMCrypto Module is in the Trusted Layer tied to the device hardware.

HOW DOES WIDEVINE DRM WORK?

Step 1: The application downloads the MPEG-DASH or CMAF manifest from the CDN and determines if the video is encrypted using Widevine. Then, the initialization data (InitData) is extracted from the content and sent to the player.

Step 2: The player sends the InitData to the Content Decryption Module (CDM).

Step 3: The CDM receives the InitData from the player and creates an encrypted "License Request" and sends it back to the player.

Step 4: After the player receives the license request, it sends it to the Widevine License Server via a proxy.

Step 5: The License Server receives the request from the player, decrypts it, extracts the InitData, and uses it to find the license from its database. The License Server then takes the Decryption Key along with license information, creates an encrypted message, and sends it back to the player.

Step 6: The player receives the encrypted message from the License Server and passes it to the CDM (via the EME).

Step 7: The CDM passes on this encrypted message to the OEMCrypto Module which resides in the Trusted layer of the device. The actual content decryption takes place in the OEMCrypto Module. In some implementations, the media decoding takes place there in the OEMCrypto module as well.

Step 8: After the content is decrypted and decoded, it is rendered on the screen and is not stored anywhere on the device.

WIDEVINE SECURITY LEVELS

Next, we look at the different security levels provided by Widevine – L1, L2, and L3. This is important on both the device and the content delivery fronts. Let us see why!

L1 OR SECURITY LEVEL 1

- L1 is the highest level of security in Widevine and provides hardware-level decryption where the content decryption, media decoding, and rendering are all done from within the Trusted Execution Environment.
- If you want to stream HD content from content providers, your device will need to meet L1 specifications.

L2 OR SECURITY LEVEL 2

- In L2, only the media decryption is performed within the TEE and the decrypted video is sent to the application for decoding and rendering. Understandably, L2 is less secure than Google Widevine L1.

L3 OR SECURITY LEVEL 3

- L3 is the least secure of all three Widevine DRM Security levels and is used in low-end hardware without a TEE and the decryption is performed in a software-CDM.
- It is common for content providers to block encrypted HD video playback in devices with L3 security.

WHERE IS WIDEVINE SUPPORTED?

Widevine Modular or simply, Widevine is supported on several platforms such as –

- Android (4.4+)
- Android TVs
- Amazon Fire TV
- Chromecast
- Smart TVs
- Browsers such as Chrome, Firefox, Edge.

MICROSOFT PLAYREADY DRM

PlayReady is a DRM technology and platform provided by Microsoft for content protection and distribution and provides a secure client-side implementation, a license server, and the handling of licenses and keys in transit. PlayReady provides useful features such as Metering, Domains and Domain-based licenses, Breach Response, Key Rotation for Live streaming, etc.

BUILDING BLOCKS OF PLAYREADY DRM

The building blocks of Microsoft PlayReady DRM are as follows –

VIDEO PACKAGER AND THE CONTENT PACKAGING SERVER

- The content which needs to be protected is packaged into either MPEG-DASH, HLS, or Microsoft Smooth Streaming (MSS) streaming formats. The input videos can be in either fmp4, mp4, or the ismv / isma container formats used in MSS.
- The packaged and encrypted content is stored in a Content Packaging Server while the license information and encryption keys are sent to the License Server.

KEY AND KEYID

- When content is encrypted using PlayReady, the Encryption Key is sent to the License Server and is a private value. The KeyID is a public value and is embedded in clear (readable) format in the manifest by a packager.

LICENSE SERVER

- The License Server's primary duty is to provide license information to an authenticated client so that the client can play back a protected video. It relies on a Key Management System (KMS) or a database to store the Keys and KeyID.

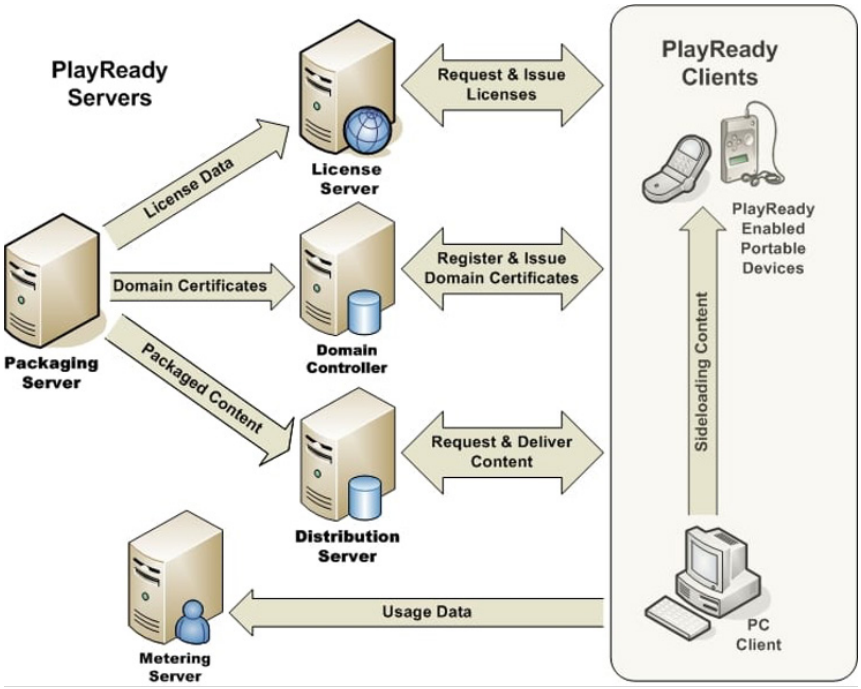
PLAYREADY DOMAIN CONTROLLER (OR DOMAIN SERVER)

- A PlayReady Domain is a feature of PlayReady that allows content providers to provide licenses bound to a group of devices, and not just one device. This group of devices is called a "Domain" and they can share the license with each other instead of contacting the License Server each time.
- The Domain Controller specifies rules that decide the definition of a Domain (e.g. identity of a single user or family). It also enforces the policy defining how many devices or PCs may join the domain, leave the domain, or renew their domain certificate.

METERING SERVERS

- PlayReady Metering is a feature of PlayReady that counts the number of times a piece of content was played which is useful in tracking the number of times the content was played, and royalty pay-outs.
- A PlayReady Metering Server aggregates the counts from all the PlayReady clients and lets the clients know that they can reset their individual counts.

HOW DOES MICROSOFT PLAYREADY DRM WORK?



PlayReady Workflow from Microsoft

Step 1: The content is first packaged, encrypted, and then sent to the Distribution Server. The license and encryption keys are sent to the License Server, and the domain information is sent to the Domain Server.

Step 2: The client informs the CDM that the content is encrypted, and the CDM generates a license request that contains the KeyID and client information. The player sends this request to the License Server.

Step 3: The License Server uses the KeyID to get the Keys from the Key Management System and sends it to the client along with other relevant license information (securely). The response from the License Server contains –

- The decryption keys.
- The licensing rights, and the right restrictions and right modifiers, also known as the conditions of the license.

Step 4: The player receives the license from the License Server and passes it to the CDM.

Step 5: The CDM (or the hardware component in some devices) will take the response from the License Server, extract the content key from it, and use the Key to decrypt, decode, and render the video. PlayReady also provides a License Store that can be implemented on the client to store the key and the rights at the client itself. This is sometimes called the Hashed Data Store or HDS.

LICENSE ACQUISITION IN PLAYREADY

PlayReady has two methods of license acquisition -

- **Proactive:** The client sends license requests to the License Server even before playback starts.
- **Reactive:** After the user presses “Play”, the client searches for the license in the License Store (a Hashed Data Store) in the client. If it finds a license for that content, it can start playback immediately. If it does not find a license, it needs to ask the License Server for a new license.

PLAYREADY SECURITY LEVELS

PlayReady has three Security Levels or SLs – SL150, SL2000, and SL3000.

SL150

- SL150 is the lowest level of security in PlayReady and is not recommended for product-releases and should only be used in closed-door testing.
- In SL150, nothing is protected (assets, clients, keys, etc) and everything can be hacked.

SL2000

- The SL2000 security level can be used in production scenarios on commercial content because the content, assets, keys, clients are protected either in software or hardware.

SL3000

- SL3000 is the most secure form of PlayReady DRM and was introduced in 2015 along with PlayReady v3 and includes hardware protection of assets, clients, and keys via the Trusted Execution Environment.

HOW ARE THE SECURITY LEVELS ENFORCED IN PLAYREADY?

The PlayReady Security Level is a property of the Client Certificate embedded in the client during the time of manufacturing. When a client makes a license request to the PlayReady server, it must indicate what its security level is. The License Server examines the client’s security level and returns the content key for the resolutions tied to the client’s security level. The License Server also specifies the MinimumSecurityLevel value that is set to either SL150, SL2000, or SL3000. The client needs to check this value and refuse to play the stream if its own security level is lower than the minimum value specified.

ADDITIONAL FEATURES OF MICROSOFT PLAYREADY

PlayReady supports Subscription, Pay-Per-View, Rental, Purchase, and Ad-based business models. Let us take a brief look at these business models.

SUBSCRIPTION

In the Subscription model, the license can be time-based or chained.

- Time-based model: Here, the license is valid only for a specified period. If the subscription is active and the license is close to expiry, then the license is automatically renewed. If the subscription is cancelled, then the license is automatically invalid.
- Chained: In this model, there is the concept of a root license and a leaf license. The root license contains the subscription's time-based policies, and the leaf licenses are tied to it. When the root license expires, the leaf licenses also expire.

PAY-PER-VIEW

In this business model, PlayReady pre-delivers the content licenses to the subscribers and acknowledges that the licenses were successfully stored on the client's device.

RENTAL

This is a time-based license with several flexible features to help you specify for how long the license is valid (1 day, or 30 days, or 24 hours from pressing play, etc.)

PURCHASE

In this business model, the assumption is that the license is issued with no expiration at all. Additionally, when a person purchases content, PlayReady enables copying to a different device, or converting the content into any other DRM scheme.

WHERE IS MICROSOFT'S PLAYREADY DRM SUPPORTED?

PlayReady is supported on several platforms and streaming formats such as,

- Xbox & PlayStation
- Chromecast, Roku, Android TV, Amazon Fire TV
- Natively on browsers such as Windows Edge and IE11
- Smart TVs (Samsung, LG, Philips, Toshiba, Panasonic.)
- MPEG-DASH, HLS, and Microsoft Smooth Streaming (MSS) streaming formats. The input videos can be in either fmp4, mp4, or ismv / isma used in MSS.
- PlayReady v4 has support for CENC-based encryption using either AES–CTR and AES–CBC encryption modes.

APPLE FAIRPLAY STREAMING

FairPlay Streaming is a DRM technology from Apple to securely deliver streaming media using the HLS (HTTP Live Streaming) protocol via encryption, secure key exchange, and on-device protection. FairPlay Streaming DRM is supported natively on iOS, iPadOS, watchOS 7, tvOS, and macOS.

BUILDING BLOCKS OF FAIRPLAY STREAMING (FPS)

In this section, let us learn about the fundamental building blocks of FairPlay Streaming DRM.

HLS PACKAGER

The first step is to package the media as specified by the HTTP Live Streaming (HLS) protocol and this is accomplished using specialized software. The packaging process involves splitting the video into small pieces or chunks and creating a file called a playlist that describes the chunks and the order in which they are to be delivered.

ENCRYPTION USING SAMPLE-AES, OR AES-128

The content is encrypted using AES-128 CBCS encryption after being packaged. Here, CBCS stands for Cipher Block Chaining. Apple FairPlay allows you to encrypt your videos using either SAMPLE-AES or AES-128. Both techniques use AES-128-bit encryption but apply it differently to the videos.

SAMPLE-AES

In this technique, only samples of the audio packets and video frames are encrypted using AES-128 with Cipher Block Chaining (CBC). By not encrypting the entire media segments, power is conserved, and the efficiency of the encryption process is improved. Despite encrypting only a portion of a segment, there is no loss in protection when you use SAMPLE-AES and it is just as secure as the alternate, AES-128 technique.

AES-128

In this technique, the entire segment (audio and video) is encrypted with a 128-bit key, Cipher Block Chaining (CBC), and Public-Key Cryptography Standards (PKCS7) padding. For further details on the types of encryption, please refer to the MPEG-2 Stream Encryption Format for HTTP Live Streaming specification from Apple Inc.

CLIENT APPLICATION

The Client Application refers to the application used to playback the videos on Apple's operating systems such as iOS, tvOS, and macOS. It is responsible for sending request messages to the License Server to obtain the decryption keys.

AVFOUNDATION

AVFoundation is a powerful framework from Apple that enables an application to play HLS streams in addition to other important functionalities. AVFoundation is defined by Apple as follows:

AVFoundation is the full-featured framework for working with time-based audio-visual media on iOS, macOS, watchOS, and tvOS. Using AVFoundation, you can easily play, create, and edit QuickTime movies and MPEG-4 files, play HLS streams, and build powerful media functionality into your apps.

APP DELEGATE

The App Delegate sits at the heart of the application and acts as a "controller" for the application. In FairPlay Streaming, the app delegate handles the communication between the player, AVFoundation Framework, and the Key Server.

KEY SERVER AND KEY SECURITY MODULE (KSM)

The Key Server manages the keys used for encrypting and decrypting FairPlay-protected content. The Key Security Module (KSM) receives & decrypts the license request from the player.

HOW DOES FAIRPLAY STREAMING WORK?

Step 1: The process starts when the user opens the app and presses "Play" to start watching the content.

Step 2: The app supplies the content's m3u8 file to AVFoundation to begin media playback.

Step 3: AVFoundation downloads the m3u8 file and parses it.

Step 4: AVFoundation searches the m3u8 file for the #EXT-X-KEY tag to determine if the video is encrypted. If it is encrypted, AVFoundation will ask the App Delegate for the Content Key to decrypt the content.

Step 5: In return, the App Delegate requests the AVFoundation Framework to generate the Server Playback Context (SPC) message.

Step 6: The App Delegate sends the SPC to the Key Server. Following this, the KSM in the Key Server unwraps the SPC and, the Key Server uses the information in the SPC to do a Content Key look-up. The Content Key is sent to the FSM, which wraps it into a CKC (Context Key Context).

Step 7: The KSM sends the CKC to the AVFoundation App Delegate. The delegate pushes the CKC into AVFoundation.

Step 8: AVFoundation uses the Content Key inside the CKC to decrypt, decode, and display the content to the user.

INTERESTING FEATURES OF FAIRPLAY STREAMING

DUAL EXPIRY WINDOWS FOR VIDEO RENTAL

Apple FairPlay has a "Dual Expiry Windows" feature that can be used with Persistent Keys for offline playback. A persistent key is a key stored securely on the device and can be used to playback rented content for a predefined time-period without needing to contact the license server (offline playback).
In the rental business model, there is a need for two expiry windows defined as follows –

- first window: when a user rents a movie, that movie can be watched within a 30-day window (for example)
- second window: once the user presses play, the movie must be watched within a 48-hour window. After this window expires, even the 30-day window expires.

To account for the "second window" rental model, FairPlay Streaming introduced the Dual Expiry Windows feature where,

- the First Key from the license server establishes the longer rental period (storage duration).
- the Second Key is obtained when the user begins playback, and it supersedes the First Key. When the second key's expiry window is exceeded, the user cannot access or playback that content any longer.

AVCONTENTKEYSESSION FOR OBTAINING ENCRYPTION KEYS

AVContentKeySession is a class in AVFoundation for handling decryption keys and was announced in WWDC 2017. It provides more control over the loading and lifecycle of Content Keys and is aimed at de-coupling key-loading from the media playback lifecycle. Using AVContentKeySession, the application can request for the content keys even before the user presses the Play button (referred to as "key preloading"). By pre-emptively loading content keys, a content provider can cut down on the start-up delay (latency) and improve the user's experience.

WHERE IS FAIRPLAY STREAMING SUPPORTED?

Support for FairPlay Streaming is available on the following platforms –

- Safari browser
- iOS Devices (natively in the Safari browser, or in a native iOS application)
- iPadOS
- Apple TV (tvOS 10.0+)
- Airplay (Apple's wireless content delivery protocol)

MULTI-DRM

Multi-DRM as the name suggests, is the use of more than one DRM technology to satisfy the mind-boggling combinations of devices, capabilities, business-rules, operating systems, and standards used in video streaming.

Even though DRM technology is rather straightforward to understand, the application of DRM in commercial video streaming is often not easy and involves a lot of tricky decisions.

Let us look at some of the decisions that are typically faced when choosing a DRM technology:

- What is the streaming protocol being used? MPEG-DASH, HLS, MSS, or a combination of the three?
- What container formats should we support? mp4 or ts for MPEG-DASH and HLS, respectively? Or CENC and CMAF?
- Which video players should we support? Web (HTML5), Android phones/TVs, Apple (iOS and tvOS), Roku, Smart TVs (Samsung, LG, etc.), Amazon Fire TV, etc.
  Which DRM should we use for each device?
- If I use CMAF and CENC, does my ecosystem support AES-CBC cbcs mode throughout? Why is this important? Well,
  - > Apple FairPlay supports only AES-CBC cbcs mode.
  - > HLS supports only AES-CBC cbcs mode (irrespective of CMAF)
  - > Widevine and PlayReady support both AES-128 CTR cenc or AES-128 CBC cbcs modes.
  - > MPEG-DASH with CMAF supports both AES-128 CTR cenc or AES-128 CBC cbcs modes.
  - > MPEG-DASH without CMAF supports only AES-128 CTR cenc mode.
- What is the impact of customers on legacy hardware?
- How do I deploy DRM at-scale without impacting the start-up delay (latency) and operational costs?
- Do I need Google Widevine, Apple FairPlay, and Microsoft's PlayReady? All three? Either one?

This is just the tip of the iceberg when it comes to designing and deploying a Studio Approved, commercial DRM solution that is designed to be secure and scalable.

In a fragmented and complex OTT ecosystem which has several inter-dependencies, it is crucial to pull in the experts and take their help in navigating a complex space. Especially, in DRM where the implications of getting it wrong can result in the loss of subscribers, or worse, intellectual property worth millions of dollars. This expertise provided by multi-DRM vendors who specialize in untangling and simplifying DRM deployments across a variety of playback platforms, streaming formats, and devices can prove to be priceless.

BUYDRM'S KEYOS PLATFORM

The KeyOS™ MultiKey™ Service is powered by Microsoft PlayReady, Google Widevine Modular, and Apple FairPlay DRM. Including support for CMAF, HLS, MPEG-DASH and Microsoft Smooth Streaming. By supporting adaptive streaming technologies MultiKey enables smooth playback of media to clients via HTTP and HTTPS.

STREAMS, DOWNLOADS, ONLINE AND OFFLINE

Support for all kinds of content delivery approaches including streaming, downloads, persistent licensing, and offline playback.

READY-TO-DEPLOY PLAYERS & PACKAGER ALLOW FOR SUPPORT ACROSS ALL MAJOR BROWSERS

From packaging to playback BuyDRM has you covered. Our packager MultiPack ingests MP4's and outputs CMAF, HLS, MPEG-DASH, MSS encrypted with Widevine, PlayReady and FairPlay. Our fully featured HTML5 Player – WebPlay, supports these file formats across all major browsers. And finally, our MultiPlay SDKs for iOS and Android include a player, download manager, and DRM library for downloads and offline support in iOS and Android apps.

BuyDRM™ is a leading global provider of Content Security Services for the entertainment, education, enterprise, and hospitality industries. As an OVHcloud company, BuyDRM's KeyOS content security platform powers many of the largest brands in media and technology.

With decades of market-leading experience implementing commercial content security solutions and media technologies, BuyDRM has amassed substantial success stories for many of today's largest brands such as ABC (Australian Broadcasting Corporation), AMPAS (The Academy), Blizzard Entertainment, Cinedigm, Crackle, Daily Rounds, Deluxe Digital, EPIX, FuboTV, Funimation, POPS Worldwide, Rakuten Viki, Redbox, SBS Belgium, Sinclair Digital and Zee5.

For more information, please visit www.buydrm.com and corporate.ovhcloud.com/en/

BuyDRM provides content security services to the world's leading media & entertainment brands

2303 R.R. 620 S. | Suite 160-155 | Austin, Texas 78734
info@keyos.com | www.BuyDRM.com | 512.377.1340

Proudly Made in Austin, Texas

BuyDRM an OVHcloud company