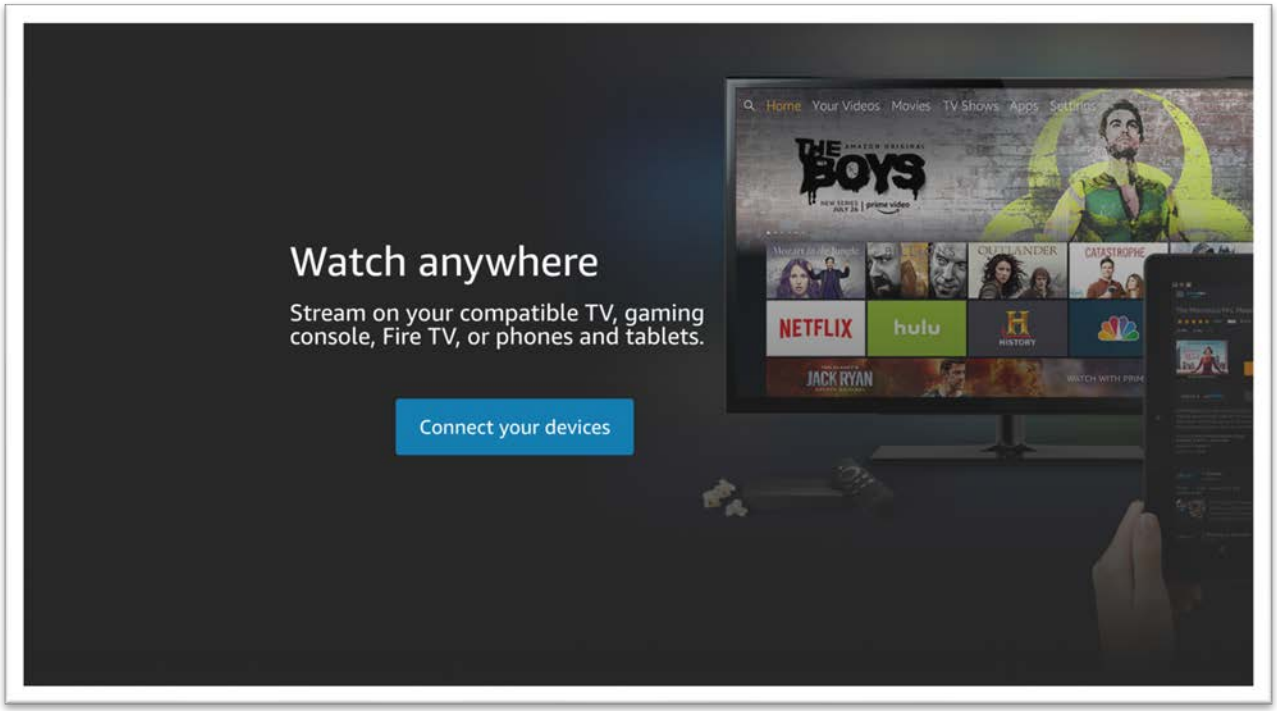
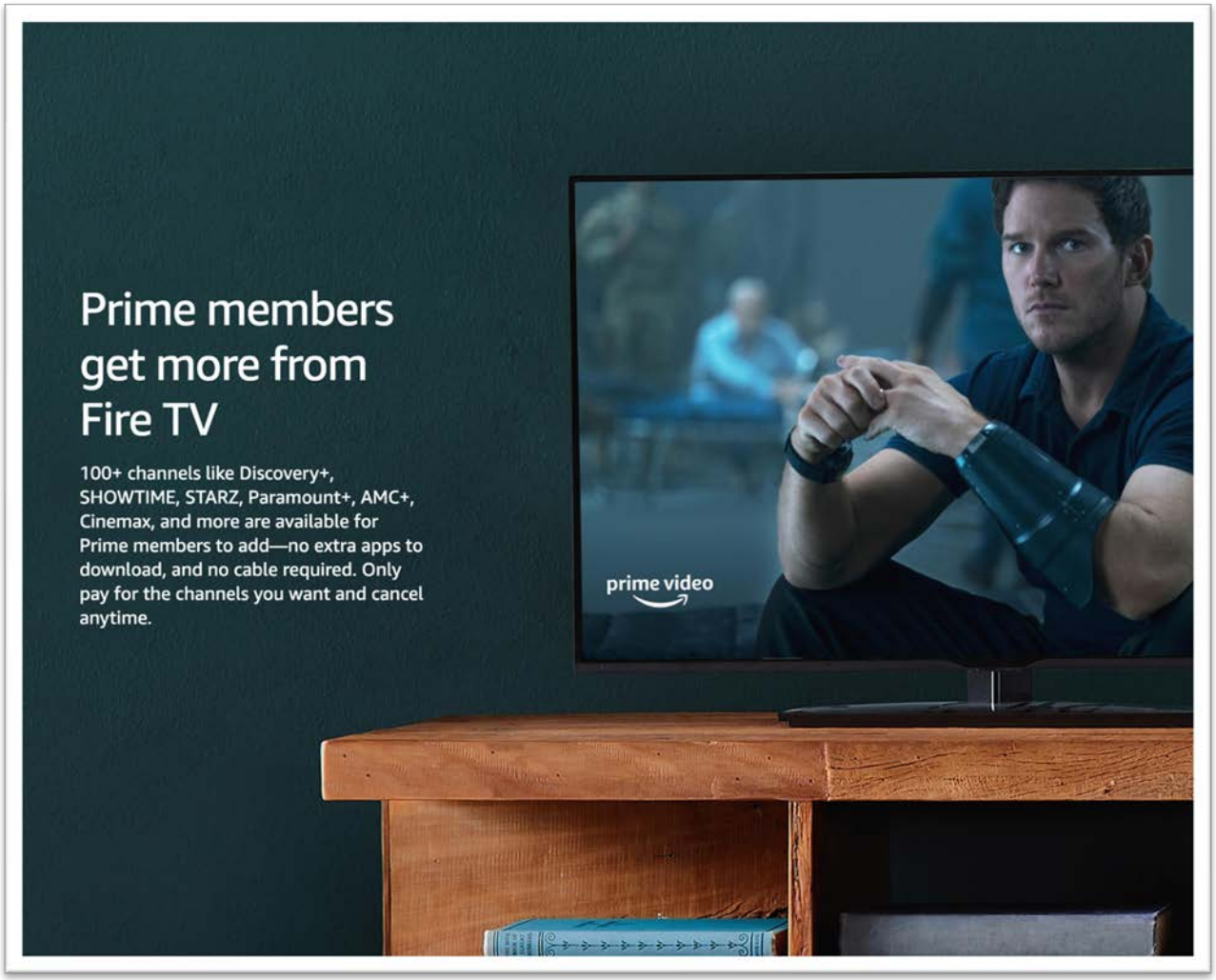
EXHIBIT 49

EXHIBIT 49
UNITED STATES PATENT NO. 11,102,553 ("553 PATENT")
INFRINGEMENT CLAIM CHART COMPARING CERTAIN AMAZON ACCUSED PRODUCTS[1] TO ASSERTED INDEPENDENT CLAIM 11

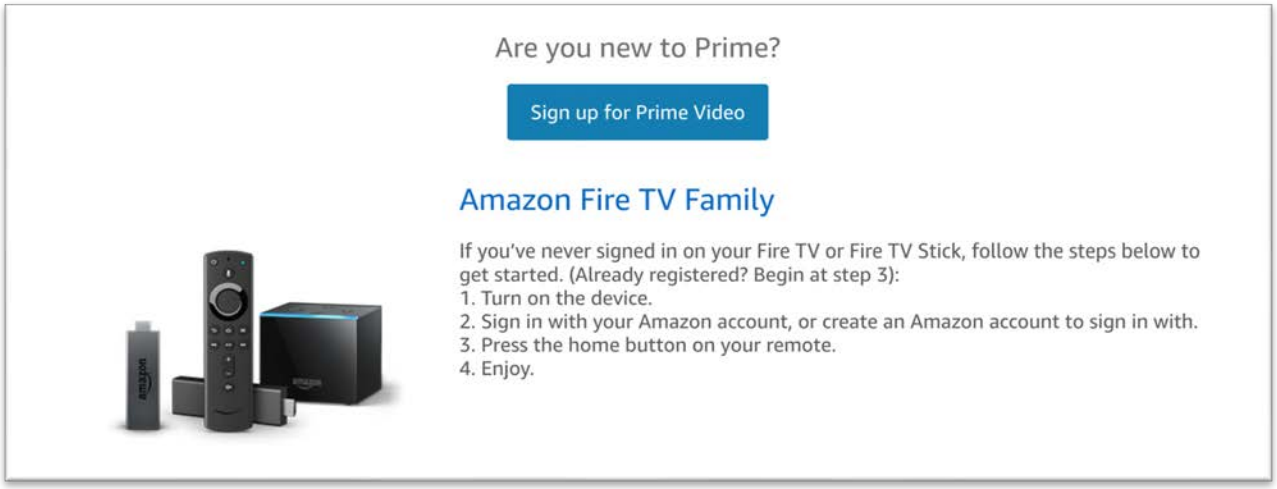
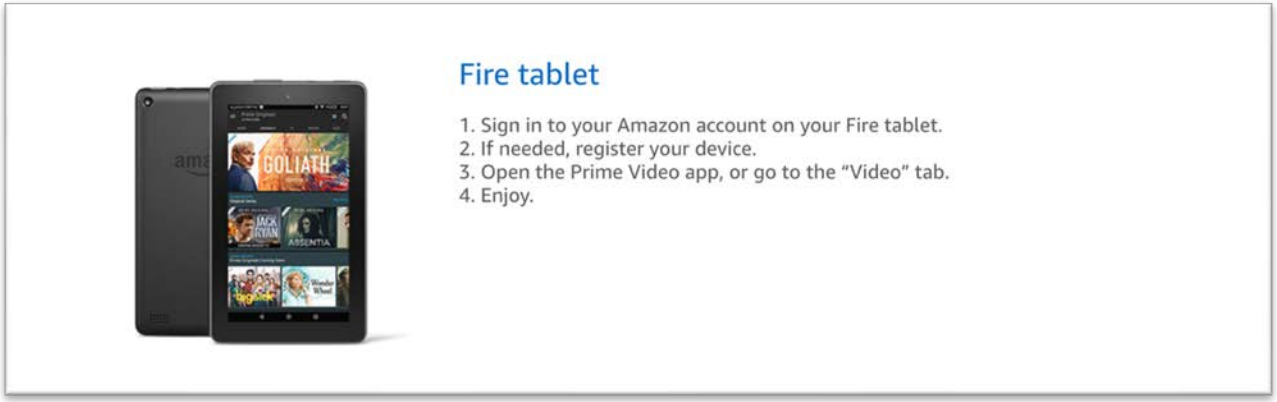
553 Patent	Amazon's Fire TV Stick (3rd Gen)
[11-pre] A playback device for playing back encrypted video, the playback device comprising:	An Amazon Fire TV Stick (3rd Gen), Build Model AFTSSS ("Fire Stick"), as exemplary of all Amazon Accused Products with the Amazon Fire OS, comprises "[a] playback device for playing back encrypted video." Amazon provides the streaming platform for its Prime Video service, which allows its consumers to "[w]atch anywhere" and "[s]tream on your compatible TV, gaming console, Fire TV, or phones and tablets." Amazon practices this claim directly when testing its Prime Video service and alternatively induces software developers and consumers to practice this claim indirectly in the manner charted herein.

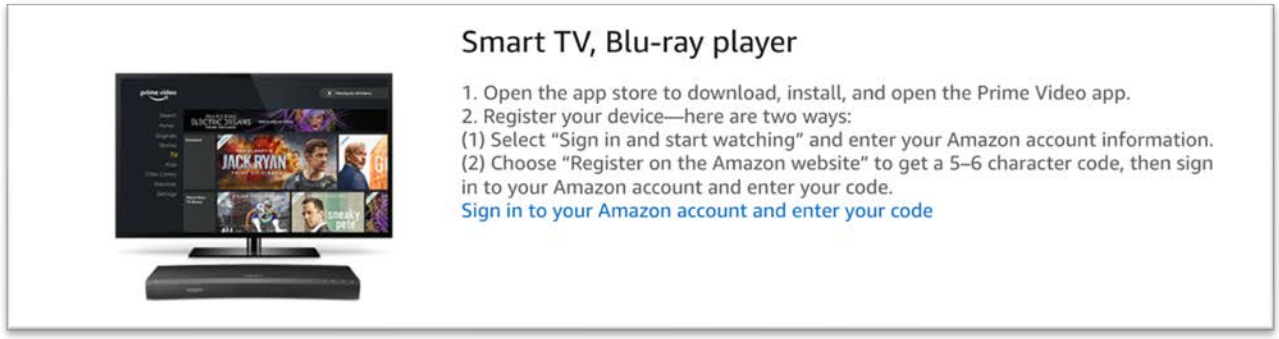
[1] The Accused Products are various generations and screen sizes of Amazon Fire, Fire HD, Fire HD Plus, Fire Kids, Fire Kids Pro, Fire HD Kids, Fire HD Kids Pro tablets (collectively, "Amazon Fire tablets"), various generations of Amazon Fire TV Stick and Fire Cube devices (collectively, "Amazon Fire TV/Cube devices"); various generations and screen sizes of Amazon Echo Show devices ("Amazon Echo Show devices"); and various screen sizes of Amazon Fire TV 4-Series and Fire TV Omni Series sets (collectively, "Amazon Fire TV sets").

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	 Source: https://www.amazon.com/gp/video/getstarted/ref=sv_atv_5.

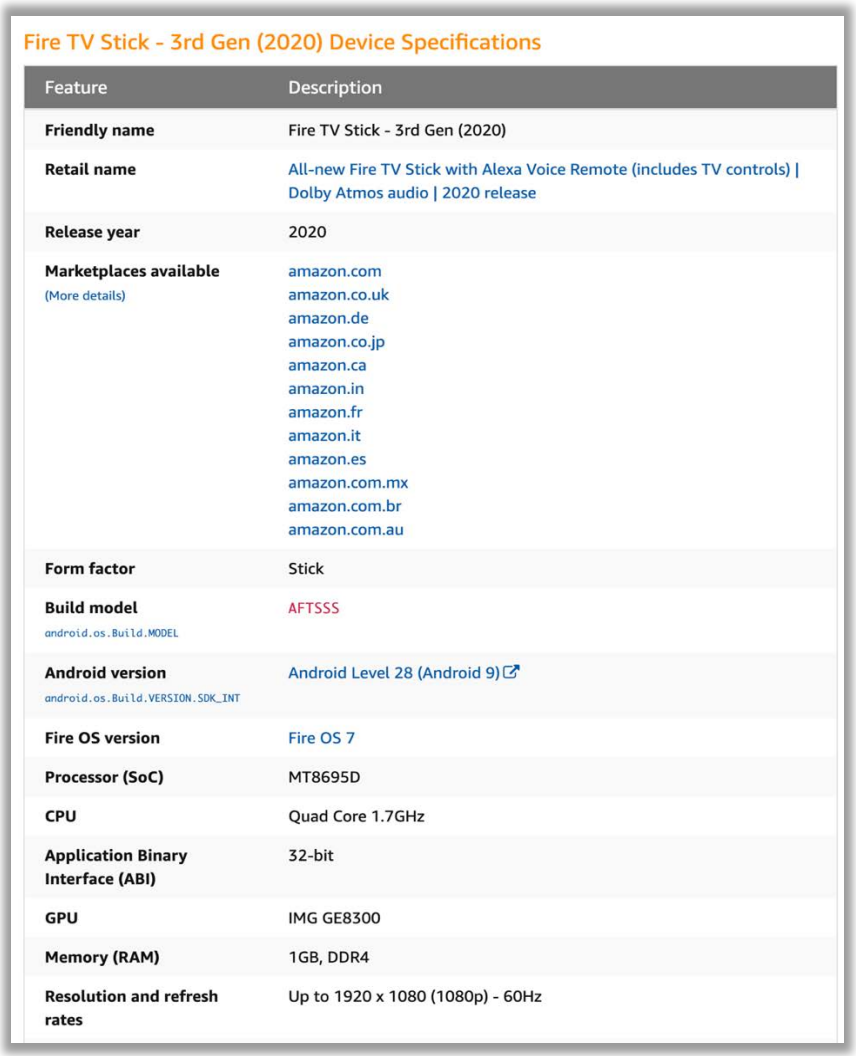
553 Patent	Amazon's Fire TV Stick (3rd Gen)
	 Source: https://www.amazon.com/fire-tv-stick-with-3rd-gen-alexa-voice-remote/dp/B08C1W5N87.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	As shown below, Amazon provides the exemplary Fire Stick for decoding multimedia files received from Amazon's Prime Video streaming platform.  Source: https://www.amazon.com/fire-tv-stick-with-3rd-gen-alexa-voice-remote/dp/B08C1W5N87. As shown below, Amazon provides consumers with instructions on how to access the Prime Video streaming platform and thereby induces consumers to practice this claim indirectly.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	 Source: https://www.amazon.com/gp/video/splash/device_linking.  Source: https://www.amazon.com/gp/video/splash/device_linking.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	 Source: https://www.amazon.com/gp/video/splash/device_linking. Thus, the Fire Stick includes "a playback device for playing back encrypted video."

553 Patent	Amazon's Fire TV Stick (3rd Gen)
[11(a)] a set of one or more processors; and	An Amazon Fire TV Stick (3rd Gen), Build Model AFTSSS ("Fire Stick"), as exemplary of all Amazon Accused Products with Amazon Fire OS, comprises "a set of one or more processors." The Fire Stick includes at least one processor. For example, the Fire Stick contains a MediaTek MT8695D quad-core 1.7GHz processor.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	

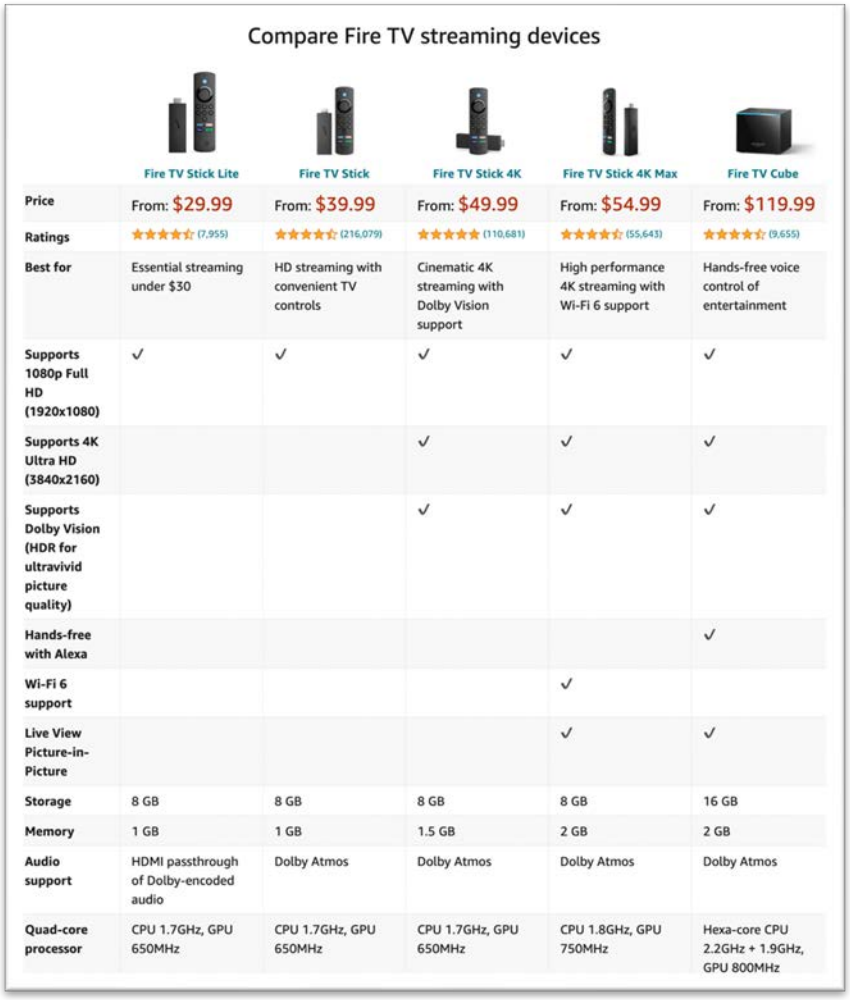
Fire TV Stick - 3rd Gen (2020) Device Specifications

Feature	Description
Friendly name	Fire TV Stick - 3rd Gen (2020)
Retail name	All-new Fire TV Stick with Alexa Voice Remote (includes TV controls) | Dolby Atmos audio | 2020 release
Release year	2020
Marketplaces available (More details)	amazon.com amazon.co.uk amazon.de amazon.co.jp amazon.ca amazon.in amazon.fr amazon.it amazon.es amazon.com.mx amazon.com.br amazon.com.au
Form factor	Stick
Build model android.os.Build.MODEL	AFTSSS
Android version android.os.Build.VERSION.SDK_INT	Android Level 28 (Android 9)
Fire OS version	Fire OS 7
Processor (SoC)	MT8695D
CPU	Quad Core 1.7GHz
Application Binary Interface (ABI)	32-bit
GPU	IMG GE8300
Memory (RAM)	1GB, DDR4
Resolution and refresh rates	Up to 1920 x 1080 (1080p) - 60Hz

Source: https://developer.amazon.com/docs/fire-tv/device-specifications-fire-tv-streaming-media-player.html.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	Thus, the Fire Stick includes "a set of one or more processors."

553 Patent	Amazon's Fire TV Stick (3rd Gen)
[11(b)] a non-volatile storage containing a playback application for causing the set of one or more processors to perform the steps of:	An Amazon Fire TV Stick (3rd Gen), Build Model AFTSSS ("Fire Stick"), as exemplary of all Amazon Accused Products with Amazon Fire OS, comprises "a non-volatile storage containing a playback application." The Fire Stick includes a non-volatile storage containing a playback application. For example, the Fire Stick, an Amazon Fire playback device, contains 8GB of eMMC flash storage.

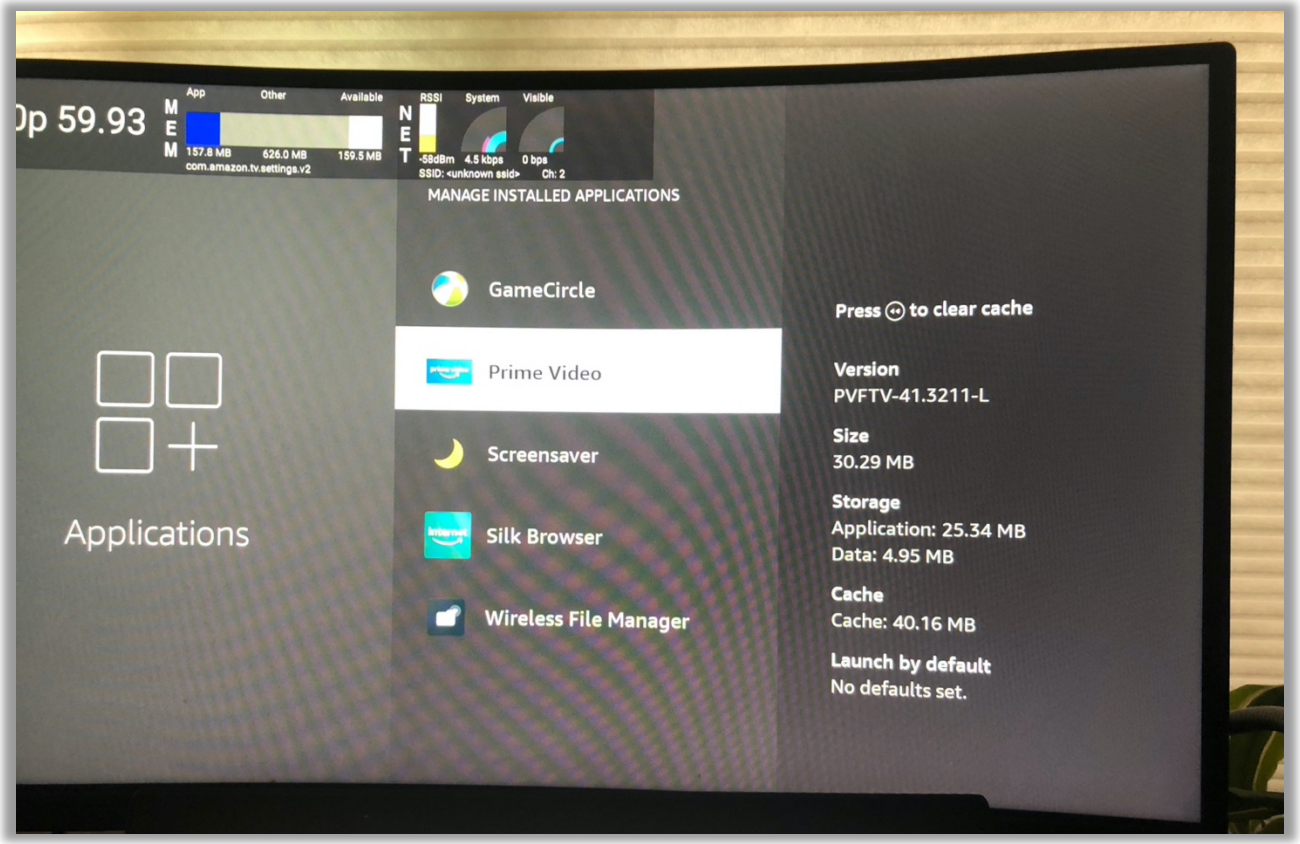
553 Patent	Amazon's Fire TV Stick (3rd Gen)
	

Compare Fire TV streaming devices

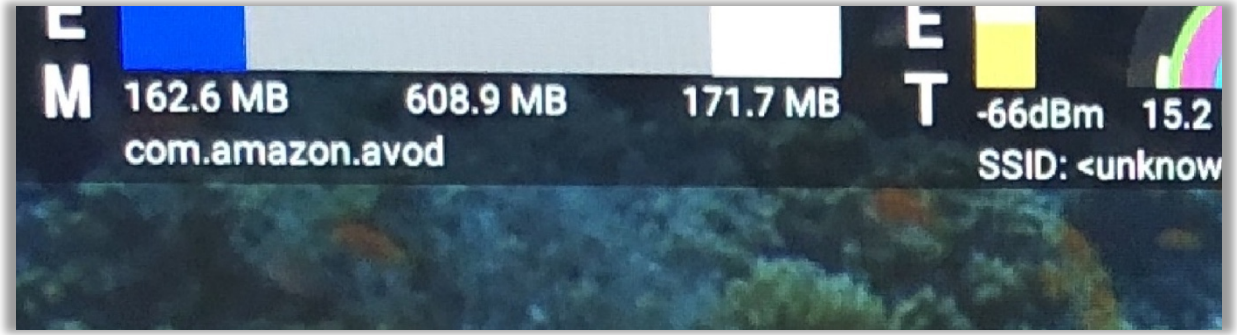
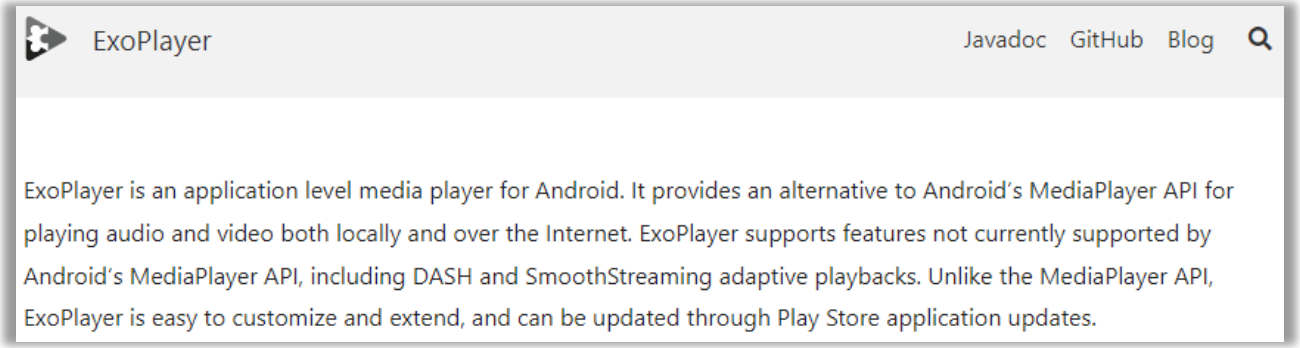
	Fire TV Stick Lite	Fire TV Stick	Fire TV Stick 4K	Fire TV Stick 4K Max	Fire TV Cube
Price	From: $29.99	From: $39.99	From: $49.99	From: $54.99	From: $119.99
Ratings	★★★★½ (7,955)	★★★★½ (216,079)	★★★★★ (110,681)	★★★★½ (55,643)	★★★★½ (9,655)
Best for	Essential streaming under $30	HD streaming with convenient TV controls	Cinematic 4K streaming with Dolby Vision support	High performance 4K streaming with Wi-Fi 6 support	Hands-free voice control of entertainment
Supports 1080p Full HD (1920x1080)	✓	✓	✓	✓	✓
Supports 4K Ultra HD (3840x2160)			✓	✓	✓
Supports Dolby Vision (HDR for ultravivid picture quality)			✓	✓	✓
Hands-free with Alexa					✓
Wi-Fi 6 support				✓	
Live View Picture-in-Picture				✓	✓
Storage	8 GB	8 GB	8 GB	8 GB	16 GB
Memory	1 GB	1 GB	1.5 GB	2 GB	2 GB
Audio support	HDMI passthrough of Dolby-encoded audio	Dolby Atmos	Dolby Atmos	Dolby Atmos	Dolby Atmos
Quad-core processor	CPU 1.7GHz, GPU 650MHz	CPU 1.7GHz, GPU 650MHz	CPU 1.7GHz, GPU 650MHz	CPU 1.8GHz, GPU 750MHz	Hexa-core CPU 2.2GHz + 1.9GHz, GPU 800MHz

Source: https://www.amazon.com/fire-tv-stick-with-3rd-gen-alexa-voice-remote/dp/B08C1W5N87. *See also* https://www.cnx-software.com/2020/09/25/the-new-amazon-fire-tv-stick-is-powered-by-mediatek-mt8695d-soc/; https://www.makeuseof.com/tag/nand-emmc-need-know-flash-memory/.

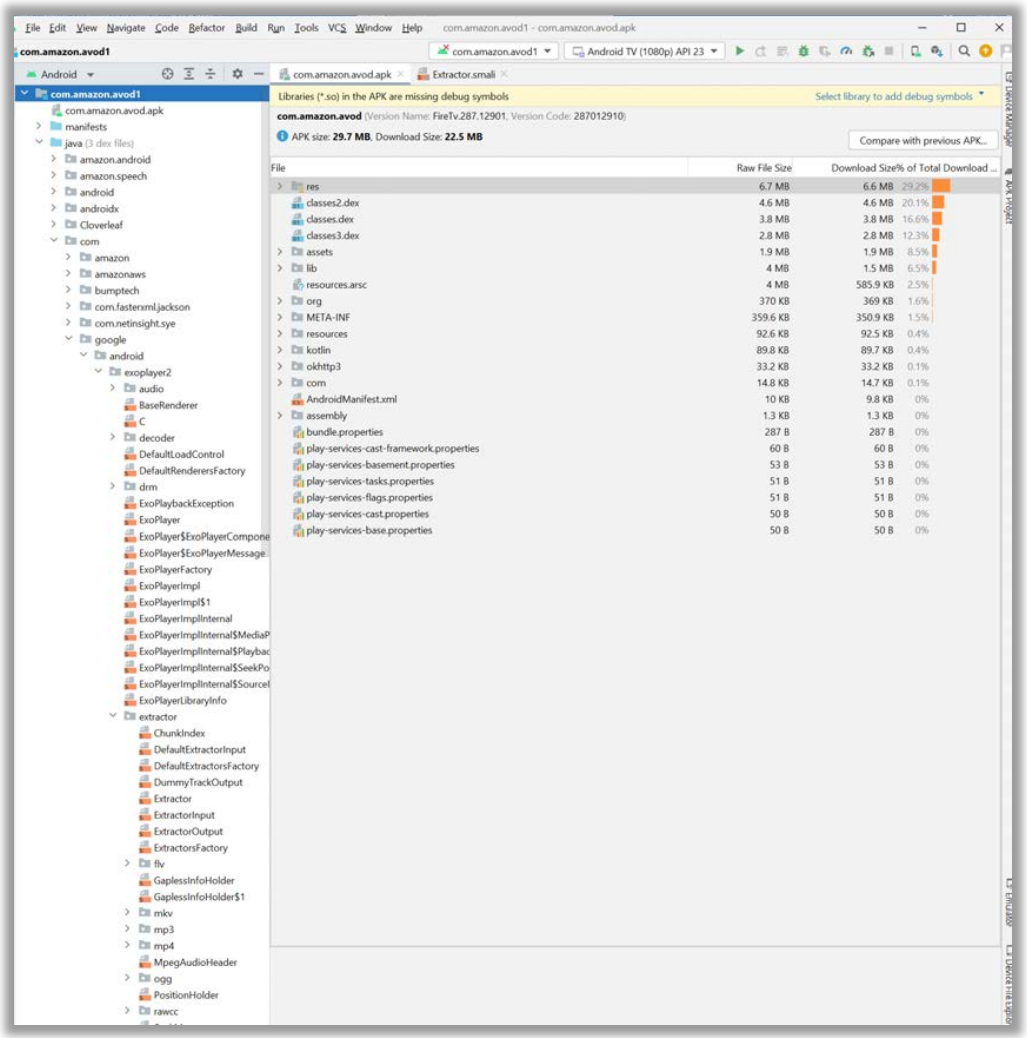
553 Patent	Amazon's Fire TV Stick (3rd Gen)
	Further, the Fire Stick "doesn't have external storage," so playback applications must reside on the Fire Stick's internal non-volatile storage. https://developer.amazon.com/docs/fire-tv/specifying-installation-location.html.  Source: https://developer.amazon.com/docs/fire-tv/specifying-installation-location.html. For example, the Fire Stick's internal non-volatile storage contains a playback application: package:/data/app/com.amazon.avod-9Mr6-mjaVpHe75uQOLftMw==/base.apk ("Prime Video application").

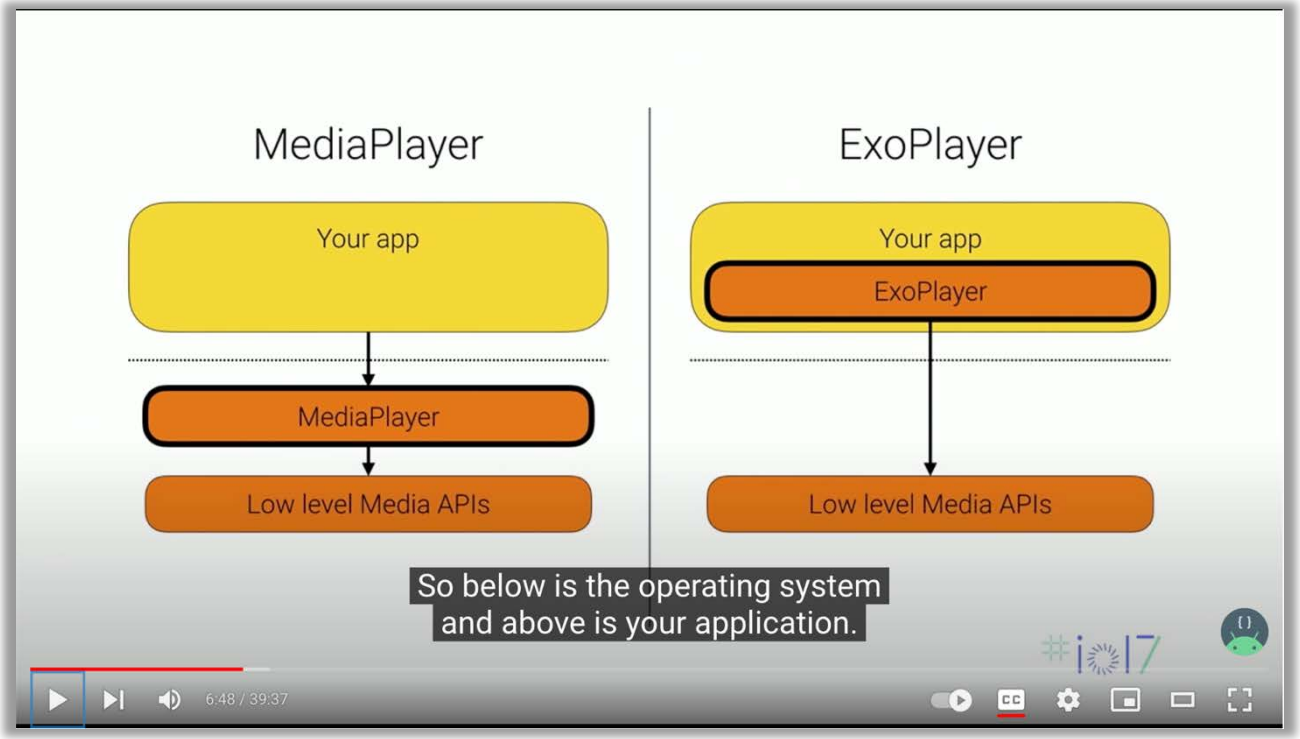
553 Patent	Amazon's Fire TV Stick (3rd Gen)
	 Source: Photograph of Fire TV settings showing the installed Prime Video application and its properties on the Fire Stick, taken by, or on behalf of, DivX. As shown below, the Prime Video application runs in the foreground of the Fire Stick when streaming Prime Video content.

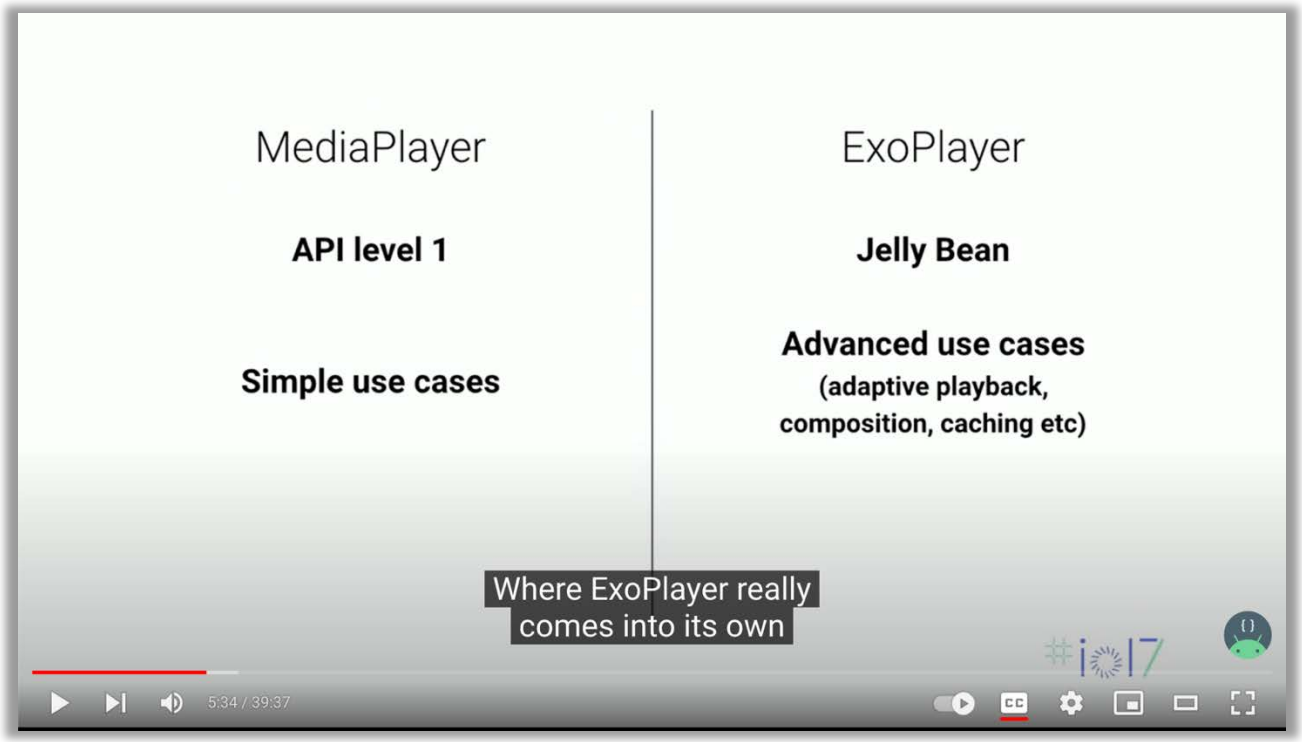
553 Patent	Amazon's Fire TV Stick (3rd Gen)
	 Source: Photograph of Prime Video playback of *Wildest Places, Wild Coasts (Episode 4)*, and overlayed System X-Ray properties on the Fire Stick, taken by, or on behalf of, DivX. For more information on System X-Ray, see https://developer.amazon.com/docs/fire-tv/system-xray.html.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	 Source: Screenshot showing name of application running in foreground in previous photograph, *i.e.*, com.amazon.avod, taken by, or on behalf of, DivX. *See* https://developer.amazon.com/docs/fire-tv/system-xray.html. The Fire Stick (in which the Prime Video application comes pre-installed) includes Android's ExoPlayer playback component ("ExoPlayer") that is contained in the non-volatile storage of the Fire Stick.  Source: https://exoplayer.dev/. As shown below in Amazon's Prime Video application, Android's ExoPlayer is also integrated in the software of various third-party applications (*e.g.*, HBO Max) for Fire OS and therefore is transmitted

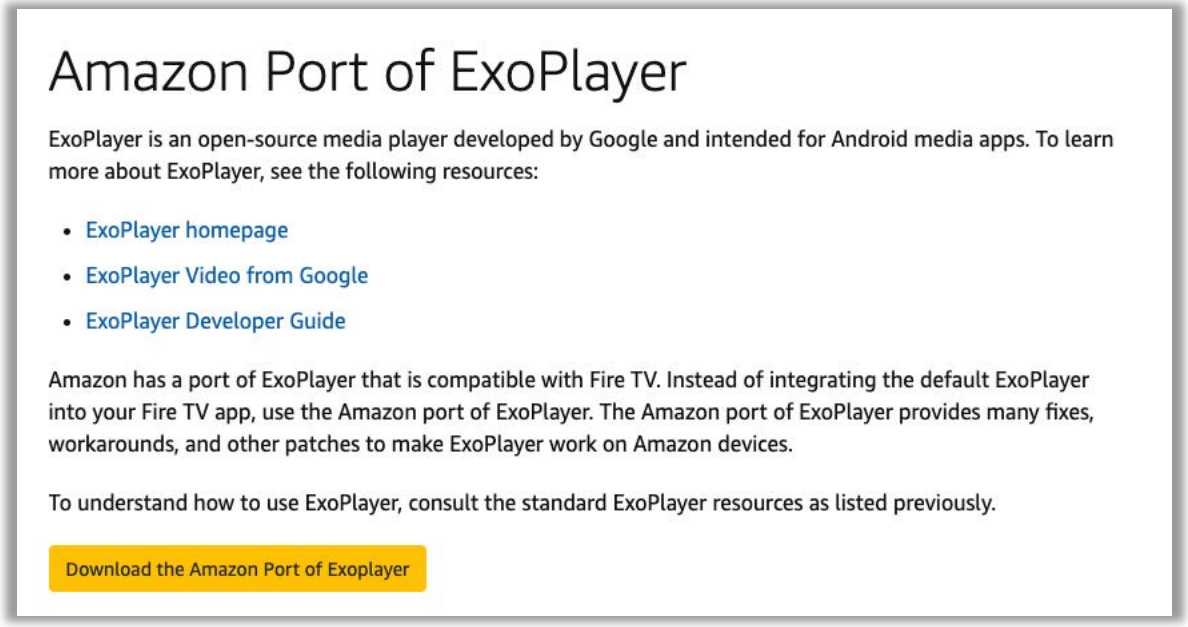
553 Patent	Amazon's Fire TV Stick (3rd Gen)
	with the third-party application (inside the application itself) when a user requests and downloads that app.



553 Patent	Amazon's Fire TV Stick (3rd Gen)
	Source: Screenshot of com.amazon.avod.apk in Android Studio showing ExoPlayer references, taken by, or on behalf of, DivX.  Source: https://www.youtube.com/watch?v=jAZn-J1I8Eg at 6:48.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	 Source: https://www.youtube.com/watch?v=jAZn-J1I8Eg at 5:34. Because Amazon Fire OS is a "fork of Android," Android's ExoPlayer lends itself to be highly compatible with Fire OS. https://developer.amazon.com/docs/fire-tv/fire-os-overview.html. Indeed, as Amazon so claims, "if your app runs on Android, it will most likely run on Amazon's Fire devices too. You can quickly check your app's compatibility with Amazon through the App Testing Service. As a developer, you might not have to adjust your Android code at all to publish your app on Amazon's platform." *Id.*

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	 Source: https://developer.amazon.com/docs/fire-tv/fire-os-overview.html. Amazon encourages software developers to download and use its port of ExoPlayer: "Instead of integrating the default ExoPlayer into your Fire TV app, use the Amazon port of ExoPlayer. The Amazon port of ExoPlayer provides many fixes, workarounds, and other patches to make ExoPlayer work on Amazon devices." https://developer.amazon.com/docs/fire-tv/media-players.html.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	 Source: https://developer.amazon.com/docs/fire-tv/media-players.html. Amazon maintains a list of licenses of third-party, open-source software that comprise its Fire TV products. *See* https://www.amazon.com/gp/help/customer/display.html?nodeId=GJZ7FQ4Z3B2Q8EJV. As shown below, Amazon lists ExoPlayer as one such licensed software component used in Fire TV.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	 Source: Screenshot of Notice Relating to Open Source Software, *available at* https://www.amazon.com/gp/help/customer/display.html?nodeId=GJZ7FQ4Z3B2Q8EJV.  Source: Screenshot of Notice Relating to Open Source Software, *available at* https://www.amazon.com/gp/help/customer/display.html?nodeId=GJZ7FQ4Z3B2Q8EJV.

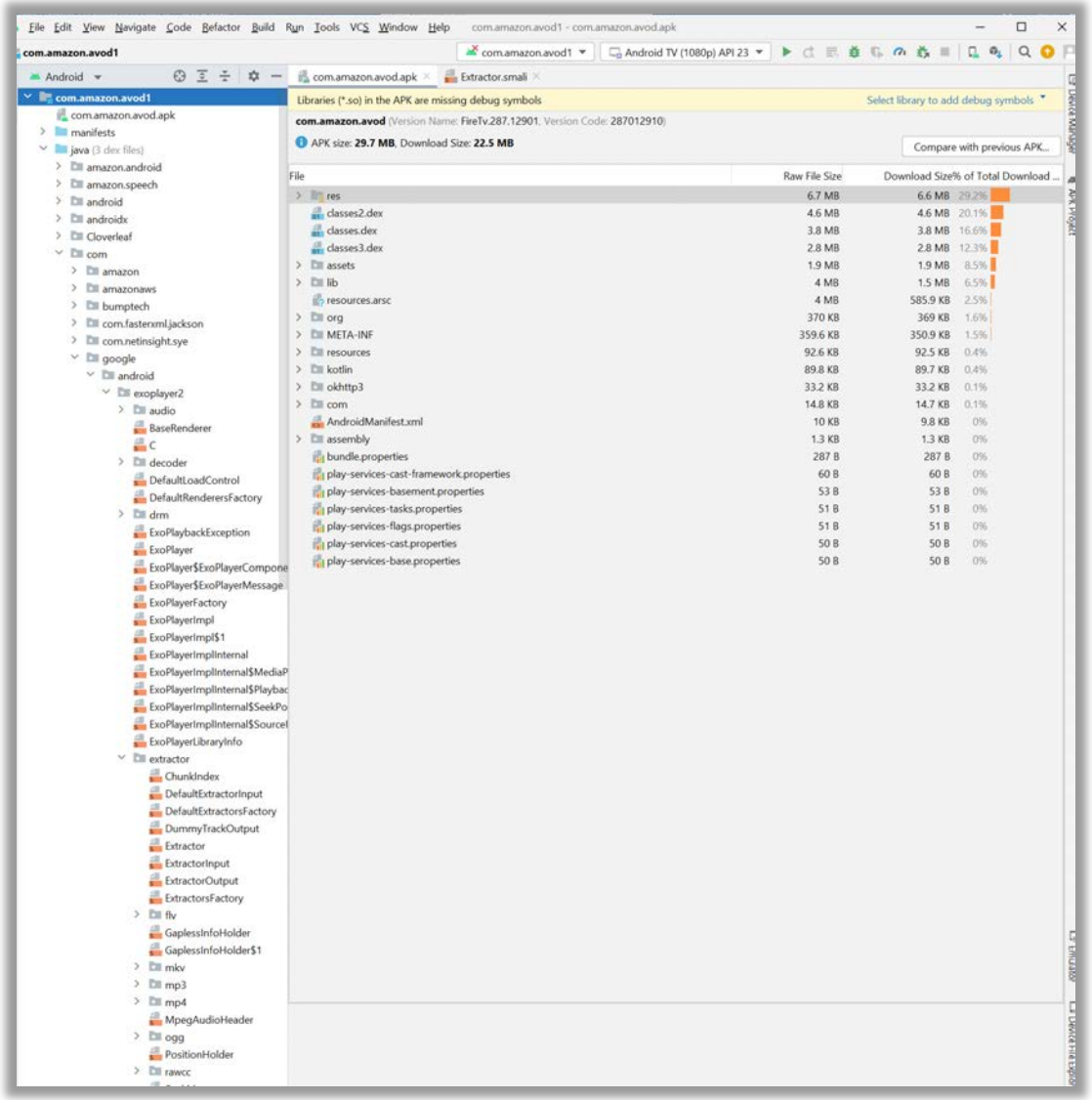
553 Patent	Amazon's Fire TV Stick (3rd Gen)
	The Fire Stick's non-volatile storage may contain other "playback application[s]" (including applications by Amazon and third parties alike) that "caus[e] the processor to perform" the recited steps, as shown further below, in the same or similar fashion, and the Prime Video application is merely representative of the other applications. Thus, the Fire Stick includes a non-volatile storage containing a playback application (*e.g.,* the Prime Video application or, alternatively, third-party streaming applications that incorporate the ExoPlayer playback component).

553 Patent	Amazon's Fire TV Stick (3rd Gen)
[11(b)(i)] obtaining, at a parser, at least a portion of a container file, wherein the at least a portion of the container file comprises:	An Amazon Fire TV Stick (3rd Gen), Build Model AFTSSS ("Fire Stick"), as exemplary of all Amazon Accused Products with Amazon Fire OS, comprises a playback application to cause the set of one or more processors to perform the steps of "obtaining, at a parser, at least a portion of a container file." For example, the Fire Stick includes a playback application such as the Prime Video application. *See* **Element [11(b)]**, *supra*. The Prime Video application (or any Amazon or third-party streaming application incorporating the ExoPlayer playback component) causes the Fire Stick to obtain a portion of a container file—*e.g.*, mp4[2] container files—with video data. As shown below in the Fire TV log[3] after initiating Prime Video playback, at timestamp 12:46:47.591, the package "AmazonVideo" initiates a "DownloadRequest" to download a mp4 container file from the following URL: `https://s3-iad-2.cf.dash.row.aiv-cdn.net/dm/2$ixOgy3TbqZyaqB6etQlGBHLFW34~/1@249c8087b7be6e96ae85a82ed64e501f/0c53/c9b1/b9df/439a-8de1-c8a86e9e8a45/2@AgBgAAACAAAJ-IAXqz7B8EC8vA==/9489fd98-158b-4940-8391-53db747ac3b2_video_11.mp4.`

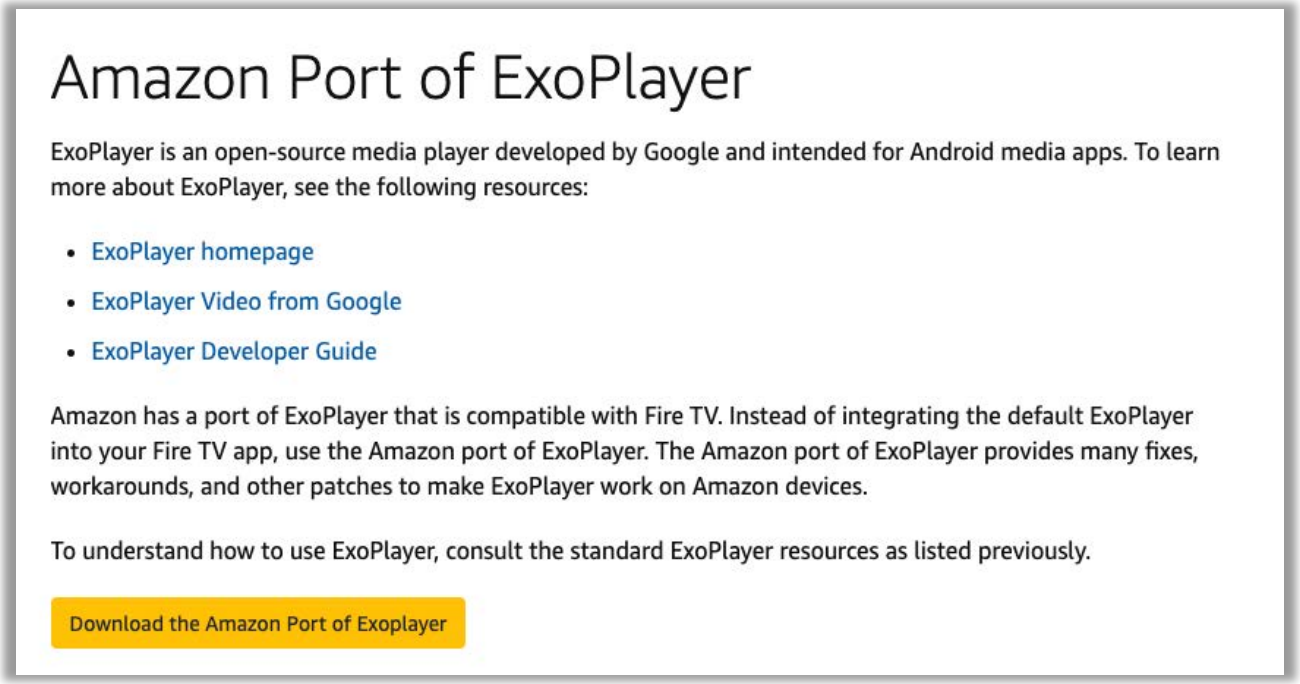
[2] MPEG-4 ("mp4") is an ISO base media file format container file standardized in ISO/IEC 14496-12 *Information technology – Coding of audio-visual objects – Part 12: ISO base media file format* ("ISO/IEC 14496-12" or "ISO BMFF Standard"). Amazon Prime Video implements MPEG-DASH (also known as "DASH," a streaming protocol as defined in ISO/IEC 23009-1 *Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats* (4th ed. 2019) ("ISO/IEC 23009-1," "DASH standard," or "MPEG-DASH standard")) using mp4 files encoded and packaged using elements and extensions specified in and consistent with the ISO/IEC 23001-7 *Information technology – MPEG systems technologies – Part 7: Common encryption in ISO base media file format files* (3d ed. 2016) ("ISO Common Encryption Standard" or "ISO/IEC 23001-7").

[3] The "log" is the system log for Android operating systems (and by extension Amazon's Fire OS because "Fire OS is a fork of Android," see https://developer.amazon.com/docs/fire-tv/fire-os-overview.html) to log messages from different processes and components residing on Fire Stick (or the subject Amazon Fire playback device) using adb logcat. For more information on logcat and log formatting, see https://developer.android.com/studio/debug/am-logcat.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	At timestamp 12:46:47.594, the package AmazonVideo successfully starts the fragment downloader. `06-20 12:46:47.591 19153 20125 I AmazonVideo: DownloadRequest downloading url: https://s3-iad-2.cf.dash.row.aiv-cdn.net/dm/2$ixOgy3TbqZyaqB6etQlGBHLFW34~/1@249c8087b7be6e96ae85a82ed64e501f/0c53/c9b1/b9df/439a-8de1-c8a86e9e8a45/2@AgBgAAACAAAJ-IAXqz7B8EC8vA==/9489fd98-158b-4940-8391-53db747ac3b2_video_11.mp4 | headers: {Range=bytes=20420-767770} | expectedSizeKB: 729 | timeoutMillis: 120000`  `06-20 12:46:47.594 19153 20050 I AmazonVideo: Start fragment downloader, succeeded: true` Source: Excerpt from log file for *Wildest Places, Wild Coasts (Episode 4)*, downloaded using the Prime Video application installed on the Fire Stick, taken by, or on behalf of, DivX. As shown below in the Fire TV log at timestamp 12:46:48.318, the Prime Application causes the set of one or more processors to perform the steps of "obtaining, at a parser" (*e.g.*, the package AmazonVideo starts parsing) "at least a portion of the container file" (*e.g.*, the fragment) after successfully starting to download the fragment. `06-20 12:46:48.318 19153 20150 I : T20150: AIVDashFragmentJNI::Java_com_amazon_avod_smoothstream_dash_Mp4FragmentJni_parse: Parsing fragment (size 48126)` Source: Excerpt from log file for *Wildest Places, Wild Coasts (Episode 4)*, downloaded using the Prime Video application installed on the Fire Stick, taken by, or on behalf of, DivX. For instance, the Prime Video application includes a parser (*e.g.*, "Extractor").

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	Source: Screenshot of com.amazon.avod.apk in Android Studio showing ExoPlayer references and Extractor, taken by, or on behalf of, DivX. The Prime Video application includes at least a portion of the ExoPlayer code base ("ExoPlayer playback component" or "ExoPlayer") utilized by streaming applications and that supports parsing container files using "Extractor." https://google.github.io/ExoPlayer/doc/reference/com/google/android/exoplayer2/extractor/Extractor.html. In addition, Amazon encourages software developers to download and use its port of ExoPlayer: "Instead of integrating the default ExoPlayer into your Fire TV app, use the Amazon port of ExoPlayer. The Amazon port of ExoPlayer provides many fixes, workarounds, and other patches to make ExoPlayer work on Amazon devices." https://developer.amazon.com/docs/fire-tv/media-players.html.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	 Source: https://developer.amazon.com/docs/fire-tv/media-players.html. Furthermore, the ExoPlayer playback component uses Android's "MediaDrm" API to support playback of digital rights management-protected content. As shown below, "[t]he MediaDrm APIs are designed to support the ISO/IEC 23001-7: Common Encryption standard, but may also be used to implement other encryption schemes." https://developer.android.com/reference/android/media/MediaDrm.html.

553 Patent	Amazon's Fire TV Stick (3rd Gen)

Digital rights management

ExoPlayer uses Android's `MediaDrm` API to support DRM protected playbacks. The minimum Android versions required for different supported DRM schemes, along with the streaming formats for which they're supported, are:

DRM scheme	Android version number	Android API level	Supported formats
Widevine "cenc"	4.4	19	DASH, HLS (FMP4 only)
Widevine "cbcs"	7.1	25	DASH, HLS (FMP4 only)
ClearKey "cenc"	5.0	21	DASH
PlayReady SL2000 "cenc"	AndroidTV	AndroidTV	DASH, SmoothStreaming, HLS (FMP4 only)

In order to play DRM protected content with ExoPlayer, the UUID of the DRM system and the license server URI should be specified **when building a media item**. The player will then use these properties to build a default implementation of `DrmSessionManager`, called `DefaultDrmSessionManager`, that's suitable for most use cases. For some use cases additional DRM properties may be necessary, as outlined in the sections below.

Source: https://exoplayer.dev/drm.html.

553 Patent	Amazon's Fire TV Stick (3rd Gen)

MediaDrm

Added in API level 18

Kotlin | **Java**

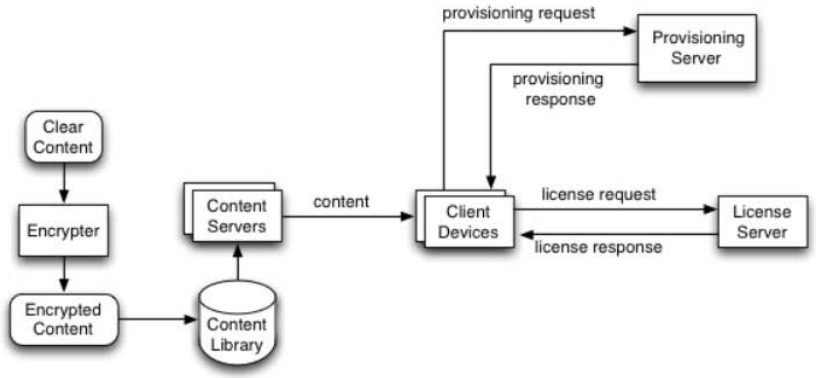
```
public final class MediaDrm
extends Object implements AutoCloseable
```

java.lang.Object
↳ android.media.MediaDrm

Source: https://developer.android.com/reference/android/media/MediaDrm.html.

MediaDrm can be used to obtain keys for decrypting protected media streams, in conjunction with `MediaCrypto`. The MediaDrm APIs are designed to support the ISO/IEC 23001-7: Common Encryption standard, but may also be used to implement other encryption schemes.

Encrypted content is prepared using an encryption server and stored in a content library. The encrypted content is streamed or downloaded from the content library to client devices via content servers. Licenses to view the content are obtained from a License Server.



Keys are requested from the license server using a key request. The key response is delivered to the client app, which provides the response to the MediaDrm API.

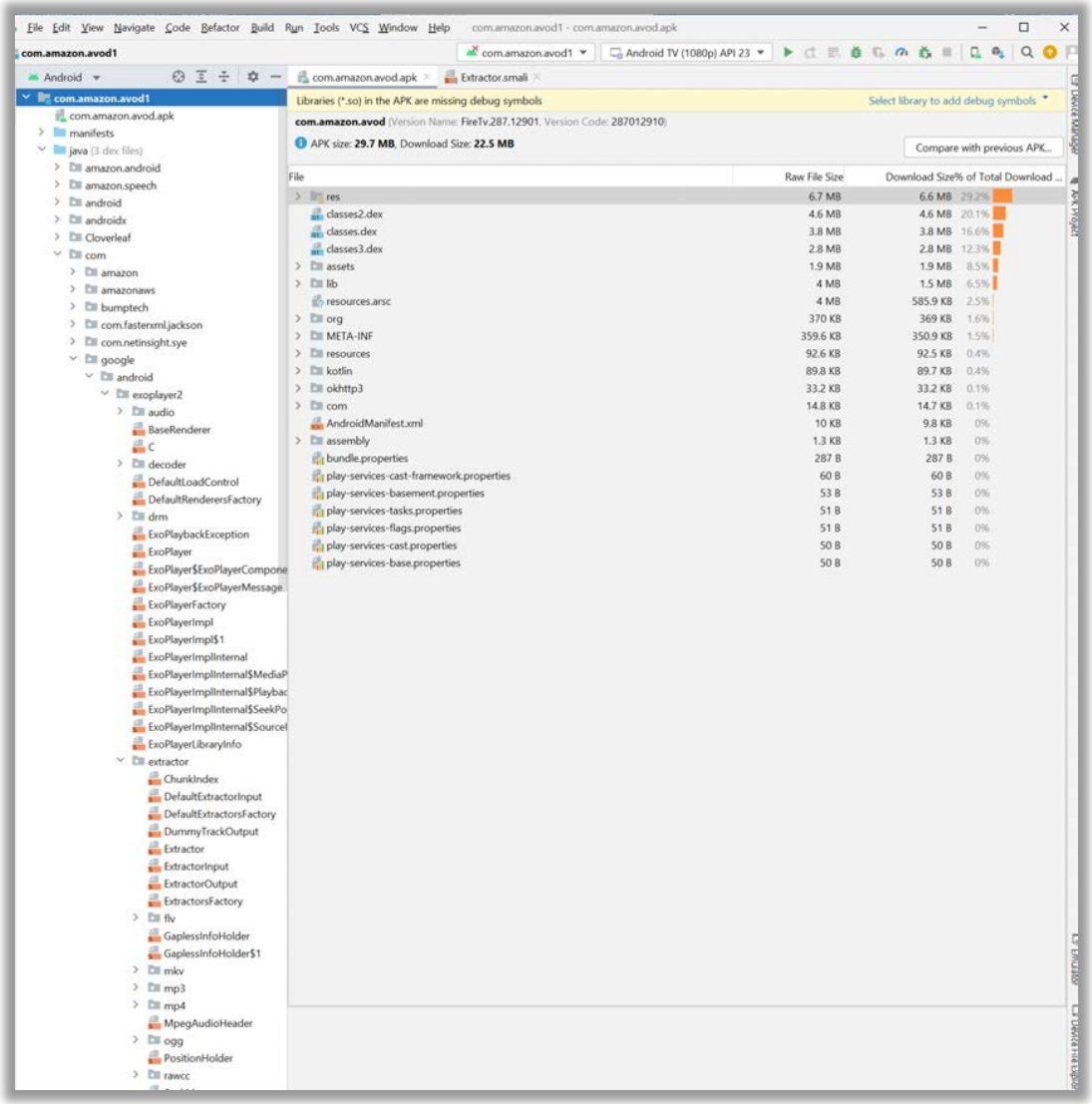
553 Patent	Amazon's Fire TV Stick (3rd Gen)
	Source: https://developer.android.com/reference/android/media/MediaDrm.html. The "MediaDrm" class specifies that the "MediaExtractor" and "MediaCodec" objects are used, as well as a "MediaCrypto" object that contains the UUID of the digital rights management (DRM) scheme. The MediaExtractor parses encoded media data which is then passed to the decoder. As shown below, "[w]hen the app has constructed MediaExtractor, MediaCodec and MediaCrypto objects, it proceeds to pull samples from the extractor and queue them into the decoder. For encrypted content, the samples returned from the extractor remain encrypted, they are only decrypted when the samples are delivered to the decoder." https://developer.android.com/reference/android/media/MediaDrm.html.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	The app first constructs `MediaExtractor` and `MediaCodec` objects. It accesses the DRM-scheme-identifying UUID, typically from metadata in the content, and uses this UUID to construct an instance of a MediaDrm object that is able to support the DRM scheme required by the content. Crypto schemes are assigned 16 byte UUIDs. The method `isCryptoSchemeSupported(UUID)` can be used to query if a given scheme is supported on the device. The app calls `openSession()` to generate a sessionId that will uniquely identify the session in subsequent interactions. The app next uses the MediaDrm object to obtain a key request message and send it to the license server, then provide the server's response to the MediaDrm object. Once the app has a sessionId, it can construct a MediaCrypto object from the UUID and sessionId. The MediaCrypto object is registered with the MediaCodec in the `MediaCodec#configure` method to enable the codec to decrypt content. When the app has constructed `MediaExtractor`, `MediaCodec` and `MediaCrypto` objects, it proceeds to pull samples from the extractor and queue them into the decoder. For encrypted content, the samples returned from the extractor remain encrypted, they are only decrypted when the samples are delivered to the decoder. MediaDrm methods throw `MediaDrm.MediaDrmStateException` when a method is called on a MediaDrm object that has had an unrecoverable failure in the DRM plugin or security hardware. `MediaDrm.MediaDrmStateException` extends `IllegalStateException` with the addition of a developer-readable diagnostic information string associated with the exception. In the event of a mediaserver process crash or restart while a MediaDrm object is active, MediaDrm methods may throw `MediaDrmResetException`. To recover, the app must release the MediaDrm object, then create and initialize a new one. As `MediaDrmResetException` and `MediaDrm.MediaDrmStateException` both extend `IllegalStateException`, they should be in an earlier catch() block than `IllegalStateException` if handled separately. Source: https://developer.android.com/reference/android/media/MediaDrm.html. As shown below, "MediaExtractor facilitates extraction of demuxed, typically encoded, media data from a data source." https://developer.android.com/reference/android/media/MediaExtractor.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	 Source: https://developer.android.com/reference/android/media/MediaExtractor.

553 Patent	Amazon's Fire TV Stick (3rd Gen)

	MediaExtractor facilitates extraction of demuxed, typically encoded, media data from a data source.
	
	It is generally used like this:
	
	`MediaExtractor extractor = new MediaExtractor();`
	`extractor.setDataSource(...);`
	`int numTracks = extractor.getTrackCount();`
	`for (int i = 0; i < numTracks; ++i) {`
	`  MediaFormat format = extractor.getTrackFormat(i);`
	`  String mime = format.getString(MediaFormat.KEY_MIME);`
	`  if (weAreInterestedInThisTrack) {`
	`  extractor.selectTrack(i);`
	`  }`
	`}`
	`ByteBuffer inputBuffer = ByteBuffer.allocate(...)`
	`while (extractor.readSampleData(inputBuffer, ...) >= 0) {`
	`  int trackIndex = extractor.getSampleTrackIndex();`
	`  long presentationTimeUs = extractor.getSampleTime();`
	`  ...`
	`  extractor.advance();`
	`}`
	
	`extractor.release();`
	`extractor = null;`
	
	Source: https://developer.android.com/reference/android/media/MediaExtractor.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	 Source: Screenshot of com.amazon.avod.apk in Android Studio showing ExoPlayer references and Extractor, taken by, or on behalf of, DivX.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	When receiving partially encrypted video streams, ExoPlayer supports the use of box structures, which contain information (*e.g.*, metadata) about a requested video. As shown below, ExoPlayer supports MPEG-DASH.

553 Patent	Amazon's Fire TV Stick (3rd Gen)

DASH

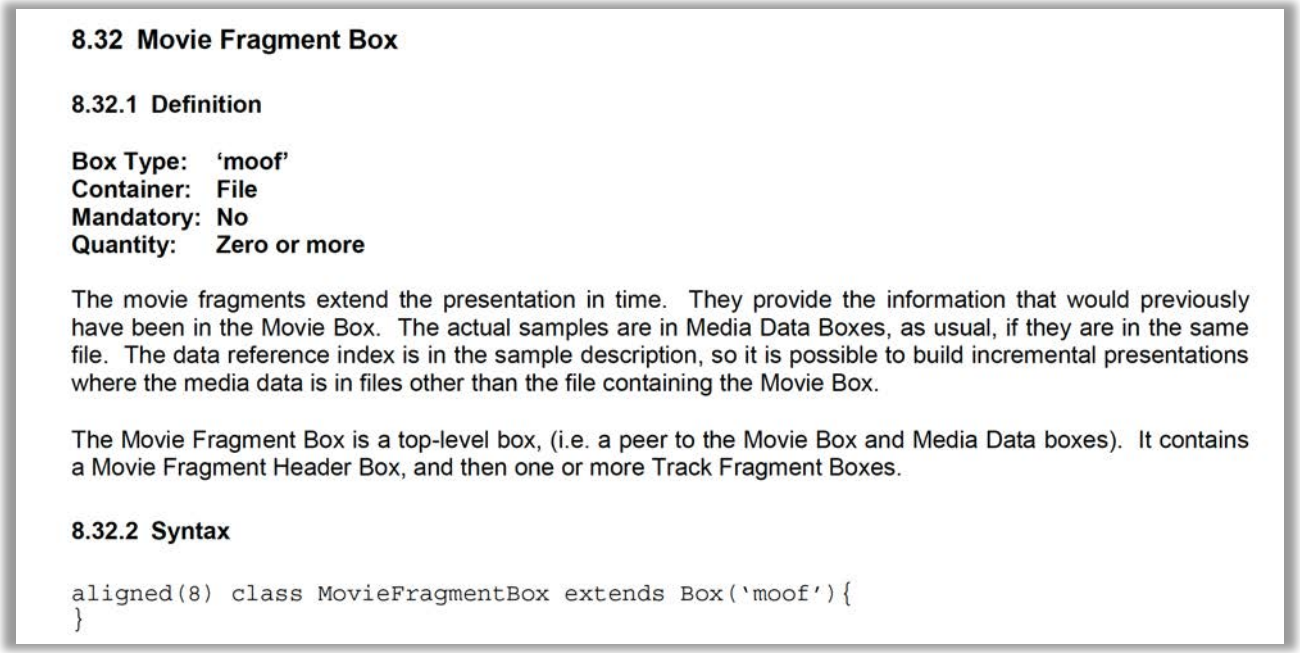
ExoPlayer supports DASH with multiple container formats. Media streams must be demuxed, meaning that video, audio and text must be defined in distinct AdaptationSet elements in the DASH manifest (CEA-608 is an exception as described in the table below). The contained audio and video sample formats must also be supported (see the sample formats section for details).

Feature	Supported	Comments
Containers		
FMP4	YES	Demuxed streams only
WebM	YES	Demuxed streams only
Matroska	YES	Demuxed streams only
MPEG-TS	NO	No support planned
Closed captions/subtitles		
TTML	YES	Raw, or embedded in FMP4 according to ISO/IEC 14496-30
WebVTT	YES	Raw, or embedded in FMP4 according to ISO/IEC 14496-30
CEA-608	YES	Carried in SEI messages embedded in FMP4 video streams
Metadata		
EMSG metadata	YES	Embedded in FMP4
Content protection		
Widevine	YES	"cenc" scheme: API 19+; "cbcs" scheme: API 25+
PlayReady SL2000	YES	Android TV, "cenc" scheme only
ClearKey	YES	API 21+, "cenc" scheme only
Live playback		
Regular live playback	YES	
Ultra low-latency CMAF live playback	YES	

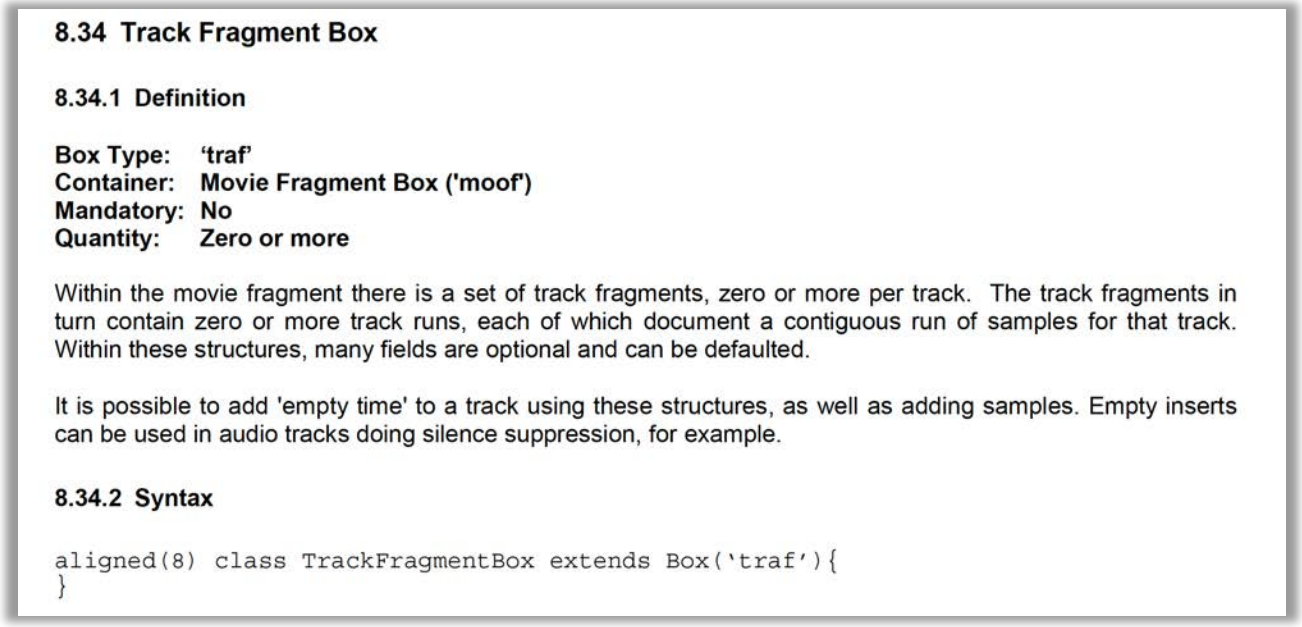
Source: https://exoplayer.dev/dash.html.

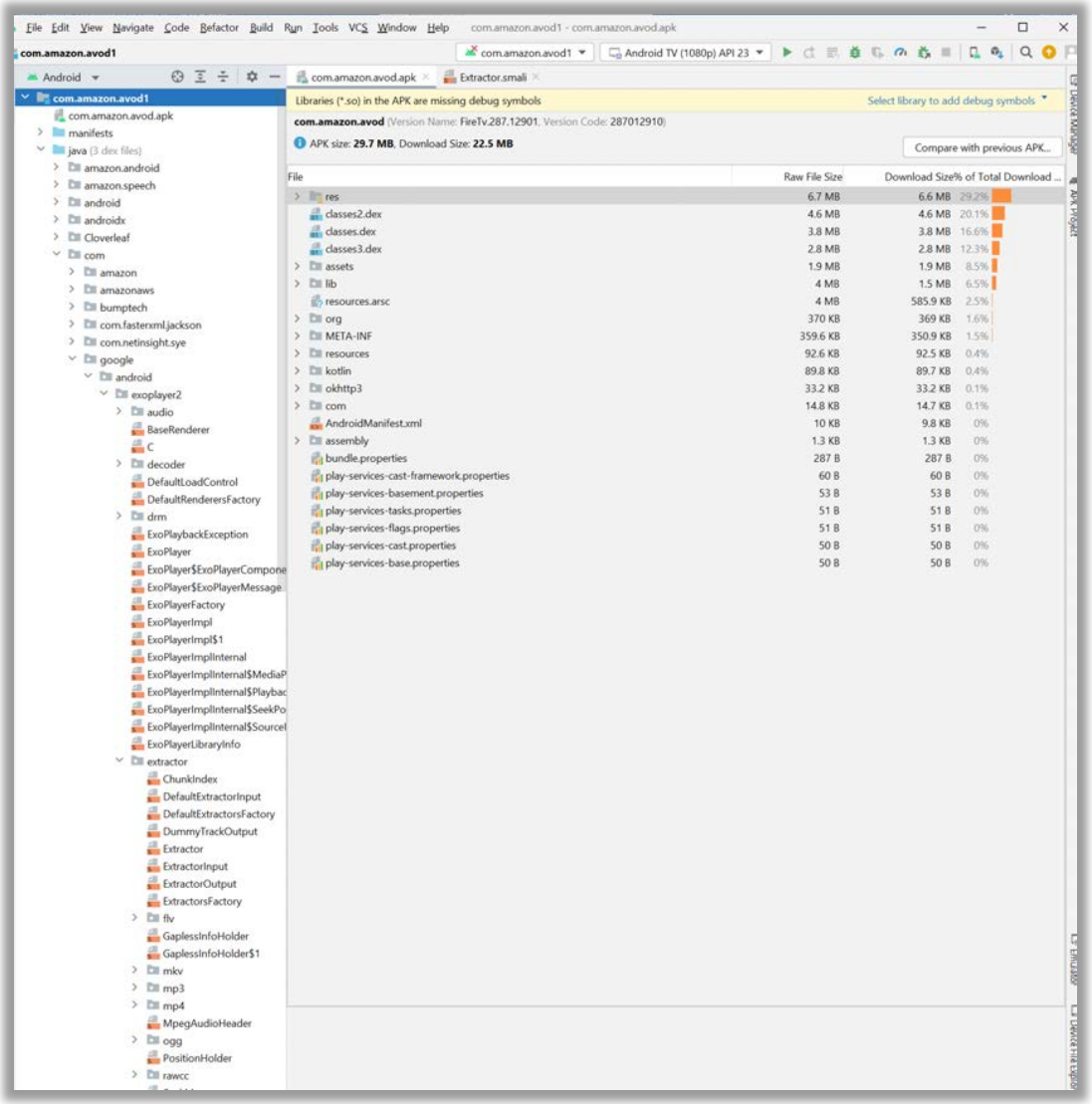
553 Patent	Amazon's Fire TV Stick (3rd Gen)
	As shown below, DASH supports ISO Base Media File Format as specified in ISO/IEC 14496-12. For example, the DASH standard states that "[a]ll Segment formats defined in subclause 6.3 shall contain one or more boxes in accordance with the box structure of the ISO base media file format ISO/IEC 14496-12." Ex. 94 (ISO/IEC 23009-1:2019(E)) at 149. **6.3 Segment formats for ISO base media file format** **6.3.1 General** This subclause defines Segment formats based on the ISO Base Media File Format as specified in ISO/IEC 14496-12. All Segment formats defined in subclause 6.3 shall contain one or more boxes in accordance with the box structure of the ISO base media file format ISO/IEC 14496-12. Refinements on generic concepts are introduced in subclause 6.3.2. Segment formats are defined for Initialization Segments (subclause 6.3.3), Media Segments (subclause 6.3.4) and Self-Initializing Media Segments (subclause 6.3.5). Bitstream Switching Segments and Index Segments are not defined for this media format. Source: Ex. 94 (ISO/IEC 23009-1:2019(E)) at 149. For example, as shown below, ISO Base Media File Format standard ("ISOBMFF") defines the Movie Fragment Box 'moof' and the Track Fragment Box 'traf,' among others.

553 Patent	Amazon's Fire TV Stick (3rd Gen)

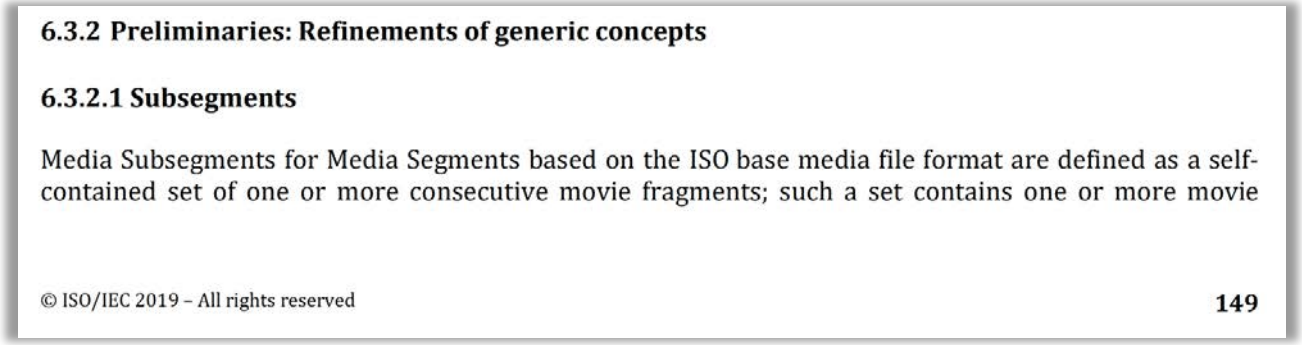
	
	8.32 Movie Fragment Box
	
	8.32.1 Definition
	
	Box Type: 'moof'
	Container: File
	Mandatory: No
	Quantity: Zero or more
	
	The movie fragments extend the presentation in time. They provide the information that would previously have been in the Movie Box. The actual samples are in Media Data Boxes, as usual, if they are in the same file. The data reference index is in the sample description, so it is possible to build incremental presentations where the media data is in files other than the file containing the Movie Box.
	
	The Movie Fragment Box is a top-level box, (i.e. a peer to the Movie Box and Media Data boxes). It contains a Movie Fragment Header Box, and then one or more Track Fragment Boxes.
	
	8.32.2 Syntax
	
	`aligned(8) class MovieFragmentBox extends Box('moof'){`
	`}`
	
	Source: Ex. 95 (IEC 14496-12 *Information technology — Coding of audio-visual objects — Part 12: ISO base media file format* (2nd Ed. 2005) ("ISO/IEC 14496-12:2005(E)") at 39.

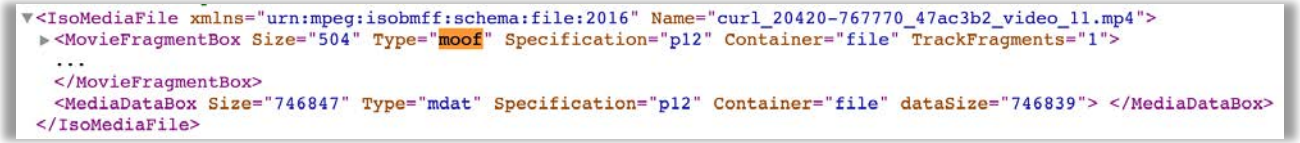
553 Patent	Amazon's Fire TV Stick (3rd Gen)

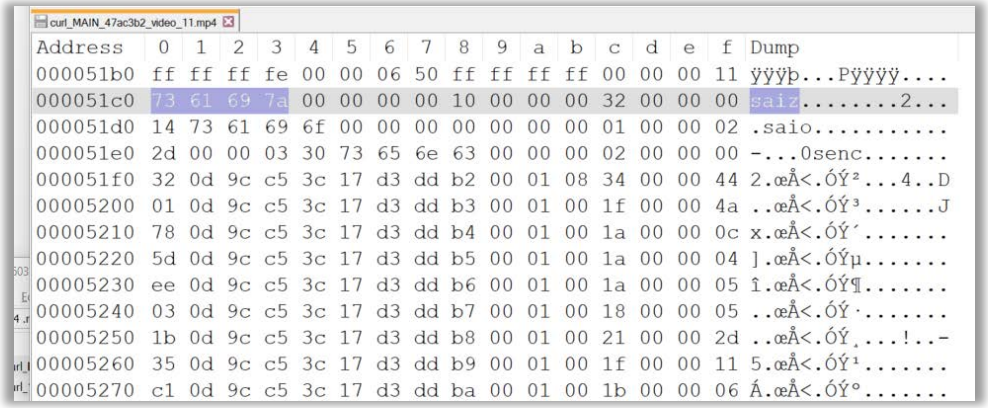
	
	8.34 Track Fragment Box
	
	8.34.1 Definition
	
	Box Type: 'traf'
	Container: Movie Fragment Box ('moof')
	Mandatory: No
	Quantity: Zero or more
	
	Within the movie fragment there is a set of track fragments, zero or more per track. The track fragments in turn contain zero or more track runs, each of which document a contiguous run of samples for that track. Within these structures, many fields are optional and can be defaulted.
	
	It is possible to add 'empty time' to a track using these structures, as well as adding samples. Empty inserts can be used in audio tracks doing silence suppression, for example.
	
	8.34.2 Syntax
	
	`aligned(8) class TrackFragmentBox extends Box('traf'){`
	`}`
	
	Source: Ex. 95 (ISO/IEC 14496-12:2005(E)) at 40.
	
	Thus, ExoPlayer supports the use of box structures when receiving partially encrypted video streams.
	
	The Prime Video application installed on the Fire Stick parses data based on the MP4 atom type.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	 Source: Screenshot of com.amazon.avod.apk in Android Studio showing ExoPlayer references and Extractor, taken by, or on behalf of, DivX.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	Thus, the Fire Stick includes a parser, and the Fire Stick obtains the "at least a portion of a container file" at the parser.  Source: Screenshot of *Wildest Places, Wild Coasts (Episode 4)* played back using the Prime Video application installed on the Fire Stick, taken by, or on behalf of, DivX. Therefore, the Fire Stick includes a playback application (*e.g.*, Prime Video application, including the ExoPlayer playback component) to cause the set of one or more processors to perform the steps of "obtaining, at a parser, at least a portion of a container file."

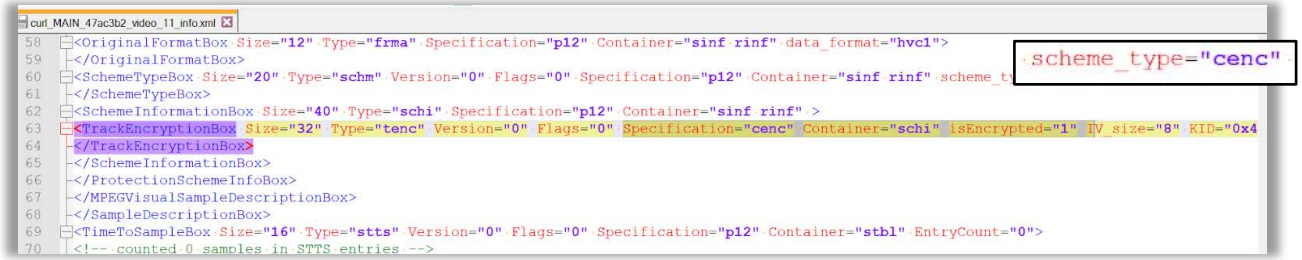
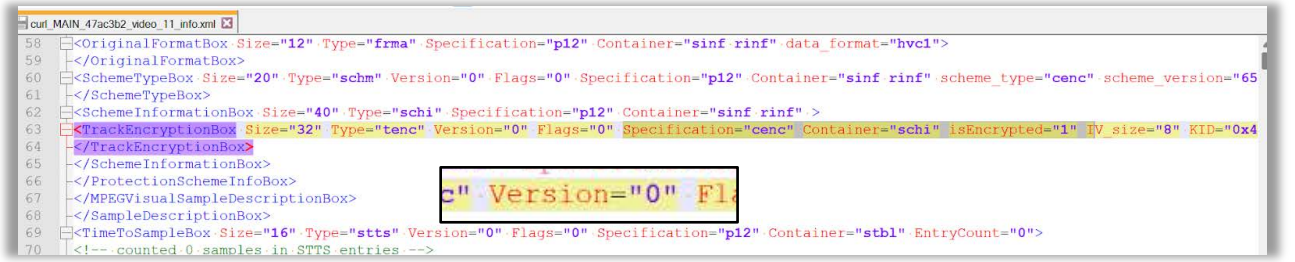
553 Patent	Amazon's Fire TV Stick (3rd Gen)
[11(b)(i)(A)] video data comprising a partially encrypted frame;	An Amazon Fire TV Stick (3rd Gen), Build Model AFTSSS ("Fire Stick"), as exemplary of all Amazon Accused Products with Amazon Fire OS, comprises a playback application to cause the set of one or more processors to perform the steps of "obtaining, at a processor, at least a portion of a container file." For example, the Fire Stick includes a playback application such as the Prime Video application. *See* **Element [11(b)]**, *supra*. The Prime Video application causes the Fire Stick to obtain a portion of a container file—such as a mp4 container file. *See* **Element [11(b)(i)]**, *supra*. "The at least a portion of the container file" (*e.g.*, the at least a portion of the mp4 container file obtained by the Fire Stick via the Prime Video application) "comprises video data comprising a partially encrypted frame." The obtained Prime Video mp4 container files include video data. The MPEG-DASH standard specifies that "Media Subsegments for Media Segments based on the ISO base media file format are defined as a self-contained set of one or more consecutive movie fragments; such a set contains one or more movie fragment boxes with the corresponding media data ('mdat') box(es). A media data box containing data referenced by a movie fragment ('moof') box shall follow that movie fragment box and precede the next movie fragment box, if any, containing information about the same track." Ex. 94 (ISO/IEC 23009-1:2019(E)) at 149-50.  **6.3.2 Preliminaries: Refinements of generic concepts** **6.3.2.1 Subsegments** Media Subsegments for Media Segments based on the ISO base media file format are defined as a self-contained set of one or more consecutive movie fragments; such a set contains one or more movie © ISO/IEC 2019 – All rights reserved 149 Source: Ex. 94 (ISO/IEC 23009-1:2019(E)) at 149.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	fragment boxes with the corresponding media data (`'mdat'`) box(es). A media data box containing data referenced by a movie fragment (`'moof'`) box shall follow that movie fragment box and precede the next movie fragment box, if any, containing information about the same track. For a Media Subsegment, the value of the `reference_type` field in the describing Segment Index (`'sidx'`) box shall be set to 0. Source: Ex. 94 (ISO/IEC 23009-1:2019(E)) at 150. As shown below, for the downloaded byte ranges 20420 to 767770, the obtained Prime Video mp4 container files contain at least one 'moof' (*e.g.*, <MovieFragmentBox…>) and 'mdat' (*e.g.*, <MediaDataBox…>) box pairing.  Source: Excerpt of mp4box output for *Wildest Places, Wild Coasts (Episode 4)*, streamed using the Prime Video application installed on the Fire Stick, taken by, or on behalf of, DivX. The obtained Prime Video mp4 container files include video data, including 'saio' (*e.g.*, <SampleAuxiliaryInfoOffsetBox…>) and 'saiz' boxes (*e.g.*, <SampleAuxiliaryInfoSizeBox…>) in the 'traf' box. — Encryption information for individual media samples – this includes Initialization Vectors and Subsample encryption data. This data is sample auxiliary information, referenced by using a Sample Auxiliary Information Sizes Box ('saiz') and a Sample Auxiliary Information Offsets Box ('saio'). See Clause 7 for further details. Source: Ex. 92 (ISO/IEC 23001-7:2016(E)) at 3.

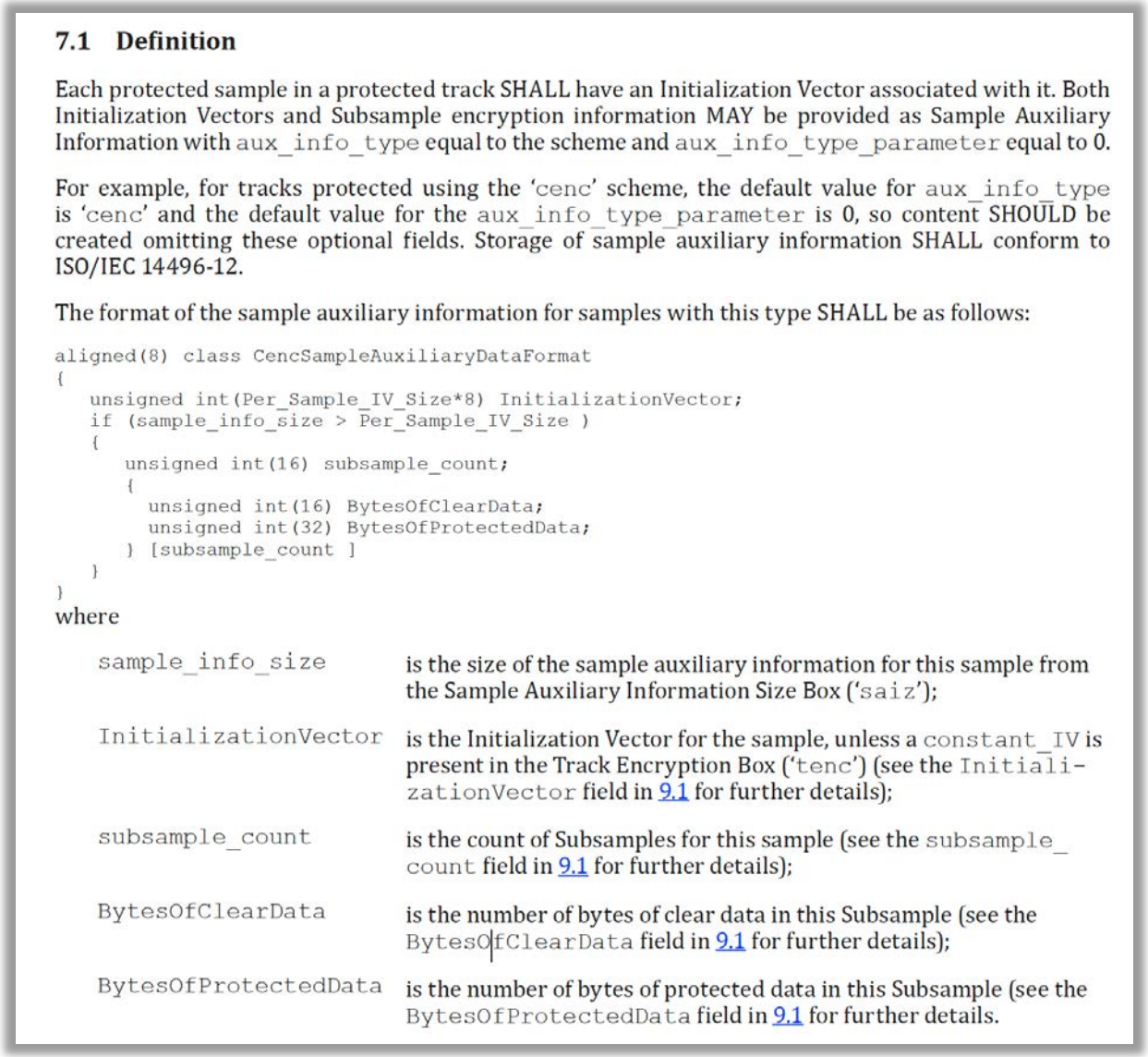
553 Patent	Amazon's Fire TV Stick (3rd Gen)
	 Source: Excerpt of mp4box output for *Wildest Places, Wild Coasts (Episode 4)*, streamed using the Prime Video application installed on the Fire Stick, taken by, or on behalf of, DivX. Direct evidence of video data includes the raw video data of the mp4 container file with the text 'saiz' and 'senc' shown via a hex editor:  Source: Hexadecimal ("hex") view of binary file for *Wildest Places, Wild Coasts (Episode 4)*, streamed using the Prime Video application installed on the Fire Stick, taken by, or on behalf of, DivX.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	The video data (*e.g.*, the obtained Prime Video video data) comprises "a partially encrypted frame." The ISO Common Encryption Specification provides a standard encryption scheme that supports multiple DRM systems: **Introduction** Common Encryption specifies standard encryption and key mapping methods that can be utilized to enable decryption of the same file using different Digital Rights Management (DRM) and key management systems. It operates by defining encryption algorithms and encryption-related metadata necessary to decrypt the protected streams, yet it leaves the details of rights mappings, key acquisition and storage, DRM content protection compliance rules, etc., up to the DRM system or systems. For instance, DRM systems is intended to support identifying the decryption key via stored key identifiers (KIDs), but how each DRM system protects and locates the KID identified decryption key is left to a DRM-specific method. Source: Ex. 92 (ISO/IEC 23001-7:2016(E)) at vi. The ISO Common Encryption Specification defines the 'cenc' encryption scheme (AES-CTR mode full sample and video NAL subsample encryption) (*i.e.*, partial frame encryption). Amazon Prime Video uses that scheme to encrypt video content. **4.2 Common encryption scheme types** Four protection schemes are specified in this edition of Common Encryption. Each scheme uses syntax and algorithms specified in Clause 5 to Clause 9, as constrained in Clause 10. They are the following: a) 'cenc' – AES-CTR mode full sample and video NAL Subsample encryption, see 10.1; b) 'cbc1' – AES-CBC mode full sample and video NAL Subsample encryption, see 10.2; c) 'cens' – AES-CTR mode partial video NAL pattern encryption, see 10.3; d) 'cbcs' – AES-CBC mode partial video NAL pattern encryption, see 10.4. Source: Ex. 92 (ISO/IEC 23001-7:2016(E)) at 3.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	For example, as shown below, the ISOBMFF standard defines a "sample" as "an individual frame of video, a time-contiguous series of video frames, or a time-contiguous compressed section of audio." Source: Ex. 95 (ISO/IEC 14496-12:2005(E)) at 2. **3.10** **Sample**: In non-hint tracks, a sample is an individual frame of video, a time-contiguous series of video frames, or a time-contiguous compressed section of audio. In hint tracks, a sample defines the formation of one or more streaming packets. No two samples within a track may share the same time-stamp. Source: Ex. 95 (ISO/IEC 14496-12:2005(E)) at 2. For the 'cenc' AES-CTR encryption scheme, as shown below, the ISO Common Encryption Specification requires the scheme_type field to be set to the four-character code 'cenc' and the version of the TrackEncryptionBox ('tenc') be set to 0: **10.1 'cenc' AES-CTR scheme** Support for the 'cenc' scheme is mandatory. This scheme uses counter-mode protection as specified in 9.3. The scheme_type field of the scheme Type Box (`schm') SHALL be set to the four character code 'cenc'. Encrypted video tracks using NAL unit structured video conforming to ISO/IEC 14496-15 SHALL be protected using Subsample encryption specified in 9.5 and SHALL NOT use pattern encryption. As a result, the fields crypt_byte_block and skip_byte_block SHALL be 0. Non-video encrypted tracks SHALL be protected using full-sample encryption as specified in 9.4. The version of the TrackEncryptionBox ('tenc') SHALL be 0. Constant IVs SHALL NOT be used; Per_Sample_IV_Size SHALL NOT be 0, except for unencrypted sample groups. For an isProtected flag of 0x1 where the scheme_type field of the scheme type box is 'cenc' (i.e. AES-CTR), counter values SHALL be unique per KID. Source: Ex. 92 (ISO/IEC 23001-7:2016(E)) at 19.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	As shown below, the SchemeTypeBox information obtained using mp4box from streamed Amazon Prime Video video content, *e.g.*, from *Wildest Places, Wild Coasts (Episode 4)*, indicates use of subsample encryption. Here, the scheme_type is set to 'cenc'.  Source: Screenshot of SchemeTypeBox information using mp4box from *Wildest Places, Wild Coasts (Episode 4)*, streamed using the Prime Video application installed on the Fire Stick, taken by, or on behalf of, DivX. As shown below, the TrackEncryptionBox information obtained using mp4box from streamed Amazon Prime Video video content, *e.g.*, from *Wildest Places, Wild Coasts (Episode 4)*, also indicates use of subsample encryption. Here, the version of the TrackEncryptionBox is set to '0'.  Source: Screenshot of TrackEncryptionBox information using mp4box from *Wildest Places, Wild Coasts (Episode 4)*, streamed using the Prime Video application installed on the Fire Stick, taken by, or on behalf of, DivX. Further, the presence of SampleAuxiliaryInfoSizeBoxes ('saiz') and SampleEncryptionBoxes ('senc') indicates that the mp4 container files are encrypted according to the ISO Common Encryption Standard. Ex. 92 (ISO/IEC 23001-7:2016(E)) at 5. Specifically, the presence of

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	BytesOfClearData and BytesOfProtectedData in the SampleEncryptionBoxes indicates the use of partial frame encryption. *Id.* at 6. "The Sample Encryption Box contains sample auxiliary information and may contain a per sample Initialization Vector for each sample, and clear and protected byte ranges of partially protected video samples ('Subsample encryption')." *Id.* BytesOfClearData "is the number of bytes of clear data in this Subsample," and BytesOfProtectedData "is the number of bytes of protected data in this Subsample." *Id.*

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	

7.1 Definition

Each protected sample in a protected track SHALL have an Initialization Vector associated with it. Both Initialization Vectors and Subsample encryption information MAY be provided as Sample Auxiliary Information with `aux_info_type` equal to the scheme and `aux_info_type_parameter` equal to 0.

For example, for tracks protected using the 'cenc' scheme, the default value for `aux_info_type` is 'cenc' and the default value for the `aux_info_type_parameter` is 0, so content SHOULD be created omitting these optional fields. Storage of sample auxiliary information SHALL conform to ISO/IEC 14496-12.

The format of the sample auxiliary information for samples with this type SHALL be as follows:

```
aligned(8) class CencSampleAuxiliaryDataFormat
{
   unsigned int(Per_Sample_IV_Size*8) InitializationVector;
   if (sample_info_size > Per_Sample_IV_Size )
   {
      unsigned int(16) subsample_count;
      {
        unsigned int(16) BytesOfClearData;
        unsigned int(32) BytesOfProtectedData;
      } [subsample_count ]
   }
}
```

where

`sample_info_size`	is the size of the sample auxiliary information for this sample from the Sample Auxiliary Information Size Box ('saiz');
`InitializationVector`	is the Initialization Vector for the sample, unless a `constant_IV` is present in the Track Encryption Box ('tenc') (see the `InitializationVector` field in 9.1 for further details);
`subsample_count`	is the count of Subsamples for this sample (see the `subsample_count` field in 9.1 for further details);
`BytesOfClearData`	is the number of bytes of clear data in this Subsample (see the `BytesOfClearData` field in 9.1 for further details);
`BytesOfProtectedData`	is the number of bytes of protected data in this Subsample (see the `BytesOfProtectedData` field in 9.1 for further details.

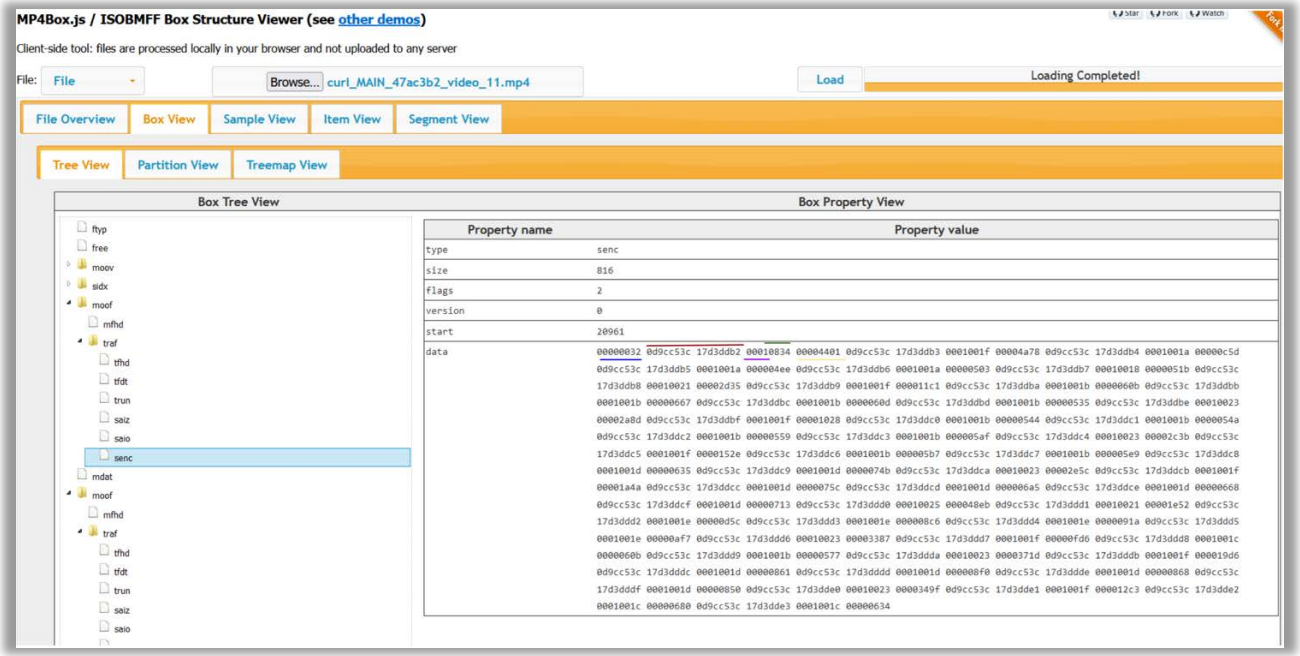
Source: Ex. 92 (ISO/IEC 23001-7:2016(E)) at 5.

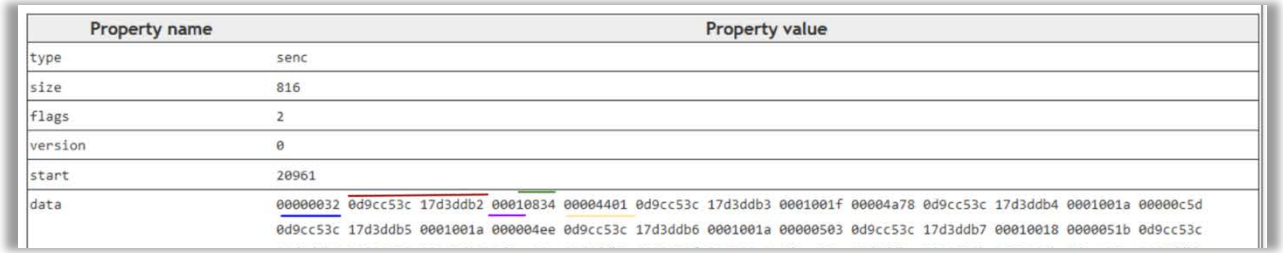
553 Patent	Amazon's Fire TV Stick (3rd Gen)

	7.2 Sample Encryption Information box for storage of sample auxiliary information
	
	7.2.1 Sample Encryption Box ('senc')
	
	Box Type: 'senc'
	
	Container: Track Fragment Box ('traf') or Track Box ('trak')
	
	Mandatory: No
	
	Quantity: Zero or one
	
	An optional storage location for Sample Auxiliary Information is the Sample Encryption Box ('senc'), specified here.
	
	The Sample Encryption Box contains sample auxiliary information and may contain a per sample Initialization Vector for each sample, and clear and protected byte ranges of partially protected video samples ("Subsample encryption"). It MAY be used when samples in a track or track fragment are protected. Storage of 'senc' in a Track Fragment Box makes the necessary Sample Auxiliary Information accessible within the movie fragment for all contained samples in order to make each track fragment independently decryptable; for instance, when movie fragments are delivered as DASH Media Segments.
	
	7.2.2 Syntax
	
	`aligned(8) class SampleEncryptionBox`
	`  extends FullBox('senc', version=0, flags)`
	`{`
	`  unsigned int(32) sample_count;`
	`  {`
	`  unsigned int(Per_Sample_IV_Size*8) InitializationVector;`
	`  if (flags & 0x000002)`
	`  {`
	`  unsigned int(16) subsample_count;`
	`  {`
	`  unsigned int(16) BytesOfClearData;`
	`  unsigned int(32) BytesOfProtectedData;`
	`  } [ subsample_count ]`
	`  }`
	`  }[ sample_count ]`
	`}`
	
	Source: Ex. 92 (ISO/IEC 23001-7:2016(E)) at 6.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	Specifically, as shown below within a SampleEncryptionBox ('senc'), each "partially encrypted frame" (*e.g.*, sample) corresponds to a SampleEncryptionEntry. Each SampleEncryptionEntry includes "NumClearBytes" and "NumEncryptedBytes" values in a SubSampleEncryptionEntry box to describe the unencrypted ("clear") and encrypted data, respectively, in the frame. As shown below, the obtained Prime Video mp4 container file contains a Sample Encryption Box 'senc' (*e.g.*, <SampleEncryptionBox…>). In this example, the Sample Encryption Box contains cryptographic information about frames (*e.g.*, samples) (sampleCount="50"). In the first Sample Encryption Entry, the IV_size="8" and IV= "0x0D9CC53C17D3DDB2." Additionally, the first Sample Encryption Entry indicates that the runs of bytes that are unencrypted ("clear") (*e.g.*, NumClearBytes="2100"), and encrypted ("protected") (*e.g.*, the NumEncryptedBytes ="17409") are shown below:  Source: Excerpt of mp4box output for *Wildest Places, Wild Coasts (Episode 4)*, streamed using the Prime Video application installed on the Fire Stick, taken by, or on behalf of, DivX.

553 Patent	Amazon's Fire TV Stick (3rd Gen)





Source: Screenshots of the same video data as the above excerpted mp4box output as viewed using MP4Box.js for *Wildest Places, Wild Coasts (Episode 4)*, streamed using the Prime Video application installed on the Fire Stick, taken by, or on behalf, of DivX

The same information is displayed in both the MP4Box.js view and excerpted mp4box output, consistent with 'cenc' AES-CTR mode full sample and video NAL subsample encryption:

553 Patent	Amazon's Fire TV Stick (3rd Gen)

	• Blue (underlined): `sample_count` (number of frames in this segment) = Hex 00000032 = Dec 50.
	• Red (overlined): IV 64-bit `IV`
	• Purple (underlined): 16-bit `subsample_count` = 1
	• Green (overlined): 16-bit `BytesOfClearData` = Hex 834 = Dec 2100
	• Yellow (underlined): 32-bit `BytesOfProtectedData` = Hex 4401= Dec 17409
	
	Thus, consistent with the mp4box output, "video data" (*e.g.*, the obtained Prime Video mp4 container file) "compris[es] a partially encrypted frame." The above information overlined in green demonstrates that the information represented by BytesOfClearData indicates that a run of 2100 bytes of data is unencrypted or "clear" and the next 17409 bytes of data in the first sample are encrypted, respectively, in the 'mdat' box that corresponds to the first frame (*e.g.*, sample).
	
	7.2.2 Syntax
	
	`aligned(8) class SampleEncryptionBox`
	`  extends FullBox('senc', version=0, flags)`
	`{`
	`  unsigned int(32) sample_count;`
	`  {`
	`  unsigned int(Per_Sample_IV_Size*8) InitializationVector;`
	`  if (flags & 0x000002)`
	`  {`
	`  unsigned int(16) subsample_count;`
	`  {`
	`  unsigned int(16) BytesOfClearData;`
	`  unsigned int(32) BytesOfProtectedData;`
	`  } [ subsample_count ]`
	`  }`
	`  }[ sample_count ]`
	`}`
	
	Source: Ex. 92 (ISO/IEC 23001-7:2016(E)) at 6.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	As shown above, the obtained video data is compliant and in accordance with Section 7.2.2 of the ISO/IEC 23001-7 standard, which identifies the syntax and assignment of individual bits in the file. For example, "[s]torage of 'senc' in a Track Fragment Box makes the necessary Sample Auxiliary Information accessible within the movie fragment for all contained samples in order to make each track fragment independently decryptable; for instance, when movie fragments are delivered as DASH Media Segments." Ex. 92 (ISO/IEC 23001-7:2016(E)) at 6. Thus, Amazon Prime Video uses partial frame encryption (*e.g.*, 'cenc' AES-CTR mode full sample and video NAL subsample encryption) to encrypt video content. The ExoPlayer documentation also specifies that the 'cenc' DRM scheme is the DRM scheme supported for AndroidTV. As previously discussed in **Element [11(b)]**, the 'cenc' DRM scheme is the DRM scheme supported for Fire OS because Amazon's Fire OS is a "fork of Android" and Amazon adapted its port of ExoPlayer so that the Prime Video application and other third-party applications can adopt ExoPlayer in Fire OS and its Amazon Fire playback devices.

553 Patent	Amazon's Fire TV Stick (3rd Gen)

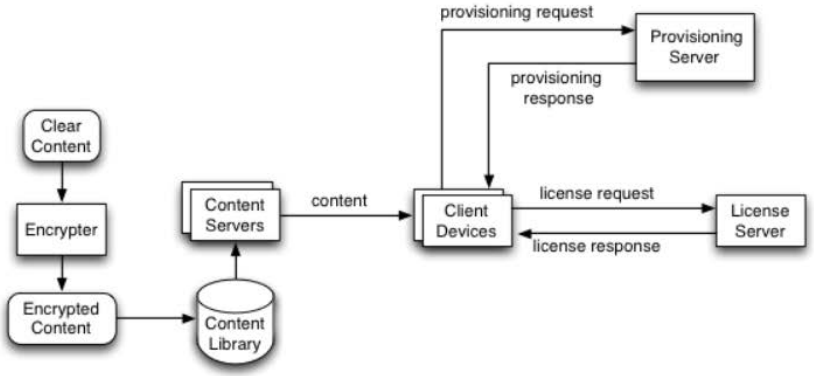
Digital rights management

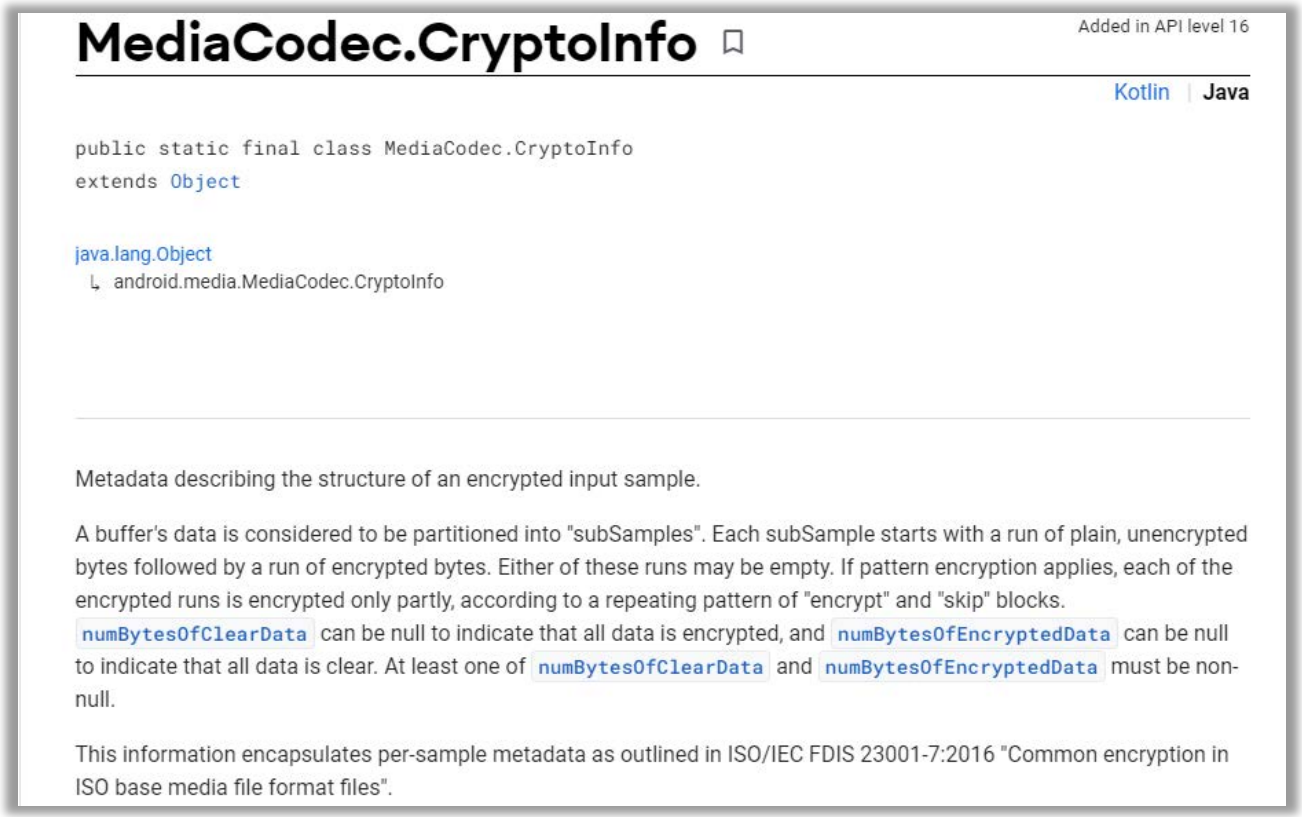
ExoPlayer uses Android's `MediaDrm` API to support DRM protected playbacks. The minimum Android versions required for different supported DRM schemes, along with the streaming formats for which they're supported, are:

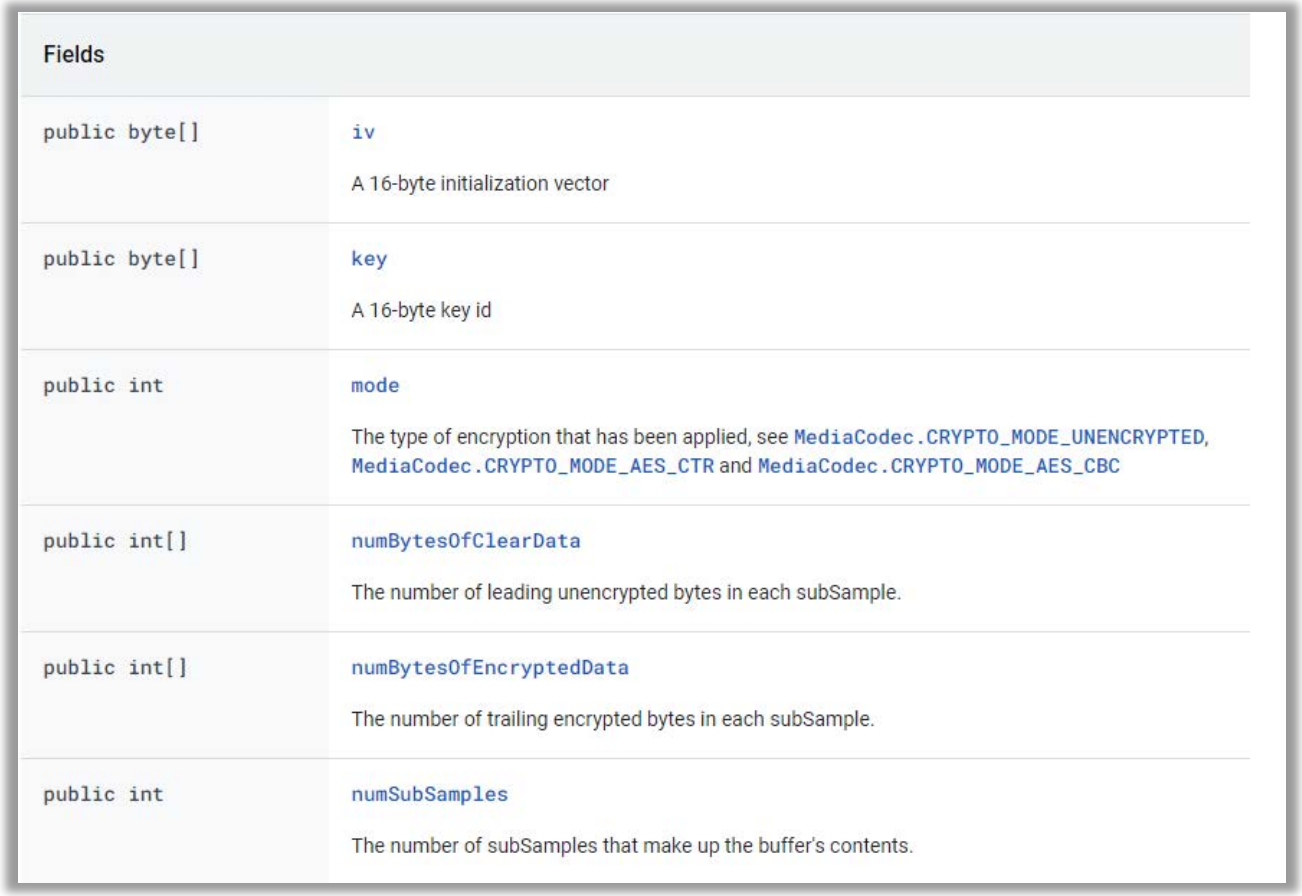
DRM scheme	Android version number	Android API level	Supported formats
Widevine "cenc"	4.4	19	DASH, HLS (FMP4 only)
Widevine "cbcs"	7.1	25	DASH, HLS (FMP4 only)
ClearKey "cenc"	5.0	21	DASH
PlayReady SL2000 "cenc"	AndroidTV	AndroidTV	DASH, SmoothStreaming, HLS (FMP4 only)

Source: https://exoplayer.dev/drm.html.

The 'cenc' DRM scheme is a DRM scheme within the ISO/IEC 23001-7 standard format, a format also supported by the "MediaDrm" API.
https://developer.android.com/reference/android/media/MediaDrm.html.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	MediaDrm can be used to obtain keys for decrypting protected media streams, in conjunction with `MediaCrypto`. The MediaDrm APIs are designed to support the ISO/IEC 23001-7: Common Encryption standard, but may also be used to implement other encryption schemes. Encrypted content is prepared using an encryption server and stored in a content library. The encrypted content is streamed or downloaded from the content library to client devices via content servers. Licenses to view the content are obtained from a License Server.  Keys are requested from the license server using a key request. The key response is delivered to the client app, which provides the response to the MediaDrm API. Source: https://developer.android.com/reference/android/media/MediaDrm.html. The ExoPlayer code base also applies the 'cenc' encryption scheme for partially encrypted frames, where each frame comprises of unencrypted ("clear") and encrypted bytes contained in the "CryptoInfo" class nested within the "MediaCodec" class.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	 MediaCodec.CryptoInfo Added in API level 16 Kotlin | Java public static final class MediaCodec.CryptoInfo extends Object java.lang.Object ↳ android.media.MediaCodec.CryptoInfo Metadata describing the structure of an encrypted input sample. A buffer's data is considered to be partitioned into "subSamples". Each subSample starts with a run of plain, unencrypted bytes followed by a run of encrypted bytes. Either of these runs may be empty. If pattern encryption applies, each of the encrypted runs is encrypted only partly, according to a repeating pattern of "encrypt" and "skip" blocks. numBytesOfClearData can be null to indicate that all data is encrypted, and numBytesOfEncryptedData can be null to indicate that all data is clear. At least one of numBytesOfClearData and numBytesOfEncryptedData must be non-null. This information encapsulates per-sample metadata as outlined in ISO/IEC FDIS 23001-7:2016 "Common encryption in ISO base media file format files". Source: https://developer.android.com/reference/android/media/MediaCodec.CryptoInfo.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	

Fields

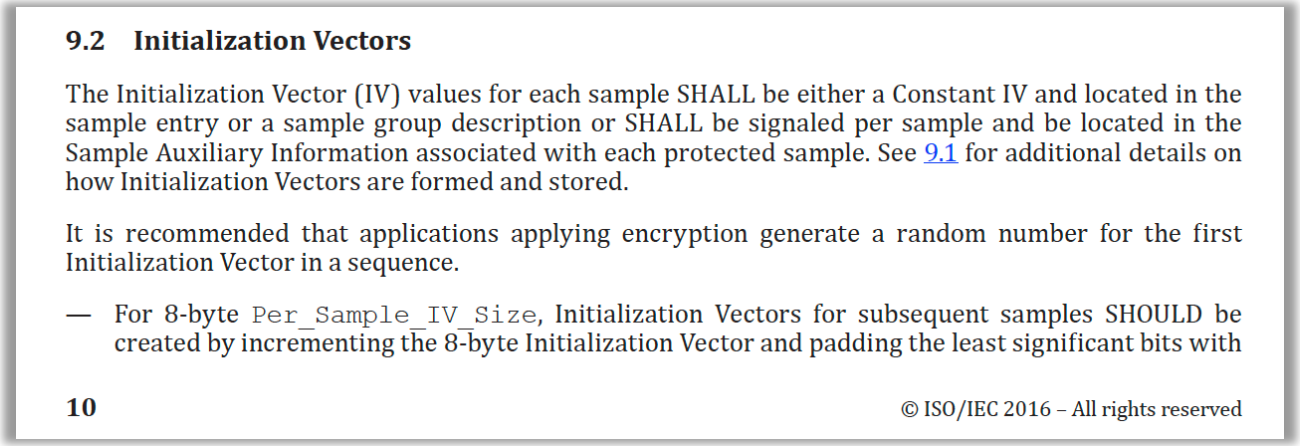
`public byte[]`	`iv` A 16-byte initialization vector
`public byte[]`	`key` A 16-byte key id
`public int`	`mode` The type of encryption that has been applied, see `MediaCodec.CRYPTO_MODE_UNENCRYPTED`, `MediaCodec.CRYPTO_MODE_AES_CTR` and `MediaCodec.CRYPTO_MODE_AES_CBC`
`public int[]`	`numBytesOfClearData` The number of leading unencrypted bytes in each subSample.
`public int[]`	`numBytesOfEncryptedData` The number of trailing encrypted bytes in each subSample.
`public int`	`numSubSamples` The number of subSamples that make up the buffer's contents.

Source: https://developer.android.com/reference/android/media/MediaCodec.CryptoInfo.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	 Source: Screenshot of *Wildest Places, Wild Coasts (Episode 4)* played back using the Prime Video application installed on the Fire Stick, taken by, or on behalf of, DivX. Thus, the Prime Video application causes the Fire Stick to obtain a portion of a container file—such as mp4 container files—with "video data comprising a partially encrypted frame."

553 Patent	Amazon's Fire TV Stick (3rd Gen)
[11(b)(i)(B)] cryptographic material comprising a wrapped frame key; and	An Amazon Fire TV Stick (3rd Gen), Build Model AFTSSS ("Fire Stick"), as exemplary of all Amazon Accused Products with Amazon Fire OS comprises a playback application to cause the set of one or more processors to perform the steps of "obtaining, at a processor, at least a portion of a container file." For example, the Fire Stick includes a playback application such as the Prime Video application. *See* **Element [11(b)]**, *supra*. The Prime Video application causes the Fire Stick to obtain a portion of a container file—such as a mp4 container file. *See* **Element [11(b)(i)]**, *supra*. "The at least a portion of the container file" (*e.g.*, the at least a portion of the mp4 container file obtained by the Fire Stick via the Prime Video application) "comprises . . . cryptographic material comprising a wrapped frame key." Accordingly, the Prime Video application causes the Fire Stick to obtain a portion of a container file—such as an mp4 container file—with "cryptographic material comprising a wrapped frame key." As discussed above in **Element [11(b)(i)(A)]**, the obtained video data comprises "a partially encrypted frame." *See* **Element [11(b)(i)(A)]**, *supra*. The ExoPlayer code base supports playback of video encrypted in accordance with the 'cenc' AES-CTR DRM scheme set forth in the ISO/IEC 23001-7 specification. *Id.* Video streams encrypted with the AES-CTR mode contain a wrapped frame key (*e.g.*, an initialization vector) for each partially encrypted frame. The information contained in 'senc' boxes discussed above in **Element [11(b)(i)(A]** also comprise cryptographic material. The obtained Prime Video mp4 container files include for each partially encrypted frame "a wrapped frame key" (*e.g.*, the initialization vector ("IV")). As shown below, the ISO Common Encryption Specification provides that the "Per_Sample_IV_Size is the size in bytes of the InitializationVector field" and the value of "8 specifies 64-bit Initialization Vectors." Ex. 92 (ISO/IEC 23001-7:2016(E)) at 9.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	9 Encryption of media data **9.1 Field semantics** Within the sample groups and sample auxiliary information used by the common encryption scheme, these fields have the following semantics: — `isProtected` is the identifier of the protection state of the samples in the track or group of samples. This flag takes the following values: — 0x0: Not protected; — 0x1: protected (as signalled by the `scheme_type` field of the scheme type box '`schm`', e.g. for `scheme_type` of '`cenc`', the track default is AES-CTR encrypted using the '`cenc`' scheme); — 0x02 – 0xFF: Reserved. — `Per_Sample_IV_Size` is the size in bytes of the `InitializationVector` field. The following are supported values: — 0 if the isProtected flag is 0x0 (Not Protected) or Constant IVs are in use; — 8 specifies 64-bit Initialization Vectors; — 16 specifies 128-bit Initialization Vectors. Source: Ex. 92 (ISO/IEC 23001-7:2016(E)) at 9. "The Initialization Vector (IV) values for each sample SHALL be either a Constant IV and located in the sample entry or a sample group description or SHALL be signaled per sample and be located in the Sample Auxiliary Information associated with each protected sample." Ex. 92 (ISO/IEC 23001-7:2016(E)) at 10.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	 **9.2 Initialization Vectors** The Initialization Vector (IV) values for each sample SHALL be either a Constant IV and located in the sample entry or a sample group description or SHALL be signaled per sample and be located in the Sample Auxiliary Information associated with each protected sample. See 9.1 for additional details on how Initialization Vectors are formed and stored. It is recommended that applications applying encryption generate a random number for the first Initialization Vector in a sequence. — For 8-byte `Per_Sample_IV_Size`, Initialization Vectors for subsequent samples SHOULD be created by incrementing the 8-byte Initialization Vector and padding the least significant bits with **10** © ISO/IEC 2016 – All rights reserved Source: Ex. 92 (ISO/IEC 23001-7:2016(E)) at 10. "Storing a unique IV per sample for both CTR and CBC mode increases cryptographic entropy and provides random access and error recovery for each sample (against continuous chaining of multiple samples). CTR mode requires a unique counter value for each cipher block sharing a key." Ex. 92 (ISO/IEC 23001-7:2016(E)) at 11.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	ISO/IEC 23001-7:2016(E) zero to construct a 16-byte number. Using a random starting value for a track introduces entropy into the Initialization Vector values and incrementing the most significant 8 bytes for each sample in sequence ensures that each 16-byte IV and CTR counter value combination is unique. The 8-byte Initialization Vector can roll over from the maximum value (0xFFFFFFFFFFFFFFFF) to the minimum value (0x0) if the maximum 8-byte value is exceeded when incrementing the 8-byte value per sample from its random starting value. 8-byte IVs are recommended for per sample IVs to reduce storage size and guarantee unique counter values for CTR mode counter blocks. — For 16-byte Per_Sample_IV_Size, Initialization Vectors for subsequent samples using CTR mode MAY be created by adding the cipher block count of the previous sample to the Initialization Vector of the previous sample. Using a random starting value introduces entropy into the Initialization Vector values and incrementing by cipher block count for each sample ensures that each CTR counter block is unique. Even though the least significant bytes of the IV (bytes 8 to 15) are incremented as an unsigned 64-bit counter in CTR mode, the Initialization Vector SHOULD be treated as a 128-bit number when calculating the next Initialization Vector from the previous 16-byte IV. CBC mode Initialization Vectors need not be unique per sample or Subsample and may be generated randomly or sequentially, e.g. a per sample IV may be equal to the cipher text of the last encrypted cipher block (a continuous cipher block chain across samples), or generated by incrementing the previous IV by the number of cipher blocks in the last sample or by a fixed amount. Each 'cbc1' IV is stored in sample auxiliary information, so its method of derivation is irrelevant for decryption. Storing a unique IV per sample for both CTR and CBC mode increases cryptographic entropy and provides random access and error recovery for each sample (against continuous chaining of multiple samples). CTR mode requires a unique counter value for each cipher block sharing a key. Some schemes use a Constant IV either as a track default or for a group of samples mapped to a sample group. It is assumed that compressed video data is random enough to allow the reuse of the same IV for each Subsample when a Constant IV is reused on multiple Subsamples and samples. A Constant IV reduces IV data size against a per-sample IV, which requires 8 bytes or 16 bytes per sample. For a fragmented file, a Constant IV typically requires one sample group box per track fragment and a sample group description box containing the sample group's IV. Source: Ex. 92 (ISO/IEC 23001-7:2016(E)) at 11.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	When operating in AES-CTR mode, "[t]he CTR mode counter block SHALL be constructed from a per sample IV and incremented as described below and in 9.2. When an 8 byte Per_Sample_IV_Size is indicated, the least significant 8 bytes of the 16 byte IV (bytes 8 to 15) SHALL be set to zero and used as a 64 bit block counter that is incremented by one for each subsequent 16 byte cipher block of encrypted sample data." Ex. 92 (ISO/IEC 23001-7:2016(E)) at 11. **9.3 AES-CTR mode counter operation** Counter-mode schemes SHALL use the Advanced Encryption Standard, specified in Federal Information Processing Standards Publication 197, FIPS-197 published by the United States National Institute of Standards and Technology (NIST) using 128-bit keys in Counter Mode (AES-CTR), as specified in Recommendation of Block Cipher Modes of Operation, NIST, NIST Special Publication 800-38A. AES-128 CTR mode is a 16 byte block cipher that can encrypt an arbitrary sized byte stream without need for padding or leaving a clear remainder when the last Block of sample data is a partial Block (1 to 15 bytes in size). Counter mode (CTR) operates by encrypting a counter block using the AES block encryption algorithm using the key indicated by KID, and then XOR-ing the result with the data to be encrypted or decrypted. The CTR mode counter block SHALL be constructed from a per sample IV and incremented as described below and in 9.2. When an 8 byte `Per_Sample_IV_Size` is indicated, the least significant 8 bytes of the 16 byte IV (bytes 8 to 15) SHALL be set to zero and used as a 64 bit block counter that is incremented by one for each subsequent 16 byte cipher block of encrypted sample data. When a 16 byte `Per_Sample_IV_Size` is indicated and the least significant 8 bytes (64 bit counter) reaches the maximum value (0xFFFFFFFFFFFFFFFF), then incrementing it SHALL reset the 8 byte block counter to zero (bytes 8 to 15) without affecting the other 8 bytes of the counter (bytes 0 to 7). Within each sample, encrypted data SHALL be a logically continuous byte sequence of 16 byte Blocks, regardless of physically interleaved clear data identified by Subsample encryption or pattern Source: Ex. 92 (ISO/IEC 23001-7:2016(E)) at 11.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	As shown below, the IV (*e.g.*, "IV") is capable of being unwrapped via AES-CTR mode by being inputted through the input block of the decryption process below:

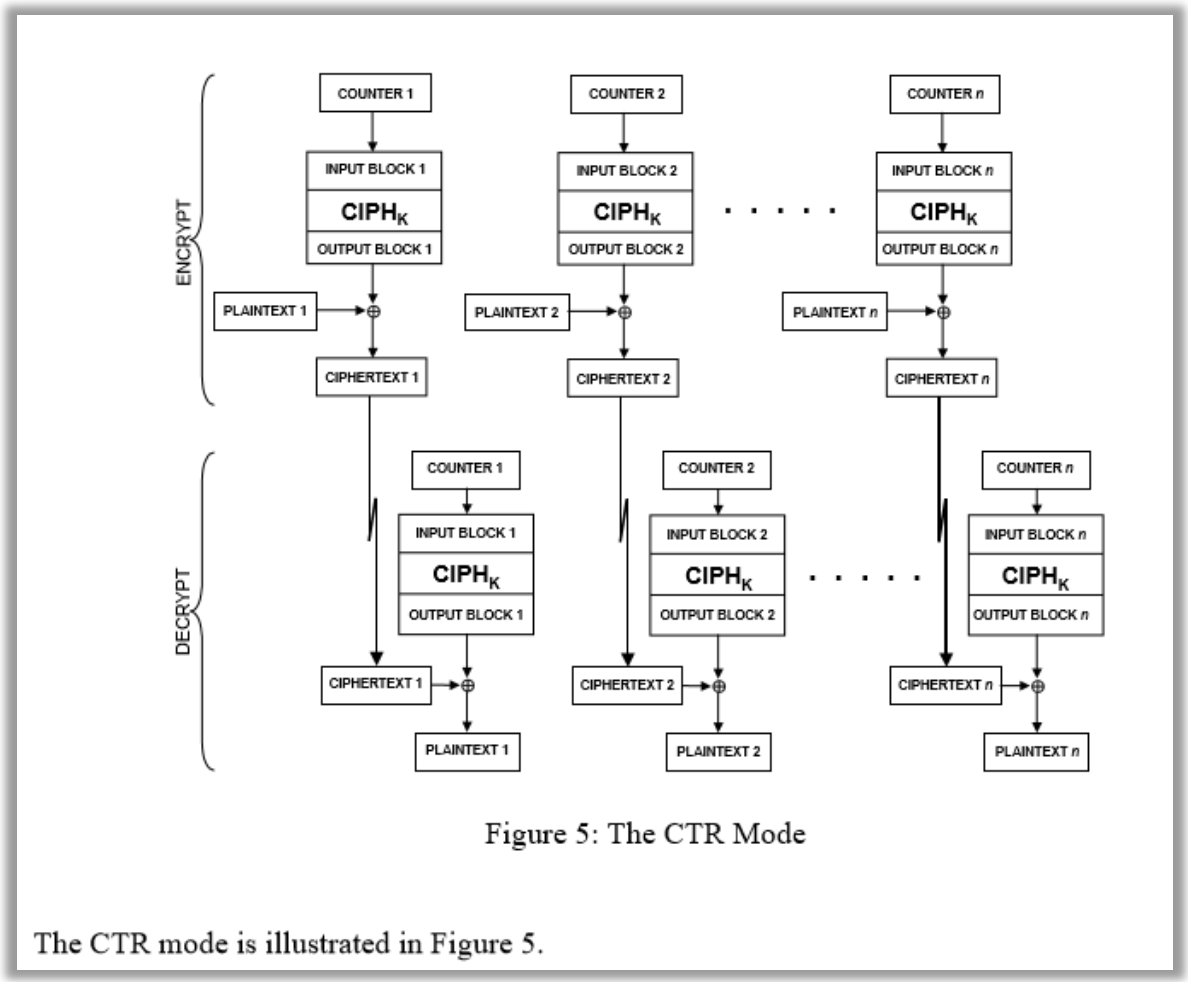


Figure 5: The CTR Mode

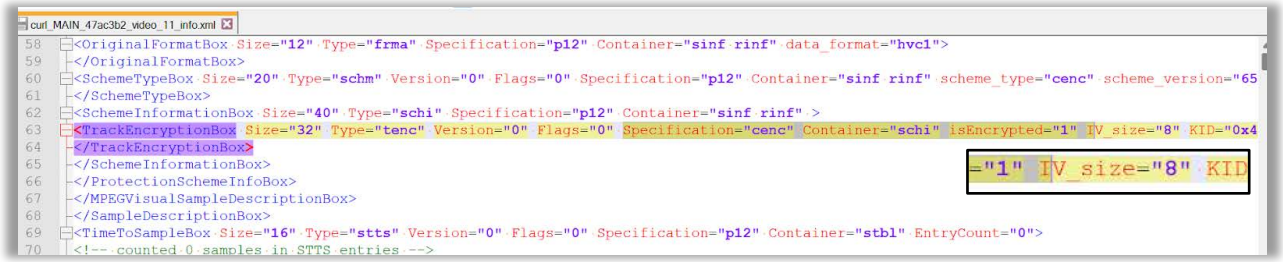
The CTR mode is illustrated in Figure 5.

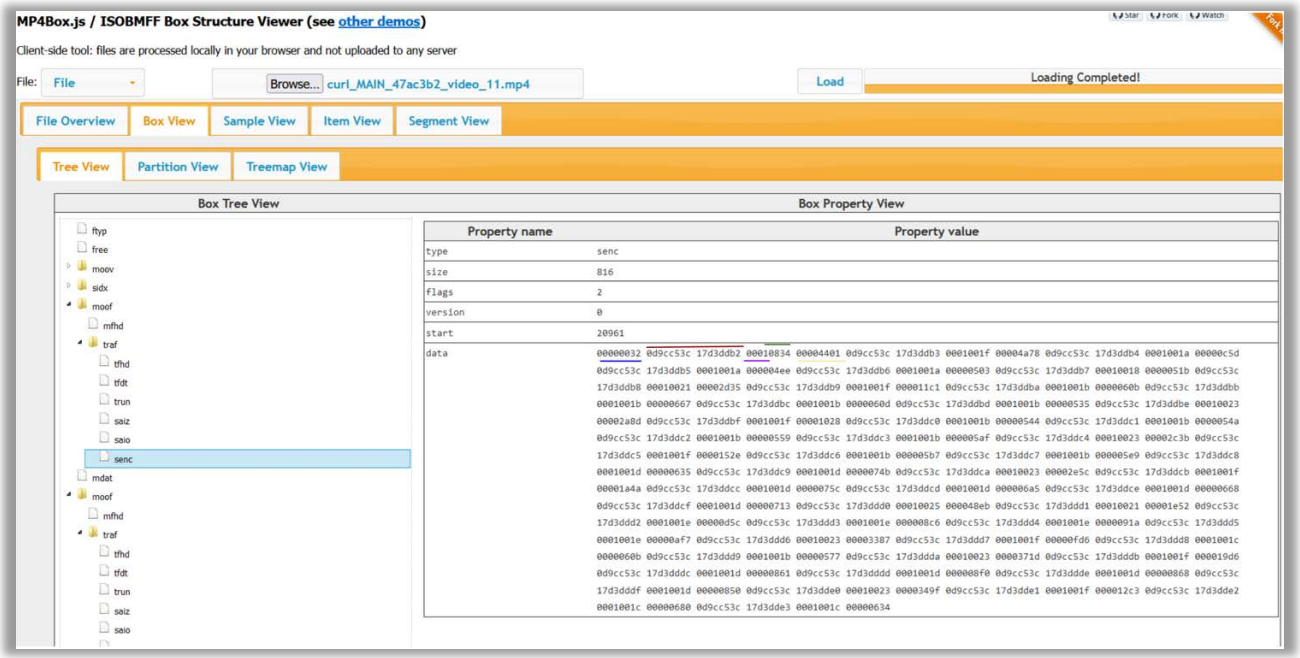
Source: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	The AES-CTR mode's output block for a particular video frame—the unwrapped frame key—is associated with at least one encrypted block within the video frame and is used by the ExoPlayer code base to access the encrypted data. AES-CTR has many properties that make it an attractive encryption algorithm for in high-speed networking. AES-CTR uses the AES block cipher to create a stream cipher. Data is encrypted and decrypted by XORing with the key stream produced by AES encrypting sequential counter block values. AES-CTR is easy to implement, and AES-CTR can be pipelined and parallelized. AES-CTR also supports key stream precomputation. Source: Internet Engineering Task Force in RFC3686, section 2.1 Counter Mode, *available at* https://www.ietf.org/rfc/rfc3686.txt.pdf. As discussed above in **Element [11(b)(i)(A)]**, "[t]he Sample Encryption Box contains sample auxiliary information and may contain a per sample Initialization Vector for each sample, and clear and protected byte ranges of partially protected video samples ('Subsample encryption')." Ex. 92 (ISO/IEC 23001-7:2016(E)) at 6.

553 Patent	Amazon's Fire TV Stick (3rd Gen)

	7.2 Sample Encryption Information box for storage of sample auxiliary information
	
	7.2.1 Sample Encryption Box ('senc')
	
	Box Type: 'senc'
	
	Container: Track Fragment Box ('traf') or Track Box ('trak')
	
	Mandatory: No
	
	Quantity: Zero or one
	
	An optional storage location for Sample Auxiliary Information is the Sample Encryption Box ('senc'), specified here.
	
	The Sample Encryption Box contains sample auxiliary information and may contain a per sample Initialization Vector for each sample, and clear and protected byte ranges of partially protected video samples ("Subsample encryption"). It MAY be used when samples in a track or track fragment are protected. Storage of 'senc' in a Track Fragment Box makes the necessary Sample Auxiliary Information accessible within the movie fragment for all contained samples in order to make each track fragment independently decryptable; for instance, when movie fragments are delivered as DASH Media Segments.
	
	7.2.2 Syntax
	
	`aligned(8) class SampleEncryptionBox`
	`  extends FullBox('senc', version=0, flags)`
	`{`
	`  unsigned int(32) sample_count;`
	`  {`
	`  unsigned int(Per_Sample_IV_Size*8) InitializationVector;`
	`  if (flags & 0x000002)`
	`  {`
	`  unsigned int(16) subsample_count;`
	`  {`
	`  unsigned int(16) BytesOfClearData;`
	`  unsigned int(32) BytesOfProtectedData;`
	`  } [ subsample_count ]`
	`  }`
	`  }[ sample_count ]`
	`}`
	
	Source: Ex. 92 (ISO/IEC 23001-7:2016(E)) at 6.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	Here, the obtained Prime Video mp4 container files use 64-bit initialization vectors. As shown below, the TrackEncryptionBox information ('tenc') obtained using mp4box from streamed Amazon Prime Video video content, *e.g.*, from *Wildest Places, Wild Coasts (Episode 4)*, indicates use of 64-bit initialization vectors. Here, the IV_size is set to '8,' which indicates use of 64-bit initialization vectors.  Source: Screenshot of TrackEncryptionBox information using mp4box from *Wildest Places, Wild Coasts (Episode 4)*, streamed using the Prime Video application installed on the Fire Stick, taken by, or on behalf of, DivX. Specifically, as shown below within a SampleEncryptionBox ('senc'), each "partially encrypted frame" (*e.g.,* sample) corresponds to a SampleEncryptionEntry. Each SampleEncryptionEntry includes a unique IV value. In the first Sample Encryption Entry, the IV_size="8" and IV= "0x0D9CC53C17D3DDB2:" 

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	Source: Excerpt of mp4box output for *Wildest Places, Wild Coasts (Episode 4)*, streamed using the Prime Video application installed on the Fire Stick, taken by, or on behalf of, DivX.   Source: Screenshots of the same video data as the above excerpted mp4box output as viewed using MP4Box.js for *Wildest Places, Wild Coasts (Episode 4)*, streamed using the Prime Video application installed on the Fire Stick, taken by, or on behalf of, DivX.

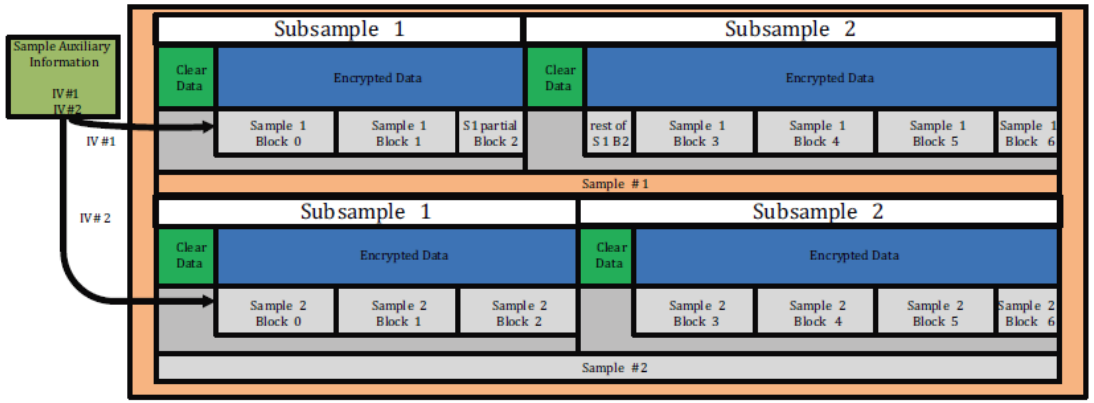
553 Patent	Amazon's Fire TV Stick (3rd Gen)
	As displayed above using MP4Box.js, the value of the IV for the first sample is "0d9cc53c17d3ddb2." Thus, the Prime Video application causes the Fire Stick to obtain a portion of a container file—such as mp4 container files—comprising "cryptographic material comprising a wrapped frame key."

553 Patent	Amazon's Fire TV Stick (3rd Gen)
[11(b)(i)(C)] a block reference that identifies an encrypted portion of the partially encrypted frame;	An Amazon Fire TV Stick (3rd Gen), Build Model AFTSSS ("Fire Stick"), as exemplary of all Amazon Accused Products with Amazon Fire OS, comprises a playback application to cause the set of one or more processors to perform the steps of "obtaining, at a processor, at least a portion of a container file." For example, the Fire Stick includes a playback application such as the Prime Video application. *See* **Element [11(b)]**, *supra*. The Prime Video application causes the Fire Stick to obtain a portion of a container file—such as a mp4 container file. *See* **Element [11(b)(i)]**, *supra*. "The at least a portion of the container file" (*e.g.*, the at least a portion of the mp4 container file obtained by the Fire Stick via the Prime Video application) "comprises: . . . a block reference that identifies an encrypted portion of the partially encrypted frame" (*e.g.*, "BytesOfClearData" (or "NumClearBytes") and "BytesOfProtectedData" (or "NumEncryptedBytes") values in a SubSampleEncryptionEntry box). Accordingly, the Prime Video application causes Fire Stick to obtain a portion of a container file—such as an mp4 container file—with "a block reference that identifies an encrypted portion of the partially encrypted frame." As shown above in **Element [11(b)(i)(A)]**, the ExoPlayer code base applies the 'cenc' AES-CTR DRM scheme in the ISO/IEC 23001-7 specification. The ISO/IEC 23001-7 specification supports subsample encryption (partial frame encryption) with the 'cenc' AES-CTR scheme, including specifying the number of bytes of protected (encrypted) data. Ex. 92 (ISO/IEC 23001-7:2016(E)) at 5, 10, 14.

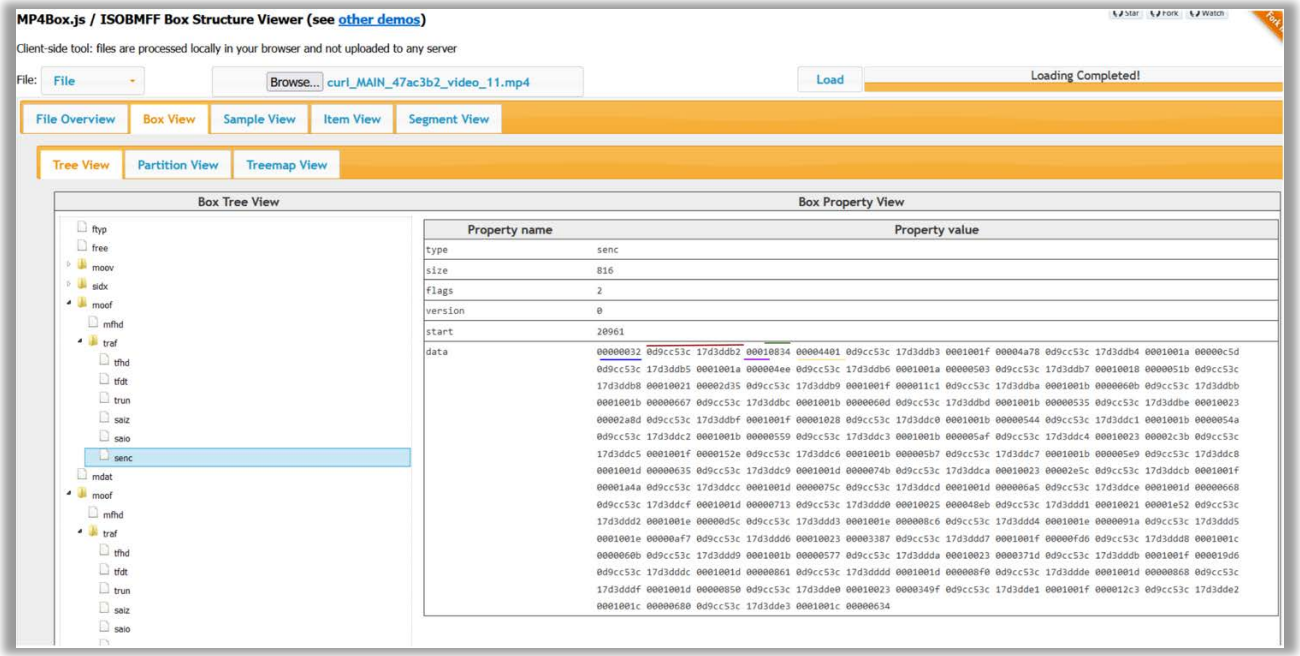
553 Patent	Amazon's Fire TV Stick (3rd Gen)

	7.1 Definition
	
	Each protected sample in a protected track SHALL have an Initialization Vector associated with it. Both Initialization Vectors and Subsample encryption information MAY be provided as Sample Auxiliary Information with `aux_info_type` equal to the scheme and `aux_info_type_parameter` equal to 0.
	
	For example, for tracks protected using the 'cenc' scheme, the default value for `aux_info_type` is 'cenc' and the default value for the `aux_info_type_parameter` is 0, so content SHOULD be created omitting these optional fields. Storage of sample auxiliary information SHALL conform to ISO/IEC 14496-12.
	
	The format of the sample auxiliary information for samples with this type SHALL be as follows:
	
	`aligned(8) class CencSampleAuxiliaryDataFormat`
	`{`
	`  unsigned int(Per_Sample_IV_Size*8) InitializationVector;`
	`  if (sample_info_size > Per_Sample_IV_Size )`
	`  {`
	`  unsigned int(16) subsample_count;`
	`  {`
	`  unsigned int(16) BytesOfClearData;`
	`  unsigned int(32) BytesOfProtectedData;`
	`  } [subsample_count ]`
	`  }`
	`}`
	
	Source: Ex. 92 (ISO/IEC 23001-7:2016(E)) at 5.

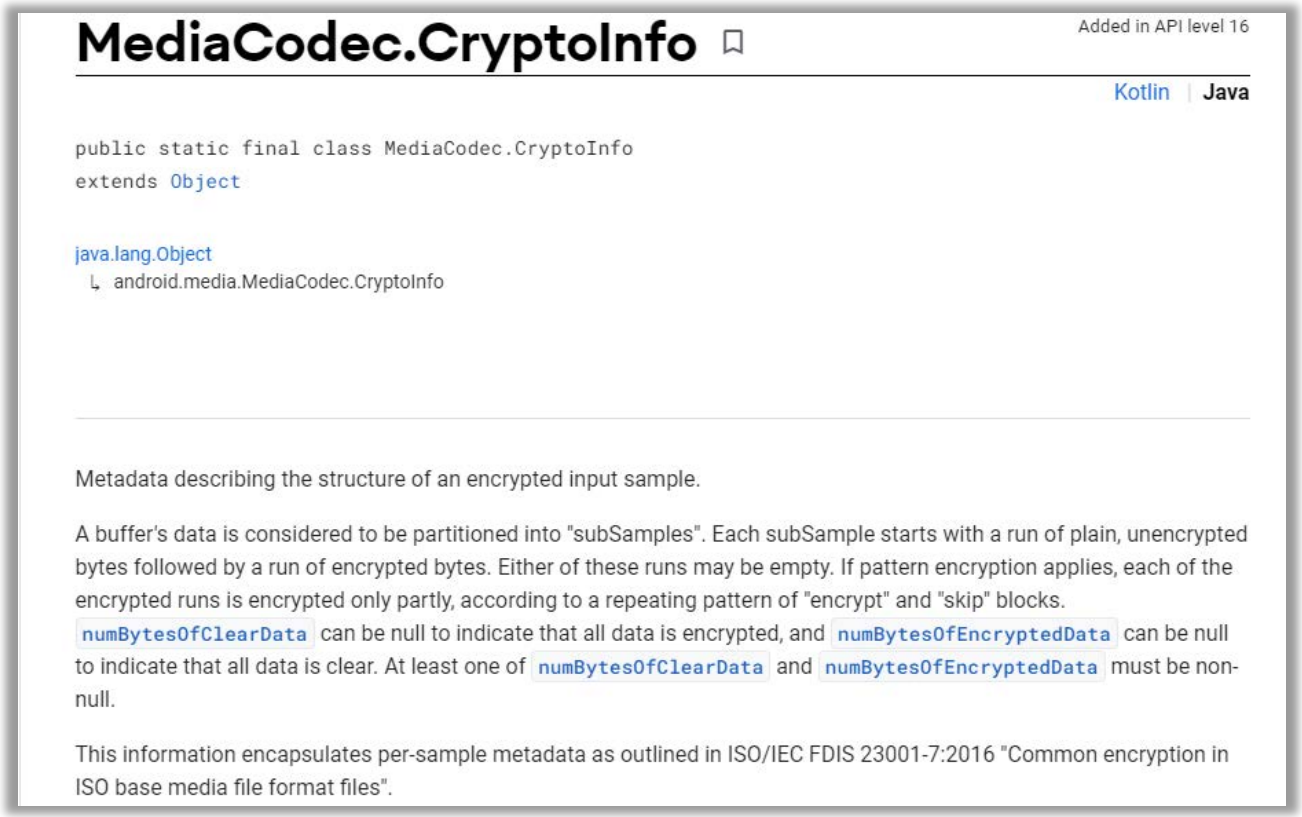
553 Patent	Amazon's Fire TV Stick (3rd Gen)
	— `BytesOfClearData` specifies the number of bytes of clear data at the beginning of this Subsample encryption entry. NOTE This value may be zero if no clear bytes exist for this Subsample. — `BytesOfProtectedData` specifies the number of bytes of protected data following the clear data. NOTE This value may be zero if no protected bytes exist for this Subsample. The Subsample encryption entries SHALL NOT include an entry with a zero value in both the `BytesOfClearData` field and in the `BytesOfProtectedData` field. The total length of all `BytesOfClearData` and `BytesOfProtectedData` in a sample SHALL equal the length of the sample. Subsample encryption entries SHOULD be as compactly represented as possible. For example, instead of two entries with {15 clear, 0 protected}, {17 clear, 500 protected}, use one entry of {32 clear, 500 protected}. If pattern-based encryption is used, then the pattern applies to the protected byte range, `BytesOfProtectedData`; otherwise, all protected bytes are encrypted. Source: Ex. 92 (ISO/IEC 23001-7:2016(E)) at 10. The wrapped frame keys (*e.g.*, "IVs") are used to decrypt the encrypted portions of the corresponding sample, as explained below. In one example, each sample comes "with a per-sample Initialization Vector and a logically continuous sequence of 16-byte cipher blocks interspersed with unencrypted byte ranges." Ex. 92 (ISO/IEC 23001-7:2016(E)) at 14.

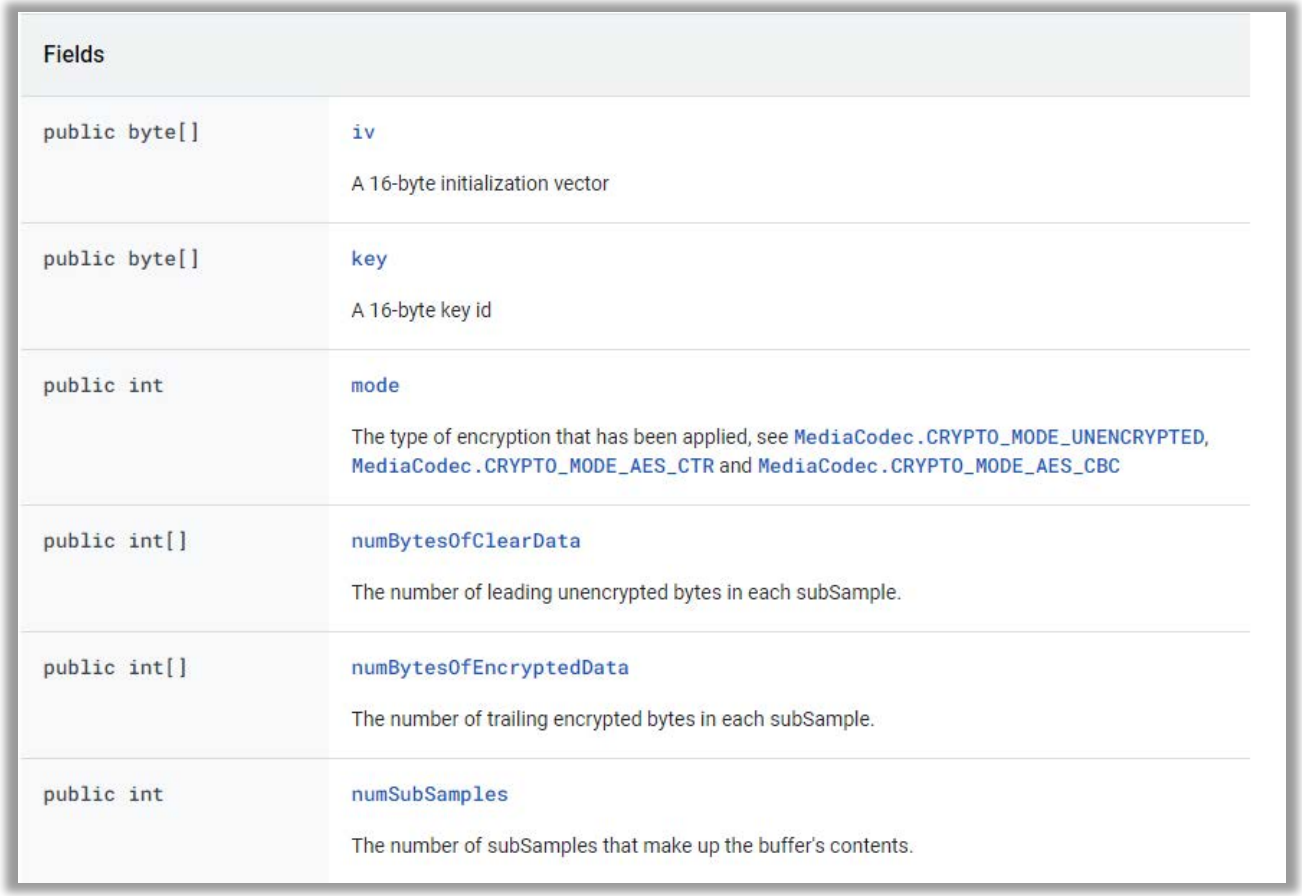
553 Patent	Amazon's Fire TV Stick (3rd Gen)
	Figure 3 is a Subsample encryption example showing two samples, each containing two Subsamples, each with a per-sample Initialization Vector and a logically continuous sequence of 16-byte cipher blocks interspersed with unencrypted byte ranges. Note that Block 2 of sample 1 is continued in the second Subsample, which is possible, but not recommended, with the scheme 'cenc', but not 'cens', 'cbc1' and 'cbcs'.  NOTE Cipher block and counter chaining is continuous from Subsample 1 to Subsample 2 in the first sample so that all cipher blocks are 16 bytes, except possibly the last cipher block in each sample. Block 6 is shown as smaller than 16 bytes to illustrate that CTR mode can encrypt cipher blocks smaller than 16 bytes without adding padding that changes the file size. Block 6 would be unencrypted if CBC mode were used. **Figure 3 — CTR Subsample encryption using one IV per sample** Source: Ex. 92 (ISO/IEC 23001-7:2016(E)) at 14. Specifically, as shown below within a SampleEncryptionBox ('senc'), each video data comprising a partially encrypted frame (*e.g.*, sample) corresponds to a SampleEncryptionEntry. Each SampleEncryptionEntry includes "NumClearBytes" and "NumEncryptedBytes" values in a SubSampleEncryptionEntry box to describe the unencrypted ("clear") and encrypted data, respectively, in the sample.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	As shown below, the obtained Prime Video mp4 container file contains a Sample Encryption Box 'senc' (*e.g.,* <SampleEncryptionBox…>). In this example, the Sample Encryption Box contains 50 frames (*e.g.*, samples) (sampleCount="50"). In the first Sample Encryption Entry, the IV_size="8" and IV= "0x0D9CC53C17D3DDB2." Additionally, for the first Sample Encryption Entry includes "a block reference that identifies an encrypted portion of the partially encrypted frame." In the SubSampleEncryptionEntry, the runs of bytes that are unencrypted ("clear") (*e.g.*, NumClearBytes="2100") and encrypted (*e.g.*, the NumEncryptedBytes ="17409") "identify an encrypted portion of the partially encrypted frame" because the run of encrypted bytes points to where the location of where the encrypted portion of the frame begins.  Source: Excerpt of mp4box output for *Wildest Places, Wild Coasts (Episode 4)*, streamed using the Prime Video application installed on the Fire Stick, taken by, or on behalf of, DivX.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	 Property name Property value type senc size 816 flags 2 version 0 start 20961 data 00000032 0d9cc53c 17d3ddb2 00010834 00004401 0d9cc53c 17d3ddb3 0001001f 00004a78 0d9cc53c 17d3ddb4 0001001a 00000c5d 0d9cc53c 17d3ddb5 0001001a 000004ee 0d9cc53c 17d3ddb6 0001001a 00000503 0d9cc53c 17d3ddb7 00010018 0000051b 0d9cc53c Source: Screenshots of the same video data as the above excerpted mp4box output as viewed using MP4Box.js for *Wildest Places, Wild Coasts (Episode 4)*, streamed using the Prime Video application installed on the Fire Stick, taken by, or on behalf of, DivX. The same information is displayed in both the MP4Box.js view and excerpted mp4box output, consistent with 'cenc' AES-CTR mode full sample and video NAL subsample encryption:

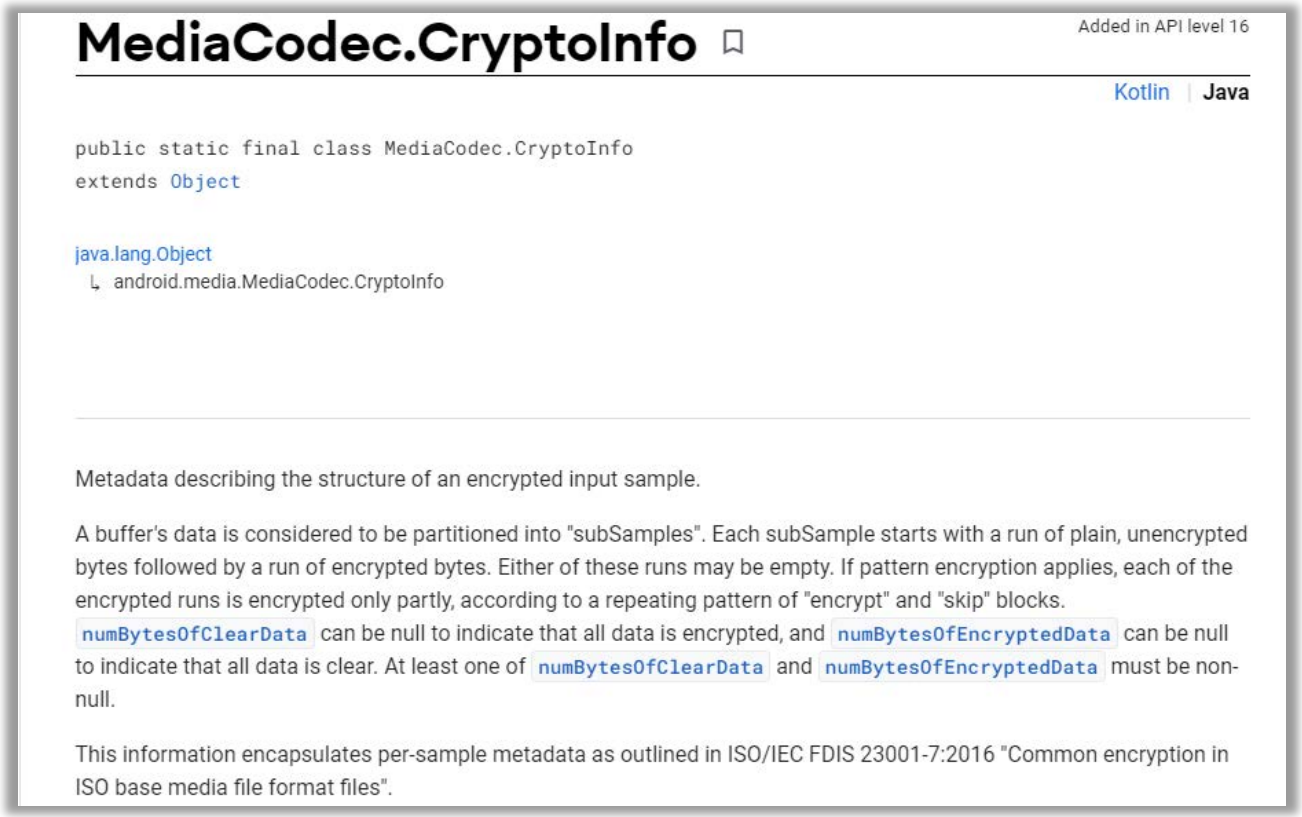
553 Patent	Amazon's Fire TV Stick (3rd Gen)
	• Blue (underlined): `sample_count` (number of frames in this segment) = Hex 00000032 = Dec 50. • Red (overlined): IV 64-bit `IV` • Purple (underlined): 16-bit `subsample_count` = 1 • Green (overlined): 16-bit `BytesOfClearData` = Hex 834 = Dec 2100 • Yellow (underlined): 32-bit `BytesOfProtectedData` = Hex 4401= Dec 17409 Further, the "MediaCodec.CryptoInfo" class used by the ExoPlayer code base specifies the portion of "the block reference that identifies an encrypted portion of the partially encrypted frame" (*e.g.*, "the number of trailing encrypted bytes in each subSample"). As shown below, each "partially encrypted frame" (*e.g.*, "subSample") "starts with a run of plain, unencrypted bytes followed by a run of encrypted bytes." https://developer.android.com/reference/android/media/MediaCodec.CryptoInfo.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	 Source: https://developer.android.com/reference/android/media/MediaCodec.CryptoInfo.

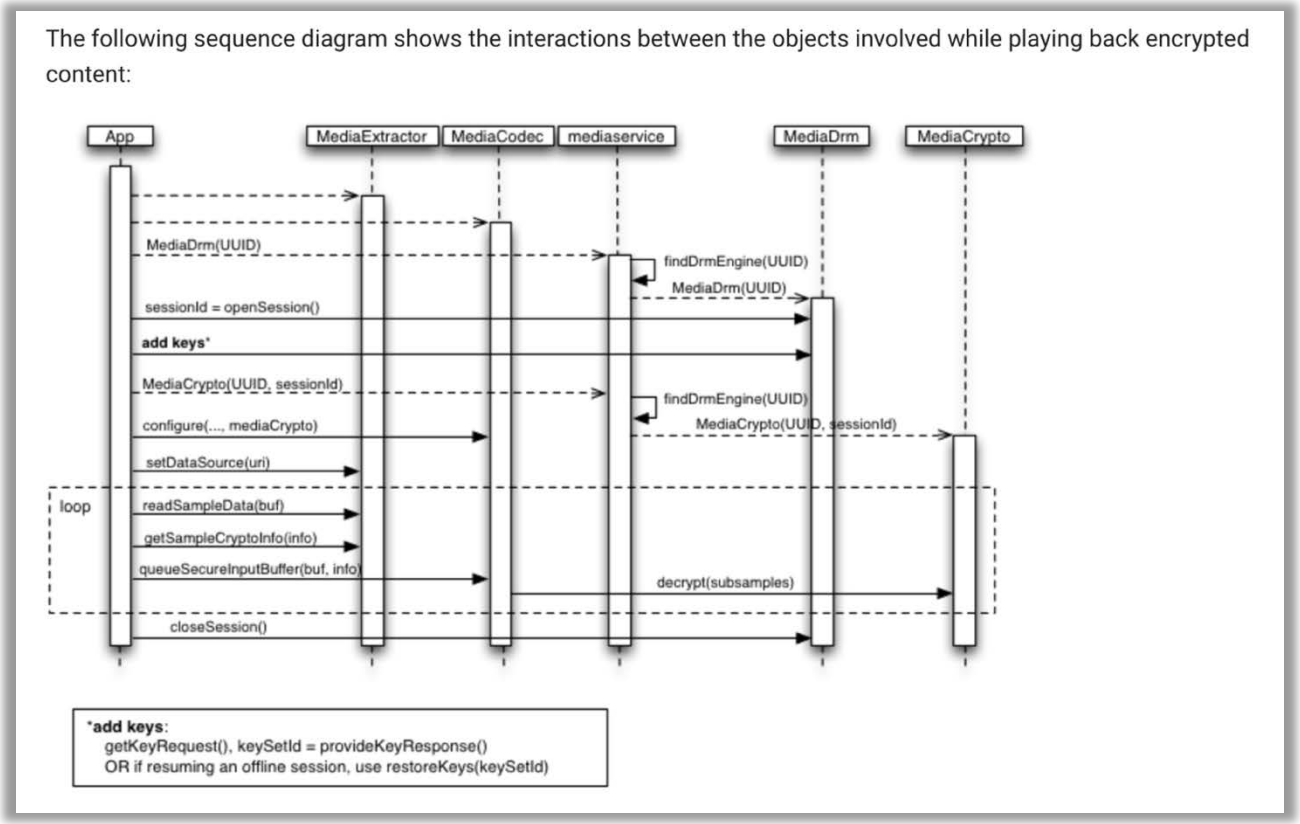
553 Patent	Amazon's Fire TV Stick (3rd Gen)
	 Fields public byte[] iv A 16-byte initialization vector public byte[] key A 16-byte key id public int mode The type of encryption that has been applied, see MediaCodec.CRYPTO_MODE_UNENCRYPTED, MediaCodec.CRYPTO_MODE_AES_CTR and MediaCodec.CRYPTO_MODE_AES_CBC public int[] numBytesOfClearData The number of leading unencrypted bytes in each subSample. public int[] numBytesOfEncryptedData The number of trailing encrypted bytes in each subSample. public int numSubSamples The number of subSamples that make up the buffer's contents. Source: https://developer.android.com/reference/android/media/MediaCodec.CryptoInfo. Thus, the Prime Video application causes Fire Stick to obtain a portion of a container file—such as an mp4 container file—with "a block reference that identifies an encrypted portion of the partially encrypted frame."

553 Patent	Amazon's Fire TV Stick (3rd Gen)
[11(b)(i)(D)] wherein the partially encrypted frame contains the encrypted portion and an unencrypted portion of data;	An Amazon Fire TV Stick (3rd Gen), Build Model AFTSSS ("Fire Stick"), as exemplary of all Amazon Accused Products with Amazon Fire OS, comprises a playback application to cause the set of one or more processors to perform the steps of "obtaining, at a processor, at least a portion of a container file." For example, the Fire Stick includes a playback application such as the Prime Video application. *See* **Element [11(b)]**, *supra*. The Prime Video application causes the Fire Stick to obtain a portion of a container file—such as a mp4 container file. *See* **Element [11(b)(i)]**, *supra*. "The at least a portion of the container file" (*e.g.*, the at least a portion of the mp4 container file obtained by the Fire Stick via the Prime Video application) "comprises: . . . a block reference that identifies an encrypted portion of the partially encrypted frame, wherein the partially encrypted frame contains the encrypted portion and an unencrypted portion of data." Accordingly, the Prime Video application causes Fire Stick to obtain a portion of a container file—such as an mp4 container file—with "a block reference that identifies an encrypted portion of the partially encrypted frame, wherein the partially encrypted frame contains the encrypted portion and an unencrypted portion of data." As discussed in **Element [11(b)(i)(C)]** above, the ISO/IEC 23001-7 specification specifies the number of bytes of protected ("encrypted") and clear ("unencrypted") data. *See* **Element [11(b)(i)(C)]**, *supra*. As shown below, the "partially encrypted frame" (*e.g.*, sample) "contains the encrypted portion and an unencrypted portion of data" (*e.g.*, "the number of bytes of protected data following the clear data" and "the number of bytes of clear data at the beginning of this Subsample encryption entry," respectively). The ISO/IEC 23001-7:2016(E) specification also specifies that "[t]he total length of all BytesOfClearData and BytesOfProtectedData in a sample SHALL equal the length of the sample," and that subsample encryption entries should use one entry ("wherein the partially encrypted frame contains the encrypted portion and an unencrypted portion of data"). Ex. 92 (ISO/IEC 23001-7:2016(E)) at 10 ("Subsample encryption entries SHOULD be as compactly represented as possible.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	For example, instead of two entries with {15 clear, 0 protected}, {17 clear, 500 protected}, use one entry of {32 clear, 500 protected}."). *Id*. — `BytesOfClearData` specifies the number of bytes of clear data at the beginning of this Subsample encryption entry. NOTE This value may be zero if no clear bytes exist for this Subsample. — `BytesOfProtectedData` specifies the number of bytes of protected data following the clear data. NOTE This value may be zero if no protected bytes exist for this Subsample. The Subsample encryption entries SHALL NOT include an entry with a zero value in both the `BytesOfClearData` field and in the `BytesOfProtectedData` field. The total length of all `BytesOfClearData` and `BytesOfProtectedData` in a sample SHALL equal the length of the sample. Subsample encryption entries SHOULD be as compactly represented as possible. For example, instead of two entries with {15 clear, 0 protected}, {17 clear, 500 protected}, use one entry of {32 clear, 500 protected}. If pattern-based encryption is used, then the pattern applies to the protected byte range, `BytesOfProtectedData`; otherwise, all protected bytes are encrypted. Source: Ex. 92 (ISO/IEC 23001-7:2016(E)) at 10. As shown below, the "partially encrypted frame" (*e.g.*, subSample) "contains the encrypted portion and an unencrypted portion of data" (*e.g.*, "a run of encrypted bytes" following "a run of plain, unencrypted bytes," respectively). The "CryptoInfo" class nested within the "MediaCodec" class used by the ExoPlayer code base specifies "the number of trailing encrypted bytes in each subSample" ("the encrypted portion…of data"). It also specifies "the number of leading unencrypted bytes in each subSample" ("the unencrypted portion of data"). Each "partially encrypted frame" (*e.g.*, subSample) "starts with a run of plain, unencrypted bytes followed by a run of encrypted bytes."

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	 Source: https://developer.android.com/reference/android/media/MediaCodec.CryptoInfo. Thus, the Prime Video application causes Fire Stick to obtain a portion of a container file—such as an mp4 container file—with "a block reference that identifies an encrypted portion of the partially encrypted frame, wherein the partially encrypted frame contains the encrypted portion and an unencrypted portion of data."

553 Patent	Amazon's Fire TV Stick (3rd Gen)
[11(b)(ii)] providing the partially encrypted frame, the cryptographic material, and the block reference from the parser to a video decoder, where the partially encrypted frame and the wrapped frame key are provided by the parser to the video decoder over an unsecured channel;	An Amazon Fire TV Stick (3rd Gen), Build Model AFTSSS ("Fire Stick"), as exemplary of all Amazon Accused Products with Amazon Fire OS, comprises a playback application to cause the set of one or more processors to perform the steps of "providing the partially encrypted frame, the cryptographic material, and the block reference from the parser to a video decoder, where the partially encrypted frame and the wrapped frame key are provided by the parser to the video decoder over an unsecured channel." For example, the Fire Stick includes a playback application such as the Prime Video application. *See* **Element [11(b)]**, *supra*. The Prime Video application causes the Fire Stick to "provid[e] the partially encrypted frame, the cryptographic material, and the block reference from the parser to a video decoder, where the partially encrypted frame and the wrapped frame key are provided by the parser to the video decoder over an unsecured channel." As discussed above in **Element [11(b)(i)]**, the ExoPlayer code base implemented by streaming applications such as the Prime Video application installed on Fire Stick supports parsing container files using "Extractors." *See* **Element [11(b)(i)]**, *supra*. The following diagram shows how Android apps use the "MediaDrm" API and "shows the interactions between the objects involved while playing back encrypted content." https://developer.android.com/reference/android/media/MediaDrm.html. Because Amazon Fire OS is a "fork of Android," the Prime Video application on the Fire Stick functions the same way. https://developer.amazon.com/docs/fire-tv/fire-os-overview.html.

553 Patent	Amazon's Fire TV Stick (3rd Gen)
	 Source: https://developer.android.com/reference/android/media/MediaDrm.html. The "MediaDrm" class and the "MediaExtractor" object are utilized to parse encoded media data. After the key is obtained via the "MediaCrypto" and "MediaCodec.CryptoInfo" objects, the "partially encrypted frame" (*e.g.*, the subSample), "the cryptographic material, and the block reference" (*e.g.*, DRM information including, the IV, the 16-bvte key id ("KID"), and numBytesofEncryptedData) are passed from the "MediaExtractor" parser to the decoder ("providing the partially encrypted frame, the cryptographic material, and the block reference from the parser to a video decoder"). The subSamples and DRM information "are only decrypted when the samples are delivered to the

553 Patent	Amazon's Fire TV Stick (3rd Gen)

	decoder" as they are passed over an unsecured channel. https://developer.android.com/reference/android/media/MediaDrm.html.
	
	When the app has constructed `MediaExtractor`, `MediaCodec` and `MediaCrypto` objects, it proceeds to pull samples from the extractor and queue them into the decoder. For encrypted content, the samples returned from the extractor remain encrypted, they are only decrypted when the samples are delivered to the decoder.
	
	Source: https://developer.android.com/reference/android/media/MediaDrm.html.
	
	As shown below in the Fire TV log excerpts, package AmazonVideo uses the "MediaDrm" API and constructs "MediaCodec" and "MediaCrypto" objects.
	
	`06-20 12:46:47.675 19153 20050 I AmazonVideo: Operation MediaDrm.openSession() PLAYREADY took duration 29 (unit MILLISECONDS, tag null)`
	`06-20 12:46:47.675 19153 20050 I AmazonVideo: MediaDrm Operation: Opened session | SessionId: sid2e | SessionType: IMMEDIATE | Scheme: PLAYREADY`
	`. . .`
	`06-20 12:46:47.770 19153 20050 I AmazonVideo: MediaDrm Operation: Generated license challenge, length: 6956 | SessionId: sid2e | SessionType: IMMEDIATE | Scheme: PLAYREADY`
	`. . .`
	`06-20 12:46:47.770 19153 20050 I AmazonVideo: MediaDrm Operation: Generated license challenge, length: 6956 | SessionId: sid2e | SessionType: IMMEDIATE | Scheme: PLAYREADY`
	`. . .`
	`06-20 12:46:48.419 19153 20050 I AmazonVideo: MediaDrm Operation: Processed license response, got back keySetId: a3NpZBQ2yw9n0jNCg2x+7XgwCLE= | SessionId: sid2e | SessionType: IMMEDIATE | Scheme: PLAYREADY`
	`. . .`

553 Patent	Amazon's Fire TV Stick (3rd Gen)

	`06-20 12:46:48.442 19153 20050 I AmazonVideo: MediaDrm Operation: KeySetId: a3NpZBQ2yw9n0jNCg2x+7XgwCLE= already restored | SessionId: sid2e | SessionType: IMMEDIATE | Scheme: PLAYREADY`
	Source: Excerpt from log file for *Wildest Places, Wild Coasts (Episode 4)*, downloaded using the Prime Video application installed on the Fire Stick, taken by, or on behalf of, DivX.
	
	`06-20 12:46:56.418 19153 20361 I AmazonVideo: MediaDrm Operation: Instantiated new MediaCrypto | SessionId: sid30 | SessionType: IMMEDIATE | Scheme: PLAYREADY`
	`. . .`
	`06-20 12:46:56.483 19153 20361 I AmazonVideo: MediaCodec configuring with format: {sample-rate=48000, mime=audio/mp4a-latm, channel-count=2}, surface: null, crypto android.media.MediaCrypto@a040796`
	`. . .`
	
	`06-20 12:46:56.560 19153 20361 I AmazonVideo: MediaCodec configuring with format: {max-height=1080, max-width=1920, mime=video/hevc, width=704, max-input-size=777600, height=396}, surface: Surface(name=null)/@0x4ff6f04, crypto android.media.MediaCrypto@a040796`
	Source: Excerpt from log file for *Wildest Places, Wild Coasts (Episode 4)*, downloaded using the Prime Video application installed on the Fire Stick, taken by, or on behalf of, DivX.
	
	Further, the partially encrypted frame, the cryptographic material, and the block reference are provided from the parser to a video decoder "over an unsecured channel."
	
	As discussed above in **Element [11(b)(i)],** the Prime Video application includes a parser (*e.g.*, "Extractor"). *See* **Element [11(b)(i)]**, *supra*. The parser (*e.g.*, "Extractor") uses an "unsecured channel." "The extractor service hosts the extractors, components that parse the various file formats supported by the media framework. The extractor service is the least privileged of all the services—it can't read [file descriptors ('FDs')] so instead it makes calls onto a Binder interface (provided to it by the media server for each playback session) to access files."