

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

NETFLIX, INC. AND HULU, LLC,
Petitioners,

v.

DIVX, LLC,
Patent Owner.

IPR2020-00558

U.S. Patent No. 10,225,588

PETITION FOR *INTER PARTES* REVIEW

OF U.S. PATENT NO. 10,225,588

TABLE OF CONTENTS

EXHIBIT LIST	4
I. Background.....	5
A. Adaptive Bitrate Streaming.....	5
B. DRM	7
1. Key Management	7
2. Partial Encryption	8
II. The '588 Patent.....	9
A. Admitted Art.....	9
B. Summary of the Alleged Invention	10
C. The Alleged Invention According to Patent Owner.....	11
D. Priority Date	11
III. Identification of Challenge	12
A. Statutory Grounds.....	12
B. Relied-Upon Prior Art.....	12
1. U.S. Patent Pub. No. 2011/0096828 (“Chen”) (Exhibit 1006)	12
2. U.S. Patent Pub. No. 2007/0083467 (“Lindahl”) (Exhibit 1007)	13
3. U.S. Patent 8,683,066 (“Hurst”) (Exhibit 1008).....	13
C. Standing.....	14
D. Discretionary Analysis for Review	14
IV. Level of Ordinary Skill.....	14

V.	Claim Construction.....	16
VI.	Challenged Claims.....	17
A.	Ground 1: Chen in View of Lindahl and Hurst.....	17
1.	Motivation to Combine	17
2.	Claim 1	22
3.	Claim 2	58
4.	Claim 3	62
5.	Claim 4	64
6.	Claim 5	64
7.	Claim 6	65
8.	Claim 7	66
9.	Claim 8	69
10.	Claim 9	71
11.	Claim 10	73
12.	Claim 11	74
13.	Claim 12	78
14.	Claims 13-22	79
15.	Claim 23	80
16.	Claim 24	83
VII.	MANDATORY NOTICES AND FEES	84
	CLAIM APPENDIX	86

EXHIBIT LIST

Exhibit No.	Description
1001	U.S. Patent No. 10,225,588 to Kiefer et al. (“the ’588 patent”)
1002	Prosecution File History for the ’588 patent
1003	Declaration of Patrick McDaniel, Ph.D.
1004	Prosecution File History for U.S. Patent Application No. 13/340,623 to Kiefer et al. (“Kiefer”)
1005	U.S. Provisional Patent Application No. 61/530,305
1006	U.S. Patent Pub. No. 2011/0096828 (“Chen”)
1007	U.S. Patent Pub. No. 2007/0083467 (“Lindahl”)
1008	U.S. Patent No. 8,683,066 (“Hurst”)
1009	U.S. Patent No. 8,849,950 (“Stockhammer”)
1010	U.S. Patent No. 8,243,924 (“Chen-924”)
1011	U.S. Patent No. 6,415,031 (“Colligan”)
1012	U.S. Patent No. 6,957,350 (“Demos”)
1013	U.S. Patent No. 7,295,673 (“Grab-673”)
1014	U.S. Patent No. 5,574,785 (“Ueno”)
1015	Divx, LLC v. Netflix Inc., No. 2:19-cv-1602-PSG, Am. Compl. (C.D. Cal Aug. 21, 2019)

The '588 patent was filed in 2011. It attempts to claim commonly-known aspects of adaptive bitrate streaming that had been widely used in the industry years before the patent was filed, including the conventional use of encryption, common keys, and file formats.

I. BACKGROUND

By 2011, Netflix and Hulu had launched video streaming services. Many technology companies, such as Microsoft, Apple and Adobe, provided tools for adaptive bitrate streaming. *See* Ex. 1009, 2:1-10; Ex. 1010, 1:31-59, 2:59-3:22, 3:34-4:16. Several common components of adaptive streaming had become standardized, including file formats, digital rights management (“DRM”) and encryption. *See* Ex. 1010, 8:47-9:8.

A. Adaptive Bitrate Streaming

Adaptive bitrate streaming compensated for changes in network bandwidth to deal with variable Internet streaming conditions. Ex. 1003, ¶70. Typically, video was encoded at several bitrates, and playback switched between them depending on the available channel bandwidth. *Id.*; *see* Ex. 1009, 2:26-30; Ex. 1010, 14:61-15:14.

Popular standards, such as MPEG Dynamic Adaptive Streaming over HTTP (“DASH”) included a Media Presentation Description (MPD) that referenced the URLs for fragments of each representation of video. Ex. 1003, ¶¶71, 74, 76; *see*

Ex. 1009, 2:19-25. Client devices used the MPD to select an appropriate representation based on network conditions, using low bitrates during network congestion or higher bandwidths if possible. *See id.*, 2:30-38; Ex. 1003, ¶71.

Video streams were typically stored in media container files. Several media container formats existed, including the ISO base media file format, sometimes referred to as the MP4 file format. *See* Ex. 1010, 2:59-4:5; Ex. 1001, 2:27-34. Container specifications defined the file format including the location of audio and video data and metadata for indexing and accessing specific portions of the content. Ex. 1003, ¶72. Adaptive streaming systems utilized these indices to jump to specific portions of the content when switching between alternate video files. *Id.*

Video streaming typically used the ubiquitous HTTP protocol, which allowed clients to request documents from servers based on a URL. Ex. 1003, ¶73. Starting with HTTP version 1.1 in 1997, the HTTP protocol allowed requests for specific byte ranges of files, sometimes referred to as partial GET requests. *Id.*; *see* Ex. 1009, 2:44-3:3. The byte-range request feature of HTTP was commonly used, along with file container indices, to transition between alternative video files in adaptive bitrate streaming systems. *Id.*

Prior art MPEG-DASH generally divided media into segments encoded at different bitrates, which were stored on servers and downloaded by clients through

HTTP GET requests, including partial GET requests as described above. Ex. 1003, ¶¶74-75; *see* Ex. 1009, 2:11-3:3, 5:34-6:8, 6:33-7:39. An MPD described the segments, typically in XML format, listing the different representations of content with URLs for clients to use with HTTP Get requests. Ex. 1003, ¶¶76-78.

B. DRM

DRM had long been used to protect video from unauthorized copying, typically through encryption so only authorized users with the proper keys could decrypt and use the content. Ex. 1003, ¶79. Most DRM systems included functionality for encryption/decryption; key management, which tracked the keys associated with encrypted media files and provided them to authorized client devices; access management; and authentication of client and server devices. Ex. 1003, ¶¶79-80.

1. Key Management

A common DRM approach was to encrypt each movie with unique keys that cannot be used to decrypt any other movie. Ex. 1003, ¶81. For prior art MPEG-DASH systems, it was common for a single key or a single set of keys to encrypt all the alternative streams for a video because it simplified key management, given that permission to adaptively stream a video generally included permission to access the alternative representations of that video. *Id.* For example, when a client began streaming a movie, the client was expected to switch between the alternative

streams of that movie depending on the available bandwidth—this was the basic premise of adaptive bitrate streaming. *See* §I.A. Because access was typically granted to alternative streams as a group, it was common to encrypt them as a group, using the same key, which simplified key management by allowing the service to send over that one key, which the client would then use to decrypt the alternative streams. *See* Ex. 1008, 18:62-67. When a set of keys was used to encrypt a stream, for example by cycling through them, it was common to use that same set of keys to encrypt the alternative streams. *See, e.g.*, Ex. 1013, 7:15-52, Fig. 9; Ex. 1003, ¶81.

To manage keys, it was common to associate each key with a unique identifier called a “key ID” that could be used to reference a particular key. This allowed systems to identify which keys were used to encrypt which files, thereby enabling the decryption of those files using the identified key. *See* Ex. 1014, 1:43-52, Fig. 4; Ex. 1013, 9:34-39, Fig. 9; Ex. 1003, ¶82. Keys were often stored in key tables alongside their key IDs, so they could be easily organized and accessed. *See id.* Since video was typically encrypted using symmetric key encryption, the same key used to encrypt data was needed to decrypt it. Ex. 1003, ¶82.

2. Partial Encryption

Partial encryption was a common DRM feature for video streaming by the 2000s. With partial encryption (also referred to as selective encryption), only

portions of a video stream were encrypted, leaving the remaining portions unencrypted. *See* Ex. 1012, 23:17-24:49; Ex. 1003, ¶83. Partial encryption balanced security with computational load. *See id.*

II. THE '588 PATENT

A. Admitted Art

The '588 patent admits that adaptive bitrate streaming was already known, where content is “encoded as alternative streams that are suitable for different network connection bandwidths.” Ex. 1001, 1:39-64; Ex. 1003, ¶85. “Typically, the source media is encoded at multiple bitrates and the playback device or client switches between streaming the different encodings depending on available resources.” *Id.*, 1:64-67. Ex. 1003, ¶86. Known “[a]daptive streaming solutions typically utilize... Hypertext Transfer Protocol (HTTP)” which “enables a playback device to request a byte range within a file.” *Id.* 2:5-7; Ex. 1003, ¶87. Thus, the '588 patent admits that in adaptive bitrate streaming, it was known for a playback device to obtain a top level index file identifying alternative streams of video and request portions of a selected stream with byte range requests (e.g., using HTTP). Ex. 1003, ¶89.

“In adaptive streaming systems, the source media is typically stored on a media server as a top level index file pointing to a number of alternate streams” and “[e]ach stream is typically stored in one or more container files.” Ex. 1001,

2:12-16; Ex. 1003, ¶88. The '588 patent admits that “several adaptive streaming solutions” are known in the art, including from Apple, Microsoft, and Adobe, where the latter two used the MP4 container format, which was one of “[t]he most commonly used media container formats...” *Id.* 2:15-34, 2:35-45; Ex. 1003, ¶¶90-91, 93.

The '588 patent further admits that known DRM “typically involve[d] the use of cryptographic information to control access to or protect a piece of content” including using “one or more encryption keys to encrypt some or all of the content”—referring here to known partial encryption. Ex. 1001, 2:50-57; Ex. 1003, ¶92.

B. Summary of the Alleged Invention

The '588 patent claims recite various details of adaptive bitrate streaming and partial encryption, such as a top level index file identifying a plurality of alternative streams of protected video, selecting a stream based on detected streaming conditions, and requesting portions of the selected stream based on the byte ranges, which the '588 patent admits was known in the art. *See* §II.A; Ex. 1001, 1:45-2:62; Ex. 1003, ¶94. The specification does not clearly identify what the alleged invention is purported to be; the “Summary of the Invention” is a recitation of embodiments that spans 5 columns. *See* Ex. 1001, cols. 2-7; Ex. 1003, ¶94.

The claims also recite basic DRM features—including partial encryption and encrypting the alternative streams using a set of common keys comprising at least one key—but the specification is sparse with details regarding whether or how this differs from what was already widely known and practiced in the art. Ex. 1003, ¶95.

C. The Alleged Invention According to Patent Owner

According to PO, the '588 patent is directed to using “common frame encryption keys to encode alternate video streams.” Ex. 1015, ¶224. According to PO, the '588 patent recites use of a “new file type” and “byte ranges” to “provide the playback device with newly available computing resources.” Ex. 1015, ¶232. No other alleged point of novelty is explained in DivX’s complaint. Ex. 1003, ¶96.

D. Priority Date

While the '588 patent cites a provisional application filed September 2011, the claim limitations regarding “receiving a container index that provides byte ranges” are unsupported by the provisional. *See* Ex. 1005. The priority date is thus December 29, 2011 at the earliest. The relied-upon references here are prior art regardless. Ex. 1003, ¶97.

III. IDENTIFICATION OF CHALLENGE

A. Statutory Grounds

Petitioner requests *inter partes* review and cancelation of the challenged claims on the following grounds:

Grounds	Claims	Statutory Basis	Prior Art
1	1-24	§103	Chen in View of Lindahl and Hurst

Lindahl and Hurst were not cited or considered during prosecution of the '588 patent. Chen was included in an information disclosure statement (“IDS”) alongside over 1,000 other references, in seven separate IDSs filed the same day, spanning 47 pages. *See* Ex. 1002, 102-107, 390-392, 405-407, 415-420, 433-454, 467-472, 485-488; Ex. 1003, ¶104. However, Chen was never specifically discussed by the examiner or raised in any office actions. *Id.*

B. Relied-Upon Prior Art¹

1. U.S. Patent Pub. No. 2011/0096828 (“Chen”) (Exhibit 1006)

Chen was filed on September 21, 2010, claims priority to September 2009 and published April 2011. Chen is prior art under at least §§102(a) and (e). Ex. 1003, ¶105.

Chen teaches HTTP streaming where video is “encoded at multiple bitrates to form different versions, or representations.” Ex. 1006, ¶[0063], Abstract. These

¹ Pre-AIA 35 U.S.C. §102 applies. MPEP §2159.02.

representations are “broken into smaller pieces, perhaps on the order of a few seconds each, to form segments.” *Id.* As the client device requests segments, it “switch[es] to different data rates based on available bandwidth.” *Id.* ¶[0064]. An MPD file identifies the alternative representations of a media item, providing their respective bitrates, URLs, and other information, which the client uses to request an appropriate segment based on the available bandwidth. *See id.*, ¶¶[0063-0064, 0066-0068]; Ex. 1003, ¶¶106-107.

2. U.S. Patent Pub. No. 2007/0083467 (“Lindahl”) (Exhibit 1007)

Lindahl was filed on October 10, 2005 and published on April 2007.

Lindahl is prior art under §§102(a), (b), and (e). Ex. 1003, ¶¶108-109.

Lindahl teaches DRM for media files, including techniques for partially encrypting portions of media files, which “substantially disrupt[s] usefulness of other parts of the media file which are not encrypted.” Ex. 1007, Abstract.

3. U.S. Patent 8,683,066 (“Hurst”) (Exhibit 1008)

Hurst was filed on August 6, 2007, published February 2009 and granted March 2014. Hurst is prior art under §§102(a), (b) and (e). Ex. 1003, ¶¶110-111.

Hurst teaches “adaptive-bitrate content streaming” where content is encoded as “a plurality of streams having varying degrees of quality and bandwidth.” Ex. 1008, 3:65-68, FIG. 2b The streams are encrypted with the same key. *Id.*, 6:59-62, 18:64-67; Ex. 1003, ¶111.

C. Standing

Petitioner certifies that Petitioner is not barred or estopped from requesting this inter partes review and the '588 patent is eligible for it.

D. Discretionary Analysis for Review

The General Plastic factors weighs in favor of institution. This is the first petition filed against the '588 patent. Institution would be an efficient use of resources. No issues are presented that would inhibit the issuance of a final determination within one year of institution. Denial of institution under §325(d) would not be appropriate because the grounds and arguments relied on by this Petition were not previously presented to the Patent Office. The underlying litigations are in their infancy with limited discovery and no claim construction briefing yet. Institution is consistent with the rationale for the one-year bar expressed in the AIA's legislative history. *See Inter Corp. v. VLSI Tech. LLC*, IPR2019-01192, Paper 15 (PTAB Jan. 9, 2020).

IV. LEVEL OF ORDINARY SKILL

A person of ordinary skill in the field of art ("POSITA") of the '588 patent would have had a bachelor's degree in mechanical engineering, electrical engineering, computer science, or a similar field with at least two years of experience in video streaming and media security or a person with a master's degree in mechanical engineering, electrical engineering, computer science, or a

similar field with a specialization in video streaming and media security. A person with less education but more relevant practical experience may also meet this standard. Ex. 1003, ¶¶65-67.

Given this level of skill in the 2011 time frame, a POSITA would have had the knowledge and skills necessary to program and build systems that meet all limitations of the challenged claims, including the following:

- Compressing and decompressing video, including by using common video standards such as MPEG and AVC. *See* Ex. 1007, ¶[0006]; Ex. 1009, 1:15-37.
- Using known video container file formats for streaming, such as the ISO base media file format and 3GPP extensions. *See* Ex. 1009, 1:15-37; Ex. 1010, 2:59-3:6; Ex. 1001, 2:27-41.
- Using known communication protocols for streaming, including Hypertext Transfer Protocol (HTTP). *See* Ex. 1009, 1:38-3:3; Ex. 1010, 3:7-22; Ex. 1001, 2:1-10.
- Using known adaptive streaming technologies to stream video over the Internet, including MPEG Dynamic Adaptive Streaming over HTTP (DASH) and adaptive bitrate streaming solutions from companies such as Microsoft, Adobe, and Apple. *See* Ex. 1009, 2:1-3:3; Ex. 1001, 2:12-34.

- Encrypting and decrypting data, including by using common symmetric key algorithms (e.g., Blowfish, AES, DES) and asymmetric key algorithms. *See, e.g.*, Ex. 1007, ¶[0008]; Ex. 1010, 7:34-42; Ex. 1001, 2:47-62.
- Using DRM for video security, including for authentication, key management (such as the assignment and identification of keys), and partial encryption of video streams. *See, e.g.*, Ex. 1011, 1:49-63, 2:1-8, 8:23-32; Ex. 1013, 6:15-56, 7:15-8:42.

Ex. 1003, ¶68; *see* §II.A.

V. CLAIM CONSTRUCTION

For all grounds, the challenged claims are invalid under their plain and ordinary meaning. Ex. 1003, ¶100-101. Petitioner's arguments are not a concession on the proper scope of any claim term or waiver of any argument that claim terms are indefinite. While there may be issues that prevent a POSITA from determining the scope of the claims with reasonable certainty, the prior art and grounds here fall squarely within the scope and nowhere near the boundaries of the challenged claims. *Id.* It is not necessary to determine indefiniteness to determine invalidity under the present grounds. *Id.*

VI. CHALLENGED CLAIMS

A. Ground 1: Chen in View of Lindahl and Hurst

1. Motivation to Combine

A POSITA would have been motivated to apply teachings from Lindahl and Hurst to Chen—specifically to combine (a) Chen’s adaptive streaming teachings with (b) Lindahl’s DRM teachings, including partial encryption and key management and (c) Hurst’s DRM teachings, as explained below. Ex. 1003, ¶¶113-128.

Adaptive bitrate streaming and DRM, as those taught by Chen, Lindahl, and Hurst, were commonly used together for video streaming, were widely recognized as complementary, and are used here for their known and conventional purposes. *See* Ex. 1010, Abstract; Ex. 1003, ¶114. The use of DRM (protecting content intellectual property) and adaptive streaming (addressing network instability) together were widely understood to be complementary requirements for paid streaming services. *Id.* It was widely known for video streaming to include these features to account for bandwidth variability over the Internet and address piracy. *See id.*; Ex. 1006, ¶¶[0103-0104]; Ex. 1007, ¶¶[0036, 0039]; Ex. 1008, 3:12-23, 6:6-58. Chen and Lindahl provide motivations to combine their teachings in this manner, and Hurst specifically teaches DRM for adaptive streaming. *See* Ex. 1008, Abstract, 1:6-10, 8:16-18, 16:61-17:58, Figure 10. Thus, a POSITA would

have been motivated to combine the references because they were complimentary and to enhance the efficiency and security of DRM for video streaming. Ex. 1003, ¶114.

While Chen accounted for the use of DRM in its teachings (see Ex. 1006, ¶[0522]), Chen did not require any particular DRM implementation, leaving particulars to the POSITA's knowledge and preferences. Ex. 1003, ¶115. Lindahl and Hurst provided such DRM teachings. Lindahl prevented unauthorized access to media while improving the computational efficiency. See Ex. 1007, ¶[0095]; Ex. 1003, ¶¶115, 121-122. Lindahl's partial encryption teachings reduce computational load while maintaining content protection, providing a balanced level of security that was well-suited for video streaming applications. Ex. 1003, ¶¶121-122. Similarly, its key management teachings provided security, simplicity and efficiency benefits. *Id.*

A POSITA would have been motivated to apply Lindahl's teachings to Chen to address piracy concerns and improve efficiency, which Chen also raises as a goal. Ex. 1003, ¶¶115, 119-120. The teachings of Chen relied on here reflect common, well-known aspects of adaptive streaming, and a POSITA would have been motivated to apply Chen's teachings in the manner explained in this Petition, to improve the efficiency of adaptive streaming, optimize the balance between bitrate and bandwidth, and improve the end-user experience with fast startup and

seek. Ex. 1003, ¶¶119-120; *see* Ex. 1006, ¶¶[0024], [0026]-[0027], [0063-0065], [0148].

Similarly, Hurst provides DRM protection, further teaching the keys used to encrypt content for adaptive streaming. Ex. 1003, ¶116. Hurst, like Chen, teaches encoding a video at different bitrates. *Id.* A POSITA would have been motivated to apply Hurst's teachings—e.g., where the alternative streams are encoded at different bitrates and encrypted as a group in the same manner, such as by using the same key—because they follow the usage model for video distribution in adaptive streaming, where the playback device gains access to the alternative streams as a group. *See id.*; Ex. 1006, ¶¶[0063-0064]; §I.B. For example, a playback device adaptively streaming a movie is typically granted access to its alternative streams for the movie because it is expected to switch between them depending on channel bandwidth. Ex. 1003, ¶116. Since the alternative streams are accessed as a group, it would have been obvious to encrypt them as a group, using the same set of keys, so that the alternative streams can be accessed as a group using that set of keys. *See id.* Thus, Hurst teaches the natural and obvious approach to DRM for adaptive streaming, which a POSITA would have been motivated to apply. *Id.*

Furthermore, the prior art provides express teaching and motivation to combine adaptive bitrate streaming, which Chen teaches, with partial encryption,

which Lindahl teaches. *See* Ex. 1010 (“Chen-924”), Abstract (teaching clients receiving content “adaptively streamed container over a network using one container and communication protocol” where “[t]he container may be received as selectively encrypted container”). Chen-924 teaches several embodiments that adaptively stream alternative representations of a video using partially encrypted container files. *See* Ex. 1010, 14:41-15:60. Thus a POSITA would have been motivated to combine Chen and Lindahl. Ex. 1003, ¶117.

Lindahl provides express motivation to combine its DRM teachings with known streaming techniques. *See* Ex. 1007, ¶[0065] (“...the download 708 of the media file can be performed by streaming the media file through the data network to the user...”). A POSITA would have been motivated to combine Lindahl’s DRM teachings with Chen’s streaming teachings. Ex. 1003, ¶118. And Lindahl comes from Apple Corporation, which was active in prior art file formats and adaptive bitrate streaming. *See* Ex. 1001, 2:23-27; Ex. 1003, ¶122.

Further, the references provide express motivation to modify their teachings. *See* Ex. 1006, ¶[0554]; Ex. 1007, ¶[0096]; Ex. 1008, 21:20-27; Ex. 1003, ¶124. While Hurst discloses other embodiments, e.g., services with standard and premium streams, those are not relied upon here, and a POSITA would have been motivated to apply Hurst’s teaching of encrypting alternative streams in the same manner, for the reasons explained above. *See* Ex. 1008, 18:66-67; Ex. 1003, ¶125.

A POSITA would have had a reasonable expectation of success in combining the combined teachings because they were widely known in the art and widely recognized as complementary and compatible techniques that were intended to be used together, and which a POSITA would have been familiar with. Ex. 1003, ¶126. By 2011, video streaming systems commonly (and almost universally) used adaptive bitrate streaming and DRM together. *Id.* For example, a video streaming system would encode multiple representations of a video at different bitrates, as taught by Chen, with the representations partially encrypted as taught by Lindahl, using common keys, as taught by Hurst. *Id.* Each aspect of the combination would be used in a conventional manner according to its known purpose. *Id.*

Furthermore, Chen and Hurst both apply their teachings to adaptive streaming frameworks such as MPEG-DASH. Ex. 1003, ¶127. A POSITA would have recognized that Lindahl's partial encryption teachings were fully compatible with Hurst's DRM teachings because the former was directed to which portions of the video stream are encrypted, while the latter was directed to the separate issue of which key to use for encryption. *Id.* Thus the combination of Chen, Lindahl and Hurst was easily within the level of ordinary skill. *Id.*

The alleged invention merely recites different aspects of video streaming that were already known and used together. Ex. 1003, ¶128. The challenged

claims are an obvious combination of prior art elements (e.g., adaptive streaming and partially encrypting alternative streams with the same set of keys) according to known methods (using Chen's adaptive bitrate streaming teachings to provide content to users, using Lindahl and Hurst's DRM teachings to encrypt content) to yield predictable results (a streaming service that accounts for network variability and protects access to content). *Id.* The combination applies known techniques (e.g., Chen's adaptive streaming teachings, and Lindahl and Hurst's DRM teachings) to improve similar devices (e.g., streaming systems) in the same way as they were known and used in the prior art. *Id.*

2. Claim 1

1[a]. A playback device for playing protected content from a plurality of alternative streams, comprising:

The combination teaches or suggests a playback device (e.g., client device) for playing protected (e.g., partially encrypted) content from a plurality of alternative streams (e.g., representations). Ex. 1003, ¶¶129-131. Chen and Lindahl teach a client device, at the consumer/user, for playing at least one of multiple representations of partially encrypted content. *See, e.g.*, Ex. 1006, ¶¶[0061], [0203], [0366]; Ex. 1007, ¶[0011]; Ex. 1003, ¶129. Hurst refers to its client device as a media player and teaches that it plays protected content from a plurality of alternative streams. *See* Ex. 1008, 1:15-19, 6:13-17, 8:47-49. A

POSITA would have recognized these teachings to refer to analogous concepts of client devices that were widely-known and widely-used types of playback devices. Ex. 1003, ¶130. It would have been obvious for the client device to play protected content from a plurality of alternative streams, as explained below. *Id.* ¶131.

[1b] a set of one or more processors; and a non-volatile storage containing an application for causing the set of one or more processors to perform the steps of:

The combination teaches or suggests a playback device (e.g., client device) comprising a set of one or more processors housed within a computer or device and a non-volatile storage (e.g., disk storage, ROM) containing an application (e.g., software, programs) for causing the set of one or more processors to perform the steps with respect to the challenged claims, as discussed herein. Ex. 1003, ¶¶132-135. All three references implement their teachings with client devices comprising a processor executing software written on disk storage or ROM. *See* Ex. 1006, ¶¶[0116]-[0118], FIG. 4:

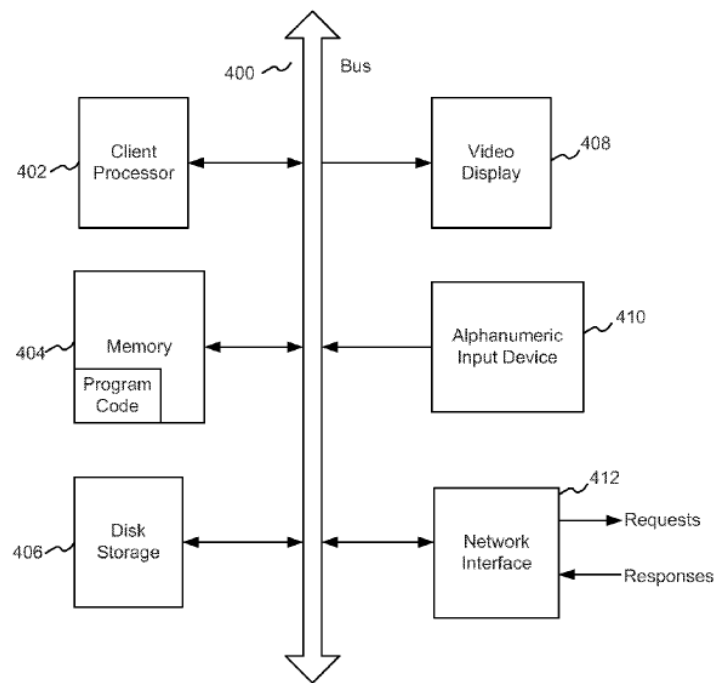


FIG. 4

See Ex. 1007, ¶[0088], FIG. 11:

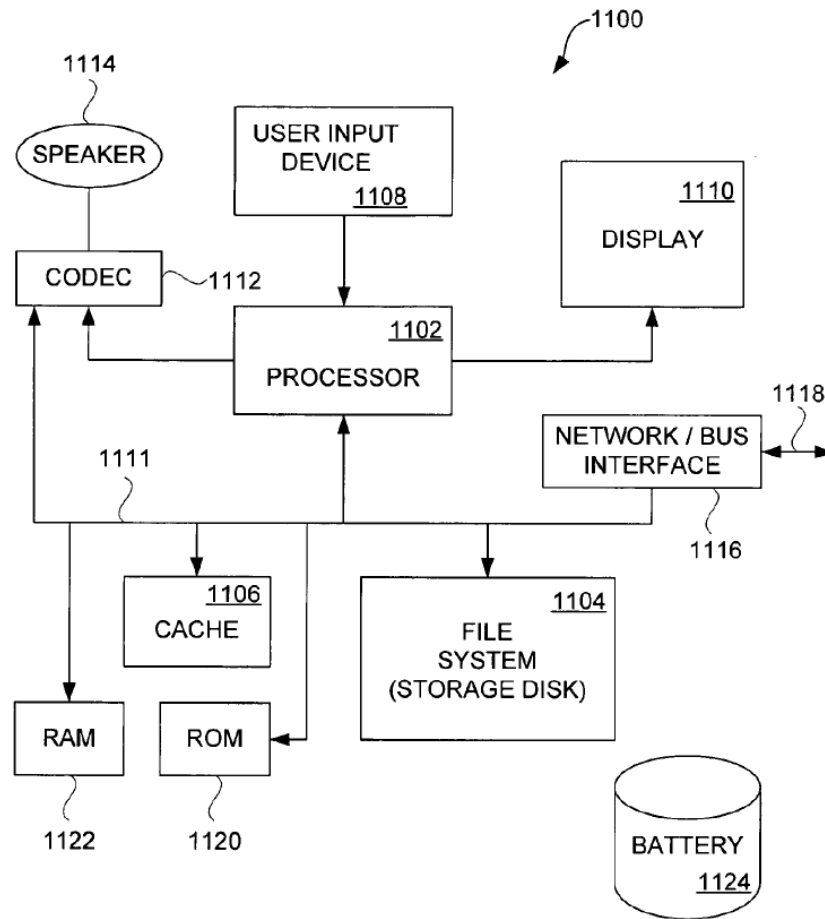


FIG. 11

See also Ex. 1008, 4:64-65, 5:26-39; Ex. 1003, ¶¶132-134.

Given these teachings, it would have been obvious for the client device to have one or more processors, with software applications stored on non-volatile storage for implementing the combined teachings of the references, including the steps described below with respect to the challenged claims. Ex. 1003, ¶135.

[1c] obtaining a top level index file identifying a plurality of alternative streams of protected video,

The combination teaches or suggests obtaining a top level index file (e.g., a Media Presentation Description “MPD” file) identifying a plurality of alternative streams (e.g., representations) of protected (e.g., partially encrypted) video. Ex. 1003, ¶¶136-140. Chen teaches a Media Presentation Description (“MPD”) file that identifies a plurality of alternative representations of a video, which are streamed to the client device. See Ex. 1006, ¶¶[0219] (“[A] ‘Media Presentation Description’ is provided that comprises information regarding the representations of the media presentation that are available to the client. **Representations may be alternatives in a sense that the client selects one out the different alternatives...**”) (emphasis added), [0227-0253] (“[A] media presentation description may contain ... Alternative Representation: Describes a list of representations.”), [0063] (“HTTP streaming is a specific type of streaming. ... The media, such as video, may be encoded at multiple bitrates to form different versions, or representations. The terms ‘version’ and ‘representation’ are used synonymously in this document.”), FIG. 5:

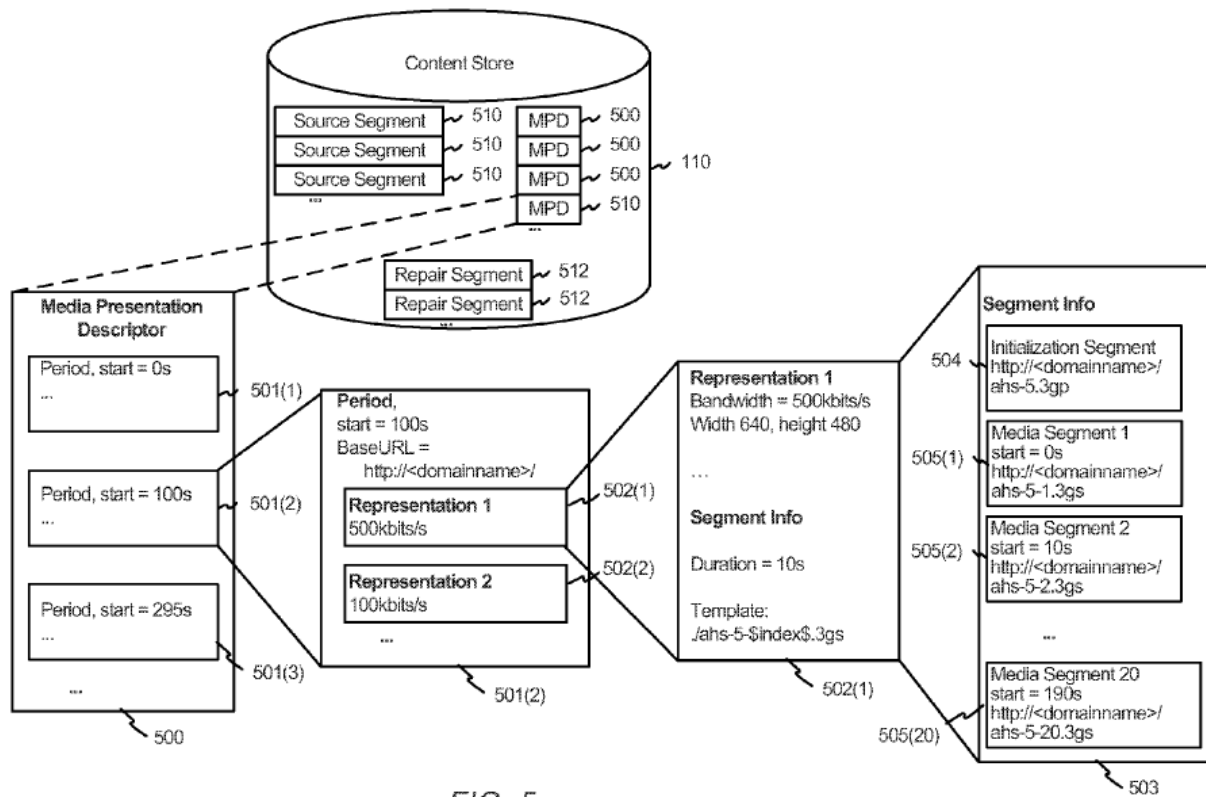


FIG. 5

See also *id.*, ¶[0225]. Chen teaches the alternative representations are encoded at different bitrates. See, e.g., Ex. 1006, ¶[0063]. Thus, a POSITA would have understood that Chen teaches obtaining a top level index file identifying a plurality of alternative streams of protected video. Ex. 1003, ¶138.

A POSITA would have been motivated to utilize Chen’s MPD teachings, with alternative streams, because that was typical and efficient for adaptive streaming, as admitted by the ’588 patent. See Ex. 1001, 2:11-14; Ex. 1003, ¶¶136-137; Ex. 1006, ¶¶[0226], [0223] (representations “typically” are alternatives); §§I, II. Its widespread use meant a substantial amount of infrastructure and support facilitated the use of alternative streams as taught by

Chen. Ex. 1003, ¶137; Ex. 1006, ¶¶[0246-0253 (discussing XML support). And Chen explains the use of alternative representations in its embodiments. *See* Ex. 1006, ¶¶[0121], [0226]; Ex. 1003, ¶137.

Hurst further renders this limitation obvious by teaching a plurality of alternative streams of protected video, including alternative representations of a content file encoded at varying bitrates and quality. *See* Ex. 1008, 6:32-8 (“...FIG. 2 b is a schematic block diagram illustrating one embodiment of a plurality of streams 202 having varying degrees of quality and bandwidth...”), FIG. 2b:

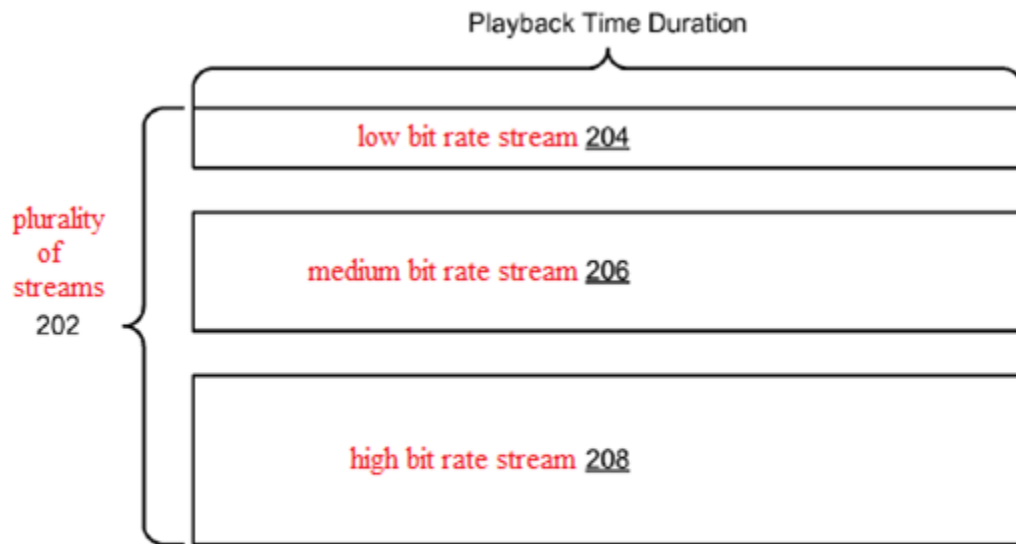


FIG. 2b
(ANNOTATIONS ADDED)

Ex. 1003, ¶139. A POSITA would have recognized Hurst’s teachings to be analogous to Chen’s and been motivated to apply them to reinforce Chen’s teachings. Ex. 1003, ¶139.

The combination teaches that each of the alternative video streams is protected, through encryption, as explained below. *See* §VI.A.2.[1d-1e].

[1d] wherein each of the alternative streams of protected video includes partially encrypted video frames
--

The combination teaches or suggests that each of the alternative streams (e.g., representations) of protected video includes partially encrypted video frames. Ex. 1003, ¶¶141-143. Lindahl teaches that portions of each frame are encrypted, while other portions remain unencrypted, providing an illustration using a “representative” frame. *See* Ex. 1007, ¶[0055] (“...The frame 520 is divided into blocks ... FIG. 5C illustrates a representative block 540 ... **only a portion of the block 540 is encrypted.** ...”), ¶[0056] (“Partial encryption and the use of encryption parameters is discussed in additional detail below ...”), FIGS. 5A-5C:

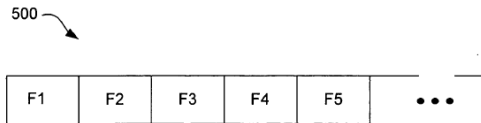


FIG. 5A

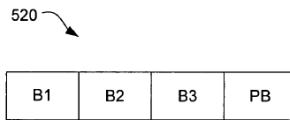


FIG. 5B

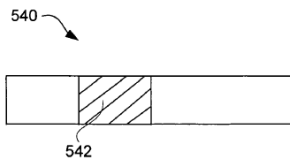


FIG. 5C

Ex. 1003, ¶141.

A POSITA would have found it obvious to partially encrypt all frames because (a) Lindahl describes a partially-encrypted frame as “a representative one of the frames from the media file...” (*see* Ex. 1007, ¶¶[0055], [0060]), and (b) it would have protected each frame, rendering each frame and the video as a whole effectively unwatchable for unauthorized users. Ex. 1003, ¶142.

[1e] that are encrypted using a set of common keys comprising at least one key,

Lindahl and Hurst both teach or suggests this limitation, and either is sufficient by itself to render the limitation obvious. Ex. 1003, ¶¶143-151.

First, Lindahl teaches or suggests that media files may be encrypted (and therefore decrypted²) with a global key. *See* Ex. 1007, ¶[0066] (“Once the media file is downloaded 708, the media file can then be decrypted 710 at the user's local machine using the cryptographic key. The global key or other cryptographic key may also be used.”); Ex. 1003, ¶144. Here, Lindahl teaches an embodiment where a “global” key is used to encrypt and decrypt all of the media files on the server. *See id.*

A POSITA would have been motivated to apply this teaching to encrypt all media files, including all alternative representations, using a global key to simplify key management particularly for content that is not particularly sensitive. Ex. 1003, ¶144. Lindahl teaches that when data “is not that sensitive ... the use of a common global key to encrypt such is adequate.” *See* Ex. 1007, ¶[0048]; Ex. 1003, ¶¶144-145. This refers to a well-known security principle: reusing a global (e.g., single) key for many videos can provide sufficient protection for content that is less sensitive, including for example (a) marketing or promotional videos, such as advertisements, commercials, etc., where a primary goal is to spread viewership of the commercials, and piracy is less of a concern, (b) non-profit materials, such

² Lindahl teaches symmetric encryption algorithms meaning the same key was used for encryption and decryption. *See* Ex. 1007, ¶[0008]; Ex. 1003, n. 3.

as educational or public awareness videos, (c) academic materials that are provided for free and (d) public access videos. Ex. 1003, ¶145. These are examples where content was provided at low cost or free, but where DRM was often still used to provide a basic level of protection to prevent crimes of convenience. *Id.* But since would-be pirates do not have much incentive to break the encryption, these usage models use DRM but do not require as rigorous security as other applications. *Id.* Thus it would have been obvious that each of the alternative streams of protected video would include partially encrypted video frames that are encrypted using a set of common keys comprising at least one global key. Ex. 1003, ¶145.

Second, the application of Lindahl to Chen leaves only a finite number of possibilities: either (1) use the same set of keys for all representations of a video, or (2) use different sets of keys. Ex. 1003, ¶146. A POSITA would have found it obvious to use the same set of keys as one of those two possibilities; recognized the design need to apply encryption to adaptive streaming because it was widely known that adaptive video streaming would require DRM; and recognized the market pressure to do so given the growth of popular commercial streaming services. *Id.*

Hurst reinforces this obviousness by teaching that the alternative “streamlets 304 of different bitrates may be encrypted with the same key.” See Ex. 1008, 18:64-67. Hurst teaches that these streamlets come from alternative streams

encoded with “varying degrees of quality and bandwidth” for “varying bit rates.”

Ex. 1003, ¶147; *see* Ex. 1008, FIG. 2b (encoding alternative low, medium, and high bitrate video streams):

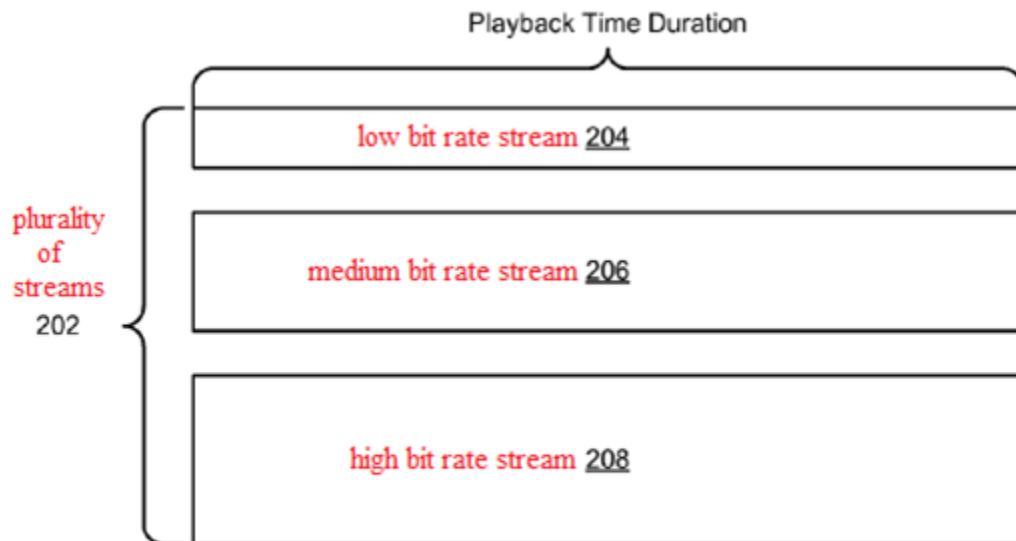


FIG. 2b
(ANNOTATIONS ADDED)

See id. 6:32-58. A POSITA would have understood Hurst’s streams to be alternative streams because they represent the same content with varying degrees of quality and bandwidth to be selected, in the alternative, based on channel bandwidth. Ex. 1003, ¶147; *see* Ex. 1008, 8:16-18, Abstract, 1:6-10, 16:61-17:58, Figure 10.

Hurst teaches that each “stream 302 divided into a plurality of source streamlets 303.” Ex. 1008, 6:59-62, FIG. 3a:

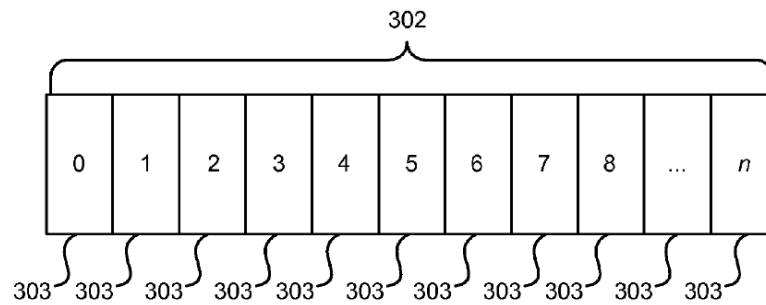


FIG. 3a

Applying Fig. 3a to 2b, Hurst illustrates the alternative low, medium and high bitrate streams divided into streamlets. *Id.*, 7:9-20, FIGS. 3b:

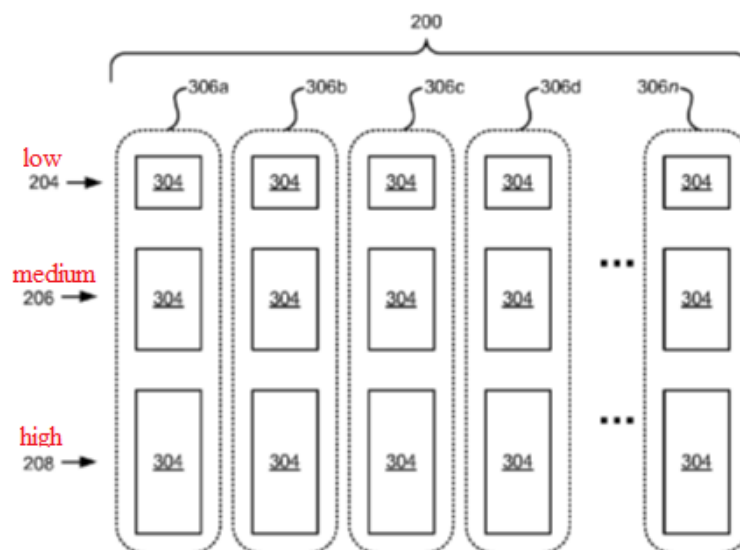


FIG. 3b

(ANNOTATIONS ADDED)

See also Ex. 1003, ¶149:

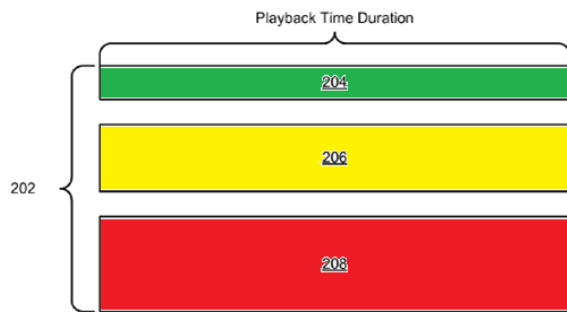


FIG. 2b
(ANNOTATIONS ADDED)

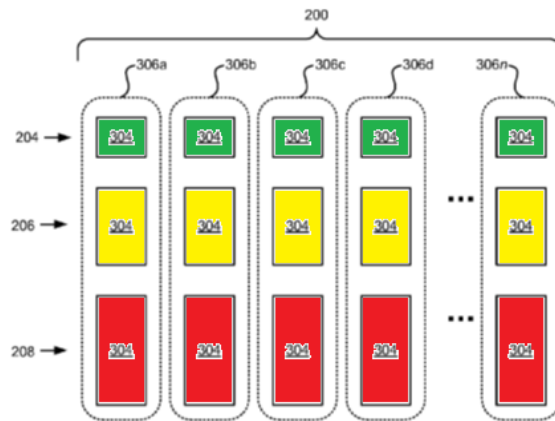


FIG. 3b
(ANNOTATIONS ADDED)

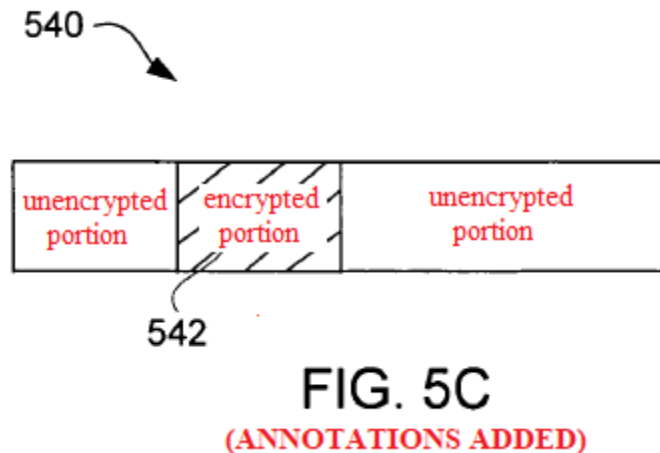
Thus Hurst teaches that each of the streamlets from the low 204, medium 206, and high 208 bitrates are encrypted using the same key. Ex. 1008, 18:64-67; Ex. 1003, ¶148.

A POSITA would have understood that Hurst's streamlets are analogous to Chen's segments because both teach video encoded at different bitrates, where the alternative streams are then divided based on time. Ex. 1003, ¶150. Both serve the same purpose for adaptive streaming. See Ex. 1006, ¶¶[0063-0065]; Ex. 1008, 7:9-20. In this context, applying Lindahl's teachings, it would have been obvious and straightforward to apply Hurst's key management teachings, with respect to the alternative streams, to Lindahl's teachings regarding cryptographic keys. See Ex. 1007, ¶¶[0064]-[0065]; Ex. 1008, 18:40-19:14; Ex. 1003, ¶150.

Thus it would have been obvious to use a common set of keys to encrypt the alternative representations of a video. Ex. 1003, ¶151.

[1f] and wherein the partially encrypted video frames contain encrypted portions and unencrypted portions of data;

The combination teaches or suggests that the partially encrypted video frames contain encrypted portions and unencrypted portions of data. Ex. 1003, ¶¶152-153. Lindahl teaches that each partially encrypted video frame contains partially encrypted blocks, which have encrypted and unencrypted portions. See *id.*; §VI.A.2.[1d]; Ex. 1007, ¶[0061] (“[E]ach partially encrypted block can have one or more portions encrypted and one or more portions unencrypted. ...”), ¶[0055] (“... only a portion of the block 540 is encrypted. More particularly, the block 540 includes an encrypted portion 542 ... those portions of the block 540 other than the encrypted portion 542 remain unencrypted.”), Fig. 5C:



Lindahl teaches “[t]he encryption parameters control the portion of a block that is to be encrypted. The remaining portions of a block that are not encrypted remain unencrypted (i.e., ‘in the clear’).” Ex. 1007, ¶[0056], ¶[0060] (“[T]he

partially encrypted blocks 600 and 620 can represent all blocks of a given frame that are partially encrypted.”), FIG. 6A-6B:

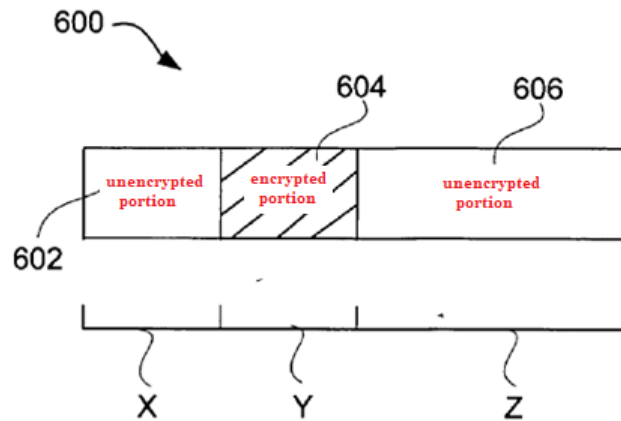


FIG. 6A
(ANNOTATIONS ADDED)

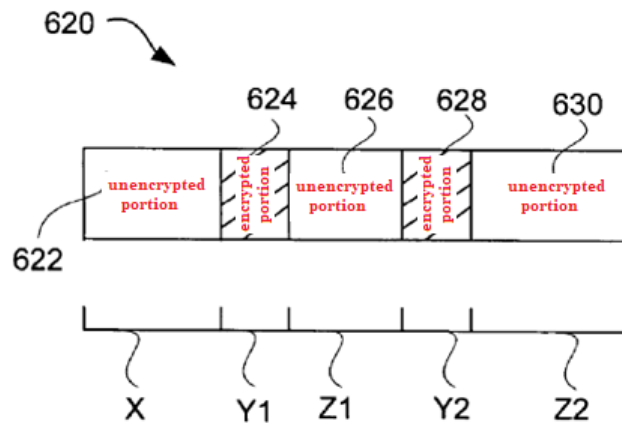


FIG. 6B
(ANNOTATIONS ADDED)

Ex. 1003, ¶153.

[1g] obtaining a copy of the set of common keys;

The combination teaches or suggests obtaining a copy of the set of common keys. Ex. 1003, ¶154-157. For example, the references teach obtaining the set of common keys to decrypt the content. Lindahl teaches that as part of playback, “the associated encryption key is obtained 706 from the selected media file” so that “the encrypted media file can be decrypted 808 using the associated encryption key.” Ex. 1007, ¶¶[0069-0070]. Similarly, Hurst teaches “[t]he DRM server 1204 is further configured to supply encryption keys to the end user upon authenticating the end user.” Ex. 1008, 18:53-55. Thus Lindahl and Hurst teach the client device obtaining the requisite keys to decrypt the selected content. Ex. 1003, ¶154.

Here, the keys associated with a particular piece of content—including its alternative streams—would have been the set of common keys (*see* §VI.A.2.[1e]), which a POSITA would have found obvious to obtain to decrypt the streams. Ex. 1003, ¶¶155-156. A POSITA would have found it obvious to decrypt the streams with the same keys used to encrypt them because the combination teaches symmetric encryption. *See* Ex. 1007, ¶¶[0008], [0049], [0066]. A POSITA would have been motivated to use symmetric keys to encrypt video data, as Lindahl teaches, because that had been the general practice for video encryption since at least the 1990s due to computational efficiency. Ex. 1003, ¶156.

Various mechanisms were known in the art for obtaining and transferring keys. *See* §I.B; Ex. 1007, ¶¶[0060, 0079]; Ex. 1003, ¶157. Thus the combination renders this limitation obvious. *Id.*

[1h] detecting streaming conditions for the playback device;

The combination teaches or suggests detecting streaming conditions for the playback device (e.g., the client device). Ex. 1003, ¶¶158-160. Chen teaches the client device³ detecting when available bandwidth so that a stream with a matching data rate can be selected. *See* Ex. 1006, ¶¶[0081] (“This selection may be based on any number of criteria, including matching of the media data rate to the available bandwidth, even when the available bandwidth is changing over time...” [0121], [0063-0065] (“...The client may switch to different data rates based on available bandwidth. ... Triggers for switching may include **buffer occupancy and network measurements**, for example. ...”) (emphasis added); Ex. 1003, ¶158. It would have been obvious that Chen teaches buffer occupancy and network measurements for detecting the available bandwidth for streaming—thereby detecting streaming conditions for the playback device. Ex. 1003, ¶160. A POSITA would have been motivated to apply these teachings because detecting streaming conditions,

³ Chen refers to the client device as a receiver. *See* Ex. 1006,, [0201] (“the client or receiver”); Ex. 1003, ¶158.

including through buffer occupancy and network measurements, was a basic, well-known and practiced function in adaptive streaming, as admitted by the '588 patent. Ex. 1003, ¶159; *see* Ex. 1001, 1:59-67. Thus Chen renders this limitation obvious. Ex. 1003, ¶160.

[1i] selecting a stream from the plurality of alternative streams of protected video based on the detected streaming conditions;
--

The combination teaches or suggests selecting a stream from the plurality of alternative streams (e.g., representations) of protected (e.g., partially encrypted) video based on the detected streaming conditions (e.g., network measurements/available bandwidth). Ex. 1003, ¶¶161-164. Chen teaches a client device selecting blocks from alternative representations, encoded at different bitrates, based on available bandwidth, which was detected in the previous step. *See* Ex. 1006, ¶¶[0080-0081] (“[T]he division of the media presentation into blocks allows each block of encoded media data to be **selected from a set of multiple available encodings. This selection may be based on any number of criteria, including matching of the media data rate to the available bandwidth ...**”) (emphasis added), [0063-0065] (“The client may switch to different data rates based on available bandwidth. ... Triggers for switching may include buffer occupancy and network measurements, for example. ...”); Ex. 1003, ¶¶161-162. Chen teaches adaptive streaming, where video is encoded at multiple bitrates to

form different representations, and a receiver selects a representation, based on its bitrate, to match the bitrate with the available bandwidth. Ex. 1003, ¶¶162-163; see Ex. 1006, FIG. 5 (illustrating different bitrates for alternative representations):

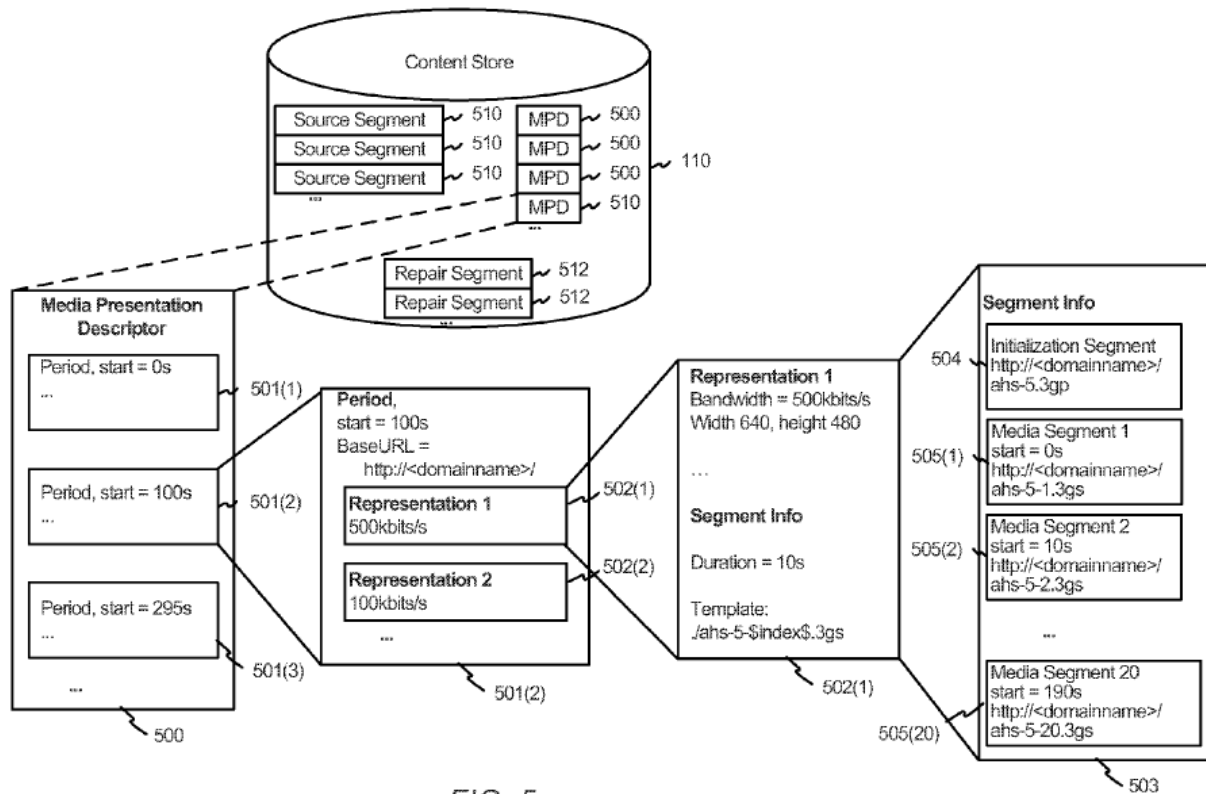


FIG. 5

A POSITA would also have found it obvious that each representation of video would be protected through encryption, including by partially encrypting its video frames. See §VI.A.2.[1d]; Ex. 1003, ¶164.

[1j] receiving a container index that provides byte ranges for portions of the selected stream of protected video within an associated container file;
--

The combination teaches or suggests receiving a container index (e.g., segment index/map) that provides byte ranges for portions (e.g., fragments/blocks) of the selected stream (e.g., representation) of protected (e.g., partially encrypted) video within an associated container file (e.g., segment). Ex. 1003, ¶¶165-174. Chen teaches that after a representation is selected, the client accesses associated segments for that representation. Ex. 1003, ¶165; *see* Ex. 1006, ¶¶[0063-0064] (“Each version or representation may be broken into smaller pieces... to form segments. Each segment may then be stored on a web server or CDN as a separate file. **On the client side, requests may then be made, using HTTP, for individual segments** that are seamlessly spliced together by the client..”) (emphasis added), Fig. 5:

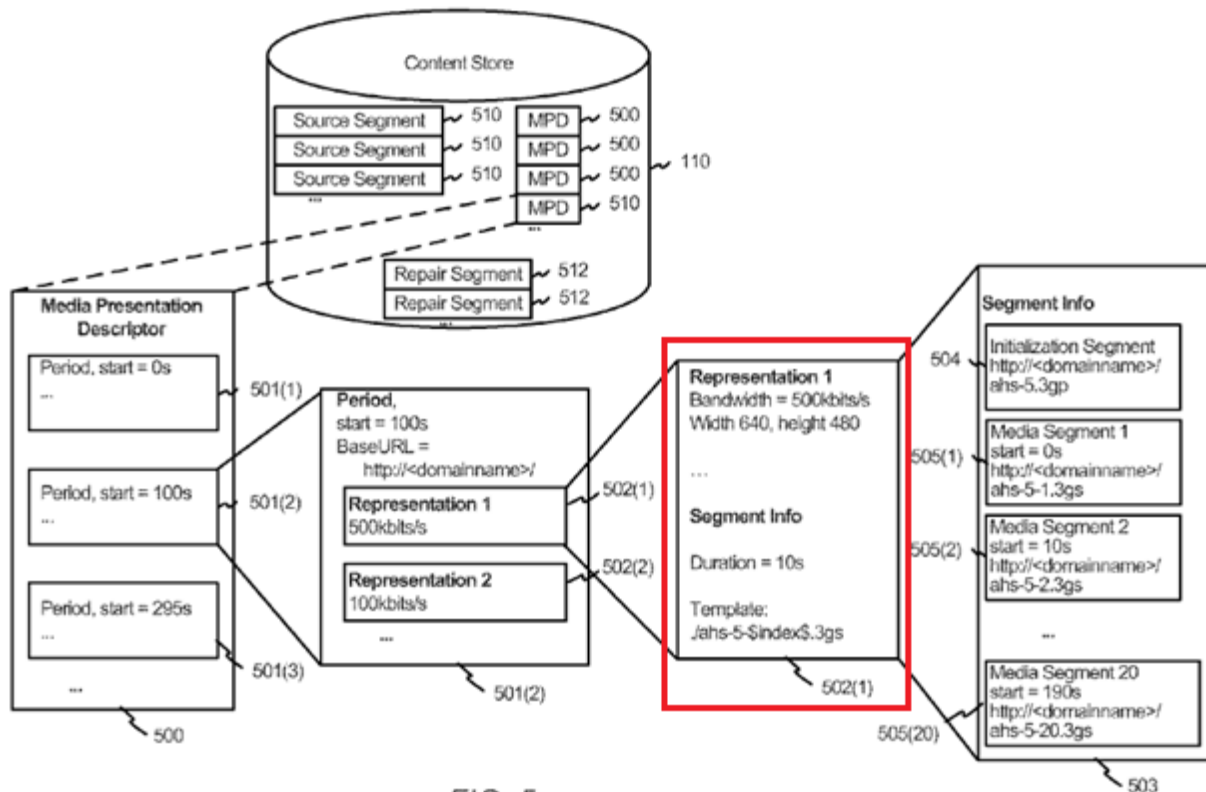


FIG. 5
(ANNOTATIONS ADDED)

Chen teaches that the segments are files. See Ex. 1006, ¶¶[0072] (“The terms ‘file’ and ‘segment’ are used synonymously in this document unless otherwise specified.”); Ex. 1003, ¶165. Specifically, the segments are container files that follow 3GPP or ISO file formats. *Id.*; see Ex. 1006, ¶[0119] (“The 3GPP File Format or any other file based on the ISO base media file format, such as the MP4 file format or the 3GPP2 file format, **may be used as the container format for HTTP streaming.**”) (emphasis added), [0175]. A POSITA would have been motivated to apply these teachings because storing video in container files—particularly the widely-used MP4 file format—was common for video streaming

before 2011, as admitted by the '588 patent. Ex. 1003, ¶165; see Ex. 1001, 2:12-34.

Chen teaches that each container file (segment) contains an index that provides byte ranges for portions of the selected stream of protected video. Ex. 1003, ¶¶165-166; see Ex. 1006, ¶¶[0119] (“**A segment index may be included in each segment to signal time offsets and byte ranges**, such that the client can download the appropriate pieces of files or media segments as required.”) (emphasis added), [0148] (“To improve bandwidth usage, segments can include an index file as metadata, where **the index file maps the byte ranges** of individual blocks with the time ranges that the blocks correspond to, called segment indexing or segment map.”) (emphasis added), [0120] (“The “moov” box may be placed at the start of the file, followed by fragment offset data, also referred to as a segment index, **containing offset information in time and byte ranges** for each fragment or at least a subset of fragments in the containing segment.”) (emphasis added), [0123], [0222] (providing “efficient mapping between time segments of the media presentation and byte ranges within the file”), [0129] (“Segment metadata may include **byte range** versus time information for requests within a segment and identification of Random Access Points (RAPS) or other seek points, where some or all of this information may be part of a **segment indexing** or segment map.”) (emphasis added), Fig. 27:

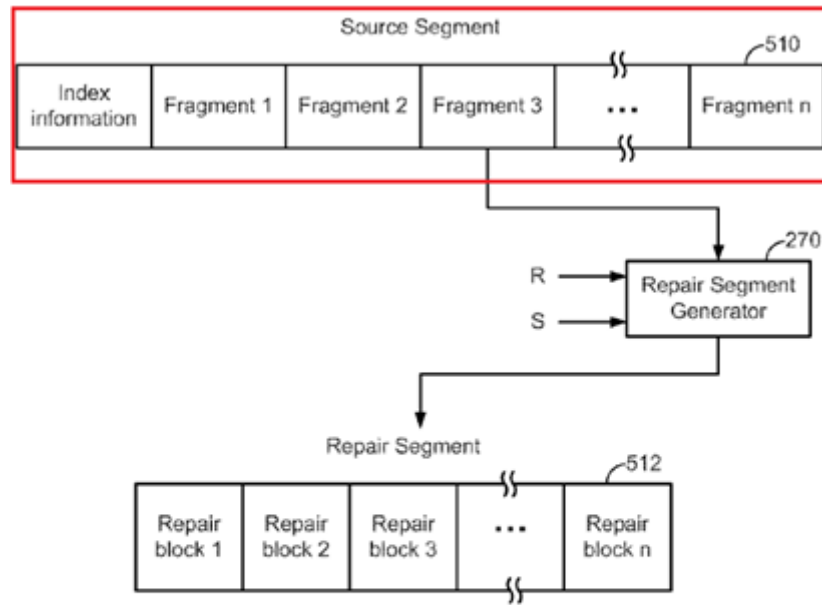


FIG. 27
(ANNOTATIONS ADDED)

FIG. 9(a):

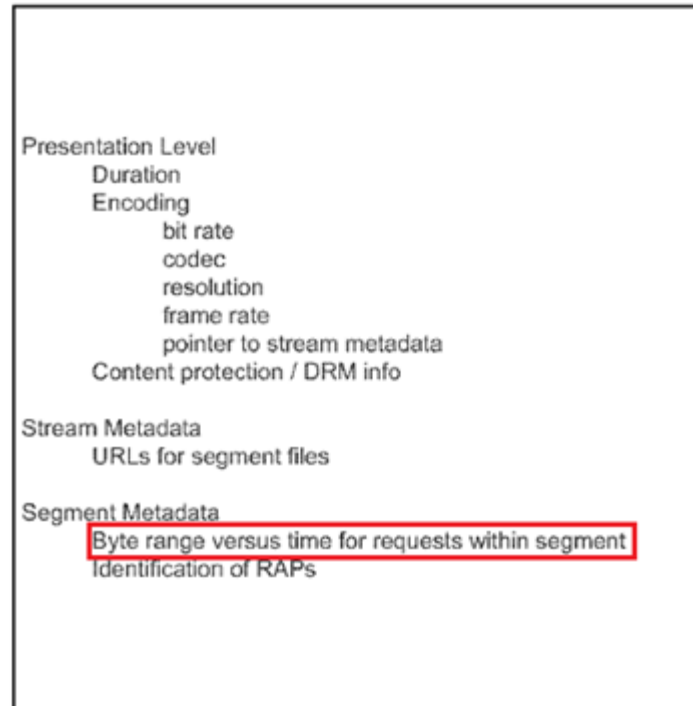


FIG. 9(a)
(ANNOTATIONS ADDED)

Thus, Chen teaches the playback device receiving a container index that provides byte ranges for portions of the selected stream of protected video within an associated container file. Ex. 1003, ¶166.

Chen's segment index provides byte ranges to facilitate adaptive streaming over HTTP, where HTTP requests include a byte range for requesting a portion of a file. See Ex. 1006, ¶¶[0111], [0119]; Ex. 1003, ¶167. Chen's technique often accesses segments in the middle of the file because the playback device jumps between streams based on bandwidth conditions, and the user may skip ahead in

the video. See Ex. 1006, ¶¶[0064], [0086]; Ex. 1003, ¶168. Thus a POSITA would have been motivated to provide byte ranges in a segment index to facilitate HTTP range requests for specific fragments within a segment file, as Chen teaches. Ex. 1003, ¶¶167-168; see Ex. 1006, ¶¶[0119-0123, 0129, 0147-0150, 0201], FIG. 6. Chen teaches the segment index/map is typically located at the beginning of the segment file, and it teaches advantages that would have motivated a POSITA to include segment indices inside segment files, including the ability to download segment maps quickly for determining offsets and byte positions of fragments. See Ex. 1006, ¶[0123]; Ex. 1003, ¶172.

Chen teaches the segment index is a “Segment Index Box (‘sidx’)” according to the ISO MP4 base file format. See Ex. 1006, ¶¶[0158-0189], [0119]; Ex. 1003, ¶169. In its Amended Complaint, PO alleged this limitation [1j] is met “because, for example, MP4 container files ... include ***container indexes in the form of an sidx box, which provides byte ranges*** for portions of a stream of protected video within the container file. ...” See Ex. 1015, ¶429 (emphasis added).

Additionally, A POSITA would also have found it obvious for the video to be protected through encryption, including by partially encrypting its video frames. See §VI.A.2.[1d]; Ex. 1003, ¶174.

[1k] requesting portions of the selected stream of protected video based on the provided byte ranges;
--

The combination teaches or suggests requesting portions (e.g., fragments) of the selected stream (e.g., representation) of protected (e.g., partially encrypted) video based on the provided byte ranges. Ex. 1003, ¶¶175-181. Chen teaches that, after selecting a representation and receiving a segment index for an associated segment, as explained above, the client requests a fragment of the associated segment based on byte ranges provided by the segment index, which is stored as metadata at the beginning of the file. See Ex. 1006, ¶¶[0070] (“The client may then request an initial portion of the file to learn the media metadata ... together with movie fragment times and byte offsets. **The client may then issue HTTP partial get requests to obtain movie fragments as required.**”) (emphasis added), [0147] (“Multiple fragments may be concatenated into a single segment and **clients may issue requests that identify the specific byte range within the segment that correspond to the required fragment** or subset of the fragment.”) (emphasis added), [0517] (“The client may, for example, initially request the ‘tidx’ or ‘sidx’ box from the beginning of the media segment using byte range requests. **By use of the ‘tidx’ or ‘sidx’ boxes, segment timing can be mapped to byte ranges of the segment. By continuously using partial HTTP requests, only the relevant**

parts of the media segment need be accessed, for improved user experience and low start-up delays.”) (emphasis added), FIG. 5:

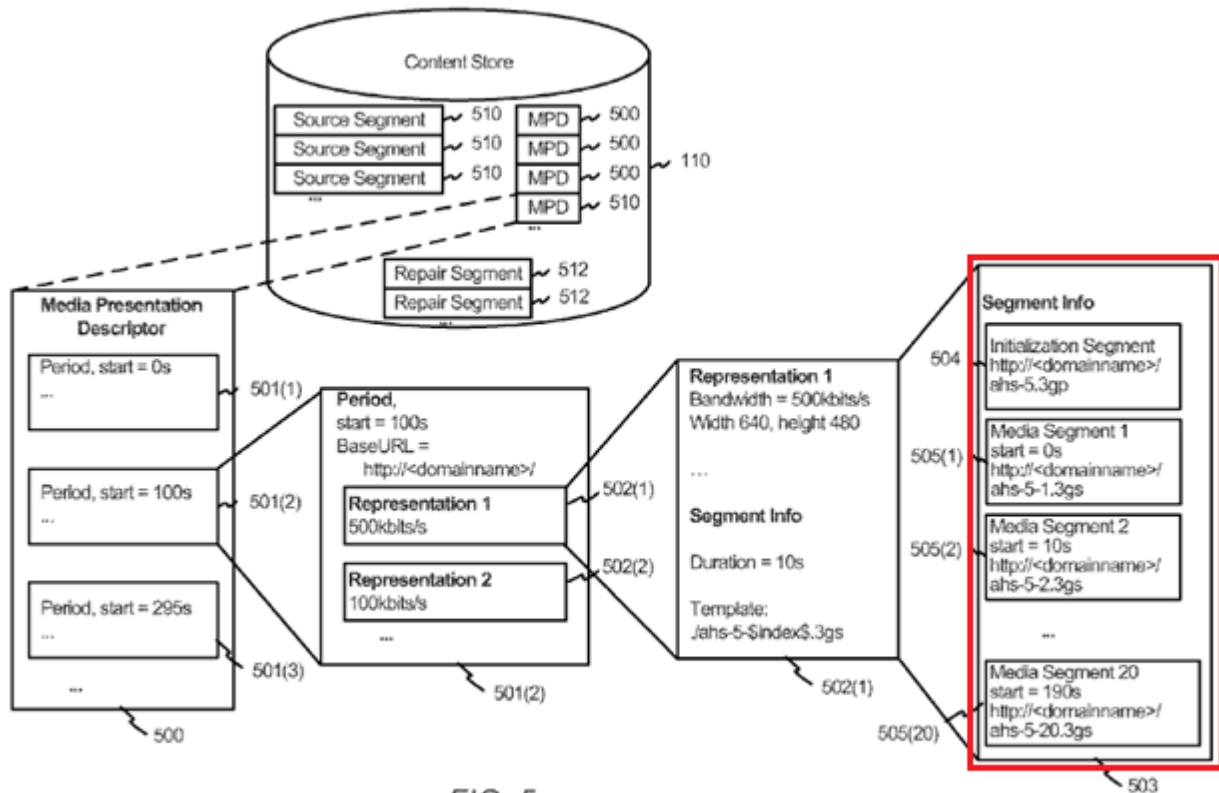


FIG. 5
(ANNOTATIONS ADDED)

Ex. 1003, ¶175. Chen teaches requests for fragments (or portions of fragments) are made using partial HTTP requests. See, e.g., 1006, ¶¶[0070], [0074], [0083]; Ex. 1003, ¶176.

Chen teaches that fragments are blocks and that both are portions of a segment (e.g., container file). See Ex. 1006, ¶¶[0074] (“The terms “block” and “fragment” are used synonymously in this document unless otherwise

specified...”), [0147]. Chen teaches that a segment file contains a container index at the front, followed by a series of fragments. *See id.*, FIG. 27:

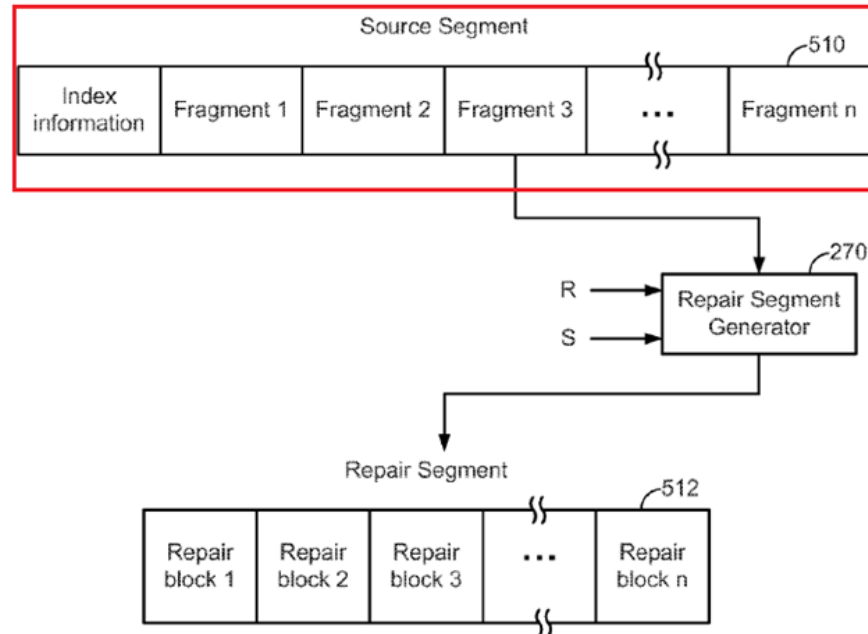


FIG. 27
(ANNOTATIONS ADDED)

Ex. 1003, ¶¶177-178; *see also* Ex. 1006, ¶[0071] (“In case that the segment format is based on the 3GP file format, then segments contain non-overlapping time slices of the movie fragments, called ‘time-wise splitting.’”), Figs. 6, 27. Therefore, Chen teaches that fragments/blocks are portions of a segment. Ex. 1003, ¶178. When Chen teaches requesting a portion of a segment, Chen teaches requesting a portion of the selected stream because a segment is a portion of the selected stream (e.g., representation). Ex. 1003, ¶178; §VI.A.2.[1i]-[1j].

Chen further explains that its request for fragments of a segment are based on byte ranges provided by a segment index. *See, e.g.*, Ex. 1006, ¶¶[0129] (“Segment metadata may **include byte range** versus time information **for requests within a segment** and identification of Random Access Points (RAPs) or other seek points, where **some or all of this information may be part of a segment indexing or segment map.**”) (emphasis added), [0517], [0529] (“[E]ach HTTP byte range request is for a portion of the byte range indicated in the segment map for the fragment.”), [0530]. A POSITA would have been motivated to follow these teachings because, e.g., they facilitate HTTP partial get requests. Ex. 1003, ¶179. Thus, the combination teaches using HTTP to request portions of the selected stream of protected video based on the provided byte ranges in the segment index (sidx). *Id.*

In its Amended Complaint, PO alleged this limitation [1k] is met by “us[ing], for example, the sidx box to make HTTP byte range requests for content.” Ex. 1015, ¶430; Ex. 1003, ¶180.

Hurst further renders this obvious with analogous teachings where “streamlet requests may comprise requesting pieces of any streamlet file...” Ex. 1008, 14:6-11, 15:10-14; Ex. 1003, ¶181.

Additionally, a POSITA would also have found it obvious for the video to be protected through encryption, including by partially encrypting its video frames.

See §VI.A.2.[1d]; Ex. 1003, ¶181. Thus the combination renders this limitation obvious. *Id.*

[11] locating encryption information that identifies encrypted portions of frames of video within the requested portions of the selected stream of protected video;

The combination teaches or suggests locating encryption information (e.g., encryption parameters) that identifies encrypted portions of frames of video within the requested portions of the selected stream (e.g., representation) of protected (e.g., partially encrypted) video. Ex. 1003, ¶¶182-187. Lindahl teaches using encryption parameters to identify encrypted portions. See Ex. 1007, ¶¶[0056] (“Partial encryption and the use of encryption parameters is discussed in additional detail below with reference to FIGS. 6A and 6B. The encryption parameters control the portion of a block that is to be encrypted...”), FIGS. 6A-6B:

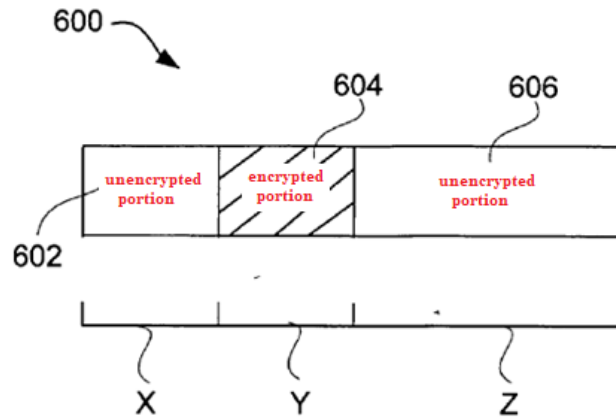


FIG. 6A

(ANNOTATIONS ADDED)

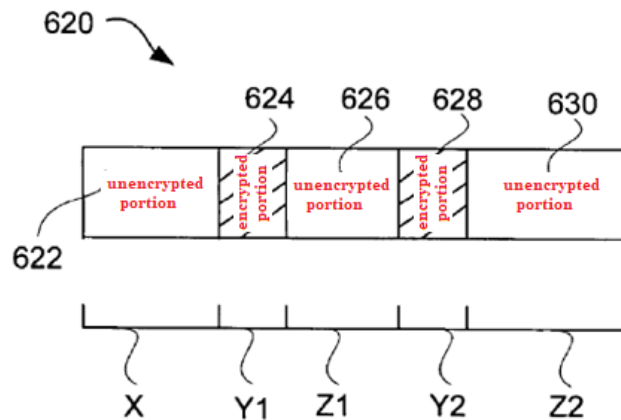


FIG. 6B

(ANNOTATIONS ADDED)

Ex. 1003, ¶182.

Lindahl teaches encryption parameters, including X, Y and Z, that identify encrypted portions of video frames. See Ex. 1007, ¶¶[0056-0059]. For the encrypted portion of Figure 6A, X provides its location, Y denotes its size, and Z

denotes the size of the unencrypted portion at the end of the block. *See id.*

¶¶[0057], [0059]; Ex. 1003, ¶183. Figure 6A contains one encrypted portion per block; Lindahl teaches another embodiment in FIG. 6B with more than one encrypted portion of a block. Like before, X, Y, and Z identify the encrypted portions of the video frame, with variable Y expanded into Y1 and Y2, and Z expanded into Z1 and Z2 to indicate the size and positions of the additional encrypted portions. *See* Ex. 1007, ¶[0058]; Ex. 1003, ¶184. Lindahl teaches that a video frames are divided into blocks. *See* Ex. 1007, ¶¶[0055, 0060-0061].

Both embodiments satisfy the challenged claims because both include partially encrypted video frames with encrypted portions and unencrypted portions of data. *See* §VI.A.2.[1d, 1f]. Both teach encryption information (e.g., encryption parameters) that identify encrypted portions of frames of video within the requested portions of the selected stream of protected video, as explained above. Ex. 1003, ¶185. A POSITA would have been motivated to use encryption parameters, as taught by Lindahl, because it provides flexibility and facilitates different levels of security by controlling the amount and portions of each frame that is encrypted. *See* Ex. 1007, ¶¶[0057-0058]; Ex. 1003, ¶186.

Lindahl teaches that the encryption parameters are “retrieved” to decrypt the media. *See* Ex. 1007, ¶[0072] (“After a media file to be decrypted has been identified 902, encryption parameters associated with the media file are retrieved

904....”); Ex. 1003, ¶187. It would have been obvious to retrieve the encryption parameters to identify the encrypted portions of the content so that they may be decrypted efficiently. *Id.*

[1m] decrypting each encrypted portion of the frames of video identified within the located encryption information using the set of common keys;

The combination teaches or suggests decrypting each encrypted portion of the frames of video identified within the located encryption information (e.g. encryption parameters) using the set of common keys. Ex. 1003, ¶¶188-191. First, Lindahl teaches that the encryption parameters are located and retrieved to decrypt the video stream, which is divided into frames and then blocks—each block is decrypted using the encryption parameters, which identify the encrypted portions of each frame. *See* Ex. 1007, ¶[0072] (emphasis added):

After a media file to be decrypted has been identified 902, encryption parameters associated with the media file are retrieved 904. The encryption parameters can be the same parameters that were utilized with respect to the media file encryption process 400 illustrated in FIG. 4. **The encryption parameters are again utilized to decrypt the media file that was previously encrypted.** These encryption parameters can also be referred to as decryption parameters. In other words, the parameters for encryption and decryption are the same. Next, the media file is examined 906 to locate frame boundaries within the media data. Then, each frame is divided 908 into a plurality

of blocks. **Each block is then decrypted 910 in accordance with the encryption parameters.**

Id., ¶¶[0056-0059]. The encryption parameters identify the encrypted portions of frames of video. *See* §VI.A.2.[11]; Ex. 1003, ¶188. Therefore, a POSITA would have found it obvious to decrypt each encrypted portion of the frames of video identified within the located encryption information to enable playback of the content. Ex. 1003, ¶191.

The combination teaches encrypting the video frames using symmetric encryption, meaning the same keys used to encrypt data are also used to decrypt the data. *See* §VI.A.2.[1g]; Ex. 1007, ¶¶[0066], [0070] (“Next, the encrypted media file can be decrypted 808 using the associated encryption key.”); Ex. 1003, ¶189. The combination further teaches that the content was encrypted using a set of common keys. *See* §VI.A.2.[1e]. Because the video frames were symmetrically encrypted using a set of common keys, a POSITA would have found it obvious to decrypt the video frames using the same set of common keys that were used to encrypt the video frames. Ex. 1003, ¶190. Here, the set of common keys would be the media file’s “associated encryption key[s]” referenced by Lindahl. *See* Ex. 1007, ¶[0070]. Thus the combination renders this limitation obvious. Ex. 1003, ¶¶190-191.

[1n] playing back the decrypted frames of video obtained from the requested portions of the selected stream of protected video.
--

The combination teaches or suggests playing back the decrypted frames of video obtained from the requested portions of the selected stream of protected video. Ex. 1003, ¶¶192-193. Chen’s teachings “enable[] clients to request blocks of the highest media quality that may be received in time to be played out with continuity.” Ex. 1006, ¶[0127]. Chen teaches that media delivery systems send content to users, who then play back and display the content on their devices. *See id.* ¶¶[0016]-[0019]. Playing back streams for users is one of the goals of a streaming system. *Id.* ¶[0061]; Ex. 1003, ¶192. Hurst further renders this obvious by teaching that the client device plays back the content. *See* Ex. 1008, 1:15-19, 6:14-22; Ex. 1003, ¶193. Thus, it would have been obvious, after decrypting frames of video obtained from the requested portions of the selected stream of protected video, as explained in the steps above, to playback those decrypted frames because that is the purpose of Chen, Hurst, and Lindahl’s teachings. Ex. 1003, ¶193; *see* Ex. 1007, ¶¶[0005], [0011].

3. Claim 2

2[a] The playback device of claim 1, wherein the step of requesting portions of the selected stream further comprises the step of obtaining a container file containing protected video from at least one of the plurality of alternative streams,

The combination teaches or suggests the playback device and the steps of claim 1 and further teaches or suggests obtaining a container file (e.g., segment) containing protected (e.g., partially encrypted) video from at least one of the plurality of alternative streams (e.g., representations). Ex. 1003, ¶¶194-196. Chen teaches that after a representation is selected, the client obtains associated segments for that representation, which are stored as container files on the server. *See* §VI.A.2.[1i]-[1j]; Ex. 1006, ¶¶[0063-0064]. Chen teaches that “[t]he terms ‘file’ and ‘segment’ are used synonymously” and further teaches that a segment can be “obtained as a result of a file download protocol request, including for example an HTTP request.” Ex. 1006, ¶[0072]; Ex. 1003, ¶194.

A POSITA would have found it obvious and been motivated to obtain a container file using a file download protocol request, such as an HTTP request, because HTTP requests were widely utilized for adaptive streaming solutions to request and obtain files. *See* §VI.A.2.[1k]; Ex. 1001, 2:1-11; Ex. 1003, ¶¶194-196; Ex. 1006, ¶[0133] (“Most CDNs will, however, serve range requests from cache if the data is available. It may therefore be advantageous to have some portion of the

client requests be for a whole segment file. These requests can later be cancelled if necessary.”). It was obvious to obtain a container file with the requested video in response to an HTTP request, because HTTP GET requests most frequently return files. *See* §I.A; Ex. 1003, ¶196.

It would have been obvious to a POSITA for the container file to contain protected video from at least one of the plurality of alternative streams. *See* §VI.A.2.[1i-1k]; Ex. 1003, ¶¶194-196. Thus, the combination teaches the playback device and further renders this limitation obvious. Ex. 1003, ¶196.

[2b] where the container file also includes encryption information that identifies portions of frames of video that are encrypted and
--

The combination teaches or suggests that the container file also includes encryption information (e.g., encryption parameters) that identifies portions of frames of video that are encrypted. Ex. 1003, ¶197. Lindahl teaches encryption parameters that identify encrypted portions of video frames (*see* §VI.A.2.[11]) and can “be provided on a per file basis” including “in the media file itself.” Ex. 1007, ¶[0053]. A POSITA would have been motivated to do so because, for example, it provides flexibility, particularly for adaptive streaming situations, to account for varying bitrates for each stream, given that the encryption parameters are provided in units of bytes. *See id.*, ¶¶[0052] (“frame size can vary... depending on target resolution.”), [0057]; Ex. 1003, ¶197. For example, this would allow for

encryption of corresponding regions of blocks in alternative streams with different bitrates. Ex. 1003, ¶197.

Furthermore, Lindahl teaches that “the encryption parameters could be provided on a per frame or block basis, e.g., with each frame or block potentially having a different set of parameters.” Ex. 1007, ¶[0053]. A POSITA would have been motivated to apply this teaching to improve the balance between computation and security by tailoring the parameters based on the data, such as encrypting higher percentages of I-frames. Ex. 1003, ¶197. Because encryption parameters vary between frames or blocks within a video, it would have been obvious to store the encryption parameters “in the media file itself.” See Ex. 1007, ¶[0053]. Thus, the combination teaches that the container file also includes encryption information (e.g., encryption parameters) that identifies portions of frames of video that are encrypted. Ex. 1003, ¶197.

[2c] a reference to at least one key from the set of common keys to utilize in accessing the encrypted portions of the frames of video.
--

The combination teaches or suggests a reference to at least one key from the set of common keys to utilize in accessing the encrypted portions of the frames of video. Ex. 1003, ¶¶198-199. Lindahl teaches locating the appropriate cryptographic key for decryption and playback. See Ex. 1007, ¶[0083] (“[I]t should be understood that since the media items are secured (i.e., encrypted), the

play module 1032 must first locate the appropriate encrypted cryptographic key before a given media item can be played.”).

Thus Lindahl teaches that the cryptographic key must be identified, and a POSITA would have been motivated and found it obvious for Lindahl’s encryption parameters to include a reference to this key, for example a key ID, to indicate which key was used to encrypt which block, because Lindahl’s use of symmetric encryption requires the same key to be used for decryption. *See* Ex. 1007, ¶[0008]; §VI.A.2.[1g]; Ex. 1003, ¶198. Lindahl teaches that blocks are decrypted using one or more cryptographic keys, in accordance with the encryption parameters; thus it would have been obvious to store a reference to the cryptographic key in the encryption parameters so that it may be used for decryption. *See* Ex. 1007, ¶[0072]; Ex. 1003, ¶198. Using key IDs would also have facilitated key management, allowing servers with many files to quickly identify and confirm the encryption keys for each file, which is why it is a common practice to include key IDs with encrypted files for video streaming DRM applications. Ex. 1003, ¶198.

Additionally, the combination teaches encrypting the alternative streams using a common set of keys comprising a plurality of keys. *See* §VI.A.2.[1e], §VI.A.12. This includes using more than one key to encrypt and decrypt blocks within the video stream. *See id.*; Ex. 1007, ¶¶[0054], [0072]. When more than one key is used to encrypt a video file, a POSITA would have been motivated and

found it obvious for Lindahl's encryption parameters to include key IDs to indicate which key should be used to decrypt which block—otherwise the decryption process would need to guess at which key was used for each block, which can become prohibitively inefficient. Ex. 1003, ¶199; *see* Ex. 1007, ¶¶[0008], [0072]; §VI.A.2.[1g]. Thus, the combination renders this limitation obvious. Ex. 1003, ¶199.

4. Claim 3

3	The playback device of claim 1, wherein the located encryption information comprises a reference to the start of an encrypted block of data.
----------	---

The combination teaches or suggests the playback device of claim 1 and further teaches that the located encryption information (e.g., encryption parameters) comprises a reference to the start of an encrypted block of data (e.g., offset). Ex. 1003, ¶¶200, 182-187. Lindahl's encryption parameters identify encrypted portions of video frames, including variable X which is an offset to where an encrypted portion of data begins. *See* §VI.A.2.[11]; Ex. 1007, ¶¶[0059] (“This initial portion, having its size denoted by the variable X, provides an offset into the partially encrypted block 600 or 620. This offset causes the encryption portion to begin somewhere internal to the block. ...”), [0057-0058]. This offset is a reference to the start of an encrypted block of data because it points to the start of the encrypted block. Ex. 1003, ¶200; *see* Ex. 1007, ¶¶[0057-0058], FIGS. 6A-6B:

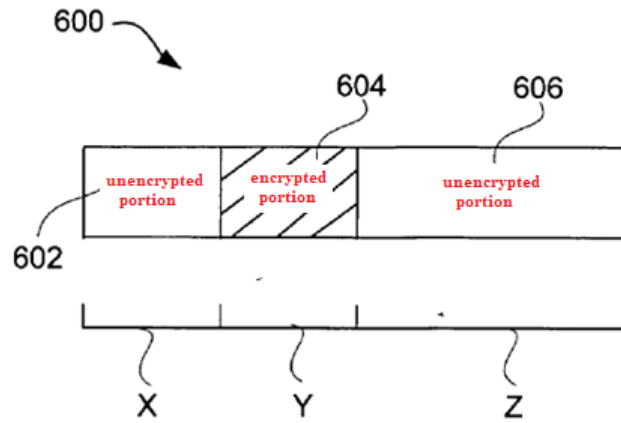


FIG. 6A

(ANNOTATIONS ADDED)

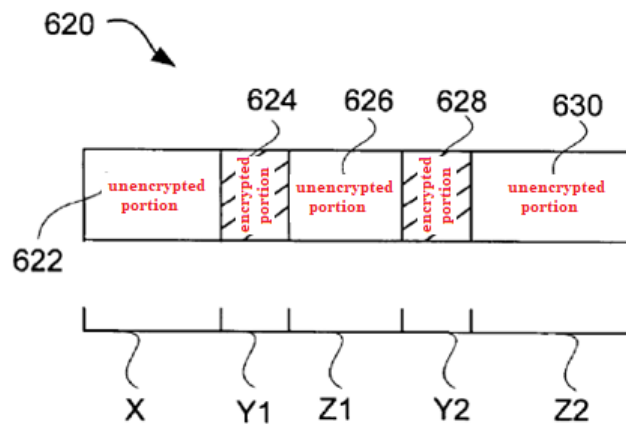


FIG. 6B

(ANNOTATIONS ADDED)

5. Claim 4

4	The playback device of claim 3, wherein the located encryption information further comprises the size of the encrypted block of data.
----------	--

The combination teaches or suggests the playback device of claim 1 and further teaches or suggests that the located encryption information (e.g., encryption parameters) further comprises the size of the encrypted block of data, including variable Y. See §VI.A.2.[11], VI.A.3.[2b]; Ex. 1007, ¶[0057] (“the size of the encrypted portion 604 is denoted by the variable Y...”), FIGS. 6A-6B; Ex. 1003, ¶¶201, 182-187.

6. Claim 5

5	The playback device of claim 3, wherein the located encryption information further comprises cryptographic information that can be utilized to access the encrypted portion of the frame.
----------	--

The combination teaches or suggests the playback device of claim 3 and further teaches or suggests that the located encryption information (e.g., encryption parameters) further comprises cryptographic information that can be utilized to access the encrypted portion of the frame. Ex. 1003, ¶202. The combination teaches the encryption parameters comprise the size and location of the encrypted portions of data, which are used to access it. *Id.*; see §VI.A.2.[11], §VI.A.3.[2b], §VI.A.4-5. And the combination teaches the encryption parameters include references, such as key IDs, to the keys in the set of common keys to be used for

decryption, which can be utilized to access the encrypted portion of the frame. See §VI.3[2c]; Ex. 1003, ¶202. Dependent claim 6 confirms that the recited “cryptographic information” can be satisfied by a reference to at least one key from the set of common keys—as explained above, this is rendered obvious by the combination. *Id.*

7. Claim 6

6	The playback device of claim 5, wherein the cryptographic information is a reference to at least one key from the set of common keys.
----------	--

The combination teaches or suggests the playback device of claim 5 and further teaches or suggests that the cryptographic information is a reference to at least one key from the set of common keys. Ex. 1003, ¶203. The combination teaches or suggests that the encryption parameters include a reference, for example a key ID, to at least one key from the set of common keys. See §VI.3.[2c]. Thus, the combination teaches or suggests the playback device of claim 5 and further teaches or suggests that the cryptographic information is a reference to at least one key from the set of common keys. Ex. 1003, ¶203.

8. Claim 7

7[a] The playback device of claim 1, wherein the application is further for causing the set of processors to perform the steps of:

The combination teaches or suggests the playback device of claim 1. See §VI.A.2; Ex. 1003, ¶204.

7[b] detecting a change in the streaming conditions;

The combination teaches or suggests this limitation. Ex. 1003, ¶205; see §VI.A.2.[1h]-[1j]. Chen teaches detecting changes in available bandwidth to select a stream with a matching data rate. See Ex. 1006, ¶¶[0081] (“This selection may be based on any number of criteria, including **matching of the media data rate to the available bandwidth, even when the available bandwidth is changing over time...**”) (emphasis added), [0063-0065], [0121]. Further, “buffer occupancy and network measurements” may trigger the client “switch[ing] to different data rates based on available bandwidth.” *Id.*, ¶¶[0063-0065]. Thus it would have been obvious for the client device to detect a change in the streaming conditions, to trigger the client to switch to representations of different bitrates. Ex. 1003, ¶205. PO admits this limitation is met, accusing in its Complaint a client that “adaptively selects the optimal stream given the instantaneous bandwidth.” Ex. 1015, ¶427; Ex. 1003, ¶205.

[7c] identifying a second alternative stream of protected video based on the detected change;

The combination teaches or suggests this limitation. Ex. 1003, ¶206; *see* §VI.A.2.[1i]. Chen teaches that video is encoded at multiple bitrates to form different representations, and the client switches to different data rates based on available bandwidth. *See id.*; Ex. 1006, ¶¶[0063-0065]. Thus, Chen teaches switching to a second alternative stream (e.g., a different representation) of protected video based on the detected change in the streaming conditions (e.g., a detected change in available bandwidth). *See* §VI.A.2.[1i]. A POSITA would have found it obvious to identify a second alternative stream in order to switch to it because the stream must be identified before it can be looked up in the MPD and requested by the playback device. Ex. 1003, ¶206.

**[7d] receiving a container index that provides byte ranges for portions of the second alternative stream of protected video within a second associated container file;
requesting portions of the second alternative stream of protected video based on the provided byte ranges;**

These limitations are the same as limitations [1j-1k] and are obvious for the same reasons. *See* §VI.A.2[1j-1k]. While claim [7d] refers to “the second alternative stream,” the obviousness analysis from claim 1 still applies because in both situations, the recited stream refers to one of the alternative streams from claim 1, and the claimed steps (“receiving a container index...” and “requesting

portions...” function similarly for each of the alternative streams. Ex. 1003, ¶207.

Claim 7’s recitation of a “second alternative stream” does not alter the analysis, other than by applying the teachings to a different one of the alternative streams.

Id.

[7e] decrypting each encrypted portion of the frames of video from the requested portions of the second alternative stream of protected video using the set of common keys; and
--

The combination teaches or suggests decrypting each encrypted portion of the frames of video identified within the located encryption information using the set of common keys. See §VI.A.2.[1m]. Claim 7 specifies this decrypting step is applied to “the requested portions of the second alternative stream of protected video.” This does not alter the obviousness analysis from claim 1, other than by applying the teachings to a different one of the alternative streams (the identified second alternative stream), because the decrypting step in claim 1 was also applied to encrypted portions of video frames from requested portions of the alternative stream of protected video requested in the preceding steps. See §VI.A.2.[1k-1m]; Ex. 1003, ¶208. Given that each of the alternative streams is encrypted in the same manner, it would have been obvious to decrypt them in the same manner. See §VI.A.2.[1e]; Ex. 1003, ¶208. Claim 7’s recitation of a “second alternative stream” does not alter the analysis, other than by applying the teachings to a different one of the alternative streams. *Id.*

[7f] playing back the decrypted frames of video from the requested portions of the second alternative stream of protected video.

This limitation is encompassed within limitation [1n] and is obvious for the same reasons. *See* §VI.A.2.[1n]. While claim 7 refers to “the second alternative stream,” the analysis from claim 1 still applies. Ex. 1003, ¶209. In both situations, the recited stream refers to one of the alternative streams from claim 1, and the claimed steps (“receiving a container index...” and “requesting portions...”) function similarly for each of the alternative streams. *Id.* The recitation of a “second alternative stream” does not alter the analysis, other than by applying the teachings to a different one of the alternative streams. *Id.*

9. Claim 8

8 The playback device of claim 1, wherein the application is further for causing the set of processors to perform the steps of: transmitting a request for content to a set of one or more content distribution servers; and receiving the content from the set of one or more content distribution servers.

The combination teaches or suggests that transmitting a request for content to a set of one or more content distribution servers (e.g., media servers) and receiving the content from the set of one or more content distribution servers (e.g., media servers). Ex. 1003, ¶¶210-212. Chen teaches requesting streaming media over a channel from one or more media servers, which a POSITA would have understood to be content distribution servers because they distribute content to

client devices. *See* Ex. 1006, ¶¶[0079] (“It is assumed in many of these examples that a client is coupled to a media server or a plurality of media servers and the client requests streaming media over a channel or a plurality of channels from the media server or the plurality of media servers.”), [0072], [0018, 0023-0024, 0063, 0079]; Ex. 1003, ¶210.

Chen teaches the use of CDNs, and a POSITA would have understood Chen’s media server teachings to be an example of this well-known content-delivery architecture, with the media servers being content distribution servers. *See* Ex. 1006, ¶[0452] (“For example this is a common method of spreading the load of requests from multiple clients across multiple physical machines. In particular this approach is commonly taken by Content Delivery Networks (CDNs).”). A POSITA would have been motivated and found it obvious to apply these teachings from Chen and use a plurality of media servers as content distribution servers because CDNs were a standard and widely-used architecture for streaming media systems. Ex. 1003, ¶211. Given Chen’s teachings, it would have been obvious for client devices to transmit a request for content to a set of one or more media servers, and then to receive the content from those servers. *Id.*

Hurst further renders this obvious. *See* Ex. 1008, 6:17-22 (“... the client module 114 may request and receive content from any of the plurality of web servers 116.”), 8:53-57 (“Additionally, since the client module 114 initiates the

request, the web server 116 is only required to retrieve and serve the requested streamlet 304. In a further embodiment, the client module 114 may be configured to retrieve streamlets 304 from a plurality of web servers 116.”). A POSITA would have recognized the web servers taught by Hurst as content distribution servers because the web servers host content and distribute the content to requesting users. *See id.*, 6:23-25; Ex. 1003, ¶212. A POSITA would have been motivated to apply the well-known content-delivery teachings from both references. *Id.*

10. Claim 9

9	The playback device of claim 8, wherein the application is further for causing the set of processors to perform the steps of: transmitting a request for cryptographic information including the set of common keys to a set of one or more DRM servers; and receiving the cryptographic information from the set of one or more DRM servers.
----------	--

The combination teaches or suggests transmitting a request for cryptographic information including the set of common keys to a set of one or more DRM servers and receiving the cryptographic information from the set of one or more DRM servers. Ex. 1003, ¶¶213-215. Hurst teaches a DRM server authenticating a user and supplying the user with appropriate encryption keys. *See, e.g.*, Ex. 1008, 18:51-55 (“The DRM server 1204 is configured to provide digital rights management capability to the system 1200. The DRM server 1204 is further

configured to supply encryption keys to the end user upon authenticating the end user.”), 19:5-8 (“The DRM server 1204 is configured to authenticate a client device and accordingly communicate the appropriate decryption key.”). It would have been obvious for Hurst’s DRM server to authenticate a user in response to a request for access to protected content (including cryptographic information needed for playback, e.g., keys, algorithms, and related parameters), which would entail the client device transmitting a request for cryptographic information, including the set of common keys that will be used to decrypt the protected content, to a set of one or more DRM servers. *See* §VI.A.2.[1m]; Ex. 1003, ¶213. After successful authentication, it would have been obvious for the DRM server to send, and for the client device to receive, the cryptographic information that will be used to decrypt the protected content. *Id.* Thus, the combination renders this limitation obvious. *Id.*

Additionally, Lindahl teaches a media distribution system that gives authorized users, such as users who purchased a media file, appropriate cryptographic information to decrypt the media file. *See* Ex. 1007, ¶¶[0039], [0044] (“... Authorized users or devices are given the appropriate cryptographic information (e.g., keys and encryption parameters) to remove the partial encryption.”). Lindahl teaches its media distribution system performs widely-known functions of a typical DRM server. Ex. 1003, ¶214. Lindahl teaches the

user's device transmitting a request, when the user purchases a media file, for cryptographic information including the set of common keys to a set of one or more DRM servers, and then receiving that cryptographic information in response. *See* Ex. 1007, ¶¶[0062-0066]. It would have been obvious that the appropriate cryptographic information would include the keys required to decrypt the content, including the set of common keys. *See* §VI.A.2.[1e, 1g]; Ex. 1003, ¶215.

11. Claim 10

10	The playback device of claim 1, wherein the application is further for causing the set of processors to perform the step of obtaining the container index from at least one file selected from the group consisting of: the associated container file containing the selected stream of protected content; and a separate file to the associated container file containing the selected stream of protected content.
-----------	---

The combination teaches this or suggests limitation, including both options of the recited group. Ex. 1003, ¶216. First, the combination teaches or suggests obtaining a container file containing a selected stream of protected content. *See* §VI.A.2.[1j], VI.A.3.[2a]. Chen teaches that the container index (e.g., segment index) is typically placed at the beginning of the segment file. *See* Ex. 1006, ¶¶[0123] (“This segment indexing may be included within the segment, typically at the beginning of the segment.”), [0120]; §VI.A.2.[1j]; Ex. 1003, ¶¶216, 172-173. Thus Chen teaches the step of obtaining the container index from the associated container file containing the selected stream of protected content. Ex. 1003, ¶216.

Further, Chen teaches or suggests providing the segment index in a separate file. *See* Ex. 1006, ¶[0123] (“The segment index may also be provided in a separate index segment or file.”); Ex. 1003, ¶¶216, 173. Thus Chen also teaches obtaining the container index from a separate file to the associated container file containing the selected stream of protected content. Ex. 1003, ¶216.

Both techniques were widely-known in the art, and a POSITA would have been motivated to apply them depending on the situation. *Id.* For example, a POSITA would have been motivated to (a) place the container index in the associated container file to quickly obtain the index when the video is requested (*see* Ex. 1006, ¶[0123]), or (b) place the container index in a separate file to provide an overall index of multiple container files to facilitate switching between streams. Ex. 1003, ¶216. Thus, the combination teaches the playback device of claim 1 performing the additional limitations of claim 10. *Id.*

12. Claim 11

11	The playback device of claim 1, wherein the common set of keys comprises a plurality of keys.
-----------	--

The combination teaches or suggests this limitation. Ex. 1003, ¶¶217-219. Lindahl teaches using more than one key to encrypt and decrypt the blocks in a video file. *See* Ex. 1007, ¶¶[0054] (“The encryption of each block in accordance with the encryption parameters is discussed in greater detail below with respect to

FIGS. 5A-5C and FIGS. 6A-6B. ... the encryption process also makes use of one or more encryption keys when encrypting each block.”), [0072] (“[T]he decryption of each block also makes use of one or more cryptographic keys...”), FIGS. 6A-6B:

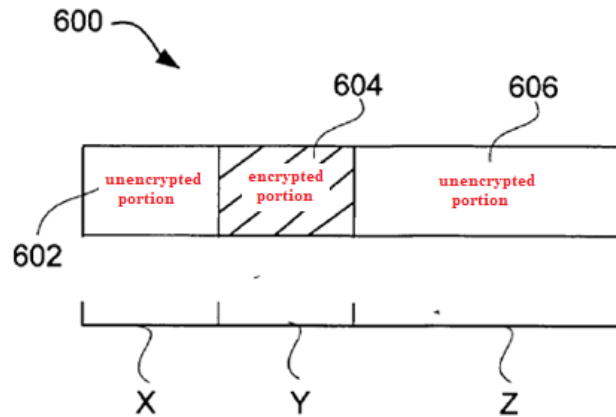


FIG. 6A

(ANNOTATIONS ADDED)

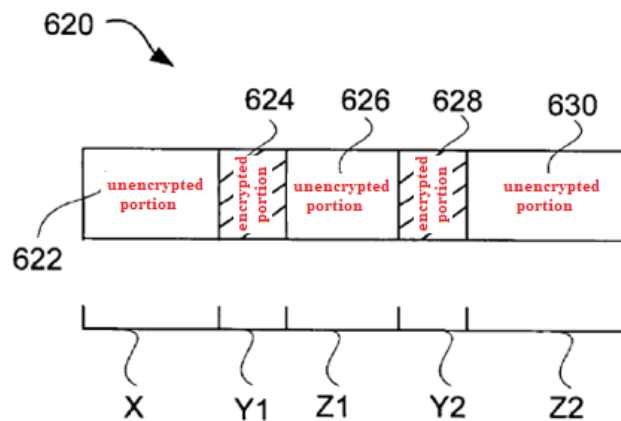


FIG. 6B

(ANNOTATIONS ADDED)

Ex. 1003, ¶217.

It would have been obvious for the different encrypted portions of FIG. 6B to be encrypted using different cryptographic keys because FIGS. 6A-6B teach one or more encrypted portions for each block, and Lindahl teaches that encryption for FIGS. 6A-6B “makes use of one or more encryption keys when encrypting each block.” *See* Ex. 1007, ¶¶[0054, 0072], FIGS. 6A-6B; Ex. 1003, ¶218. Thus Lindahl teaches using more than one encryption key to encrypt a video. *Id.*; *see* Ex. 1007, ¶¶[0054, 0057-0059]. A POSITA would have been motivated to apply these teachings because using more than one encryption key for a video increases security—each key is used to encrypt a smaller amount of data, making it more difficult to access the media, and even if one key is compromised, it only decrypts a part of the video stream. Ex. 1003, ¶218. It was common in the art to use multiple keys in this manner, going back years if not decades, with the keys stored and organized in key tables. *See id.*; §I.

Hurst teaches or suggests that the alternative streams of content, such as video files, may be encrypted in the same manner. *See* Ex. 1008, 18:62-67; Ex. 1003, ¶219. For example, if a stream is encrypted with one key, Hurst teaches encrypting each of the alternative streams with the same key. *See* Ex. 1008, 18:62-67; §VI.A.2.[1e]. Here, Lindahl teaches encrypting a video using a plurality of keys, which in the context of this combination would have suggested encrypting a

video stream using a plurality of keys. When combined with Lindahl, Hurst's teachings would have suggested encrypting each of the alternative streams in the same manner, with that same plurality of keys. See Ex. 1007, ¶¶[0054], [0072]; Ex. 1008, Fig. 3b; Ex. 1003, ¶219. This concept is illustrated below, with an example of at least 3 keys (K1, K2, K3) used to encrypt each low bitrate stream (row 204); given Hurst's teachings, it would have been obvious to use that same set of keys to encrypt the medium (row 206) and high bitrate (row 208) streams for the same content. *Id.*:

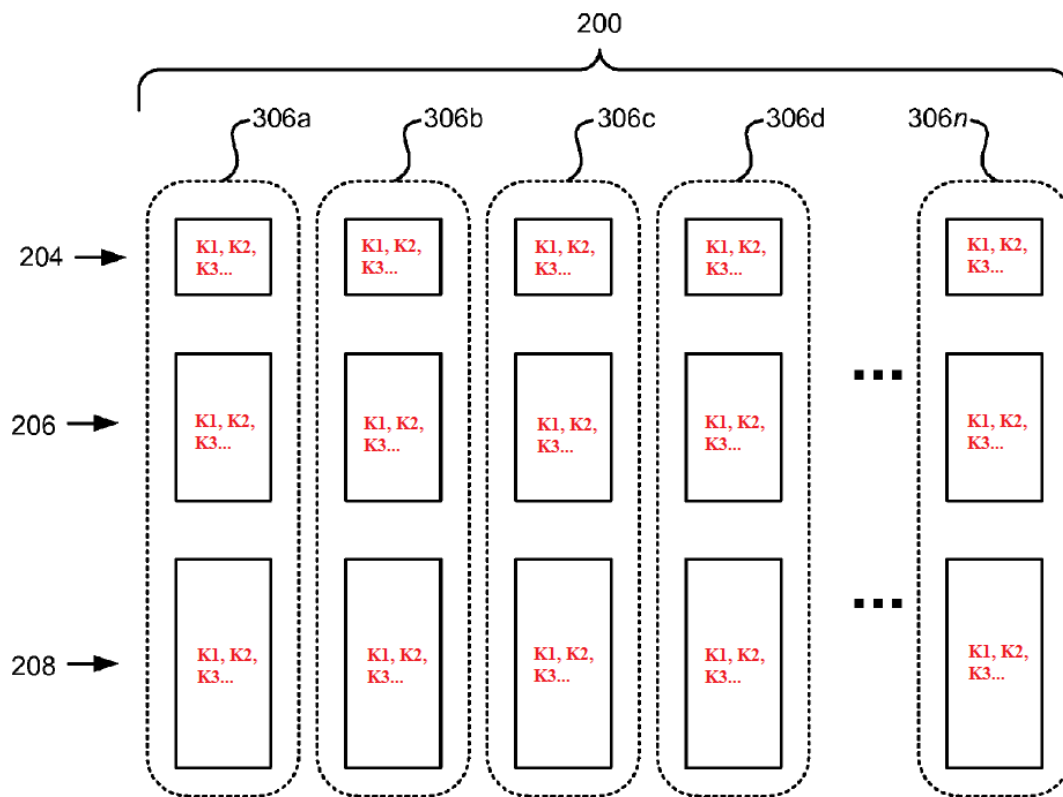


FIG. 3b
(ANNOTATIONS ADDED)

A POSITA would have been motivated to do this because using a plurality of keys, as taught by Lindahl, increases security, and applying the same set of keys to each alternative stream, as taught by Hurst, aligns with the usage model for adaptive streaming where access is typically granted to the alternative streams as a group. Ex. 1003, ¶219; *see* §VI.A.1. Furthermore, encrypting the alternative streams in the same manner is one of a finite number of possibilities that would have at least been obvious to try. *See* §VI.A.2.[1e]; Ex. 1003, ¶¶146, 219. Thus, given the combined teachings of the references, it would have been obvious for the partially encrypted video frames to be encrypted using a set of common keys comprising a plurality of keys. Ex. 1003, ¶219.

13. Claim 12

Claim 12 recites the same limitations as claim 1 except for “using a decoder” and is obvious for the same reasons explained above. *See* §VI.A.2.1[a]-[1n]; Ex. 1003, ¶220. Given the obviousness of an apparatus performing the steps of claim 12, explained above for claim 1, it would have been obvious to perform these steps as a method. Ex. 1003, ¶220.

The combination further teaches or suggests playing back decrypted frames of video using a decoder. Ex. 1003, ¶221. Chen and Lindahl both teach decoders in the claimed manner and render this limitation obvious. *See* Ex. 1006, ¶[0106]; Ex. 1007, ¶[0091]; Ex. 1003, ¶221. Hurst further renders this obvious by teaching

encoding of a plurality of streams. *See, e.g.*, Ex. 1008, 8:19-25, 16:3-5. Hurst teaches a media player for playing music and videos. *See* 1008, 1:15-19. It would have been obvious that playing back an encoded stream requires decoding the encoded stream. Ex. 1003, ¶222. It would have been obvious to use a decoder to play back decrypted frames of video, because this is one of the known (if not principal) functions of decoders; thus this limitation is obvious. Ex. 1003, ¶223.

14. Claims 13-22

Claims 13, 14, 15, 16, 17, 18, 19, 20, 21, and 22 recite the same limitations as claims 2, 3, 4, 5, 6, 7, 8, 9, 10, and 11, respectively, but as method claims. Each of claims 13-22 is obvious for the same reasons explained above for claims 2-11, respectively. *See* §§VI.A.3-12; Ex. 1003, ¶¶224-233. Given the obviousness of an apparatus performing these steps, explained above, it would have been obvious to perform these steps as methods. Ex. 1003, ¶224-233.

15. Claim 23

23 **The playback device of claim 1, wherein:
the container index is part of a hierarchical index; and
the application is further for causing the set of processors to perform
the step of:
obtaining a lower layer index that identifies the location of frames
within a specific requested portion of the selected stream of protected
video; and
identify partially encrypted video frames from within the specific
requested portion of the selected stream using the lower layer index.**

The combination teaches or suggests the playback device of claim 1 and further teaches or suggests that the container index is part of a hierarchical index, and the step of obtaining a lower layer index that identifies the location of frames within a specific requested portion (e.g., fragment) of the selected stream of protected video (e.g., the selected representation). Ex. 1003, ¶¶234-237. First, Chen teaches hierarchical indexing. See Ex. 1006, ¶¶[0148] (“The indexing can be simple, wherein the time and byte ranges of each block are absolute relative to the start of the file, or **they can be hierarchical**, wherein some blocks are grouped into parent blocks (and those into grandparent blocks, etc.) and the time and byte range for a given block is expressed relative to the time and/or byte range of the block's parent block.”) (emphasis added), [0164] (“A Time Index Box with targettype of Time Index Box can be used to construct **a hierarchical time index**, allowing users of the file to quickly navigate to the required portion of the index.”) (emphasis added), FIG. 7(b):

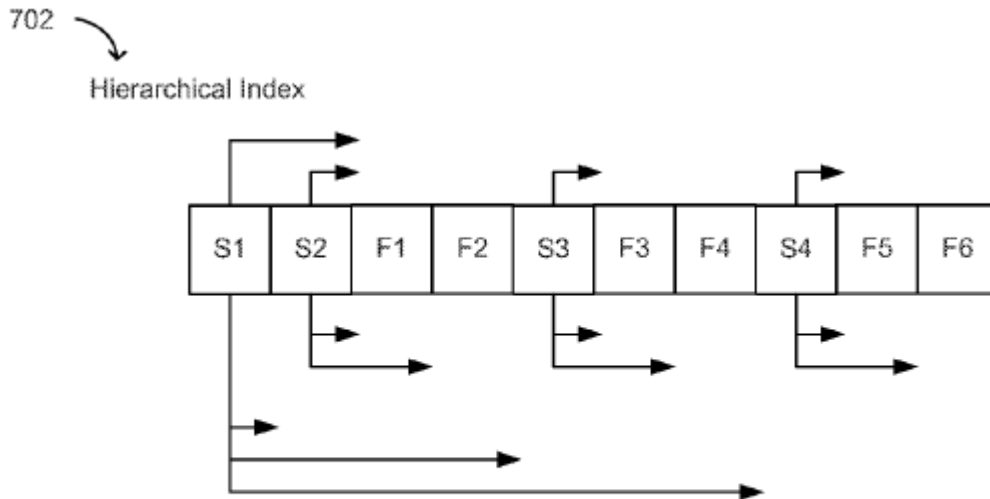


FIG. 7(b)

Ex. 1003, ¶234. Thus Chen teaches, and it would have been obvious, for Chen’s container index to be part of a hierarchical index. *Id.*

Chen further teaches a “seek point structure preferably is used to determine the block structure. ... The blocks for a given version of the content do not overlap and every frame of the content is contained within exactly one block of each version.” Ex. 1006, ¶[0197]. A POSITA would have been motivated for Chen’s hierarchal index to identify the location of frames within blocks because (a) it allows playback devices to seek to specific points, such as a scene or arbitrary location the user wants to access, and (b) it allows client devices to switch between alternative streams at any frame within the video, providing greater flexibility and allowing faster adjustments for changes to streaming conditions. *See id.*; Ex. 1003, ¶235. Furthermore, it would have been obvious for Chen’s hierarchal index to

identify the location of frames within blocks to facilitate decrypting and decoding of those frames, whose boundaries and locations typically needed to be identified, as Lindahl teaches. *See* Ex. 1007, ¶¶[0014-0016] (“As a method for decrypting media data, one embodiment of the invention includes at least the acts of... examining the media data in the media file to locate frame boundaries for the frames within the media data.”), [0052], [0072]; Ex. 1003, ¶235.

It would have been obvious for a lower layer index to contain this identification information for the location of frames because within the hierarchal index, the lower layer is closer to the frames. *See* Ex. 1006, ¶[0148]; Ex. 1003, ¶235. Given Chen’s teachings, it would have been obvious for the client device, when attempting to access a specific part of the video or switch between alternative streams, to obtain a lower layer index that identifies the location of frames within a specific requested portion of the selected stream of protected video—and then use that index to identify partially encrypted video frames from within the specific requested portion of the selected stream to access that frame on the client device. Ex. 1003, ¶¶235-236; *see* §VI.A.2.[1k].

The combination teaches encryption parameters that identifies the encrypted portions of data (*see* §VI.A.2.[1l], §VI.A.3.[2b]), and that each of the alternative streams of protected video includes partially encrypted video frames (*see* §VI.A.2.[1d]). The lower layer index identifies the location of frames within the

specific requested portion of the selected stream, and thus, it identifies partially encrypted video frames from within the specific requested portion of the selected stream. Ex. 1003, ¶237.

16. Claim 24

Claim 24 recites the same limitations as claim 23 and is obvious for the same reasons explained above. *See* §VI.A.15; Ex. 1003, ¶238. Given the obviousness of an apparatus performing these steps, explained above, it would have been obvious to perform these steps as methods.

VII. MANDATORY NOTICES AND FEES

Real Party In Interest: The real parties-in-interests here are Netflix, Inc. and Hulu, LLC. No other parties directed, controlled, or funded this Inter Partes Review proceeding (IPR).

Petitioner, Hulu, LLC, is a real party-in-interest. The following entities own ten percent or more of the stock of Petitioner Hulu, LLC: The Walt Disney Company and Comcast Corporation. Neither of the parties listed above as part-owners of Hulu controlled or funded this IPR, nor did they contribute to the preparation of this IPR in any way.

Related Matters: The '588 patent is the subject of the following civil actions and administrative proceedings:

- *DivX, LLC v. Netflix, Inc.*,
Case No. 2:19-cv-01602 (C.D. Cal.)
- *DivX, LLC v. Hulu, LLC*,
Case No. 2:19-cv-01606 (C.D. Cal.)

Lead and Backup Counsel: Pursuant to 37 C.F.R. §42.8(b)(3) and 42.10(a), Petitioner designates Harper Batts (Reg. No. 56,160) as lead counsel and Jeffrey Liang (Reg. No. 69,043) and Chris Ponder (Reg. No. 77,167) as back-up counsel, each of Sheppard, Mullin, Richter & Hampton LLP.

Service Information: Petitioner consents to service by electronic mail

addressed to:

HBatts@sheppardmullin.com,
CPonder@sheppardmullin.com,
JLiang@sheppardmullin.com, and
LegalTm-DivX-IPR@sheppardmullin.com

Petitioner's counsel may also be served by mail or hand delivery at
Sheppard, Mullin, Richter & Hampton LLP, 379 Lytton Ave., Palo Alto, CA
94301. Petitioner's counsel may be reached by telephone at (650) 815-2673 and
by facsimile at (650) 815-4668.

Fees: The Office is authorized to charge fees for this Petition to Deposit
Account 50-4561, Ref#16VF297423.

Respectfully,

SHEPPARD, MULLIN,
RICHTER & HAMPTON LLP

/Harper Batts/
Harper Batts (Reg. No. 56,160)

CLAIM APPENDIX

1[a]. A playback device for playing protected content from a plurality of alternative streams, comprising:

[b] a set of one or more processors; and

a non-volatile storage containing an application for causing the set of one or more processors to perform the steps of:

[c] obtaining a top level index file identifying a plurality of alternative streams of protected video,

[d] wherein each of the alternative streams of protected video includes partially encrypted video frames

[e] that are encrypted using a set of common keys comprising at least one key,

[f] and wherein the partially encrypted video frames contain encrypted portions and unencrypted portions of data;

[g] obtaining a copy of the set of common keys;

[h] detecting streaming conditions for the playback device;

[i] selecting a stream from the plurality of alternative streams of protected video based on the detected streaming conditions;

[j] receiving a container index that provides byte ranges for portions of the selected stream of protected video within an associated container file;

[k] requesting portions of the selected stream of protected video based on the provided byte ranges;

[l] locating encryption information that identifies encrypted portions of frames of video within the requested portions of the selected stream of protected video;

[m] decrypting each encrypted portion of the frames of video identified within the located encryption information using the set of common keys; and

[n] playing back the decrypted frames of video obtained from the requested portions of the selected stream of protected video.

2[a]. The playback device of claim 1, wherein the step of requesting portions of the selected stream further comprises the step of obtaining a container file containing protected video from at least one of the plurality of alternative streams,

[b] where the container file also includes encryption information that identifies portions of frames of video that are encrypted and

[c] a reference to at least one key from the set of common keys to utilize in accessing the encrypted portions of the frames of video.

3. The playback device of claim 1, wherein the located encryption information comprises a reference to the start of an encrypted block of data.

4. The playback device of claim 3, wherein the located encryption information further comprises the size of the encrypted block of data.

5. The playback device of claim 3, wherein the located encryption information further comprises cryptographic information that can be utilized to access the encrypted portion of the frame.

6. The playback device of claim 5, wherein the cryptographic information is a reference to at least one key from the set of common keys.

7[a]. The playback device of claim 1, wherein the application is further for causing the set of processors to perform the steps of:

[b] detecting a change in the streaming conditions;

[c] identifying a second alternative stream of protected video based on the detected change;

[d] receiving a container index that provides byte ranges for portions of the second alternative stream of protected video within a second associated container file;

requesting portions of the second alternative stream of protected video based on the provided byte ranges;

[e] decrypting each encrypted portion of the frames of video from the requested portions of the second alternative stream of protected video using the set of common keys; and

[f] playing back the decrypted frames of video from the requested portions of the second alternative stream of protected video.

8. The playback device of claim 1, wherein the application is further for causing the set of processors to perform the steps of:

transmitting a request for content to a set of one or more content distribution servers; and

receiving the content from the set of one or more content distribution servers.

9. The playback device of claim 8, wherein the application is further for causing the set of processors to perform the steps of:

transmitting a request for cryptographic information including the set of common keys to a set of one or more DRM servers; and

receiving the cryptographic information from the set of one or more DRM servers.

10. The playback device of claim 1, wherein the application is further for causing the set of processors to perform the step of obtaining the container index

from at least one file selected from the group consisting of: the associated container file containing the selected stream of protected content; and a separate file to the associated container file containing the selected stream of protected content.

11. The playback device of claim 1, wherein the common set of keys comprises a plurality of keys.

12[a]. A method for playing protected content from a plurality of alternative streams on a playback device, the method comprising:

[c] obtaining a top level index file identifying a plurality of alternative streams of protected video,

[d] wherein each of the alternative streams of protected video includes partially encrypted video frames

[e] that are encrypted using a set of common keys comprising at least one key,

[f] and wherein the partially encrypted video frames contain encrypted portions and unencrypted portions of data;

[g] obtaining a copy of the set of common keys;

[h] detecting streaming conditions for the playback device;

[i] selecting a stream from the plurality of alternative streams of protected video based on the detected streaming conditions;

[j] receiving a container index that provides byte ranges for portions of the selected stream of protected video within an associated container file;

[k] requesting portions of the selected stream of protected video based on the provided byte ranges;

[l] locating encryption information that identifies encrypted portions of frames of video within the requested portions of the selected stream of protected video;

[m] decrypting each encrypted portion of the frames of video identified within the located encryption information using the set of common keys; and

[n] playing back the decrypted frames of video obtained from the requested portions of the selected stream of protected video using a decoder.

13[a]. The method of claim 12, wherein requesting portions of the selected stream further comprises obtaining a container file containing protected video from at least one of the plurality of alternative streams,

[b] where the container file also includes encryption information that identifies portions of frames of video that are encrypted and

[c] a reference to at least one key from the set of common keys to utilize in accessing the encrypted portions of the frames of video.

14. The method of claim 12, wherein the located encryption information comprises a reference to the start of an encrypted block of data.

15. The method of claim 14, wherein the located encryption information further comprises the size of the encrypted block of data.

16. The method of claim 14, wherein the located encryption information further comprises cryptographic information that can be utilized to access the encrypted portion of the frame.

17. The method of claim 16, wherein the cryptographic information is a reference to at least one key from the set of common keys.

18[a]. The method of claim 12 further comprising:

[b] detecting a change in the streaming conditions;

[c] identifying a second alternative stream of protected video based on the detected change;

[d] receiving a container index that provides byte ranges for portions of the second alternative stream of protected video within a second associated container file;

requesting portions of the second alternative stream of protected video based on the provided byte ranges;

[e] decrypting each encrypted portion of the frames of video from the requested portions of the second alternative stream of protected video using the set of common keys; and

[f] playing back the decrypted frames of video from the requested portions of the second alternative stream of protected video.

19. The method of claim 12 further comprising:

transmitting a request for content to a set of one or more content distribution servers; and

receiving the content from the set of one or more content distribution servers.

20. The method of claim 19 further comprising:

transmitting a request for cryptographic information including the set of common keys to a set of one or more DRM servers; and

receiving the cryptographic information from the set of one or more DRM servers.

21. The method of claim 12 further comprising obtaining the container index from at least one file selected from the group consisting of: the associated container file containing the selected stream of protected content; and a separate

file to the associated container file containing the selected stream of protected content.

22. The method of claim 12, wherein the common set of keys comprises a plurality of keys.

23. The playback device of claim 1, wherein:
the container index is part of a hierarchical index; and
the application is further for causing the set of processors to perform the step of:

obtaining a lower layer index that identifies the location of frames within a specific requested portion of the selected stream of protected video; and
identify partially encrypted video frames from within the specific requested portion of the selected stream using the lower layer index.

24. The method of claim 12, wherein:
the container index is part of a hierarchical index; and
the method further comprises:
obtaining a lower layer index that identifies the location of frames within a specific requested portion of the selected stream of protected video; and identify partially encrypted video frames from within the specific requested portion of the selected stream using the lower layer index.

CERTIFICATE OF SERVICE

The undersigned hereby certifies that on February 15, 2020, true and correct copies of the foregoing Petition for *Inter Partes* Review of U.S. Patent No. 10,225,588 and Exhibits 1001-1015 were served in its entirety on the Patent Owner at the following address of record as listed on PAIR via Priority Mail Express®:

KPPB LLP

2190 S. Towne Centre Place, Ste 300
Anaheim, CA 92806

DIVX, LLC

4350 La Jolla Village Drive, Ste 950
San Diego, CA 92122

The undersigned certifies that true and correct copies of the Petition for *Inter Partes* Review of U.S. Patent No. 10,225,588 and Exhibits 1001-1015 were also sent via electronic mail to the attorneys of record for Plaintiff in the concurrent litigation matters:

Daniel L. Allender – DAllender@RobinsKaplan.com

Roman M. Silberfeld – RSilberfeld@RobinsKaplan.com

Christopher A. Seidl – CSeidl@RobinsKaplan.com

Aaron R. Fahrenkrog – AFahrenkrog@RobinsKaplan.com

Bryan J. Mechell – BMechell@RobinsKaplan.com

Emily J. Tremblay (Pro Vac Vice) – etremblay@robinskaplan.com

Mary Pheng (Pro Hac Vice) – mpheng@robinskaplan.com

Rajin S. Olson (Pro Hac Vice) – rolson@robinskaplan.com

Shui Li (Pro Hac Vice) – sli@robinskaplan.com

William E. Manske (Pro Hac Vice) – wmanske@robinskaplan.com

David M. Stein – dstein@ggtriallaw.com

SHEPPARD, MULLIN,
RICHTER & HAMPTON LLP

/Harper Batts/

Harper Batts (Reg. No. 56,160)

Attorney for Petitioner

Date: February 15, 2020

379 Lytton Ave.
Palo Alto, CA 94301
Tel: (650) 815-2673

**CERTIFICATE OF COMPLIANCE WITH TYPE-VOLUME
LIMITATION, TYPEFACE REQUIREMENTS, AND TYPE STYLE
REQUIREMENTS**

1. This Petition complies with the type-volume limitation of 14,000 words, comprising 13,983 words, as counted using the Microsoft Word software that was used to prepare this paper, excluding the parts exempted by 37 C.F.R. § 42.24(a).

2. This Petition complies with the general format requirements of 37 C.F.R. § 42.6(a) and has been prepared using Microsoft® Word 2016 in 14-point Times New Roman.

Respectfully submitted,

SHEPPARD, MULLIN,
RICHTER & HAMPTON LLP

/Harper Batts/
Harper Batts (Reg. No. 56,160)
Attorney for Petitioner

Date: February 15, 2020

379 Lytton Ave.,
Palo Alto, CA 94301
(650) 815-2673