

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Google LLC,
Petitioner,

v.

Ericsson AB
Patent Owner.

Case IPR2025-00003
Patent No. 9,313,178

PATENT OWNER'S RESPONSE

TABLE OF CONTENTS

I.	Introduction	1
II.	The ’178 Patent	3
III.	The Petition does not meet “the content information including content encryption keys. . .” as recited in all claims	6
	A. The Petition’s theory	7
	B. Peterka’s teachings contradict and refute the Petition’s theory	9
	1. Peterka’s Pay-Per-View model cannot meet the challenged claims	10
	2. Peterka’s Subscription model cannot meet the challenged claims	12
	3. The Petition’s theory also fails under Peterka’s Pay-By-Time model	14
IV.	The Petition does not meet “content encryption <i>key identifiers</i> ” as recited in all claims	37
	A. Peterka’s message sequence numbers or timestamps are <i>message identifiers</i> , not “content encryption <i>key identifiers</i> ”	39
	B. The Petition fails to demonstrate obviousness with respect to “content encryption key identifiers”	40
V.	The Petition’s expiration time theory does not meet “detecting content encryption key rotation boundaries”	43
VI.	The Petition’s notification theory does not meet the “issuing requests” requirement of all challenged claims	52
VII.	Conclusion	56

EXHIBIT LIST

Exhibit No.	Description
2001	Declaration of Kayvan B. Noroozi in Support of Motion for Admission <i>Pro Hac Vice</i>
2002	“Check.” Cambridge Dictionary, Cambridge University Press, (https://dictionary.cambridge.org/us/dictionary/english/check), accessed Jan. 16, 2025
2003	“Detect.” Cambridge Dictionary, Cambridge University Press, (https://dictionary.cambridge.org/us/dictionary/english/detect#google_vignette), accessed Jan. 16, 2025
2004	U.S. Pat. No. 10,136,452, titled “Enhanced PRACH for standalone contention based communications including unlicensed spectrum”
2005	Confidential Settlement Agreement
2006	Confidential Arbitration Agreement
2007	U.S. Patent Application Publ. No. 2014/0237243 (publication of U.S. Application No. 14/266,368 filed April 30, 2014)
2008	U.S. provisional application Ser. No. 61/500,316, filed Jun. 23, 2011
2009	U.S. Patent Application Publ. No 2012/0331293(Publication of U.S. patent application Ser. No.: 13/530,997, filed on Jun. 22, 2012)
2010	Declaration of John Villasenor, Ph.D. in support of Patent Owner's Response
2011	Deposition Transcript of Mr. Anthony J. Wechselberger (deposition taken May 29, 2025)

I. Introduction

The Petition's theories as to the '178 Patent's challenged claims rest on fundamental misinterpretations of its lead reference, Peterka, and of the scope of multiple claim limitations.

Limitations [1B] and [16C] of all challenged claims require that a single live media playback request yield a single response comprising *multiple* "content encryption keys." The Petition contends that Peterka meets that requirement by sending *multiple* messages, each containing one content key, to a client in response to a *single* client-initiated "pull" request for content. That is incorrect. Section III, *infra*. As demonstrated in detail below, Peterka consistently teaches that its "pull" model only provides *one* content key per client request. *Id.* The Petition further entirely misinterprets Peterka's Figure 7B, alleging that it shows two content keys provided in response to a single request. To the contrary, Peterka expressly teaches that the first two content keys (CK0 and CK1) in Figure 7B are provided *without any request at all*. *Id.* Content Key 2 (CK2) is the one and *only* key that clients obtain through request, which does not meet the challenged claims. *Id.*

Limitations [1B] and [16C] further require "content encryption key identifiers." The '178 Patent's specification teaches that the recited key identifiers

must be *assigned* to the content encryption keys themselves. Section IV, *infra*. The Petition, however, relies on timestamps or sequence numbers assigned to the *messages* that carry Peterka's keys, and not to the content keys, as the challenged claims require. *Id.* While Peterka's approach in that respect fails to meet the claims, the Petition itself contends that Peterka's alternative approach uniquely identifies content encryption keys. Pet. 25. Thus, the ordinary artisan would have had no reason to add the recited "key identifiers" of the challenged claims to Peterka, as the Petition separately alleges. Section IV, *infra*.

The challenged claims further require "detecting content encryption key rotation boundaries" and "issuing requests to a license server . . . to retrieve a second encryption key[.]" As shown below, the '178 Patent teaches that Figure 3 provides the process for detection of rotation boundaries. Section V, *infra*. That process necessarily requires checking for a key change notification, in addition to considering any key expiration time, in order to determine the appropriate "rotation boundary" for a current key. *Id.* The Petition's reliance on Peterka's expiration times alone thus cannot meet the "detecting" requirement of the claims. *Id.* On the other hand, the Petition's alternative reliance on Peterka's key change signaling method cannot meet the "issuing requests" limitation, because in that embodiment Peterka's

clients do not **request** a new content encryption key. Section VI, *infra*. Rather, the new content encryption key is provided to them without request, **before** a key change has been signaled. *Id.*

Accordingly, the Petition fails at multiple levels, and the Board should find all challenged claims not unpatentable.

II. The '178 Patent

United States Patent 9,313,178 (“the '178 Patent”), titled “Method and System for Secure Over-The-Top Live Video Delivery,” is directed to, among other things, “over-the-top (OTT) media delivery and more specifically key rotation for live streaming media.” Ex. 1001, 1:15-17.

The '178 Patent teaches that “[a]s content delivery models move away from streaming distribution over private networks to Web-based delivery of files over the public Internet, referred to as over-the-top (OTT) delivery, traditional content protection paradigms must be modified to support new delivery protocols, e.g., HTTP Live Streaming. For live streaming content with long or indefinite durations, use of a single encryption key for the entire duration increases the probability that the key may be compromised.” Ex. 1001, 1:18-26. “The generation of new strongly random values for use as content encryption keys and the rotation of content

encryption keys provides protection from content encryption keys being compromised in long lived streams.” *Id.* at 3:2-5.

“A history of individual content encryption keys and the order in which they were generated is maintained.” *Id.* at 3:28-29. “Each content encryption key in the history is assigned a unique identifier which is referred to herein as the content encryption key identifier.” *Id.* at 3:29-32.

“In one embodiment, when initiating playback of the stream, the client requests the current content encryption key, the next future content encryption key, and the fixed duration lifespan of the keys. The client uses the content encryption keys to decrypt the associated content.” *Id.* at 3:47-52.

“The transitioning between use of different keys is [] referred to herein as key ‘rotation.’” *Id.* at 1:63:65. “In step 310, the client 110 checks the expiration of current content key to see if a new key should be applied. There are two cases—normal period-based expiration, and an explicit key change notification received in step 308 that applies to the current segment being decrypted.” *Id.* at 11:3-7.

Claim 1 of the '178 Patent recites:

1PRE. A method for handling secure distribution of content comprising:

[1A] initiating a media playback request and receiving a playback request response;

[1B] parsing content information from the playback request response, the content information including content encryption keys, content encryption key identifiers, and content encryption key expiration times;

[1C] retrieving content and manifest files from a content delivery server;

[1D] detecting content encryption key rotation boundaries between periods of use of different content encryption keys in decrypting retrieved content;

[1E] issuing requests to a license server ahead of a key rotation boundary to retrieve a second content encryption key to be used after a content encryption key rotation boundary is reached; and

[1F] applying the second key for content decryption after the key rotation boundary is reached.

Claim 16 recites:

16PRE. A computerized device operable as a client for handling secure distribution of content, comprising:

[16A] memory operative to store computer program instructions; one or more processors; input/output interface circuitry; and interconnect circuitry coupling the memory, processors and input/output interface circuitry together,

[16B] wherein the processors are operative to execute the computer program instructions from the memory to cause the computerized device to: initiate a media playback request and receive a playback request response;

[16C] parse content information from the playback request response, the content information including content encryption keys, content encryption key identifiers, and content encryption key expiration times;

[16D] retrieve content and manifest files from a content delivery server;

[16E] detect content encryption key rotation boundaries between periods of use of different content encryption keys in decrypting retrieved content;

[16F] issue requests to a license server ahead of a key rotation boundary to retrieve a second content encryption key to be used after a content encryption key rotation boundary is reached; and

[16G] apply the second key for content decryption after the key rotation boundary is reached.

III. The Petition does not meet “the content information including content encryption keys. . .” as recited in all claims

Claim 1 of the '178 Patent, limitation [1A], recites “initiating a media playback request and receiving a playback request response.” Claim 16, limitation [16B], recites the same requirements.

Limitation [1B] further recites “parsing content information from the playback request response, the content information including content encryption keys.” Limitation [16C] recites the same.

Under limitations [1B] and [16C], “the playback request response” must comprise *multiple* “content encryption keys.” The Petition agrees. Pet. 23-25; Ex. 1003 ¶¶83, 86. Thus, as the Petition states, the challenged claims require that *one* “request” yields *one* “response” comprising *multiple* “content encryption keys” in the one response. Pet. 23-25; Ex. 1003 ¶¶83, 86.

Claim 1 limitations [1D] and [1F] further specify that the “content encryption keys” must be used/applied for decrypting content. *See* Pet. 23-24. Claim 16 limitations [16E] and [16G] recite the same.

A. The Petition’s theory

Under the Petition’s theory, Peterka’s “client-initiated requests to access media, e.g., program content,” are each the “initiating a media playback request” of limitations [1A] and [16B]. Pet. 19-20; Ex. 1003 ¶¶ 76-77.

More specifically, the Petition relies exclusively on Peterka’s “*pull*” model for key requests in order to meet the “initiating a media playback request” requirement of the claims. Pet. 12, 18.

To meet the recited “content encryption key” of the challenged claims, the Petition relies exclusively on Peterka’s “content key” because Peterka’s “content key” is used to decrypt the content, as the challenged claims require. Pet. 23-24; Ex. 2011 at 37:14-20.

14	Q. Only Peterka's content keys are the
15	content encryption keys of your theory, right?
16	ATTORNEY SARWAR: Objection.
17	Form.
18	THE WITNESS: That's correct.
19	That was sufficient to support my opinions
20	of invalidity.

The Petition explains that in Peterka, a content key is sent to the client in an ECM or EMM message for content decryption, with only *one* content key sent in each ECM/EMM message. Pet. 21-22.

The Petition then alleges that when Peterka’s client makes a *single* client-initiated “pull” request, it will receive *multiple* ECM/EMM messages in succession (each message containing one “content key”). Pet. 21-22. The Petition contends these multiple ECM/EMM messages constitute a single response that comprises the required *multiple* “content encryption keys” of limitations [1B] and [16C]. *Id.*

On those bases, the Petition states that Peterka teaches providing *multiple* “content keys” in response to a *single* “request” for live media, as claimed by the ’178 Patent. Pet. 21-22.

B. Peterka’s teachings contradict and refute the Petition’s theory

Contrary to the Petition’s assertions, Peterka’s “pull” embodiment does *not* teach providing *multiple* “content keys” in response to *one* client-initiated request for program content. Ex. 2010 ¶ 30. Rather, in all instances Peterka’s “pull” model provides *one* content key in response to each request for program content, and the Petition thus fails as to all challenged claims on that basis. *Id.*

Peterka discloses a multicasting system for distributing multimedia program content to client devices. Ex. 1005 ¶¶ 2, 6. Peterka teaches that multicasting “is similar to the existing TV broadcast” and “delivers the same content to one or more consumers at the same time,” such as “live content.” Ex. 1005 ¶ 58.

Peterka discloses three “purchase models” through which its system can provide content to clients: Pay-Per-View, Subscription, and Pay-By-Time. Ex. 1005 ¶¶ 63-82. Peterka’s message structures, key types, and other teachings are all directed to implementing its three “purchase models.” Ex. 2010 ¶ 32. A close analysis of Peterka’s teachings as to its different purchase models exposes the

Petition's erroneous characterization as to (1) *when* Peterka makes a client-initiated "pull" request for a content key; and (2) *how* Peterka responds to each such "pull" request.

1. Peterka's Pay-Per-View model cannot meet the challenged claims

In Peterka's "**Pay-Per-View**" model, the client "purchases a single event" in its entirety. Ex. 1005 ¶ 63. To access and decrypt the pay-per-view content, the client "pulls" *one* "program key," and *one* "content key," *i.e.*, "*the* content key for that pay per view program." Ex. 1005 ¶105 (emphasis added); Ex. 2010 ¶ 33. Peterka defines a "program key" or "PK" as the key-type "used to give access to an individual program event purchased using the pay-per-view option." *Id.* ¶98. Peterka defines a "content key" as the key-type "used to encrypt the content itself," *id.* ¶101, and to decrypt the content. *Id.* ¶¶106, 116, 121, 127; Pet. 11, 23-24. Under the Petition's theory, *only* the "content key"—and not the "program key"—constitutes a "content encryption key" for purposes of the challenged claims. Pet. 23-24; Ex. 2011 at 37:14-20.

14	Q. Only Peterka's content keys are the
15	content encryption keys of your theory, right?
16	ATTORNEY SARWAR: Objection.
17	Form.
18	THE WITNESS: That's correct.
19	That was sufficient to support my opinions
20	of invalidity.

Peterka expressly teaches that each “pull” request for pay-per-view content results in a response containing only *one* content key. *Id.* ¶ 105; Ex. 2010 ¶ 34. Peterka states: “Content keys, i.e., the keys used to encrypt the program content itself, are *not required to change* during a pay per view event. A client can pull a program key which is used to encrypt *the* content key for *that* pay per view program. Thus, *no other keys are required* during the pay per view event[.]” *Id.* (emphasis added). As Peterka states, that single content key comes along with a “program key” (PK) because a “program key” is the key-type used in Peterka to encrypt the content key for a pay per view event. *Id.* ¶¶ 98, 101, 105. Peterka describes this approach as “a multi-tier key hierarchy to accommodate the various purchase options such as pay per view or pay by time.” *Id.* ¶94. Thus, although Peterka teaches providing *two* keys in response to a pay-per-view request—a program key and a content key—only *one*

of those keys is relevant to the Petition's theory. Pet. 23-24; Ex. 2011 at 37:14-20; Ex. 2010 ¶ 34.

Because Peterka's Pay-Per-View model provides only *one* content key in response to *one* client-initiated "pull" request for content, Peterka's pay-per-view distribution model cannot meet limitations [1B] and [16C] and thus any challenged claim. Ex. 2010 ¶ 35.

2. Peterka's Subscription model cannot meet the challenged claims

Peterka's "**Subscription**" distribution model likewise teaches that a single content key is provided in response to a single client request. Ex. 2010 ¶ 36.

Under that model, the client "subscribes to the service," and receives access to a variety of content. *Id.* ¶68. "[I]n the initial request under the subscription model, the client *requests* the subscription and pulls *a* service key encrypted under the unique key for that client." *Id.* ¶105 (emphasis added). The client thus receives *one* "service key," *id.*, which is Peterka's key-type for the subscription distribution model. *Id.* ¶96. Along with that service key, the client also receives a single "content key" that is "encrypted under a SK [service key] for those who are subscribers to the

service.” *Id.* ¶101. Thus, the client’s initial subscription request only yields **one** content key. Ex. 2010 ¶ 36.

Peterka never teaches providing more than one content key in response to a single service key request under the Subscription model. Ex. 2010 ¶ 36. To the contrary, Peterka states that a client that has received its initial service key “may come back to the service and watch the content **without** negotiating **any** new keys” until its first service key has expired. Ex. 1005 ¶68 (emphasis added); *id.* ¶96 (“Once the client has **the** key, the client can decrypt the content of this service **until the SK expires.**”) (emphasis added). Since the client does not need to negotiate “**any** new keys” during the lifespan of its first service key, the client does not receive more than **one** content key in response to an initial subscription model request. Ex. 2010 ¶ 37. This single service key and content key pairing has “a duration from several days to several months.” *Id.* ¶96; Ex. 2010 ¶ 37.

To receive a next service key upon expiration of the initial service key, “the client **requests** the next version of the SK from the caching server.” *Id.* ¶96 (emphasis added). The next single content key would again be sent encrypted under the new service key. *Id.* ¶ 101; Ex. 2010 ¶ 39.

Thus, under the Subscription model (like the Pay-Per-View model), a client must make a new “pull” “request” for each content key it requires, and the client does not receive multiple “content encryption keys” in response to a single “pull” request as the Petition contends. Ex. 2010 ¶ 40.

Accordingly, Peterka's Subscription distribution model also cannot meet limitations [1B] and [16C], and thus the challenged claims. *Id.*

3. The Petition's theory also fails under Peterka's Pay-By-Time model

Under the Pay-By-Time model, a client purchases one or more time segments of content, and “pays for the duration of the content that was actually watched.” Ex. 1005 ¶53. “At the time the viewer negotiates the purchase options with the OCS [origin content server], the viewer may choose to select the pay-by-time option if offered. The viewer will be told what the pay periods are and what the cost of each pay period is. The viewer's selection will be included in a secure object.” *Id.* ¶ 81 (bracketing added).

Under the Pay-By-Time model, the first segment of content that the client may access is a “**free preview**,” as shown in Figs. 7A and 7B, below.

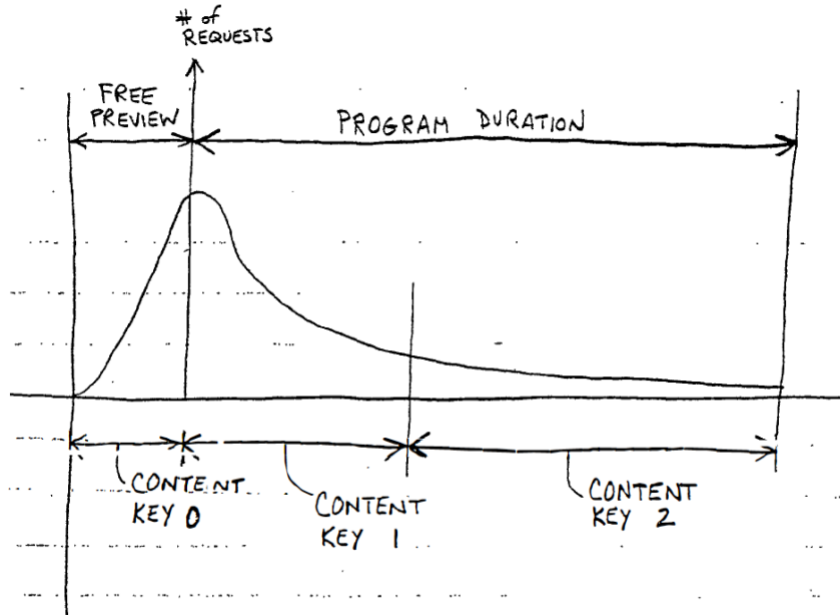


Fig. 7A

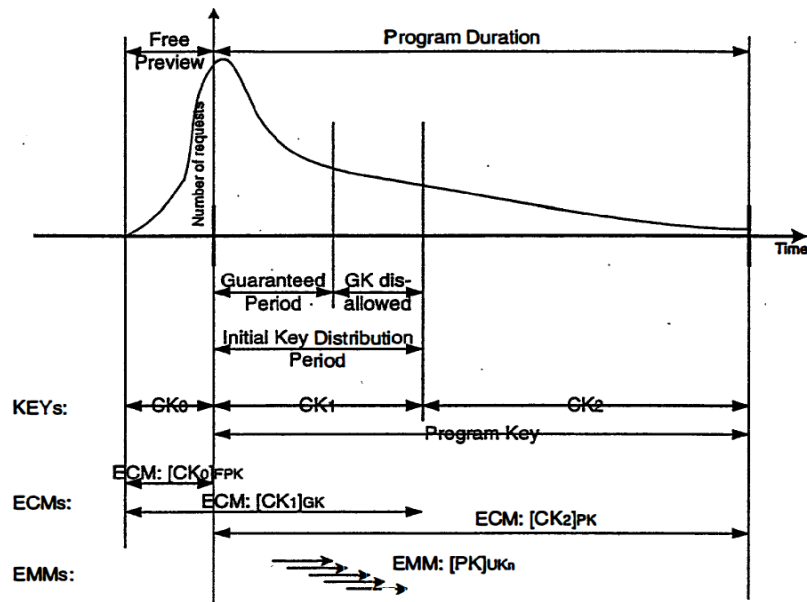


Fig. 7B

This “free preview” can be “a movie trailer prior to broadcast of the actual movie or an advertisement for the program material or a first portion of the actual movie.” *Id.* ¶ 9.

To access the free preview, Peterka teaches that the client receives any necessary content key *without* making any “request” for that content key. Ex. 2010 ¶ 44. Peterka states that the content key for the free preview “is provided *ahead of time*, for example, during provisioning.” *Id.* ¶83 (emphasis added); Ex. 2010 ¶ 44; Ex. 1005 ¶84 (“a free preview key is provided *ahead of time*, e.g., during provisioning, to a client as shown in block 420.”) (emphasis added); *id.* ¶85 (“In block 408, a content key is provided. The content key is encrypted with a free preview key as shown in block 412. The encrypted content key is then provided to the plurality of clients as shown by block 416.”); *id.* ¶103 (defining “Free Preview Key (FPK)” as a key “used to distribute the CK [content key] or the PSK [program segment key] that in turn encrypts the CK for the content during the initial free preview period. This may be a *fixed key known to all clients or distributed during provisioning.*”) (emphasis and bracketing added).

The *unrequested* content key provided for the “free preview” segment allows the client “to decrypt the program material and obtain a preview of the program

without charge.” *Id.* ¶ 83. Peterka thus provides the content key for the free preview to clients without request because clients do not request a “free preview”; the free preview is provided to them to motivate a purchase of paid content. Ex. 2010 ¶ 44.

In Figs. 7A and 7B, this first *unrequested* content key for the “free preview” segment is labeled as CK0, “content key 0,” or “Key 0.” Ex. 1005 Figs. 7A & 7B, ¶¶ 92, 93, 122. Because Peterka provides the content key for the free preview segment without request, this key (e.g., CK0 in Figs. 7A & 7B) cannot meet limitations [1B] and [16C] and thus cannot meet the challenged claims.

After the “free preview,” the next content segment a client can access in the Pay-By-Time model is an “**Initial Viewing Period.**” Ex. 1005 ¶90 (describing the “initial viewing period” as immediately following the “free preview period”); *id.* ¶93 (with respect to Fig. 7A, describing the segment after the “free preview period” as the “initial viewing period”).

In Figure 7B, this “initial viewing period” is labeled as the “initial key distribution period.” *Id.* ¶131 (with respect to Figure 7B, stating that the period defined by CK1 “is called the initial viewing period, or *equivalently*, the initial key distribution period.”) (emphasis added).

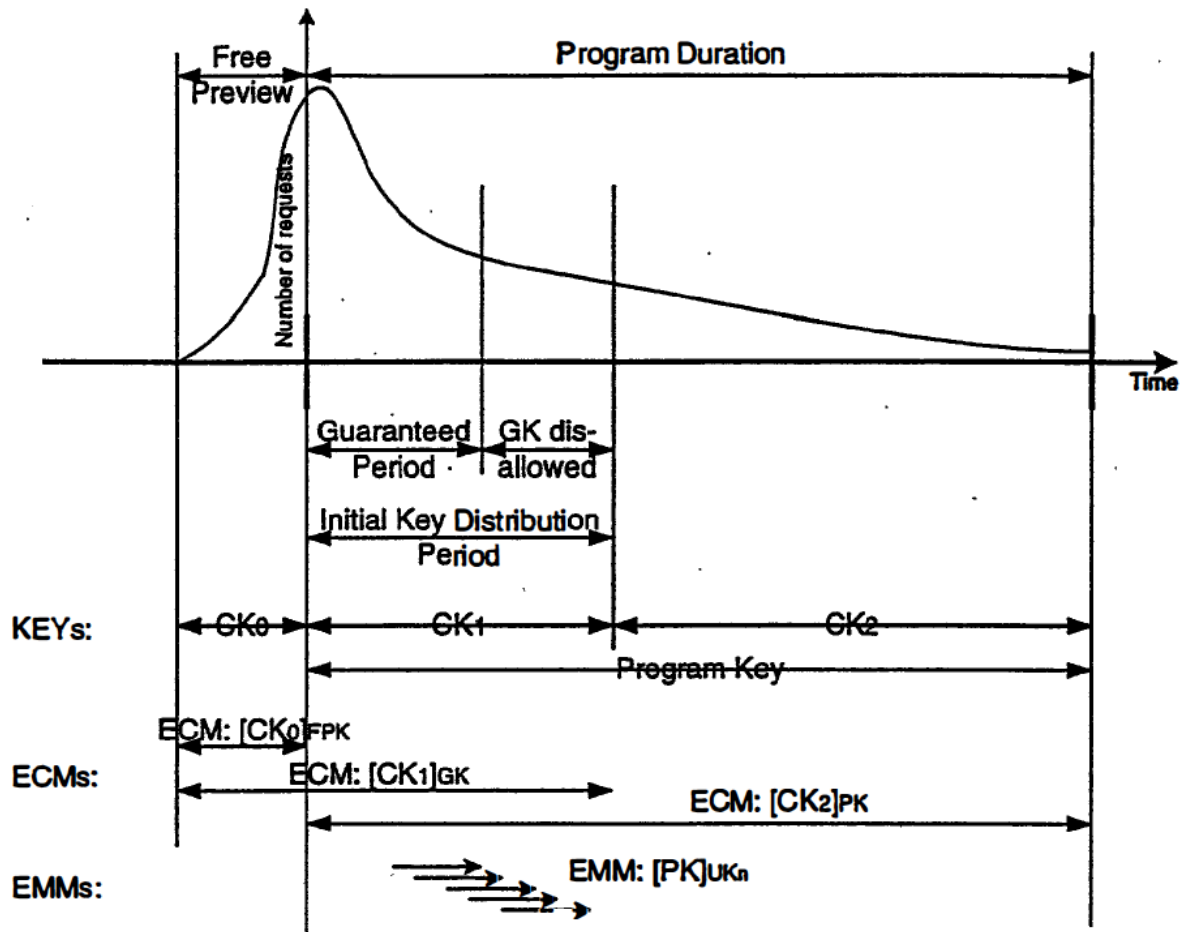


Fig. 7B

To access the “initial viewing period,” clients are provided a “group key” and a “content key” encrypted under the group key. *Id.* ¶127 (describing the key distributed to clients for accessing the “initial viewing period” as “referred to herein as a group key,” and further teaching that a “second key,” “referred to as a content

key,” is encrypted under the group key and provided to clients for decrypting the “initial viewing period” content); *id.* ¶102 (defining “Group Key” or “GK” as the “key used to distribute the CK [content key] or the PSK [program segment key] that in turn encrypts the CK for the initial viewing period.”) (bracketing added); *id.* ¶93 (with respect to Fig. 7A, stating: “Key 1 would represent a group key itself or a content encryption key encrypted under the group key used during the initial viewing period.”); *id.* ¶122 (“Fig. 7B shows an entitlement control message in which a content key is encoded with a group key (ECM:[CK1]GK).”); Ex. 2010 ¶

Like the free preview, Peterka also gives clients the content key for the “initial viewing period” *without* request, and this content key is also provided to clients ahead of time, encrypted under the Group Key. Ex. 2010 ¶ 50; Ex. 1005 ¶102 (“Clients will get the GK [group key] ahead of program events sold on PPV [pay-per-view] or PBT [pay-by-time] basis, for instance during the provisioning.”); *id.* ¶127 (“One method of implementing an initial viewing period is to distribute a common key to potential participants ahead of time, e.g., *prior to distribution* of program content. Such a key is referred to herein as a group key.”) (emphasis added); *id.* (“Since *every client* would receive a group key in such a situation, all clients

could, in theory, receive the first part of the content for free.”)¹ (emphasis added); *id.* ¶128 (“the group key is distributed *ahead of time*”); *id.* ¶59 (stating that the “group key” is “*known in advance* to registered clients”) (emphasis added); *id.* ¶127 (further teaching that the “content key” for decrypting the “initial viewing period” is also provided to clients with the group key, prior to any request for the content from the clients); *id.* ¶102 (defining “Group Key” or “GK” as the “key used to distribute the CK [content key] or the PSK [program segment key] that in turn encrypts the CK for *the initial viewing period*.”) (emphasis added); *id.* ¶93 (with respect to Fig. 7A, stating: “Key 1 would represent a group key itself or a content encryption key encrypted under the group key used during the initial viewing period.”); *id.* ¶122 (“Fig. 7B shows an entitlement control message in which a content key is encoded with a group key (ECM:[CK1]GK).”).

Thus, in Peterka’s Figs. 7A and 7B, Content Key 1 (CK1)—the content key for the initial viewing period or initial key distribution period—is provided to

¹ Peterka’s reference to “first part of the content” here refers to the initial viewing period, as the surrounding teachings in Paragraph 127 make clear; it does not refer to the earlier “free preview period,” which is not necessarily “the content,” but may instead be a trailer or advertisement for the content. Ex. 1005 ¶ 9.

Peterka's clients without request. Ex. 2010 ¶ 51. At this stage there has been no client-initiated content key request, and thus there has been no "pull." *Id.*

Peterka's request-less provisioning model for the initial viewing period content key (CK1 in Fig. 7B) is by explicit design. Ex. 1005 ¶90; Ex. 2010 ¶ 52.

Most clients are expected to request (i.e., purchase) Pay-By-Time content toward the end of the free preview period and in the beginning of the "initial viewing period" (equivalent to the "initial key distribution period" in Fig. 7B). *Id.* ¶91. Peterka shows this in Figs. 7A and 7B, where a line graph representing "number of requests" peaks at the beginning of the "initial viewing period," and remains high for the initial portion of that segment. Figs. 7A & 7B.

Peterka teaches that if clients have not already been provided the content key for the initial viewing period (i.e., CK1 in Figs. 7A and 7B), many clients who choose to purchase content will face a service interruption because there would not be "enough time to communicate with the server, and receive the key necessary to decrypt" that segment. *Id.* ¶90; Ex. 2010 ¶ 52. "The initial viewing period concept invention provides a way to enable continuous viewing despite the latency in server key distribution." Ex. 1005 ¶90. As Peterka explains, providing the content key to the "initial viewing period" to clients *without request* "allow[s] viewers to receive

the beginning of a program during this period when the server is distributing keys to those who have purchased the program material.” *Id.* ¶126.

Peterka recognizes that its request-less provisioning of the initial viewing period content key means that all clients can view that content segment for free. *Id.* ¶127 (“Since every client would receive a group key in such a situation, all clients could, in theory, receive the first part of the content for free.”); *id.* ¶133 (“a first portion of the program content is multicast to a plurality of clients at no charge”). But Peterka teaches dynamically defining the length of the “initial viewing period” based on client demand in order to mitigate that problem. *Id.* ¶ 133; Ex. 2010 ¶ 52.

Peterka’s *unrequested* content key for the “initial viewing period” segment (e.g., CK1 in Figs. 7A & 7B) also cannot meet limitations [1B] and [16C] and thus cannot meet the challenged claims. Ex. 2010 ¶ 53.

In Peterka, the first *requested* content key that a client can receive for Pay-By-Time content is the content key for the segment *after* the “initial viewing period.” *Id.* ¶93; Ex. 2010 ¶ 54. For purposes of Figures 7A and 7B, this first requested content key is labeled **Content Key 2** (CK2) or Key 2. Ex. 1005 ¶93.

Peterka teaches that “Key 2 would be the *actual* content encryption key delivered *only* to those clients that *purchased* the content. Thus clients that tune to

the content first view a free preview, and make a decision to buy. Those who make such requests use the group key [already provided] to allow their viewing experience to continue. The server would deliver Key 2 during this time, so that those viewers could continue viewing the remainder of the content in a continuous fashion.” *Id.* (emphasis and bracketed text added).

Thus, under the Pay-By-Time model, the client’s first “*request*” for content corresponds to its decision to purchase a content segment, and the first content key provided in *response* to such a purchase request is **Content Key 2** (CK2). *Id.* ¶¶ 93, 129-131; Ex. 2010 ¶ 54.

As noted, the Petition explicitly relies exclusively on Peterka’s “pull” model for content key distribution. Pet. 11-12 (“The Grounds herein rely on Peterka’s *pull model*.”) (emphasis Petition’s); Pet. 18 (“Every element of claims 1 and 16 is satisfied by or obvious over Peterka’s pull method.”).

In the context of Peterka’s Pay-By-Time distribution scheme, Peterka’s “pull” model only provides *one* content key in response to each client request. Ex. 2010 ¶ 55. Peterka teaches that in “pull” mode, “a server receives a request for *a* cryptographic key,” and “logs the request for *the* key,” *i.e.*, *one* key. Ex. 1005 ¶106 (emphasis added). “[*T*]he key and its expiration time are distributed to the client in

response to the request by the client.” *Id.* Under the pull model, “the responsibility of determining when *a* new key is required can be passed to the client,” and “program content is distributed for decryption by the client utilizing *the* cryptographic key that was requested.” *Id.* (emphases added). Peterka later reiterates: “Under a pull key distribution model, each client requests *a* new key and is individually responded to by the key distribution server.” *Id.* ¶110 (emphasis added). Peterka further states that a client “can monitor the expiration time of the program segment key and send an *individual* key update request before *the* key expires. Again, this is a way of implementing the ‘pull’” model under Peterka’s teachings. *Id.* ¶114 (emphasis added).

Thus, under Peterka’s “pull” mode, each client request receives *one* key in response to each request, and a client must make a new request each time it seeks a new key. Ex. 2010 ¶55. Peterka’s Figure 8, which depicts the operation of its “pull” mode, confirms that teaching. *Id.*

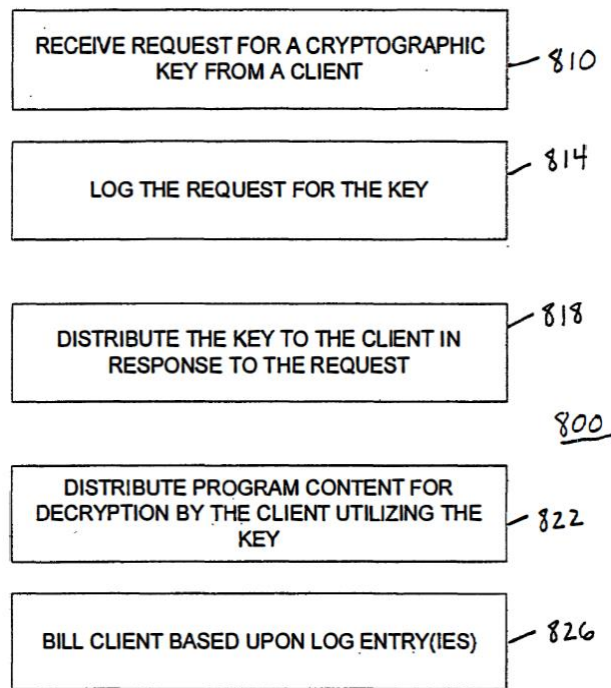


Fig. 8

A client that purchases content through Peterka's Pay-By-Time distribution scheme using Peterka's "pull" mode would thus request a *single* new content key for each next paid content segment it wishes to purchase, and would receive that *single* content key in response. Ex. 2010 ¶ 56.

Accordingly, the Petition is incorrect in its assertion that "Peterka teaches distributing *multiple* keys (e.g., multiple ECMs) to a client—e.g., for the current *and* next media segment—in response to such a request." Pet. 12 (emphasis original)

(citing Ex. 1005 ¶¶ 80, 130, Fig. 7B, Ex. 1003 ¶¶ 50-51); Pet. 20 (additionally citing Ex. 1005 ¶ 53, Ex. 1003 ¶ 77); Pet. 22, 25-26.

As shown, Peterka does **not** teach distributing multiple **content** keys (the only relevant key type) in response to a **single** “pull” request, as the Petition alleges and its theory requires. Ex. 2011 at 37:14-20; Ex. 2010 ¶¶ 57-58.

In Fig. 7B, neither the first ECM carrying Content Key 0 (CK0) encrypted under a free preview key (FPK), nor the second ECM carrying Content Key 1 (CK1) encrypted under a group key (GK), was provided in response to any **request**. Ex. 2010 ¶ 59. Rather, as shown above through extensive teachings in Peterka, both content keys were provided “ahead of time,” before any client-initiated request for content. *Id.*

The first **requested** content key in Figure 7B is Content Key 2 (CK2). *Id.* But Content Key 2 is also the **last** content key that will be provided in the embodiments of Figures 7A and 7B. Ex. 1005 ¶92 (“Content keys 1 and 2 illustrate an example where only two keys are required to decrypt the remainder of the program content.”);

Ex. 2010 ¶ 59. Content Key 2 is thus the *only requested* content key in Figures 7A and 7B. Ex. 2010 ¶ 59.

The Petition thus incorrectly relies on Figures 7A and 7B as disclosing the multiple “content encryption keys” requirement of the challenged claims, and those figures and the associated teachings cannot demonstrate unpatentability. *Id.*

In particular, the Petition is entirely incorrect in stating that Figure 7B “shows two content keys CK0 and CK1 distributed in separate ECMs for *each request* received during a free preview period [], and two content keys CK1 and CK2 distributed in separate ECMs for *each request* received during an initial key distribution period [].” Pet. 20-22 (emphasis added); Ex. 1003 ¶ 78; Ex. 2010 ¶ 63. In Figure 7B, CK0 and CK1 are both distributed *without request*, as shown. *Id.* CK2 is the only single content key distributed in response to a request, which fails to meet the claims. *Id.*

The Petition’s reliance on Peterka’s statement that “each rabbit is given keys to the current as well as the next segment,” Ex. 1005 ¶80, is similarly unavailing. There, Peterka does not teach *one* request resulting in a response comprising *two* content keys to two segments. Ex. 2010 ¶ 60. Rather, that passage explains that the key for the next segment will be provided during the current segment to avoid service

interruptions. *Id.*; Ex. 1005 ¶ 80. This is consistent with the rest of Peterka's teachings. For instance, in Figure 7B, the content key for the initial viewing period (Content Key 1) is provided during the free preview period to avoid service interruptions. Ex. 2010 ¶ 60. Although each "pull" request results in one content key, the requests and responses are timed so that the next key arrives during the period for the current key to avoid service interruptions. As a result, the client has two content keys during a given segment—the key for the current segment and the key for the next segment. Ex. 2010 ¶ 60.

The Petition's reliance on Peterka's paragraph 130 fails for similar reasons. Paragraph 130 is directed to Peterka's "guaranteed period." As shown below in Fig. 7B, the "guaranteed period" is a sub-set of the initial viewing period (or "initial key distribution period"). Ex. 1005 ¶ 130 ("the initial key distribution period is shown lasting longer than the guaranteed period").

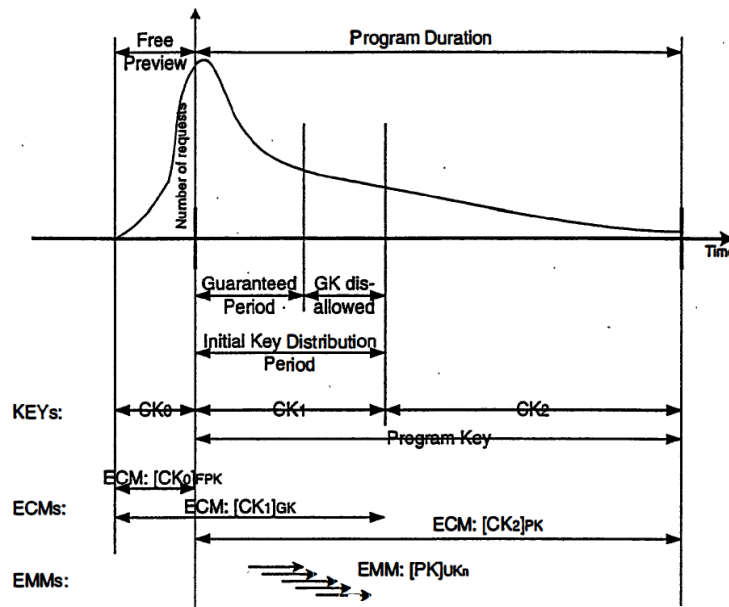


Fig. 7B

The “guaranteed period” defines the time in which clients can request *a single* content key for the *next* segment (Content Key 2 or CK2) and receive that *single* key *during* the *current* segment (the initial viewing period) to avoid service interruptions. *Id.* ¶130 (“Thus, the *next* content key will have been obtained by a purchasing user *prior* to the initial key distribution period *elapsing*.”) (emphasis added); *id.* ¶131 (“Thus, to provide a satisfying user experience, all clients who

request access to an event . . . during the ‘guaranteed period’ will be guaranteed to receive the content without interruption.”); Ex. 2010 ¶ 62.

Like paragraph 80, paragraph 130 simply teaches providing one content key for a next segment during a current segment (in response to one request) to avoid service interruptions. Ex. 2010 ¶ 62.

The Petition’s reliance on Peterka’s paragraph 53 is even farther afield. Pet. 20. That paragraph provides a very general description of Peterka’s pay-by-time distribution model; it says nothing about providing multiple content keys in response to one request, as the Petition contends. Ex. 2010 ¶ 64.

The Petition additionally misinterprets Peterka’s paragraph 105, which states that, under the “pull” model, “each client keeps track of the keys and their expiration times and actively requests new keys before the current keys expire so as to avoid service interruptions.” Ex. 1005 ¶105 (emphasis added).

That statement refers generically to “keys,” does not refer to “**content** keys,” which is the only relevant key type for the Petition’s theory. Pet. 23-24; Ex. 2011 at 37:14-20; Ex. 2010 ¶ 65.

Peterka teaches that **content** keys are encrypted under at least one **other** key corresponding to the relevant distribution model for the requested content. Ex. 1005

¶101 (Teaching that the content key can be distributed “(1) encrypted under the PSK for those viewers who select the pay-by-time option, (2) encrypted under the PK or UK for users who select the PPV option; or (3) encrypted under a SK for those who are subscribers to the service.”); Ex. 2010 ¶ 65.

Peterka's reference to requesting *multiple* “keys” in connection with its “pull” model, Ex. 1005 ¶105, refers to requesting and obtaining *one* content key and at least *one* other non-content key under which the content key is encrypted, such as a “program key” for pay-per-view content. Ex. 2010 ¶ 65; Ex. 1005 ¶105 (“A client can pull a program key which is used to encrypt *the* content key for that pay per view program. Thus, *no other keys are required* during the pay per view event[.]”) (emphasis added).

Peterka depicts this approach in Figure 7B, which shows that content key 2 (CK2) is provided in an ECM message encrypted under a program key (PK), which is the key for pay-per-view distribution. Although two keys are sent in the ECM message that transports CK2, there is only *one* content key. Fig. 7B. Accordingly, Peterka's paragraph 105 does not teach that a client's single “pull” request receives

multiple content keys in response, as the Petition’s theory requires. Ex. 2011 at 37:14-20; Ex. 2010 ¶ 65.

The Petition additionally contends that it would have been obvious to an ordinary artisan “to package multiple keys/entitlement messages (ECMs or EMMs) into a ‘single’ ‘combined’ ‘multicast message,’” Pet. 22, 28 (citing Ex. 1003 ¶¶ 81, 101). This allegation likewise fails to cure the Petition’s deficiency.

As a threshold matter, the Petition only presents this obviousness theory contingent on an interpretation of “*response*” that requires a single message. Pet. 22, 27-28. Patent Owner has not relied on such an interpretation of “response.” Rather, it has demonstrated that each “pull” request in Peterka yields only one content key in “response,” sent in only one message. Accordingly, the Petition’s obviousness theory is irrelevant and non-responsive to the argument presented here.

Nonetheless, the Petition’s obviousness allegation also fails on the merits in view of each of Peterka’s distribution models.

In Peterka’s pay-per-view model, the client only receives one *requested* content key (e.g., CK2 in Figs. 7A and 7B), which lasts for the remaining duration

of the pay-per-view event. Ex. 2010 ¶ 67. There are thus no other content keys to package and send in response to a client request for pay-per-view content. *Id.*

Under Peterka's Subscription model, the client receives a single content key that lasts for several days to several months. *Id.* ¶ 68. There is thus no need to send the client additional content keys for future subscription periods that are many days or months away. *Id.*; Ex. 1005 ¶ 96 (teaching that clients can request the next content key "adequately far enough in advance so as to avoid overloading the server.").

Moreover, the Petition's obviousness modification would cause problems of over-inclusivity and under-inclusivity in content key distribution for both Peterka's pay-by-time and subscription models. Ex. 2010 ¶ 71.

The Petition's proposed modification requires sending clients a "single combined ***multicast*** message" containing multiple content encryption keys for successive segments or subscription periods. Pet. 22, 27-28. A "multicast message" is a single message that is sent to ***multiple*** clients simultaneously. Ex. 2010 ¶ 70.

Sending "a single combined multicast message" with ***multiple*** content keys for multiple segments (pay-by-time model) or multiple subscription periods (subscription model) to multiple clients would give certain clients access to additional future content that they ***did not purchase***. Ex. 2010 ¶ 72. This would be

over-inclusive and would defeat the purpose of Peterka's pay-by-time and subscription models. *Id.* An ordinary artisan would not find that approach desirable or "efficient." *Id.*

At the same time, the multicast message for multiple future content keys would be sent *before* eventual new clients have joined the multicast distribution group. The Petition's modification would thus also cause under-inclusivity in content key distribution. Ex. 2010 ¶ 73.

This is illustrated by Peterka's use-case examples selling segments "fashion shows" and "sustained sports events (e.g. Olympics)" under the pay-by-time scheme. *Id.* ¶78. These are events with long durations that do "not have a well defined start or end time or a self-contained plot[.]" *Id.* Long after the start of the live stream event, new clients will decide to purchase discrete content segments, such as a particular Olympic event. Ex. 2010 ¶ 73. Under the Petition's modification, Peterka's system will have *already* sent out the content key(s) for the segment(s) these new clients seek in a prior multicast message that *did not include them*. Ex.

2010 ¶ 73. As a result, a new message will have to be sent, contrary to the Petition's alleged "efficiency" motivation. *Id.*

Nor does Peterka itself teach that the Petition's modification is a "viable option," as the Petition contends. Pet. 22 (citing Ex. 1005 ¶¶ 104, 121); Ex. 2010 ¶ 74.

Peterka's paragraph 104 is directed to sending individual keys intended for "*individual clients*" and encrypted under each client's "Unique Key" or "UK." Ex. 1005 ¶ 104; Ex. 2010 ¶ 76. Peterka teaches that these keys can be sent individually in a unicast message, Ex. 1005 ¶ 104, but teaches they can also be "combined into a single multicast message" and sent to multiple clients at once, leaving each intended client to identify and obtain the key directed to it. *Id.* ¶104; Ex. 2010 ¶ 76. So that clients can find the individual key directed *only* to them within the bundled multicast, "a user identification" must often be included in the individual messages (EMMs) that are bundled together. *Id.* ¶120; Ex. 2010 ¶ 76.

This teaching in Peterka is thus analogous to sending seven boxes each containing seven keys for seven different individuals to seven different addresses.

Ex. 2010 ¶ 76. Since each key is only useful to *one* recipient, no recipient would obtain anything they were not supposed to have. *Id.*

The Petition's modification, by contrast, requires placing multiple *content* keys for multiple *successive content periods* into one message and multicasting that message to multiple clients at once. Ex. 2010 ¶ 77. In this scenario, all clients receive multiple keys *they can use* to decrypt multiple successive content segments or subscription periods, whereas not all clients will necessarily have purchased all of that content, thus defeating Peterka's distribution models. *Id.* An ordinary artisan would not have been motivated to modify Peterka in that manner. *Id.*

Finally, the Petition's reliance on Suzuki to support its obviousness theory is likewise unavailing. Ex. 2010 ¶ 78. Suzuki's paragraphs 76 and 77 do not teach transmitting multiple "*content encryption keys*," they only disclose providing "key *information* relating to the key currently in use (key A) and the next key to be used (key B)." Ex. 1006 ¶¶ 76-77; Ex. 2010 ¶ 78. And Suzuki is not directed to a pay-by-time, pay-per-view, or subscription distribution model, in which a content key grants access to content that different clients must individually decide to purchase in discrete units or increments, as in Peterka. Ex. 2010 ¶ 78. Accordingly, Suzuki could

not motivate providing Peterka's clients with unpurchased content for segments or subscription periods that were designed to require a purchase. *Id.*

Accordingly, the Petition does not meet or render obvious limitations [1B] and [16C] under its theories with respect to any Grounds. *Id.*

IV. The Petition does not meet “content encryption *key identifiers*” as recited in all claims

Claim 1 limitation [1B] and claim 16 limitation [16C] further recite “parsing content information from the playback request response, the content information including . . . content encryption *key identifiers*.”

The '178 Patent states: “A history of individual content encryption keys and the order in which they were generated is maintained. Each content encryption key in the history is assigned a unique identifier which is referred to herein as the content encryption key identifier.” Ex. 1001 at 3:28-32.

In view of that language, the specification imposes two requirements for “content encryption key identifiers.” Ex. 2010 ¶¶ 82-83. First, each content encryption key identifier must be “*unique*” in the context of *all* content encryption keys maintained in the “history.” *Id.* Second, each content encryption key identifier

must be “**assigned**” to the corresponding content encryption key **itself**. *Id.* This second requirement is further confirmed by the words “**key** identifier.”

The Petition contends that Peterka’s teaching of “a sequence number or a time stamp” assigned to ECM/EMM messages, Ex. 1005 ¶121, meets the recited “content encryption key identifier” of the challenged claims. Pet. 25. The Petition contends that Peterka’s ECM/EMM time stamps or sequence numbers are unique to each ECM/EMM message, and each ECM/EMM “carries a different content encryption key.” *Id.* Mr. Wechselberger further states that Peterka’s message sequence numbers or timestamps “**can be used** to uniquely identify the key that was in the ECM or EMM.” Ex. 1003 ¶ 91. Alternatively, the Petition alleges it would have been obvious to modify Peterka to “use a key identifier to identify the content key in an ECM/EMM.” Pet. 26. As another alternative, the Petition alleges it would have been obvious to combine Peterka with Suzuki to add key identifiers to Peterka’s client request response. Pet. 51-53. And with respect to claims 7-8 and 19, the Petition’s Ground 5 further proposes a four-reference combination that would include the use of timestamps as content key encryption identifiers to allow clients “that held keys

for multiple future segments” to determine “which key to apply at the next rotation boundary.” Pet. 69.

A. Peterka’s message sequence numbers or timestamps are *message* identifiers, not “content encryption *key* identifiers”

Contrary to the requirement that a “content encryption key identifier” is an identifier *assigned* to the content encryption key itself, Ex. 2010 ¶¶ 80-83, Peterka’s sequence numbers or timestamps are only assigned to its ECM/EMM *messages*, and not the content key included in those messages. Ex. 1005 ¶121 (teaching that the disclosed sequence number or time stamp is provided under “a format for such messages”); Ex. 2010 ¶ 89. As the Petition itself states, “the sequence number/timestamp is unique to *each* ECM/EMM[.]” Pet. 25 (emphasis added). Peterka’s sequence numbers or time stamps are thus not “content encryption *key* identifiers”; they are only *message* identifiers, which cannot meet the challenged claims. Ex. 2010 ¶ 89.

Indeed, Google’s expert admitted at his deposition that Peterka’s sequence numbers or time stamps are “*tied to the ECM or EMM[.]*” Ex. 2011 at 38:14-20 (emphasis added). And he further stated in his declaration that Peterka’s sequence numbers or time stamps “*can be used* to uniquely identify the key that was in the

ECM or EMM,” Ex. 1003 ¶91 (emphasis added), thus acknowledging that they are not assigned to the relevant *key* in Peterka’s message. Ex. 2010 ¶ 90.

The Petition’s obviousness proposal further confirms its concession that Peterka’s message time stamps or sequence numbers are not themselves “key identifiers.” The Petition states: “POSITAs would have been motivated . . . to implement Peterka’s client *to receive and use a key identifier* to identify the content key in an ECM/EMM.” Pet. 26 (emphasis added). If Peterka already had such key identifiers, the Petition would not have proposed an obviousness modification for adding them. Ex. 2010 ¶ 91.

Separately, in Peterka’s “pull” model, each content key request receives only one key in response. Ex. 2010 ¶ 92. Peterka thus also does not teach a request response that includes *multiple* “content encryption key identifiers” as the challenged claims require. *Id.*

B. The Petition fails to demonstrate obviousness with respect to “content encryption key identifiers”

The Petition’s obviousness contentions as to “key identifiers” likewise fail.

The Petition first contends that Peterka’s Figure 7B “*shows* delivery of content keys with unique designations—CK0, CK1, CK2,” and that this would have motivated the ordinary artisan to “use a key identifier to identify the content key in

an ECM/EMM.” Pet. 26 (emphasis Petition’s). Not so. Peterka uses CK0, CK1, and CK2 as descriptive labels for its patent figures and teachings; it does not disclose or suggest sending any such labels to clients as key identifiers. Ex. 2010 ¶ 94.

The Petition further contends that “POSITAs would have recognized the need to differentiate between keys” in Peterka’s unmodified system, and would have been motivated to use key identifiers as a solution. Pet. 26.

But the Petition itself defeats that motivation by explaining that Peterka can *already* identify content keys by *using* the allegedly unique sequence numbers or time stamps assigned to each *message* carrying a single content key. Pet. 25; Ex. 2010 ¶ 94. Peterka can thus differentiate between keys using message identifiers, and has no need for additional “key identifiers” as recited in the challenged claims. Ex. 2010 ¶ 94.

For the same reasons, the Petition likewise defeats its own motivation as to the combination of Peterka and Suzuki. Pet. 52-53 (contending “[a] POSITA would have recognized the importance of [key] identifiers to differentiate amongst keys, and that providing unique values (*such as* key numbers) to keys would ensure that the system does not confuse keys.”). Since the Petition states Peterka’s message

identifiers already serve that purpose, there would be no motivation to add key identifiers Peterka does not need to solve a problem it does not have. Ex. 2010 ¶ 95.

Finally, the Petition's Ground 5 modification based on the combination of Peterka, Bocharov, Suzuki, and Peterka308 likewise rests on a false factual premise. Pet. 69-73. The Petition contends that applying time stamps to Peterka's content keys as "key identifiers" would have been useful "for a client that held keys for *multiple future segments*, because it could readily compare the current playback location of a media stream to the respective location information of the future segment keys to determine which key to apply at the next key rotation boundary." Pet. 69-70 (emphasis added). The Petition further states: "Implementing Peterka308's teaching would have beneficially allowed clients to more effectively manage keys obtained for segments *multiple periods in advance*." Pet. 70 (emphasis added).

But as shown extensively above, Peterka does *not* teach or render obvious providing a client with multiple content keys for "multiple future segments" or "multiple periods in advance." Ex. 2010 ¶ 97. Accordingly, an ordinary artisan could

not have been motivated to modify Peterka to better manage multiple content keys obtained multiple segments in advance. *Id.*

For the reasons above, the Petition thus does not meet the “key identifiers” requirement of all challenged claims on any Grounds.

V. The Petition’s expiration time theory does not meet “detecting content encryption key rotation boundaries”

A. Relevant claim language and claim interpretation

All challenged claims require “detecting content encryption key rotation boundaries between periods of use of different content encryption keys in decrypting retrieved content.” Claim 1 limitation [1D]; claim 16 limitation [16E].

The ’178 Patent teaches that “[t]he transitioning between use of different keys is [] referred to herein as key ‘rotation.’” *Id.* at 1:63:65. A key rotation “boundary” occurs at the point where key rotation should occur. Ex. 2010 ¶ 100.

A content encryption key can expire for two reasons: normal-period based expiration (in which the expiration time assigned to the key is reached) and “an explicit key change notification” that requires a key change outside the period-based boundary that was otherwise defined. *Id.* at 11:3-7. (“In step 310, the client 110 checks the expiration of current content key to see if a new key should be applied. There are two cases—normal period-based expiration, and an explicit key change

notification received in step 308 that applies to the current segment being decrypted.”).

For normal period-based expiration, “the workflow manager generates content encryption keys with a fixed duration lifespan on a fixed periodic basis.” *Id.* at 3:9-11. “[T]he license server distributes fixed duration lifespan information to client.” *Id.* at 3:46-47.

For an explicit key change notification, “a key change is signaled, i.e., a key change [is] requested outside of any fixed duration lifespan,” for instance, “for security reasons[.]” *Id.* at 10:39-43.

Significantly, “the workflow manager may initiate content encryption key rotation at *any* time, *outside* of the fixed duration lifespan of the existing key.” *Id.* at 3:53-55 (emphasis added).

Since any “content encryption key rotation boundary” in the ’178 Patent can be defined by a notification for early, non-period-based key rotation, the client cannot reliably identify content encryption key rotation boundaries by simply monitoring the expiration time of the current key. Ex. 2010 ¶ 105.

At the same time, the client also cannot *only* look to express key-change notifications, since key change notifications are not sent for all keys, and the rotation

boundary for a given key—after checking for any notifications and considering any such notifications—*may* instead be defined by its expiration time. Ex. 2010 ¶ 105.

Accordingly, the '178 Patent teaches “a process *for detecting the rotation of content encryption keys.*” Ex. 1001 at 9:47-49 (emphasis added). This is the teaching of Figure 3, and illustrated by that figure. *Id.* (“FIG. 3 is a flow chart 300 describing a process for detecting the rotation of content encryption keys and retrieving updated content encryption keys.”); Fig. 3; Ex. 2010 ¶ 106.

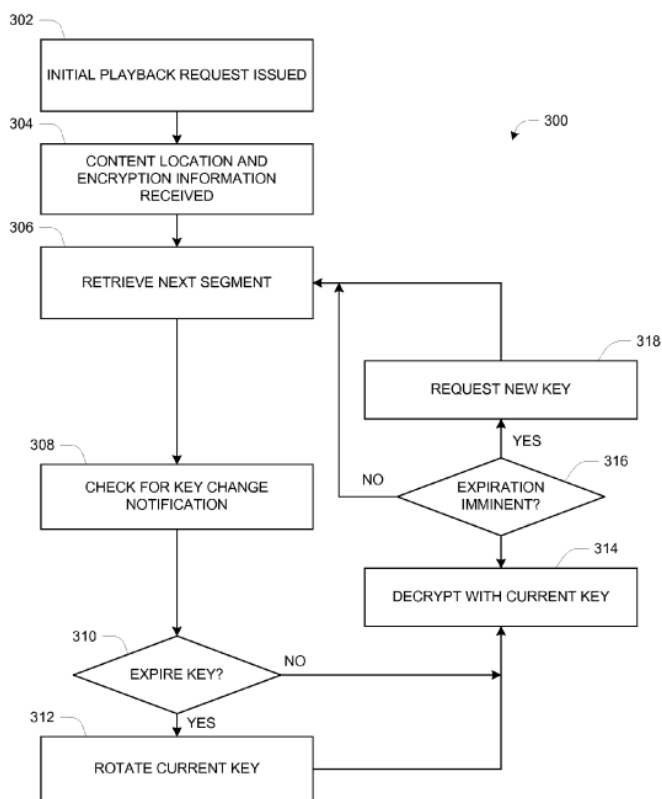


Fig. 3

As Figure 3 shows above at step 308, the process for “**detecting**” a key rotation boundary always requires that the client “check for key notification.” Fig. 3; Ex. 2010 ¶ 107. That step is not optional. To that end, the ’178 Patent teaches: “Once the new segment is retrieved, processing proceeds to step 308 where the client 110 checks to see if a key change is signaled, i.e., a key change was requested outside of any fixed duration lifespan.” Ex. 1001 at 10:39-42. The patent further teaches: “If a content encryption key change request is detected, the client notes the need to expire the current content encryption key and the content encryption key identifier of the new key to be used. The client 110 then proceeds to step 310.” *Id.* at 10:65-11:2.

Thus, under the “process for detecting the rotation of content encryption keys” taught in the ’178 Patent, *id.* at 9:47-49 (emphasis added), the client must always check for an explicit key change notification. Ex. 2010 ¶ 108.

“In step 310, the client 110 checks the expiration of current content key to see if a new key should be applied. There are **two** cases—normal period-based expiration, and an explicit key change notification received in step 308 that applies to the current segment being decrypted.” *Id.* at 11:2-7 (emphasis added). Once again, identifying the correct rotation boundary requires considering **both** the key’s normal-

period based expiration time **and** checking for “an explicit key change notification.”

Id.; Ex. 2010 ¶ 109.

“In step 316, the client 110 checks to see if the current content encryption key is going to expire in the near future **or** if a content encryption key change request for a future segment was detected in step 308. [. . .] If the current content encryption key is not close to expiring **and** a content encryption key change request was not detected in step 308, processing proceeds back to step 306 where the client 110 begins retrieving the next segment. If the current content encryption key is within a fixed threshold of expiring **or** a content encryption key change request was detected in step 308, processing proceeds to step 318 where the client 110 requests a new content encryption key from the license server 106.”
Id. at 11:38-53 (emphasis added).

Once again, the teaching as to step 316 reiterates that the process for detecting a content encryption key rotation boundary in the '178 Patent requires checking both a key's expiration time **and** checking for an explicit key-change notification for the

key and then identifying the applicable rotation boundary in view of at least those two items of information. *Id.* at 10:39-42, 11:3-7, 11:38-41; Ex. 2010 ¶ 111.

Notably, the '178 Patent never teaches “detecting” a “rotation boundary” *without* checking for a key change notification and considering the information provided in any such notification, and based only on consideration of an expiration time. Ex. 2010 ¶ 111.

Other language in the challenged claims further supports the above understanding of the “detecting” limitation. The challenged claims earlier recite “*parsing* content information from the playback request response, the content information including . . . content encryption key *expiration times*.” Claim 1 limitation [1B], Claim 16 limitation [16C] (emphasis added).

The ordinary artisan would have understood that once the content encryption key expiration times have been “*parsed*,” the client has identified the expiration time associated with the relevant content encryption key. Ex. 2010 ¶ 112.

Google’s expert admitted to that fact under cross-examination. Ex. 2011 at 35:6-36:8 (Q: “In Peterka’s system, once the ECMs or EMMs have been parsed to distinguish, for example, the content encryption key expiration times, does Peterka’s system, at that point, know what the content key expiration times are for each key?”

A: “Yes, it will. Because Peterka explains that that information is part of what’s delivered in the ECM, along with the key.”).

That reality further supports the conclusion that the claims’ separate requirement of “***detecting*** content encryption key rotation boundaries” is not met by identifying a key’s expiration time alone. Such an understanding would render the “detecting” limitation redundant to the “parsing” limitation, which is disfavored. *Power Mosfet Techs., L.L.C. v. Siemens AG*, 378 F.3d 1396, 1410 (Fed. Cir. 2004) (“interpretations that render some portion of the claim language superfluous are disfavored.”).

The claims’ use of the phrase “content encryption key ***rotation boundary***” further supports understanding the “detecting” limitation to require more than merely monitoring or keeping track of an expiration time. Ex. 2010 ¶ 114. What must be “detected” is a “rotation boundary,” not merely an “expiration time,” and this requires checking for both an express notification and an expiration time, analyzing the resulting information, and determining the appropriate rotation boundary in view of the results. Ex. 2010 ¶ 114.

B. The Petition's expiration-time theory does not meet the "detecting" limitation

The Petition relies exclusively on Peterka's "pull" model to meet the "detecting" limitation. Pet. 12, 18, 32-34.

The Petition quotes Peterka as teaching that, in its pull model, "each client keeps track of the keys and their *expiration times* and actively requests new keys before the current keys *expire* so as to avoid service interruptions." Pet. 32 (quoting Ex. 1005 ¶ 105) (emphasis modified). The Petition also quotes Peterka as teaching that clients "monitor" expiration times. Pet. 32.

The Petition then concludes, without further analysis, that "Peterka thus discloses clients that *detect* upcoming *content encryption key rotation boundaries* based on monitored key expiration times that allow the client to request new keys ahead of the upcoming boundary." Pet. 32 (emphasis Petition's).

The Petition thus alleges that by simply identifying a key's expiration time and monitoring that expiration time as part of Peterka's pull model, Peterka discloses its clients "*detecting* content encryption key rotation boundaries."

The Petition is incorrect. Ex. 2010 ¶ 117. As shown above, simply identifying a key's expiration time and monitoring that expiration time does not meet "detecting

content encryption key rotation boundaries” as claimed and taught in the ’178 Patent. Section V.A, *supra*.

Moreover, Peterka teaches that when the client requests keys under the “pull” model, “the server need not monitor the status of the client’s keys; rather, the responsibility of determining when a new key is required” is “passed to the client,” Ex. 1005 ¶ 106, and clients make that determination *exclusively* based on the current key’s “expiration time.” *Id.* ¶ 105 (teaching that under the “pull” model, “each client keeps track of the keys and their expiration times and actively requests new keys before the current keys *expire* so as to avoid service interruptions.”) (emphasis added); Ex. 2010 ¶ 118.

Peterka’s pull embodiment does thus not teach clients checking for and considering *multiple* items of information to “detect” the appropriate rotation boundary, as the challenged claims of the ’178 Patent require. Ex. 2010 ¶ 118.

Accordingly, the Petition does not and could not meet the “detecting” limitations of [1D] and [16E] based on Peterka’s “pull” embodiment. Ex. 2010 ¶ 119.

VI. The Petition's notification theory does not meet the "issuing requests" requirement of all challenged claims

In alternative to its expiration-time based theory, the Petition separately argues that Peterka meets the "detecting" limitation through Peterka's teaching of signaling a key change for a content key. Pet. 33; Ex. 1005 ¶¶ 116-117.

Claim 1 limitation [1E], however, recites: "issuing *requests* to a license server ahead of a key rotation boundary to retrieve a second content encryption key to be used after a content encryption key rotation boundary is reached." (emphasis added). Claim 16 limitation [16F] recites the same.

Peterka's signaled key change teachings do not meet the "issuing requests" requirement of the challenged claims. Ex. 2010 ¶¶ 120-129.

In its signaling embodiment, Peterka teaches that "when a new key is implemented, the implementation will be signaled to the clients so that the clients can begin utilizing the new content key with which they *have been provided*." Ex. 1005 ¶ 116 (emphasis added). Peterka further reiterates that the new content key "has recently been distributed to the client." *Id.* ¶117. In this embodiment, clients

thus always receive the “new content key” *before* they receive a content key-change signal. Ex. 2010 ¶ 122.

Peterka’s “pull” model, however, provides no mechanism for a client to reliably request and obtain the necessary “new content key” *before* the client has received a signal to change keys. Ex. 2010 ¶¶ 122-124. Rather, under the “pull” model, a client only requests a new key based on the *expiration time* of the current key. *Id.* ¶¶ 105-106 (“Under the pull model, each client keeps track of the keys and their expiration times and actively requests new keys before the current keys *expire* so as to avoid service interruptions.”); (“In block 818, the key and its expiration time are distributed to the client in response to the request by the client. Thus the server need not monitor the status of the client’s keys; rather, the responsibility of determining when a new key is required can be passed to the client.”).

Moreover, in Peterka’s “pull” model, “the server need not monitor the status of the client’s keys[.]” *Id.* ¶ 106. But in Peterka’s signaling embodiment, the server *does* monitor the status of the keys: it is the server that determines when a new key is required, and it is the server that tells clients when a new key must be used. *Id.* ¶ 116 (“when a new content key is implemented, the implementation will be signaled to the clients so that the clients can begin using the new content key[.]”); *id.* ¶ 117

(“a predetermined bit can be used to indicate if an old or current content key should be used as opposed to a new content key”); Ex. 2010 ¶ 125. Peterka’s signaling method is thus incompatible with Peterka’s “pull” method. Ex. 2010 ¶ 125.

Rather, Peterka’s server must “push” a new content key to clients ahead of a signaled key change, without clients issuing a request for that new content key. Ex. 2010 ¶ 126. The Petition’s signaling method theory thus does not meet limitations [1E] and [16F] of all challenged claims. *Id.*

For the same reasons, the Petition’s argument that “signaling when to use a key in the client’s possession *does not require* Peterka’s server to *push* updated keys” is incorrect in the context of Peterka’s teaching. Pet. 33-34 (emphasis Petition’s). Peterka does not teach clients “pulling” or “requesting” new content keys in the context of a signaled key change; it only teaches doing so based on expiration times. Ex. 2010 ¶ 127. And Peterka’s signaling method requires that clients receive the new content key *before* receiving a key change signal, precluding clients from making a “pull” request *in response* to a key change signal. *Id.* Accordingly, the

delivery method for new content keys in connection with a signaled key change is a “push.” *Id.*

Nor would it have been obvious to use “Peterka’s key change signaling with Peterka’s pull model,” as the Petition alleges. Pet. 34. As noted, under Peterka’s pull model, clients only request new keys based on upcoming expiration times, *not* signaled key changes. Ex. 1003 ¶ 115; Ex. 2010 ¶¶ 123, 127. Peterka’s servers, however, can signal a content key change at any time, and the client must always have the “new content key” *before* the key change signal. Ex. 2010 ¶ 128. Peterka’s “pull” model cannot ensure that clients will have a new content key before a signaled key change, and the “pull” model is thus incompatible with Peterka’s signaled key change method. *Id.* Accordingly, the ordinary artisan would not have been motivated to combine Peterka’s pull method with its signaled key change method as the Petition has proposed. *Id.*

The Petition’s theories based on Peterka’s signaled key change teaching thus do not meet or render obvious the “issuing requests” limitation of all challenged claims.

* * *

In sum, the Petition does not meet or render obvious the challenged claims based on either its expiration time theory or its key change signaling theory.

VII. Conclusion

For the reasons above, the Board should find all challenged claims not unpatentable.

Respectfully submitted,

/Kayvan B. Noroozi/
Kayvan B. Noroozi, Admitted *Pro Hac Vice*
Peter C. Knops, Reg. No. 37,659
Noroozi PC
11601 Wilshire Blvd., Suite 2170
Los Angeles, CA 90025

Attorneys for the Patent Owner

Date: July 2, 2025

CERTIFICATION OF WORD COUNT

The undersigned hereby certifies that the portions of the above-captioned **PATENT OWNER'S RESPONSE** specified in 37 C.F.R. § 42.24 have 10,304 words, in compliance with the 14,000 word limit set forth in 37 C.F.R. § 42.24(b)(1). This word count was prepared using Microsoft Word 365.

/Kayvan Noroozi/
Kayvan Noroozi
11601 Wilshire Blvd., Suite 2170
Los Angeles, CA 90025
Tel.: 310-975-7074

Counsel for Patent Owner

Date: July 2, 2025

CERTIFICATE OF SERVICE

The undersigned hereby certifies that the foregoing **PATENT OWNER'S Response** was served electronically via e-mail on July 2, 2025, on the following counsel of record for Petitioner:

Scott McKeown, Reg. No. 42866
SMcKeown-PTAB@wolfgreenfield.com
Anant K. Saraswat, Reg. No. 76,050
ASaraswat-PTAB@wolfgreenfield.com
Turhan F. Sarwar, pending admission pro hac vice
TSarwar-PTAB@wolfgreenfield.com
Elizabeth A. DiMarco, pending admission pro hac vice
Elizabeth.DiMarco@wolfgreenfield.com

Wolf, Greenfield & Sacks, P.C.
600 Atlantic Avenue Boston, MA 02210-2206
Telephone: 617-646-8000
Facsimile: 617-646-8646

/Kayvan Noroozi/
Kayvan Noroozi
Noroozi PC
11601 Wilshire Blvd., Suite 2170
Los Angeles, CA 90025
Tel.: 310-975-7074

Counsel for Patent Owner

Date: July 2, 2025