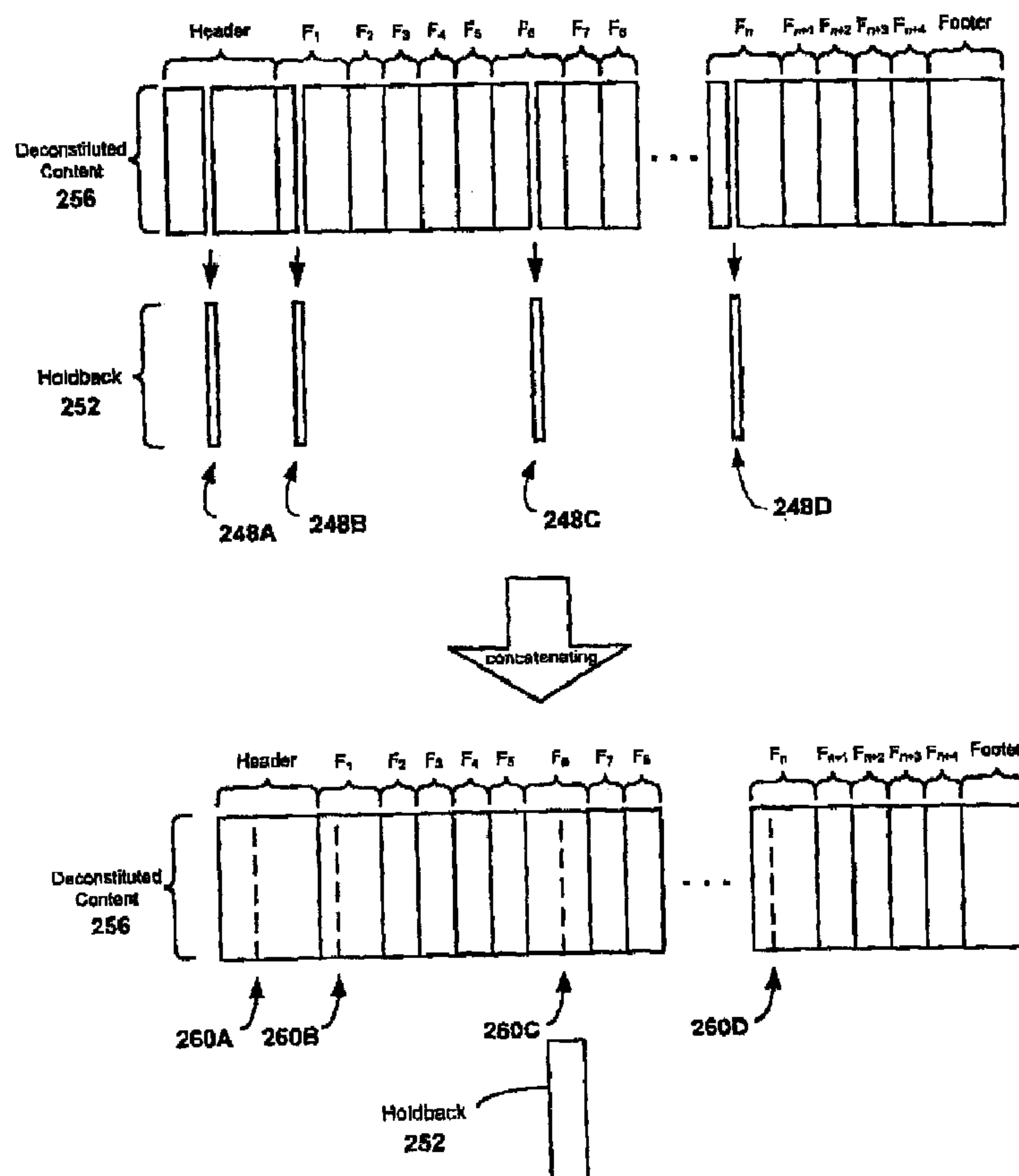




(86) **Date de dépôt PCT/PCT Filing Date:** 2007/04/13
(87) **Date publication PCT/PCT Publication Date:** 2007/10/25
(45) **Date de délivrance/Issue Date:** 2016/11/08
(85) **Entrée phase nationale/National Entry:** 2009/10/13
(86) **N° demande PCT/PCT Application No.:** CA 2007/000612
(87) **N° publication PCT/PCT Publication No.:** 2007/118311
(30) **Priorité/Priority:** 2006/04/13 (US11/403,640)

(51) **Cl.Int./Int.Cl.** *H04L 9/00* (2006.01),
G06F 21/10 (2013.01), *H04L 12/16* (2006.01),
H04L 9/32 (2006.01)
(72) **Inventeurs/Inventors:**
EPSTEIN, EARL HOWARD, CA;
EPSTEIN, JOHNNY STUART, CA
(73) **Propriétaire/Owner:**
6548601 CANADA INC., CA
(74) **Agent:** MCMILLAN LLP

(54) **Titre : PROCÉDE ET APPAREIL POUR FOURNIR DU CONTENU CODE**
(54) **Title: METHOD AND APPARATUS FOR DELIVERING ENCODED CONTENT**



(57) **Abstr  /Abstract:**

A method for delivering encoded content comprises extracting a holdback from the encoded content, thereby damaging the encoded content. The damaged encoded content is distributed. The holdback is transmitted to enable reintegration of the holdback with the damaged encoded content thereby to restore the encoded content.



(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
25 October 2007 (25.10.2007)

PCT

(10) International Publication Number
WO 2007/118311 A1

(51) International Patent Classification:

H04L 9/00 (2006.01) *H04L 9/32* (2006.01)
H04L 12/16 (2006.01) *G06F 21/24* (2006.01)

(74) Agents: RUSTON, David A. et al.; Sim & McBurney, 330
University Avenue, 6 Floor, Toronto, Ontario M5G 1R7
(CA).

(21) International Application Number:

PCT/CA2007/000612

(22) International Filing Date: 13 April 2007 (13.04.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

11/403,640 13 April 2006 (13.04.2006) US

(71) Applicant (for all designated States except US): **6548601**
CANADA INC. [CA/CA]; 66 Hewitt Avenue, Apartment
3, Toronto, Ontario M6R 1Y3 (CA).

(72) Inventors; and

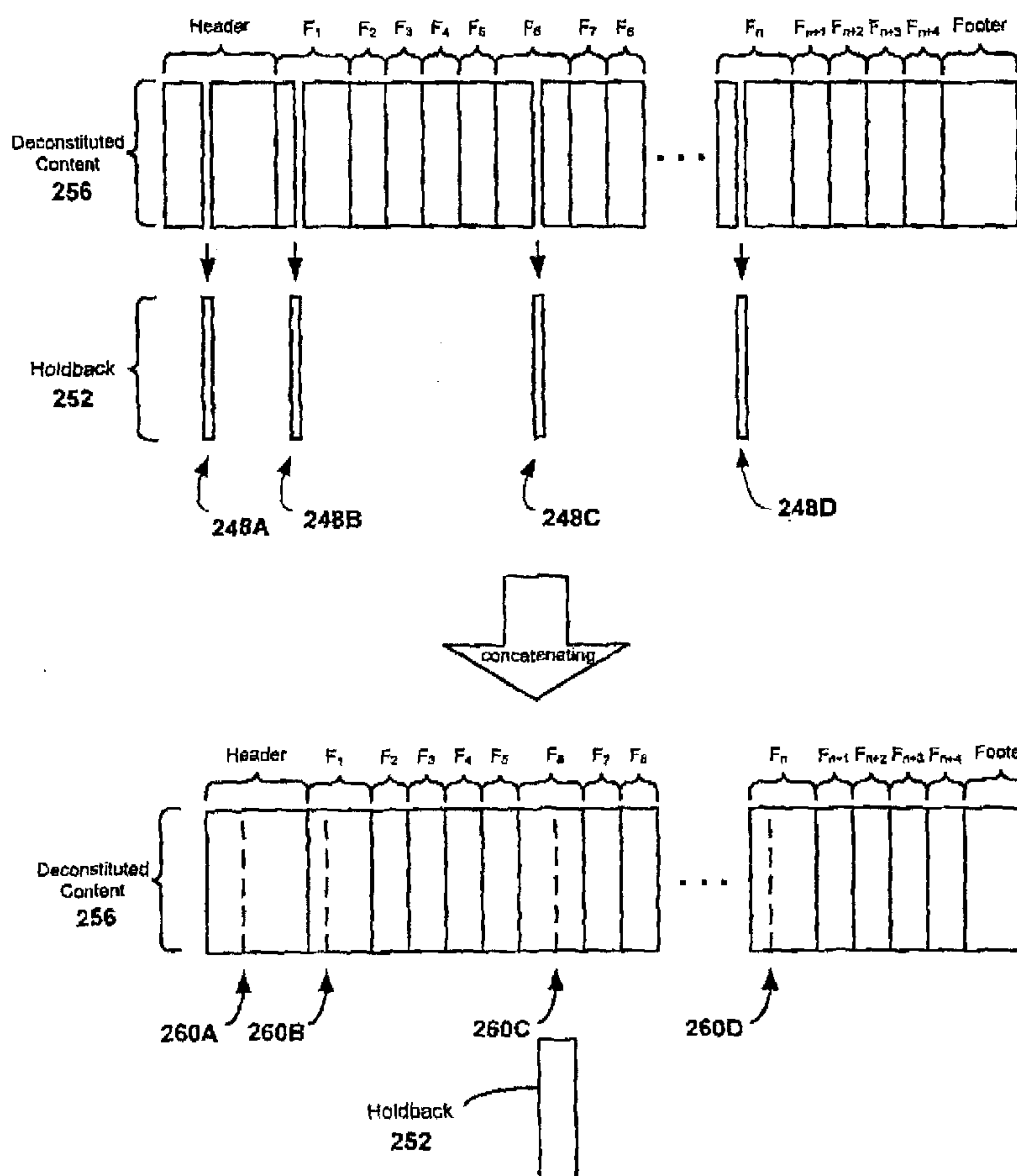
(75) Inventors/Applicants (for US only): **EPSTEIN, Johnny**
Stuart [CA/CA]; 77 Carlton Street, Suite 901, Toronto,
Ontario M5B 2J7 (CA). **EPSTEIN, Earl Howard**
[CA/CA]; 66 Hewitt Avenue, Apartment 3, Toronto,
Ontario M6R 1Y3 (CA).

(81) Designated States (unless otherwise indicated, for every
kind of national protection available): AE, AG, AL, AM,
AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH,
CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES,
FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN,
IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR,
LS, LT, LU, LY, MA, MD, MG, MK, MN, MW, MX, MY,
MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS,
RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN,
TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every
kind of regional protection available): ARIPO (BW, GH,
GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM,
ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),
European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI,
FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL,

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR DELIVERING ENCODED CONTENT



(57) Abstract: A method for delivering encoded content comprises extracting a holdback from the encoded content, thereby damaging the encoded content. The damaged encoded content is distributed. The holdback is transmitted to enable reintegration of the holdback with the damaged encoded content thereby to restore the encoded content.

WO 2007/118311 A1



PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Published:

— *with international search report*

**METHOD AND APPARATUS FOR DELIVERING ENCODED
CONTENT**

Field of the Invention

[0001] The present invention relates generally to content delivery and in particular, to a method and apparatus for delivering encoded content.

Background of the Invention

5 [0002] The digital representation of content is known. Content includes, but is not limited to, music, video, program code, text and graphical documents, images, interactive presentations, etc. During digital representation, the content is generally encoded in accordance with a pre-set standard to create encoded content, such as a file or streaming media. Each standard generally
10 specifies a protocol for encoding content such that it may be stored or transmitted, and a protocol for decoding content that has been encoded to reconstruct the content for playback. These standards are known as document types, and may involve protocols called codecs. The encoded content can be stored on digital media such as for example, a hard disk drive, a floppy disk, an
15 optical media disk, flash memory, volatile memory or, alternatively, can be transmitted via a communications network. As both storage and network bandwidth have associated costs, such codecs are generally designed to compress the digital representation of content while maintaining a desired level of quality. For music, a number of codecs exist, including MP3, AAC and WAV. Similarly, for still images, codecs include, but are not limited to, JPEG, GIF,
20 PNG and TIFF. A number of codecs exist for video, including MPEG-2, MPEG-4, AVI and WMV. Similarly, other encoding schemes for other types of computer-readable content exist, such as plain text, document files (e.g., Microsoft Word and Excel documents), program files (e.g., executables and
25 dynamic link libraries) and interactive media (e.g., Microsoft PowerPoint and Macromedia Shockwave files).

[0003] A feature of encoded content is that there are few limitations on the ability to reproduce any number of identical copies of the encoded content

- 2 -

and distribute the copied encoded content freely without identifying the source of the copies. This ability to make unlicensed copies of encoded content is an artifact of how computers operate. The replication of encoded content is also an artifact of how the majority of communications networks are designed, in
5 that they reproduce data transmitted across them, regardless of the type of data.

[0004] In many circumstances, it is desired by the owners of encoded content to limit unauthorized access, copying and dissemination of encoded content. There are, however, no widely-implemented mechanisms in current
10 communications protocols or hardware that control the authorized use of the data being processed. That is, computers that use communications protocols to connect to networks, such as the Internet, and that exchange encoded content, do not have sufficient logical controls automatically to determine the proprietorship, source or rights associated with encoded content that is being
15 processed. As a result, the computers lack the ability to govern encoded content distribution.

[0005] In order to control the distribution and use of encoded content, some content proprietors have implemented digital rights management systems. Digital rights management systems store encoded content in an
20 encrypted digital format that corresponds to an encryption key, and rely on a non-standard application to decrypt the encoded content at the time of presentation or playback. These systems, however, typically require the use of specialized players and/or content viewers, hereinafter referred to as "decoders", that are capable of decrypting the encoded content, thereby limiting
25 the selection of decoders available to end-users and/or causing compatibility issues. For example, music content licensed to a person by a proprietor employing a particular digital rights management scheme may only be accessible via a particular decoder application on a particular operating system, and may not be decodable via a traditional hardware appliance, such as a
30 compact disk player.

[0006] Further, as the encoded content in such digital rights management systems includes the totality of the data representing the content,

WO 2007/118311

PCT/CA2007/000612

- 3 -

successful cryptanalytic attacks can ultimately permit access to the entire content.

[0007] There are a number of other schemes for restricting access to content which require specialized devices such as non-standard decoders or physical hardware. Still other schemes employ traditional cryptography and are therefore, vulnerable to the same class of cryptanalytic attacks referred to above. In these cases, only the computational problem of generating the correct decryption key needs to be solved in order for an attacker to gain unrestricted access to the entirety of the content, which can then be copied in a repudiable manner.

[0008] Current systems for distributing authorized encoded content may include license identifiers only as a non-essential part or extension of the functional part of the data structure of an encoding scheme, like a credit at the end of a film, or a notice in the headers of a file. As the content itself is not affected, however, the content can be separated reasonably easily from the identifiers.

[0009] File-sharing services, such as Internet-based Kazaa and Limewire, are under pressure from content proprietors to distinguish between encoded content which may be shared freely from that for which the proprietors' permission is required. Without the ability to preclude the unauthorized distribution of encoded content, such services risk liability.

[0010] Since existing popular codecs for encoded content do not have standardized features to identify the terms of authorized use and/or the individual licensed to use the encoded content, there is no consistent or automatic way for file-sharing services to assess whether or not the sharing of encoded content via their networks is unauthorized.

[0011] Ultimately, in these cases, since the end-user possesses the entirety of the content, the content can potentially be decrypted and copied or distributed in their entirety in a repudiable manner.

WO 2007/118311

PCT/CA2007/000612

- 4 -

Summary of the Invention

[0012] Accordingly, in one aspect there is provided a method of delivering encoded content, comprising:

- 5 extracting a holdback from said encoded content, thereby
- damaging said encoded content;
- distributing said damaged encoded content; and
- transmitting said holdback to enable reintegration of said holdback with said damaged encoded content thereby to restore said encoded content.

- 10 [0013] The holdback can be modified such that a distinct copy of the restored encoded content is generated when the holdback is reintegrated with the damaged encoded content. The holdback can be modified to include steganographically-embedded information. The holdback can be modified such that when the holdback is reintegrated into the damaged encoded content, the
- 15 quality of the restored encoded content is not significantly decreased. The holdback can be modified to include information identifying an end-user authorized to restore the encoded content, such as by identifying a record in a license database identifying the end-user. The license database can also identify distribution rights for the encoded content the end-user is authorized to
- 20 restore. The holdback may comprise a plurality of discrete portions of the encoded content. During the transmitting, the holdback is transmitted over a communication network. The transmitting is also performed subsequent to the distributing and in response to receipt of an authorized access request.

- [0014] In one embodiment, the distributing comprises distributing an
- 25 optical media disk including the damaged encoded content. Alternatively, the damaged encoded content can be distributed over a communication network.

- [0015] In one embodiment, the damaged encoded content is encrypted prior to the distributing. A second holdback is extracted from the encrypted damaged encoded content prior to the distributing. In another embodiment, the
- 30 encoded content is encrypted prior to the extracting.

WO 2007/118311

PCT/CA2007/000612

- 5 -

[0016] According to another aspect, there is provided a method of delivering encoded content, comprising:

generating a unique identifier;

embedding said unique identifier in encoded content selected

5 from a library of encoded content; and

transmitting said encoded content to an end-user associated with the unique identifier.

[0017] The generating is performed in response to an authorized access request granted to the end-user. In one embodiment, the unique identifier

10 identifies the end-user. The unique identifier may also identify a record in a license database identifying the end-user. The license database may also identify distribution rights for encoded content the end-user is authorized to access.

[0018] According to yet another aspect, there is provided a method of delivering encoded content, comprising:

15

dividing said encoded content into segments;

generating a set of distinct instances of at least one of said segments;

selecting one of said distinct instances from each set of distinct

20 instances; and

reassembling said encoded content using said selected distinct instances.

[0019] In one embodiment, a set of distinct instances is generated for each of the segments. Each set of distinct instances is encrypted after the

25 generating. The selected distinct instances are decrypted before reassembly. The distinct instances can be delivered to a client prior to selection, and the selected distinct instances can be communicated to the client after selection, wherein the client performs the reassembly.

[0020] According to yet another aspect, there is provided a method of authenticating encoded content, comprising:

30

analyzing encoded content to determine at least one characteristic thereof;

WO 2007/118311

PCT/CA2007/000612

- 6 -

transmitting said at least one characteristic for the purpose of authentication; and

receiving authentication of said encoded content if said at least one characteristic signifies that said encoded content is authentic.

5 [0021] During the analyzing, a hash calculation is performed on at least a portion of the encoded content. The header of the encoded content can be parsed to read an identifier therein. The encoded content can be parsed to locate a brand and an identifier read, if the brand is present in the encoded content.

10 [0022] According to yet another aspect, there is provided a computer-readable medium, comprising:

encoded content having an identifier embedded therein, said identifier corresponding with a record in a license database identifying an end-user to which said encoded content is licensed.

15 [0023] The encoded content can be customized such that it becomes unique to the end-user. The encoded content enables authorized extraction of information that can identify the end-user, or identify the content as having been tampered with in an unauthorized way.

[0024] According to yet still another aspect, there is provided an
20 encoded content delivery apparatus, comprising:

a data extractor extracting data from said encoded content, thereby damaging said encoded content and conditioning it for distribution; and

a reintegration map generator creating a reintegration map
25 identifying insertion locations for the extracted data to enable the damaged encoded content to be restored.

[0025] According to still yet another aspect, there is provided a system for delivering encoded content, comprising:

at least one server extracting a holdback from stored encoded
30 content to be delivered to an end-user thereby damaging said encoded content, said at least one server preparing said damaged encoded content for distribution to said end-user in response to an authorized request, said at

WO 2007/118311

PCT/CA2007/000612

- 7 -

least one server further transmitting said holdback to said end-user to enable said end-user to reintegrate said holdback with said damaged encoded content thereby to restore said encoded content.

[0026] In accordance with still yet another aspect, there is provided a process for encrypting and encoding uniquely identifiable bits of data into a file format of a digital media file such that the uniquely identifiable bits are not obvious or identifiable by an end-user accessing the digital media file as intended, are not extractable by end-users attempting a cryptanalytic attack, yet are easily identifiable by the digital media file owner, licensed distributor, or other authorized monitor.

[0027] In accordance with still yet another aspect, there is provided a distribution system that encodes, distributes, and licenses encoded content by pre-distributing or delivering only a portion of the encoded content to an end-user, while withholding the remainder of it, ensuring that the end-user must obtain the withheld portion to make the encoded content usable. Information identifying the end-user and possibly other attributes is obtained and embedded into the withheld portion before the withheld portion is provided to the end-user for reintegration with the remainder of the encoded content.

20 Brief Description of the Drawings

[0028] Embodiments will now be described, by way of example only, with reference to the attached drawings, wherein:

[0029] Figure 1 shows a system for delivering encoded content;

[0030] Figure 2 is a flowchart of the encoded content delivery method employed by the system of Figure 1;

[0031] Figure 3 is a flowchart showing the steps performed during extraction of a holdback from encoded content;

[0032] Figure 4 is an abstract datagram of an MPEG-2 encoded file;

[0033] Figure 5 is a holdback reintegration map determined for video encoded using MPEG-2;

WO 2007/118311

PCT/CA2007/000612

- 8 -

- [0034] Figure 6 is an abstract schematic representation of the extraction of the holdback from the MPEG-2 encoded file of Figure 4 corresponding to the set of re-assembly instructions of Figure 5;
- [0035] Figure 7 is a flowchart of the steps performed during encrypting of
5 encoded content;
- [0036] Figure 8 is a flowchart of the steps performed during requesting of authorized access to encoded content;
- [0037] Figure 9 is a window presenting a list of encoded content that can be selected for accessing;
- 10 [0038] Figure 10 shows the system of Figure 1 with additional components for operating a peer-to-peer file-sharing network;
- [0039] Figure 11 is a flowchart showing the steps performed during checking of authorization to access encoded content in the system of Figure 10;
- 15 [0040] Figure 12 illustrates the steps performed during encoded content license checking;
- [0041] Figure 13 is a flowchart of an alternative encoded content delivery method;
- [0042] Figure 14 shows the steps performed during analysis of encoded
20 content;
- [0043] Figure 15A is an abstract diagram of encoded content;
- [0044] Figure 15B illustrates segmentation of the encoded content of Figure 15A;
- [0045] Figure 16 illustrates instances of the encoded content segments
25 of Figure 15B;
- [0046] Figure 17 illustrates the extraction of holdbacks from the instances of Figure 16;
- [0047] Figure 18 shows an exemplary content map;
- [0048] Figure 19 is a flowchart of the steps performed during the
30 decrypting, reconstituting and reassembling of encoded content; and
- [0049] Figure 20 shows the encoded content after reintegration of the holdback and assembly from the encoded content segment instances.

- 9 -

Detailed Description of the Embodiments

[0050] Turning now to Figure 1, a system for delivering encoded content is shown and is generally identified by reference numeral 24. In this embodiment, critical data is extracted from the encoded content, thereby to
5 render the encoded content incomplete and as a result, damaging the ability to decode the content. In other words, by extracting the critical data, the encoded content either simply cannot be accessed or played or if it can be accessed or played, the quality of the encoded content is significantly compromised or degraded. The incomplete encoded content is then encrypted using a cipher
10 block chain approach and, once again, critical data is extracted from the encrypted incomplete encoded content to damage further the encoded content such that decryption of the encoded content is made infeasible. The encrypted incomplete encoded content is thus "deconstituted", and can only be "reconstituted" when the critical data is reintegrated. The deconstituted
15 encoded content is then pre-delivered to an end-user. Upon licensing the encoded content, the critical data initially extracted from the encoded content is modified such that when the critical data is reintegrated with the incomplete encoded content, a distinct copy of the encoded content is generated. The critical data necessary to reconstitute the deconstituted encoded data is then
20 delivered to the end-user thereby enabling reconstitution of the encoded content.

[0051] The deconstituted encoded content is delivered in the form of an optical media disk 28 to an end-user 32, for reading using a personal computer ("PC") 36. The optical media disk 28 includes one or more files that represent
25 the deconstituted encoded content; that is, the encrypted incomplete encoded content from which has been extracted critical data. PC 36 is in communication with an authorization server 40 via the Internet 44.

[0052] The authorization server 40 is an enterprise-level server that includes web server and database server functionality. The authorization
30 server 40 manages an end-user database 48, a content completion database 52, a security database 56 and a license database 60. The end-user database

WO 2007/118311

PCT/CA2007/000612

- 10 -

48 stores various information about the end-users 32 of the system 24, including End-User ID, a password or passwords, addresses, information required to process payments, a transaction history and/or credit balances. This information is collected by the authorization server 40 during registration of an end-user together with credit purchases made by the end-user that are used to access encoded content. The End-User ID is a unique identifier selected for the end-user 32.

[0053] The content completion database 52 stores critical data of the encoded content held back (hereinafter referred to as a "holdback", or more particularly a "first holdback") and information associated with the first holdback. The security database 56 stores encryption keys for the encoded content on the optical media disk 28. In addition, the security database 56 stores critical data extracted from the encrypted incomplete encoded content (hereinafter referred to as a "second holdback") and second holdback reintegration maps that specify how to reintegrate second holdbacks into the encrypted deconstituted encoded content.

[0054] The license database 60 stores information about each license issued, including the End-User ID, a Content ID, a list of license parameters, etc. The license database 60 also includes a content map for each license issued. All other information stored concerning issued licenses, and license parameters, is accessed through the license database 60 using the content map as a key.

[0055] The general method of delivering encoded content 100 to an end-user 32 using the system 24 is shown in Figure 2. Upon authorization of the end-user 32, a customized first holdback is generated for the end-user. The customized first holdback contains information such that, when the first holdback is reintegrated with the incomplete encoded content, a customized copy of the encoded content is created. In particular, the first holdback is customized with an identifier referred to as the content consumer transaction identifier ("CCTID") that is a collision-resistant hash of the End-User ID, the Content ID, a set of license parameters, etc.

- 11 -

- [0056] The type of encoded content to be delivered to the end user is then determined and a marker referred to hereinafter as a "brand" as well as Content ID are embedded in the encoded content (step 110). For most types of encoded content, the brand and Content ID are embedded in one or more unused header fields. The brand indicates that the encoded content has been processed using the method described herein. The Content ID is a unique identifier generated for the encoded content to distinguish it from the encoded content. Next, the first holdback is extracted from the encoded content (step 120) thereby to yield incomplete encoded content.
- 10 [0057] Figure 3 shows the steps performed during the extraction of the first holdback from the encoded content at step 120. Initially a holdback scheme corresponding to the encoding type is selected (step 121). The holdback scheme is a set of rules that are used to specify how the first holdback is to be determined for the particular content encoding type. In some circumstances, it is desirable to select a first holdback that is small in size in comparison to the size of the encoded content, yet damages the content when extracted therefrom. The holdback scheme is also selected based on suitability for receiving the CCTID.
- 15 [0058] An exemplary abstract datagram 200 for a generic encoded content file (such as MPEG-2) is shown in Figure 4 in order to illustrate how a holdback scheme is defined. The datagram includes a header 204, a sequence of content frames 208 and a footer 212. The header 204 contains structural parameters for the sequence of content frames 208. In MPEG-2 encoded content, the sequence of content frames is organized to permit random access and synchronize with audio streams. The content frames 208 contain the actual video content (i.e. information) per se and comprise a number of logical units, or video frames of a video sequence. Some of the video frames namely F1, F6 and Fn, are I-frames 216 that represent complete images. The remaining video frames are difference frames (P-frames and B-frames) 220 that represent changes in video information subsequent to the I-frames. By grouping a series of video frames that have a desired level of common visual elements, each video frame subsequent to the first video frame
- 20
- 25
- 30

WO 2007/118311

PCT/CA2007/000612

- 12 -

in the series can be characterized by those portions that differ from the I-frame 216 or the immediately-preceding frame. That is, if there is only movement in the foreground of a series of video frames representing a scene of a movie, the first video frame would include an entire image and subsequent difference
5 frames would only include information about those portions of the image that have changed (objects in the foreground and background revealed by movement of the foreground objects).

[0059] The holdback scheme that is selected takes into consideration the type of encoding used to encode the content. In some cases as mentioned
10 above, it is desirable to select a first holdback that is both small in size and, when removed from the encoded content, damages the quality of the content for the end-user. To damage the content decoded from generic encoded content, it is generally sufficient to remove a plurality of blocks, each comprising one or more bytes, from random locations in the encoded content. However, it
15 is also desired to remove some blocks suitable for embedding the CCTID. In general, all blocks are classified into three types, namely (a) those blocks that cannot change (e.g., file structural parameters); (b) those blocks that are interdependent with other bytes (e.g., cyclic redundancy checks); or (c) those blocks that are free to change. The first holdback comprises the union of an
20 arbitrary selection of as many blocks as are needed of each type to render the encoded content incomplete.

[0060] This is achieved in different ways for different encoding types. For content encoded using the MPEG-2 standard, the defined holdback scheme specifies that a block of one or more bytes in length is to be removed
25 from a random location in the header 204, and a block of one or more bytes in length is to be removed from each I-frame 216. When the first holdback is extracted from MPEG-2 encoded content using this holdback scheme, the quality of the encoded content is sufficiently damaged to render the encoded content incomplete.

30 [0061] Once the holdback scheme has been selected, the candidate locations at which the CCTID may be embedded and the portion(s) of the encoded content to be held back are selected (step 122). The candidate

WO 2007/118311

PCT/CA2007/000612

- 13 -

CCTID embedding locations are determined in accordance with the holdback scheme. It can be desirable to embed the CCTID in the encoded content such that the CCTID cannot be readily removed without damaging the encoded content. In addition, it can be desirable to embed the CCTID in a location

5 where it is relatively undetectable to an end-user in comparison to other locations in the encoded content so that there is little or no significant degradation of the content quality. As the CCTID is embedded into the first holdback, the candidate locations for embedding the CCTID are selected in order to achieve these goals within the constraints of the holdback scheme. As

10 the CCTID is embedded in the first holdback prior to delivery of the first holdback to the end-user, the candidate locations for embedding the CCTID and the portions of the encoded content to be held back are selected such that the candidate CCTID embedding locations are within at least one of the portions of the encoded content to be held back. The candidate CCTID

15 embedding locations are determined in terms of absolute locations in the portions of the encoded content to be held back (i.e., the first holdback). The number of candidate CCTID embedding locations selected well exceeds the number of locations used to embed the CCTID for any one particular end-user. As a result, there will likely be little overlap in the CCTID embedding locations

20 selected for two end-users. In this manner, the embedded CCTIDs are made more resilient against tampering. Further, the set of locations where the CCTID is embedded in the encoded content can be distinct for each end-user, thereby also providing information which can be used to identify the end-user(s) for which the original encoded content was generated.

25 [0062] One potential attack directed to making the CCTID illegible would be to combine two or more separate distinct copies of the same encoded content to identify bits that differ between the copies and either average the differing values or replace them with random bits. In either case, by comparing the resultant copy of the encoded content to the original file, the locations at

30 which the CCTIDs were embedded may be determined.

[0063] It has been found that if the CCTID is embedded into a relatively complex portion of the encoded content, its detectability is decreased. In

WO 2007/118311

PCT/CA2007/000612

- 14 -

images and/or video frames, such areas of relative complexity could be areas of high texture that do not conform to a regular pattern. For example, greyscale values may be shifted up or down a small increment in such areas of relative complexity during the embedding of the CCTID. By performing such analysis
5 prior to CCTID embedding, an in-depth analysis can be performed to locate those areas of the encoded content which are more suitable for receiving the CCTID.

[0064] Once the holdback scheme and candidate CCTID embedding locations have been selected, the first holdback is determined in accordance
10 with the selected holdback scheme and is selected to encompass the candidate CCTID embedding locations. Thereafter, a first holdback reintegration map is determined and registered, along with the candidate CCTID embedding locations (step 123). In particular, the first holdback reintegration map identifies portions of the first holdback to be inserted and
15 locations in the deconstituted encoded content at which the portions of the first holdback are to be reinserted.

[0065] Figure 5 illustrates a portion of an exemplary first holdback reintegration map 224 used in the system 24. The first holdback reintegration map 224 comprises holdback portion insertion parameters 228, each including
20 an insertion location 232 and a holdback portion length 236. Each insertion location 232 identifies where a portion of the first holdback is to be inserted. The corresponding holdback portion length 236 specifies what portion of the first holdback is to be inserted at the insertion location 232. The portion of the first holdback to be inserted at the location specified in the first set of holdback
25 portion insertion parameters 228 is the portion of the first holdback of the holdback portion length 236 at the start of the encoded content. Subsequent holdback portion insertion parameters correspond to subsequent portions of the first holdback commencing where the previous first holdback portions ended.

[0066] Once the first holdback reintegration map and the candidate
30 CCTID embedding locations are determined and registered, the first holdback is extracted from the encoded content (step 124). During extraction of the first holdback, the first holdback portions are extracted and concatenated to form a

WO 2007/118311

PCT/CA2007/000612

- 15 -

contiguous first holdback. The remaining portions of the encoded content are concatenated to form the incomplete encoded content. As a result, the incomplete encoded content is damaged.

[0067] Figure 6 illustrates the extraction of the first holdback determined in accordance with the first holdback reintegration map 224 of Figure 5 for the encoded content of Figure 4. A plurality of segments 248A to 248D are extracted from the encoded content and concatenated to collectively form a first holdback 252. In particular, segment 248A is extracted from the header 204. The remaining segments of the encoded content are concatenated collectively to form incomplete encoded content 256. Demarcation lines 260A to 260D symbolically indicate the former locations of the first holdback segments 248A to 248D respectively in the incomplete encoded content.

[0068] Once the first holdback has been extracted from the encoded content to render it incomplete, the incomplete encoded content is encrypted and a second holdback is extracted from the encrypted incomplete encoded content (step 130) thereby to yield the deconstituted encoded content.

[0069] Figure 7 illustrates the method of encrypting the incomplete encoded content and extraction of the second holdback therefrom at step 130. Initially, an encryption key is randomly generated and used to encrypt the incomplete encoded content 256 prior to storage on the optical media disk 28 (step 131). The incomplete encoded content 256 is encrypted using a symmetric cipher in cipher block chain mode. In cipher block chain mode, the encryption of any particular block of data is dependent on the encrypted value of all the preceding data. As a result, the omission of any segment of the encrypted encoded content makes decryption of the encoded content infeasible. In addition, the name and extension of the file are changed so as to obscure the encoded content. Once encrypted, the second holdback is extracted from the encrypted incomplete encoded content (step 132). The second holdback is an arbitrarily selected set of data blocks that are extracted from the encrypted incomplete encoded content. A second holdback reintegration map is generated to indicate how the second holdback is to be reintegrated with the encrypted incomplete encoded content to restore the

WO 2007/118311

PCT/CA2007/000612

- 16 -

encrypted incomplete encoded content to its original form. By extracting the second holdback from the encrypted incomplete encoded content, the deconstituted encoded content is formed; that is, it is damaged and cannot be restored to its original form without the extracted data.

5 [0070] With the deconstituted encoded content formed, the first holdback reintegration map, the candidate CCTID embedding locations, the encryption key, the first holdback, the second holdback and the second holdback reintegration map are stored (step 133). In particular, the first holdback reintegration map, the candidate CCTID embedding locations and the
10 first holdback are stored in the content completion database 48 and the encryption key, the second holdback and the second holdback reintegration map are stored in the security database 52. The second holdback, the second holdback reintegration map, the encryption key, the first holdback and the first holdback reintegration map are all that is required by the PC 36 to reconstruct
15 the deconstituted encoded content. The candidate CCTID embedding locations enable the authorization server 40 to select locations and embed the CCTID in the first holdback prior to its delivery to the PC 36.

[0071] Following the above, the encrypted deconstituted encoded content is placed onto the optical media disk 28 and distributed to the end-user
20 32 (step 140). Distribution of the optical media disk 28 can be performed in a number of ways, including its inclusion with periodicals, manual distribution at an event, mail distribution via a subscriber list, etc. The optical media disk 28 includes the deconstituted encoded content and a content access application for reconstituting the encoded content. In addition, the optical media disk 28 is
25 provided with a media identifier. The media identifier is registered with the authorization server 40 along with a list of the deconstituted encoded content on the optical media disk 28.

[0072] In order for the end-user 32 of the PC 36 to be able to enjoy any of the deconstituted encoded content on the optical media disk 28, the end-
30 user 32 must submit an authorization request to the authorization server 40 (step 150). Generation of the authorization request occurs via the content access application. This application automatically launches when the optical

- 17 -

media disk 28 is inserted into PC 36 and permits the end-user 32 to obtain authorization for accessing the deconstituted encoded content stored on the optical media disk 28. In order to submit the authorization request, the end-user 32 logs in to the authorization server 40 by entering the End-User ID and password at a login screen.

5 [0073] The steps performed during the authorization process are illustrated in Figure 8. Initially, the content access application generates a list of previously reconstituted encoded content stored on the PC 36 (step 151). The content access application then queries the optical media disk 28 to obtain the media identifier, and transmits the media identifier to the authorization server 10 40 via the Internet 44, along with the End-User ID to determine what deconstituted encoded content is available on the optical media disk 28, and what deconstituted encoded content on the optical media disk 28 the end-user 32 is currently licensed to access. The authorization server 40 then responds 15 with a list identifying the deconstituted encoded content that is available on the optical media disk 28, along with the deconstituted encoded content the end-user 32 is currently licensed to access.

[0074] Upon obtaining the list of deconstituted encoded content, the content access application parses the list and presents the parsed list of 20 deconstituted encoded content available on the optical media disk 28 to the end-user 32 (step 152). The parsed list of deconstituted encoded content identifies previously reconstituted encoded content that already exists and is stored by the PC as well as deconstituted encoded content that the end-user 32 is currently eligible to license.

25 [0075] Figure 9 illustrates an exemplary content selection window 300 presenting a parsed list of deconstituted encoded content 304, music videos in this case, that is available on the optical media disk 28, along with the cost of a license to access the deconstituted encoded content. Check boxes beside each music video permit the end-user 32 to indicate the encoded content to be 30 accessed. An accept button 308 and a cancel button 312 permit acceptance or cancellation of any selections made by the end-user 32. The list of music videos is sorted by genre and artist.

WO 2007/118311

PCT/CA2007/000612

- 18 -

[0076] Referring to Figures 8 and 9, when the end-user 32 selects deconstituted encoded content 304 using the accept button 308, the content access application receives the selection of deconstituted encoded content that the end-user 32 wishes to access (step 153). Following this, an authorization confirmation dialog box (not shown) appears, prompting the end-user 32 to confirm that credit is to be spent on the selected deconstituted encoded content. Upon confirmation, selection of deconstituted encoded content is finalized.

[0077] The content access application in turn communicates with the authorization server 40 requesting access to the selected encoded content (step 154). The request from the PC 36 includes the unique End-User ID of the end-user 32 and the Content ID(s) for the selected deconstituted encoded content.

[0078] The end-user 32 is then authorized to access the requested encoded content by the authorization server 40 (step 155). At this point, a number of events occur. The authorization server 40 receives the request from the PC 36 to access encoded content and authorizes the end-user 32 for the selected encoded content. The End-User ID transmitted with the request to access encoded content is used to retrieve end-user information from the end-user database 48. The end-user database 48 contains information for each end-user 32 including information required to process a payment, a transaction history and a credit balance. In order to obtain authorization to access encoded content, the end-user purchases credit via a website operated by the authorization server 40. The end-user database 48 is then updated to reflect the new credit. If the end-user has sufficient credit to purchase access to the selected encoded content (that is, a license), a transaction history is updated, and the end-user's credit balance is debited the appropriate amount for the selected encoded content. If the end-user does not have sufficient credit to purchase access to all of the selected encoded content, the authorization server 40 directs the content access application to display a message that insufficient credit is available, with a link to a page where the end-user can purchase additional credit.

- 19 -

[0079] Once authorization has been granted, the authorization server 40 generates a customized first holdback (step 160). The authorization server 40 generates a CCTID for the license by calculating a collision-resistant hash of the End-User ID, the Content ID and license information, and then stores the
5 CCTID in the license database 60.

[0080] The authorization server 40 uses the Content ID to retrieve the corresponding first holdback, first holdback reintegration map and the candidate CCTID embedding locations from the content completion database 52. In addition, the authorization server 40 retrieves the encryption key, the second
10 holdback and the second holdback reintegration map from the security database 56. The authorization server 40 then selects a subset of the candidate CCTID embedding locations for the particular end-user and registers the selected subset in the license database 60. The authorization server 40 then embeds the CCTID into the first holdback at the selected candidate
15 CCTID embedding locations. During embedding of the CCTID in the first holdback, error-checking codes frequently used in media files to identify corrupt frames are adjusted to reflect the embedding of the CCTID. MPEG-2 for instance allows a cyclic redundancy check.

[0081] The customized first holdback is then bundled by the
20 authorization server 40 with the first holdback reintegration map, the encryption key, the second holdback and the second holdback reintegration map to generate a customized reconstitution package. The authorization server 40 then transmits the customized reconstitution package to the PC 36.

[0082] When the PC 36 receives the customized reconstitution package
25 containing the customized first holdback, the first holdback reintegration map, the encryption key, the second holdback and the second holdback reintegration map (step 170), the PC 36 parses the customized reconstitution package to obtain the encryption key, the second holdback and the second holdback reintegration map. The PC 36 then reintegrates the second holdback in
30 accordance with the second holdback reintegration map and decrypts the incomplete encoded content using the encryption key provided in the reconstitution package (step 180). The PC 36 then reconstitutes the

- 20 -

Incomplete encoded content by inserting the first holdback into the incomplete encoded content using the first holdback reintegration map (step 190). The reconstituted or restored encoded content is thereafter stored on non-volatile storage of the PC 36 for accessing at a later time.

5 [0083] Once the deconstituted encoded content has been reconstituted, the encoded content is generally as it was before it was deconstituted, with the exception that the CCTID has been steganographically embedded in the encoded content such that it is generally undetectable by the end-user 32.

[0084] As will be appreciated, by distributing encoded content in a
10 deconstituted form, the end-user is not provided with all of the data required to enable decoding of the encoded content prior to authorization. By cipher block chain encrypting the incomplete encoded content and extracting a second holdback, decrypting the incomplete encoded content to its original form is made infeasible. Further, as the end-user does not possess all of the encoded
15 content prior to reconstitution, the end-user cannot reconstruct the entire encoded content from the deconstituted encoded content. Only when the deconstituted encoded content is reintegrated with the second holdback, decrypted and reconstituted by integrating the first holdback can it be properly decoded and enjoyed by the end-user.

20 [0085] Due to the manner in which the CCTID is embedded in the encoded content, wherein actual content is changed, the CCTID in the resultant encoded content is resilient to a decoding/re-encoding attack. Further, as the CCTID is embedded in a set of locations that is distinct for each end-user, the encoded content is resilient to combination attacks where two legitimate copies
25 of the encoded content licensed to two end-users are combined in some manner, and generally can permit identification of both end-users.

[0086] It is of interest in many cases for administrators of networks, either physical such as a local area network, or virtual such as a file-sharing network operated over the Internet, to ensure that the encoded content being
30 shared on their networks is properly licensed. For such cases, a method and apparatus are provided to enable the authentication of encoded content.

- 21 -

[0087] Figure 10 shows a system 400 that is similar to the system 24 of Figure 1 and additionally comprises an arbiter computer 404 and a remote personal computer ("remote PC") 408 operated by a remote user 412. Both the arbiter computer 404 and the remote PC 408 are in communication with the PC 36. The end-user 32 of the PC 36 can offer to make available encoded content stored thereon to other users via a file-sharing network operated via the arbiter computer 404. Upon launch of a file-sharing application, PC 36 communicates to the arbiter computer 404 a list of encoded content that the end-user 32 has selected for sharing. This list is combined with lists of encoded content made available by other end-users to form an aggregate list of encoded content available on the file-sharing network. The remote user 412 can connect to the file-sharing network by executing a file-sharing application on the remote PC 408. The remote PC 408 connects to the arbiter computer 404 and communicates a list of encoded content that the remote user 412 has selected for sharing. In order to locate and retrieve encoded content from the file-sharing network, the remote user 412 specifies query criteria via the file-sharing application which, in turn, queries the arbiter computer 404. The arbiter computer 404 responds with a list of encoded content matching the query of the remote user 412. Upon selection by the remote user 412 of encoded content to be downloaded, the remote PC 408 requests the selected encoded content from the arbiter computer 404. The arbiter computer 404 then requests the selected encoded content from the PC 36 for transmission to the remote PC 408.

[0088] Figure 11 illustrates the steps performed in order to authenticate and control the distribution of encoded content via the system 400. The arbiter computer 404 is configured to monitor and control traffic and storage of encoded content on the file-sharing network. Upon receiving the encoded content (step 510), the arbiter computer 404 determines whether the encoded content is branded (step 520). If the encoded content is branded, the arbiter computer 404 determines if the branded encoded content is authenticated and if distribution permissions for the branded encoded content are available (step 530). At step 530, if the arbiter computer 404 determines that the encoded

- 22 -

content is not authentic or that the encoded content may not be redistributed, the arbiter computer 404 prohibits transmission of the encoded content (step 540). If, instead, the arbiter computer 404 determines that the encoded content is authenticated and redistribution is permitted, or if the encoded content is
5 determined not to be branded at step 520, the arbiter computer 404 permits transmission of the encoded content (step 550).

[0089] Figure 12 illustrates the steps performed in order to determine whether encoded content is authenticated. Initially, the arbiter computer 404 determines whether the brand and the Content ID appear to be valid (step 531).
10 If it appears that a brand was embedded in the encoded content, and the brand and/or the Content ID have been tampered with, the encoded content is deemed inauthentic (step 532). In this case, as the encoded content is deemed inauthentic, distribution of the encoded content is not permitted. If, instead, the brand and the Content ID appear to be valid at step 531, the arbiter
15 computer 404 then performs a hash calculation on the encoded content (step 533). The arbiter computer 404 provides the result to the authorization server 40 along with the Content ID parsed from the encoded content (step 534). Upon receiving the result from the arbiter computer 404, the authorization server 40 checks the license database 60 for a record including the Content ID and the hash calculation result
20 and the hash calculation result (step 535). If the authorization server 40 does not find a record including the Content ID and the hash calculation result registered in the license database 60, the encoded content is deemed to be inauthentic. However, if the authorization server 40 finds a record including the Content ID and the hash calculation result in the license database 60, the
25 encoded content is deemed to be authenticated and the authorization server 40 retrieves any permissions associated with the license from the record. Once the encoded content is determined to be authentic or inauthentic, the authorization server 40 responds back to the arbiter computer 404 with the result including any permissions which affect distribution (step 536).

30 [0090] In some cases, it is desirable to pre-customize portions of the encoded content that are distributed to end-users and still retain the ability to create a distinct version of the encoded content for a particular end-user.

- 23 -

- [0091]** Turning now to Figure 13, an alternative method of delivering encoded content is shown and is generally identified by reference numeral 600. For ease of discussion, the method 600 will be described with respect to video content that is encoded using the MPEG-2 standard. The method 600
- 5 commences with the analysis of the encoded content (step 610). During analysis of the encoded content as shown in Figure 14, the encoding type that is used to encode the content is determined (step 611). A modification scheme corresponding to the encoding type determined at step 611 is then selected (step 612). The locations of where modifications may be made to the encoded
- 10 content in accordance with the modification scheme depend on the type of encoding employed. For example, in the case of MPEG-2-encoded video, modifications are made to visual content and/or parameters of the I-frames and/or difference frames. These regions of the encoded content are identified by parsing the encoded content, and are flagged at this stage.
- 15 **[0092]** The encoded content is then analyzed to determine candidate locations for modifications (step 613). As it is desirable not to significantly affect the experience of an end-user, the encoded content is parsed and analyzed to identify candidate locations therein where modifications may be made in a manner that is not readily detectable to the end-user. These regions
- 20 of the encoded content are flagged as candidate locations for modifications.
- [0093]** In particular, the modification scheme used to modify the encoded content in this embodiment is based on adjustment of the bias of frames. The bias of the frames provides a baseline against which image data is compared in order to characterize it during encoding. Accordingly, video
- 25 frames are identified as candidate locations for modifications.
- [0094]** Referring again to Figure 13, once the encoded content has been analyzed, a brand and the Content ID are embedded into the encoded content (step 620). The encoded content is then segmented (step 630). The size of the segments is generally pre-determined, but can be modified depending on
- 30 the position of candidate locations in the encoded content in order to ensure that each segment contains a desired number of candidate locations. By providing a plurality of candidate locations for each segment, the variations

WO 2007/118311

PCT/CA2007/000612

- 24 -

possible of each segment through modification are numerous. Further, the probability of detection of such modifications by an end-user can be decreased.

[0095] Figures 15A and 15B represent encoded content 800 before and after segmentation respectively. In the illustrated example, the encoded

5 content 800 has been divided into forty (40) segments 804. Each segment 804 includes a set of video frames.

[0096] Referring again to Figure 13, when the encoded content has been segmented, a set of distinct instances of each segment is generated (step 640). Each segment instance is generated by modifying the segment in at least one
10 of the candidate locations identified in step 610. As there are a number of candidate locations for each segment, the segment instances are readily distinguished from each other.

[0097] In order to modify a segment to generate a particular segment instance thereof, the bias of one or more frames of the segment is modified.
15 As there are a number of frames in each segment, and as the bias of the frames can be modified to varying degrees, a number of combinations of distinct segment instances can be generated for each segment.

[0098] Figure 16 illustrates the generation of a plurality of segment instances 808 for each segment 804 of the encoded content 800. As shown,
20 ninety-nine (99) segment instances 808 of each of the forty (40) segments 804 are generated. Segment instance 808a is the first instance of a second segment 804a of the encoded content. Segment instance 808b is the fourth instance of a second segment 804b of the encoded content. Each segment instance is represented by two (2) coordinates 812. The first coordinate refers
25 to the number of the segment to which the segment instance is related. The second coordinate is an identifier for the particular instance of the segment. As segment instances are generated for a segment, they are numbered sequentially, starting at 1.

[0099] Once the segment instances have been generated, hash
30 calculations are performed on each of the segment instances and the results are stored in the license database 56. These results are used to authenticate the encoded content in a manner similar to that previously described.

- 25 -

- [00100]** Referring again to Figure 13, once the plurality of segment instances 808 has been generated, first holdbacks are extracted from each segment instance (step 650). Each first holdback refers to a portion of encoded content that is withheld so as to not provide all of the encoded content, and to damage it. In this particular case, a first holdback is a portion of a segment instance that is arbitrarily selected and extracted from the segment instance. The first holdback extracted from a segment instance can be a single, contiguous set of bits or, alternatively, can be a number of bits extracted from a variety of positions within the segment instance.
- 5
- [00101]** During extraction of the first holdbacks from the segment instances, the first holdbacks are registered in the content completion database 52, along with the Content ID associated with the encoded content, and the coordinates associated with the particular segment instance from which each first holdback has been extracted. In addition, first holdback reintegration maps for reintegrating each first holdback into the corresponding segment instance are also stored in the content completion database 52. Each first holdback reintegration map specifies how the corresponding first holdback is to be reintegrated with the segment instances. In particular, each first holdback reintegration map indicates a set of positions in the corresponding segment instance, and the length of the portion of the corresponding first holdback to be inserted at the particular positions.
- 10
- [00102]** Figure 17 illustrates the plurality of segment instances 808 after extraction of first holdbacks 812. As shown, the first holdbacks 812 can be extracted from differing positions in each of the segment instances 808. The segment instances represent incomplete encoded content as they do not contain all of the information needed to decode the encoded content.
- 15
- [00103]** After extraction of the first holdbacks, the incomplete segment instances 808 are each cipher block chain encrypted using separate symmetric encryption keys and second holdbacks are extracted (step 660) to yield the deconstituted encoded content. Second holdback reintegration maps are generated to specify how the second holdbacks are to be reintegrated with the encrypted incomplete segment instances. As a result of the encryption, the
- 20
- 25
- 30

WO 2007/118311

PCT/CA2007/000612

- 26 -

- segment instances are put into a non-standard file format and are given names and extensions that do not enable them to be readily associated with the encoded content to which they are related. By extracting the second holdbacks from the encrypted incomplete segment instances, their decryption without the
- 5 second holdbacks is made infeasible. The encryption keys, the second holdbacks and the second holdback reintegration maps are then registered by the authorization server 40 in the security database 56, along with the Content ID of the encoded content and the two coordinates of the segment instances to which they correspond.
- 10 [00104] Referring back to Figure 13, with the deconstituted encoded content available, it is placed onto an optical media disk and distributed to the end-user in the manner previously described (step 670). When an end-user requests access to the encoded content and is authorized (step 680), the authorization server 40 generates a distinct content map for the license and
- 15 registers it in the license database 48 (step 690). The content map is a mapping of the segment instances to direct the PC 36 to recombine certain segment instances to generate a copy of the encoded content that is distinct for the particular end-user, albeit incomplete due to the absence of the first holdbacks.
- 20 [00105] An exemplary content map 900 is shown in Figure 18. The content map is a sequenced list of segment instances that are to be combined to form a distinct customized copy of the encoded content. The order of the elements correspond with the segments to which they apply and each element indicates which instance is to be used for the particular segment. In content
- 25 map 900, the first element, "3", indicates that the third segment instance is to be used for the first segment of the encoded content. Likewise, the second segment instance is to be used for the second segment of the encoded content as indicated by the "2" in the second position, and so on.
- [00106] The authorization server 40 registers the license and the content
- 30 map in the license database 60. The authorization server 40 registers the End-User ID, the Content ID of the newly licensed encoded content, the content

- 27 -

map and a set of license data. The license data can include particulars of the license terms, the date the license was issued, etc.

[00107] The authorization server 40 then retrieves the first holdbacks, the first holdback reintegration maps, the encryption keys, the second holdbacks
 5 and the second holdback reintegration maps that correspond to the particular segment instances selected for the end-user 32 from the content completion database 52 and the security database 56 respectively, and creates and transmits a custom reconstitution package to the PC 36 (step 700). The custom reconstitution package includes the content map.

10 [00108] Upon receiving the custom reconstitution package, the PC 36 reconstitutes the encoded content (step 710). Figure 19 better illustrates the process of reconstituting the encoded content. When the PC 36 receives the content map, the encryption keys, the second holdbacks, the second holdback reintegration maps, the first holdbacks and the first holdback reintegration maps
 15 (step 711), the PC 36 reintegrates the second holdbacks with the encrypted incomplete segment instances on the optical media disk 28 specified by the content map using the second holdback reintegration maps (step 712). The PC 36 then decrypts the specified incomplete segment instances using the encryption keys (step 713). Next, the PC 36 reintegrates the first holdbacks
 20 into the specified incomplete segment instances in accordance with the first holdback reintegration maps (step 714). Upon reintegration of the first holdbacks, the encoded content is reassembled from the segment instances in accordance with the content map (step 715) i.e. is reconstituted. The reconstituted encoded content is thereafter stored on non-volatile storage of the
 25 PC 36 for accessing at a later time.

[00109] Figure 20 illustrates exemplary customized encoded content 1000. In this example, the completed encoded content 1000 is generated from the segment instances of Figure 16 and the content map of Figure 18.

[00110] In an alternative embodiment, the encoded content in its entirety
 30 can be stored by the authorization server 40. Upon authorization access being granted to an end-user, a distinct customized copy of the encoded content is generated and made available to the user. The customized copy is generated

WO 2007/118311

PCT/CA2007/000612

- 28 -

at the time of authorization. The customization can be steganographically placed in the encoded content. In this case, pre-analysis to identify areas of the encoded content that are suitable for customization can enable rapid steganographic customization. The customization can also provide an identifier
5 of the license via visual, audio or other suitable markings. The end-user can be provided with an interface for selecting encoded content that they wish to access, such as a webpage. Upon selecting encoded content that the end-user wishes to access, a customized copy of the selected encoded content is generated and delivered to the end-user.

10 [00111] In the above-described embodiments, if the encoded content is copied by or from the licensee, the customization(s) to the encoded content can permit proving that the copy was made from an instance of the encoded content licensed to that particular end-user.

[00112] While embodiments have been described above with reference to
15 the figures, those of skill in the art will appreciate that alternatives are available. For example, while deconstituted encoded content is described as being distributed on an optical media disk, other methods of distribution can be employed. For example, the deconstituted encoded content can be distributed electronically via a content distribution network and downloaded to the PC.
20 Deconstituted encoded content can be pre-delivered and cached on a PC prior to receiving an authorization request to access the particular encoded content.

[00113] The decoder may be made aware of the customization and the end-user to which the encoded content is licensed (and can be made to permit/deny use if the registered end-user of the application is not the end-user
25 licensed for the encoded content).

[00114] Data can be extracted and reintegrated any number of times and in any number of manners. It may be desirable simply to cipher block chain encrypt the encoded content and then extract data therefrom without extracting a first holdback from the encoded content prior to encryption. Alternatively,
30 data can be extracted from the encoded content without the encoded content being subsequently encrypted. Instead of specifying absolute addresses where the extracted data is to be reintegrated, relative addresses can be used.

WO 2007/118311

PCT/CA2007/000612

- 29 -

[0100] The first and second holdbacks can be extracted from the encoded content in other manners. For example, in extracting a holdback, an XOR function can be applied to portions of the encoded content, thereby leaving the size of the encoded content intact. In this case, reintegration of the holdback would comprise performing an XOR function on those portions of the encoded content specified by the reconstituted package.

[0101] Customization of encoded content can be performed by the PC or other locations. The authorization server can provide instructions to the PC to customize encoded content available to the PC. The instructions can be encrypted to retain the confidentiality of the information. The application executing on the PC to perform such customizations can be made robust so as to resist tampering and the encoded content can be stored in a secure manner prior to customization. In the scenario where there is a trusted client with knowledge of the embedding process that is licensed to generate copies of encoded content for end-users, unregistered copies may appear for authentication by the arbiter computer. In these cases, the arbiter computer can determine the CCTIDs generated and register each copy of the encoded content as it becomes aware of them. The trusted client embeds both the identity of the trusted client and the end-users, along with a serial number so that the number of copies generated can be counted. In this manner, the encoded content generated by the trusted client can be tracked.

[0102] In some cases, it may be desirable to simply deconstitute the encoded content and not customize it. The reconstitution data in this scenario can be delivered upon licensing of the encoded content.

[0103] The CCTID can be digitally signed by the authorization server with standard private encryption key technology, enabling authentication of the CCTID. Instead of customizing the encoded content with a CCTID or the like, the transaction particulars can be embedded directly into the holdback and/or the encoded content itself.

[0104] By generating a hash calculation for the particular encoded content and generating and storing the results for each end-user in the license database as encoded content is licensed, the authorization server can quickly

WO 2007/118311

PCT/CA2007/000612

- 30 -

determine whether the encoded content upon which the hash calculation was performed matches the end-user. If the End-User ID and Content ID are not found in a record of the license database, or if the result of the hash calculation provided by the application does not match that for the End-User, the

5 authorization server can generate a result indicating that the end-user is not authorized to access the encoded content and return the result to the application.

[0105] It can be desirable in some cases to perform hash calculations only on subsets of the encoded content where the information gathered by
10 performing hash calculations on a subset of the encoded content is sufficient for the intended purpose.

[0106] Other characteristics of the encoded content can be modified in order to differentiate the encoded content. For example, the keystoneing of video frames can be altered slightly.

15 [0107] Another method of modifying encoded content to create distinct instances while not significantly changing the end-user experience is by looking for and altering data in the encoded content that is superfluous. Many encoding standards call for compression algorithms whereby the resulting encoded content includes superfluous data. If little or no superfluous data
20 exists in encoded content, it can be desirable to insert some otherwise superfluous space into which an identifier can be embedded.

[0108] Encoded content is usually compressed by the encoder, which makes choices affecting compression ratios when encoding the content, (and this also trades off quality when using lossy compression). Since these choices
25 may all be made at compression time, it is possible to select the compression ratio such that there is always sufficient superfluous space remaining for customization.

[0109] Further, it is possible to embed a brand and Content ID into and/or customize encoded content by embedding one or more robust
30 watermarks in the encoded content at compression time. This information can then be read from the encoded content by "partially trusted" agents, such as peer-to-peer ("P2P") file-sharing services and the like. The information in the

- 31 -

watermark can be date-stamped and digitally signed. A robust digital watermark may be embedded by any algorithm that satisfies an arbitrary set of requirements for robustness. For example, exemplary robust watermarking approaches are described in the document entitled "Digital Watermarking

- 5 Schemes for Multimedia Authentication", Chang-Tsun Li, Department of Computer Science, University of Warwick, 15 February 2004.

[0110] Other methods of inserting brands and/or Content IDs will occur to those skilled in the art. For example, the content can be altered in a manner that is relatively innocuous to an end-user by modifying least-significant bits
10 within the file to satisfy a parity test.

[0111] For video or still-image content, areas of relatively high complexity provide camouflage to minor modifications made. Similarly, in audio content, portions of relatively complex sounds enable modifications to be made without significantly impacting the experience for an end-user.

- 15 [0112] Where the encoded content is customized to embed a CCTID or the like, the customization may include redundant data to identify the CCTID so that the CCTID is still legible if some of the customizations in the encoded content are tampered with.

[0113] Where the encoded content is divided into a set of segments,
20 distinct instances can be generated for a subset of the segments, with the remainder of the segments being used in their original form. In this case, content is reassembled from the segments used in their original form together with one of the instances corresponding with each segment for which instances were generated. In this manner, storage requirements for the unassembled
25 components (the original segments and segment instances) can be reduced.

[0114] While the method and system disclosed above are described with respect to monitoring file-sharing traffic, encoded content registered in storage or communicated via another type of network can be scanned and authenticated in a similar manner.

- 30 [0115] Although preferred embodiments have been described with reference to the drawings, those of skill in the art will appreciate that variations

WO 2007/118311

PCT/CA2007/000612

- 32 -

and modifications may be made without departing from the
thereof as defined by the appended claims.

scope

What is claimed is:

1. A method of delivering encoded content, comprising:
extracting, using a computing device, a selected subset of data of said encoded
content that is to be delivered, thereby rendering the remainder of the encoded content
5 damaged;
encrypting said damaged remainder of the encoded content using the computing
device;
prior to authorization, distributing said encrypted damaged remainder of the
encoded content; and
10 upon authorization, transmitting the extracted selected subset of data to one or
more recipients of the distributed damaged remainder of the encoded content for
reintegration with said distributed damaged remainder of the encoded content.
2. The method of delivering encoded content of claim 1, further comprising:
modifying the extracted selected subset of data, using the computing device, such
15 that for each recipient a distinct instance of said encoded content is created when the
extracted selected subset of data is reintegrated with said distributed damaged remainder
of the encoded content.
3. The method of delivering encoded content of claim 2, wherein during said
modifying the extracted selected subset of data is modified to include steganographically-
20 embedded information.

4. The method of delivering encoded content of any one of claims 2 to 3, wherein during said modifying the extracted selected subset of data is modified to include information identifying an authorized end-user.
5. The method of delivering encoded content of claim 4, wherein said information identifies a record in a license database identifying said authorized end-user.
6. The method of delivering encoded content of claim 5, wherein said license database also identifies distribution rights for said encoded content that the end-user is authorized to restore.
7. The method of delivering encoded content of claim 2 wherein during said modifying the extracted selected subset of data is modified to include information identifying license terms.
8. The method of delivering encoded content of any one of claims 1 to 7, wherein said encrypting utilizes a cipher block chain method.
9. The method of delivering encoded content of any one of claims 1 to 8, wherein said selected subset of data comprises a plurality of discrete portions of said encoded content.
10. The method of delivering encoded content of any one of claims 1 to 9, wherein said distributing comprises distributing said damaged remainder of the encoded content on digital storage media.

11. The method of delivering encoded content of any one of claims 1 to 9, wherein said distributing comprises distributing said damaged remainder of the encoded content over a communication network.
12. The method of delivering encoded content of any one of claims 1 to 11, wherein
5 said transmitting comprises transmitting the extracted selected subset of data over a communication network.
13. The method of delivering encoded content of any one of claims 1 to 7, further comprising, prior to said distributing:
extracting a second subset of selected data from said encrypted damaged
10 remainder of the encoded content; and
upon authorization, transmitting the extracted selected second subset of data for reintegration with said distributed damaged remainder of the encoded content.
14. The method of delivering encoded content of claim 13 wherein said transmitting is performed in response to receipt of an authorized access request.
- 15 15. The method of delivering encoded content of claim 13 wherein the extracted selected subset of data comprises discrete data segments of said encoded content and a reintegration map specifying locations of said damaged remainder of the encoded content into which said data segments are to be inserted.
- 20 16. The method of claim 15 wherein said extracted second subset of selected data comprises discrete data segments of said encrypted damaged remainder of the encoded content and a reintegration map specifying locations of said encrypted damaged remainder of the encoded content into which said discrete segments are to be inserted.

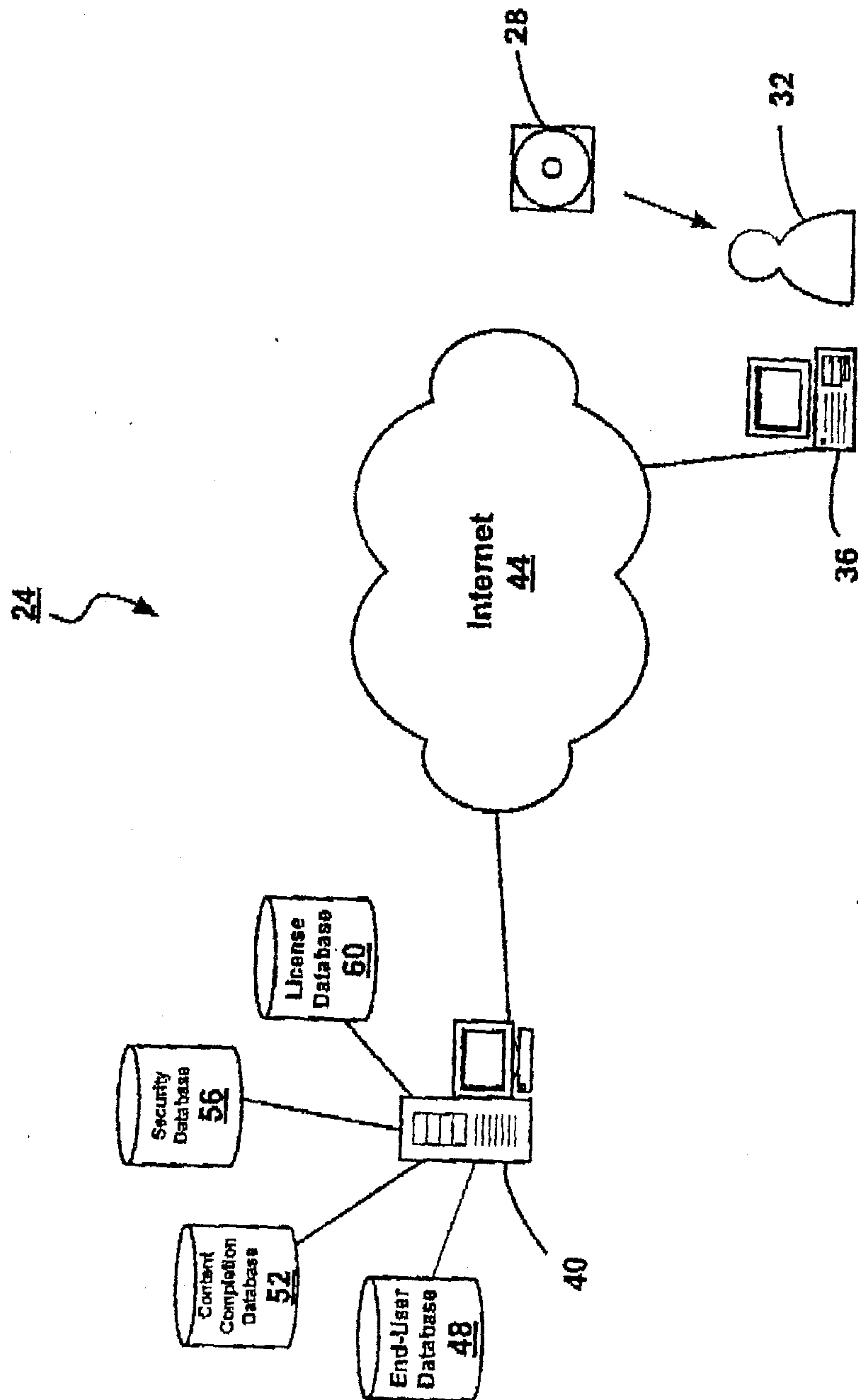
17. The method of delivering encoded content of claim 1 further comprising:
prior to said extracting, encrypting said encoded content using the computing
device.
18. The method of delivering encoded content of claim 16 wherein said extracted
5 selected subset further comprises an identifier unique to said end-user.
19. The method of delivering encoded content of claim 1 wherein during the
extracting, the selected subset of data is extracted from the encoded content using an
extraction scheme that is selected based on encoded content type.
20. The method of delivering encoded content of claim 1 wherein said extracting
10 comprises extracting a plurality of selected data segments from the encoded content.
21. The method of delivering encoded content of claim 20 wherein the data segments
are extracted from various locations in the encoded content.
22. The method of delivering encoded content of claim 1 further comprising
concatenating said damaged remainder of the encoded content after said extracting and
15 prior to said distributing.
23. The method of delivering encoded content of claim 1 further comprising:
upon authorization, transmitting reconstitution instructions specifying at least one
location within said distributed damaged remainder of the encoded content into which
said extracted selected subset of data is to be inserted.

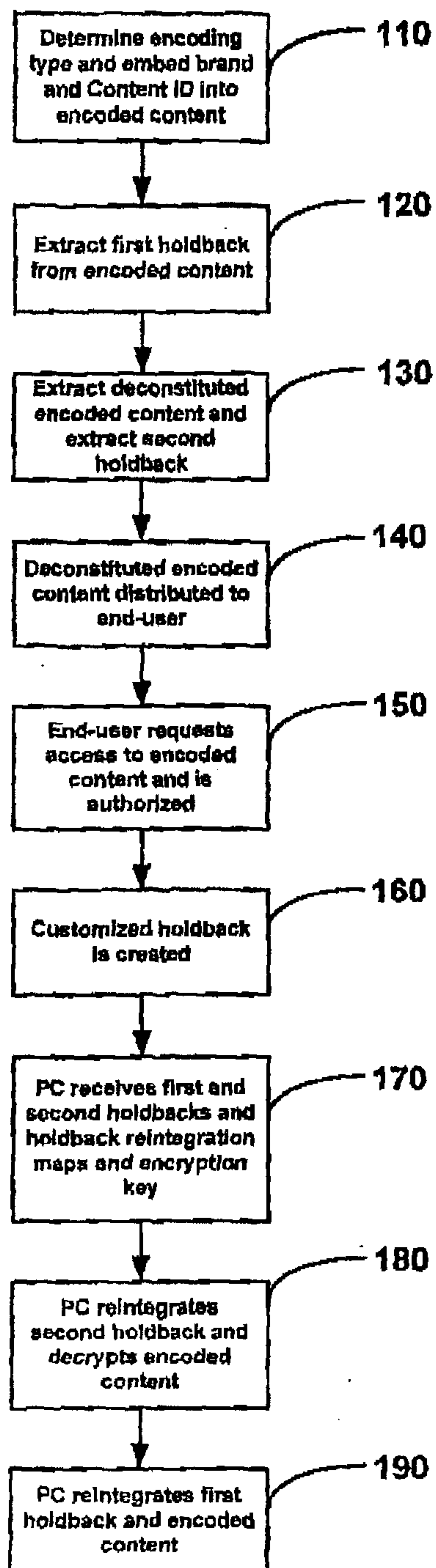
24. A system for delivering encoded content, comprising:

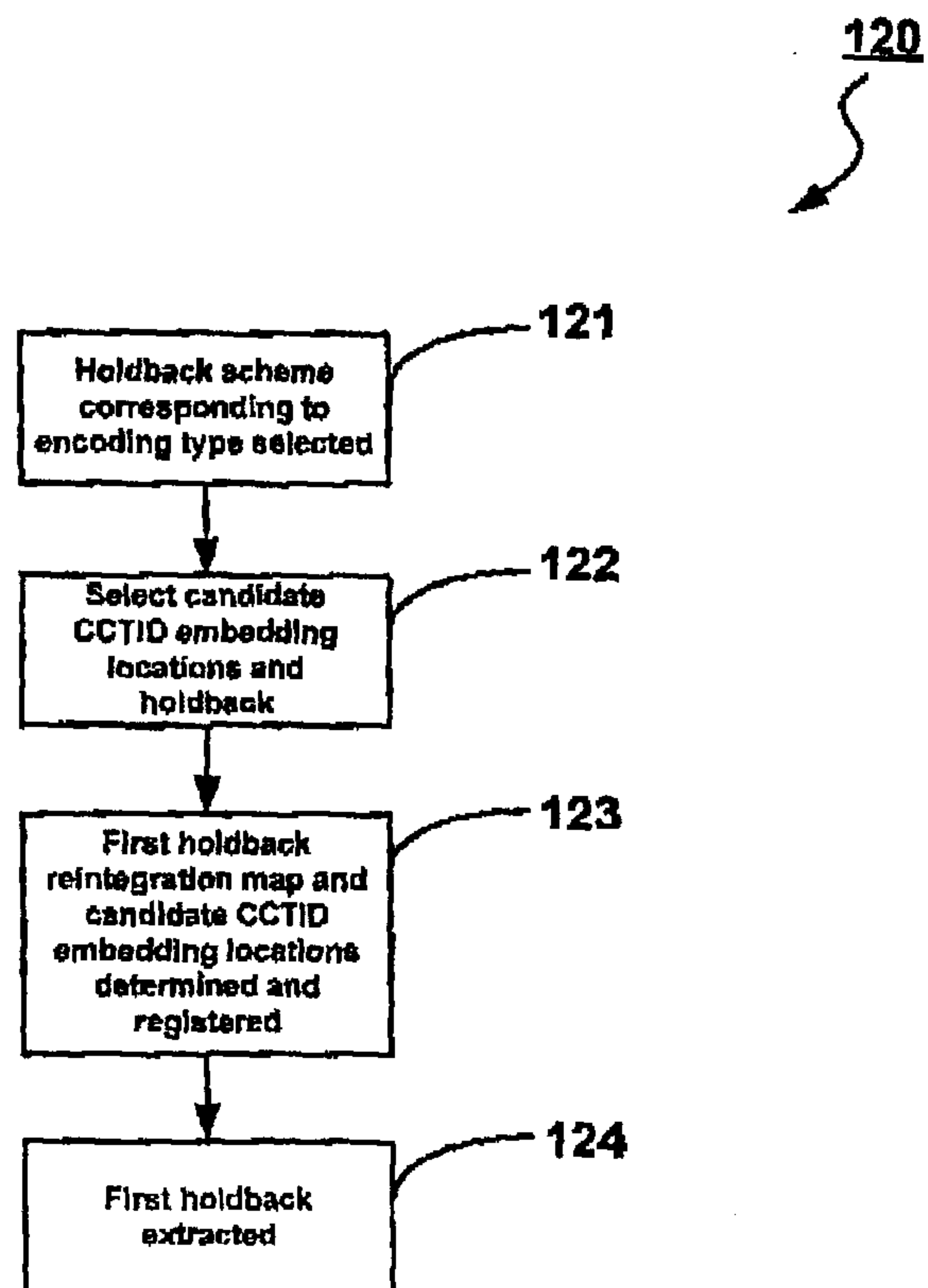
at least one server extracting a selected subset of data of stored encoded content that is to be delivered to an end-user thereby damaging said encoded content, said at least one server preparing said damaged encoded content for distribution to said end-user in response to an authorized request, said at least one server further transmitting said extracted selected subset of data to said end-user to enable said end-user to reintegrate said the extracted selected subset of data with said damaged encoded content thereby to restore said encoded content.

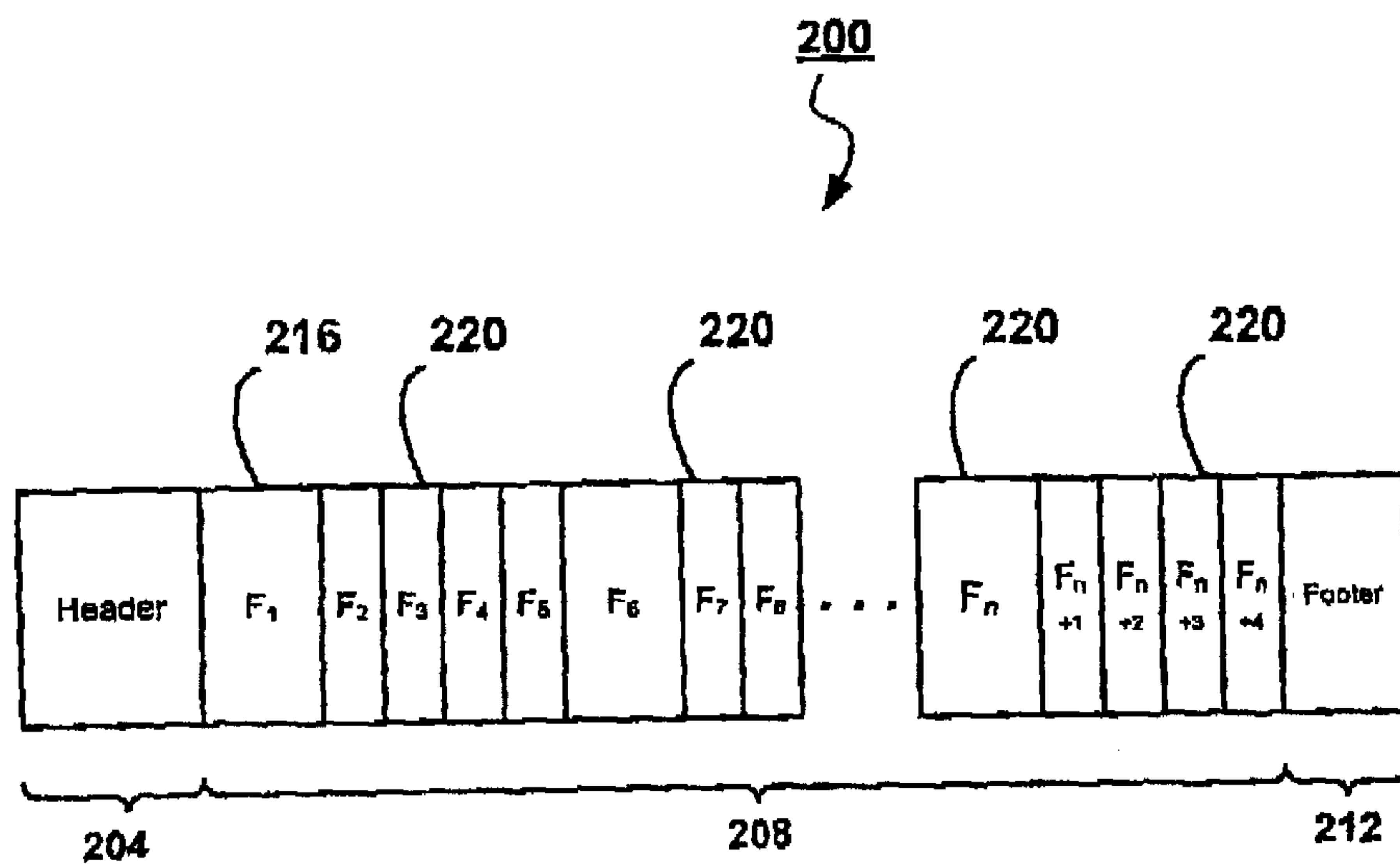
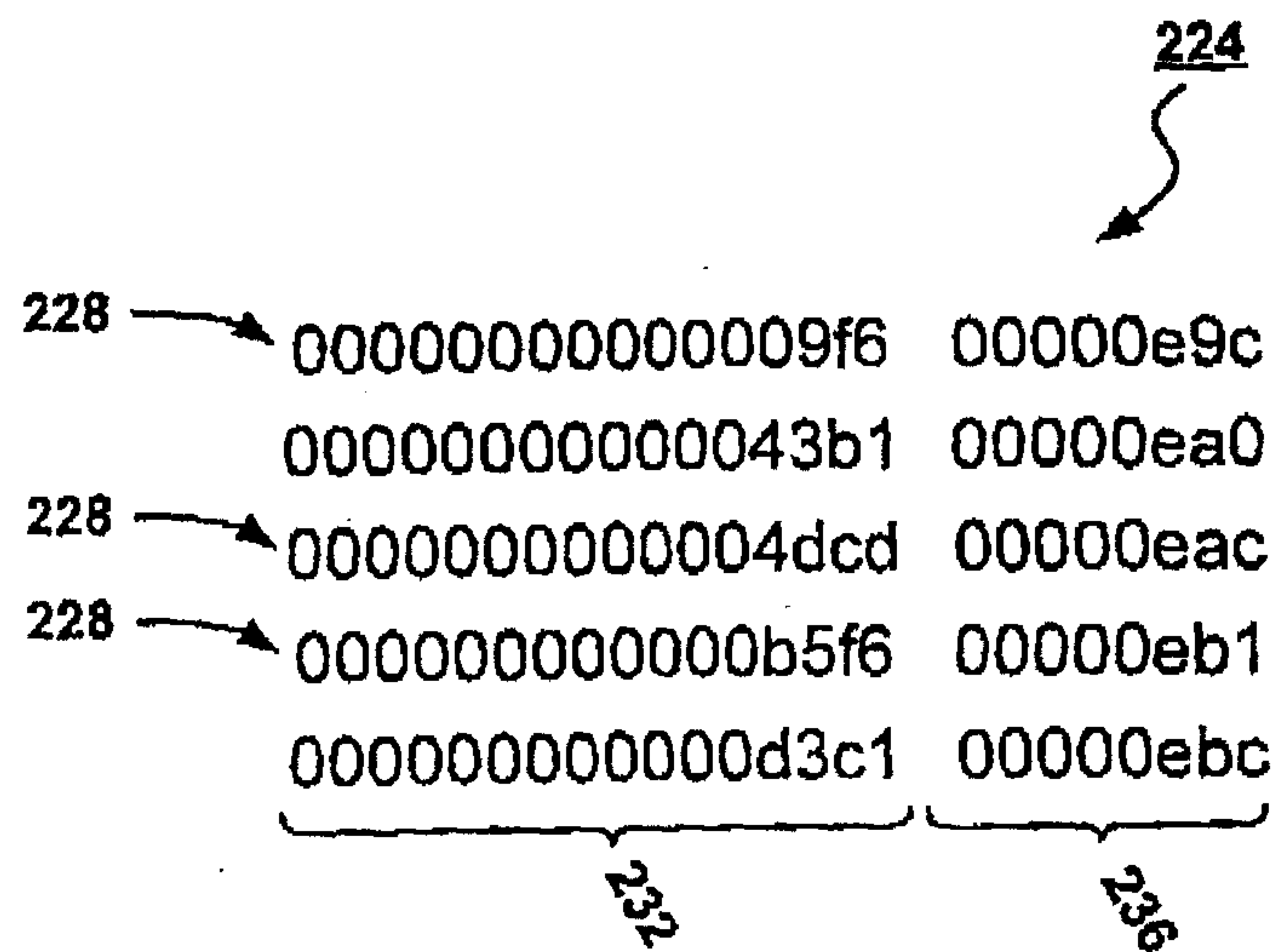
10

1/19

**FIG. 1**

**FIG. 2**

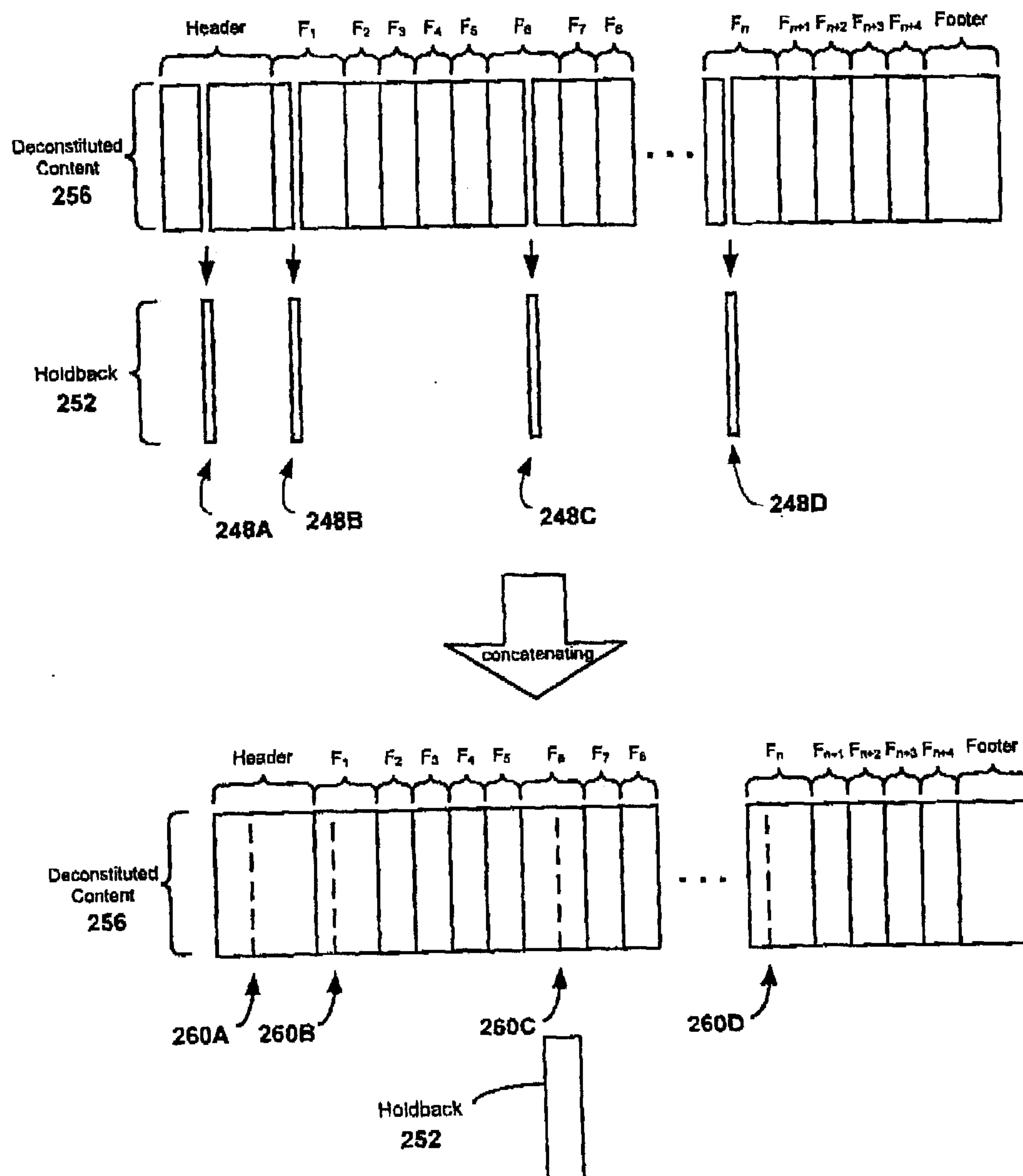
**FIG. 3**

**FIG. 4****FIG. 5**

WO 2007/118311

PCT/CA2007/000612

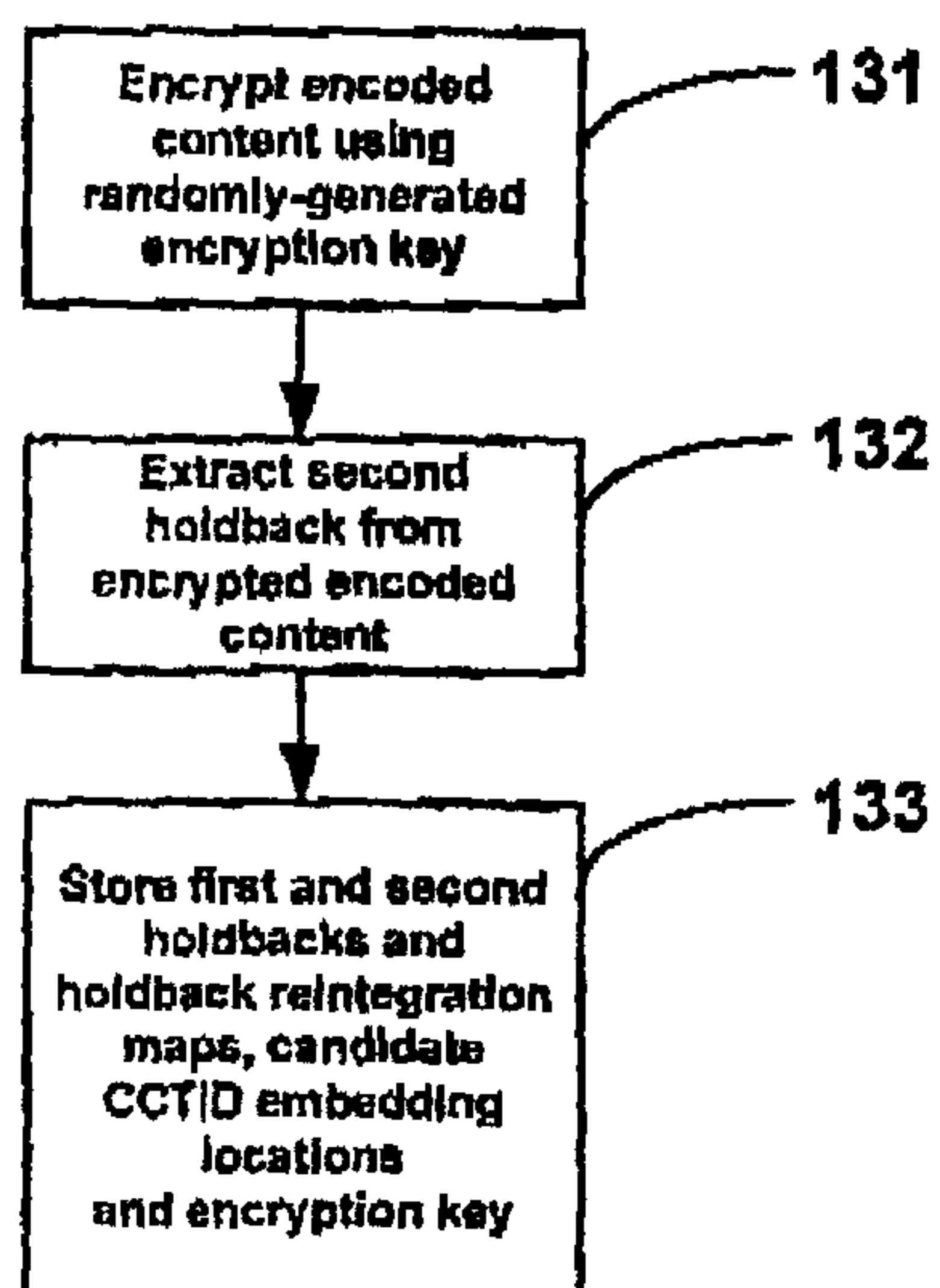
5/19

**FIG. 6**

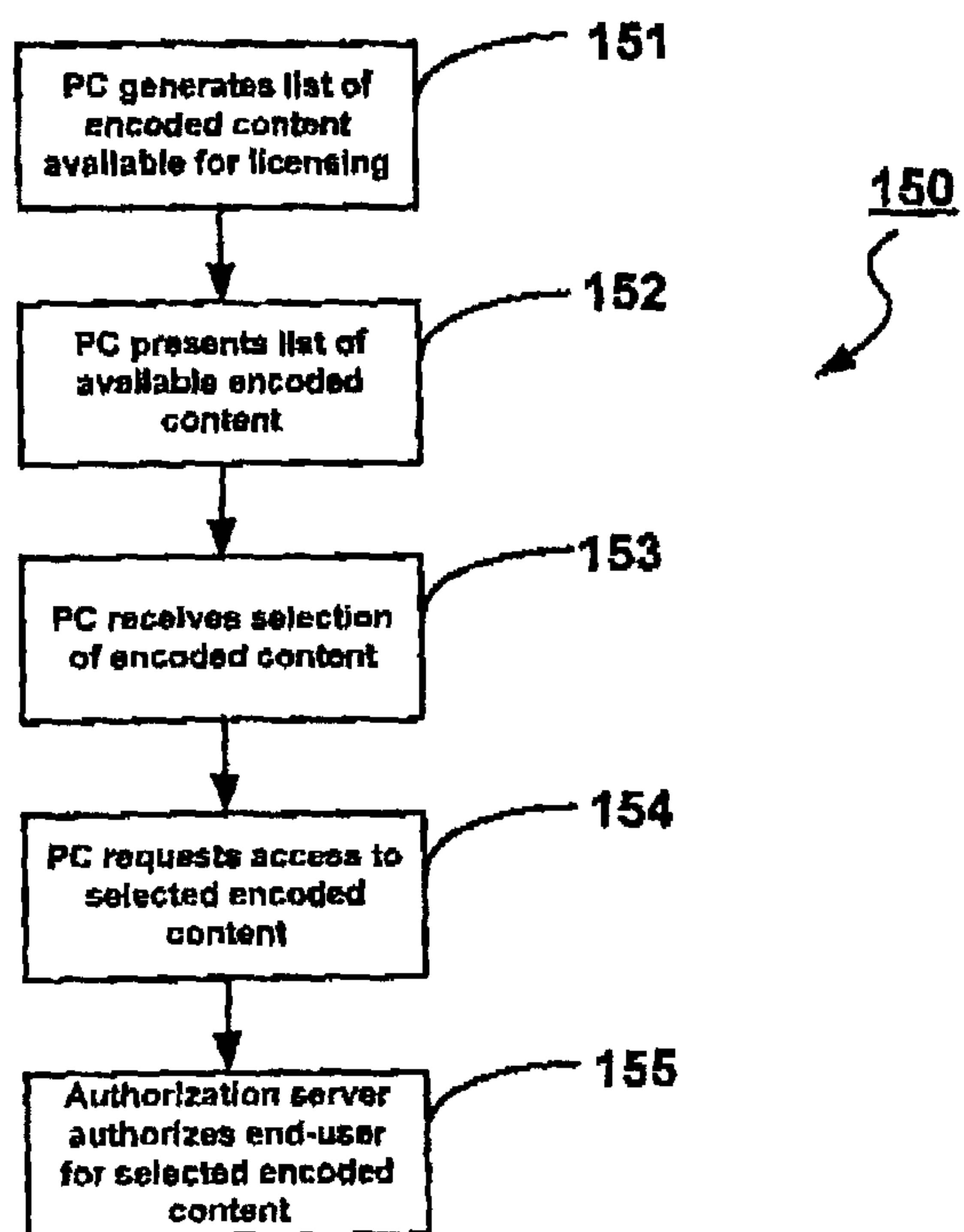
WO 2007/118311

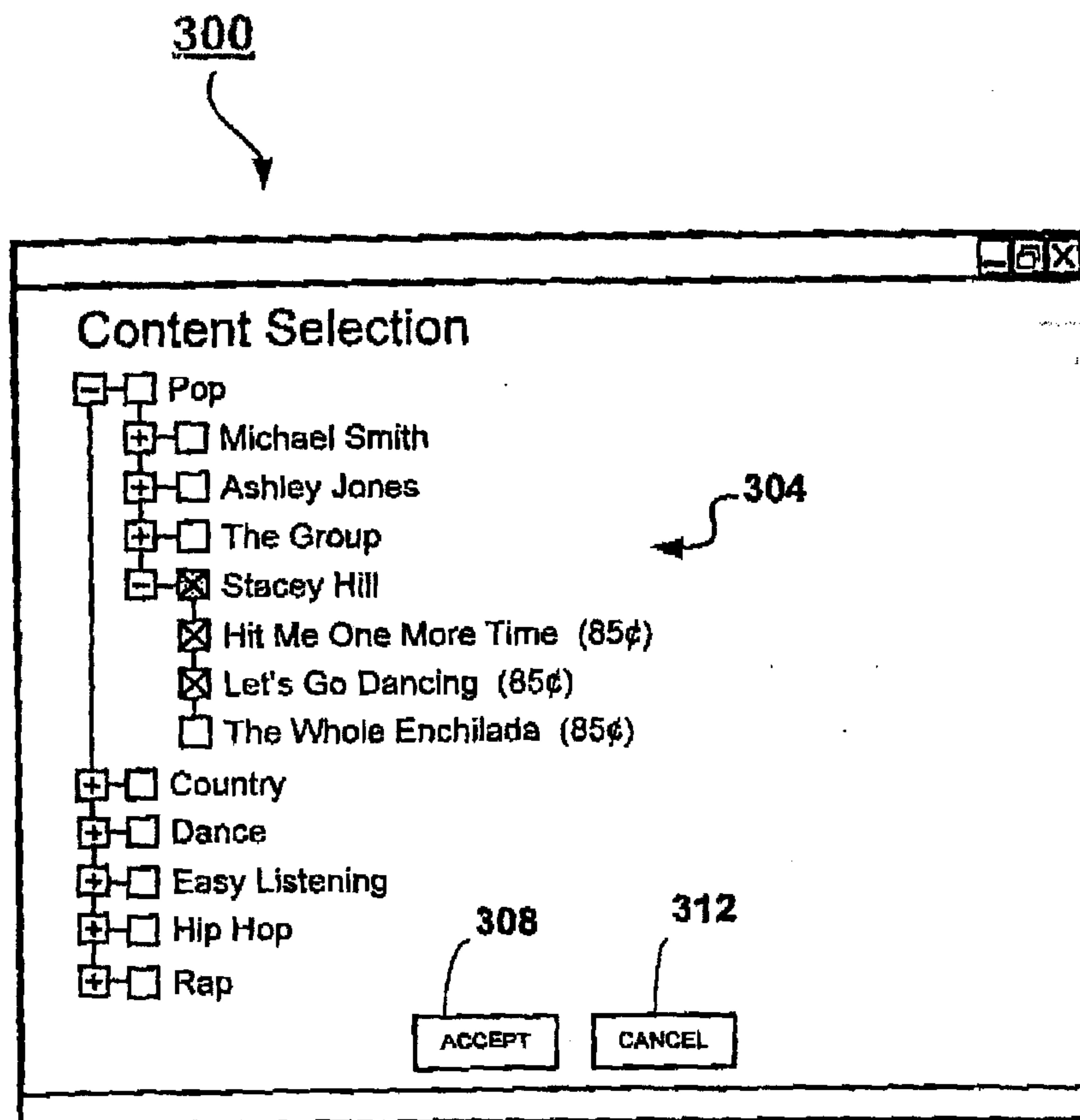
PCT/CA2007/000612

6/19

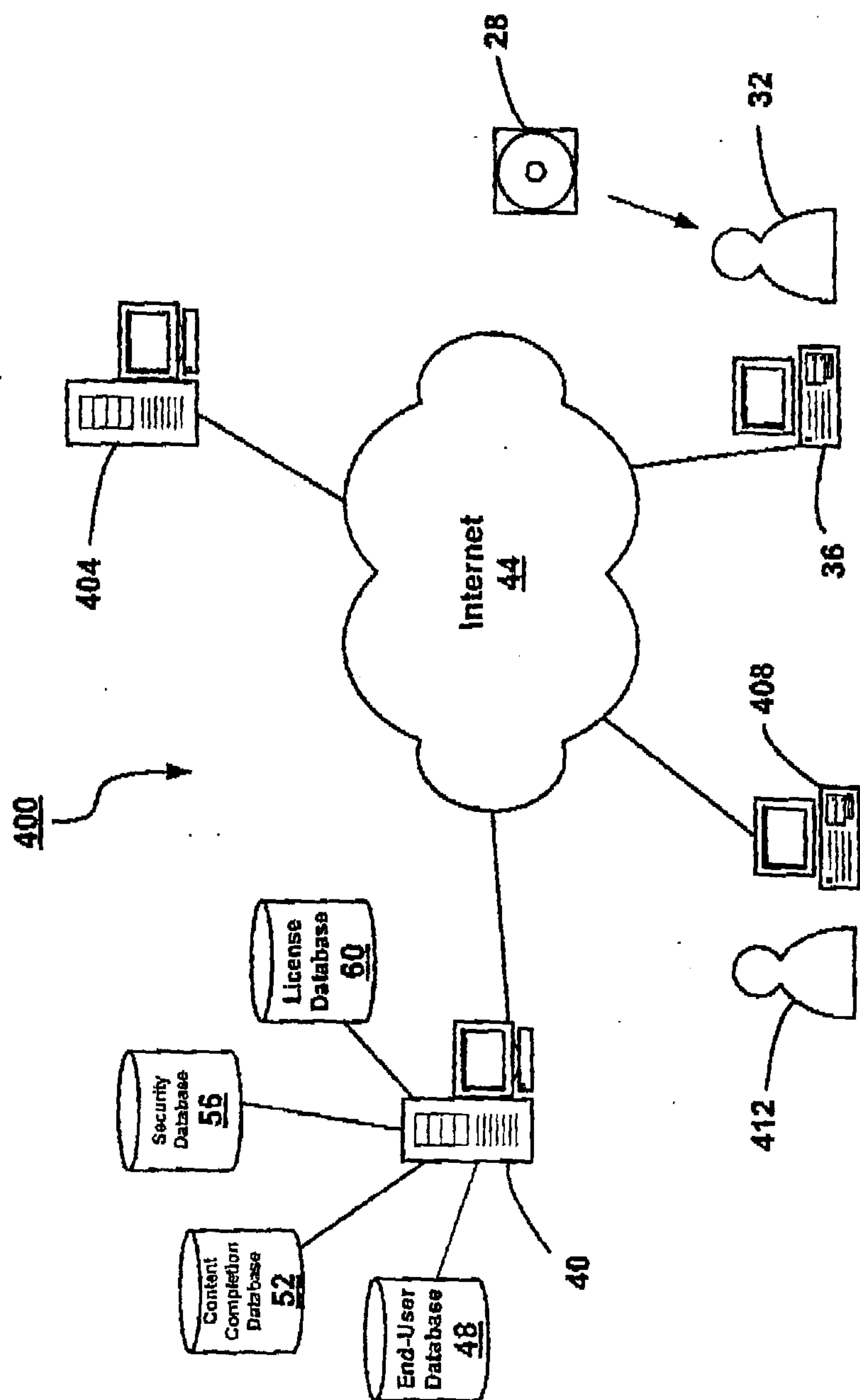
**FIG. 7**

7/19

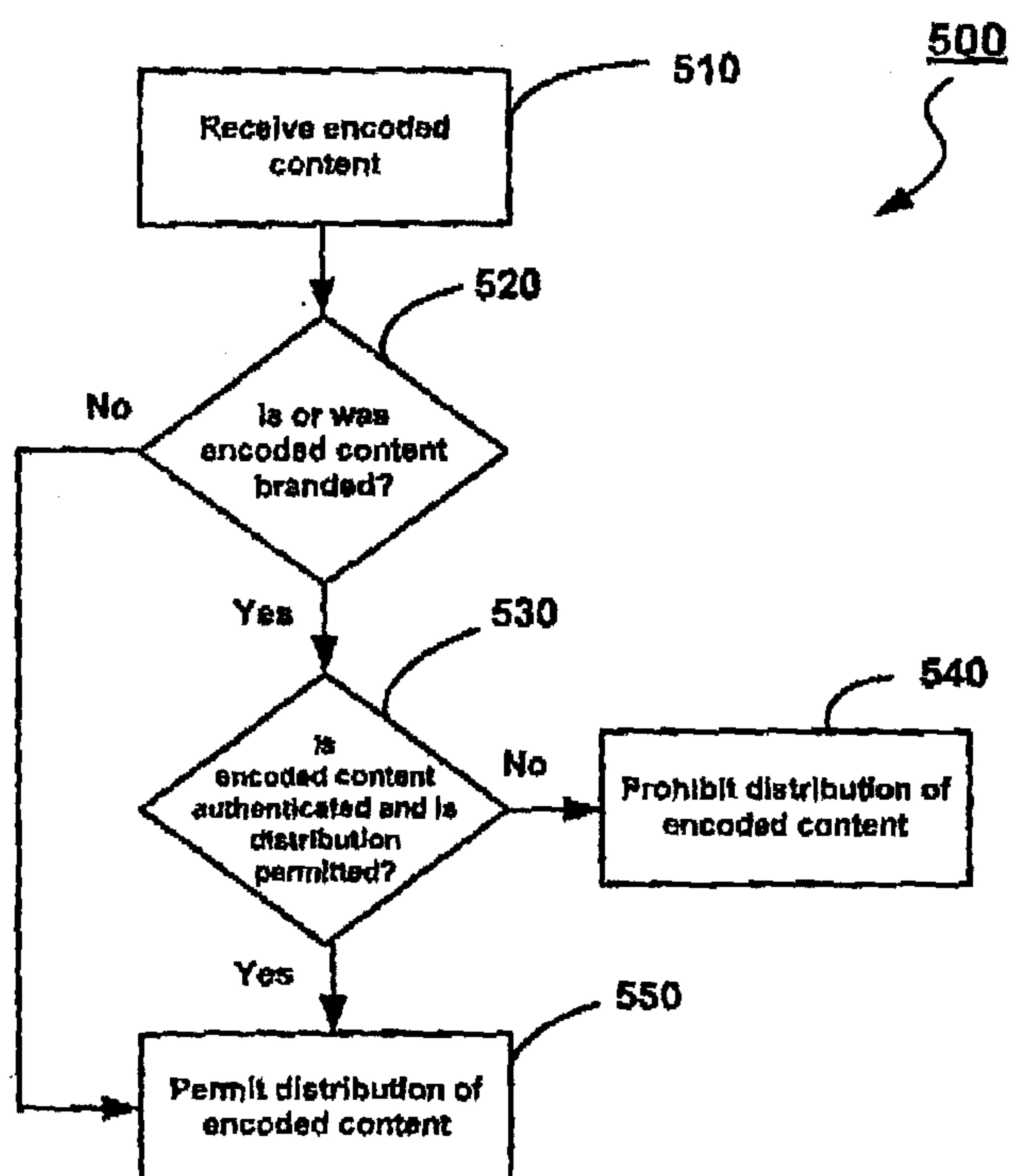
**FIG. 8**

**FIG. 9**

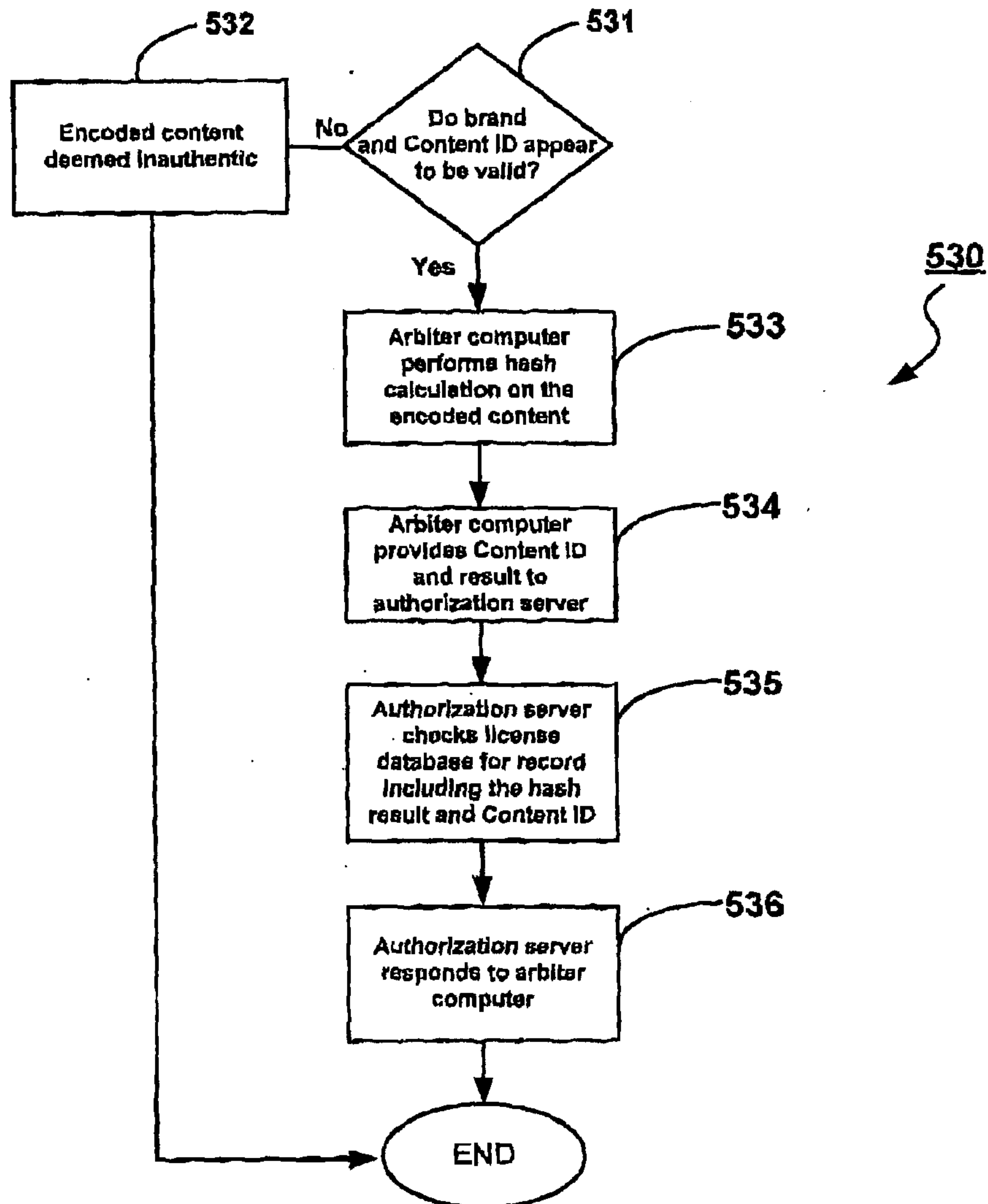
9/19

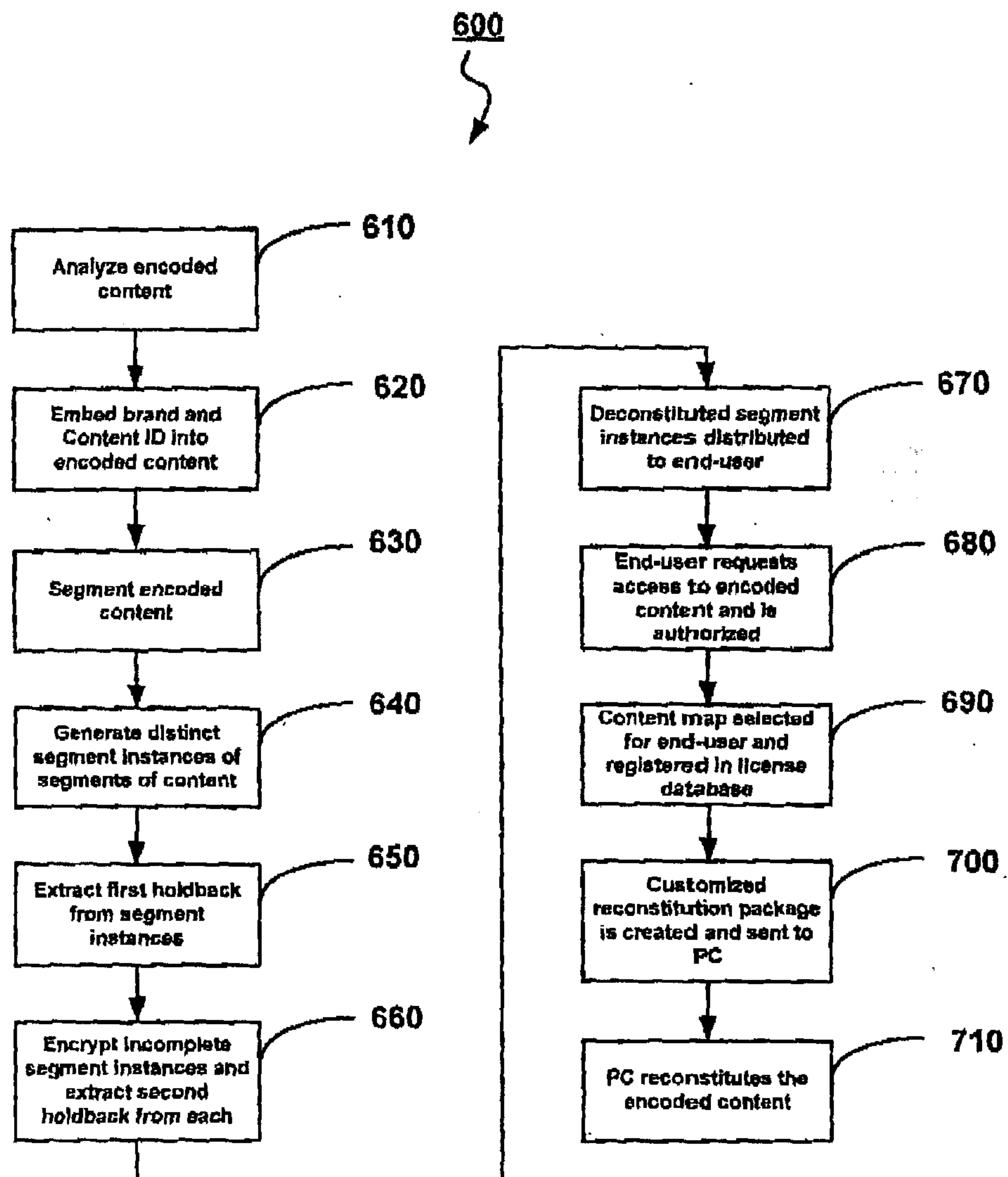
**FIG. 10**

10/19

**FIG. 11**

11/19

**FIG. 12**

FIG. 13

WO 2007/118311

PCT/CA2007/000612

13/19

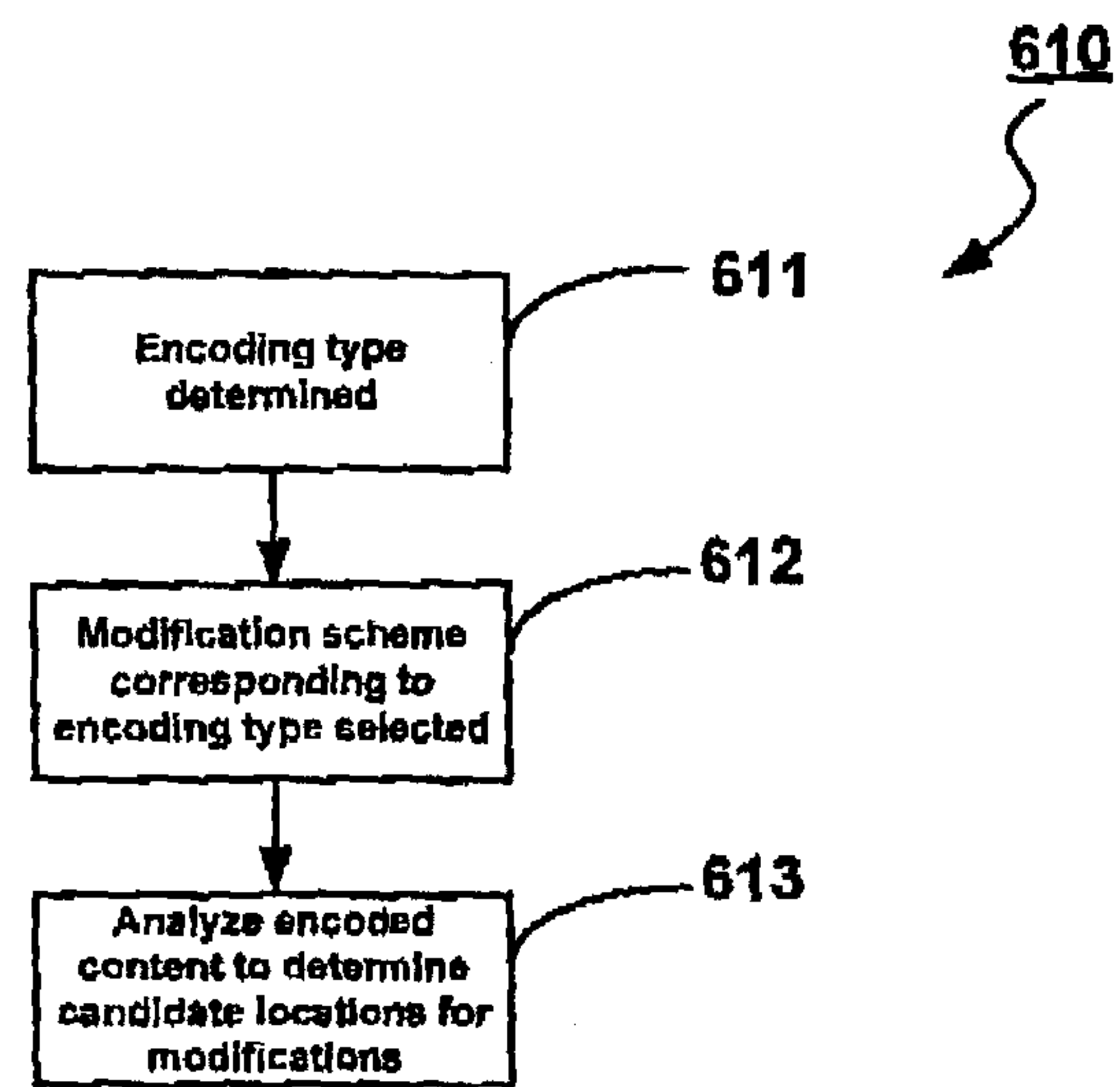


FIG. 14

WO 2007/118311

PCT/CA2007/000612

14/19

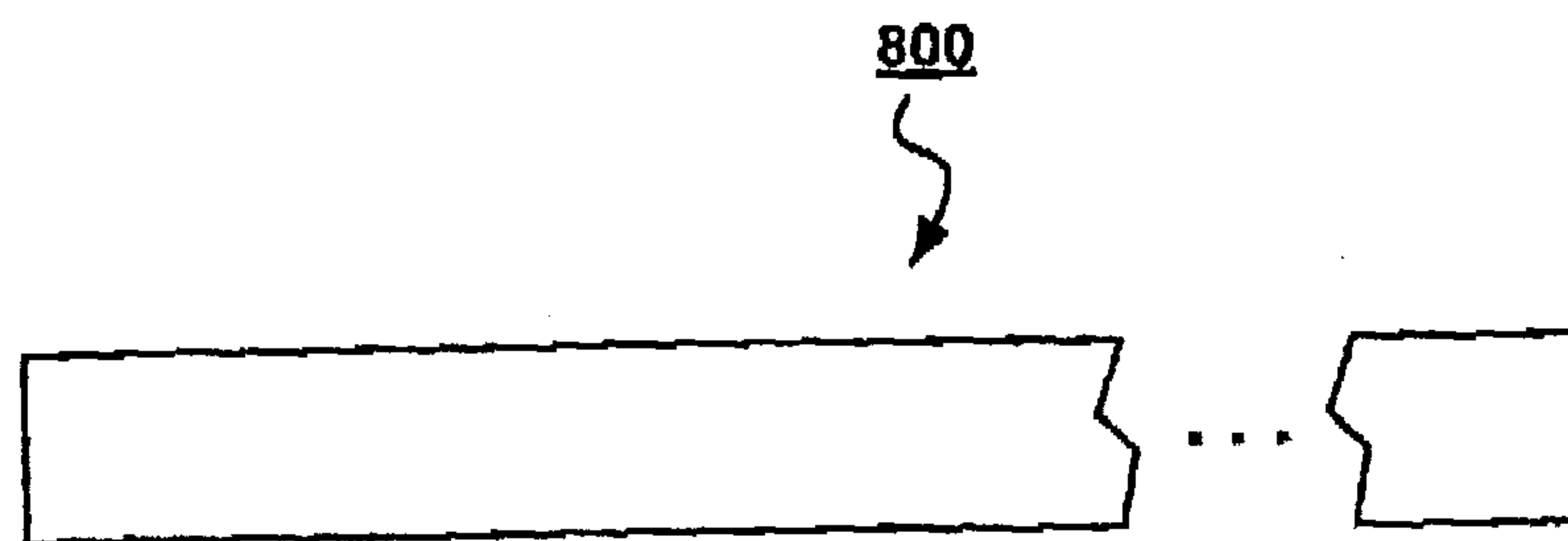


FIG. 15A

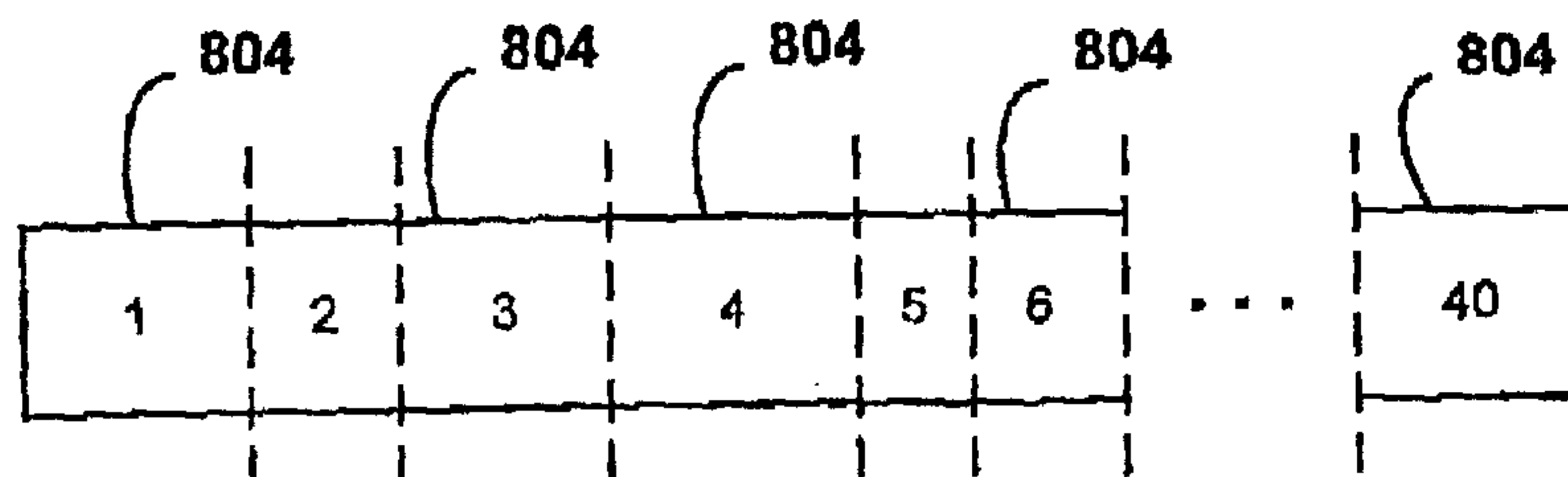
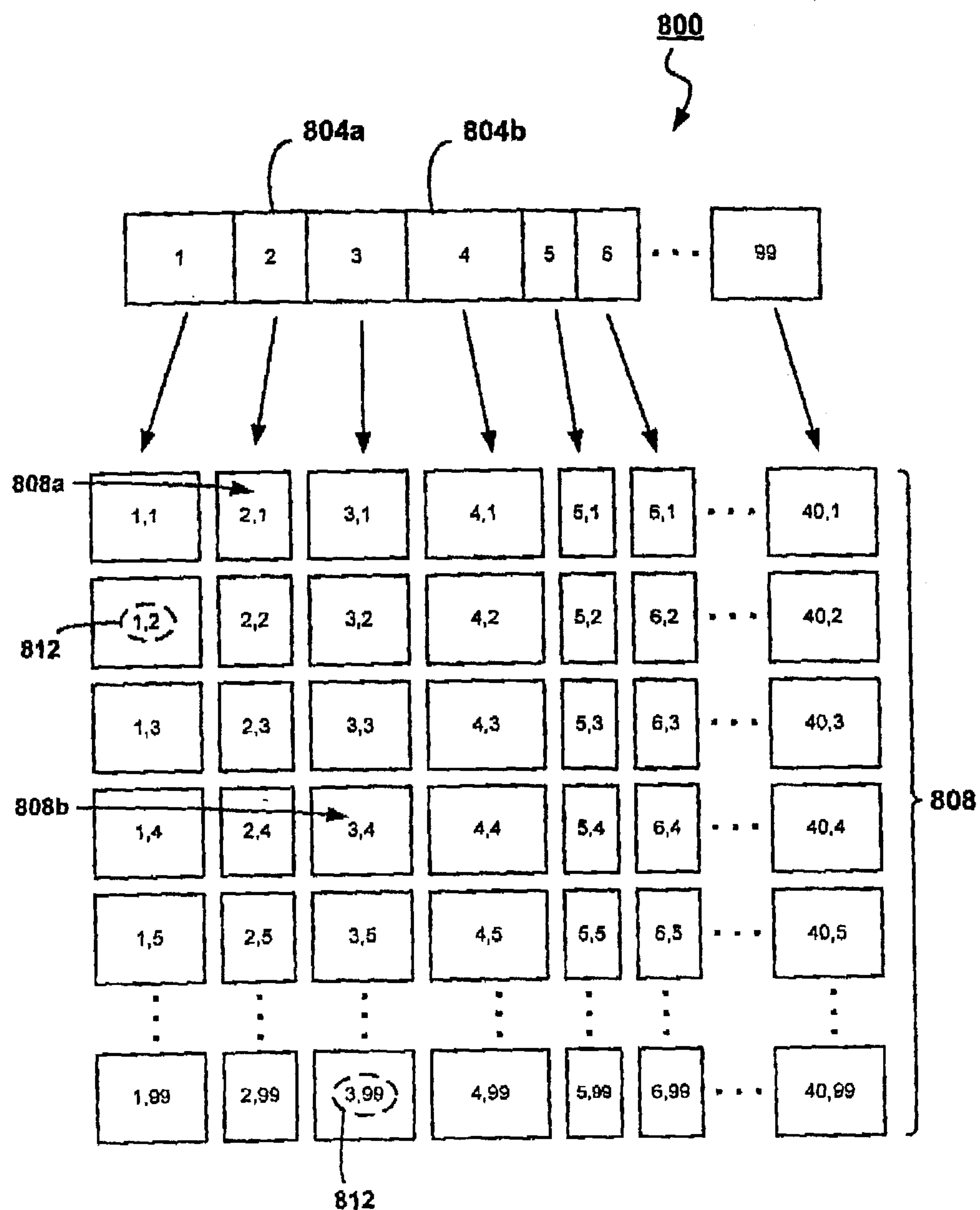
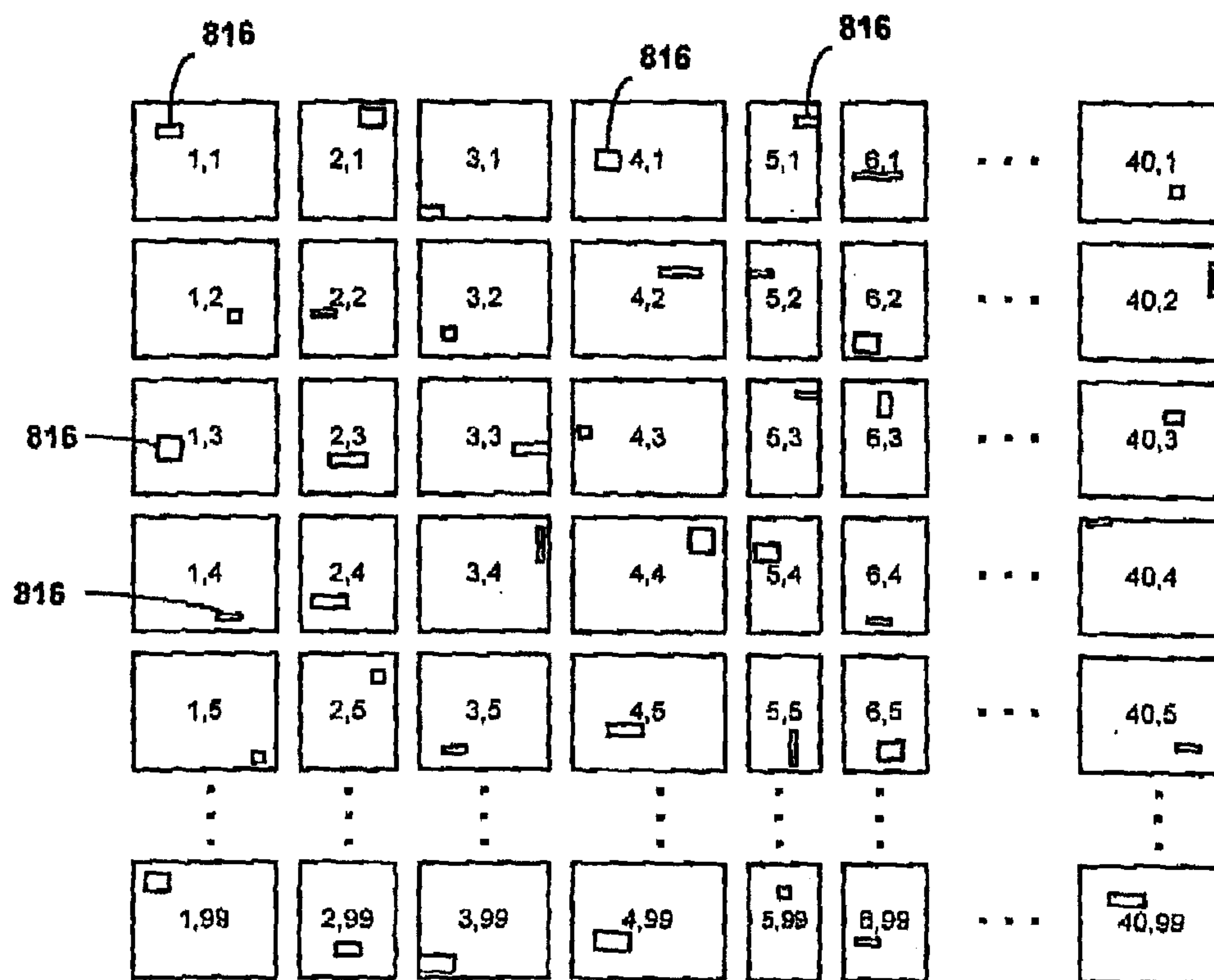


FIG. 15B

**FIG. 16**

16/19

**FIG. 17**

WO 2007/118311

PCT/CA2007/000612

17/19

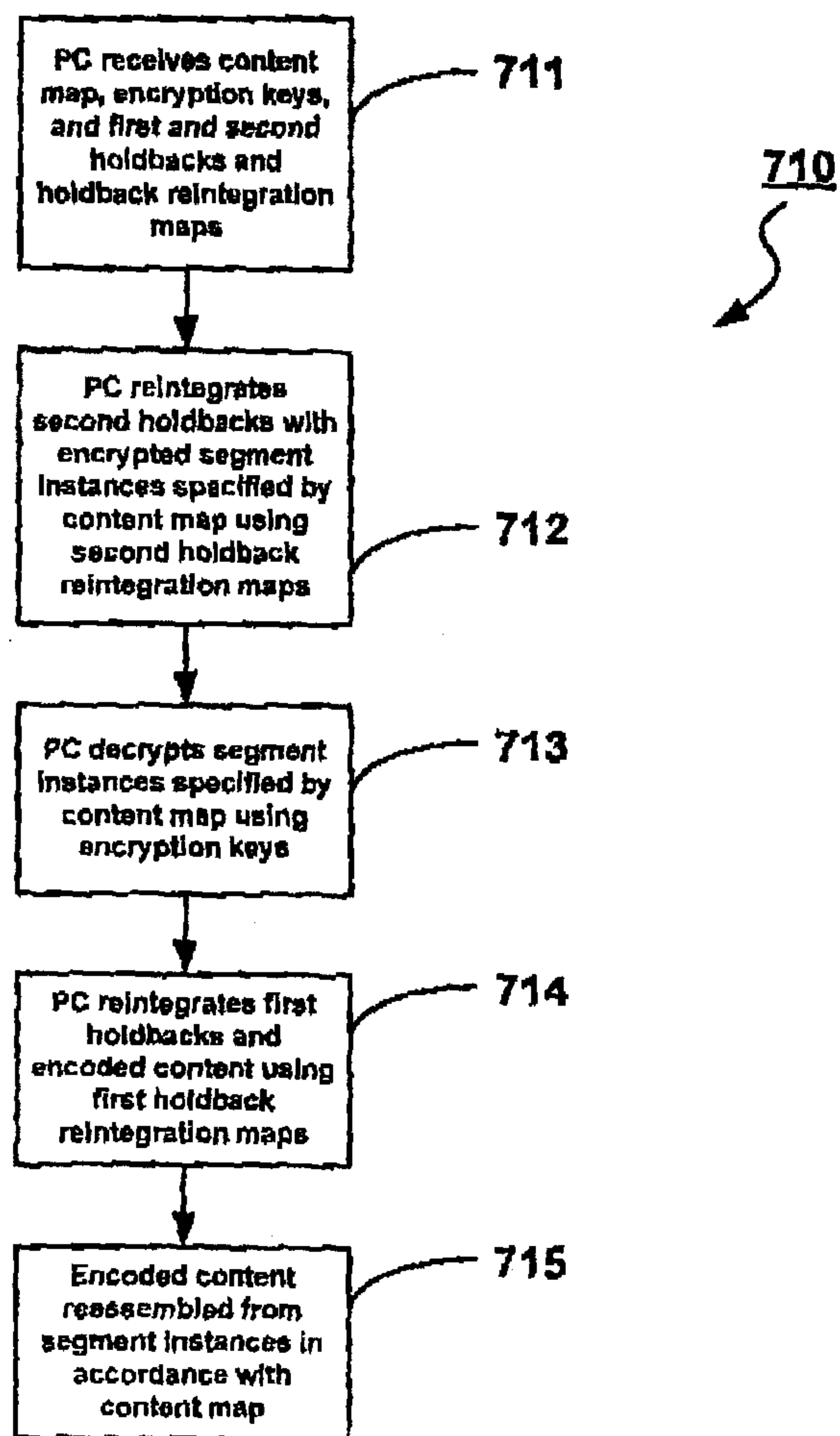
900



3, 2, 5, 1, 2, 2, ..., 4

FIG. 18

18/19

**FIG. 19**

WO 2007/118311

PCT/CA2007/000612

19/19

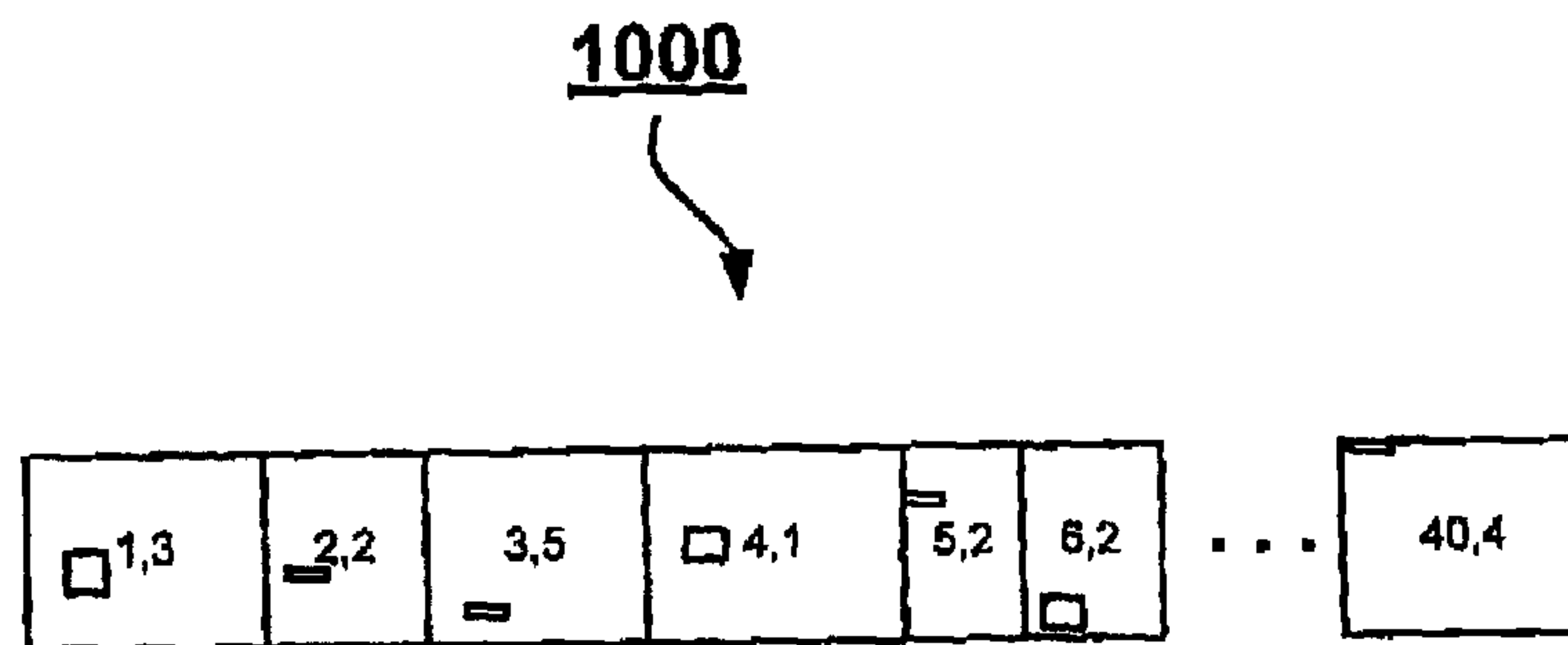


FIG. 20

