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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 February 2007 (22.02.2007)

PCT

(10) International Publication Number
WO 2007/021830 A1

(51) International Patent Classification:
G06F 17/00 (2006.01) *H04L 9/32* (2006.01)

(21) International Application Number:
PCT/US2006/031185

(22) International Filing Date: 10 August 2006 (10.08.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/201,751 11 August 2005 (11.08.2005) US

(71) Applicant *(for all designated States except US)*: **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) Inventors: **KLEMETS, Anders E.**; One Microsoft Way, Redmond, Washington 98052-6399 (US). **ALKOVE, James M.**; One Microsoft Way, Redmond, Washington 98052-6399 (US). **BHATT, Sanjay**; One Microsoft Way, Redmond, Washington 98052-6399 (US). **OLIVEIRA, Eduardo P.**; One Microsoft Way, Redmond, Washington 98052-6399 (US). **PAKA, Anand**; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, 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, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

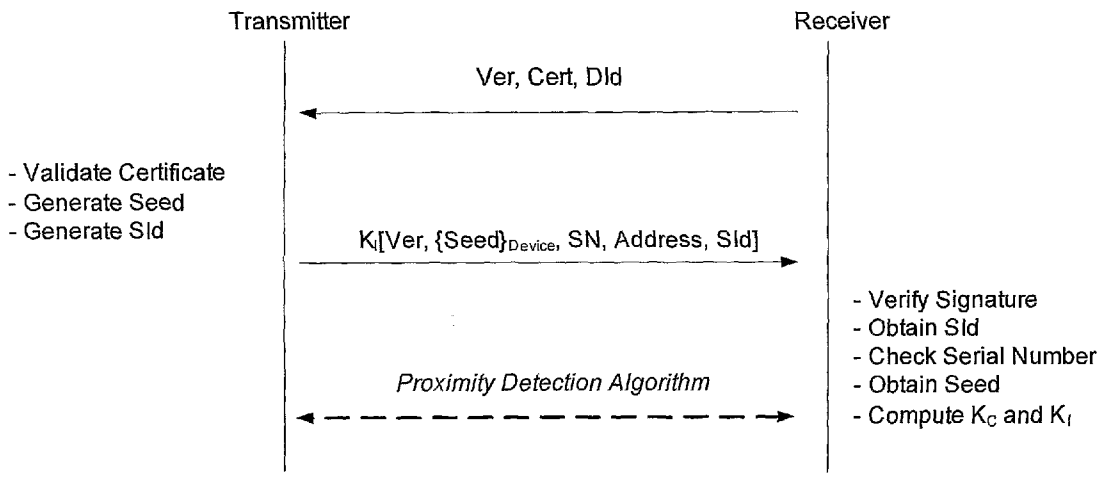
(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:
— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:
— *with international search report*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

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.

(54) Title: PROTECTING DIGITAL MEDIA OF VARIOUS CONTENT TYPES



(57) Abstract: Systems and/or methods ("tools") are described that enable a digital rights management policy to be associated with digital media having an arbitrary content type or transfer control protocol. In some embodiments, the tools encrypt data segments of a media file and add a descriptor to each of those segments. These descriptors can enable a receiver of the encrypted media file to decrypt the file and consume it according to the correct digital rights management policy.

WO 2007/021830 A1

PROTECTING DIGITAL MEDIA OF VARIOUS CONTENT TYPES

BACKGROUND

Digital Rights Management (DRM) refers to techniques that are used to protect content, such as by controlling or restricting the use of digital media content on electronic devices. One characteristic of DRM is that it can bind the media content to a given machine or device. Thus, a license that pertains to a particular piece of content and that defines rights and restrictions associated with the piece of content will typically be bound to the given machine or device. As a result, a user may not take the piece of content and move it to another machine in order to playback the content.

Current DRM techniques have limitations. They are often compatible with only two types of protocols for transferring digital media—HTTP and RTSP. But other protocols may now or in the future be better suited for transferring digital media. Also, content protected by DRM may be limited to a particular content type. One particular content type—ASF files—permits only one set of rights and restrictions, i.e. "policies", to apply to an entire ASF file. For example, when a video file is rendered, either Macrovision may be required to be enabled on an analog video output for the whole file, or it may not be required at all.

SUMMARY

Systems and/or methods ("tools") are described that enable a digital rights management policy to be associated with digital media having an arbitrary content type or transfer control protocol. In some embodiments, the tools encrypt data segments of a media file and add a descriptor to each of those segments. These descriptors can enable a receiver of the encrypted media file to decrypt the file and consume it according to the correct digital rights management policy.

This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is

not intended to identify key or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 illustrates an exemplary registration procedure of a protocol with which the inventive embodiments can be employed in one embodiment.

Fig. 2 illustrates an exemplary proximity detection procedure of a protocol with which the inventive embodiments can be employed in one embodiment.

Fig. 3 illustrates an exemplary session establishment procedure of a protocol with which the inventive embodiments can be employed in one embodiment.

Fig. 4 illustrates an exemplary data transfer procedure of a protocol with which the inventive embodiments can be employed in one embodiment.

Fig. 5 illustrates aspects of a streaming protocol with which the inventive embodiments can be utilized in accordance with one embodiment.

Fig. 6 illustrates aspects associated with root licenses and leaf licenses, in accordance with one embodiment.

Fig. 7 illustrates aspects associated with root licenses and leaf licenses, in accordance with one embodiment.

Fig. 8 illustrates an exemplary single media file having seven portions associated with five different exemplary digital rights management policies.

Fig. 9 illustrates the single media file of Fig. 8 with exemplary data segments associated with Key IDs (KIDs).

Fig. 10 illustrates, for a data segment shown in Fig. 9, a packet in accordance with one embodiment.

Fig. 11 is a flow diagram showing one manner in which the tools enable content-independent encryption and decryption.

Fig. 12 illustrates sample encryption in accordance with one embodiment.

Fig. 13 is a flow diagram showing communication of root and leaf licenses in accordance with one embodiment.

DETAILED DESCRIPTION

Overview

Tools are described that enable a digital rights management policy to be associated with digital media having an arbitrary content type or transfer control protocol. In some embodiments, the tools encrypt data segments of a media file and add a descriptor to each of those segments. These descriptors can enable a receiver of the encrypted media file to decrypt the file and consume it according to the correct digital rights management policy.

In the discussion that follows, a section entitled "Content Security and License Transfer Protocol" is provided and describes one particular system in which the inventive techniques can be employed. Following this, sections entitled "RTSP" and "HTTP" are provided to give the reader who is unfamiliar with these protocols understanding of the inventive techniques in these spaces.

Following this section, a section entitled "Root and Leaf Licenses" is provided and describes the notion of an initial, root license enabling multiple other licenses for a media file. Following this section, a section entitled "A Single, Encrypted Media File with Multiple Leaf Licenses" is provided and describes how a media file can be associated with more than one digital rights management policy using leaf licenses associated with portions of the media file.

Following these sections, two sections, the first entitled "Descriptors" and the second entitled "Content-Independent Data Encryption" describe descriptors for data segments of a media file and manners in which the tools may use these descriptors to enable encryption of a media file regardless of its type of digital content. The last section, "Using Root and Leaf Licenses" describes one way in which the tools may use root and leaf licenses.

Content Security and License Transfer Protocol

The following provides a discussion of an exemplary protocol that provides security and transfers licenses for content flowing over digital links. This protocol constitutes but one exemplary protocol with which the various inventive techniques

can be employed. It is to be appreciated and understood that other protocols can be utilized without departing from the spirit and scope of the claimed subject matter.

The following cryptographic notation is used in this description:

K{data}	data is encrypted with secret key K.
K[data]	data is signed with secret key K.
$\{data\}_{Device}$	data is encrypted with the device's public key.
$[data]_{Device}$	data is signed with the device's private key.

In this particular protocol, there are five primary procedures: *Registration, Revalidation, Proximity Detection, Session Establishment,* and *Data Transfer.*

In the *Registration* procedure, a transmitter (i.e. a device that has content that is to be transmitted to another device) can uniquely and securely identify an intended receiver (i.e. a device to which content is to be transmitted). In this particular protocol, the transmitter maintains a database with registered receivers and ensures that no more than a small predetermined number of receivers are used simultaneously. During the registration process, the transmitter also employs a *Proximity Detection* procedure to ensure that the receiver is located "near" the transmitter in the network, in order to prevent wide distribution of protected content.

The *Revalidation* procedure is utilized to ensure that the receiver continues to be "near" the transmitter. Content is not delivered to receivers unless they have been registered or revalidated within a predetermined period of time in the past.

The *Session Establishment* procedure is used whenever the receiver requests content from the transmitter. The transmitter enforces that devices must be registered and recently validated before the *Session Establishment* can be completed.

Once the session is established, the *Data Transfer* of the requested content can take place in a secure way. The receiver may reuse the session to retrieve specific portions of the content (seeking), but must establish a new session in order to retrieve a different content.

Consider now the Registration procedure in connection with Fig. 1 and the table just below that describes the various messages that are passed between the transmitter and the receiver during registration.

Message	Value	Description
Registration Request Message	Ver	8-bit Protocol Version
	Cert	XML digital certificate of the Receiver.
	DId	128-bit Serial Number.
Registration Response Message	Ver	8-bit Protocol Version
	{ Seed }Device	128-bit Seed used to derive the Content Encryption key and Content Integrity key.
	SN	128-bit Serial Number.
	Address	Address of transmitter's incoming and outgoing proximity packets socket.
	SId	128-bit Random Session Id.
Proximity Detection Algorithm		The Proximity Detection Algorithm is executed out-of-band.

Here, the receiver sends a registration request message that contains, among other information, the receiver's digital certificate. Responsive to receiving the registration request message, the transmitter validates the receiver's certificate, generates a seed and a random session ID, returning the same in the form indicated above to the receiver in a registration response message. The receiver then validates the transmitter's signature, obtains the session ID and performs the other actions indicated in the figure. The receiver and the transmitter can then undergo a proximity detection process which is described below.

With regard to *Revalidation*, the same procedures as outlined above are performed, with the difference being that during *Revalidation*, the receiver is already registered in the database.

With regard to *Proximity* Detection, consider the following in connection with Fig. 2.

During the *Proximity Detection* procedure, the receiver sends to the transmitter a message containing the Session Id indicated in a Proximity Detection Initialization

Message. The transmitter then sends to the receiver a message containing a Nonce (128-bit random value), and measures the time it takes for the receiver to reply with the nonce encrypted using a Content Encryption key. Finally, the transmitter sends a message to the receiver indicating if the proximity detection was successful or not.

The receiver may repeat the process until it has a confirmation that the proximity detection succeeded. When this particular protocol is used over IP-based networks, the proximity detection messages are exchanged over UDP. The receiver learns the transmitter's address via the Registration Response message. The receiver's address does not need to be separately communicated since it can be determined by inspecting the incoming IP header of the UDP packet that carries the Proximity Detection Initialization Message.

The following table describes the messages that are exchanged during Proximity Detection:

Message	Value	Description
Proximity Start Message	SId	Same 128-bit Session Id value sent by the transmitter.
Proximity Challenge Message	Seq	8-bit incremental sequence number.
	SId	Same 128-bit Session Id.
	Nonce	128-bit Random Value.
Proximity Response Message	Seq	Same sequence number determined by the transmitter.
	SId	Same 128-bit Session Id.
	KC{Nonce}	128-bit Nonce encrypted using the Content Encryption key.
Proximity Result Message	SId	Same 128-bit Session Id.
	Result	Status code indicating the success or failure of the registration procedure.

With regard to *Session Establishment*, consider the following in connection with Fig. 3 and the table just below which describes messages that are exchanged during *Session Establishment.*

Message	Value	Description	
License Request Message	Ver	8-bit Protocol Version	
	Cert	XML digital certificate of the Receiver.	
	SN	128-bit Serial Number.	
	Action	Requested usage for the content. Ex.: "Play", "Copy" or "Burn".	
	RId	128-bit random Rights Id.	
	VCRL	Version of the receiver's CRL.	
License Response Message	Ver	8-bit Protocol Version	
	CRL	Transmitter's CRL. Only sent in case it has a higher version number than the receiver's CRL and the receiver component also has transmitting capabilities.	
	License	KC (encrypted with receiver's public key)	128-bit Random Content Encryption key.
		KI (encrypted with receiver's public key)	128-bit Random Content Integrity key.
		VCRL	Version of the transmitter's CRL.
		RId	Same 128-bit random Rights Id sent by the receiver.
		SN	128-bit Serial Number.

In this example, a License Request Message is sent from the receiver to the transmitter and contains the information described above. In response, the transmitter can send a License Response Message that contains the information described above.

In this particular example, the License is represented in XMR format and includes a Content Encryption key, a Content Integrity key, a Version of the Transmitter's CRL, a 128-bit Rights Id and a 128-bit Serial Number. The License also contains an OMAC calculated using the Content Integrity key using OMAC.

With regard to the *Data Transfer* procedure, consider the following in connection with Fig. 4. Once the *Session Establishment* is complete, the data transfer is executed in a control protocol specific manner. Both the *Data Transfer* request and response must be specifically defined for the control protocol and content type. This is conceptually represented in Fig. 4.

Having now provided a brief overview of an exemplary protocol with which the inventive embodiments can be employed, consider now some background information on RTSP.

RTSP

The Real Time Streaming Protocol or RTSP is an application-level protocol for control over the delivery of continuous media (e.g., data with real-time properties like streaming), as will be appreciated by the skilled artisan. RTSP provides an extensible framework to enable controlled, on-demand delivery of real-time data, such as audio and video. Sources of data can include both live data feeds and stored clips. This protocol is intended to control multiple data delivery sessions, provide a means for choosing delivery channels such as UDP, multicast UDP and TCP, and provide a means for choosing delivery mechanisms based upon RTP.

RTSP establishes and controls either a single or several time-synchronized streams of continuous media such as audio and video. It does not typically deliver the continuous streams itself, although interleaving of the continuous media stream with the control stream is possible. In other words, RTSP acts as a "network remote control" for multimedia servers.

The set of streams to be controlled is defined by a presentation description. In RTSP, there is no notion of an RTSP connection; instead, a server maintains a session labeled by an identifier. An RTSP session is in no way tied to a transport-level connection such as a TCP connection. During an RTSP session, an RTSP client may open and close many reliable transport connections to the server to issue RTSP requests. Alternatively, it may use a connectionless transport protocol such as UDP, as will be appreciated by the skilled artisan.

The streams controlled by RTSP may use RTP, but the operation of RTSP does not depend on the transport mechanism used to carry continuous media.

Consider now a typical RTSP request/response exchange in connection with Fig. 5, between a client/receiver 500 and a server/transmitter 502.

Preliminarily, the RTSP requests/responses have headers which, for the sake of brevity, are not described. In RTSP, a client/receiver 500 typically issues what is known as a DESCRIBE request which is directed to retrieving a description of a

presentation or media object identified by a request URL from server 502. The server 502 responds with a description of the requested resource which is represented in the SESSION DESCRIPTION PROTOCOL (SDP). The DESCRIBE response (SDP) contains all media initialization information for the resource(s) that it describes.

Next, client 500 sends a SETUP request for a URI that specifies the transport mechanism to be used for the streamed media. In the Fig. 5 example, a SETUP request is sent for both audio and video. Client 500 also indicates, in the SETUP request, the transport parameters that it will be utilizing. A transport header in the SETUP request specifies the transport parameters acceptable to the client for data transmission. The RESPONSE from server 502 contains the transport parameters selected by the server. The server also generates session identifiers in response to the SETUP requests.

At this point, the client can issue a PLAY request which tells the server to start sending data via the mechanism specified in the SETUP. Responsive to receiving a PLAY request, the server can start streaming the content which, in this example, is the audio/video content. In this example, the streaming content is encapsulated using RTP packets and is sent over UDP, as will be appreciated by the skilled artisan.

The RTSP protocol has other methods of interest which include PAUSE, TEARDOWN, GET_PARAMETER, SET_PARAMETER, REDIRECT, and RECORD. For additional background on RTSP, the reader should consult the RTSP RFC, Schulzrinne, H., Rao, A., and R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC 2326, available at http://www.ietf.org/rfc/rfc2326.txt, April 1998.

Root and Leaf Licenses

In the illustrated and described embodiment, the notion of a *root license* and *leaf licenses* are employed. Here, the root license is utilized to set up and securely deliver a content key (a *root content key*) to the client/receiver so that the client/receiver can decrypt subsequently-delivered leaf license(s). Once the root content key is securely delivered to the client/receiver, content keys for various leaf licenses (*leaf content keys*) can be encrypted by the server/transmitter using the root content key sent to the client/receiver. Using the root content key, the client can

decrypt the leaf content keys and associated policies in the leaf licenses. Each of the leaf licenses also have a unique identifier capable of associating the leaf license with a portion of a media file. Here the unique identifier is referred to as the Key ID, or KID and for each leaf license numbered 1 to n ($leaf_1$, $leaf_2$, ... $leaf_n$), $KID_{leaf\text{-}n}$.

To provide but one example of how this particular scheme can be implemented, consider the following in connection with Fig. 6. In this particular example, the system of Fig. 6 is configured to use 1024-bit RSA keys for public key cryptographic operation and 128-bit AES keys for symmetric cryptographic operations. Of course, this is provided as but one example and is not intended to limit application of the claimed subject matter.

In this example, client/receiver 600 has a public/private key pair 650 and the server/transmitter 602 has the client/receiver's public key. In this example, each of the client/receiver's public and private keys is a 1024-bit RSA key. Using the client/receiver's public key, the server/transmitter builds a root license that contains a root content key that is encrypted with the client/receiver's public key. The root content key is a 128-bit AES content key. This root license is then sent to the client/receiver. In Fig. 6, this is shown as the first communication that takes place between the client/receiver and server-transmitter, where the encrypted root content key is represented as $\{\text{content key}_{root}\}_{CLIENT}$. It is to be appreciated, however, that other communication prior to the illustrated communication can take place.

Having received the encrypted root content key from the server/transmitter, the client/receiver can now decrypt the root content key using its private key and can securely store the decrypted root content key for future use.

At this point, consider what has occurred. The server/transmitter has securely communicated a key to the client/receiver that can now serve as the basis for subsequent cryptographic operations. More specifically, consider now that multiple, particular policies may pertain to multiple, particular pieces of DRM-protected content in a single media file. In this case, the server/transmitter can prepare multiple leaf licenses each containing a digital rights management policy and an encrypted version of a particular leaf content key. In this example, each leaf content key is a 128-bit AES content key that has been encrypted using the root content key. Thus, the

computational complexity and expense experienced and incurred by the client/receiver associated with decrypting new and additional leaf content keys is reduced over that associated with 1024-bit RSA key operations because now, the client/receiver only needs to decrypt using a 128-bit AES content key (i.e. the root content key).

HTTP

Having now discussed the notion of a root and leaf license and how each can be employed in the contexts described above, consider now how the root and leaf license can be delivered using HTTP.

When HTTP is utilized for carrying DRM-protected content, the client issues two requests to the server/transmitter. First, the client issues a POST request to retrieve a root license. Second, the client issues a GET request for retrieving the DRM-protected content. The client issues the requests in this example because in HTTP, the server typically cannot initiate communication with a client.

Specifically, consider Fig. 7 in connection with the following discussion. When a client wishes to receive a root license, it issues a POST request to the server. The POST request contains a license request message, as discussed above. Responsive to receiving this communication, the server responds with a license response message that contains a root license which, in at least one embodiment, is expressed in XMR. Having received the root license and processed it accordingly, the client issues a GET request to the server asking for the DRM-protected content. Responsive to the GET request, the server replies with segments of the requested content interleaved with one or more license response messages. The license response messages each contain a leaf license that pertains to a particular portion of the DRM-protected content. Any suitable mechanism or interleaving technique can be used for formulating the server's reply.

As but one implementation example in one particular context, consider the following.

In but one example, a four-byte framing header is used to encapsulate data and control blocks. The framing header contains a one byte ASCII dollar sign (0x24),

followed by a one byte block type identifier, followed by a two byte length of the encapsulated data, represented in network byte order.

Sections	Fields
Header	8-bit ASCII dollar sign (0x24)
	8-bit Block Type
Data Length	16-bit Length of the encapsulated data

A Control block uses an ASCII 'c' character (0x63) as its type identifier. This block contains a message, typically a License Response message.

A Data block uses an ASCII 'd' character (0x63) as its type identifier. This block contains a Data Segment descriptor immediately followed by media data.

The Data Segment descriptor can be associated with content that is encrypted or in the clear. An encrypted flag in the descriptor conveys this information. A Data Segment descriptor is associated with a portion of the transmitted file to which, if encrypted, a single policy and content encryption key apply. In other words, the content encryption key and policies cannot be changed within the segment.

In accordance with one embodiment, a typical HTTP response with link encryption is comprised of the following blocks:

1. Control block [$c] carrying a License Response message with a Chained License.
2. One or more Data blocks [$d].

In case there is a key or policy change during the transmission of the file, then the following steps are added:

3. A new Control block [$c] carrying a License Response message with a new Chained License.
4. One or more Data blocks [$d].

Note that steps 3 and 4 may occur multiple times in the case of multiple key or policy changes.

A Single, Encrypted Media File with Multiple Leaf Licenses

The tools enable a single encrypted media file to have portions associated with different policies. The single encrypted media file may be of an arbitrary content type (e.g., ASF, MPEG, WAV, or other files) and be transferred using various control protocols.

In the following illustrated and described embodiment of Figure 8, a single, encrypted media file 800 has seven portions 802, 804, 806, 808, 810, 812, and 814. Assume that this media file is a media program about the history of music videos. The first portion is an introduction to music videos, the second is a music video, the third an advertisement, the fourth is another music video, the fifth is another music video, the sixth is another advertisement, and the seventh is a conclusion to the program.

Here the creator of the media program desires to have different rights for various portions. The creator may be willing to permit users of the media program to play the introduction and conclusion portions and copy them a certain number of times. The creator may not be willing to grant the same rights to the music videos; assume here that the creator of the program does not own these music videos, and so they are subject to different policies of use. The creator may also be willing to have the advertisements used freely—and thus they may be copied, used, and played in any way a user likes.

To govern the usage of each of these portions, each is associated with a policy. Here the policy is in a leaf license having a KID and content key. Assume that one root license and five leaf licenses are received for this media program. The leaf licenses are shown in Figure 8 at 816, 818, 820, 822, and 824. Each of the leaf licenses has a unique KID (KID_1, KID_2, KID_3, KID_4, and KID_5) and a unique leaf content key (leaf content key_1, leaf content key_2, leaf content key_3, leaf content key_4,

and leaf content key_5). Each leaf license also contains a policy ($policy_1$, $policy_2$, $policy_3$, $policy_4$, and $policy_5$) permitting or excluding certain rights for using the media of each of the associated portions. These leaf licenses are expressed in XMR (eXtensible Media Rights), though other languages may also be used.

The first policy (that of leaf license #1) permits media associated with it to be played up to ten times and copied up to three times. This policy permits, therefore, the introduction and the conclusion of the program to be played and copied a certain number of times.

The second policy permits media associated with it to be played only once and not copied. Thus, the first music video of the program can only be played once. If a user attempts to play the entire program a second time, this video will not play.

The third policy permits media associated with it to be used in any way desired. The policy itself can set this out—that there are no restrictions on the play, copying, or other use of associated media. In some embodiments, however, the portions of the media may instead be in the clear (not encrypted). An example of this is described below. In either case, both the first and second advertisements may be used in any way desired.

The fourth policy permits media associated with it to by played as many times as a user likes, but cannot be copied. Thus, the second music video can be played but not copied.

The fifth policy permits media associated with it to be played as many times as a user likes and copied, but only as an analog file. Thus, the third music video may be played, and copied in a certain way only.

The association between each of the portions and the licenses are shown in Figure 8 with dashed lines. Ways in which the tools may establish this association are set forth in greater detail below.

Descriptors

The tools can associate policies with portions of a single media file. Continuing the illustrated and described embodiment of Figure 8, each of the portions is associated with a policy through a leaf license. To better explain how this

association may be established, one portion of single media file 800 is illustrated in greater detail.

Fig. 9 illustrates media file 800 with fourth portion 808 expanded to show one way in which this portion can be associated with a policy. Here the media file is received with the portions generally in order. In this example the root license is received, followed by a first leaf license, followed by the first portion of the media file, followed by the second leaf license, followed by the second portion, and so on. Here the leaf licenses are not all received prior to receiving the beginning of the media file as described above. Because of this, the first and third leaf licenses may be sent again prior to the portion associated with them (thus, the first leaf license may be sent before the first portion and again before the seventh portion).

When a new policy is to be followed for a portion of the media file, a new leaf license (here fourth leaf license 822) is sent prior to the portion of the media associated with the fourth leaf license.

Here the leaf license is sent as part of a control block 902, followed by data segments 904-914 of fourth portion 808. In RTSP, however, the licenses are delivered in SDP descriptors or ANNOUNCE messages. This particular embodiment focuses on use of HTTP, though use and communication of leaf licenses and data may also use RTSP, such as is set forth in the description relating to Fig. 7 above. The control block comprises leaf license 822 of Figs. 8 and 9. The leaf license has the leaf content key_4, the $policy_4$, and the KID_4. Once received, the fourth leaf license can be decrypted using the root content key. The KID can be sent in the clear or encrypted but capable of being decrypted.

Each of the data segments is associated with a policy, here data segments 904-914 are associated with the corresponding fourth policy. This association is established with the KID of the fourth leaf license. The KID, or an identifier associated with the KID, is stored in each data segment. The KID can be a relatively short piece of information, even an integer taking up less than a byte of memory. Thus, the receiver can associate the data segment with the appropriate policy based on the KID indicating the appropriate policy.

The descriptor can be used with various control and data protocols and packet structures now in existence or that may be created in the future. One such exemplary data protocol is RTP. Here the descriptor is oriented appended to the end of each packet. In another embodiment, an HTTP control protocol is used. Here the descriptor is oriented appended at the beginning of each frame.

Fig. 10 illustrates a descriptor associating a data segment with a leaf license in accordance with RTSP.

In this example, data segment 1000 can include an RTP payload format header 1008 and payload data 1010. Here the payload data and payload format header are encrypted, an example of which is described as part of Fig. 11 below.

Here the descriptor is appended to the end of the payload data according the RTP protocol, though it can be placed at any suitable location permitted by the data protocol. Placing the descriptor at the end of the payload data can mitigate backward compatibility issues, as will be appreciated by the skilled artisan.

In this embodiment, the RTP packet—with the exception of the RTP header—is associated with the descriptor 1012. Descriptor 1012, in turn, carries with it the encryption parameters that can be used in a decryption process that enables payload data 1010 and RTP payload format header 1008 to be decrypted (e.g., the Initialization Vector (IV) associated with the fourth leaf content key). In this particular example, a single policy and content encryption key applies to the payload data 1010.

In accordance with one embodiment, descriptor 1012 comprises a data structure as follows:

Sections	Fields
Flags	8-bit Flags
Extensions	8-bit Number of Extensions
	Multiple Variable Length Extensions
Length	Data Segment Descriptor Length

In this example, the Flags section is a bit-field indicating attributes of the Data Segment. The following bit is currently defined: Bit 0 (Encrypted Data.) When this

bit is set to 1, it indicates that the Data Segment is in encrypted form. Otherwise, the Data Segment is in the clear.

The extension section comprises the KID and IV; here the KID is the KID_4 and the IV is associated with the leaf content key_4.

With regard to the Extensions section, the Number of Extensions field indicates the number of variable length extensions included in this descriptor. With regard to the Variable Length Extension field, each extension has the following format:

Fields
8-bit Extension Type
16-bit Extension Length
Variable Length Extension

In accordance with one embodiment, the KID and IV are defined as follows:

KID

Extension Type: Must be set to 1 for Key ID Extension.

Extension Length: Must be set to 16, which represents 128 bits (16 bytes).

Extension: Must contain the Key ID value for the encrypted media delivered in conjunction with this descriptor. This extension is only used when the Encrypted Data flag is set to 1.

Initialization Vector (IV)

Extension Type: Must be set to 2 for Initialization Vector Extension.

Extension Length: Must be set to 8, which represents 64 bits (8 bytes).

Extension: Must contain the Initialization Vector for the encrypted media delivered in conjunction with this descriptor. This extension is only used when the Encrypted Data flag is set to 1.

With regard to the Length section, in this embodiment, this section must contain the total length of the descriptor in bytes. This length does not include the size of the media data delivered in conjunction with this descriptor.

Content-Independent Data Encryption

Fig. 11 is a flow diagram that describes steps in a method in accordance with one embodiment. This method can be performed in connection with any suitable hardware, software, firmware or combination thereof. In one embodiment, the method can be implemented in connection with systems, such as those illustrated and described above. Additionally, in the discussion that follows, some of the acts that are performed are depicted as being performed by a server/transmitter, and other acts are depicted as being performed by a client/receiver. Examples of server/transmitters and client/receivers are provided above.

Step 1102 receives a media file. The media file can have any content type permitting the media file to be broken into data segments, encrypted, transmitted, received, and decrypted. It can be, for instance, an ASF, MPEG2 TS, MPEG2 ES, or WAV file.

Step 1104 divides the media file into data segments. These data segments can comprise packets, other pieces of data, or frames conforming to various controls protocols, such as RTP or HTTP.

Step 1106 encrypts each data segment. Step 1106 may do so according to any of the embodiments described herein. Thus, it may encrypt the payload data with a leaf content key and encrypt that leaf content key with a root content key. With the root content key, a receiver may later decrypt the leaf content key and use that leaf content key to decrypt the payload data.

In one embodiment, step 1106 encrypts each data segment or part thereof using an AES in Counter mode. Fig. 12 illustrates a process for encrypting a single data segment using this technique. In this embodiment, Counter mode creates a stream of bytes that are then XOR'd with the clear text bytes of the data segment to create the encrypted data segment. The Key Stream Generator uses an AES round to generate 16-byte blocks of key stream at a time. The inputs to the AES round are the Content

Encryption key (K_C) (e.g., the leaf content key) and the 128-bit concatenation of a Data Segment ID and the block number within the data segment.

The output of key stream generator should be XOR'd byte by byte with the data from the corresponding block (i) of the data segment. In the case that the data segment is not evenly divisible by 16 bytes only the valid bytes of the media data from the last block should be XOR'd with the key stream and retained for the encrypted data segment.

Step 1108 adds a descriptor to each encrypted data segment. The descriptor can comprise a KID, IV, or other elements set forth herein. Each descriptor indicates an associated digital rights management policy by which the payload data of the data segment should be governed. This digital rights management policy, according to one embodiment above, is contained within a previously-received leaf license. Each descriptor can also indicate a content key (e.g., a particular leaf content key) usable to decrypt the data segment.

Note that the result of these steps can be a media file of an arbitrary content type broken into data segments, each data segment encrypted and having a descriptor by which the encrypted data can later be associated with a digital rights management policy.

In one embodiment, the descriptor contains a length indicator. With this length indicator, a receiver of an encrypted data segment can determine when the descriptor ends or begins. This length indicator permits the descriptor to be added to an encrypted data segment at various locations in the data segment or its packet. For the RTP protocol, for instance, the descriptor is added to the end of an RTP packet having the data segment. For the HTTP protocol, for instance, the descriptor is added to the beginning of the frame having the data segment. Note that the descriptor, by having a discernable length, can be added to various portions of a data segment and thus enable use of the descriptor with various transfer protocols.

Step 1110 transmits the encrypted data segments (and clear data segments, if any) with descriptors to a receiver. The receiver is enabled to orient (e.g., place in correct order) the data segments in manners known in the art. The receiver may decrypt the data segments using a content key associated with the data segments.

Further, the receiver, using the descriptor, can determine what rights policy should be used with the media file or a portion thereof. If the media file has portions that should be governed by different rights policies, this method can also divide the data segments based on their portion of the media file and assign different descriptors to data segments of different portions in step 1104.

Step 1112 receives and decrypts the encrypted data segments. A receiver (such as client/receiver 500 or 600) decrypts the data segments and assigns the appropriate rights policy to them based on their descriptor. In one embodiment, the receiver decrypts the data segments using an Initialization Vector in the descriptor. The receiver determines the appropriate leaf content key based on the KID, which it then uses to decrypt the data segments after decrypting the leaf content key with a root content key.

Step 1114 associates each data segment with a rights policy. In one embodiment, the receiver does so using a Key ID (KID) found in the descriptor and in the leaf license having the rights policy.

Using Root and Leaf Licenses

Fig. 13 is a flow diagram that describes steps in a method in accordance with one embodiment. This method can be performed in connection with any suitable hardware, software, firmware or combination thereof. In one embodiment, the method can be implemented in connection with systems, such as those illustrated and described above. Additionally, in the discussion that follows, some of the acts that are performed are depicted as being performed by a server/transmitter, and other acts are depicted as being performed by a client/receiver. Examples of server/transmitters and client/receivers are provided above.

Step 1300 encrypts a root content key using a public key of a client/receiver. Any suitable content key can be utilized with but one example being given above. Step 1302 sends a root license containing the encrypted root content key to a client/receiver. Any suitable method can be utilized to implement this step. In the discussion that follows, two specific examples that draw upon two different protocols are provided. It is to be appreciated and understood that these constitute examples and

are not intended to limit application of the claimed subject matter to only the specific protocols that are described.

Step 1304 receives the root license sent by the server/transmitter and step 1306 decrypts the encrypted root content key. In this example, this step is performed by using the client/receiver's private key to decrypt the encrypted root content key.

Step 1308 prepares a leaf license and encrypts a leaf content key with the root content key. Step 1310 sends the leaf license to the client/receiver. Recall that the leaf license can and typically does contain policies for DRM-protected content. It should be understood and appreciated that steps 1308 and 1310 can be executed multiple times for a given piece of DRM-protected content. That is, for each portion having a different policy, a corresponding leaf license can be prepared and sent to the client/receiver.

Step 1312 receives the leaf license and step 1314 decrypts the leaf content key using the root content key that was previously received. Step 1316 then uses the decrypted leaf content key to decrypt content. It also associates the appropriate leaf license with a portion of the media file (if the media file has portions) using a descriptor described above.

It is to be appreciated and understood that steps 1312, 1314 and 1316 can be performed for each new leaf license that is received by the client/receiver.

Conclusion

This document describes techniques by which a digital rights management policy may be associated with digital media having an arbitrary content type or transfer control protocol. In some cases this enables a receiver of an encrypted media file to decrypt the file and consume portions of the file according to different digital rights management policies. In some cases this also permits a transmitter to encrypt many different types of media files with one set of techniques. Although the invention has been described in language specific to structural features and/or methodological steps, it is to be understood that the invention defined in the appended claims is not necessarily limited to the specific features or steps described. Rather, the

specific features and steps are disclosed as preferred forms of implementing the claimed invention.

CLAIMS

1. A computer-implemented method comprising:

encrypting data segments of a media file having an arbitrary content type to provide encrypted data segments; and

adding descriptors to the encrypted data segments, each encrypted data segment's descriptor indicating an associated digital rights management policy for the encrypted data segment and Key ID usable to decrypt the encrypted data segment.

2. The method of claim 1, wherein the act of encrypting encrypts the data segments using an AES in counter mode.

3. The method of claim 1, wherein the descriptors comprise Initialization Vectors, each encrypted data segment's Initialization Vector associated with the content key usable to decrypt the encrypted data segment.

4. The method of claim 1, wherein each descriptor has a Key ID that is also in its associated digital rights management policy.

5. The method of claim 1, wherein each encrypted data segment's descriptor comprises a length indicator enabling differentiation between the encrypted data segment and its descriptor.

6. The method of claim 1, wherein the media file comprises a first portion and a second portion, the first portion associated with a first digital rights management policy and the second portion associated with a second digital rights management policy, and wherein the act of adding adds a first descriptor indicating the first digital rights management policy to encrypted data segments of the first portion and adds a second descriptor indicating the second associated digital rights management policy to encrypted data segments of the second portion.

7. The method of claim 1, wherein the encrypted data segment comprises a packet conforming to an RTP data protocol or a frame conforming to an HTTP protocol.

8. The method of claim 7, wherein the act of adding concatenates the descriptor to the end of the encrypted data segment if the encrypted data segment comprises a packet conforming to an RTP protocol or the beginning of the encrypted data segment if the encrypted data segment comprises a frame conforming to an HTTP protocol.

9. A system comprising one or more computer-readable media, the computer-readable media comprising a digital media file comprising data segments, each data segment added to a descriptor and comprising encrypted payload data, each data segment's descriptor having encryption parameters enabling decryption of the data segment's payload data and association of the data segment's payload data with a digital rights management policy.

10. The system of claim 9, wherein each descriptor's encryption parameters comprise a Key ID also comprised by the digital rights management policy.

11. The system of claim 9, wherein each data segment's payload data is encrypted using a content key and the data segment's descriptor comprises a Key ID associated with the content key, the content key usable to decrypt the payload data.

12. The system of claim 9, wherein the digital media file further comprises first and second portions, the first portion having first data segments and the second portion having second data segments, at least one of the first data segment's descriptors enabling the first portion to be associated with a first digital rights management policy and at least one of the second data segment's descriptors enabling the second portion to be associated with a second digital rights management policy.

13. The system of claim 9, wherein each data segment's payload data is encrypted and its descriptor is not.

14. The system of claim 9, wherein the digital media file further comprises a data segment added to a descriptor and comprising clear payload data, the data segment's descriptor enabling association of the data segment's clear payload data with another digital rights management policy.

15. The system of claim 9, wherein the digital media file is streaming media.

16. A computer-implemented method comprising:

transmitting an encrypted media file having a first portion and a second portion to a receiver;

enabling the receiver of the encrypted media file to decrypt the first portion and the second portion; and

enabling the receiver to associate the first portion with a first rights policy indicating permitted usage of the first portion, and the second portion with a second rights policy indicating permitted usage of the second portion.

17. The method of claim 16, further comprising transmitting the first rights policy and the second rights policy to the receiver.

18. The method of claim 16, wherein the act of enabling the receiver to associate builds a first descriptor into the first portion and a second descriptor into the second portion, the first descriptor indicating that the first portion is associated with the first rights policy and the second descriptor indicating that the second portion is associated with the second rights policy.

19. The method of claim 18, wherein the act of enabling the receiver to decrypt builds a first Key ID into the first descriptor and a second Key ID into the second descriptor, the first Key ID usable to decrypt the first portion and the second Key ID usable to decrypt the second portion.

20. The method of claim 16, further comprising receiving a media file of an arbitrary content type and encrypting the media file to provide the encrypted media file.

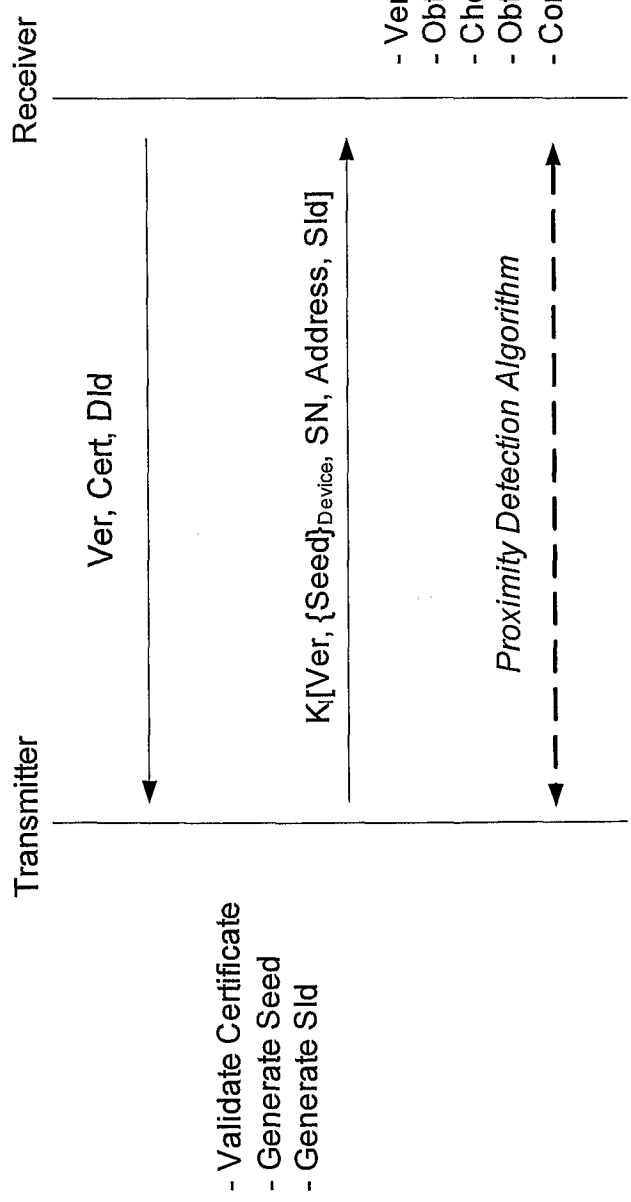

Transmitter
Receiver
- Validate Certificate
- Generate Seed
- Generate SId
Ver, Cert, DId
K_I[Ver, {Seed}$_{Device}$, SN, Address, SId]
Proximity Detection Algorithm
- Verify Signature
- Obtain SId
- Check Serial Number
- Obtain Seed
- Compute K_C and K_I

Fig. 1

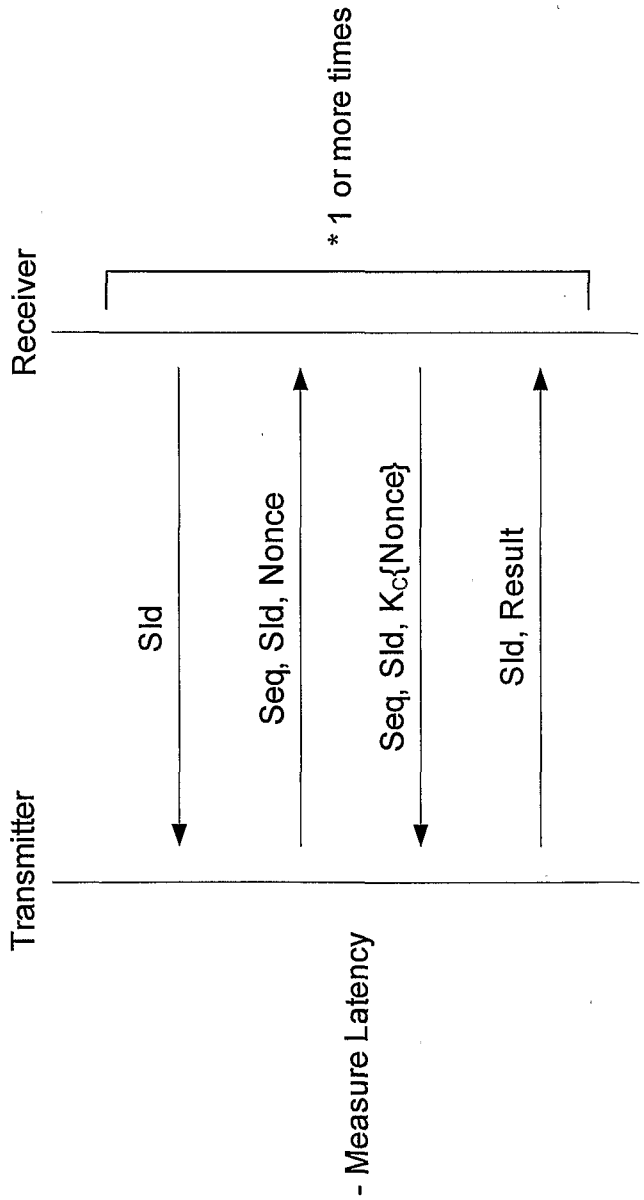

Receiver
Transmitter
* 1 or more times
SId
Seq, SId, Nonce
Seq, SId, K_{C}{Nonce}
SId, Result
- Measure Latency

Fig. 2

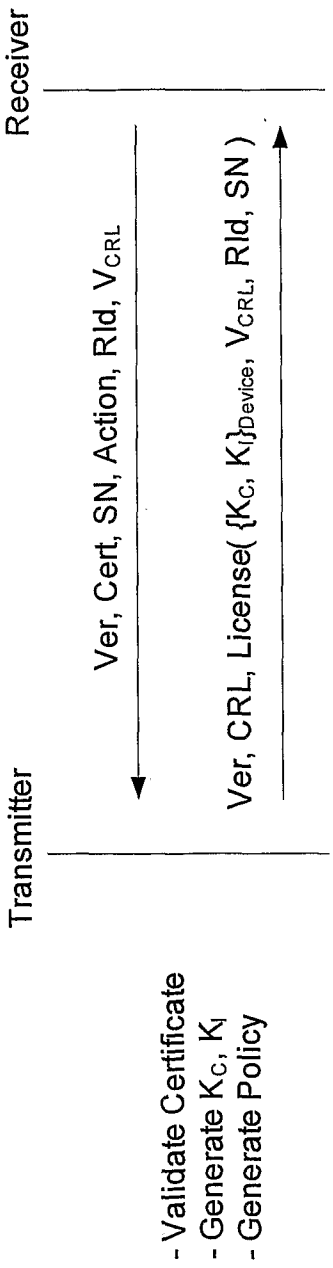

Receiver
Transmitter
Ver, Cert, SN, Action, RId, V_{CRL}
Ver, CRL, License($\{K_C, K_I\}_{Device}$, V_{CRL}, RId, SN)
- Validate Certificate
- Generate K_C, K_I
- Generate Policy

Fig. 3

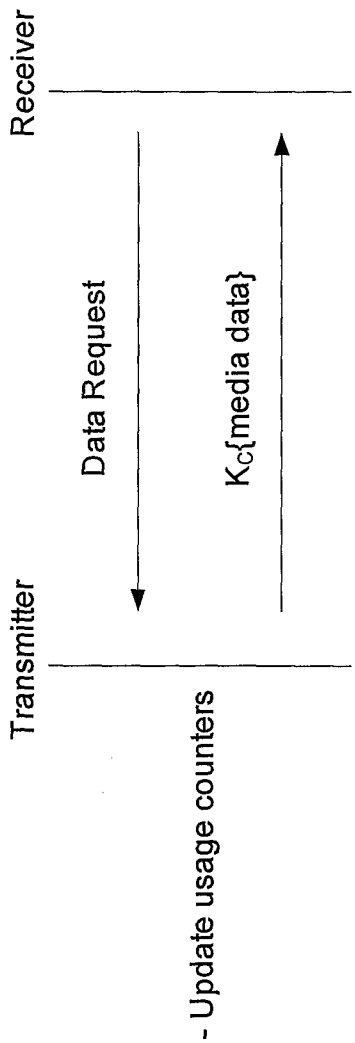

Receiver
Data Request
K_C{media data}
Transmitter
- Update usage counters

Fig. 4

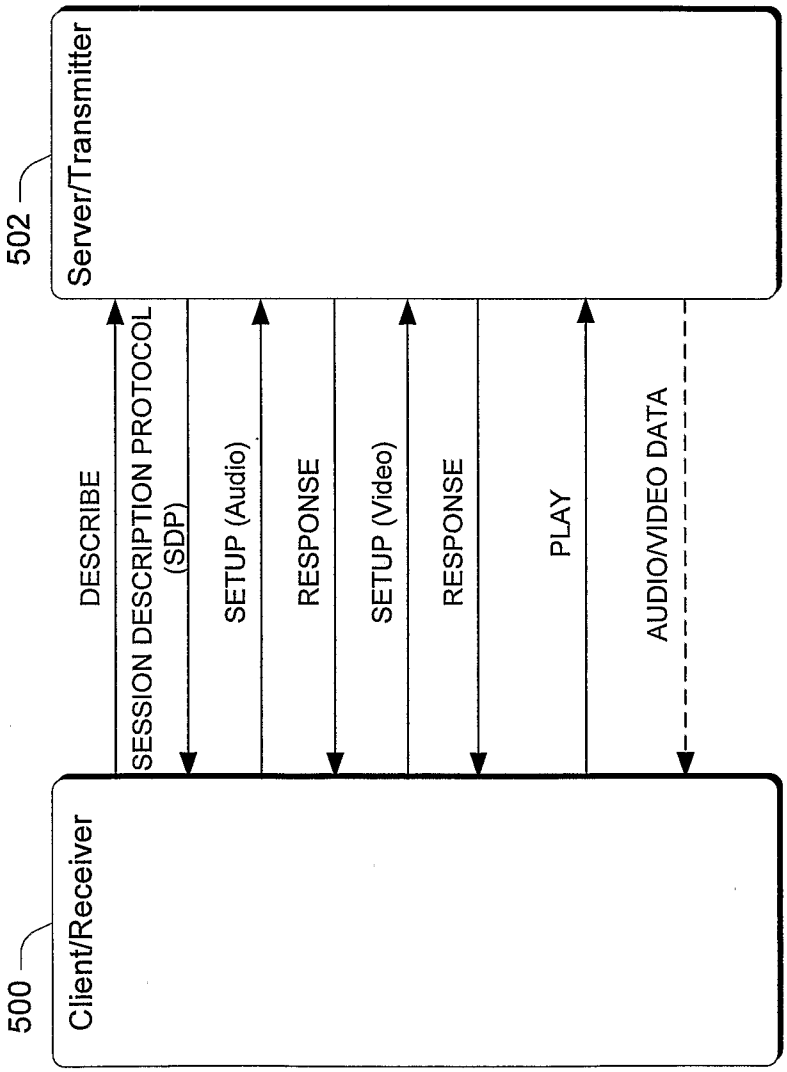

Server/Transmitter
502
Client/Receiver
500
DESCRIBE
SESSION DESCRIPTION PROTOCOL (SDP)
SETUP (Audio)
RESPONSE
SETUP (Video)
RESPONSE
PLAY
AUDIO/VIDEO DATA

Fig. 5

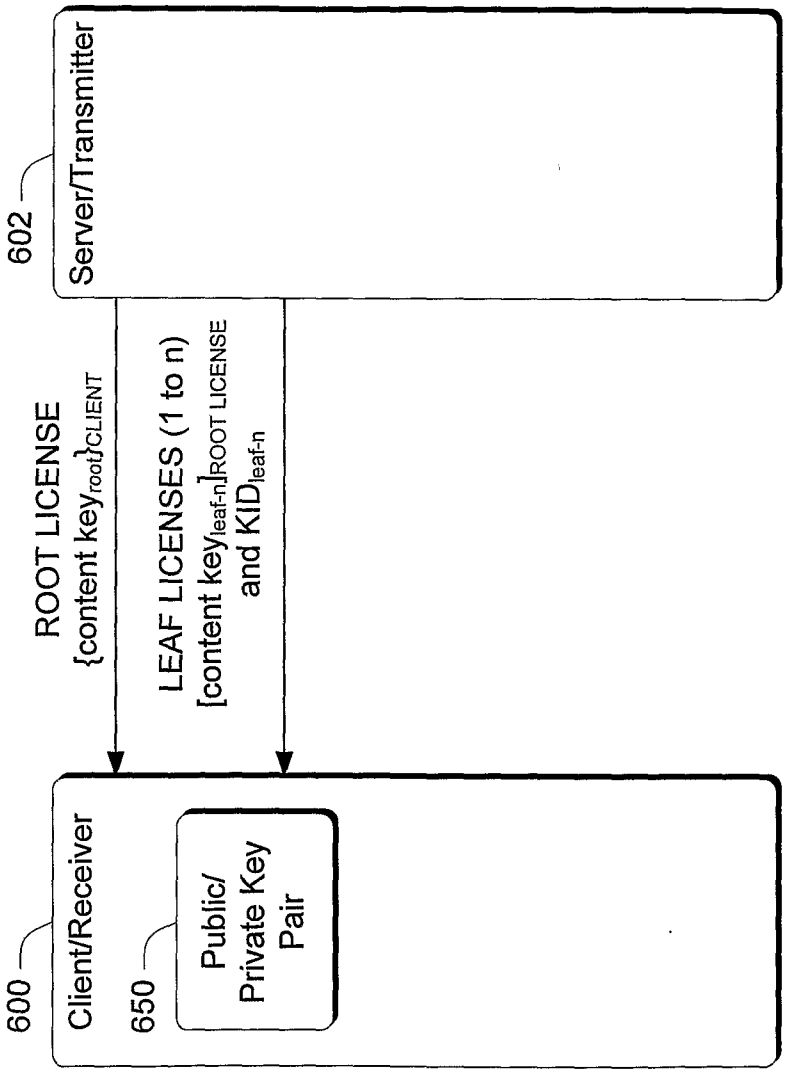

602
Server/Transmitter
ROOT LICENSE
{content key$_{root}$}$_{CLIENT}$
LEAF LICENSES (1 to n)
[content key$_{leaf-n}$]$_{ROOT\ LICENSE}$
and KID$_{leaf-n}$
600
Client/Receiver
650
Public/
Private Key
Pair

Fig. 6

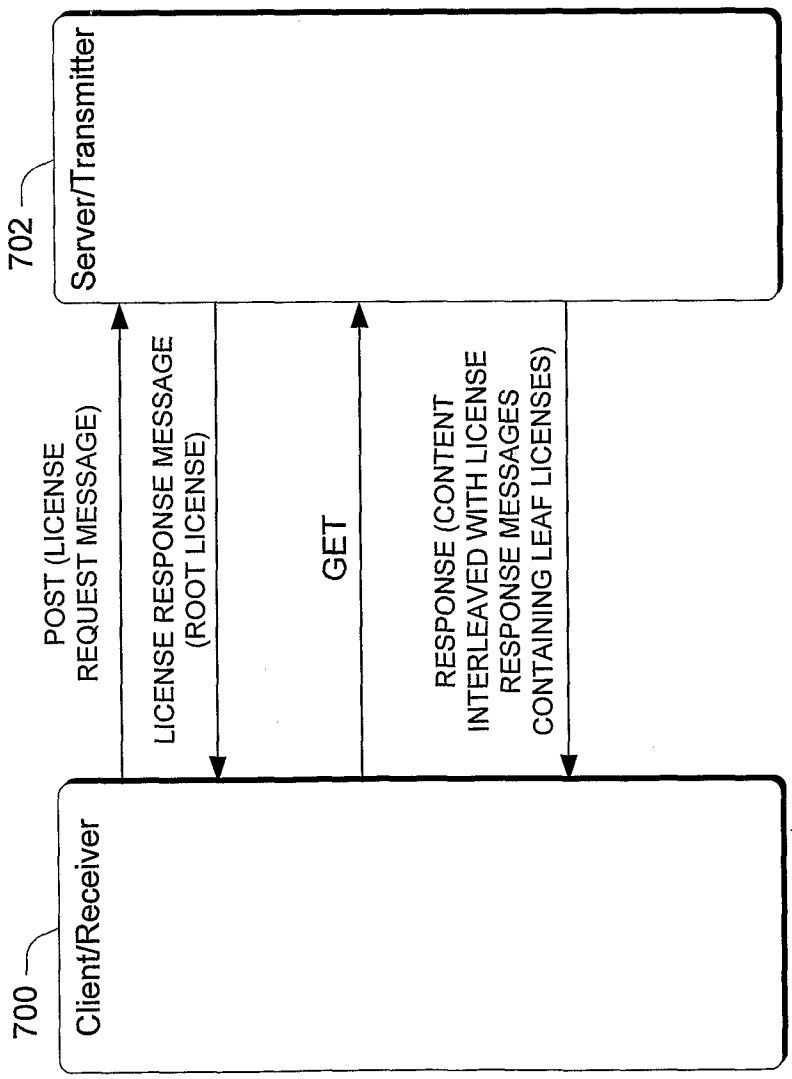

Server/Transmitter
702
Client/Receiver
700
POST (LICENSE REQUEST MESSAGE)
LICENSE RESPONSE MESSAGE (ROOT LICENSE)
GET
RESPONSE (CONTENT INTERLEAVED WITH LICENSE RESPONSE MESSAGES CONTAINING LEAF LICENSES)

Fig. 7

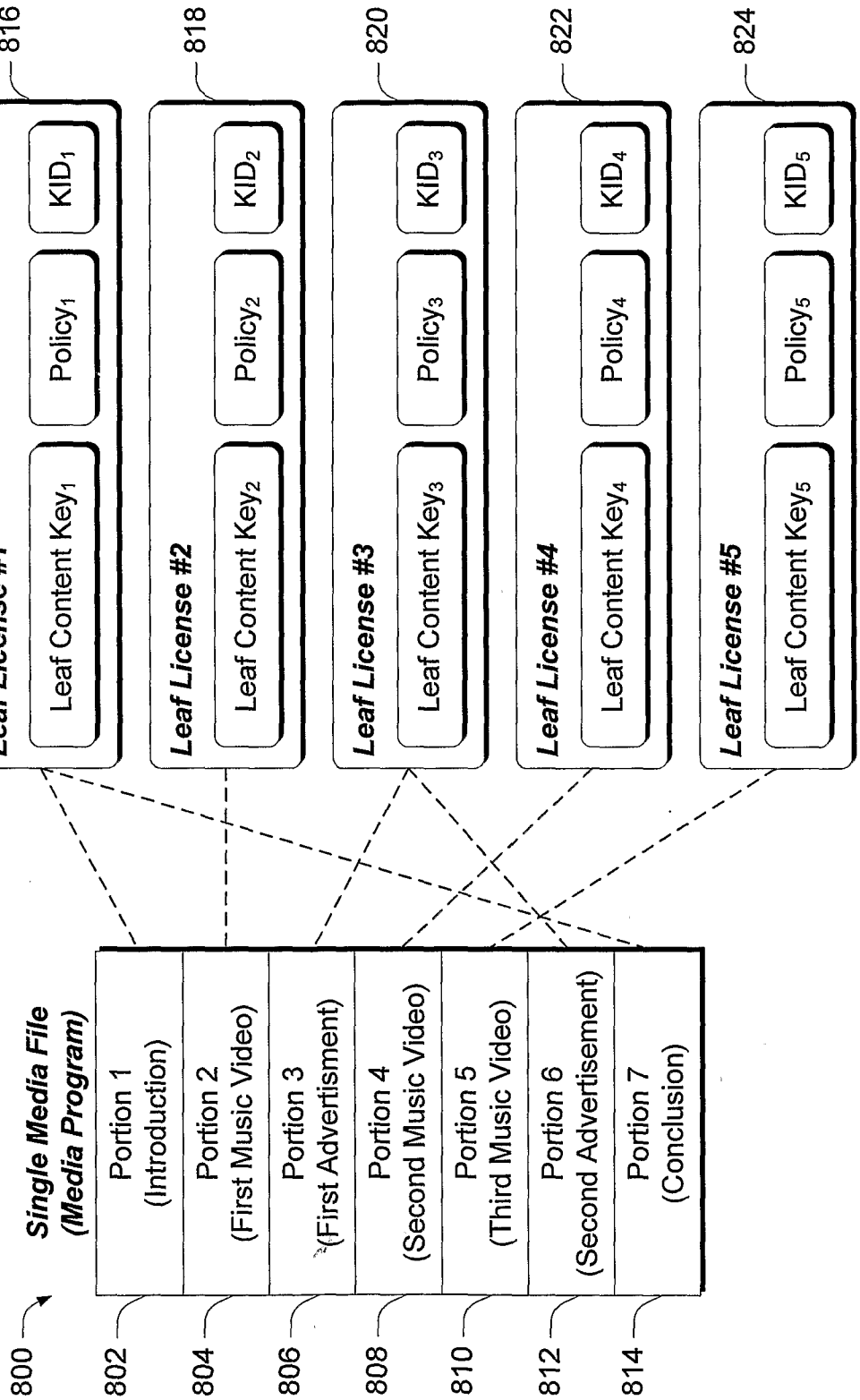

800
Single Media File
(Media Program)
802
Portion 1
(Introduction)
804
Portion 2
(First Music Video)
806
Portion 3
(First Advertisment)
808
Portion 4
(Second Music Video)
810
Portion 5
(Third Music Video)
812
Portion 6
(Second Advertisement)
814
Portion 7
(Conclusion)
816
Leaf License #1
Leaf Content Key1
Policy1
KID1
818
Leaf License #2
Leaf Content Key2
Policy2
KID2
820
Leaf License #3
Leaf Content Key3
Policy3
KID3
822
Leaf License #4
Leaf Content Key4
Policy4
KID4
824
Leaf License #5
Leaf Content Key5
Policy5
KID5

Fig. 8

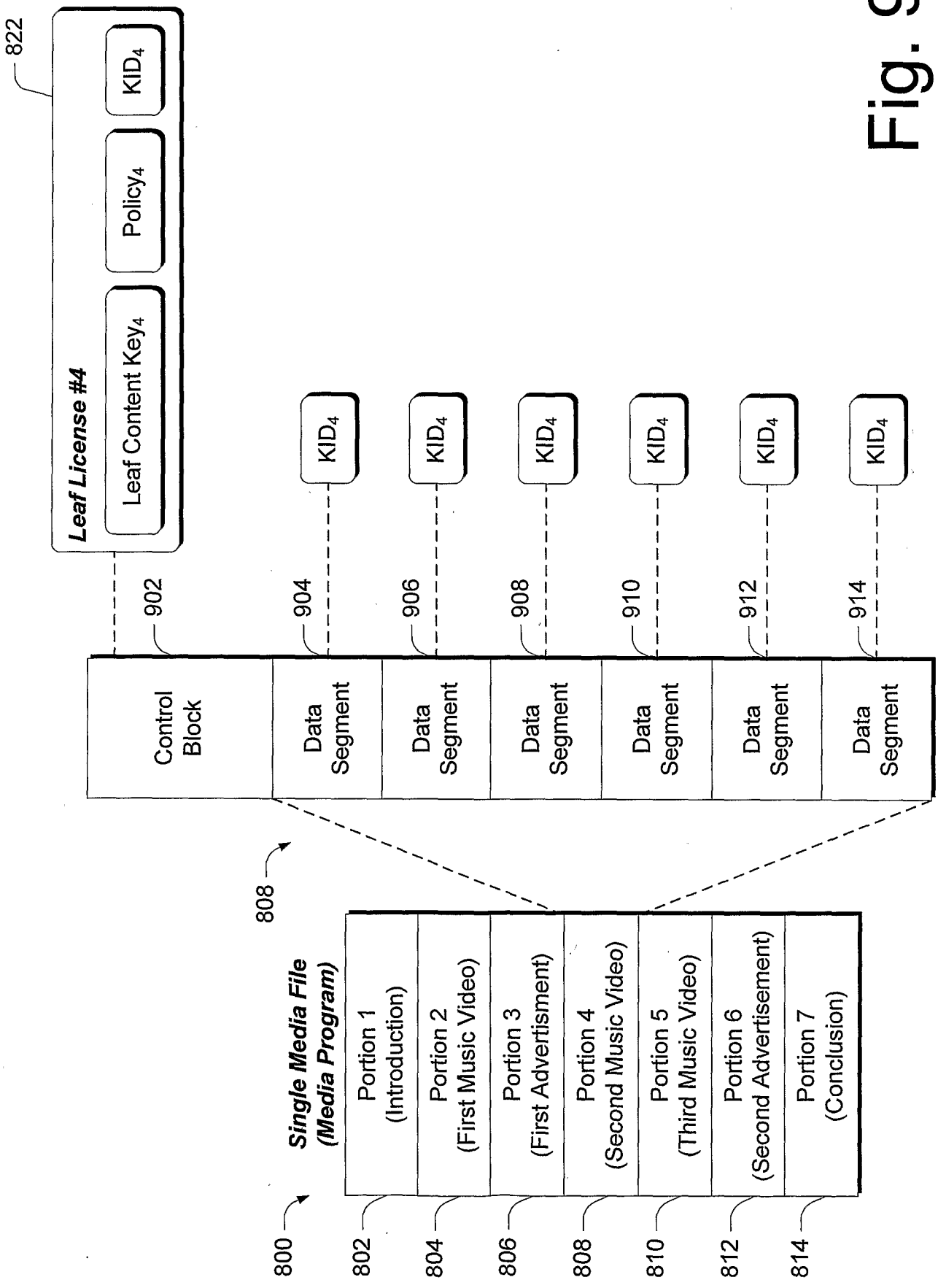

822
Leaf License #4
Leaf Content Key4
Policy4
KID4
902
Control Block
904
906
908
910
912
914
Data Segment
KID4
808
Single Media File (Media Program)
800
802
804
806
808
810
812
814
Portion 1 (Introduction)
Portion 2 (First Music Video)
Portion 3 (First Advertisment)
Portion 4 (Second Music Video)
Portion 5 (Third Music Video)
Portion 6 (Second Advertisement)
Portion 7 (Conclusion)

Fig. 9

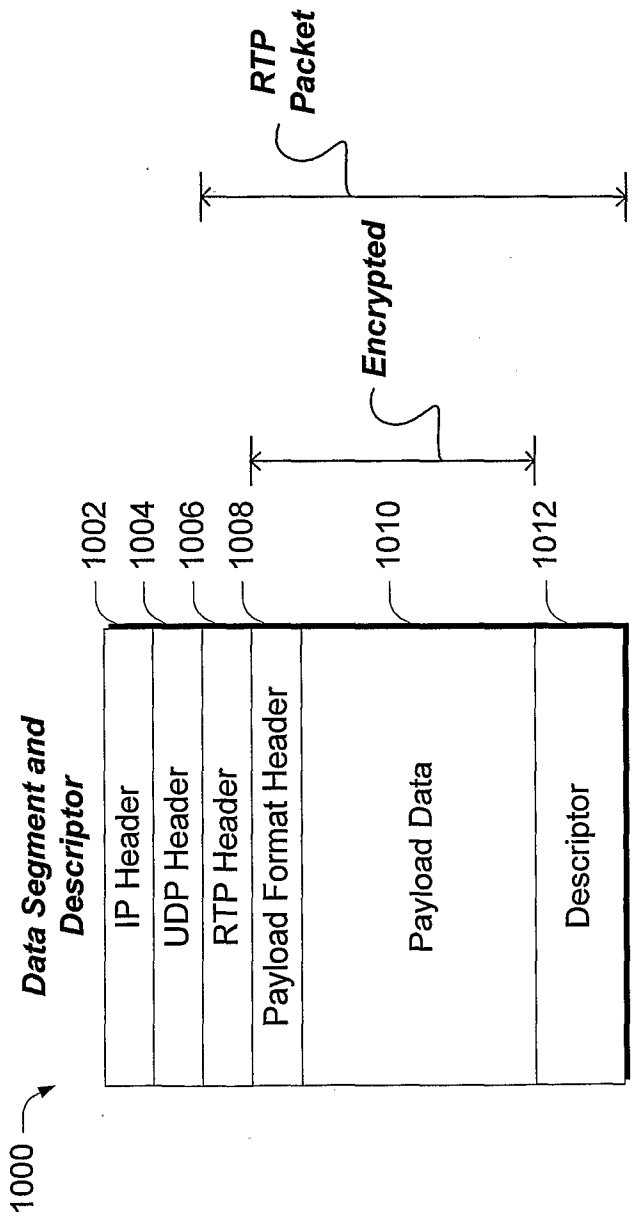

1000
Data Segment and Descriptor
IP Header
UDP Header
RTP Header
Payload Format Header
Payload Data
Descriptor
1002
1004
1006
1008
1010
1012
Encrypted
RTP Packet

Fig. 10

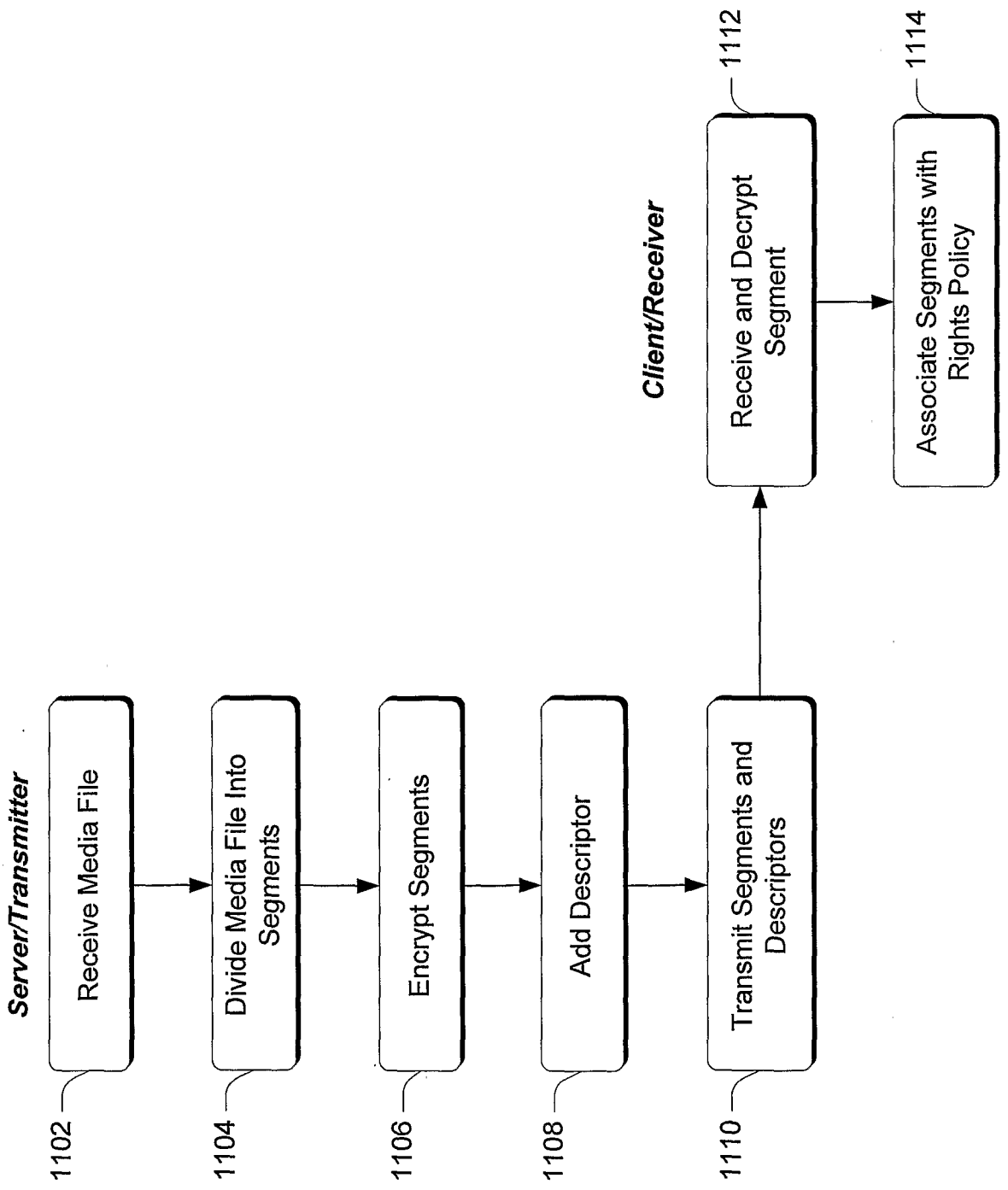

Server/Transmitter
Receive Media File
Divide Media File Into Segments
Encrypt Segments
Add Descriptor
Transmit Segments and Descriptors
Client/Receiver
Receive and Decrypt Segment
Associate Segments with Rights Policy
1102
1104
1106
1108
1110
1112
1114

Fig. 11

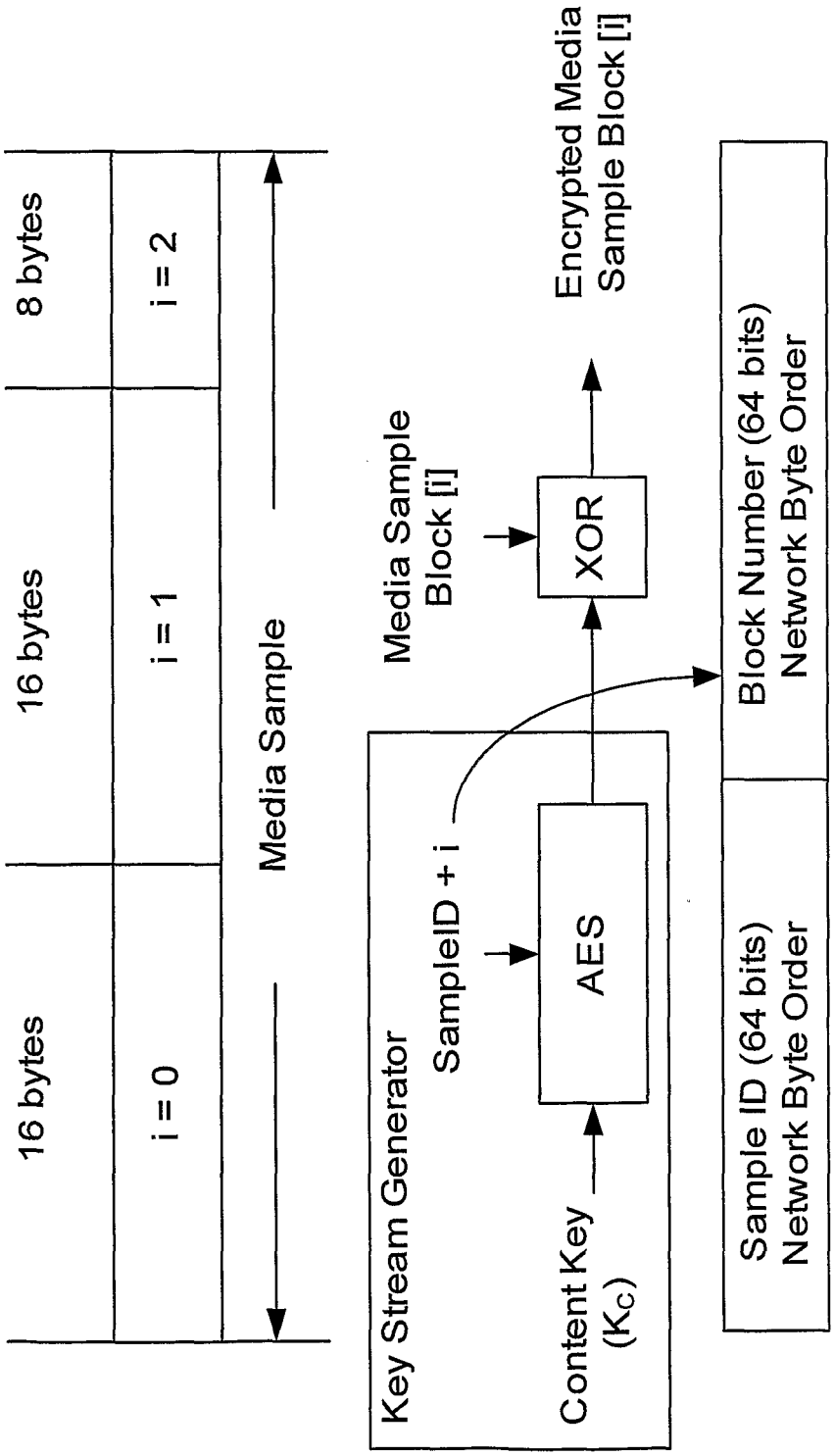

16 bytes
16 bytes
8 bytes
i = 0
i = 1
i = 2
Media Sample
Key Stream Generator
SampleID + i
AES
Content Key
(K_C)
Media Sample
Block [i]
XOR
Encrypted Media
Sample Block [i]
Block Number (64 bits)
Network Byte Order
Sample ID (64 bits)
Network Byte Order

Fig. 12

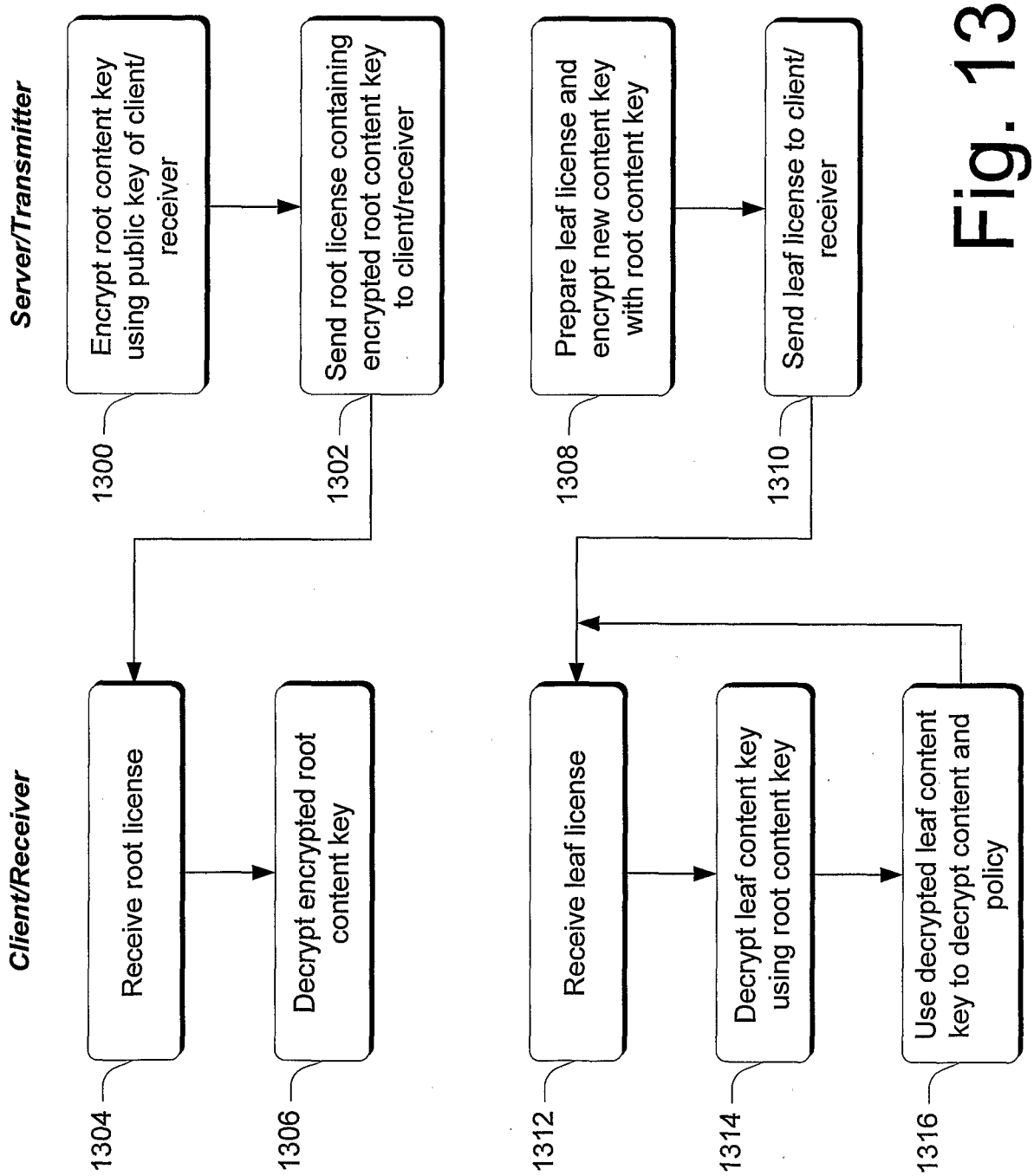

Server/Transmitter
Client/Receiver
Encrypt root content key using public key of client/ receiver
Send root license containing encrypted root content key to client/receiver
Receive root license
Decrypt encrypted root content key
Prepare leaf license and encrypt new content key with root content key
Send leaf license to client/ receiver
Receive leaf license
Decrypt leaf content key using root content key
Use decrypted leaf content key to decrypt content and policy
1300
1302
1304
1306
1308
1310
1312
1314
1316

Fig. 13

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2006/031185

A. CLASSIFICATION OF SUBJECT MATTER

G06F 17/00(2006.01)i, H04L 9/32(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC8 H04N 7/173; IPC8 H04L 9/32; IPC8 H04L 9/00; IPC8 G06F 17/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean patents and applications for inventions since 1975.
Korean utility models and applictions for utility models since 1975.
Japanese utility models and application for utility models since 1975.

Electronic data base consulted during the intertnational search (name of data base and, where practicable, search terms used)

e-KIPASS "ENCRYPTION, DRM, MEDIA, KEY, DESCRIPTOR"

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	WO 2004/030364 A1 (MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD.) 8 APR. 2004 See page 1,line 28-page2,line 4; page 10,line 2-line 7; page 13, lines 15-14, line 6; fig. 3; fig. 4.	1-20
Y	WO 2004/023717 A2 (SONY ELEC. INC.) 18 MAR. 2004 See abstract; claim 1.	1-20
A	US 2002/0002674 A1 (GRIMES, T. et al.) 3 JAN. 2002 See abstract	1-20
A	US 2004/0143736 A1 (CROSS, D. B. et al.) 22 JUL. 2004 See abstract	1-20

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
16 JANUARY 2007 (16.01.2007)	**16 JANUARY 2007 (16.01.2007)**

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
920 Dunsan-dong, Seo-gu, Daejeon 302-701,
Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

YUK, SEONG WON

Telephone No. 82-42-481-8213

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/US2006/031185

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO2004030364A1	08.04.2004	CN1613257A	04.05.2005
		EP1547387A1	29.06.2005
		JP16145867	20.05.2004
		JP2004145867A2	20.05.2004
		KR2005061395A	22.06.2005
		US20040249759A1	09.12.2004
		US2004249759AA	09.12.2004
WO2004023717A2	18.03.2004	AU2003234690A1	03.11.2003
		AU2003268468A1	29.03.2004
		AU2003296903A1	04.05.2004
		CA2413807A1	02.07.2003
		CA2437014A1	09.03.2004
		CA2480964A1	30.10.2003
		CA2498326A1	18.03.2004
		CA2498346A1	29.04.2004
		CN1659819A	24.08.2005
		CN1682486A	12.10.2005
		EP01495575A1	12.01.2005
		EP01543650A2	22.06.2005
		JP2005525010T2	18.08.2005
		JP2005538453T2	15.12.2005
		KR20040098074	18.11.2004
		KR20050046750	18.05.2005
		US2003145329A1	31.07.2003
		US2003152224A1	14.08.2003
		US2003156718A1	21.08.2003
		US2003159139A1	21.08.2003
		US2003174837A1	18.09.2003
		US2004047470A1	11.03.2004
		US2004073917A1	15.04.2004
		US2005028193A1	03.02.2005
		US2005192904A1	01.09.2005
		US7039938BB	02.05.2006
		US7120250BB	10.10.2006
		US7151833BB	19.12.2006
		WO2003090401A1	30.10.2003
		WO2004023717A3	29.12.2004
		WO2004036892A2	29.04.2004
		WO2004036892A3	23.06.2005
US20020002674A1	03.01.2002	US07036011	25.04.2006
		US2002002674AA	03.01.2002
		US7036011BB	25.04.2006
US20040143736A1	22.07.2004	US2004143736AA	22.07.2004