Description

METHOD AND DEVICE FOR CONSUMING RIGHTS OBJECTS HAVING INHERITANCE STRUCTURE

Technical Field

[1] Methods and apparatuses consistent with the present invention relate to digital rights management, and more particularly, to consuming a rights object (RO) having an inheritance structure in an environment where the RO is distributed over a plurality of devices.

Background Art

[2] Recently, digital rights management (hereafter, referred to as DRM) has been actively researched and developed. Commercial services using DRM have already been used or will be used. DRM needs to be used because of the following various characteristics of digital content; digital content can be copied and easily distributed.

[3] There were several efforts to protect digital content. Conventionally, digital content protection has been concentrated on preventing non-permitted access to digital content, permitting only people who paid charges to access the digital content. Thus, people who paid charges to the digital content are allowed to unencrypted digital content while people who did not pay charges are not allowed to. In this case, when a person paid charges intentionally distributes the digital content to other people, however, the people can use the digital content without paying charges.

[4] To solve this program, DRM was introduced. In DRM, any one is allowed to freely access encoded digital content, but a license referred to as an RO is needed to decode and execute the digital content.

[5] Referring to FIG. 1, a device 10 obtains digital content from a content provider 20. Digital content provided by the content provider 20 is encrypted, and thus, in order to use the digital content, an RO corresponding to the digital content is needed.

[6] The device 10 can obtain the RO with a license to play the encrypted content from received from an RO issuer 30. To this end, a user should pay charges. The encrypted digital content is decrypted using a key contained in the RO.

[7] The RO issuer 30 makes the content provider 20 prepare an RO issuing detail report. The RO issuer 30 and the content provider 20 may be the same authority.

[8] After obtaining the RO, the device 10 consumes the RO to use the encrypted digital content.

[9] The encrypted digital content can be freely copied and distributed to another device (not shown). However, since an RO contains constraint information such as Count, Interval or Copy, unlike the encrypted digital content, the RO has a limitation in its

reuse or replication. Accordingly, the digital content can be more effectively protected by using DRM.

[10] Recent DRM techniques enable content object playback services to be provided to users on a subscription basis by playing back content objects using a plurality of ROs having an inheritance structure. A device generally needs two or more ROs that are in inheritance relationship to use an encrypted content object. A plurality of ROs needed for using a single encrypted content object may be stored in different devices. Therefore, it is necessary to develop methods to efficiently consume a plurality of ROs needed for using a single encrypted content object.

Disclosure of Invention

Technical Problem

[11] Recently, methods for facilitating the storage and distribution of ROs by managing the ROs with the aid of a portable storage device such as a memory stick or a multimedia card have been developed. However, portable storage devices generally provide poorer operation capabilities than digital content playback devices. Therefore, it is necessary to develop methods to reduce the operation load of a portable storage device when a device consumes a plurality of ROs that are stored in the portable storage device and have an inheritance structure.

Technical Solution

[12] The present invention provides a method and device for enabling a plurality of ROs distributed over a plurality of devices to be efficiently consumed.

[13] According to an aspect of the present invention, there is provided a master device including a storage module which stores a plurality of ROs having an inheritance structure, a detection module which searches the storage module for an RO needed for playing back a predetermined content object, a message generation module which, if the detection module fails to discover the RO needed for playing back the predetermined content object from the storage module, generates an RO request message, a device interface module which transmits the RO request message to a slave device and receives the RO needed for playing back the predetermined content object from the slave device, and a playback module which plays back the predetermined content object using at least one of the RO detected from the storage module and the RO received from the slave device.

[14] According to another aspect of the present invention, there is provided a method of consuming a plurality of ROs including obtaining a plurality of ROs having a predetermined inheritance structure from a slave device, and playing back a predetermined content object using the ROs.

Description of Drawings

[15] The above and other aspects of the present invention will become more apparent by describing in detail exemplary embodiments thereof with reference to the attached drawings in which:

[16] FIG. 1 is a diagram of a typical DRM system;

[17] FIG. 2 is a diagram of a DRM system according to an exemplary embodiment of the present invention;

[18] FIG. 3 is a flowchart illustrating authentication between a master device and a slave device according to an exemplary embodiment of the present invention;

[19] FIG. 4 is a flowchart illustrating a method of obtaining a child RO according to an exemplary embodiment of the present invention;

[20] FIG. 5 is a flowchart illustrating a method of obtaining a parent RO according to an exemplary embodiment of the present invention;

[21] FIG. 6 is a diagram of a parent RO according to an exemplary embodiment of the present invention;

[22] FIG. 7 is a diagram of a child RO according to an exemplary embodiment of the present invention;

[23] FIG. 8 is a block diagram of a master device according to an exemplary embodiment of the present invention;

[24] FIG. 9 is a block diagram of a slave device according to an exemplary embodiment of the present invention;

[25] FIG. 10 is a flowchart illustrating a method of consuming ROs having an inheritance structure according to an exemplary embodiment of the present invention;

[26] FIG. 11 is a flowchart illustrating the providing of ROs by a slave device according to an exemplary embodiment of the present invention.

Mode for Invention

[27] Advantages and features of the present invention and methods of accomplishing the same may be understood more readily by reference to the following detailed description of exemplary embodiments and the accompanying drawings. The present invention may, however, be embodied in many different forms and should not be construed as being limited to the exemplary embodiments set forth herein. Rather, these exemplary embodiments are provided so that this disclosure will be thorough and complete and will fully convey the concept of the invention to those skilled in the art, and the present invention will only be defined by the appended claims. Like reference numerals refer to like elements throughout the specification.

[28] Several terms used herein will first be described in a brief manner for better understanding of the present description. Thus, it should be noted that this description will not be interpreted to limit the protection scope of the present invention as long as there is no express limitation on them. - Master Device

[29] A master device is a device which can be connected to a slave device and can play back a content object using a plurality of ROs having a reciprocal inheritance structure. The master device may be a portable content playback device such as a mobile phone, a personal digital assistant (PDA), or an MP3 player or a non-portable content playback device such as a desktop computer or a digital TV. - Slave Device

[30] A slave device is a storage device including a non-volatile memory such as a flash memory to/from which data can be written/read, has predetermined data operation capabilities, and can be easily connected to/disconnected from a master device. The slave device may be a portable storage device such as a smart media card, a memory stick, a CF card, an XD card, or a multimedia card. However, the slave device may be a device other than those set forth herein or may be a device of a similar type to that of a master device. - Content Objects

[31] Content objects are encrypted digital content. Here, digital content may be a moving picture, a still picture, audio data, a game, or text data. - ROs

[32] ROs are licenses for using content objects. ROs comprise at least one of a content encryption key, constraint information, and permission information. In the present invention, ROs are classified into parent ROs and child ROs. Parent ROs have an inheritance relationship with respective corresponding child ROs. Parent ROs comprise at least one of constraint information and permission information, while child ROs comprise content object identifiers and content encryption keys. For a better understanding of the present invention, it will now be assumed that child ROs, instead of parent ROs, comprise constraint information, but the present invention is not restricted to it. Therefore, entire constraint information or part of the constraint information may be included in child ROs. The functions of constraint information included in child ROs can be fully understood by understanding the functions of constraint information included in parent ROs. Each parent ROs and its corresponding child ROs have the same inheritance relationship identifiers so that the inheritance relationships between the parent RO and the corresponding child ROs can be identified by the inheritance relationship identifiers. A single parent RO may have an inheritance relationship with a plurality of child ROs. Also, a single child RO may have an inheritance relationship with a plurality of parent ROs. - Content Encryption Keys

[33] Content encryption keys are keys used for decrypting content objects and have predetermined binary values. Content encryption keys are included in child ROs, and child ROs comprise content object identifiers identifying content objects that can be decrypted using content encryption keys included in the child ROs. - Permission Information

[34] Permission information indicates play-back and reproduction types of a content object.

[35] Example of the play back include 'Play', 'Display', 'Execute', and 'Print'. The Play component indicates a right to express content in an audio/video format. In addition, the Display component indicates a right to display a content object through a visual device, and the Print component indicates a right to generate a hard copy of a content object. For example, in a case where the content object is a moving picture or music, at least one of the display component and print component may be set as permission information of an RO to be consumed to play the content object. The Execute component indicates a right to execute a content object such as games and other application programs. For example, in a case where the content object is a JAVA game, the Execute component may be set as permission information of an RO to be consumed to play the JAVA game.

[36] Meanwhile, examples of the duplication include the Copy component and the Move component. The copy component and the move component are rights to move an RO from one device to another device and to store the same. The Move component deactivates the original RO in the current device, while the Copy component does not deactivate the original RO in the current device. Here, deactivation may mean deletion of an RO. - Constraint Information

[37] Constraint information refers to constraints allowing an RO to be played back and one or more pieces of constraint information may be set. Examples of the constraint information include Count constraint, Datetime constraint, Interval constraint, Accumulated constraint, and so on.

[38] Here, the Count constraint specifies the count of permissions granted to a content object. When the count constraint is set to 10, the master device is allowed to play the content object 10 times until the count constraint for the RO is consumed.

[39] The Datetime constraint specifies duration for permission and selectively contains a start element or an end element. When an RO with a Datetime constraint set is consumed, the master device can play a content object after and before a time/date specified by a start item of the datetime constraint. For example, when the start item is set to 00:00:00 (hour: minute: second) 2005-12-01 (year-month-day), the master device cannot access to and consume the RO for play the content object before 00:00:00 2005-12-08.

[40] The Interval constraint specifies a time interval at which an RO can be executed for the corresponding content object. When a start element is contained in the Interval constraint, consumption of the content object is permitted during a period of time specified by a duration element contained in the Interval constraint after a specified time/date. For example, let the interval constraint of be one week. When the master device consumes an RO on and after 00:00:00 2005-12-01 to play a content object, consumption of the RO for playing the content object is permitted by 00:00:00

2005-12-08.

[41] The Accumulated constraint specifies a maximum time interval for an accumulated measured period of time while the RO is executed for the corresponding content object. When an RO has the accumulated constraint set to 10, the master device can play a content object for 10 hours. In this instance, the master device is not limited by the Count or Datetime. - Public-Key Cryptography

[42] This is also referred to as asymmetric cryptography because encryption is made when a key used in decrypting data and a key used in encrypting the data constitute different encryption keys. In public-key cryptography, an encryption key consists of a pair of a public key and a private key. The public key is not necessary to be kept in secret, i.e., the public is easily accessible thereto while the private key must be known only to a specific device. This public key encryption algorithm has been disclosed to the general public but a third person cannot know or hardly know the original content with encryption algorithm, encryption key and ciphered text. There are examples of public key encryption algorithm such as Diffie-Hellman, RSA, El Gamal, Elliptic Curve, etc. - Symmetric-Key Cryptography

[43] This is also referred to as secret key cryptography, wherein encryption is made when a key used to encrypt data and a key used to decrypt the data constitute the same encryption key. As an example of such symmetric key encryption, data encryption standard (DES) method is used most generally, but application adopting advanced encryption standard (AES) method has recently been increased. - Random Number

[44] A random number is a sequence of numbers or characters with random properties. Since it costs a lot to generate a complete random number, a pseudo-random number may be used. - Connection

[45] When a master device is connected with a slave device in the present invention, they are electrically connected with each other via each wired medium. However, this is just an example, and 'being connected' simply implies that two devices can communicate with each other through a wireless medium in a non-contact state.

[46] Terms specifically defined above will be described below when necessary.

[47] FIG. 2 is a diagram of a DRM system according to an exemplary embodiment of the present invention.

[48] Referring to FIG. 2, the DRM system includes a master device 100 and a slave device 200.

[49] A user can obtain content objects from a content provider 20 via the master deice 100 in the same manner as in a conventional typical DRM system. In addition, the user can purchase ROs needed for playing back content objects from an RO issuer 30. The purchased ROs may be stored in the master device 100 and then moved to/copied to the slave device 200. The slave device 200 may be equipped with one or more ROs

when manufacturing the slave device 200.

[50] If the slave device 200 stores ROs, the master device 100 may play back content objects by being connected to the salve device 200 and consuming the ROs stored in the slave device 200. In order to play back a content object, the master device 100 may use a plurality of ROs having an inheritance structure. Here, the ROs having an inheritance structure may comprise at least one parent RO and at least one child RO. For convenience, it will now be assumed that a content object is played back by consuming a parent RO and a child RO having an inheritance relationship, but the present invention is not restricted to it.

[51] The master device 100 and the slave device 200 may authenticate each other before being connected to each other and exchanging data with each other. The authentication between the master device 100 and the slave device 200 is carried out for determining whether the master device 100 and the slave device 200 are legitimate and maintaining the security of data transmitted between the master device 100 and the slave device 200. This will now be described in detail with reference to FIG. 3.

[52] FIG. 3 is a flowchart illustrating an authentication procedure according to an exemplary embodiment of the present invention.

[53] In the illustrative embodiment, a subscript 'M' of data indicates that the data is possessed or generated by a master device 100 and a subscript 'S' of data indicates that the data is possessed or generated by a slave device 200.

[54] In operation S10, the master device 100 sends an authentication request to the slave device 200. When requesting authentication, the master device 100 may send the slave device 200 a $certificate_M$, which was issued to the master device 100 by a certification authority. The $certificate_M$ is signed with a digital signature of the certification authority and contains a device ID_M and the $public\ key_M$.

[55] In operation S12, the slave device 200 verifies whether the $certificate_M$ of the master device 100 is valid using a certificate revocation list (CRL). If the $certificate_M$ is registered in the CRL, the slave device 200 may reject the authentication with the master device 100. If the $certificate_M$ is not registered in the CRL, the slave device 200 obtains the $public\ key_M$ using the $certificate_M$ of the master device 100.

[56] If it is determined that the master device 100 is verified as an authenticated device, that is, the $certificate_M$ of the master device 100 is valid, in operation S14, the slave device 200 generates a $random\ number_S$. In operation S16, the generated $random\ number_S$ is encrypted using the $public\ key_M$.

[57] In operation S20, the slave device 200 performs an authentication response procedure. During the authentication procedure, the slave device 200 sends a $certificate_S$, which was issued to the slave device 200 by the certification authority, and the encrypted $random\ number_S$. The $certificate_S$ is signed with a digital signature of the

certification authority and contains an ID_S and public key_S of the slave device 200.

[58] In operation S22, the master device 100 receives the $certificate_S$ and encrypted $random\ number_S$ and authenticates the slave device 200 by verifying the $certificate_S$, and decrypts the encrypted $random\ number_S$ using its own private key_M. Here, the master device 100 obtains the public key_S of the slave device 200 using the $certificate_S$ of the slave device 200. In addition, verification of the $certificate_S$ may also be performed on the slave device 200 using CRL.

[59] If the slave device 200 is verified as an authenticated device using the $certificate_S$ of the slave device 200, in operation S24, the master device 100 generates a $random\ number_M$. In operation S26, the generated $random\ number_M$ is encrypted using the public key_S of the slave device 200.

[60] Thereafter, the master device 100 requests the slave device 200 for an authentication end procedure in operation S30. When requesting for the authentication end procedure, the master device 100 sends the encrypted $random\ number_M$ to the slave device 200.

[61] In operation S32, the slave device 200 receives the encrypted $random\ number_M$ and decrypts the $random\ number_M$ using its private key_S.

[62] Accordingly, the master device 100 and the slave device 200 share each other's random numbers, that is, $random\ number_M$ and $random\ number_S$.

[63] As a result, the master device 100 and the slave device 200 sharing each other's random numbers generate their session keys in operations S40 and S42. Here, in order for the master device 100 and the slave device 200 to generate their session keys, the same algorithm may be used. Therefore, the master device 100 and the slave device 200 share the same session key.

[64] After authentication is completed, encrypting and decrypting the data to be transmitted between the master device 100 and the slave device 200 using their session keys can further provide for ensured security in data transmission. In several exemplary embodiments that are described below, unless otherwise noted, it is to be understood that the master device 100 and the slave device 200 encrypt and decrypt the data to be transmitted to each other using each session key generated by the authentication.

[65] After completing the authentication procedure, the master device 100 may obtain one or more ROs having an inheritance structure from the slave device 200. The ROs having an inheritance structure may comprise a parent RO and a child RO. The master device 100 cannot play back a content object using only one of a parent RO and a child RO. In other words, the master device 100 needs both the parent RO and a child RO to play back a content object. In detail, the master device 100 can play back a content object identified by a content object identifier included in a child RO by using a

content encryption key included in the child RO while referencing permission information and constraint information included in a parent RO. Therefore, the master device 100 may obtain one of a parent RO and a child RO or both of the parent RO and the child RO from the slave device 200 depending on whether the slave device 200 includes only one or both of the parent RO and the child RO. The obtaining of a parent RO and a child RO from the slave device 200 by the master device 100 will now be described in detail with reference to FIGS. 4 and 5.

[66] FIG. 4 is a flowchart illustrating a method of obtaining a child RO according to an exemplary embodiment of the present invention. Referring to FIG. 4, in operation S410, a master device 100 which wishes to play back a predetermined content object sends a request for a child RO to a slave device 200. In operation S420, the slave device transmits the requested child RO to the master device 100. In operation S420, the slave device may allow the master device 100 to access the requested child RO without directly transmitting the requested child RO to the master device 100. The master device 100 may transmit an identifier of the predetermined content object to the slave device 200 when sending the child RO request to the slave device 200. Then, the slave device 200 detects for, i.e., searches for, the requested child RO using the identifier of the predetermined content object.

[67] FIG. 5 is a flowchart illustrating a method of obtaining a parent RO according to an exemplary embodiment of the present invention. Referring to FIG. 5, in operation S510, a master device 100 which wishes to play back a predetermined content object sends a request for a parent RO to a slave device 200. In operation S520, the slave device transmits the requested parent RO to the master device 100. In operation S520, the slave device may allow the master device 100 to access the requested parent RO without directly transmitting the requested parent RO to the master device 100. The master device 100 may transmit an inheritance relationship identifier specifying an inheritance relationship in which the requested RO is involved to the slave device 200 when sending the parent RO request to the slave device 200. Then, the slave device 200 searches for the requested parent RO using the inheritance relationship identifier.

[68] As illustrated in FIGS. 4 and 5, the master device 100 may issue queries to the slave device 200 to obtain a child RO and a parent RO.

[69] As described above, the master device 100 may store one or both of a parent RO and a child RO needed for playing back a content object, in which case, the master device 100 performs one or both of the method illustrated in FIG. 4 and the method illustrated in FIG. 5.

[70] A parent RO according to an exemplary embodiment of the present invention and a child RO according to an exemplary embodiment of the present invention are illustrated in FIGS. 6 and 7, respectively.

[71] FIG. 6 is a diagram of a parent RO according to an exemplary embodiment of the present invention which is expressed in a mark-up language. Referring to FIG. 6, the parent RO comprises permission information 620 and constraint information 630. A Play field 622 is set in the permission information 620, and the constraint information 630 comprises Datetime constraint information 632.

[72] FIG. 7 is a diagram of a child RO according to an exemplary embodiment of the present invention which is expressed in a mark-up language. Referring to FIG. 7, the child RO comprises a content object identifier 730 and a content encryption key 720.

[73] The parent RO illustrated in FIG. 6 comprises an inheritance relationship identifier 610 (SubscriptionGUID), and the child RO illustrated in FIG. 7 comprises an inheritance relationship identifier 710 (SubscriptionGUID). Since the inheritance relationship identifier 610 and the inheritance relationship identifier 710 are identical, it is determined that the parent RO illustrated in FIG. 6 and the child RO illustrated in FIG. 7 have an inheritance relationship with each other. If the master device 100 obtains both the parent RO illustrated in FIG. 6 and the child RO illustrated in FIG. 7 from the slave device 200, the master device 100 can play back a content object identified by the content object identifier 730 included in the child RO using the content encryption key 720 included in the child RO while referencing the permission information 620 and the constraint information 630 included in the parent RO.

[74] The structures of the master device 100 and the slave device 200 will now be described in detail with reference to FIGS. 8 through 11.

[75] FIG. 8 is a block diagram of a master device 100 according to an exemplary embodiment of the present invention. Referring to FIG. 8, the master device 100 comprises a device interface module 110, an encryption/decryption module 120, a storage module 130, a detection module 140, a message generation module 140, a playback module 160, a user input module 170, and a control module 180.

[76] The device interface module 110 transmits data to or receives data from a slave device 200. The master device 100 may be connected to the slave device 200 via the device interface module 110.

[77] The encryption/decryption module 120 performs encryption/decryption on data. In detail, the encryption/decryption module 120 encrypts data to be transmitted to the slave device 200 or decrypts data received from the slave device 200 upon the request of the control module 180. The encryption/decryption module 120 may be able to perform not only a public key-based encryption method but also a private key-based encryption method. For this, one or more encryption/decryption modules 120 may be included in the master device 100.

[78] The encryption/decryption module 120 may generate random numbers needed for performing authentication between the master device 100 and the slave device 200.

[79] The storage module 130 stores content objects, parent ROs, child ROs, a CRL, and a certificate of the master device 100. For this, the storage module 130 may comprise a storage medium such as a hard disc or a flash memory.

[80] The detection module 140 searches the storage module 130 for a child RO and a parent RO needed for playing back a content object.

[81] The message generation module 150 generates a child RO request message and a parent RO request message. The child RO request message may comprise a content object identifier, and the parent RO request message may comprise an inheritance relationship identifier. The message generation module 150 may generate the child RO request message and the parent RO request message when the detection module 140 fails to discover a child RO and a parent RO needed for playing back a content object from the storage unit 130.

[82] The playback module 160 plays back a content object using a child RO and a parent RO. For example, the playback module 160 may comprise a moving picture decoding module, in which case, the playback module 160 can play back a content object comprising MPEG-compressed moving picture data.

[83] The user input module 170 receives a command or request from a user. For this, the user input module 170 may comprise an input unit such as a keypad, a touch pad, or a touch screen.

[84] The control module 180 controls the device interface module 110, the encryption/decryption module 120, the storage module 130, the detection module 140, the message generation module 150, the playback module 160, and the user input module 170. Therefore, the control module 180 may serve as a DRM agent that generally manages a DRM operation performed by the master device 100. In addition, as described above with reference to FIG. 3, the control module 180 may control authentication between the master device 100 and the slave device 200 when the master device 100 is connected to the slave device 200.

[85] FIG. 9 is a block diagram of a slave device 200 according to an exemplary embodiment of the present invention. Referring to FIG. 9, the slave device 200 comprises a device interface module 210, an encryption/decryption module 220, a storage module 230, a detection module 240, and a control module 250.

[86] The device interface module 210 transmits data to or receives data from a master device 100. The slave device 200 can be connected to the master device 100 via the device interface module 210.

[87] The encryption/decryption module 220 performs encryption/decryption on data. In detail, the encryption/decryption module 220 encrypts data to be transmitted to the master device 100 or decrypts data received from the master device 100 upon the request of the control module 250. The encryption/decryption module 220 may be able

to perform not only a public key-based encryption method but also a private key-based encryption method. For this, one or more encryption/decryption modules 220 may be included in the slave device 200.

[88] The encryption/decryption module 220 may generate random numbers needed for performing authentication between the master device 100 and the slave device 200.

[89] The storage module 230 stores content objects, parent ROs, child ROs, a CRL, and a certificate of the slave device 200. For this, the storage module 230 may comprise a storage medium such as a hard disc or a flash memory. The parent ROs and the child ROs stored in the storage module 230 may be moved from or copied from the master device 100 or may be those which were provided to the slave device 200 when the slave device 200 was manufactured.

[90] The detection module 240 searches the storage module 230 for a child RO or a parent RO when a child RO request message or a parent RO request message is received from the master device 100.

[91] The control module 250 controls the device interface module 210, the encryption/ decryption module 220, the storage module 230, and the detection module 240. Therefore, the control module 250 may serve as a DRM agent that generally manages a DRM operation performed by the slave device 200. In addition, as described above with reference to FIG. 3, the control module 250 may control authentication between the master device 100 and the slave device 200 when the slave device 200 is connected to the master device 100.

[92] In FIGS. 8 and 9, the term 'module' means, but is not limited to, a software or hardware component, such as a Field Programmable Gate Array (FPGA) or Application Specific Integrated Circuit (ASIC), which performs certain tasks. A module may advantageously be configured to reside on the addressable storage medium and configured to execute on one or more processors. Thus, a module may include, by way of example, components, such as software components, object-oriented software components, class components and task components, processes, functions, attributes, procedures, subroutines, segments of program code, drivers, firmware, microcode, circuitry, data, databases, data structures, tables, arrays, and variables. The functionality provided for in the components and modules may be combined into fewer components and modules or further separated into additional components and modules. In addition, the components and modules may be implemented such that they execute one or more CPUs in a communication system.

[93] The operations of the master device 100 and the slave device 200 will now be described in further detail with reference to FIGS. 10 and 11.

[94] FIG. 10 is a flowchart illustrating a method of consuming ROs according to an exemplary embodiment of the present invention. Referring to FIG. 10, the method is

performed by a master device 100. In operation S810, the master device 100 receives a request for a content object from a user via a user input module 170. In operation S815, a detection module 140 searches a storage module 130 for a predetermined child RO containing a content encryption key needed for playing back the requested content object. In detail, the detection module 140 may obtain a content object identifier from the requested content object and searches for the predetermined child RO with reference to the content object identifier. Child ROs stored in the storage module 130, like the child RO illustrated in FIG. 7, comprise a content object identifier 730. Therefore, the detection module 140 examines the content object identifiers of the child ROs stored therein and searches for a child RO whose content object identifier is identical to the content object identifier obtained from the requested content object.

[95] In operation S820, if the predetermined child RO is not detected from, i.e., found in, the storage module 130, the master device 100 sends a request for the predetermined child RO to a slave device 200. For this, in operation S825, a message generation module 150 generates a child RO request message, and in operation S830, the device interface module 110 transmits the child RO request message to the slave device 200. The child RO request message may comprise a content object identifier.

[96] In operation S835, the device interface module 110 receives the predetermined child RO from the slave device 200, and then, a process of obtaining a parent RO is performed. However, if no child RO is received from the slave device 200 (for example, if the predetermined child RO is not received within a predetermined amount of time or if a message indicating that the predetermined child RO has not yet been detected from the slave device 200 is received), the method is terminated because the master device 100 cannot play back the requested content object.

[97] In operation S840, if the predetermined child RO is detected from the storage module 130 or is received from the slave device 200, the detection module 140 searches the storage module 130 for a parent RO that has an inheritance relationship with the predetermined child RO. For this, the detection module 140 may obtain an inheritance relationship identifier from the predetermined child RO and searches the storage module 130 for a parent RO whose inheritance relationship identifier is identical to the obtained inheritance relationship identifier.

[98] If the parent RO having an inheritance relationship with the predetermined child RO is not detected from the storage module 130 in operation S845, a process of sending a request for the parent RO having an inheritance relationship with the predetermined child RO to the slave device 200 is performed. For this, in operation S850, the message generation module 150 generates a parent RO request message, and in operation S855, the device interface module 110 transmits the parent RO request message to the slave device 200. The parent RO request message may comprise an in-

heritance relationship identifier.

[99] Thereafter, if no parent RO is received from the slave device 200 (for example, if the parent RO having an inheritance relationship with the predetermined child RO is not received within a predetermined amount of time or if a message indicating that the parent RO having an inheritance relationship with the predetermined child RO has not yet been detected from the slave device 200 is received), the method is terminated because the master device 100 cannot play back the requested content object.

[100] In operation S865, if the predetermined child RO is detected from the storage module 130 or is received from the slave device 200, a playback module 160 plays back the requested content object using the predetermined child RO and the parent RO having an inheritance relationship with the predetermined child RO.

[101] As described above with reference to FIG. 10, the master device 100 may issue a request for a child RO and a request for a parent RO having an inheritance relationship with the child RO separately to the slave device 200.

[102] FIG. 11 is a flowchart illustrating the providing of ROs to a master device 100 by a slave device 200 according to an exemplary embodiment of the present invention.

[103] Referring to FIG. 11, in operation S910, a device interface module 210 receives a request for a child RO from the master device 100. In operation S920, a detection module 240 searches a storage module 230 for a child RO requested by the master device 100. In detail, the detection module 240 may obtain a content object identifier from the child RO request message and searches the storage module 230 for a child RO whose content object identifier is identical to the obtained content object identifier.

[104] In operation S940, if the requested child RO is detected from the storage module 230 in operation S930, the device interface module 210 transmits the detected child RO to the master device 100.

[105] In operation S950, the device interface module 210 receives a parent RO request message from the mater device 100. In operation S960, the detection module 240 searches the storage module 230 for a parent RO requested by the master device 100. In detail, the detection module 24 may obtain an inheritance relationship identifier from the parent RO request message and searches the storage module 230 for a parent RO whose inheritance relationship identifier is identical to the obtained inheritance relationship identifier.

[106] If the requested parent RO is detected from the storage module 230 in operation S970, the device interface module 210 transmits the detected parent RO to the master device 100 in operation S980.

[107] Operations S940 and S950 have been described above as being operations in which the slave device 200 actively transmits the requested child RO and the requested parent RO to the master device 100, but the present invention is not restricted to it. For

example, a control module 250 of the slave device 200 may allow the master device 100 to access the requested child RO and the requested parent RO.

[108] As described above, all the queries for obtaining a content object identifier from a content object to detect a child RO, for obtaining an inheritance relationship identifier from the child RO to search for a parent RO, and for obtaining the child RO and the parent RO from the slave device 200 are sent by the master device 100. Therefore, in a case where the master device 100 plays back a content object using a plurality of ROs that are stored in the slave device 200 and have an inheritance structure, the slave device 200 can detect the ROs by performing a minimum number of processes. Therefore, it is possible to considerably reduce the operation load of the slave device 200.

Industrial Applicability

[109] As described above, the method and the device for consuming a plurality of ROs having an inheritance structure in an environment where the ROs are distributed over a plurality of devices have the following aspects.

[110] First, even in an environment where a content object and a parent RO and a child RO needed for playing back the content object are stored in different devices, it is possible to play back the content object using an inheritance relationship between the parent RO and the child RO.

[111] Second, even when a plurality of ROs having an inheritance structure are stored in a slave device which has poorer operation capabilities than a master device, it is possible to facilitate the consumption of the ROs using the operation capabilities of the master device.

[112] While the present invention has been particularly shown and described with reference to exemplary embodiments thereof, it will be understood by those of ordinary skill in the art that various changes in form and details may be made therein without departing from the spirit and scope of the present invention as defined by the following claims. Therefore, it is to be understood that the above-described exemplary embodiments have been provided only in a descriptive sense and will not be construed as placing any limitation on the scope of the invention.

Claims

[1] 1. A master device comprising:

a storage module which stores a plurality of rights objects (ROs) having a plurality of inheritance structures;

a detection module which searches the storage module for an RO corresponding to a predetermined content object;

a message generation module which generates an RO request message if the detection module fails to discover the RO corresponding to the predetermined content object from the storage module;

a device interface module which transmits the RO request message to a slave device and receives the RO corresponding to the predetermined content object from the slave device; and

a playback module which plays back the predetermined content object using the RO corresponding to the predetermined content object from at least one of the storage module and the slave device.

2. The master device of claim 1, wherein the plurality of ROs comprises at least one of a child RO and a parent RO, the child RO having a first inheritance relationship with the parent RO and the parent RO having a second inheritance relationship with the child RO,

wherein the child RO comprises at least one of an identifier of the predetermined content object, a content encryption key corresponding to the predetermined content object, and constraint information regarding use restriction of the child RO, and

wherein the parent RO comprises at least one of the constraint information and permission information regarding playback methods for playing back the predetermined content, the playback methods being obtainable from the child RO.

3. The master device of claim 2, wherein the message generation module generates a child RO request message, the device interface module receives a first child RO corresponding to the child RO request message from the slave device, and the playback module plays back the predetermined content object using the first child RO and the parent RO detected from the storage module by the detection module.

4. The master device of claim 3, wherein the child RO request message comprises a first identifier of the predetermined content object, the first child RO is detected from the slave device with reference to the first identifier of the predetermined content object, and the detection module searches for the parent RO using an inheritance relationship identifier included in the first child RO.

5. The master device of claim 2, wherein the message generation module generates a child RO request message and a parent RO request message, the device interface module receives a first child RO corresponding to the child RO request message and a first parent RO corresponding to the parent RO request message from the slave device, and the playback module plays back the predetermined content using the first child RO and the first parent RO.

6. The master device of claim 5, wherein the child RO request message comprises a first identifier of the predetermined content object, the parent RO request message comprises an inheritance relationship identifier included in the first child RO, the first child RO is detected from the slave device with reference to the first identifier of the predetermined content object, and the first parent RO is detected from the slave device using the inheritance relationship identifier.

7. The master device of claim 2, wherein the message generation module generates a parent RO request message, the device interface module receives a first parent RO corresponding to the parent RO request message from the slave device and the playback module plays back the predetermined content object using the first parent RO.

8. The master device of claim 7, wherein the parent RO request message comprises the inheritance relationship identifier included in the child RO detected from the storage module by the detection module, and the first parent RO is detected from the slave device using the inheritance relationship identifier included in the child RO.

9. The master device of claim 1, wherein the slave device is a portable storage device.

10. A method of consuming a plurality of Rights Objects (ROs), the method comprising:

obtaining a plurality of ROs having predetermined inheritance structures from a slave device; and

playing back a predetermined content object using at least one of the plurality of ROs.

11. The method of claim 10, wherein the plurality of ROs comprises at least one of a child RO and a parent RO, the child RO having a first inheritance relationship with the parent RO and the parent RO having a second inheritance relationship with the child RO,

wherein the child RO comprises at least one of an identifier of the predetermined content object, a content encryption key corresponding to the predetermined content object, and constraint information regarding use restriction of the child RO, and

wherein the parent RO comprises at least one of the constraint information and permission information regarding playback methods for playing back the predetermined content, the playback methods being obtainable from the child RO.

12. The method of claim 11, wherein the obtaining comprises:
sending a request for the child RO to the slave device; and
receiving the child RO from the slave device, and
wherein the playing back comprises playing back the predetermined content object using the child RO.

13. The method of claim 12, wherein the sending the request for the child RO comprises transmitting an identifier of the predetermined content object to the slave device, and the child RO is detected from the slave device with reference to the identifier of the predetermined content object.

14. The method of claim 12, further comprising:
sending a request for the parent RO to the slave device; and
receiving the parent RO from the slave device.

15. The method of claim 14, wherein the sending the request for the parent RO comprises:
obtaining an inheritance relationship identifier from the child RO, the inheritance relationship identifier identifying the parent RO; and
transmitting the inheritance relationship identifier to the slave device, the parent RO being detected from the slave device using the inheritance relationship identifier.

16. The method of claim 11, further comprising:
sending a request for the parent RO to the slave device; and
receiving the parent RO from the slave device,
wherein the playing back comprises playing back the predetermined content object using the child RO and the parent RO.

17. The method of claim 16, wherein the sending the request for the parent RO comprises:
obtaining an inheritance relationship identifier identifying the parent RO from the child RO; and
transmitting the inheritance relationship identifier to the slave device, and
the parent RO being detected from the slave device using the inheritance relationship identifier.

18. The method of claim 10, wherein the slave device is a portable storage device.