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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 April 2004 (29.04.2004)

PCT

(10) International Publication Number
WO 2004/036449 A1

(51) International Patent Classification[7]: G06F 17/00

(21) International Application Number:
PCT/KR2003/000713

(22) International Filing Date: 9 April 2003 (09.04.2003)

(25) Filing Language: Korean

(26) Publication Language: English

(30) Priority Data:

60/418,160	15 October 2002 (15.10.2002)	US
60/425,259	12 November 2002 (12.11.2002)	US
10-2003-0013002	3 March 2003 (03.03.2003)	KR

(71) Applicant: SAMSUNG ELECTRONICS CO., LTD. [KR/KR]; 416, Maetan-dong, Yeongtong-gu, Suwon-si, 442-742 Gyeonggi-do (KR).

(72) Inventor: CHOI, Yang-Lim; 210-1509 Hanshin Apt., 124, Imae-dong, Bundang-gu, Seongnam-si, 463-060 Gyeonggi-do (KR).

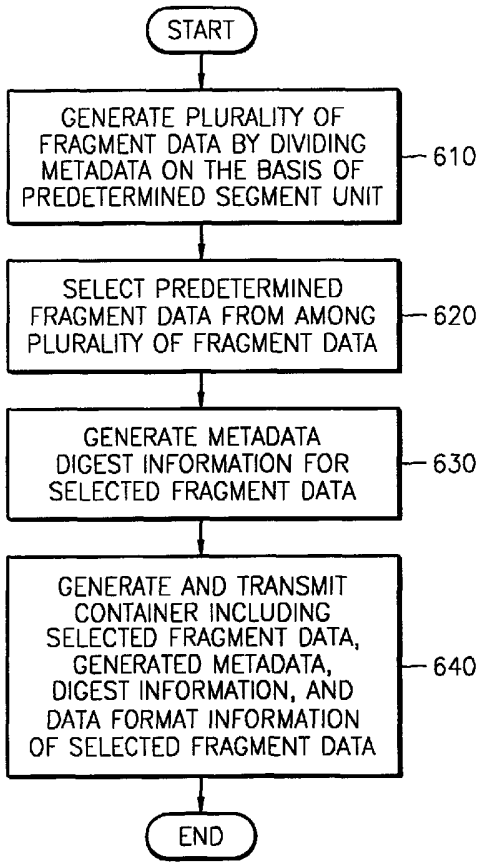
(74) Agent: LEE, Young-Pil; The Cheonghwa Building, 1571-18, Seocho-dong, Seocho-gu, Seoul 137-874 (KR).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE,

[Continued on next page]

(54) Title: METHOD FOR MANAGING METADATA



(57) Abstract: A method for managing metadata in a metadata transmission server is provided. The method involves generating a plurality of fragment data by partitioning metadata to be transmitted on the basis of a predetermined segment unit, selecting predetermined fragment data among the plurality of fragment data, generating metadata-related information using the selected fragment data, and transmitting the selected fragment data and the metadata-related information with data format information indicating the type of the selected fragment data.

WO 2004/036449 A1

ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

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

Published:

— *with international search report*

METHOD FOR MANAGING METADATA

Technical Field

The present invention relates to a method for managing metadata in a transmission server and a client that receives the metadata, and more particularly, to a method for managing metadata including authentication of a message source, and message integrity and confidentiality until a client receives the metadata.

Background Art

In a multimedia system, such as a broadcasting system where data is transmitted from a server to a client or a video-on-demand service system where data is transmitted through interactions between the server and the client, a service provider provides multimedia content and its related metadata to a client.

The metadata transmitted to the client may be used for various purposes. For example, the metadata can be used by the client to select multimedia content to be reproduced, recorded, or transmitted.

In recent years, the amount and complexity of data that can be contained in metadata used by a client of a broadcasting system have increased. Thus, there has been an increasing demand for security of such metadata. In particular, in a case where metadata is generated and then transmitted to a client from a transmission server, it is very important to authenticate a source of the metadata and verify whether or not the integrity and confidentiality of the metadata have been affected during the transmission process. However, a metadata management method that enables effective metadata authentication has not yet been proposed.

Disclosure of the Invention

The present invention provides a method for managing metadata to be transmitted in a metadata transmission server so that authentication of the metadata to be transmitted can be effectively performed.

The present invention also provides a method for managing in a client metadata received from a transmission server so that authentication of the received metadata can be effectively performed.

According to an aspect of the present invention, there is provided a method for managing metadata in a metadata transmission server. The method involves (a) generating a plurality of fragment data by partitioning metadata to be transmitted on the basis of a predetermined segment unit, (b) selecting predetermined fragment data among the plurality of fragment data, (c) generating metadata-related information using the selected fragment data, and (d) transmitting the selected fragment data and the metadata-related information with data format information indicating the type of the selected fragment data.

According to another aspect of the present invention, there is provided a method for managing metadata in a client that receives the metadata. The method involves (a) reading predetermined fragment data and its corresponding metadata-related information and data format information indicating the type of the predetermined fragment data from the received metadata, (b) generating metadata-related information using the predetermined fragment data and its corresponding data format information, and (c) determining whether or not the received metadata has been authenticated by comparing the metadata-related information generated in step (b) with the metadata-related information read in step (a).

According to still another aspect of the present invention, there is provided a method for managing metadata in a client that receives the metadata. The method involves (a) receiving fragment data of the received metadata, metadata-related information, data format

information indicating the type of the fragment data, metadata authentication information, and an encrypted first encryption key, (b) generating metadata-related information using the fragment data of the received metadata and its corresponding data format information, (c) decrypting the encrypted first encryption key using a second encryption key stored in the client, (d) generating metadata authentication signature information using the generated metadata-related information and the decrypted first encryption key, and (e) determining whether or not the received metadata has been authenticated by comparing the generated metadata authentication signature information with the received metadata authentication signature information.

The present invention relates to a method for managing metadata in a transmission server and a client device, which is capable of identifying whether or not metadata has been damaged during being transmitted from the transmission server to the client device and effectively verifying which service provider or metadata content provider has transmitted the corresponding metadata to the client device.

Brief Description of the Drawings

FIG. 1 is a block diagram illustrating metadata authentication levels;

FIG. 2 is a diagram illustrating a method of transmitting data using different transmission units;

FIG. 3 is a diagram illustrating the format of a metadata container used for metadata container-level authentication in a unidirectional channel;

FIG. 4 is a diagram illustrating a SOAP message used for metadata container-level authentication in a bi-directional channel;

FIG. 5 is a block diagram illustrating a metadata classification method using index information of metadata;

FIG. 6 is a flowchart of a method for managing metadata in a

metadata transmission server according to a preferred embodiment of the present invention;

FIG. 7 is a flowchart of a method for managing metadata in a metadata client according to a preferred embodiment of the present invention;

FIG. 8 is a flowchart of a method for managing metadata in a metadata transmission server according to another preferred embodiment of the present invention;

FIG. 9 is a flowchart of a method for managing metadata in a metadata client according to another preferred embodiment of the present invention;

FIG. 10 is a diagram illustrating the format of a data container in a unidirectional channel; and

FIG. 11 is a diagram illustrating a SOAP message in a bi-directional channel.

Best mode for carrying out the Invention

When metadata is received, it is necessary to authenticate the received metadata. Metadata authentication may be performed at a transmission level or a source level.

In particular, transmission-level metadata authentication includes authentication of a message source, and message integrity and confidentiality. Here, the message source is not a source from which a message, i.e., metadata content, is generated but a source from which the message is transmitted.

For example, in a case where a metadata content provider 120 and a service provider 140, such as SK Telecom Corp., are separately provided as shown in FIG. 1, it can be verified through transmission-level authentication of a message source whether metadata A received by a client 160 has been transmitted from the service provider 140.

In addition, transmission-level authentication of message integrity

verifies whether or not the metadata A has been changed in the process of transmitting the metadata A from the service provider 140 to the client 160.

Transmission-level authentication of message confidentiality verifies whether or not the metadata A has not yet been disclosed to a third party during the transmission process. These three transmission-level authentication processes are performed using an SSL/TLS algorithm in a TCP/IP protocol, a DTCP algorithm in an IEEE 1394 protocol, and an HDCP algorithm in a DVI protocol.

Like the transmission-level authentication, source-level metadata authentication also includes authentication of a message source, and message integrity and confidentiality.

In particular, source-level authentication of a message source verifies a source from which a message, i.e., metadata content, is generated. For example, as shown in FIG. 1, source-level authentication of a message source of the metadata A shows that the metadata A received by the client 160 has been transmitted from the metadata content provider 120.

Source-level authentication of message integrity verifies whether or not the metadata A has been changed in the process of transmitting the metadata A from the metadata content provider 120 to the client 160.

Source-level authentication of message confidentiality verifies whether or not the metadata A has not yet been disclosed to a third party during the transmission of the metadata A between the metadata content provider 120 and the client 160.

When such source-level metadata authentication is performed, transmission-level metadata authentication may not need to be performed.

FIGS. 2(a) through 2(c) show a method of transmitting data in different transmission units in a physical layer.

More specifically, FIG. 2(a) illustrates transmission packets that

are subject to transmission-level metadata authentication. Transmission-level metadata authentication is performed on each transmission packet shown in FIG. 2(a). Each transmission packet has a binary XML format.

FIG. 2(c) illustrates metadata subject to source-level metadata authentication. The metadata shown in FIG. 2(c) has a text XML format.

FIG. 2(b) illustrates metadata containers subject to metadata container-level authentication. Each predetermined semantic unit of metadata is contained in a metadata container. Examples of such metadata container are shown in FIGS. 3 and 4.

FIG. 3 is a diagram illustrating the format of a metadata container subject to metadata container-level authentication in a unidirectional channel. As shown in FIG. 3, a metadata container includes a header, fragment data, and metadata authentication information, and the header contains control information used for metadata container-level authentication.

The control information includes first control information F_1, second control information F_2, third control information F_3, fourth control information F_4, and fifth control information F_5. The control information ranging from the first control information F_1 through the fifth control information is comprised of a signal or a flag.

The first control information F_1 is an authentication flag indicating whether or not metadata container-level authentication has been performed on the fragment data. Here, the metadata container-level authentication may be performed using a media authentication code (MAC) or digital signature algorithm (DSA).

The second control information F_2 is information on a specific algorithm used for generating metadata container-level authentication information. The second control information F_2 may be represented by a set of binary codes. The relationship between the specific

algorithm and the binary codes is defined in advance and is rendered to a server providing services and a client receiving metadata containers.

The third control information F_3 is data format information showing in detail the way to apply the specific algorithm to the fragment data. The fragment may have a binary XML format or a text XML format, and thus the method of applying the specific algorithm to the fragment data varies depending on the format of the fragment data.

Metadata authentication information in the present invention is comprised of values obtained by substituting metadata into a hash function, i.e., hash values. Therefore, authentication information of fragment data having a text XML format has nothing to do with authentication information of fragment data having a binary XML format, and this is the reason why the third control information F_3 is necessary. In other words, there is a need to identify the format of metadata used to obtain hash values in order to determine whether or not an authentication signature is valid based upon metadata included in a metadata container received by a client and the hash values.

The fourth control information F_4 is encryption key information concerning metadata authentication. The encryption key information is inserted into the metadata container together with metadata and then directly transmitted from a server to a client. Alternatively, the encryption key information may be transmitted from the server to the client via an additional security channel.

The fifth control information F_5 is an authentication level flag indicating a level of metadata authentication that has been performed. For example, when the fifth control information F_5 is set to '0', it indicates that transmission-level metadata authentication has been performed. When the fifth control information F_5 is set to '1', it indicates that source-level metadata authentication has been performed. With the help of the authentication level flag indicating whether or not transmission-level or source-level metadata authentication has been

performed, it is possible to determine, using an application program of a client, how much reliable the metadata transmitted from a server is. Based on the reliability of the received metadata, it can be further determined whether to use the received metadata or not based on the reliability of the received metadata.

The metadata container includes a fragment data storage region where at least one fragment data is contained. A predetermined semantic unit of metadata, for example, fragment data, such as information on a program, is inserted into the metadata container in the present embodiment. However, the metadata container of the present invention may also be used to selectively carry arbitrary units of metadata. In addition, a group of related metadata is transmitted from a service provider to a client, while being carried by a series of metadata containers. Furthermore, one metadata container includes one or more metadata fragments. For example, one of the metadata fragment data may be a sub-tree of an XML tree structure representing the entire metadata.

The metadata container-level authentication information includes metadata digest information and metadata authentication signature information.

The metadata digest information represents a value obtained by substituting one of the fragment data stored in the fragment data storage region into a unidirectional function, such as a hash function specified in the second control information F_2. Each metadata digest information is related to its corresponding fragment data using a predetermined pointer. For example, first metadata digest information is related to the first fragment data using the predetermined pointer. In the present embodiment, a hash function has been used to generate the metadata digest information. Sometimes, however, other functions having the same characteristics as a unidirectional function, such as a hash function, can also be used to obtain the metadata digest information.

The metadata authentication signature information is a value obtained by substituting the metadata digest information and an encryption key K into a unidirectional function, for example, the hash function specified in the second control information F_2. Each metadata authentication signature information, like each metadata digest information, is related to its corresponding fragment data using a predetermined pointer. For example, first metadata authentication signature information is related to the first fragment data using the predetermined pointer. In the present embodiment, a hash function has been used to generate the metadata authentication signature information. Sometimes, however, other functions having the same characteristics as a unidirectional function, such as a hash function, can also be used to obtain the metadata digest information.

FIG. 4 is a diagram illustrating the format of an SOAP envelope used for metadata container-level authentication in a bi-directional channel. As shown in FIG. 4, authentication-related information is included in an SOAP header, and metadata fragment data is included in the body of the SOAP envelope.

Among pieces of the authentication-related information contained in the SOAP header, 'Algorithm ID' information, 'SignatureValueBaseType' information, and 'KeyInfo' information correspond to the second control information F_2, the third control information F_3, and the fourth control information F_4, respectively, of FIG. 3. 'Digest' information and 'SignatureValue' information correspond to the metadata digest information and the metadata authentication signature information, respectively, described above with reference to FIG. 3. 'AuthenticationLevel' information specifies a level of metadata authentication and corresponds to an authentication level flag, i.e., the fifth control information F_5 of FIG. 3.

As shown in FIGS. 3 and 4, it is possible to effectively perform encryption management and metadata management by inserting

fragment data obtained by partitioning the metadata on the basis of a predetermined semantic unit into a metadata container.

For example, by allotting indexing information to each fragment data and using an index list stored in an index list storing unit, it is possible to store only predetermined metadata selected from among all metadata input into a cache 520 in a data storage 540 of FIG. 5. In addition, since metadata is partitioned into predetermined semantic units, such as program information, segment information, and so on, as shown in FIG. 4, it is possible to selectively encrypt the metadata fragment data on a predetermined semantic unit-by-predetermined semantic unit basis.

FIG. 6 is a flowchart of a metadata container-level authentication method using the metadata container shown in FIGS. 3 and 4. More specifically, FIG. 6 is a flowchart of the operation of the metadata content provider 120 or the service provider 140 of FIG. 1.

Referring to FIG. 6, in step 610, a plurality of fragment data are generated by dividing metadata on the basis of a predetermined semantic unit. Each fragment data generated in the present embodiment is a predetermined semantic unit of metadata that has a predetermined meaning, like program information.

In step 620, predetermined fragment data is selected from among the plurality of fragment data generated in step 610.

In step 630, metadata digest information is generated by substituting the selected fragment data into a hash function, for example, a secured hash algorithm, such as SHA-1. In the present embodiment, a hash function is used to generate message digest information. Sometimes, however, other functions having the same characteristics as a unidirectional function, such as a hash function, can also be used.

In step 640, a metadata container, including the selected fragment data, the generated metadata digest information, and data format information indicating whether the format of the selected fragment data is binary XML or text XML, is generated and then transmitted to a client.

Here, it is necessary to specify the format of the selected fragment data using the data format information because two different types of fragment data can bring about two different types of metadata digest information in step 620 even though the two fragment data are basically the same.

Examples of the metadata container generated in step 640 are shown in FIGS. 3 and 4. In step 640, a predetermined authentication flag is set so as to indicate that metadata container-level authentication has been performed on fragment data of metadata carried by the metadata container.

Algorithm information that has been used to generate the metadata digest information may be inserted into the metadata container. For example, in a case where the metadata digest information is generated in step 630 using a hash function, algorithm information indicating that the hash function has been used as an authentication information generation algorithm is inserted into the metadata container. However, in a case where the algorithm information is already well known to both a server and a client, there is no need to insert such algorithm information into the metadata container.

Furthermore, it is also possible to insert a flag specifying a metadata authentication level into the metadata container together with the data formation information of the selected fragment data. The flag specifies whether metadata authentication using the metadata container has been performed at a transmission level or at a source level.

In a case where a plurality of fragment data are inserted into the metadata container, metadata digest information corresponding to each of the plurality of fragment data is contained in the metadata container, and so is pointer information indicating a relationship between each of the plurality of fragment data and its corresponding metadata digest information.

In addition, in a case where a plurality of fragment data are

inserted into the metadata container, indexing information for each of the plurality of fragment data is also contained in the metadata container.

FIG. 7 is a flowchart of a metadata container-level authentication method using the metadata container shown in FIGS. 3 and 4. More specifically, FIG. 7 is the flowchart of the operation of the client 160 of FIG. 1. Referring to FIG. 7, in step 710, a metadata container is received from the metadata content provider 120 or the service provider 140.

In step 720, first control information F_1, i.e., an authentication flag, of a header of the received metadata container is read.

In step 730, if the result of reading the authentication flag shows that metadata container-level authentication has been performed on fragment data contained in the metadata container, the method moves on to step 740. Otherwise, the method moves on to step 742.

In step 740, an algorithm used for generating metadata digest information included in the metadata container is read by identifying second control information F_2, i.e., an algorithm used for generating authentication information. In the present embodiment, the algorithm used for generating authentication information is a hash function. In a case where the algorithm used for generating authentication information is determined in advance and known to both the metadata content provider 120 (or the service provider 140) and the client 160, the process of reading the algorithm used for generating authentication information can be omitted.

In step 740, the format of fragment data, used in computing metadata digest information included in the metadata container, is identified by recognizing third control information F_3, i.e., metadata format information.

In step 742, such metadata container-level authentication is completed.

In step 750, predetermined fragment data of metadata and its

corresponding metadata digest information are read.

In step 760, metadata digest information is generated based on the fragment data and the data format information read in step 740 by using the algorithm used for generating metadata digest information, for example, a hash function.

In step 770, it is determined whether or not the metadata container-level authentication has been performed on metadata transmitted from the metadata content provider 120 or the service provider 140 by comparing the metadata digest information generated in step 760 with the metadata digest information of the predetermined fragment data read in step 750.

A metadata authentication level flag may be further included in the metadata container transmitted from the metadata content provider 120 or the service provider 140. In this case, it is possible to figure out whether the metadata container-level authentication is a transmission-level metadata authentication or a source-level metadata authentication by reading the metadata authentication level flag using an application program of the client 160. In addition, it is also possible to determine whether to use the metadata transmitted from the metadata content provider 120 or the service provider 140 or not based upon the reliability of the metadata.

FIG. 8 is a flowchart of a metadata container-level authentication method using the metadata container shown in FIGS. 3 and 4. More specifically, FIG. 8 is the flowchart of the operation of the metadata content provider 120 or the service provider 140 shown in FIG. 1.

Referring to FIG. 8, in step 810, a plurality of fragment data are generated by partitioning metadata on the basis of a predetermined semantic unit. Each fragment data generated in the present embodiment is a predetermined semantic unit, such as program information.

In step 820, predetermined fragment data among the plurality of

fragment data is selected.

In step 830, metadata digest information is generated by substituting the selected fragment data into a hash function. In the present embodiment, a hash function is used to generate the metadata digest information. However, other functions having the same characteristics as a unidirectional function, such as a hash function, can also be used.

In step 840, a metadata authentication signature is generated by substituting the metadata digest information generated in step 830 and an encryption key K into the hash function. The encryption key K is specific to the service provider 140. In the present embodiment, a hash function is used to generate the metadata digest information. However, other functions having the same characteristics as a unidirectional function, such as a hash function, can also be used. The encryption key K used to generate the metadata authentication signature is encrypted using another encryption key L. Hereinafter, an encrypted encryption key value obtained using the encryption key L will be represented by E(K). The encrypted encryption key value E(K) is transmitted to the client 160, being carried by a metadata container. Alternatively, the encrypted encryption key value E(K) is transmitted to the client 160 via a security channel. The encryption key L is transmitted to the client 160 via another security channel.

In step 850, a metadata container including the metadata digest information, the metadata authentication signature, and data format information of the selected fragment data is generated and then transmitted to the client 160.

Examples of the metadata container generated in step 850 are shown in FIGS. 3 and 4. In step 850, metadata container-level authentication flag is allotted to the generated metadata container so as to indicate that metadata container-level authentication has been performed on fragment data of metadata carried by the metadata

container.

Information on an algorithm used for generating the metadata digest information may be inserted into the metadata container.

In addition, the data format information of the selected fragment data indicates whether the format of the selected fragment data used for generating the metadata digest information and the authentication information is binary XML or text XML.

In a case where a plurality of fragment data are inserted into the metadata container, metadata digest information and metadata authentication signature information for each of the plurality of fragment data are also included in the metadata container. In addition, pointer information indicating a relationship between each of the plurality of fragment data and its corresponding metadata digest information and metadata authentication signature information is further included in the metadata container.

FIG. 9 is a flowchart of a metadata container-level authentication method using the metadata container shown in FIGS. 3 and 4. More specifically, FIG. 9 is a flowchart of the operation of the client 160 of FIG. 1.

Referring to FIG. 9, in step 910, a metadata container is received from the metadata content provider 120 or the service provider 140.

In step 920, first control information included in a header of the metadata container, i.e., an authentication flag, is read.

In step 930, if the result of reading the authentication flag shows that metadata container-level authentication has been performed on fragment data contained in the metadata container, the method moves on to step 940. Otherwise, the method moves on to step 942.

In step 940, an algorithm used for generating metadata digest information included in the metadata container is read by recognizing second control information F_2, i.e., an algorithm used for generating authentication information. In the present embodiment, the algorithm

used for generating authentication information is a hash function. In a case where the algorithm used for generating authentication information is determined in advance and known to both the metadata content provider 120 (or the service provider 140) and the client 160, the process of reading the algorithm used for generating authentication information can be omitted.

In step 940, the format of fragment data, used in computing metadata digest information included in the metadata container, is identified by recognizing third control information F_3, i.e., metadata format information.

In step 942, metadata container-level authentication is completed.

In step 950, predetermined fragment data of metadata contained in the metadata container, and its corresponding metadata digest information, metadata authentication signature information, and data format information are read.

In step 960, metadata digest information is generated based upon the predetermined fragment data and its corresponding data format information read in step 950 by using the algorithm read in step 940, for example, a hash function.

In step 970, an encryption key K that has been encrypted is decrypted using another encryption key L stored in the client 160. The encryption key L has been transmitted from the metadata content provider 120 or the service provider 140 to the client 160.

In step 980, a metadata authentication signature S is generated using the metadata digest information generated in step 960 and the decrypted key K.

In step 990, it is determined whether or not a metadata authentication signature received by the client 160 is verified by comparing the metadata authentication signature S generated in step 980 with the metadata authentication signature information read in step 950.

The metadata container may further include an authentication level flag indicating the level of metadata authentication performed on the metadata container, in which case the metadata authentication level is read using an application of the client 160 and depending on the metadata authentication level, it is determined whether to use metadata contained in the metadata container or not.

In addition, various methods for testing message integrity are available. One of those various methods is cryptography using a public key. According to this method, a service provider possesses a pair of keys (K_s, K_p) and signs a message using the key K_s. Here, K_s indicates a secret key, and K_p indicates a public key. A client can obtain the public key K_p through reliable sources. Therefore, in a case where the client receives a metadata container with the service provider's signature, the client figures out who the service provider that has transmitted the metadata container is and obtains the public key K_p corresponding to the identified service provider. The client verifies whether or not the received signature is valid using the public key K_p.

Hereinafter, requisites for metadata authentication and a metadata authentication method for preserving the security of metadata will be described in greater detail in the following paragraphs.

In order to maintain the security of metadata, it is necessary to authorize metadata access and use, preserve metadata integrity and confidentiality, and effectively protect the binary format or text format of subgroups of the metadata.

Authorization of access to the entire metadata or part of it must be performed according to predetermined authorization rules. This metadata access authorization process is performed on each application or each metadata.

Various operations including 'view', 'modify', and 'copy' are carried out based on accessing of the entire metadata or part of it. 'View' is one of the simplest examples of metadata use and is simply performed

by accessing the metadata. On the other hand, in the case of modifying or copying all or part of the metadata, a metadata file management system is required. In addition, in the case of copying the metadata using a remote application, for example, in the case of transmitting the metadata from a client to a service provider, a request for the metadata and transmission of the requested metadata and its source authentication information are required.

In addition, it is necessary to preserve metadata confidentiality in order to preserve the security of metadata. In some cases, metadata may include highly confidential or private data. For this or other reasons, metadata needs to be encrypted before being transmitted or stored so that it can be prevented from being undesirably exposed to the public. In other words, even during transmitting metadata, the confidentiality of the metadata can be preserved by performing transmission-level encryption on the metadata, i.e., encrypting a transmission unit or container of the metadata. In addition to the transmission-level encryption of the metadata, source-level encryption of the metadata can solve all possible problems concerning the confidentiality of metadata at a transmission level or a storage level.

Hereinafter, the security of metadata in a unidirectional channel environment concerning a conditional access system and a bi-directional channel (TLS) environment will be described in greater detail.

Here, the unidirectional channel environment concerning a conditional access system includes terrestrial broadcasting, such as ATSC or DVB, satellite broadcasting, such as Direct TV, cable TV, and IP-multicasting. In the unidirectional channel environment concerning a conditional access system, a unidirectional channel is used except for a case where data exchanges, such as transactions, are carried out using a return channel. Functions provided in the unidirectional channel environment concerning a conditional access system are as follows.

A receiver and a transmitter with hardware devices automatically

authorize each other. In addition, the receiver and the transmitter are enabled to share a common secret via a predetermined channel. Here, the common secret represents a code shared by the receiver and the transmitter. Packet payload is encrypted and transmitted. Later, the encrypted packet payload is decrypted using the common secret or using a key decrypted with the use of the common secret.

In the bi-directional channel environment, a handshake protocol is used and a server and a client authorize each other by exchanging and authenticating certificates issued by a third certificate authorization organization. A common secret is shared by the client and the server, and a session key is generated later. Packet payload is encrypted using the session key and then transmitted. The encrypted packet payload is decrypted using the session key. Source authentication may be performed using an algorithm, such as DSA or MAC.

In addition, in the bi-directional channel environment, authorization of the client and the server is performed through the authentication and exchange of certificates issued a third certificate authorization organization. The confidentiality of data transmitted between the client and the server to the other is preserved through encryption of packet payload and message authentication.

In order to keep metadata secured during the transmission of the metadata, the common secret needs to be shared by the receiver and the transmitter in a safe manner so that the receiver and the transmitter can authorize each other and data transmitted there between and data can be encrypted and then transmitted.

Hereinafter, a method for protecting metadata at a transmission level or at a source level will be described in greater detail.

As for the protection of metadata during the transmission of the metadata, authorization of a receiver and a transmitter is carried out at a transmission level, and authorization of the metadata and preservation of the confidentiality of the metadata are carried at a broadcasting system

level.

For example, in a unidirectional channel, each SOAP message consisting of a head and a body can be used as a unit of protection, as shown in FIG. 10. On the other hand, in a bi-directional channel, data signature information is transmitted using a SOAP message, in which case the data information is included in the body of the SOAP message, as shown in FIG. 11. Data contained in the body of the SOAP message can be encrypted.

Hereinafter, a method for preserving metadata integrity and confidentiality and controlling metadata access and use in a broadcasting system, which is classified as metadata protection at a source level, will be described.

The preservation of metadata integrity and confidentiality in a broadcasting system is enabled by allotting an authentication signature to metadata and encrypting the metadata. Given that the entire metadata is not always subjected to such an encryption process because of no need to preserve the integrity of the entire metadata, it is necessary to represent specific portions of the metadata that have been encrypted or authenticated with a predetermined pointer, and this process can be performed at a source level where the predetermined pointer can be maintained by using a right management protection (RMP) system. By using a source level signature, a metadata source can be practically authenticated. Of course, the metadata must include such information as a source authentication signature.

In order to control metadata access and usage, a standard description of metadata access and usage rights and implementation thereof are required. A standard description may have an XML schema format or may assume the form of an element of a set of data having a predetermined meaning. Such a standard description may be generated using a conventional markup language, such as XrML, XACML, or SAML. A license description and a usage rule can be

isolated from metadata. In a case where there are many metadata fragments, usage information of which is worth describing, access/usage control can be performed in such a simple way as follows. Once access to an application is authorized, the application is believed to operate following predetermined usage rules set as default values.

In this case, an application program interface (API) of an RMP system is used to access or use metadata. The API is needed when access/usage control information is managed by a TVA RMP system. For example, the API issues and authorizes a request for accessing metadata. In addition, the API modifies, copies, and exports metadata.

As described above, there are several types of authentication that can be performed at a predetermined structure level, and they are transmission-level authentication, metadata container-level authentication or SOAP message-level authentication, and source-level authentication.

In the case of source-level authentication, authentication information on specific portions of metadata that have been authenticated is provided using a pointer. In the case of a SOAP message-level authentication, authentication information is included in a header of a SOAP message together with a pointer for part of metadata contained in the body of the SOAP message or a pointer for the entire metadata.

In a case where only metadata integrity is requested to be preserved during transmission of metadata, only transmission-level authentication is required. On the other hand, in a case where there is a need to secure transmission independence, metadata container-level authentication or SOAP message-level authentication can satisfy the need. The size of metadata contained in a metadata container or a body of an SOAP message is much larger than the size of a transmission packet. Therefore, transmission-level authentication helps reduce a system's load, in which case a security channel is not

necessary.

Authentication of a metadata source requires metadata container-level authentication and SOAP message-level authentication. The syntax of a metadata container enabling source authentication is shown in FIG. 11.

In order to perform source authentication on metadata at each node between a source and a final destination, source authentication information needs to be provided to each node between the source and the final destination.

More specifically, metadata is authenticated at a predetermined node between a source and a final destination using authentication information transmitted from a previous node, new authentication information is generated, and the metadata and the new authentication information are passed on to a next node. Alternatively, metadata is authenticated at a predetermined node using authentication information transmitted from a previous node, and the metadata and the authentication information are directly passed on to a next node so that the metadata can be authenticated again at the next node using the authentication information.

Accordingly, in the case of transmitting metadata from a source to a final destination while source-level-authenticating the metadata at each node between the source and the final destination, a flag or a signal, indicating whether new authentication information is generated or not after the metadata is authenticated at a predetermined node using authentication information transmitted from a previous node, can be inserted into the authentication information. The flag or signal indicating the presence of source authentication information helps a receiver determine whether to accept the corresponding metadata or not.

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.

The above-described embodiments of the present invention can be realized as computer-readable codes written on a computer-readable recording medium. The computer-readable recording medium includes all kinds of storages where computer-readable data can be stored, such as a ROM, a RAM, a CD-ROM, a magnetic tape, a hard disk, a floppy disk, a flash memory, an optical data storage, and a carrier wave, such as data transmission through the Internet. The computer-readable recording medium can be distributed over computer systems connected via a network so that the computer-readable codes written on the computer-readable recording medium can be executed in an independent manner.

Industrial Applicability

As described above, the method for managing metadata according to the present invention makes it possible to authenticate metadata at a metadata container level. Therefore, it is possible to carry out transmission-level authentication in any channel environment. In addition, the present invention makes it possible to selectively carry out either transmission-level authentication or source-level authentication or both by inserting data format information indicating the format of metadata into a metadata container. Considering that the size of a metadata container-level packet is larger than the size of a transmission-level packet, the present invention reduces the number of packets to be transmitted, thus simplifying a system.

What is claimed is:

1. A method for managing metadata in a metadata transmission server, comprising:

(a) generating a plurality of fragment data by partitioning metadata to be transmitted on the basis of a predetermined segment unit;

(b) selecting predetermined fragment data among the plurality of fragment data;

(c) generating metadata-related information using the selected fragment data; and

(d) transmitting the selected fragment data and the metadata-related information with data format information indicating the type of the selected fragment data.

2. The method of claim 1, wherein the selected fragment data, the metadata-related information, and the data formation information of the selected fragment data are transmitted in a metadata container.

3. The method of claim 1, wherein the data format information indicates whether the selected fragment data has a binary XML format or a text XML format.

4. The method of claim 1, wherein the plurality of fragment data are predetermined semantic units of the metadata.

5. The method of claim 2, wherein an authentication level flag specifying a metadata authentication level is further contained in the metadata container.

6. The method of claim 1, wherein the metadata-related information is metadata digest information obtained by substituting the selected fragment data into a unidirectional function.

7. The method of claim 6, wherein the unidirectional function is a hash function.

8. The method of claim 1 further comprising generating metadata authentication signature information using the metadata-related information and a first encryption key and inserting the metadata authentication signature information in the metadata container containing the selected fragment data.

9. The method of claim 8, wherein the metadata authentication signature information is obtained by substituting the metadata-related information and the first encryption key into a unidirectional function.

10. The method of claim 9 further comprising encrypting the first encryption key using a second encryption key and inserting the encrypted first encryption key into the metadata container containing the selected fragment data.

11. The method of claim 2, wherein the plurality of fragment data and their respectively metadata-related information are inserted into the metadata container, and each of the plurality of fragment data and its corresponding metadata-related information are connected to each other by pointer information.

12. The method of claim 8, wherein the plurality of fragment data and their respective metadata-related information and metadata authentication signature information are inserted into the metadata container, and each of the plurality of fragment data and its corresponding metadata-related information and metadata authentication

signature information are connected to one another by pointer information.

13. A method for managing metadata in a client that receives the metadata, comprising:

(a) reading predetermined fragment data and its corresponding metadata-related information and data format information indicating the type of the predetermined fragment data from the received metadata;

(b) generating metadata-related information using the predetermined fragment data and its corresponding data format information; and

(c) determining whether or not the received metadata has been authenticated by comparing the metadata-related information generated in step (b) with the metadata-related information read in step (a).

14. The method of claim 13, wherein the predetermined fragment data and its corresponding metadata-related information and data format information are received in a metadata container.

15. The method of claim 13, wherein the data format information indicates whether or not the predetermined fragment data has a binary XML format or a text XML format.

16. The method of claim 13, wherein the fragment data is a predetermined semantic unit of the received metadata.

17. The method of claim 14, wherein an authentication level flag is further included in the metadata container.

18. The method of claim 13, wherein the metadata-related information is metadata digest information obtained by substituting the

predetermined fragment data into a unidirectional function.

19. The method of claim 18, wherein the unidirectional function is a hash function.

20. The method of claim 14, wherein a plurality of fragment data and their respective metadata-related information are included in the metadata container, and each of the plurality of fragment data and its corresponding metadata-related information is connected to each other by pointer information.

21. A method for managing metadata in a client that receives the metadata, comprising:

(a) receiving fragment data of the received metadata, metadata-related information, data format information indicating the type of the fragment data, metadata authentication information, and an encrypted first encryption key;

(b) generating metadata-related information using the fragment data of the received metadata and its corresponding data format information;

(c) decrypting the encrypted first encryption key using a second encryption key stored in the client;

(d) generating metadata authentication signature information using the generated metadata-related information and the decrypted first encryption key; and

(e) determining whether or not the received metadata has been authenticated by comparing the generated metadata authentication signature information with the received metadata authentication signature information.

22. The method of claim 21, wherein the metadata-related

information is metadata digest information obtained by substituting the fragment data into a unidirectional function.

23. The method of claim 22, wherein the unidirectional function is a hash function.

24. The method of claim 21, wherein the generated metadata authentication signature information is obtained by substituting the generated metadata-related information and the decrypted first encryption key into a unidirectional function.

25. The method of claim 24, wherein the unidirectional function is a hash function.

26. The method of claim 21, wherein the fragment data, its corresponding metadata-related information, data format information, and metadata authentication signature information, and the encrypted first encryption key are received, being contained in a metadata container.

27. The method of claim 21, wherein the data format information indicates whether the fragment data used to generate the metadata-related information has a binary XML format or a text XML format.

28. The method of claim 21, wherein the fragment data is a predetermined semantic unit of the received metadata.

29. The method of claim 26, wherein an authentication level flag is further included in the metadata container.

30. The method of claim 26, wherein a plurality of fragment data and their respective metadata-related information and metadata authentication signature information are inserted into the metadata container, and each of the plurality of fragment data and its corresponding metadata-related information and metadata authentication signature information are connected to one another by pointer information.

FIG. 1

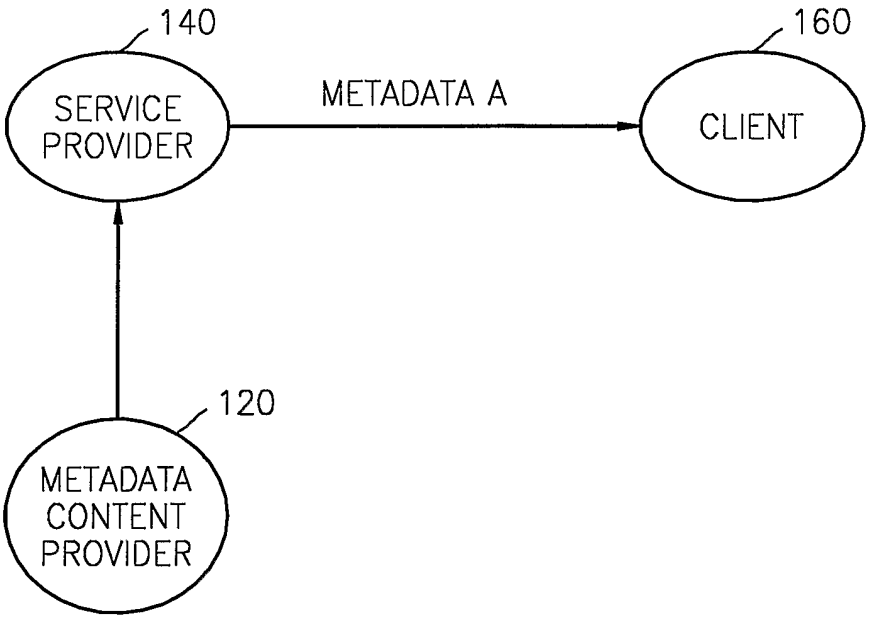

140
SERVICE PROVIDER
METADATA A
160
CLIENT
120
METADATA CONTENT PROVIDER

FIG. 2

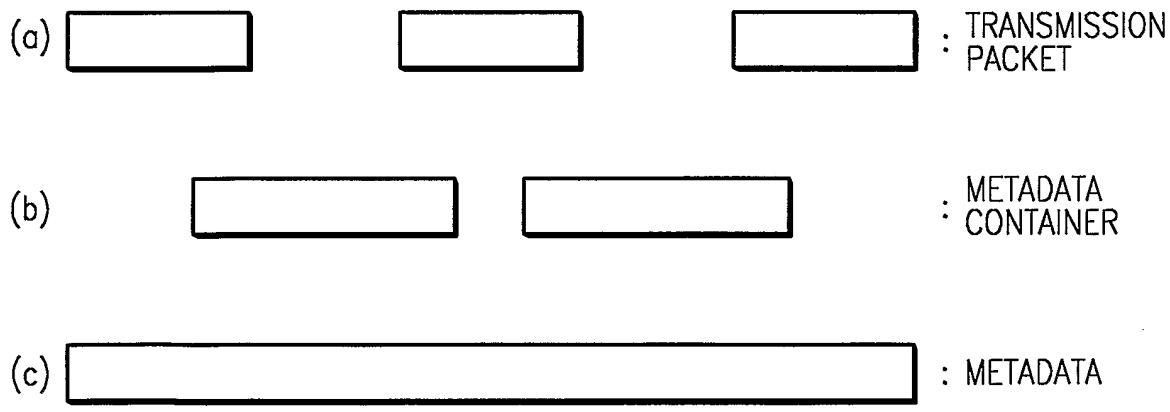

(a)
: TRANSMISSION PACKET
(b)
: METADATA CONTAINER
(c)
: METADATA

FIG. 3

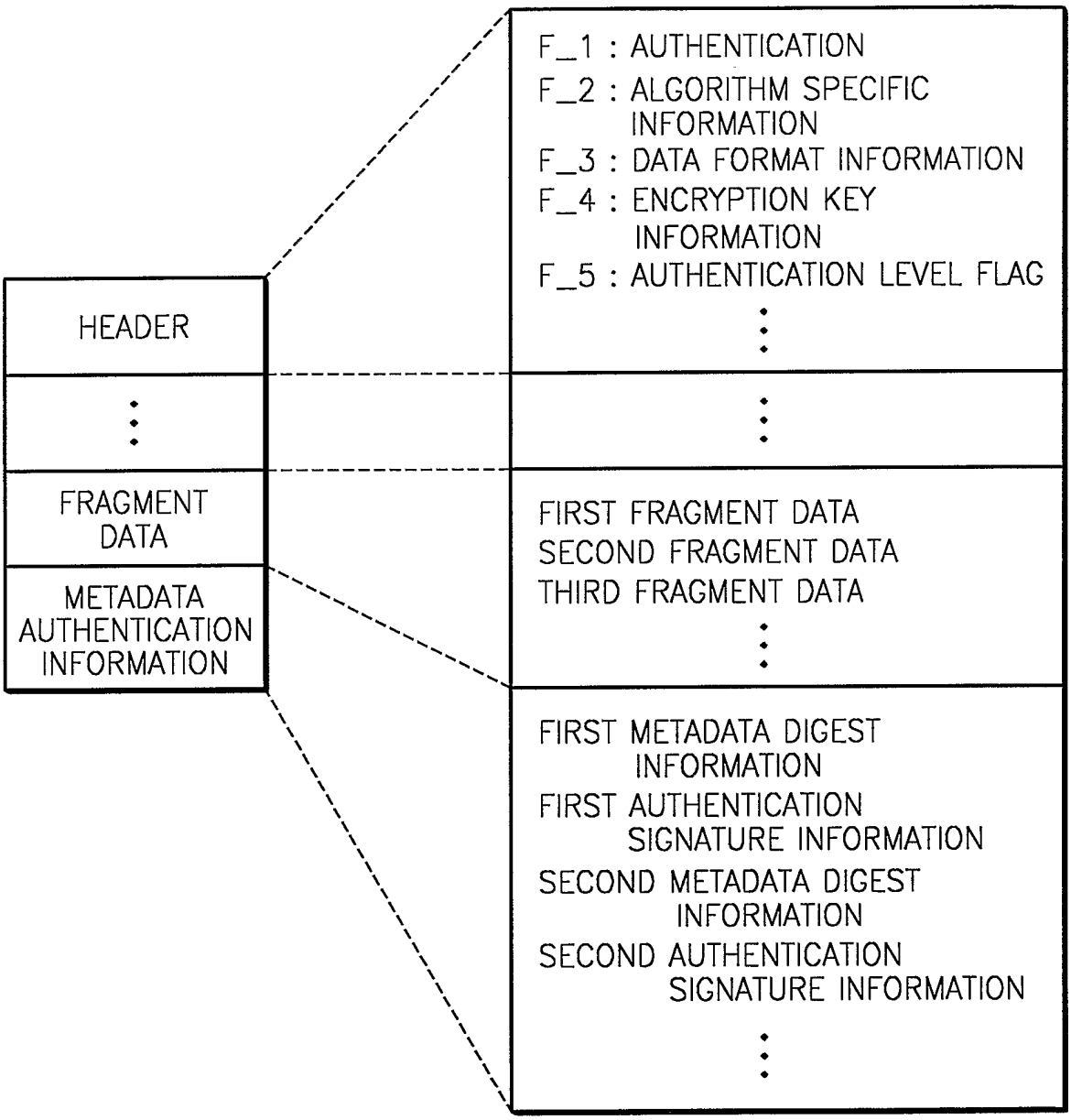

HEADER
FRAGMENT DATA
METADATA AUTHENTICATION INFORMATION
F_1 : AUTHENTICATION
F_2 : ALGORITHM SPECIFIC INFORMATION
F_3 : DATA FORMAT INFORMATION
F_4 : ENCRYPTION KEY INFORMATION
F_5 : AUTHENTICATION LEVEL FLAG
FIRST FRAGMENT DATA
SECOND FRAGMENT DATA
THIRD FRAGMENT DATA
FIRST METADATA DIGEST INFORMATION
FIRST AUTHENTICATION SIGNATURE INFORMATION
SECOND METADATA DIGEST INFORMATION
SECOND AUTHENTICATION SIGNATURE INFORMATION

FIG. 4

```
<SOAP:Envelope ...>
  <SOAP:Header>
      <Signature fragrefID = 1>
        <Algorithm ID=1>
        <Digest> ... <\Digest>
        <SignatureValue>  ...  </SignatureValue>
        <KeyInfo>  ... </KeyInfo>
        <SignatureValueBaseType>Text</SignatureValueBaseType>
        <AuthenticatioinLevel>Transport</AuthenticationLevel>
      </Signature>
  </SOAP:Header>
<SOAP:Body>
  <TVAmetadataFragment id=1>
      <ProgramInformation>...</ProgramInformation>
  </TVAmetadataFragment>
  <TVAmetadataFragment id=2>
      <SegmentInformation>...</SegmentInformation>
  </TVAmetadataFragment>
</SOAP:Body>
</SOAP:Envelope>
```

FIG. 5

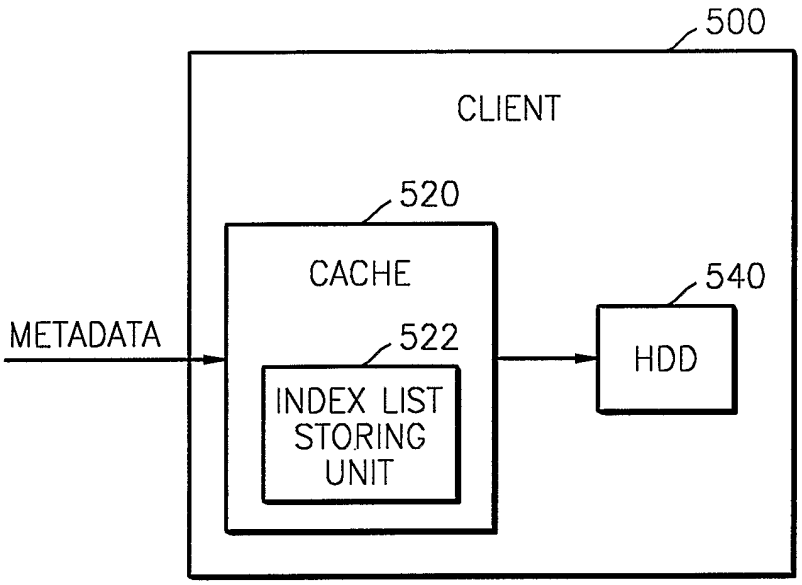

500
CLIENT
520
CACHE
540
METADATA
522
HDD
INDEX LIST STORING UNIT

FIG. 6

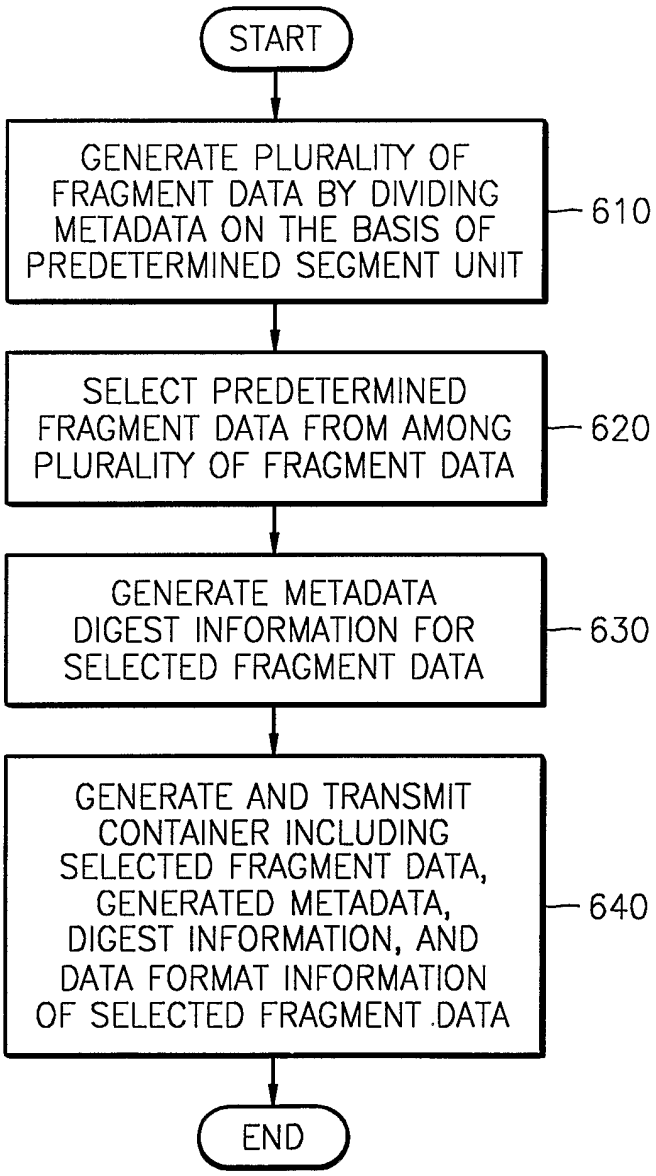

START
GENERATE PLURALITY OF FRAGMENT DATA BY DIVIDING METADATA ON THE BASIS OF PREDETERMINED SEGMENT UNIT
610
SELECT PREDETERMINED FRAGMENT DATA FROM AMONG PLURALITY OF FRAGMENT DATA
620
GENERATE METADATA DIGEST INFORMATION FOR SELECTED FRAGMENT DATA
630
GENERATE AND TRANSMIT CONTAINER INCLUDING SELECTED FRAGMENT DATA, GENERATED METADATA, DIGEST INFORMATION, AND DATA FORMAT INFORMATION OF SELECTED FRAGMENT DATA
640
END

FIG. 7

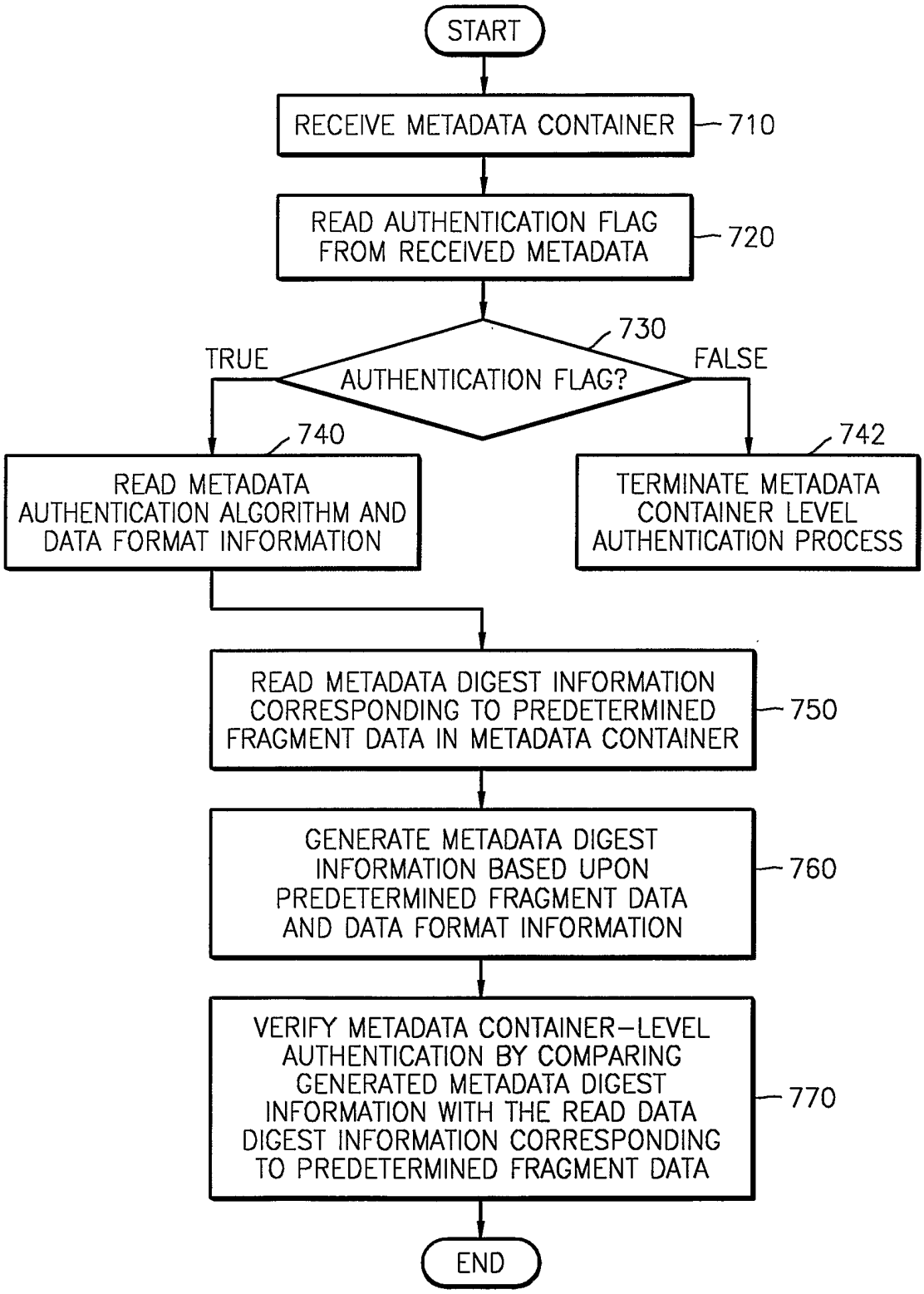

START
RECEIVE METADATA CONTAINER
710
READ AUTHENTICATION FLAG FROM RECEIVED METADATA
720
730
AUTHENTICATION FLAG?
TRUE
FALSE
740
READ METADATA AUTHENTICATION ALGORITHM AND DATA FORMAT INFORMATION
742
TERMINATE METADATA CONTAINER LEVEL AUTHENTICATION PROCESS
READ METADATA DIGEST INFORMATION CORRESPONDING TO PREDETERMINED FRAGMENT DATA IN METADATA CONTAINER
750
GENERATE METADATA DIGEST INFORMATION BASED UPON PREDETERMINED FRAGMENT DATA AND DATA FORMAT INFORMATION
760
VERIFY METADATA CONTAINER-LEVEL AUTHENTICATION BY COMPARING GENERATED METADATA DIGEST INFORMATION WITH THE READ DATA DIGEST INFORMATION CORRESPONDING TO PREDETERMINED FRAGMENT DATA
770
END

FIG. 8

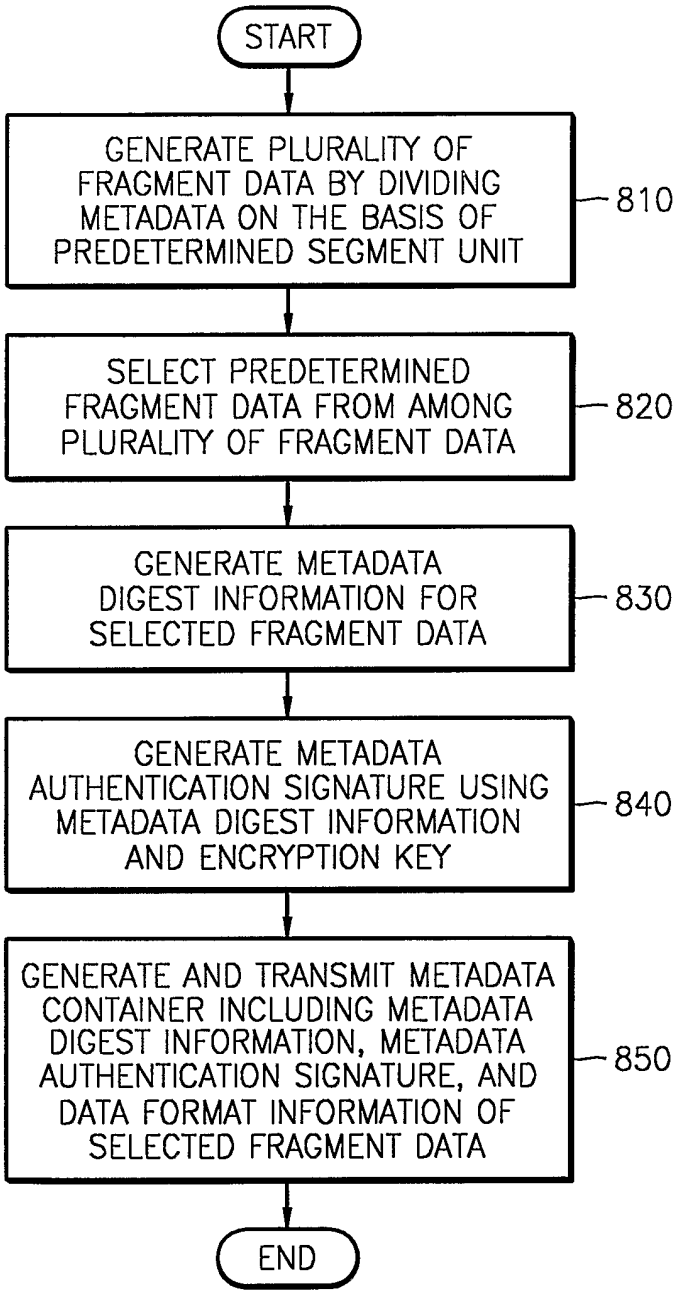

START
GENERATE PLURALITY OF FRAGMENT DATA BY DIVIDING METADATA ON THE BASIS OF PREDETERMINED SEGMENT UNIT
810
SELECT PREDETERMINED FRAGMENT DATA FROM AMONG PLURALITY OF FRAGMENT DATA
820
GENERATE METADATA DIGEST INFORMATION FOR SELECTED FRAGMENT DATA
830
GENERATE METADATA AUTHENTICATION SIGNATURE USING METADATA DIGEST INFORMATION AND ENCRYPTION KEY
840
GENERATE AND TRANSMIT METADATA CONTAINER INCLUDING METADATA DIGEST INFORMATION, METADATA AUTHENTICATION SIGNATURE, AND DATA FORMAT INFORMATION OF SELECTED FRAGMENT DATA
850
END

FIG. 9

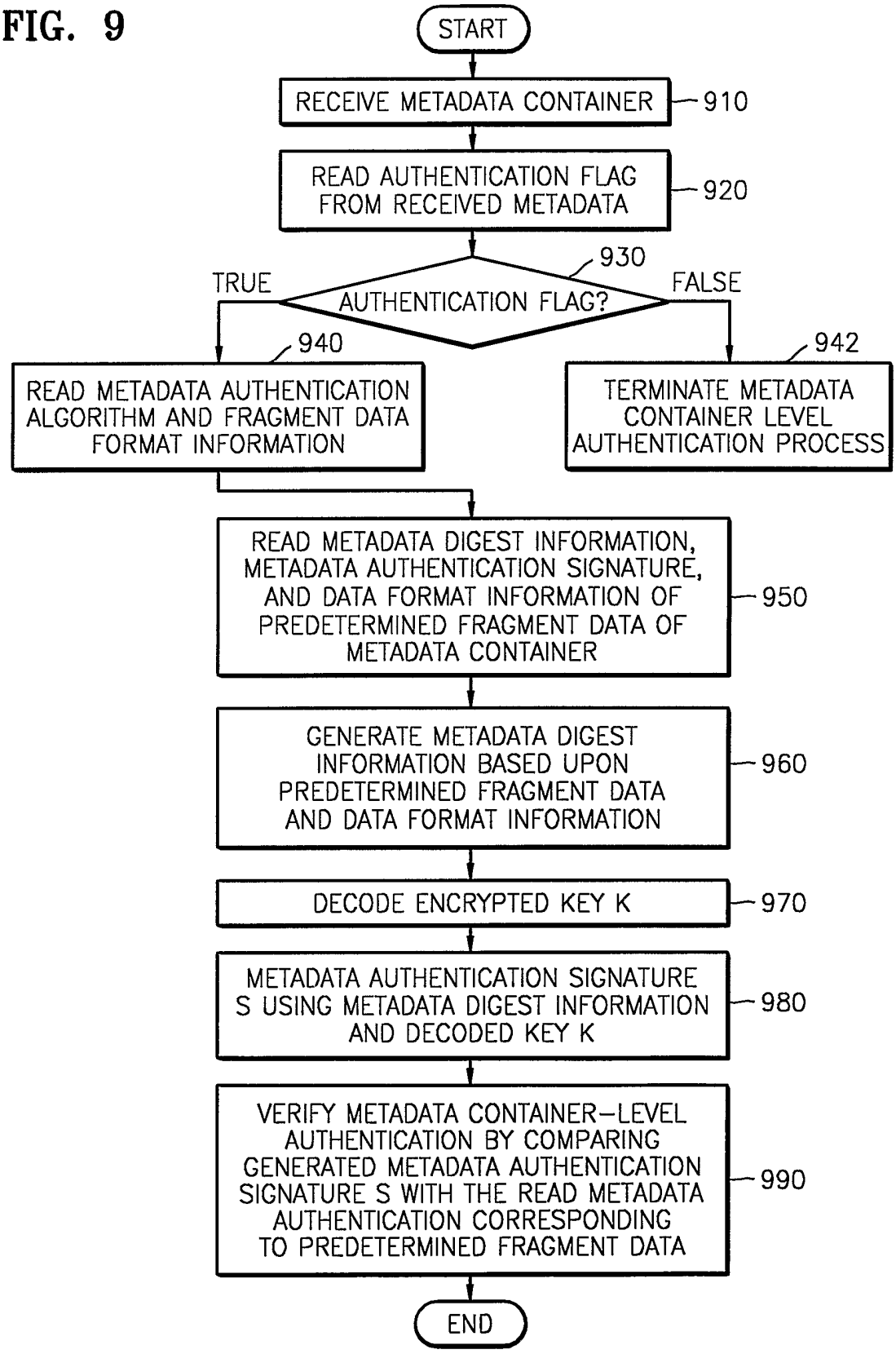

START
RECEIVE METADATA CONTAINER
910
READ AUTHENTICATION FLAG FROM RECEIVED METADATA
920
930
AUTHENTICATION FLAG?
TRUE
FALSE
940
READ METADATA AUTHENTICATION ALGORITHM AND FRAGMENT DATA FORMAT INFORMATION
942
TERMINATE METADATA CONTAINER LEVEL AUTHENTICATION PROCESS
READ METADATA DIGEST INFORMATION, METADATA AUTHENTICATION SIGNATURE, AND DATA FORMAT INFORMATION OF PREDETERMINED FRAGMENT DATA OF METADATA CONTAINER
950
GENERATE METADATA DIGEST INFORMATION BASED UPON PREDETERMINED FRAGMENT DATA AND DATA FORMAT INFORMATION
960
DECODE ENCRYPTED KEY K
970
METADATA AUTHENTICATION SIGNATURE S USING METADATA DIGEST INFORMATION AND DECODED KEY K
980
VERIFY METADATA CONTAINER-LEVEL AUTHENTICATION BY COMPARING GENERATED METADATA AUTHENTICATION SIGNATURE S WITH THE READ METADATA AUTHENTICATION CORRESPONDING TO PREDETERMINED FRAGMENT DATA
990
END

FIG. 10

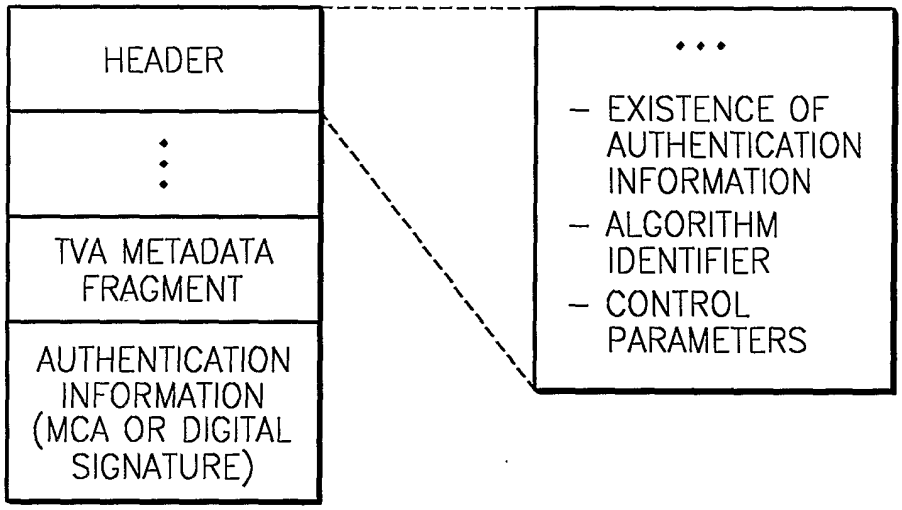

HEADER
TVA METADATA FRAGMENT
AUTHENTICATION INFORMATION (MCA OR DIGITAL SIGNATURE)
• • •
– EXISTENCE OF AUTHENTICATION INFORMATION
– ALGORITHM IDENTIFIER
– CONTROL PARAMETERS

FIG. 11


```
<?xml version="1.0" encoding="utf-8"?>
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
               xmlns:xsd="http://www.w3.org/2001/XMLSchema"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <SOAP:Header>
  <wssec:credentials xmlns:wssec="http://schemas.xmlsoap.org/ws/2001/10/security">
   <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="SigningCertificate">
    <ds:X509Data>
     <ds:X509Certificate>MIIH1zCCBr+gAwIBA...</ds:X509Certificate>
    </ds:X509Data>
   </ds:KeyInfo>
  </wssec:credentials>

  <wssec:integrity xmlns:wssec="http://schemas.xmlsoap.org/ws/2001/10/security">
   <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
     <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/xml-exc-c14n#"/>
     <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
     <ds:Reference URI=""/>
      <ds:Transforms>
       <ds:Transform Algorithm="http://schemas.xmlsoap.org/2001/10/security
         #RoutingSignatureTransform"/>
       <ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
      </ds:Transforms>
      <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <ds:DigestValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</DigestValue>
     </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>aiYECAxNqK2PivQaRweWajXup5zJa...</ds:SignatureValue>
    <ds:KeyInfo>
       <wssec:licenseLocation>
        #SigningCertificate
      </wssec:licenseLocation>
    </ds:KeyInfo>
     </ds:Signature>

  </wssec:integrity>
 </SOAP:Header>

<SOAP:Body>
 <TVAmetadataFragment>
  <ProgramInformation>.....</ProgramInformation>
      <enc:EncryptedData>
     <enc:EncryptionMethod Algorithm="xxx_algorithm"/>
     <ds:KeyInfo>.....</ds:KeyInfo>
     <enc:CipherData>
      <enc:CipherValue>9osy8Tw2+HcSHftHg...</enc:CipherValue>
     </enc:CipherData>
    </enc:EncryptedData>

    <enc:EncryptedKey>
          <enc:EncryptionMethod Algorithm="yyy_algorithm"/>
     <ds:KeyInfo>
      <ds:KeyName>Public/Private Key for TVA metadata</ds:KeyName>
       </ds:KeyInfo>
       <enc:CipherData>
        <enc:CipherValue>CCBPowCwYDVR0PBA...</enc:CipherValue>
         </enc:CipherData>
     </enc:EncryptedKey>
  </TVAmetadataFragment>
 </SOAP:Body>
</SOAP:Envelope>
```

INTERNATIONAL SEARCH REPORT

International application No.
PCT/KR03/00713

A. CLASSIFICATION OF SUBJECT MATTER

IPC7 G06F 17/00

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)
IPC7 G06F17/00, G06T1/00, H04L12/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
KR, JP IPC as above

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	KR 10-2002-70477 A (DIGIMAC CORP.) 09.SEP.2002 See the whole document	1-30
Y	KR 10-2000-33213 A (K.T.) 15.JUNE.2000 See the whole document	1-30
A	KR10-2002-63830 A (E.T.R.I) 05.AUG.2002 See the whole document	1-30
A	KR10-2002-45328 A (INHA UNIV.) 19.JUNE See the whole document	1-30

☐ 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
26 MAY 2003 (26.05.2003)	27 MAY 2003 (27.05.2003)

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
SONG, Dae Jong
Telephone No. 82-42-481-5992