

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
19 July 2007 (19.07.2007)

PCT

(10) International Publication Number
WO 2007/081163 A1

- (51) International Patent Classification:
H04L 29/06 (2006.01) H04L 9/32 (2006.01)
- (21) International Application Number:
PCT/KR2007/000200
- (22) International Filing Date: 11 January 2007 (11.01.2007)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
10-2006-0003337 11 January 2006 (11.01.2006) KR
10-2006-0003340 11 January 2006 (11.01.2006) KR
10-2006-0028355 29 March 2006 (29.03.2006) KR
- (71) Applicant: SAMSUNG ELECTRONICS CO., LTD.
[KR/KR]; 416, Maetan-dong, Yeongtong-gu, Suwon-si,
Gyeonggi-do 442-742 (KR).
- (72) Inventors: SONG, Young-Joo; #305-1702, Dong-
suwon LG Village 3-cha, Mangpo-dong, Yeong-
tong-gu, Suwon-si, Gyeonggi-do 443-773 (KR). JUNG,
Ki-Ho; #506-1404, Jugong 5-danji, Byeoryang-dong,
Gwacheon-si, Gyeonggi-do 427-732 (KR). KIM,
Mun-Churl; C/O Information & Communications
University, 103-6, Munji-dong, Yuseong-gu, Daejeon
305-732 (KR). HENDRY, Hendry; #127, Dormitor Of

C/O Information & Communications University, 119,
Munji-dong, Yuseong-gu, Daejeon 305-73 (KR).

(74) Agent: LEE, Keon-Joo; Mihwa Bldg. 110-2, Myon-
gryun-dong 4-ga, Chongro-gu, Seoul 110-524 (KR).

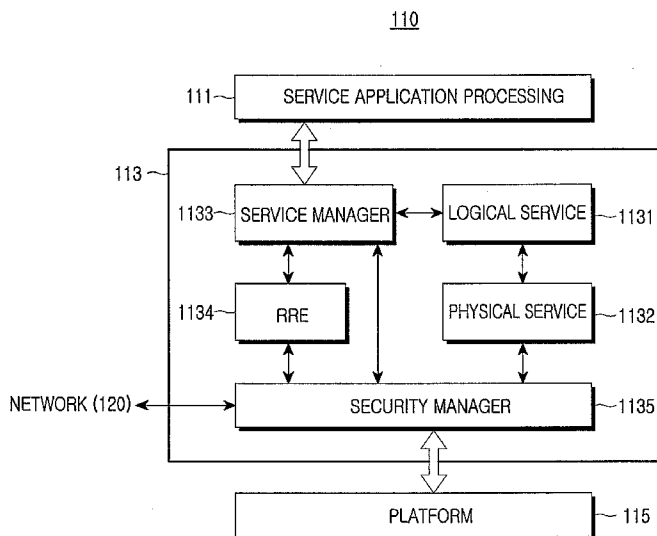
(81) Designated States (unless otherwise indicated, for every
kind of national protection available): AE, AG, AL, AM,
AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN,
CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI,
GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS,
JP, KE, KG, KM, KN, KP, KZ,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

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

Published:
— with international search report

[Continued on next page]

(54) Title: SECURITY MANAGEMENT METHOD AND APPARATUS IN MULTIMEDIA MIDDLEWARE, AND STORAGE
MEDIUM THEREFOR



(57) Abstract: A security management method in multimedia middleware of a terminal that can download a multimedia service provided from an external entity. The security management method includes if there is an access request for a particular multimedia service, determining whether there is a corresponding multimedia service; if the access-requested multimedia service does not exist in middleware of the terminal, performing negotiation for secure session setup with the external entity; setting up a secure session to the external entity using a security parameter selected as a result of the negotiation; and receiving information for the download from the external entity through the secure session, and determining whether to execute the download.

WO 2007/081163 A1



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.

**SECURITY MANAGEMENT METHOD AND APPARATUS IN
MULTIMEDIA MIDDLEWARE, AND STORAGE MEDIUM THEREFOR**

BACKGROUND OF THE INVENTION

5

1. Field of the Invention

The present invention relates generally to security technology related to multimedia middleware, and in particular, to a security management method and apparatus in multimedia middleware, capable of effectively managing security-related operations during interaction between multimedia middleware of a terminal and a multimedia service provider (or repository) providing multimedia contents. In addition, the present invention relates to a method and apparatus for security management related to multimedia middleware in a terminal.

15

2. Description of the Related Art

Generally, middleware refers to a software layer for interworking, such as data exchange, between an application and a network for providing identification, authentication, control and multimedia services related to devices in the network, or between applications. That is, when an application is connected to the network or applications are directly connected to each other, codes should generally be added to all of the associated applications so that an application can communicate with network devices or with another application. The use of the middleware can omit the troublesome operation of adding codes to the associated applications, because the middleware supports interworking between an application and a network, or between applications.

Of various middleware technologies, multimedia middleware technology supports, for example, on-line content, Internet broadcast, and digital data broadcast reception, and a typical communication network using the multimedia middleware can include a mobile communication network that provides various packet services to user terminals such as mobile phones. For example, for the mobile communication network, manufacturers of the conventional mobile terminals comprehensively design and manufacture mobile terminals including necessary hardware and software therein, taking the functions and services required by users into account. However, to meet the rapid growth of the mobile

35

communication market and diversified user demands, there is a need for frequent design change. Therefore, the conventional terminal design method could not provide sufficient flexibility for accommodating the market needs.

5 Recently, therefore, with the use of middleware technology for
interfacing between a platform and an upper application of the terminal, the
mobile terminal manufacturers, recognizing the limitations of the conventional
terminal design method, now provide support for various multimedia applications
without greatly modifying the design of the mobile terminal every time they
10 install a new application.

 However, multimedia middleware technology has managed a multimedia-
related security processing function with no device capable of managing a
security process, included in the middleware, without using a separate security
15 process during communication with an external entity such as the multimedia
service provider, or without using a universal procedure even though there is a
security process. In addition, the conventional technology may not prevent spread
of malicious codes and viruses using the security loophole, and should
inefficiently manage the middleware due to resource waste, high complexity of a
20 security processing function, and the like, caused by the absence of the universal
security procedure.

 In addition, the absence of the standardized security management method
and procedure disables an automatic negotiation process through the middleware
25 during communication of a mobile terminal that uses various security algorithms,
thereby causing an unnecessary waste of time and terminal resources for setting
up a new secure channel, and increasing complexity of the negotiation process.

SUMMARY OF THE INVENTION

30 An object of the present invention is to address at least the problems
and/or disadvantages above and to provide at least the advantages described
below. Accordingly, an object of the present invention is to provide a method and
apparatus for security management between a terminal using multimedia
35 middleware and an external service provider (or repository), and a storage

medium therefor.

Another object of the present invention is to provide a security management method and apparatus for a middleware service in a terminal using multimedia middleware, and a storage medium therefor.

Further, another object of the present invention is to provide a metadata structure capable of specifying a security algorithm and protocol, which are being used or can be used in multimedia middleware.

According to one aspect of the present invention, there is provided a security management method in multimedia middleware of a terminal that can download a multimedia service provided from an external entity. The security management method includes if there is an access request for a particular multimedia service, determining whether there is a corresponding multimedia service; if the access-requested multimedia service does not exist in middleware of the terminal, performing negotiation on secure session setup with the external entity; setting up a secure session to the external entity using a security parameter selected as a result of the negotiation result; and receiving information for the download from the external entity through the secure session, and determining whether to execute the download.

According to another aspect of the present invention, there is provided a security management apparatus in multimedia middleware of a terminal that can download a multimedia service provided from an external entity. The security management apparatus includes a security manager for, if a requested particular multimedia service does not exist in middleware of the terminal, performing negotiation on secure session setup with the external entity, and setting up a secure session to the external entity using a security parameter selected as a result of the negotiation result; and a service manager for, if there is an access request for the particular multimedia service, determining whether there is the particular multimedia service, receiving information for the download from the external entity through the secure session, and determining whether to execute the download.

According to a further aspect of the present invention, there is provided a security management method performed in multimedia middleware in a terminal. The security management method includes if there is an access request to a particular multimedia service from a particular application program, checking a license list including its access right information; and if there is an access right of the particular application program, delivering a handler for the particular multimedia service to the particular application program. The license list includes information indicating an access right of application programs for middleware services supportable by the terminal.

According to yet another aspect of the present invention, there is provided a security management method performed in multimedia middleware in a terminal. The security management method includes: if there is an access request to a particular platform resource from a particular multimedia service, generating an access identifier thereof; and checking a license list including access right information based on the access identifier; and if there is an access right to the particular multimedia service, delivering system resource to the particular multimedia service. The license list comprises information indicating an access right to multimedia services for platform resources supported by the terminal.

According to still another aspect of the present invention, there is provided a security management apparatus included in multimedia middleware in a terminal. The security management apparatus includes a service manager for receiving an access request to a particular multimedia service from a particular application program, and delivering a handler for the particular multimedia service to the particular application program, if there is an access right to the particular application program; and a security manager for checking a license list including access right information for the particular application program in response to a request of the service manager. The license list comprises information indicating an access right to application programs for middleware services supportable by the terminal.

According to still another aspect of the present invention, there is provided a security management apparatus included in multimedia middleware in

5 a terminal. The security management apparatus includes a security manager for, if there is an access request to a particular platform resource from a particular multimedia service, generating an access identifier thereof, and if there is an access right to the particular multimedia service, delivering system resources to the particular multimedia service; and an (M3W Run-time Environment) RRE for checking a license list including access right information based on the access identifier in response to a request of the security manager. The license list comprises information indicating an access right to multimedia services for platform resources supported by the terminal.

10

BRIEF DESCRIPTION OF THE DRAWINGS

15 The above and other objects, features and advantages of the present invention will become more apparent from the following detailed description when taken in conjunction with the accompanying drawings in which:

FIG. 1 is a block diagram illustrating a configuration of a network system to which a user terminal including a multimedia middleware apparatus according to the present invention makes access;

20

FIG. 2 is a block diagram illustrating a structure of a user terminal including a multimedia middleware apparatus according to the present invention;

FIG. 3 is a flowchart illustrating a download process with an external entity in a multimedia framework according the present invention;

25

FIG. 4 is a flowchart illustrating a detailed operation of the Location Sub-process of FIG. 3;

FIG. 5 is a flowchart illustrating a detailed operation of the Decision Sub-process of FIG. 3;

FIG. 6 is a diagram illustrating a data structure of the service information (Service Profile) described in FIG. 5;

30

FIG. 7 is a diagram illustrating a data structure of the execution environment information (Target Profile) described in FIG. 5;

FIG. 8 is a diagram illustrating a data structure of the security information (Security Profile) described in FIG. 5;

35

FIG. 9 is a flowchart illustrating a detailed operation of the Transfer Sub-process in FIG. 3;

FIG. 10 is a flowchart illustrating a negotiation process for communication with an external entity according to the present invention;

FIG. 11 is a diagram illustrating the overall structure of metadata that has defined a security protocol and algorithm according to the present invention;

5 FIG. 12 is a diagram illustrating a metadata structure of the Data Transfer Protocols in FIG. 11;

FIG. 13 is a diagram illustrating a metadata structure of the Authentication Protocols in FIG. 11;

10 FIG. 14 is a diagram illustrating a metadata structure of the Signature Types 1130 in FIG;

FIG. 15 is a diagram illustrating a metadata structure of the Encryption Types in FIG. 11;

15 FIG. 16 is a flowchart illustrating a security process during access to an internal service in a multimedia middleware apparatus according to the present invention;

FIG. 17 is a flowchart illustrating a security process during access to an internal resource in a multimedia middleware apparatus according to the present invention;

20 FIG. 18 is a diagram illustrating the overall data structure for access right expression according to the present invention;

FIG. 19 is a diagram illustrating a data structure for expression of an access principal (Principal) according to the present invention;

FIG. 20 is a diagram illustrating a data structure for expressing access types (AccessType) according to the present invention;

25 FIG. 21 is a diagram illustrating a data structure for expressing an access object (AccessObject) according to the present invention;

FIG. 22 is a diagram illustrating a data structure for expressing access conditions (Conditions) according to the present invention;

30 FIG. 23 is a diagram illustrating the overall data structure of an access right expression group according to the present invention; and

FIG. 24 is a diagram illustrating a basic structure of expressing a right using a license structure of Right Expression Language (REL) technology according to the present invention.

Preferred embodiments of the present invention will now be described in detail with reference to the annexed drawings. In the following description, a detailed description of known functions and configurations incorporated herein has been omitted for clarity and conciseness.

5

The present invention will be described in two aspects.

10 A first aspect of the present invention described with reference to FIGs. 1 to 15 provides a security management method and apparatus for handling security problems which may occur when a user terminal requiring a multimedia application communicates with a multimedia service provider (or repository), and an open multimedia middleware structure including a security management means. The security management means is for authentication, security parameter exchange and secure channel setup during service exchange with a multimedia
15 service provider, which is an external entity.

To this end, the present invention provides a metadata (or MPEG Multimedia Middleware Logical Service (M3WLS)) structure that defines a security protocol and algorithm for performing a security procedure which may
20 occur when a user terminal requiring a multimedia application communicates with a multimedia service provider, which is an external entity, as shown in, for example, FIGs. 11 to 15. The metadata structure that has defined the security protocol and algorithm is a data structure including specification on Data Transfer Protocols, Authentication Protocols, Signature Types, and Encryption Types
25 during service exchange between a mobile terminal including multimedia middleware according to the present invention and a multimedia service provider, and secure channel setup and data exchange between multimedia service entities is achieved using the metadata according to the present invention.

30 A second aspect of the present invention described with reference to FIGs. 16 to 24 provides a security management method and apparatus for performing Governed Access through License List management when an application program/middleware service makes access to internal service/internal resource in a user terminal requiring a multimedia application, and for verifying whether the
35 application program/middleware service has a right to use, and a multimedia

middleware structure including a security management means.

That is, with the use of the metadata structure, the second aspect provides a procedure for verifying whether an application program has an access and use right for an internal service when a multimedia terminal provides a middleware service needed by the multimedia application program, and for verifying whether the middleware service has an access and use right for the internal service when the middleware service uses internal resources of a terminal platform, and also provides an access right.

The metadata expressing the access right specifies, for example, access principal (Principal), access object (AccessObject), access condition (Condition), and access type (AccessType), in order to express "what principal can make what access for what object in what condition," and with the use of it, provides a method and apparatus for checking an access and use right between an application program and an internal service, i.e. between middleware services, or between a middleware service and a platform resource.

In the present invention, the metadata, which is systematized data, i.e. data for describing other data, means data added to contents according to a stated rule so as to allow a terminal to efficiently search for desired information from the large amount of information. Further, in the present invention, the term "service" or "multimedia service" will be construed as a concept including a multimedia middleware service.

FIG. 1 is a block diagram illustrating a configuration of a network system to which a user terminal including a multimedia middleware apparatus according to the present invention makes access, and the block diagram shows a system application framework including a user terminal 110, a network 120, and a multimedia service provider (or repository) 130.

In FIG. 1, the user terminal 110 refers to a wire terminal (for example, a computer having a network access means), or a wireless terminal (for example, cellular phone, PCS, PDA, IMT-2000, PDA phone, and smart phone). The network 120 can be various wire/wireless communication networks. The

multimedia service provider 130 can be a server that provides a multimedia application to the user terminal 110, or a storage medium containing a multimedia application. The multimedia application can be execution codes for content, metadata or multimedia middleware service, i.e. can be service implementation codes. The user terminal 110 includes a multimedia middleware block 113 that performs a security process of processing metadata and execution codes received from an upper layer through a service application processing block 111 for executing a multimedia application, the network 120, or a storage medium; performing a security process which may occur during service exchange with the multimedia service provider 130 according to the first aspect of the present invention; verifying whether the application program has an access and use right for the middleware service according to the second aspect of the present invention; and verifying whether the middleware service has an access and use right for resources of a platform, and also includes a platform block 115 in which a basic operating system (OS) supporting a multimedia service of the user terminal 110 is installed.

The term "security" in the first aspect of the present invention means an operation of setting up a secure session when a terminal downloads a service from the external multimedia service provider 130, and the term "security" in the second aspect of the present invention means an operation of verifying an access and use right (hereinafter an "access right") between application program in a terminal, middleware service, and resources.

FIG. 2 is a block diagram illustrating a structure of a user terminal including a multimedia middleware apparatus according to the present invention.

Referring to FIG. 2, a multimedia middleware block 113 constituting the multimedia middleware apparatus is connected to a service application processing block 111 via an application layer interface (not shown), and is connected to a platform block 115 via a platform layer interface (not shown). The multimedia middleware block 113 includes a logical service block 1131 composed of service metadata, a physical service block 1132 composed of service implementation codes, a service manager 1133 for management of the logical service and the physical service, an M3W Run-time Environment (RRE) 1134, and a security

manager 1135.

In the first aspect of the present invention, the service manager 1133 corresponding to a controller of the multimedia middleware block 113 performs management on the service metadata of the multimedia middleware. That is, the metadata necessary for the multimedia service transmitted from the multimedia service provider 130 via the network 120 of FIG. 1 is received through a service metadata transceiver (not shown), and the service manager 1133 checks service implementation codes of the metadata and downloads corresponding service implementation codes.

The service manager 1133 acquires a Service Profile (hereinafter "service information") as information on a corresponding service from the multimedia service provider 130, acquires a Target Profile (hereinafter "execution environment information") as information on an execution environment of the corresponding service from the RRE 1134, and acquires a Security Profile (hereinafter "security information") as information on a security method from the security manager 1135.

To this end, the security manager 1135 sets up a communication channel (secure session) through a predetermined security procedure with the multimedia service provider 130, and the service manager 1133 downloads the service information, execution environment information, and security information from the multimedia service provider 130. The download operation is performed when a user has requested access to a specific service but there is no corresponding service in the middleware of the terminal.

A detailed description of an operation performed in the service manager 1133, the RRE 1134 and the security manager 1135 for verification of an access and use right for internal service/internal resource in the terminal supporting the multimedia middleware according to the second aspect of the present invention will be made later.

With reference to FIGs. 3 to 8, a description will first be made of a multimedia middleware security management method in which the multimedia

middleware block 113 sets up a security-established session (hereinafter “secure session”) by acquiring security information from the external multimedia service provider 130, and performs download of a new service.

5 FIG. 3 is a flowchart illustrating a download process with an external entity in a multimedia framework according to the present invention.

10 Referring to FIG. 3, even though a service application processing block 111 of a user terminal 110 has sent an access request for a particular service to a service manager 1133 in a multimedia middleware block 113, if there is no corresponding service in the middleware, the service manager 1133 first undergoes a download process for acquisition of the corresponding service from a multimedia service provider 130. For the download process, in step 301, the service manager 1133 sends a network address request for the multimedia service provider 130, which is an external entity that can provide the corresponding service, to an RRE 1134.

15 Thereafter, the service manager 1133 acquiring a network address of the multimedia service provider 130 first sends in step 303 a secure connection request to a security manager 1135 before it communicates with the corresponding multimedia service provider 130. The security manager 1135 performs a Location Sub-process 303 of setting up a communication channel through a predetermined security procedure, with the multimedia service provider 130.

20 In step 305, the service manager 1133 performs a Decision Sub-process of acquiring corresponding service information (Service Profile) from the multimedia service provider 130, acquiring execution environment information (Target Profile) from the RRE 1134, and acquiring security information (Security Profile) from the security manager 1135.

25 Thereafter, in a Transfer Sub-process of step 307, the service manager 1133 determines whether to perform or cancel download of the corresponding service, by analyzing the service information, execution environment information and security information acquired in step 305. If the service manager 1133

30

35

determines to perform the download, it starts the Transfer Sub-process. In this case, the service manager 1133 transfers its right for a secure session to the RRE 1134, and the RRE 1134 downloads the corresponding service through communication with the multimedia service provider 130. If the download service ends, the service manager 1133 sends a close request for the secure session to the security manager 1135.

If the Transfer Sub-process ends in this way, the service manager 1133 transfers in step 309 the newly acquired service handler to the service application processing block 111.

FIG. 4 is a flowchart illustrating a detailed operation of the Location Sub-process of FIG. 3, and this includes the network address acquisition process for the multimedia service provider 130, described in FIG. 3, and a security establishment process between the security manager 1135 and the multimedia service provider 130.

In step 401, the service manager 1133 sends a network address request for the multimedia service provider (repository) 130 that can provide the corresponding service, to the RRE 1134. In step 403, the RRE 1134 provides a network address connectable to the corresponding multimedia service provider (repository) 130 to the service manager 1133. Thereafter, in step 405, the service manager 1133 transfers the network address of the multimedia service provider 130 to the security manager 1135 before it communicates with the corresponding multimedia service provider 130, and then sends a security establishment request thereto.

In step 407, the security manager 1135 provides information on security parameters for security establishment, such as security parameter list, session ID and the like, to the multimedia service provider 130 having the address provided from the service manager 1133. In step 409, the multimedia service provider 130 selects a parameter and a session ID to be used for actual communication among the security parameters provided from the security manager 1135, and sends a response to the security manager 1135. In step 411, the security manager 1135 sends a session connection response to the multimedia service provider 130. If the

selection of the security parameters ends in this way, the security manager 1135 provides in step 413 the corresponding information to the service manager 1133 for secure session connection, and the service manager 1133 sends a connection request for the security-established communication session, i.e. secure session, to the security manager 1135.

FIG. 5 is a flowchart illustrating a detailed operation of the Decision Sub-process of FIG. 3.

Through the secure session with the multimedia service provider (repository) 130, which was set up through the Location Sub-process of FIG. 4, the service manager 1133 sends a request for corresponding service information (Service Profile) to the multimedia service provider 130 in step 501, and acquires the corresponding service profile information from the multimedia service provider 130 in step 503. Thereafter, in steps 505 and 507, the service manager 1133 sends a request for execution environment information (Target Profile) for the corresponding service to the RRE 1134 and acquires the execution environment information (Target Profile) from the RRE 1134. In steps 509 and 511, the service manager 1133 sends a request for security information (Security Profile) for the corresponding service to the security manager 1135, and acquires the security information (Security Profile) from the security manager 1135. The security information (Security Profile) can be determined, for example, through the negotiation performed in steps 409 and 411 between the terminal and the external entity.

Thereafter, in step 513, by analyzing the acquired service information, execution environment information and the security information, the service manager 1133 determines whether the new service requested by the user coincides with the current execution environment resources, and whether the security information is valid, and then determines whether to download the corresponding service.

FIG. 6 is a diagram illustrating a data structure of the service information (Service Profile) described in FIG. 5.

Referring to FIG. 6, service information (Service Profile) 600 according to the present invention includes size information (ServiceNumberOfBytes) 610 of a service to be transmitted from the multimedia service provider 130 to the user terminal 110, capacity information (InstallationSize) 620 of a disk required for installation of the service, size (InstantiationSize) 630 of resources necessary for service execution, capacity information (MemorySize) 640 of a memory required for execution of the service, computing specification information (ComputingCapability) 650 specifying the CPU and OS specifications required for execution of the service, and description information (FreeDescription) 660 describing other requirements related to the corresponding service. "Instantiation" in the InstantiationSize 630 means a physical size of at least one execution file related to the service profile.

FIG. 7 is a diagram illustrating a data structure of the execution environment information (Target Profile) described in FIG. 5. Referring to FIG. 7, execution environment information (Target Profile) 700 according to the present invention includes available storage space information (AvailableStorageSize) 710, available memory capacity information (AvailableMemorySize) 720, size information (InstantiationSize) 730 of resources necessary for service execution, computing specification information (ComputingCapability) 740 specifying the CPU and OS specifications, and description information (FreeDescription) 750 describing other requirements related to the corresponding service.

FIG. 8 is a diagram illustrating a data structure of the security information (Security Profile) described in FIG. 5. Referring to FIG. 8, security information (Security Profile) according to the present invention includes integrity mechanism information (IntegrityMechanism) 810 for a digital signature algorithm, data encryption algorithm information (DataEncryptionAlgorithm) 820, and description information (FreeDescription) 830 describing other requirements related to the corresponding service.

FIG. 9 is a flowchart illustrating a detailed operation of the Transfer Sub-process in FIG. 3.

If the service manager 1133 determines to download the corresponding

service in the Decision Sub-process of FIG. 5, the RRE 1134 receives the download decision from the service manager 1133 in step 901, and sends a download request for the corresponding service to the multimedia service provider (repository) 130 in step 903, thereby starting the Transfer Sub-process.
5 Then the service manager 1133 transfers a right for the secure session to the RRE 1134, and the RRE 1134 sends a signal indicating the ready to receive the service to the multimedia service provider 130

In step 905, upon receipt of the ready-to-receive signal, the multimedia service provider 130 sends the corresponding service to the terminal. In step 907,
10 if the download of the service ends, the RRE 1134 provides the corresponding information to the service manager 1133. Thereafter, in step 909, the service manager 1133 sends a close request for the secure session to the security manager 1135. In step 911, the security manager 1135 closes the secure session established according to procedure of Fig.4 and provides information indicating the release of
15 the secure connection to the service manager 1133.

With reference to FIGs. 10 to 15, a description will now be made of a metadata structure that has defined the security protocol and algorithm of the
20 present invention, such that a terminal including multimedia middleware performs a security procedure which may occur during communication with the multimedia service provider, which is an external entity.

Generally, the metadata, which is systematized data; i.e. data for
25 describing other data, means data added to multimedia content according to a stated rule in order to efficiently search for desired information from a large amount of information.

FIG. 10 is a flowchart illustrating a negotiation process for
30 communication with an external entity according to the present invention.

Referring first to FIG. 2, even though the service application processing block 111 of the user terminal 110 has sent an access request for a particular service to the service manager 1133 in the multimedia middleware block 113, if
35 there is no corresponding service in the middleware, the service manager 1133

first undergoes a download process for acquisition of the corresponding service. Referring to FIG. 10, in the download process, in steps 1001 and 1003, the service manager 1133 sends a network address request for the multimedia service provider (repository) 130 to the RRE 1134 in the same manner as the operation
5 (Location Sub-process) of step 303, and receives the network address of the corresponding multimedia service provider 130.

Thereafter, in step 1005, the service manager 1133 sends a request for a secure connection to the security manager 1135 before it downloads the
10 corresponding service from the multimedia service provider 130. In steps 1007 and 1009, the security manager 1135 sets up a communication channel through a negotiation procedure for secure session setup with the multimedia service provider 130. In step 1007, the security manager 1135 requests the negotiation by transmitting a list of security algorithms and protocols to the multimedia service
15 provider 130 or a security management means (not shown) of the multimedia service provider 130. The multimedia service provider 130 receiving the list selects its preferred algorithm and protocol by analyzing the security algorithm and protocol list used by the entity from which the negotiation request was received, and then sends a response. This algorithm and protocol negotiation is
20 repeated until the negotiation is completed.

FIG. 11 is a diagram illustrating the overall structure of metadata that has defined a security protocol and algorithm according to the present invention.

Referring to FIG. 11, a brief definition is given of a data structure of
25 metadata 1100 that includes Data Transfer Protocols 1110 used for service exchange between the user terminal 110 and the multimedia service provider 130, Authentication Protocols 1120, Signature Types 1130 and Encryption Types 1140.

The security manager 1135 of FIG. 2 transmits the metadata 1100 defined
30 in FIG. 11 to the multimedia service provider 130, selects its preferred algorithm and protocol through the negotiation procedure according to step 1007 of FIG. 10, and sets up a secure channel using the selected algorithm and protocol.

FIG. 12 is a diagram illustrating a metadata structure of the Data Transfer
35

- 17 -

Protocols 1110 in FIG. 11, and this specifies a list of data transfer protocols 1110a supported by the multimedia middleware of the present invention.

Referring to FIG. 12, the list of the transfer protocols 1110a can be enumerated from 1 to infinity, and each protocol is composed of a pair of Uniform Resource Identifiers (URIs) and preferences of the protocol algorithm recommended in World Wide Web Consortium (W3C). The preference of the protocol is expressed with an integer, and for the protocol preferred by the corresponding middleware, the preference has a greater integer.

FIG. 13 is a diagram illustrating a metadata structure of the Authentication Protocols 1120 in FIG. 11, and this specifies a list of authentication protocols 1120a supported by the multimedia middleware of the present invention.

Referring to FIG. 13, the list of the authentication protocols 1120a can be enumerated from 1 to infinity, and each protocol is composed of a pair of URIs and preferences of the protocol algorithm recommended in W3C. The preference of the protocol is expressed with an integer, and for the authentication algorithm preferred by the corresponding middleware, the preference has a greater integer.

FIG. 14 is a diagram illustrating a metadata structure of the Signature Types 1130 in FIG. 11, and this specifies a list of signatures supported by the multimedia middleware of the present invention.

The Signature Types 1130 of FIG. 14 includes an algorithm list for each of Canonicalization Methods 1130a, Signature Methods 1130b, Transform Methods 1130c, and Digest Methods 1130d.

It is noted that the Transform Methods 1130c shown by a dotted line are optional. The Canonicalization Methods 1130a is a list of Canonicalization Algorithms used for electronic signatures provided by the middleware. The Signature Methods 1130b is a list of signature generation and verification algorithms provided by the middleware, the Transform Methods 1130c is a list of Transform Method algorithms for electronic signatures provided by the

middleware, and the Digest Methods 1130d is a list of Digest algorithms for electronic signatures provided by the middleware. Each of the algorithm lists can be enumerated from 1 to infinity, and each list is composed of a pair of URIs and preferences of the protocol algorithm. The preference of the protocol is expressed
5 with an integer, and for the electronic signature algorithm preferred by the corresponding middleware, the preference has a greater integer.

FIG. 15 is a diagram illustrating a metadata structure of the Encryption Types 1140 in FIG. 11, and this specifies a list of encryption algorithms supported
10 by the multimedia middleware of the present invention.

Referring to FIG. 15, a list of the Encryption Types 1140 includes an algorithm list for each of Encrypted Data Encoding Types 1140a, Encryption Methods 1140b, Digest Methods 1140c, and Key Agreement Methods 1140d. It is
15 noted that the Key Agreement Methods 1140d shown by a dotted line are optional. Each of the encryption algorithm lists can be enumerated from 1 to infinity, and each list is composed of a pair of URIs and preferences of the encryption algorithm. The preference of the algorithm is expressed with an integer, and for the encryption algorithm preferred by the corresponding middleware, the
20 preference has a greater integer.

Table 1 to Table 4 below illustrate the exemplary overall structure of the security data of FIG. 11 defined in the present invention, with eXtensible Markup Language (XML) description.
25

Table 1

```

<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://mccb.icu.ac.kr/M3WSecManProfile"
5  xmlns:m3w="http://
mccb.icu.ac.kr/M3WsecManProfile" xmlns="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <element name="SupportedSecurityEnvironment" type="
10 SupportedSecurityEnvironmentType"/>
  <complexType name=" SupportedSecurityEnvironmentType ">
    <sequence>
      <element ref="m3w:DataTransferProtocols"/>
      <element ref="m3w:AuthenticationProtocols"/>
15  <element ref="m3w:SignatureTypes"/>
      <element ref="m3w:EncryptionTypes"/>
    </sequence>
  </complexType>
  <element name="DataTransferProtocols"
20 type="m3w:DataTransferProtocolsType" />
  <complexType name="DataTransferProtocolsType">
    <sequence>
      <element ref="m3w:TransferProtocol" maxOccurs="unbounded"/>
    </sequence>
25 </complexType>
  <element name="AuthenticationProtocols"
type="m3w:AuthenticationProtocolsType" />
  <complexType name=" AuthenticationProtocolsType">
    <sequence>
30  <element ref="m3w:
AuthenticationProtocol" maxOccurs="unbounded"/>
    </sequence>

```

Table 2

```

</complexType>
  <element name="TransferProtocols" type="m3w:MethodType" />
  <element name="AuthenticationProtocols" type="m3w:MethodType" />
5  <element name="Method" type="m3w:MethodType" />
  <complexType name="MethodType">
    <attribute name="algorithm" thpe="anyURI" use="required"/>
    <attribute name="preference" thpe="Integer" use="required"/>
  </complexType>
10 <element name="SignatureTypes" type="m3w:SignatureTypesType" />
  <complexType name="SignatureTypesType">
    <sequence>
      <element ref="m3w:CanonicalizationMethods"/>
      <element ref="m3w:SignatureMethods"/>
15  <element ref="m3w:TransformMethods" minOccurs="0"/>
      <element ref="m3w:DigestMethods"/>
    </sequence>
  </complexType>
  <element name="CanonicalizationMethods"
20 type="m3w:CanonicalizationMethodsType" />
  <complexType name="CanonicalizationMethodsType">
    <sequence>
      <element ref="m3w:Method" maxOccurs="unbounded"/>
    </sequence>
25

```

Table 3

```

</complexType>
  <element name="SignatureMethods" type="m3w:SignatureMethodsType" />
  <complexType name="SignatureMethodsType">
5     <sequence>
        <element ref="m3w:Method" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
  <element name="TransformMethods" type="m3w:TransformMethodsType" />
10 <complexType name="TransformMethodsType">
    <sequence>
        <element ref="m3w:Methods" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
15 <element name="DigestMethods" type="m3w:DigestMethodsType"/>
  <complexType name="DigestMethodsType">
    <sequence>
        <element ref="m3w:Methods" maxOccurs="unbounded"/>
    </sequence>
20 </complexType>
  <element name="EncryptionTypes" type="m3w:EncryptionTypesType"/>
  <complexType name="EncryptionTypesType">
    <sequence>
        <element ref="m3w:EncryptedDataEncodingTypes"/>
25        <element ref="m3w:EncryptionMethods"/>
        <element ref="m3w:DigestMethods"/>
        <element ref="m3w:KeyAgreementMethods" minOccurs="0"/>
    </sequence>
30

```

Table 4

```

</complexType>
  <element name="EncryptedDataEncodingTypes"
type="m3w:EncryptedDataEncodingTypesType"/>
5   <complexType name="EncryptedDataEncodingTypesType" >
      <sequence>
          <element ref="m3w:
EncryptedDataEncodingType" maxOccurs="unbounded"/>
      </sequence>
10  </complexType>
      <element ref="EncryptedDataEncodingType" type="m3w:MethodType"/>
      <element
ref="EncryptionMethods" type="m3w:EncryptionMethodsType"/>
      <complexType name="EncryptionMethodsType" >
15    <sequence>
          <element ref="m3w: Method" maxOccurs="unbounded"/>
      </sequence>
      </complexType>
      <element
20 ref="m3w:KeyAgreementMethods" type="m3w:KeyAgreementMethodsType"/>
      <complexType name="m3w:KeyAgreementMethodsType">
          <sequence>
              <element ref="m3w: Method" maxOccurs="unbounded"/>
          </sequence>
25  </complexType>

```

```
</schema>
```

Table 5 to Table 7 below illustrate an exemplary list of the security data having the metadata structure of FIG. 11 defined in the present invention, with
30 XML description.

Table 5

```

<?xml version="1.0" encoding="UTF-8"?>
<SupportedSecurityEnvironment
xmlns="http://mccb.icu.ac.kr/M3WSecManProfile"
35 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

```


- 23 -

```
xsi:schemaLocation=="http://mccb.icu.ac.kr/M3WSecManProfile  
M3WSecManProfile.xsd">
```

```
<DataTransferProtocols>
```

```
<TransferProtocol algorithm="http" preference="1"/>
```

5

```
</DataTransferProtocols>
```

```
<AuthenticationProtocols>
```

```
<AuthenticationProtocols
```

```
algorithm="myAuthenticationMethod" preference="1"/>
```

```
</AuthenticationProtocols>
```

10

Table 6

```

<SignatureTypes>
  <CanonicalizationMethods>
    <Method algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
5 20010315" preference="2"/>
    <Method algorithm="http://www.w3.org/TR/2001/10/xml-exc14n#"
preference="1"/>
  </CanonicalizationMethods>
  <SignatureMethods>
10  <Method algorithm="http://www.w3.org/2000/09/xmlsig#dsa-sha 1"
preference="2"/>
    <Method algorithm="http://www.w3.org/2000/09/xmlsig#rsa-sha 1"
preference="1"/>
  </SignatureMethods>
15  <TransformMethods>
    <Method algorithm="http://www.w3.org/TR/1999/REC-xslt-
19991116" preference="1"/>
    <Method algorithm="http://www.w3.org/TR/1999/REC-xpath-
19991116" preference="1"/>
20  </TransformMethods>
  <DigestMethods>
    <Method algorithm="http://www.w3.org/2000/09/xmlsig#sha 1"
preference="1"/>
  </DigestMethods>
25  <SignatureTypes>

```

Table 7

```

<EncryptionTypes>
  <EncryptedDataEncodingTypes>
    <EncryptedDataEncodingType
30 algorithm="http://www.w3.org/2000/09/xmlsig#base64" preference="1"/>
  </EncryptedDataEncodingTypes>
  <EncryptionMethods>
    <Method
algorithm="http://www.w3.org/2001/04/xmlenc#triptedes-cbc" preference="1"/>
35  <Method algorithm="http://www.w3.org/2001/04/xmlenc#aes128-

```

- 25 -

```

cbc" preference="2"/>
      <Method algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"
preference="3"/>
      <Method algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-
5 mgf1p" preference="4"/>
    </EncryptionMethods>
    <DigestMethods>
      <Method algorithm="http://www.w3.org/2000/09/xmlsig#sha 1"
preference="1"/>
10    <KeyAgreementMethods>
      <Method algorithm="http://www.w3.org/2001/04/xmlenc#dh"
preference="1"/>
    </KeyAgreementMethods>
  </EncryptionTypes>
15 </SupportedSecurityEnvironment>

```

The program or algorithm that provides open middleware enabling efficient management of the metadata structure for the multimedia middleware service of the present invention can be stored in a storage medium that can be installed or mounted in a particular apparatus, and the data in the storage medium can be read using an apparatus such as a computer.

With reference to FIGs. 16 and 17, a description will now be made of an operation of performing Governed Access through License List management during internal service or internal resource access in the middleware structure of the present invention. To this end, in addition to the security management for communication with an external entity, the security manager 1135 of FIG. 2 not only takes charge of security maintenance and management for an internal service through a security procedure when an application program executed in an upper layer accesses an internal service, i.e. middleware service, through the service application processing block 111, but also takes charge of security maintenance and management of internal resources through the security procedure when the middleware service accesses the platform block 115 including the internal resources.

35

The internal service, which is a kind of middleware program provided by the middleware needed by a user application, is a software concept, and the internal resource is a hardware concept such as a memory of a platform needed by the middleware while performing a service.

5

FIG. 16 is a flowchart illustrating a security process during access to an internal service in a multimedia middleware apparatus according to the present invention.

10

Referring to FIG. 16, the security manager 1135 takes charge of internal security maintenance and management through a security procedure when the service application processing block 111 for processing a multimedia service requests a particular service in middleware. Therefore, upon receipt of a request for a multimedia service from a user, the service application processing block 111 sends a request for execution of a new service to the service manager 1133 in step 1601. In order to check an access right for the corresponding service, the service manager 1133 sends a check request for the access right to the security manager 1135 in step 1603. The security manager 1135 checks in step 1605 whether the service application processing block 111 has an access right for the corresponding service by checking a license list in which an access right is expressed, and sends a response including the result to the service manager 1133 in step 1607. The service manager 1133 initializes the corresponding service in step 1609 according to the response result provided in step 1607 from the security manager 1135, and transfers a handler for the corresponding service to the service application processing block 111 in step 1611. Through this process, the service manager 1133 can check whether the requested service can be processed in the corresponding application program.

15

20

25

30

35

Therefore, the security manager 1135 should have a license list for an access right according to the present invention, and can check an access right for the requested service and provide a response to the result to the service manager 1133. In addition, the service manager 1133 inquires of the security manager 1135 about the access right upon receipt of every request for a particular service from the service application processing block 111, and then can or cannot provide the service according to the result.

- 27 -

FIG. 17 is a flowchart illustrating a security process during access to an internal resource in a multimedia middleware apparatus according to of the present invention.

5 Referring to FIG. 17, for access to a particular platform, a particular service first sends a request for a new resource access right to the security manager 1135 in step 1701. Then the security manager 1135 generates a new access ID in step 1703 by checking an access policy list, and provides information indicating the issuance of the new access ID to the RRE 1134 in step 10 1705. The RRE 1134 checks the access request in step 1707 by checking the license list in which the access right is expressed. In addition, the security manager 1135, after providing information indicating the issuance of the new access ID to the RRE 1134 in step 1705, provides the new access ID to the service in step 1709. The service receiving the new access ID sends an access 15 request for a particular platform resource to the RRE 1134 in step 1711. Then the RRE 1134 provides a response to the access request in step 1713. The license list to which the security manager 1135 makes reference in FIG. 16 can be the same as the license list to which the RRE 1134 makes reference in FIG. 17.

20 FIG. 18 is a diagram illustrating the overall data structure for access right expression according to of the present invention. With reference to FIG. 18, a description will now be made of the overall data structure for access right expression according to of the present invention.

25 Referring to FIG. 18, the overall structure of an access right (AccessRight) 1801 is shown. A lower layer of the access right (AccessRight) 1801 includes a principal (m3w-r:Principal) 1811 indicating an access principal (Principal) for storage of the multimedia middleware, access types (m3w-r:AccessTypes) 1812 and 1813 of the multimedia middleware, an access object (m3w-r:AccessObject) 1814 of the multimedia middleware, and access conditions (m3w-r:Conditions) 1815 of the multimedia middleware. 30

A description of notations shown in an access right type (AccessRightType) 1810 is added to the bottom of FIG. 18. That is, the drawing 35 shows the case where a non-displayed sub-folder(s) is included or a plurality of

sub-folders are included, and the case where the sub-folders are displayed partially or fully. In addition, a line connected to the access right (AccessRight) 1801 is shown together with a notation indicating the presence of a plurality of rights.

5

FIG. 19 is a diagram illustrating a data structure for expression of an access principal (Principal) according to the present invention.

10 An access-granted principal is specified in the data structure for the access principal (Principal), i.e. principal type (PrincipalType) 1910, shown in FIG. 19. The Principal can be the application program or the service described in FIGs. 16 and 17. The data structure for expressing the Principal includes an ID (Identifier) 1911 for the Principal, and granted key information (KeyHolder) 1912 generated from an electronic signature method.

15

FIG. 20 is a diagram illustrating a data structure for expressing access types (AccessType) according to the present invention.

20 Referring to FIG. 20, reference numerals 2001 and 2002 specify access types granted to the Principal. The access type can be granted one type or a plurality of types. That is, in an access type denoted by reference numeral 2003, there is shown that a plurality of types can be granted to the service application processing block 111. For example, each type can have Open, Write, Read, and Execute values.

25

FIG. 21 is a diagram illustrating a data structure for expressing an access object (AccessObject) according to the present invention.

30 Referring to FIG. 21, an access object granted to the Principal is specified. An access object (AccessObject) 2101 can be a service or a platform resource. The data for expressing the access object includes an ID (Identifier) 2111 of the access object, granted key information (KeyHolder) 2112 generated from an electronic signature method, and platform resources (PlatformResources) 2113. In FIG. 21, the platform resources (PlatformResources) 2113 can be replaced with actual resources such as a file system (FileSystem) 2114, Memory 2115, and
35 Socket 2116.

FIG. 22 is a diagram illustrating a data structure for expressing access conditions (Conditions) according to the present invention.

5 Referring to FIG. 22, Conditions 2201 on which a granted Principal can access the access object is specified. A detailed example of the Conditions is shown by reference numeral 2210. A Condition 2211 can be replaced with Size 2213 that specifies accessible file system and memory size in bytes, or with an access count (AccessCount) 2212 that specifies an accessible count.

10

FIG. 23 is a diagram illustrating the overall data structure of an access right expression group according to the present invention.

15 Referring to FIG. 23, there is shown a data structure for expressing an access right to a plurality of access objects that one Principal can have, or for expressing a group of a plurality of access rights that one access object can have. An access right group (AccessRightGroup) 2301 has one of PrincipalGroup 2311 and AccessObjectGroup 2321, and AccessRights 2331 described in FIG. 18. That is, the AccessRightGroup 2301, if it has the PrincipalGroup 2311, has a value of AccessObject (see reference numeral 1814 of FIG. 18) but does not have Principal (see reference numeral 1811 of FIG. 18) among the access right expressions. However, if the AccessRightGroup 2301 has the AccessObjectGroup 2321, it has a value of the Principal but does not express the AccessObject among the access right expressions. In Fig.23, detailed description with regard to
20 PrincipalType 2310, ObjectType 2320 and AccessRightType 2330 are the same as Fig. 19, Fig. 21 and Fig. 18. Therefore, a description is omitted.

25

Table 8 to Table 11 below illustrate the overall structure of access right expression data of FIG. 18 defined in the present invention, with XML description.

Table 8

```

<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://mccb.icu.ac.kr/M3WAccessRightExpression"
xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:m3w-r="http://mccb.icu.ac.kr/M3WAccessRightExpression"
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <import namespace="http://www.w3.org/2000/09/xmldsig#"
schemaLocation="dsig.xsd"/>
  <element name="AccessRight" type="m3w-r:AccessRightType"/>
  <complexType name="AccessRightType">
    <sequence>
      <element ref="m3w-r:Principal" minOccurs="0"/>
      <choice>
        <element ref="m3w-r:AccessType"/>
        <element ref="m3w-r:AccessTypes"/>
      </choice>
      <element ref="m3w-r:AccessObject" minOccurs="0"/>
      <element ref="m3w-r:Conditions" minOccurs="0"/>
    </sequence>
  </complexType>
  <element name="AccessRightGroup" type="m3w-r:AccessRightGroupType"/>
  <complexType name="AccessRightGroupType">
    <sequence>
      <choice>
        <element ref="m3w-r:PrincipalGroup"/>
        <element ref="m3w-r:AccessObjectGroup"/>
      </choice>
      <element ref="m3w-r:AccessRights"/>
    </sequence>
  </complexType>
  <element name="AccessRights" type="m3w-r:AccessRightsType"/>
  <complexType name="AccessRightsType">
    <sequence>
      <element ref="m3w-r:AccessRight" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
  <element name="AccessTypes" type="m3w-r:AccessTypesType"/>
  <complexType name="AccessTypesType">
    <sequence>
      <element ref="m3w-r:AccessType" maxOccurs="unbounded"/>
    </sequence>
  </complexType>

```


Table 9

<pre> </sequence> </complexType> <element name="AccessType" type="m3w-r:AccessTypeType"/> <simpleType name="AccessTypeType"> <restriction base="string"> <enumeration value="Open"/> <enumeration value="Write"/> <enumeration value="Read"/> <enumeration value="Execute"/> </restriction> </simpleType> <complexType name="ObjectType"> <sequence> <choice> <element name="Identifier" type="anyURI"/> <element ref="m3w-r:KeyHolder"/> <element ref="m3w-r:PlatformResource"/> </choice> </sequence> <attribute name="type" type="m3w-r:ObjectTypeType" use="required"/> </complexType> <simpleType name="ObjectTypeType"> <restriction base="string"> <enumeration value="Platform"/> <enumeration value="Service"/> <enumeration value="Application"/> </restriction> </simpleType> <element name="PlatformResource" type="m3w-r:PlatformResourceType"/> <complexType name="PlatformResourceType"> <element name="FileSystem" type="m3w-r:FileSystemType" substitutionGroup="m3w-r:PlatformResource"/> <complexType name="FileSystemType"> <complexContent> <extension base="m3w-r:PlatformResourceType"> <attribute name="location" type="anyURI" use="required"/> </extension> </complexContent> </complexType> </pre>

Table 10

<pre> <element name="Memory" type="m3w-r:MemoryType" substitutionGroup="m3w-r:PlatformResource"/> <complexType name="MemoryType"> <complexContent> <extension base="m3w-r:PlatformResourceType"> <attribute name="address" type="string" use="required"/> </extension> </complexContent> </complexType> <element name="Socket" type="m3w-r:SocketType" substitutionGroup="m3w-r:PlatformResource"/> <complexType name="SocketType"> <complexContent> <extension base="m3w-r:PlatformResourceType"> <attribute name="port" type="integer" use="required"/> </extension> </complexContent> </complexType> <element name="KeyHolder" type="m3w-r:KeyHolderType"/> <complexType name="KeyHolderType"> <sequence> <element name="info" type="dsig:KeyInfoType"/> </sequence> </complexType> <element name="Principal" type="m3w-r:PrincipalType"/> <complexType name="PrincipalType"> <complexContent> <restriction base="m3w-r:ObjectType"> <sequence> <choice> <element name="Identifier" type="anyURI"/> <element ref="m3w-r:KeyHolder"/> </choice> </sequence> </restriction> </complexContent> </complexType> </pre>
--

Table 11

```

<element name="AccessObject" type="m3w-r:ObjectType"/>
<element name="PrincipalGroup" type="m3w-r:PrincipalType"/>
<element name="AccessObjectGroup" type="m3w-r:ObjectType"/>
<element name="Conditions" type="m3w-r:ConditionsType"/>
<complexType name="ConditionsType">
  <sequence>
    <element ref="m3w-r:Condition" minOccurs="0"/>
  </sequence>
</complexType>
<element name="Condition" type="m3w-r:ConditionType"/>
<complexType name="ConditionType">
  <element name="Size" type="m3w-r:SizeType"
substitutionGroup="m3w-r:Condition"/>
  <complexType name="SizeType">
    <simpleContent>
      <extension base="integer">
        <attribute name="unit" type="m3w-r:UnitType"
use="required"/>
      </extension>
    </simpleContent>
  </complexType>
<simpleType name="UnitType">
  <restriction base="string">
    <enumeration value="byte"/>
    <enumeration value="kilobyte"/>
    <enumeration value="megabyte"/>
    <enumeration value="gigabyte"/>
  </restriction>
</simpleType>
<element name="AccessCount" type="m3w-r:AccessCountType" substitutionGroup="m3w-
r:Condition"/>
<complexType name="AccessCountType">
  <simpleContent>
    <extension base="integer"/>
  </simpleContent>
</complexType>
</schema>

```

5 The program or algorithm that provides a middleware structure capable of efficiently managing the access right expression structure for the multimedia middleware service can be stored in the storage medium that can be installed or mounted in a particular apparatus, and the data in the storage medium can be read using an apparatus such as a computer.

FIG. 24 is a diagram illustrating a basic structure of expressing a right using a license structure of Right Expression Language (REL) technology

according to the present invention.

The current REL technology, which is technology corresponding to MPEG-21 Part 5, uses a machine-readable right expression language capable of expressing various rights and granted details for the content using the terms defined in Right Data Dictionary (RDD) of MPEG-21 Part 6. The use of this language can provide a flexible and interoperable mechanism that can transparently and popularly use digital resources while protecting the distributed published digital artistic creations such as movies, music, electronic books, games and the like, and maintaining the rights, conditions and charges of the content. The REL is based on XrML 2.0 by ContentGuard Co., and a right expression language that expresses the rights and the usage rule related to the use and distribution of the content with the XML can be an example of the REL.

A Title 2411 of FIG. 24 is a field for specifying a title of license 2401, a grant 2412 and a grant group (grantGroup) 2413 are fields for specifying a user or a user group granted for use of the license 2401, an issuer 2414 is a field for specifying the original owner of the license 2401, and an encrypted license (encryptedLicense) 2415 is a data structure for specifying an encrypted license.

Table 12 below shows a definition of prefixes and namespaces corresponding thereto, used for extending the right expression for a multimedia middleware to the REL technology.

Table 12

Prefix	Name	Corresponding namespace
r	REL Core	urn:mpeg:mpeg21:2003:01-REL-R-NS
sx	REL Standard Extension	urn:mpeg:mpeg21:2003:01-REL-SX-NS
mx	REL Multimedia Extension	urn:mpeg:mpeg21:2003:01-REL-MX-NS
mdx	REL Middleware Extension	urn:mpeg:mpeg21:2003:01-REL-MDX-NS

A first prefix 'r' defined in Table 12 is for specifying core expression of the REL technology. A prefix 'sx' is for specifying extension technology for specifying rights other than the core expression. A prefix 'mx' is an extension prefix for specifying multimedia-related rights. A prefix 'mdx', which is a prefix newly added by the present invention, is an extension prefix for specifying middleware-related rights. The 'mdx' information can be provided to a terminal along with, for example, the service information (Service Profile) described in FIG. 5.

Table 13 and Table 14 below illustrate the overall structure and fields of the license FIG. 24, and an element with an extended namespace 'mdx' attached for the multimedia middleware in Table 13 and Table 14 is an element newly added to the current REL by the present invention.

Table 13

r:license	r:license includes the following sub-elements: r:title, r:grant or r:grantGroup, r:issuer. r:license can be encrypted into r:encryptedLicense and stored.
r:grant	It means each r:grant right expression.
R:grantGroup	It is a group of each r:grantGroup r:grant.
r:issuer	Original owner of license
r:encryptedLicense	Encrypted license
r:title	Title of license
r:grant	r:grant has the following sub-elements: R:principal, r:right, r:resource r:condition.
r:principal r:keyHolder mdx:identifier	r:principal means Principal and has r:keyholder and mdx:identifier. mdx:identifier, which is an element newly added to the conventional REL by the present invention, is an element

		for specifying an identifier of principal.
r:right mx:execute mx:install mx:modify mx:uninstall mdx:listenSocket mdx:open mdx:read mdx:write		r:right has multimedia extension rights mx such as execute, install, modify and uninstall, as rights given by license, and middleware extension rights mdx such as open, read and write are added thereto by the present invention.

Table 14

r:resource r:grant r:grantGroup r:principal r:keyHolder mdx:identifier r:propertyAbstract sx:name sx:dnsName sx:emailName mdx:platformResource	r:resource has values of r:grant, r:grantGroup, r:principal (r:keyHolder, mdx:identifier), r:propertyAbstract (sx:name, x:dnsName, sx:emailName). mdx:identifier, which is an element newly added to the conventional REL by the present invention, is an element for specifying an identifier of a resource. mdx:platformResource is an element for expressing a list of platform resources.
r:conditions r:allconditions r:validityInterval sx:exerciseLimit sx:territory mdx:accessAddress mdx:accessPort mdx:accessSize	r:Condition, which is a condition granted according to the present license, specifies each of granted access address, access port and access resource size.
r:encryptedGrant	Encrypted grant
r:grantGroup	It is a grant group and can have r:principal or r:resource, r:condition, r:grant as a sub-element.
r:encryptedGrantGroup	Encrypted grant group

5 Table 15 below shows a structure of an element for specifying a platform resource added to r:resource of Table 14, and a Description for specifying it in the XML.

Table 15

Diagram	platformResource			
Substitute Group	r:condition			
Attributes	Name	Type	Use	Semantics
	type	PlatformResourceTypeType	Required	The type of the platform's resource.
Source	<pre> <xsd:complexType name="PlatformResource"> <xsd:complexContent> <xsd:extension base="r:Resource"> <xsd:attribute name="type" type="mdx:PlatformResourceTypeType"/> </xsd:extension> </xsd:complexContent> </xsd:complexType> <xsd:simpleType name="PlatformResourceTypeType"/> <xsd:restriction base="xsd:string"> <xsd:enumeration value="filesystem"/> <xsd:enumeration value="memory"/> <xsd:enumeration value="socket"/> </xsd:restriction> </xsd:simpleType> <xsd:element name="platformResource" type="mdx:PlatformResource" substitutionGroup="r:resource"/> </pre>			

5 Table 16 below shows a structure of an ID (identifier) added to r:resource and r:principal of Table 14, and a Description for specifying it in the XML.

Table 16

Diagram	identifier
Substitute Group	r:principal

Children	-
Source	<xsd:element name="identifier" type="xsd:anyURI" substitutionGroup="r:principal"/>

Table 17 below shows a structure of an open right element added to r:right of Table 13, and a Description for specifying it in the XML.

5

Table 17

Diagram	open
Substitute Group	r:right
Children	-
Source	<pre> <xsd:complexType name="Open"> <xsd:complexContent> <xsd:extension base="r:Right"/> </xsd:complexContent> </xsd:complexType> <xsd:element name="open" type="mdx:Open" substitutionGroup="r:right"/> </pre>

Table 18 below shows a structure of a read right element added to r:right of Table 13, and a Description for specifying it in the XML.

10

Table 18

Diagram	read
Substitute Group	r:right
Children	-
Source	<pre> <xsd:complexType name="Read"> <xsd:complexContent> <xsd:extension base="r:Right"/> </xsd:complexContent> </pre>

	<pre> </xsd:complexType> <xsd:element name="read" type="mdx:Read" substitutionGroup="r:right"/> </pre>
--	--

Table 19 below shows a structure of a write right element added to r:right of Table 13, and a Description for specifying it in the XML.

5

Table 19

Diagram	write
Substitute Group	r:right
Children	-
Source	<pre> <xsd:complexType name="Write"> <xsd:complexContent> <xsd:extension base="r:Right"/> </xsd:complexContent> </xsd:complexType> <xsd:element name="write" type="mdx:Write" substitutionGroup="r:right"/> </pre>

Table 20 below shows a structure of a listen socket (ListenSocket) right element added to r:right of Table 13, and a Description for specifying it in the XML.

10

Table 20

Diagram	ListenSocket
Substitute Group	R:right
Children	-
Source	<pre> <xsd:complexType name="ListenSocket"> <xsd:complexContent> <xsd:extension base="r:Right"/> </xsd:complexContent> </pre>

	<pre> </xsd:complexType> <xsd:element name="red" type="mdx:Read" substitutionGroup="r:right"/> </pre>
--	---

5 Table 21 below shows a structure of an access address (accessAddress) condition element added to r:condition of Table 14, and a Description for specifying it in the XML.

Table 21

Diagram	accessAddress			
Substitute Group	r:condition			
Attributes	Name	Type	Use	Semantics
	location	anyURI	Required	Access-granted location, for example, "c:\\myfile\\temp.txt"
Source	<pre> <xsd:complexType name="AccessAddress"> <xsd:complexContent> <xsd:extension base="r:Condition"> <xsd:attribute name="location" type="xsd:anyURI" use="required"/> </xsd:extension> </xsd:complexContent> </xsd:complexType> <xsd:element name="accessAddress" type="mdx:AccessAddress" substitutionGroup="r:condition"/> </pre>			

10 Table 22 below shows a structure of an access port (accessPort) condition element added to r:condition of Table 14 and a Description for specifying it in the XML.

Table 22

Diagram	accessPort
Substitute	r:condition

Group				
Attributes	Name	Type	Use	Semantics
		number	integer	Required
Source	<pre> <xsd:complexType name="Accessport"> <xsd:complexContent> <xsd:extension base="r:Condition"> <xsd:attribute name="number" type="xsd:integer" use="required"/> </xsd:extension> </xsd:complexContent> </xsd:complexType> <xsd:element name="accessPort" type="mdx:AccessPort" substitutionGroup="r:condition"/> </pre>			

Table 23 below shows a structure of an access size (accessSize) condition element added to r:condition of Table 14, and a Description for specifying it in the XML.

5

Table 23

Diagram	accessSize
Substitute Group	r:condition
Source	<pre> <xsd:complexType name="AccessSize"> <xsd:simpleContent> <xsd:extension base="xsd:integer"> <xsd:simpleContent> </xsd:complexType> <xsd:element name="accessSize" type="mdx:AccessSize" substitutionGroup="r:condition"/> </pre>

10 The second aspect of the present invention can be easily compatible with the REL technology because it can add only the new prefixes and namespaces to the REL technology as a right expression language for middleware by extending

the current REL technology. Therefore, this embodiment can achieve both multimedia right expression and middleware right expression with one right expression language, so it can be efficiently used for the middleware for the multimedia service.

5

As can be understood from the foregoing description, in processing multimedia contents, an open multimedia terminal according to the present invention can simply perform security maintenance and management during interoperation between multimedia middleware apparatuses, or between a multimedia middleware apparatus and an external entity.

10

Further, in processing a multimedia service, a middleware apparatus in an open multimedia terminal can provide a standardized data structure for specifying a necessary security-related elements and methods.

15

In addition, a terminal provides a block for security management to multimedia middleware, and has interrelation with the block, thereby increasing security.

20

Moreover, a middleware structure can easily perform security management through license list management during internal service or internal resource access.

25

While the invention has been shown and described with reference to a certain preferred embodiment thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the spirit and scope of the invention as defined by the appended claims.

WHAT IS CLAIMED IS:

1. A security management method in multimedia middleware of a terminal that can download a multimedia service provided from an external entity, the method comprising:

if there is an access request for a particular multimedia service, determining whether there is a corresponding multimedia service;

if the access-requested multimedia service does not exist in middleware of the terminal, performing negotiation for secure session setup with the external entity;

setting up a secure session to the external entity using a security parameter selected as a result of the negotiation; and

receiving information for the download from the external entity through the secure session, and determining whether to execute the download.

2. The security management method of claim 1, wherein the security parameter is expressed as metadata.

3. The security management method of claim 1, wherein the selected security parameter comprises at least one security parameter selected by the external entity in a security parameter list transmitted from the terminal to the external entity.

4. The security management method of claim 1, wherein the security parameter for negotiation comprises at least one of Data Transfer Protocols, Authentication Protocols, Signature Types, and Encryption Types.

5. The security management method of claim 4, wherein the security parameter further comprises Uniform Resource Identifiers (URIs) for a corresponding protocol/algorithm and Preferences for the multimedia middleware.

6. The security management method of claim 4, wherein the security parameter comprises a list of protocols/algorithms supported by the multimedia middleware.

7. The security management method of claim 4, wherein the Signature Types comprises at least one of:

Canonicalization Methods indicating a list of canonicalization algorithms;

5 Signature Methods indicating a list of generation and verification algorithms;

Transform Methods indicating a list of transform algorithms; and

Digest Methods indicating a list of digest algorithms;

10 all of which are used for signature provided by the multimedia middleware.

8. The security management method of claim 4, wherein the Encryption Types comprises at least one of:

Encrypted Data Encoding Methods;

Encryption Methods;

15 Digest Methods; and

Key Agreement Methods;

all of which are supported by the multimedia middleware.

9. The security management method of claim 1, further comprising:
20 if the access-requested multimedia service does not exist before the negotiation, acquiring a network address of the external entity; and

25 providing a plurality of security parameters supported by the multimedia middleware to the external entity for connection of the secure session to the external entity.

10. The security management method of claim 1, wherein the determination of whether to execute the download further comprises:

receiving a Service Profile of the multimedia service desired to be downloaded from the external entity through the set secure session;

30 checking a Target Profile related to the Service Profile and a Security Profile related to the secure session;

determining to download the multimedia service based on the Service Profile, the Target Profile and the Security Profile; and

35 downloading the multimedia service through the secure session.

11. A security management apparatus in multimedia middleware of a terminal that can download a multimedia service provided from an external entity, the apparatus comprising:

5 a security manager for, if a requested particular multimedia service does not exist in middleware of the terminal, performing negotiation for secure session setup with the external entity, and setting up a secure session to the external entity using a security parameter selected as a result of the negotiation; and

10 a service manager for, if there is an access request for the particular multimedia service, determining whether there is the particular multimedia service, receiving information for the download from the external entity through the secure session, and determining whether to execute the download.

12. The security management apparatus of claim 11, wherein the selected security parameter comprises at least one security parameter selected by the external entity in a security parameter list transmitted from the terminal to the external entity.

13. The security management apparatus of claim 11, wherein the security parameter for negotiation comprises at least one of Data Transfer Protocols, Authentication Protocols, Signature Types, and Encryption Types.

14. The security management apparatus of claim 13, wherein the security parameter further comprises Uniform Resource Identifiers (URIs) for a corresponding protocol/algorithm and Preferences for the multimedia middleware.

15. The security management apparatus of claim 13, wherein the security parameter comprises a list of protocols/algorithms supported by the multimedia middleware.

16. The security management apparatus of claim 13, wherein the Signature Types comprises at least one of:

Canonicalization Methods indicating a list of canonicalization algorithms;

Signature Methods indicating a list of generation and verification algorithms;

35 Transform Methods indicating a list of transform algorithms; and

Digest Methods indicating a list of digest algorithms;
all of which are used for signature provided by the multimedia
middleware.

5 17. The security management apparatus of claim 13, wherein the
Encryption Types comprises at least one of:
 Encrypted Data Encoding Methods;
 Encryption Methods;
 Digest Methods; and
10 Key Agreement Methods;
 all of which are supported by the multimedia middleware.

 18. The security management apparatus of claim 11, wherein if the
particular multimedia service does not exist, the service manager acquires a
15 network address of the external entity from an M3W Run-time Environment
(RRE) in the terminal.

 19. The security management apparatus of claim 11, wherein the
security manager provides a plurality of security parameters supported by the
20 multimedia middleware to the external entity for connection of the secure session
to the external entity.

 20. The security management apparatus of claim 11, wherein the
service manager comprises:
25 receiving a Service Profile of the multimedia service desired to be
downloaded from the external entity through the set secure session;
 checking a Target Profile related to the Service Profile from the RRE in
the terminal;
 checking a Security Profile related to the secure session from the security
30 manager;
 determining whether to download the multimedia service based on the
Service Profile, the Target Profile and the Security Profile.

 21. A storage medium in which multimedia middleware capable of
35 downloading a multimedia service is stored, the storage medium comprising:

a security manager for, if a requested particular multimedia service does not exist in the multimedia middleware, performing negotiation for secure session setup with an external entity from which the multimedia service can be downloaded, and setting up a secure session to the external entity using a security parameter selected as a result of the negotiation; and

a service manager for, if there is an access request for the particular multimedia service, determining whether there is the particular multimedia service, receiving information for the download from the external entity through the secure session, and determining whether to execute the download.

22. The storage medium of claim 21, wherein the selected security parameter comprises at least one security parameter selected by the external entity in a security parameter list transmitted from the terminal to the external entity.

23. The storage medium of claim 21, wherein the security parameter for negotiation comprises at least one of Data Transfer Protocols, Authentication Protocols, Signature Types, and Encryption Types.

24. The storage medium of claim 23, wherein the security parameter further comprises Uniform Resource Identifiers (URIs) for a corresponding protocol/algorithm and Preferences for the multimedia middleware.

25. The storage medium of claim 23, wherein the security parameter comprises a list of protocols/algorithms supported by the multimedia middleware.

26. A security management method performed in multimedia middleware in a terminal, the method comprising:

if there is an access request to a particular multimedia service from a particular application program, checking a license list including its access right information; and

if the particular application program is granted to access to the particular multimedia service, delivering a handler for the particular multimedia service to the particular application program;

wherein the license list includes information indicating an access right of application programs for middleware services supportable by the terminal.

27. The security management method of claim 26, wherein the information indicating the access right is expressed as metadata, and comprises information indicating at least one of an access principal, an access type, an access object, and an access condition of the multimedia middleware.

5

28. The security management method of claim 27, wherein the access principal is the application program, and the information indicating the access principal comprises an identifier of the access principal and key information granted with an electronic signature method.

10

29. The security management method of claim 27, wherein the access principal is the application program, and the information indicating the access type indicates that at least one of Open, Write, Read, and Execute is granted.

15

30. The security management method of claim 27, wherein the access principal is the application program, and the information indicating the access object comprises an identifier of the multimedia service and key information granted with an electronic signature method.

20

31. The security management method of claim 27, wherein the access principal is the application program, and the information indicating the access condition comprises at least one of accessible file system, memory size, and access count.

25

32. A security management method performed in multimedia middleware in a terminal, the method comprising:

if there is an access request to a particular platform resource from a particular multimedia service, generating an access identifier thereof; and

30

checking a license list including access right information based on the access identifier; and

if the particular multimedia service is granted to access to the particular platform resource, delivering system resource to the particular multimedia service;

35

wherein the license list comprises information indicating an access right of multimedia services for platform resources supported by the terminal.

33. The security management method of claim 32, wherein the information indicating the access right is expressed as metadata, and comprises information indicating at least one of an access principal, an access type, an access object, and an access condition of the multimedia middleware.

5

34. The security management method of claim 33, wherein the access principal is the multimedia service, and the information indicating the access principal comprises an identifier of the access principal and key information granted with an electronic signature method.

10

35. The security management method of claim 33, wherein the access principal is the multimedia service, and the information indicating the access type indicates that at least one of Open, Write, Read, and Execute is granted.

15

36. The security management method of claim 33, wherein the access principal is the multimedia service, and the information indicating the access object comprises an identifier of the platform resource and key information granted with an electronic signature method.

20

37. The security management method of claim 33, wherein the access principal is the multimedia service, and the information indicating the access condition comprises at least one of accessible file system, memory size, and access count.

25

38. A security management apparatus included in multimedia middleware in a terminal, the apparatus comprising:

a service manager for receiving an access request to a particular multimedia service from a particular application program, and delivering a handler for the particular multimedia service to the particular application program, if the particular application program is granted to access to the particular multimedia service; and

30

a security manager for checking a license list including access right information for the particular application program in response to a request of the service manager;

35

wherein the license list comprises information indicating an access right

of application programs for middleware services supportable by the terminal.

39. The security management apparatus of claim 38, wherein the information indicating the access right is expressed as metadata, and comprises
5 information indicating at least one of an access principal, an access type, an access object, and an access condition of the multimedia middleware.

40. The security management apparatus of claim 39, wherein the access principal is the application program, and the information indicating the
10 access principal comprises an identifier of the access principal and key information granted with an electronic signature method.

41. The security management apparatus of claim 39, wherein the access principal is the application program, and the information indicating the
15 access type indicates that at least one of Open, Write, Read, and Execute is granted.

42. The security management apparatus of claim 39, wherein the access principal is the application program, and the information indicating the
20 access object comprises an identifier of the multimedia service and key information granted with an electronic signature method.

43. The security management apparatus of claim 39, wherein the access principal is the application program, and the information indicating the
25 access condition comprises at least one of accessible file system, memory size, and access count.

44. A security management apparatus included in multimedia
30 middleware in a terminal, the apparatus comprising:

a security manager for, if there is an access request to a particular platform resource from a particular multimedia service, generating an access
35 identifier thereof, and if the particular multimedia service is granted to access to the particular platform resource, delivering system resources to the particular multimedia service; and

an M3W Run-time Environment (RRE) for checking a license list

including access right information based on the access identifier in response to a request of the security manager;

wherein the license list comprises information indicating an access right of multimedia services for platform resources supported by the terminal.

5

45. The security management apparatus of claim 44, wherein the information indicating the access right is expressed as metadata, and comprises information indicating at least one of an access principal, an access type, an access object, and an access condition of the multimedia middleware.

10

46. The security management apparatus of claim 45, wherein the access principal is the multimedia service, and the information indicating the access principal comprises an identifier of the access principal and key information granted with an electronic signature method.

15

47. The security management apparatus of claim 45, wherein the access principal is the multimedia service, and the information indicating the access type indicates that at least one of Open, Write, Read, and Execute is granted.

20

48. The security management apparatus of claim 45, wherein the access principal is the multimedia service, and the information indicating the access object comprises an identifier of the platform resource and key information granted with an electronic signature method.

25

49. The security management apparatus of claim 45, wherein the access principal is the multimedia service, and the information indicating the access condition comprises at least one of accessible file system, memory size, and access count.

30

50. A storage medium in which multimedia middleware is stored, the storage medium comprising:

a service manager for receiving an access request to a particular multimedia service from a particular application program of a terminal, and delivering a handler for the particular multimedia service to the particular

35

application program, if the particular application program is granted to access to the particular multimedia service; and

a security manager for checking a license list including access right information for the particular application program in response to a request of the service manager;

wherein the license list comprises information indicating an access right to application programs for middleware services supportable by the terminal.

51. A storage medium in which multimedia middleware is stored, the storage medium comprising:

a security manager for, if there is an access request to a particular platform resource of a terminal from a particular multimedia service, generating an access identifier thereof, and if the particular multimedia service is granted to access to the particular platform resource, delivering system resources of the terminal to the particular multimedia service; and

an M3W Run-time Environment (RRE) for checking a license list including access right information based on the access identifier in response to a request of the security manager;

wherein the license list comprises information indicating an access right to multimedia services for platform resources supported by the terminal.

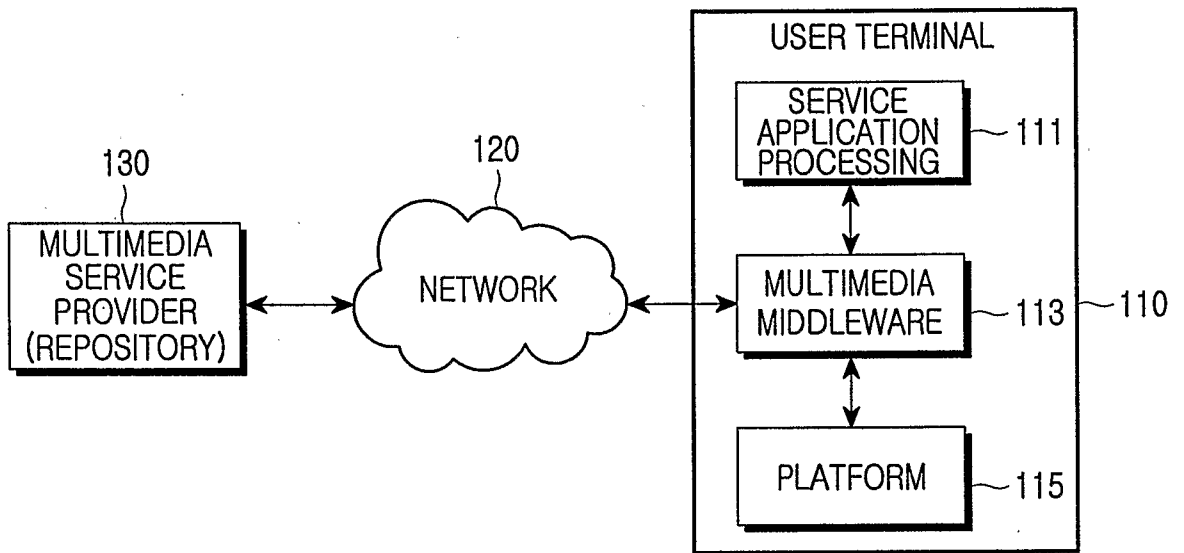


FIG.1

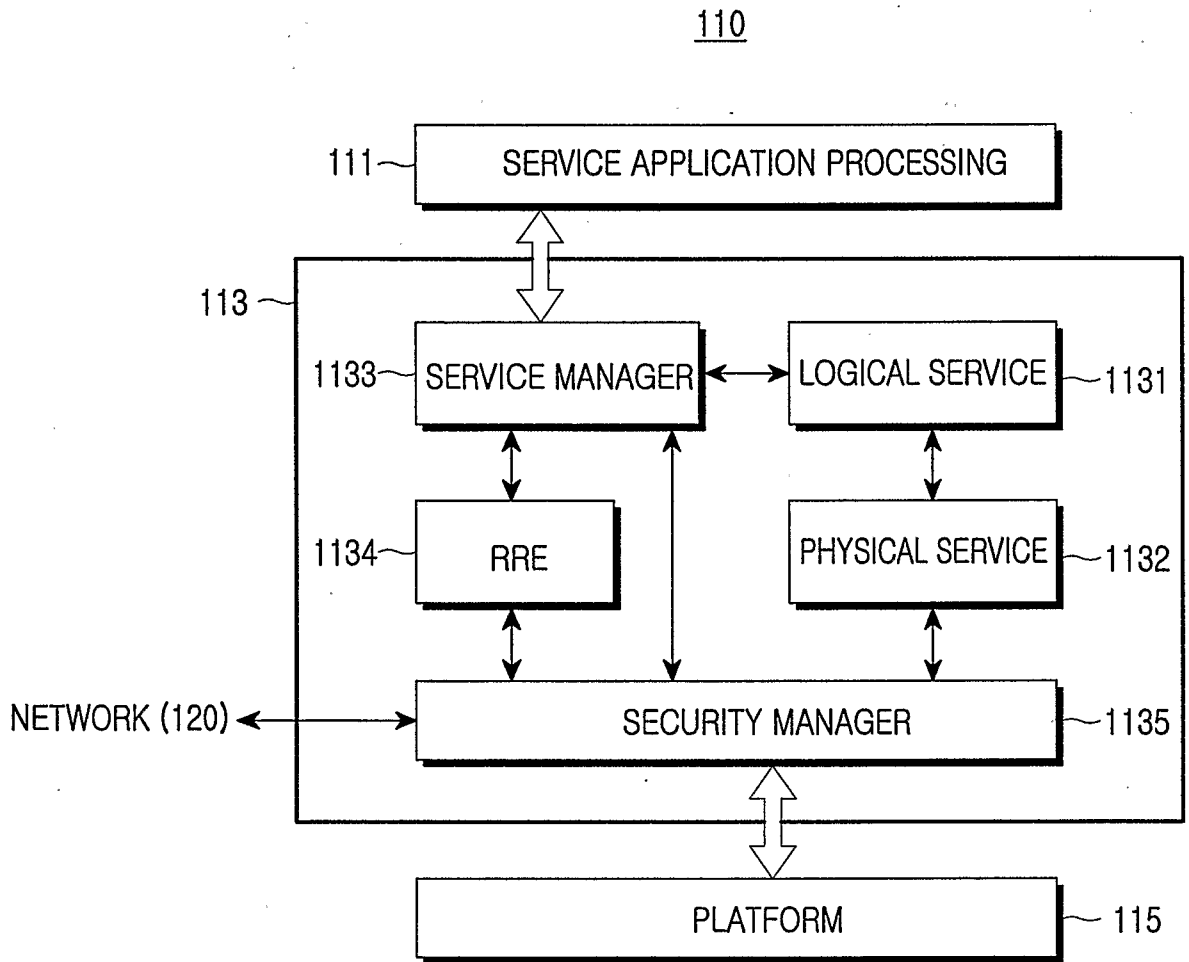


FIG.2

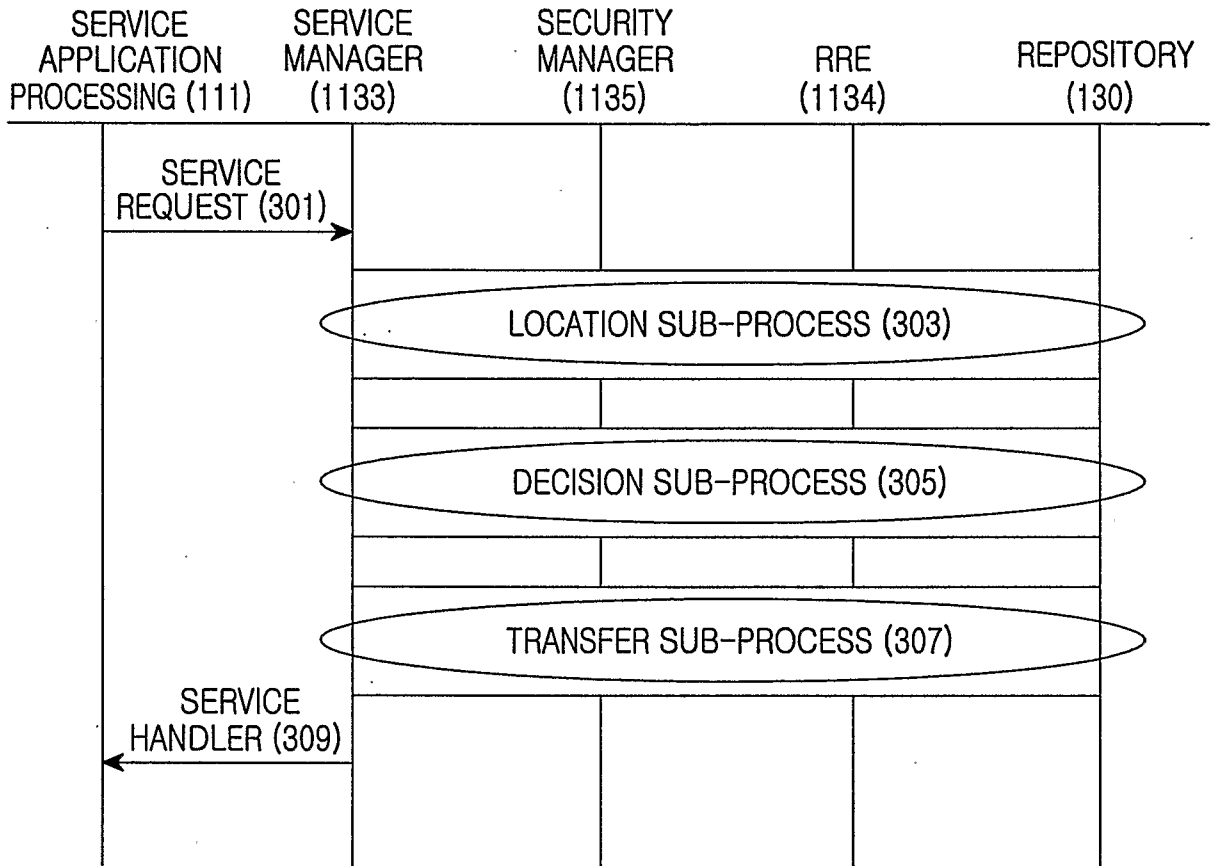


FIG.3

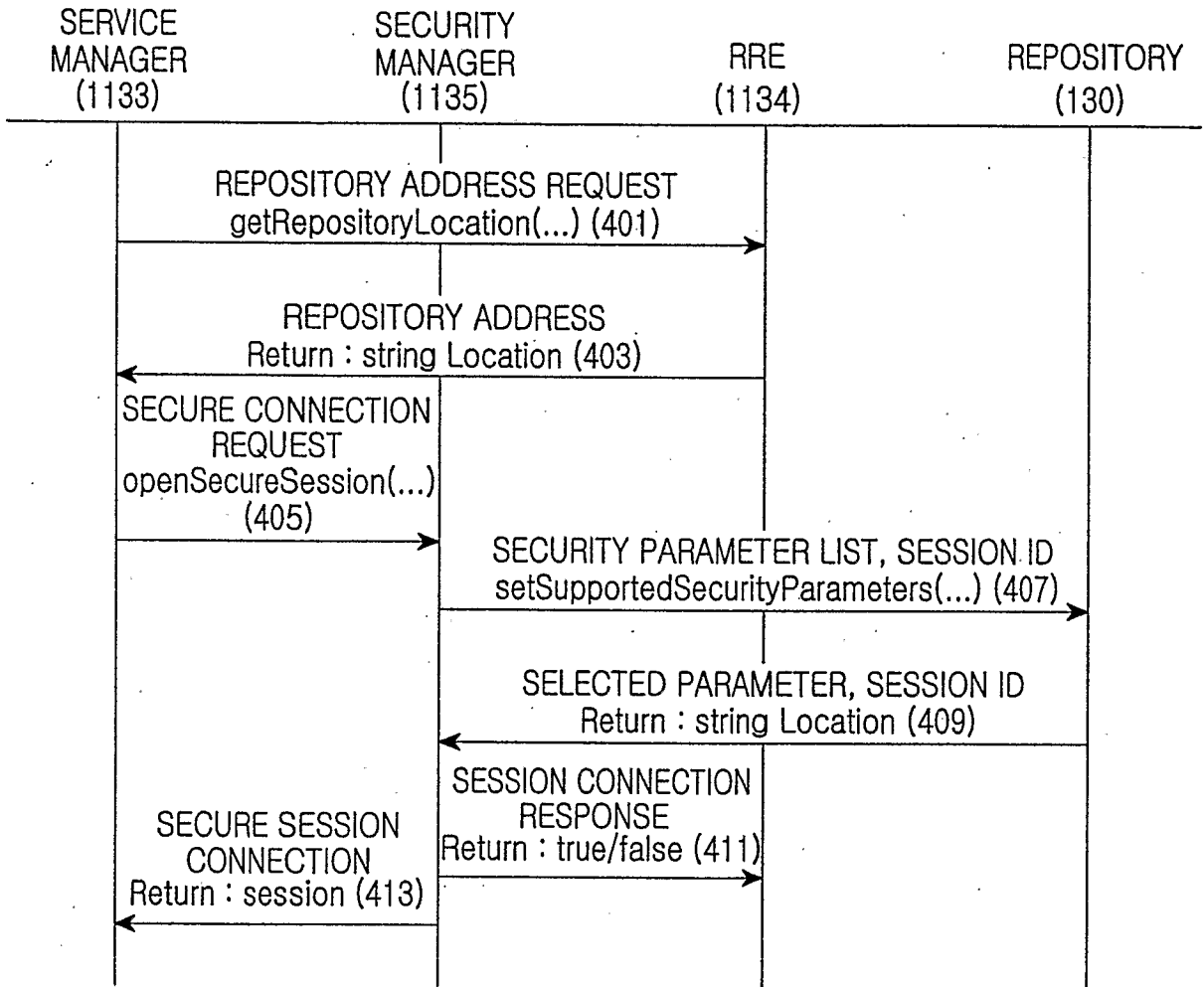


FIG.4

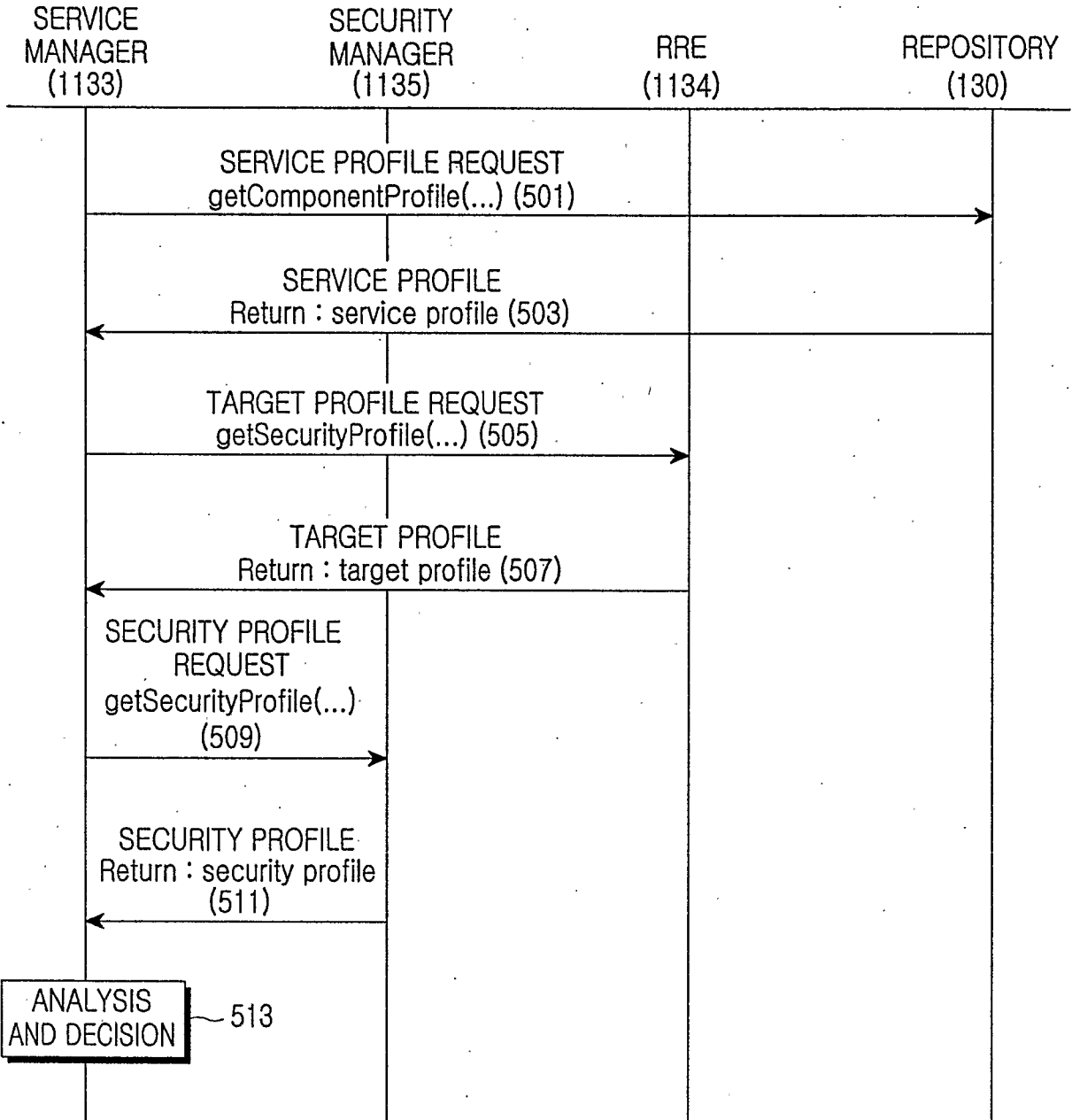


FIG.5

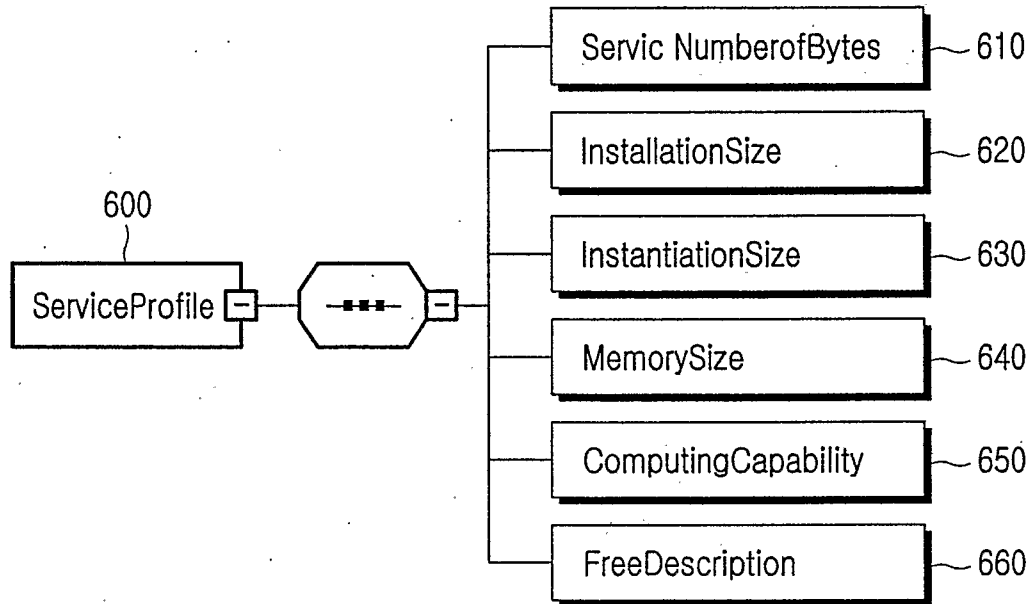


FIG.6

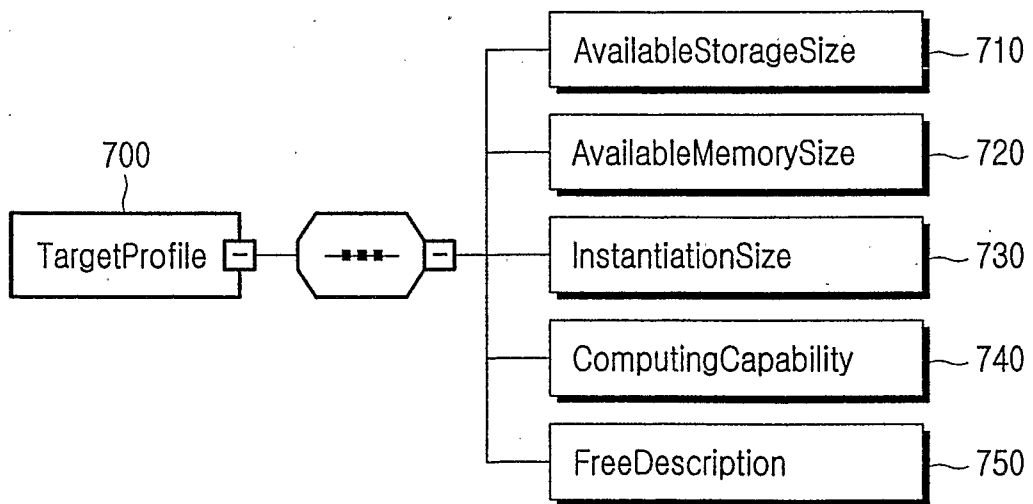


FIG.7

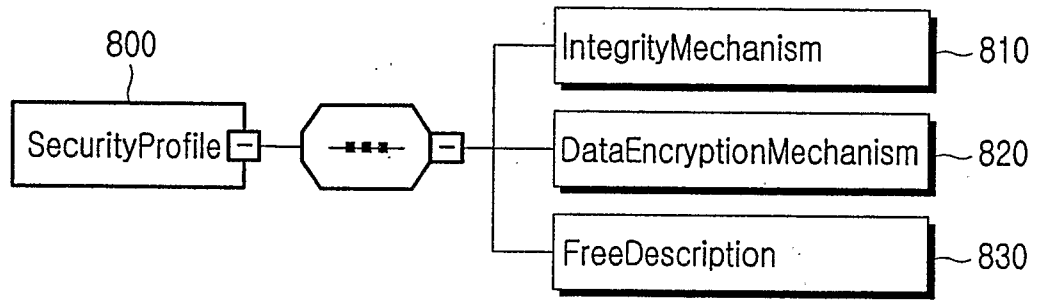


FIG. 8

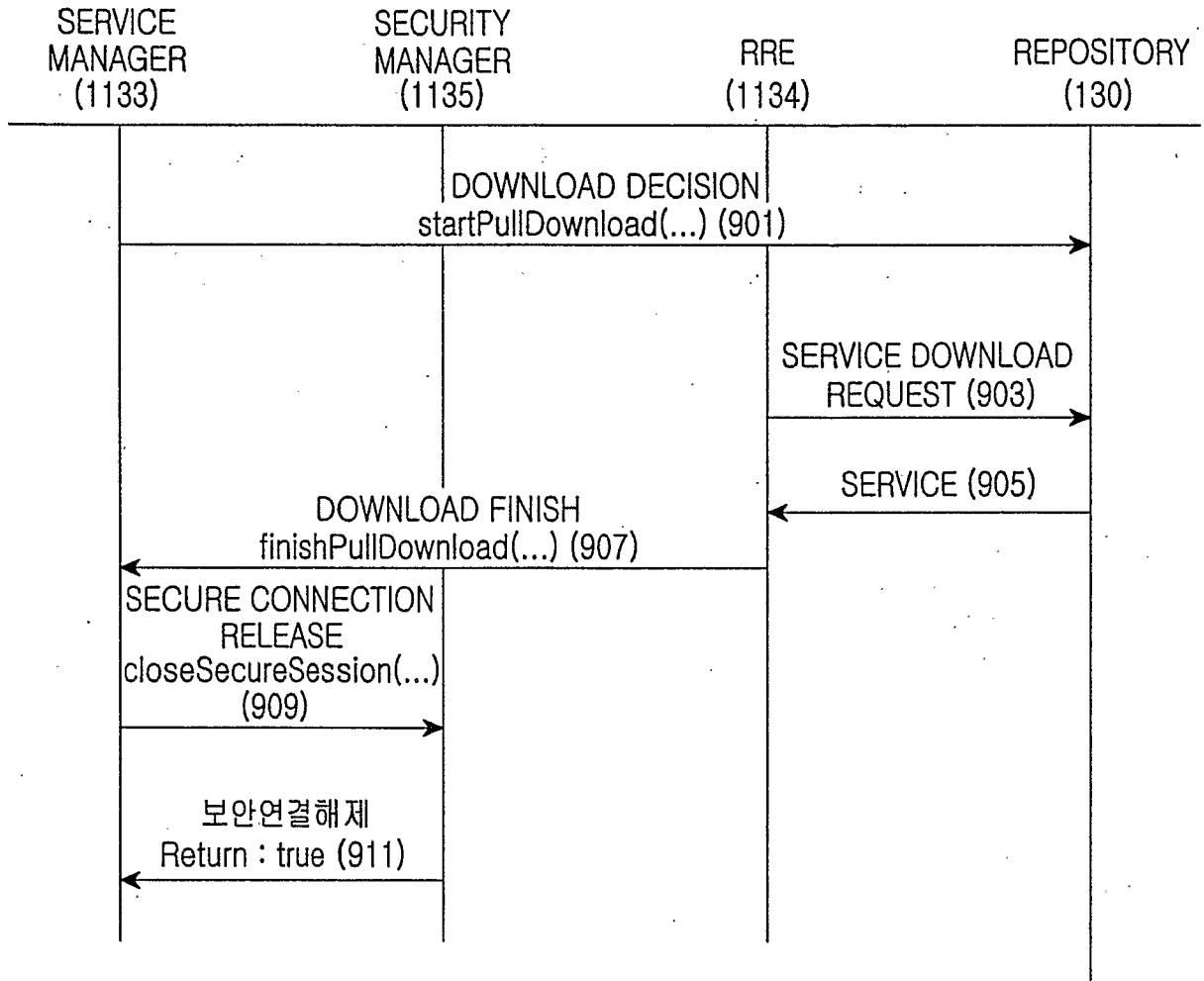


FIG.9

9/21

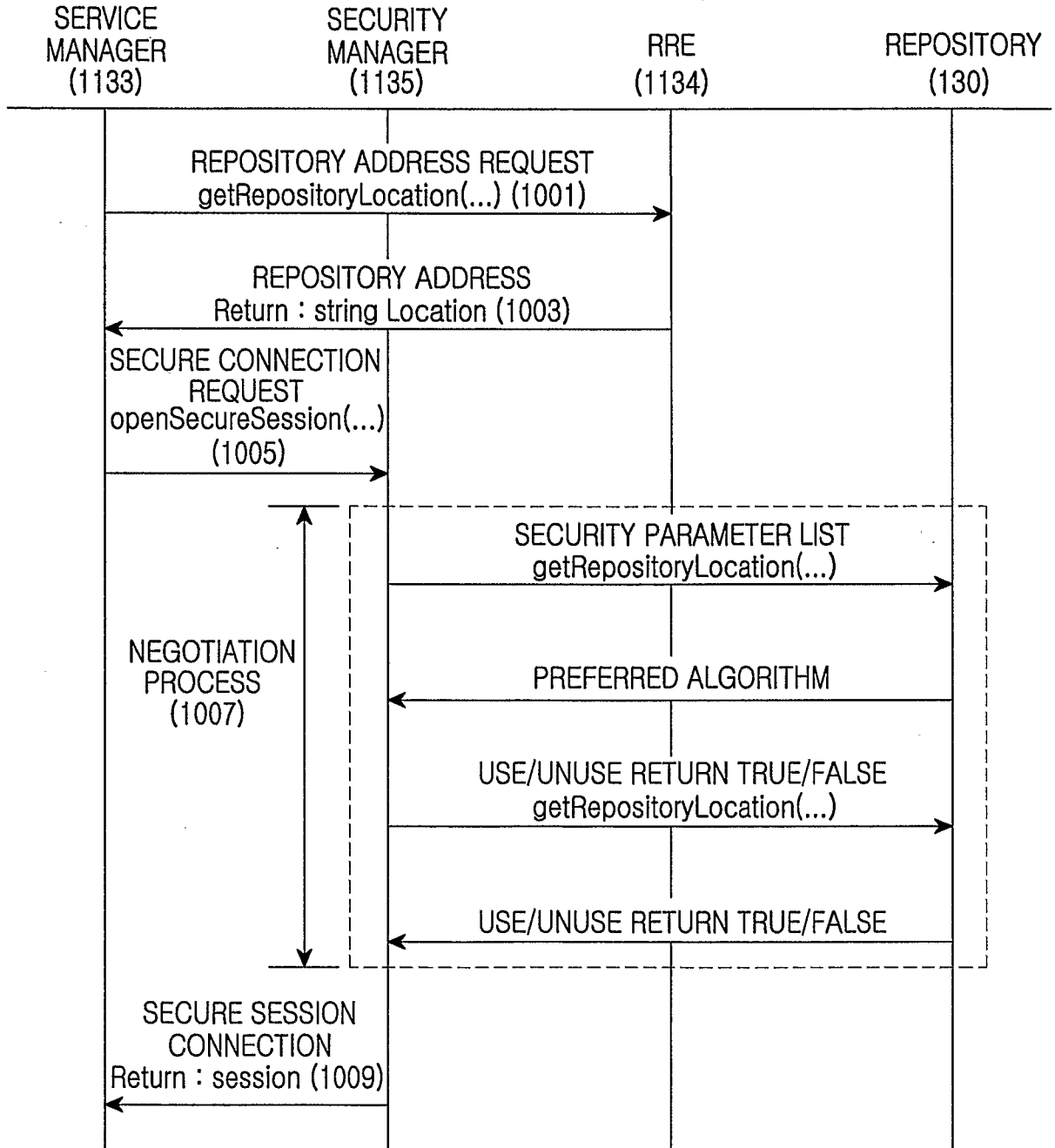


FIG.10

10/21

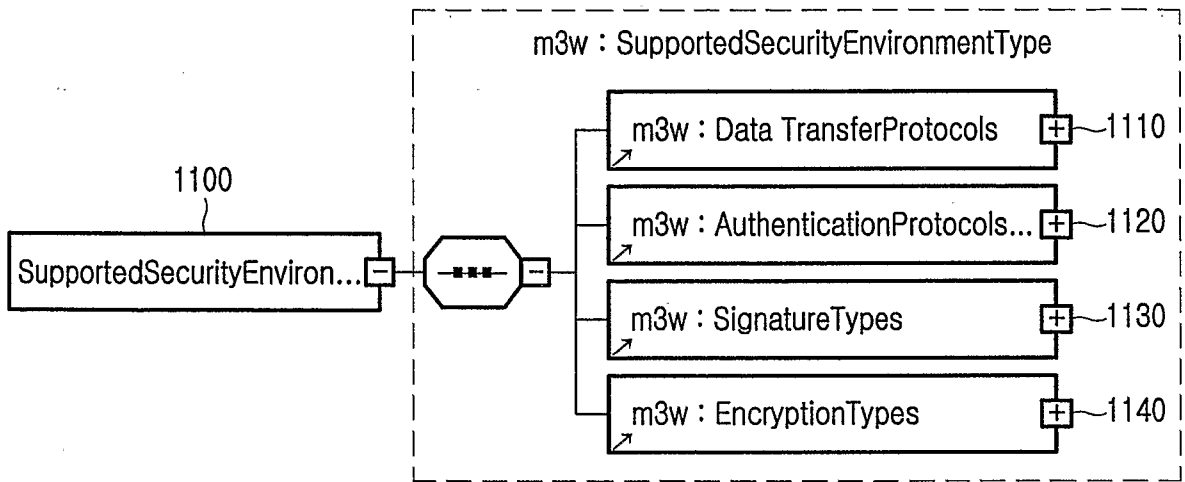


FIG. 11

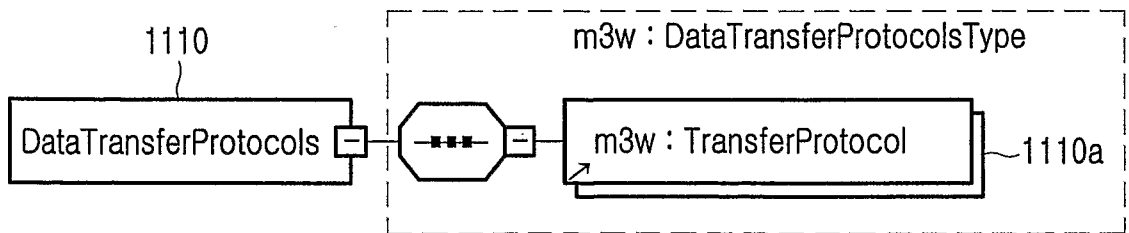


FIG. 12

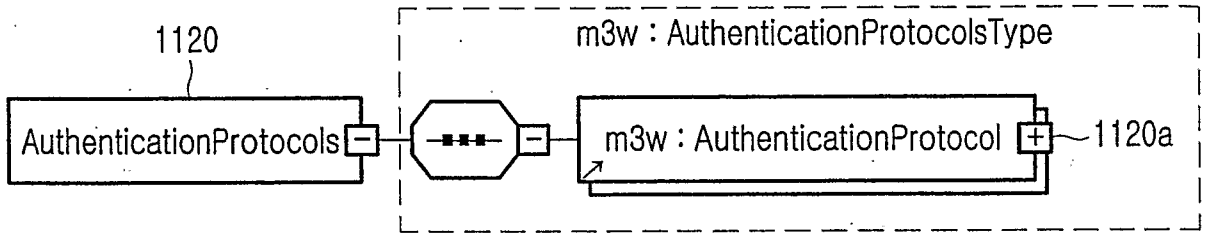


FIG.13

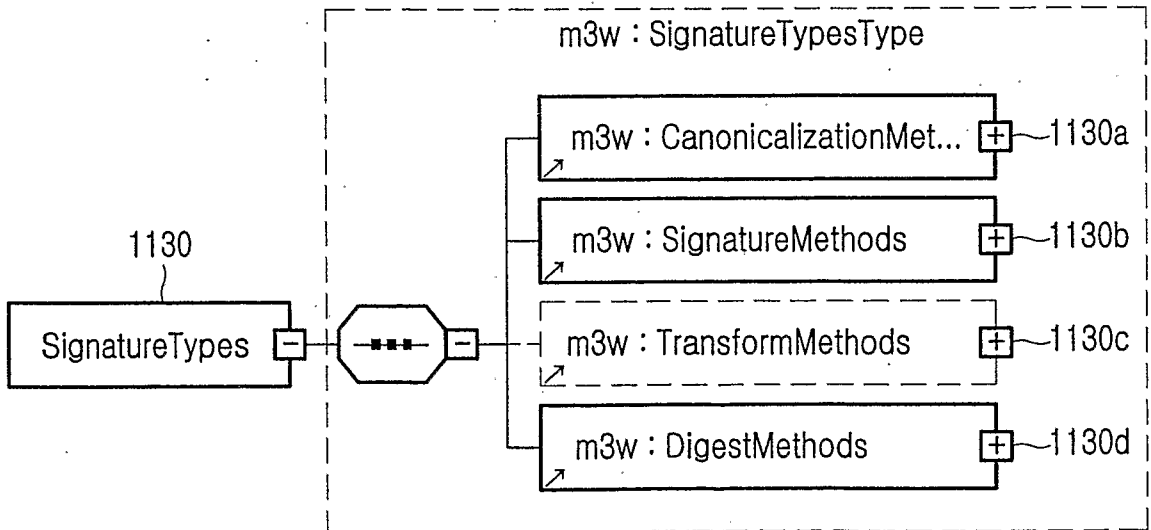


FIG.14

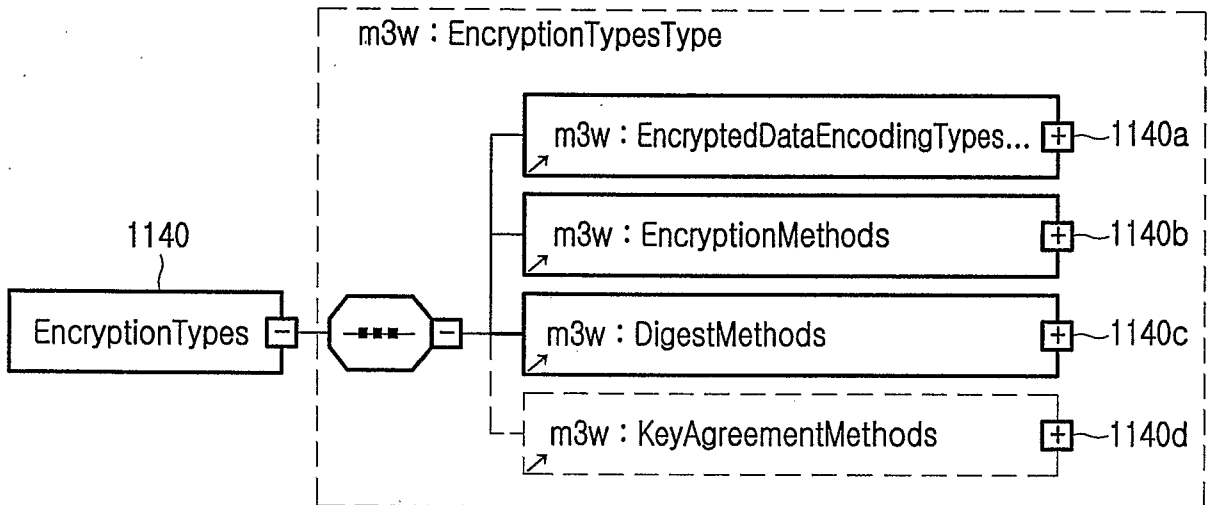


FIG.15

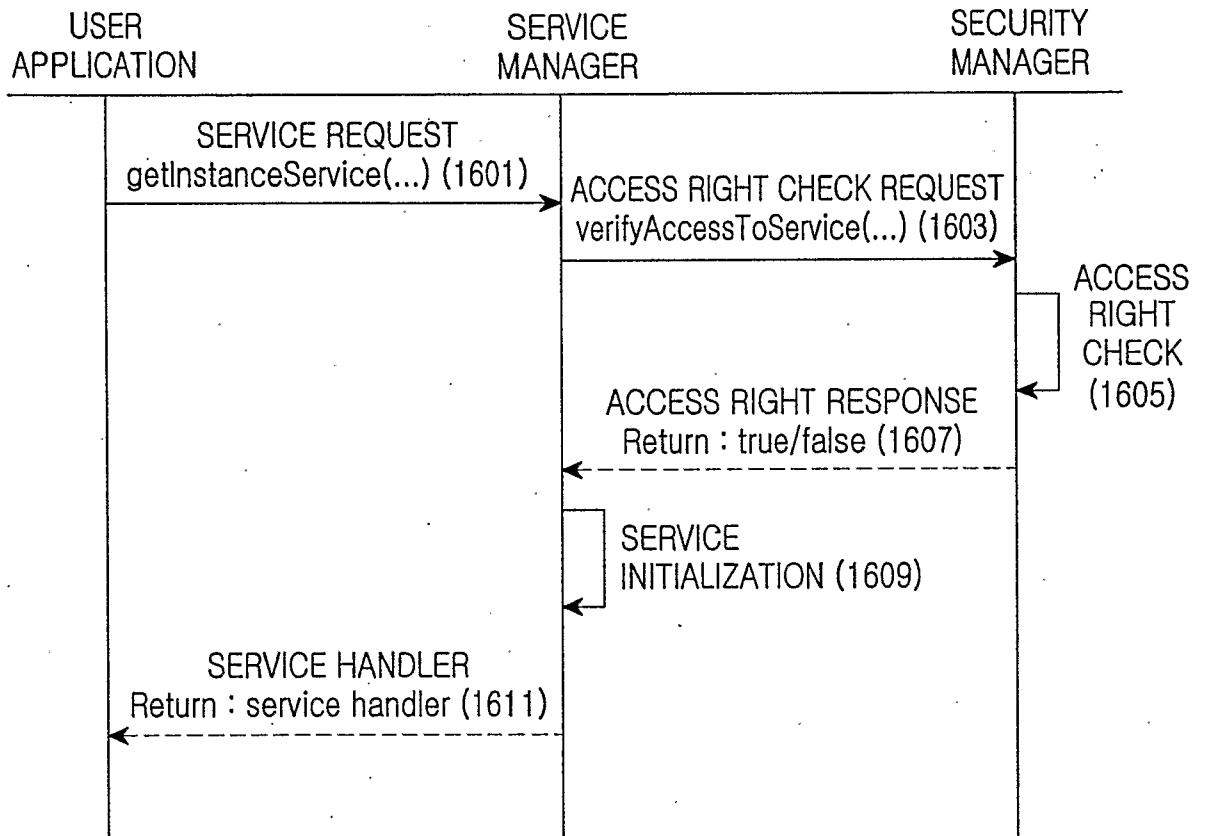


FIG.16

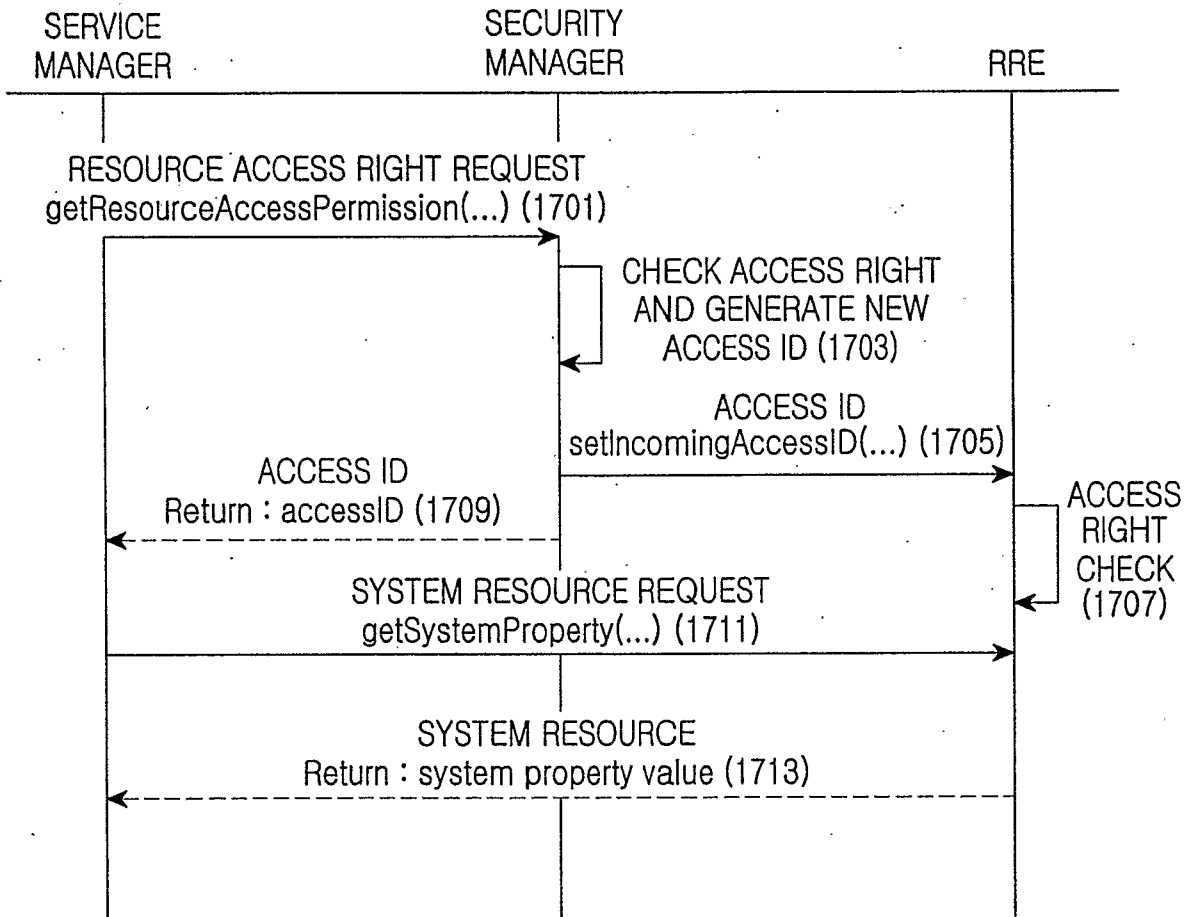


FIG.17

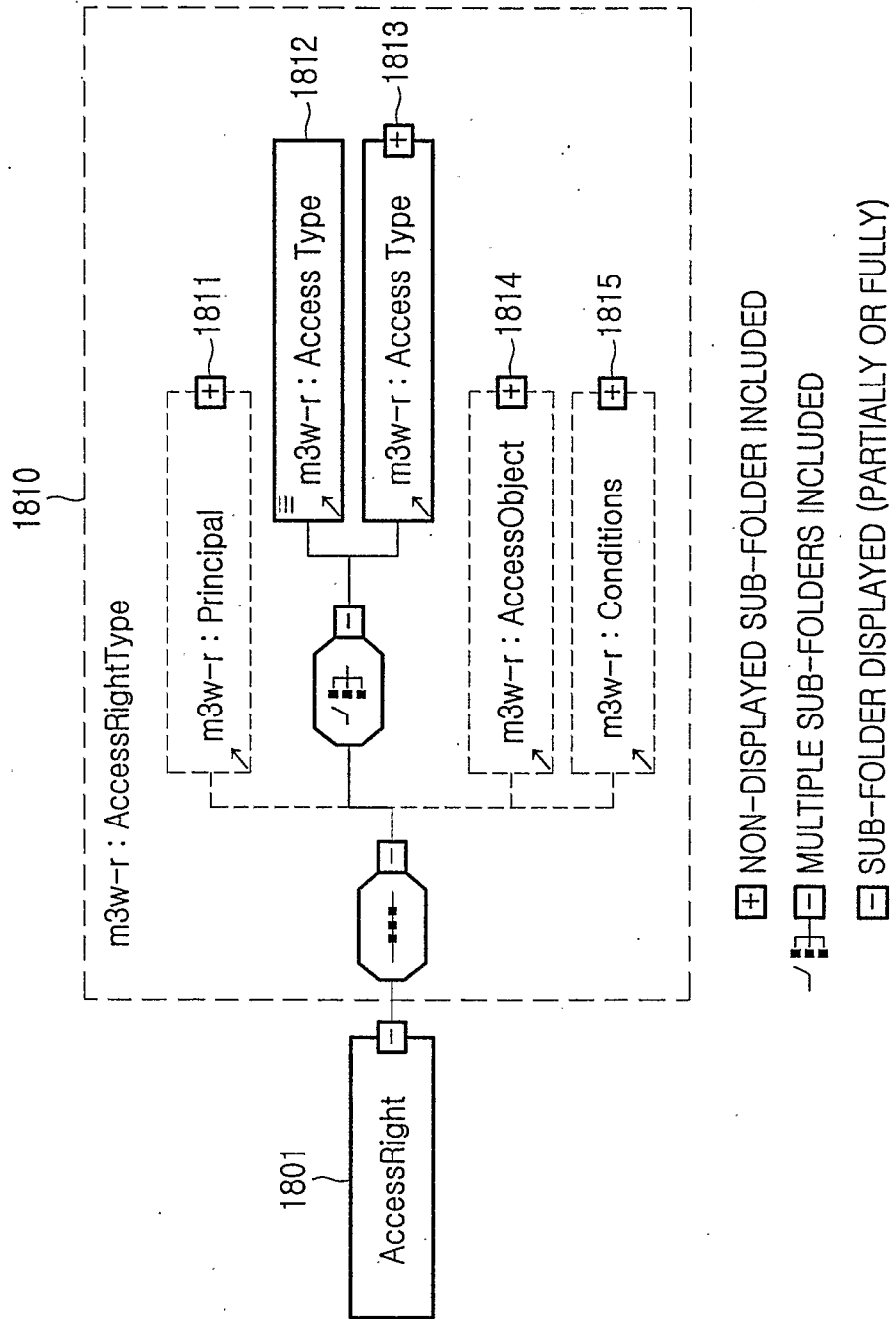
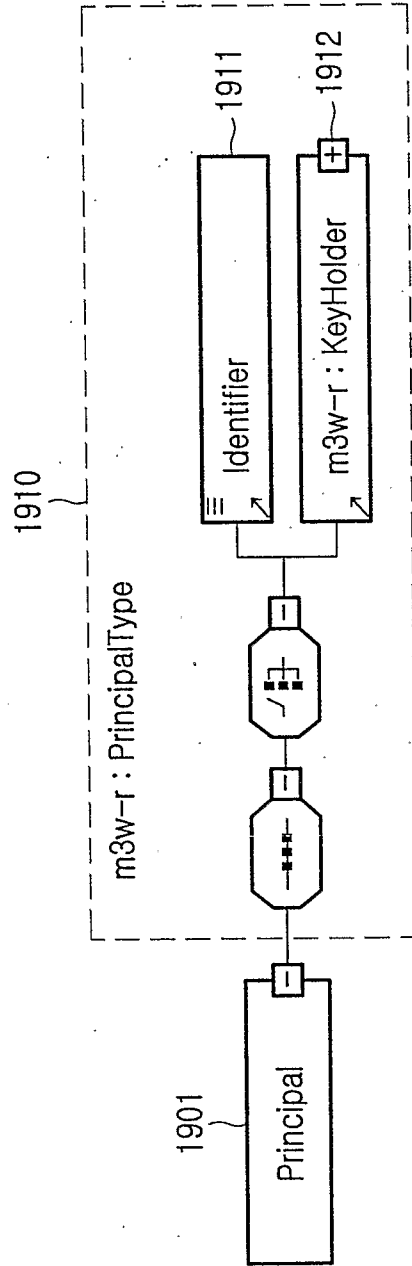


FIG.18



- ☐ NON-DISPLAYED SUB-FOLDER INCLUDED
- ☐☐ MULTIPLE SUB-FOLDERS INCLUDED
- ☐ SUB-FOLDER DISPLAYED (PARTIALLY OR FULLY)

FIG.19

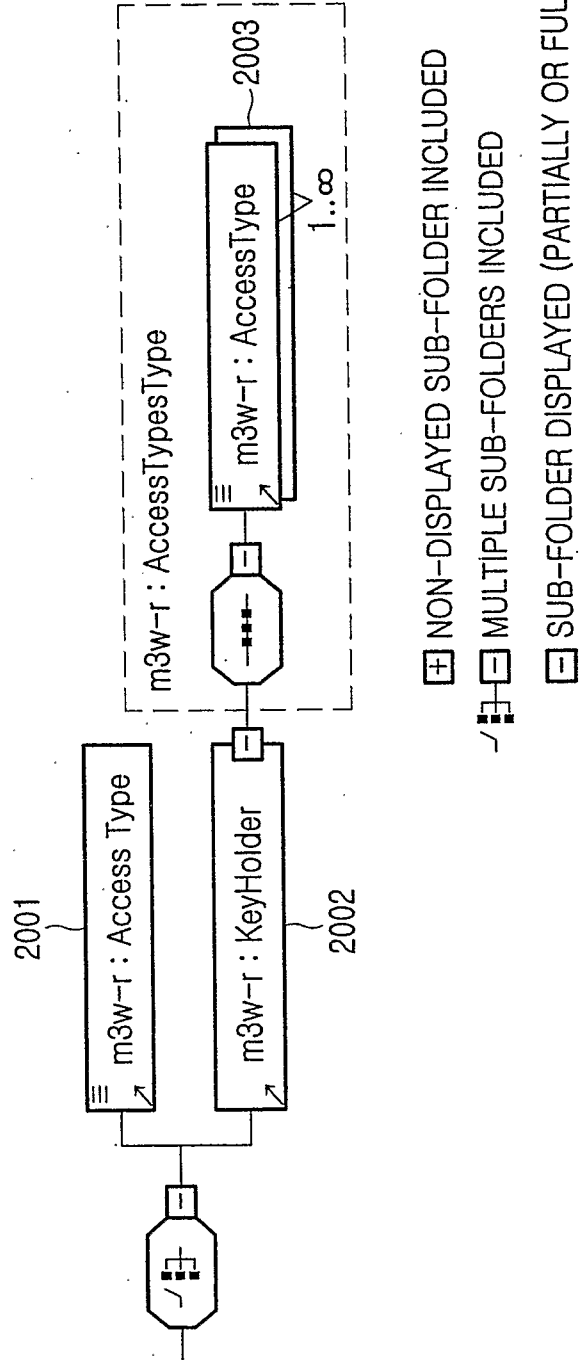


FIG.20

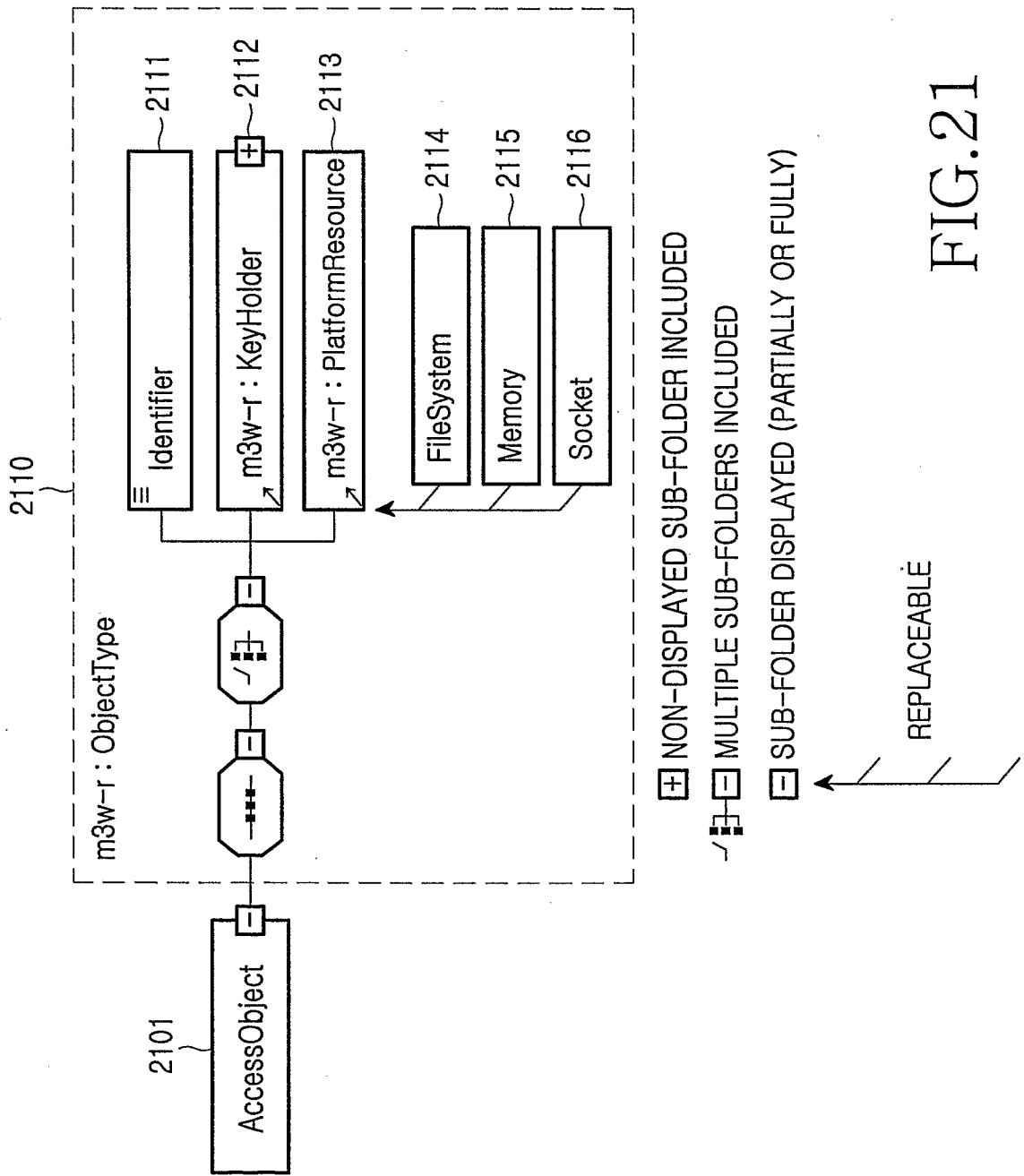


FIG.21

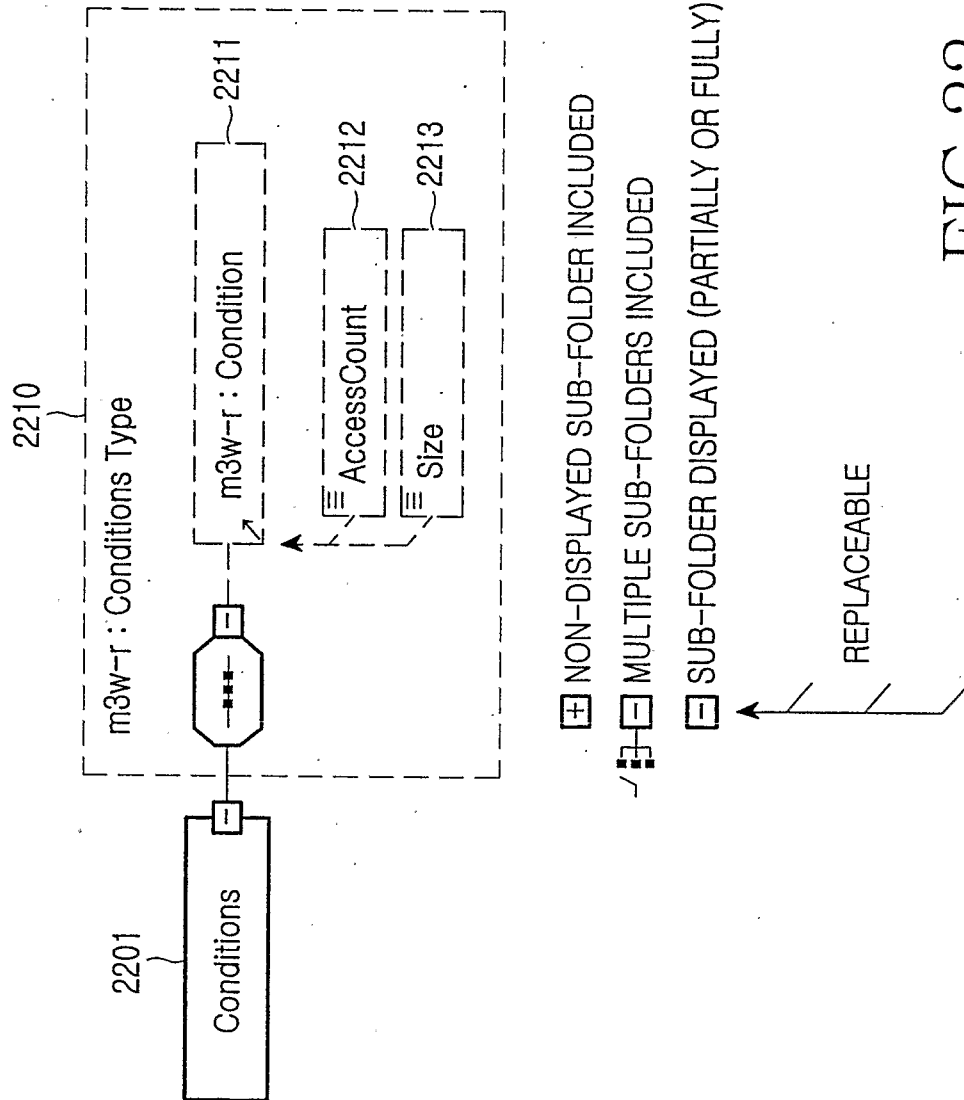


FIG.22

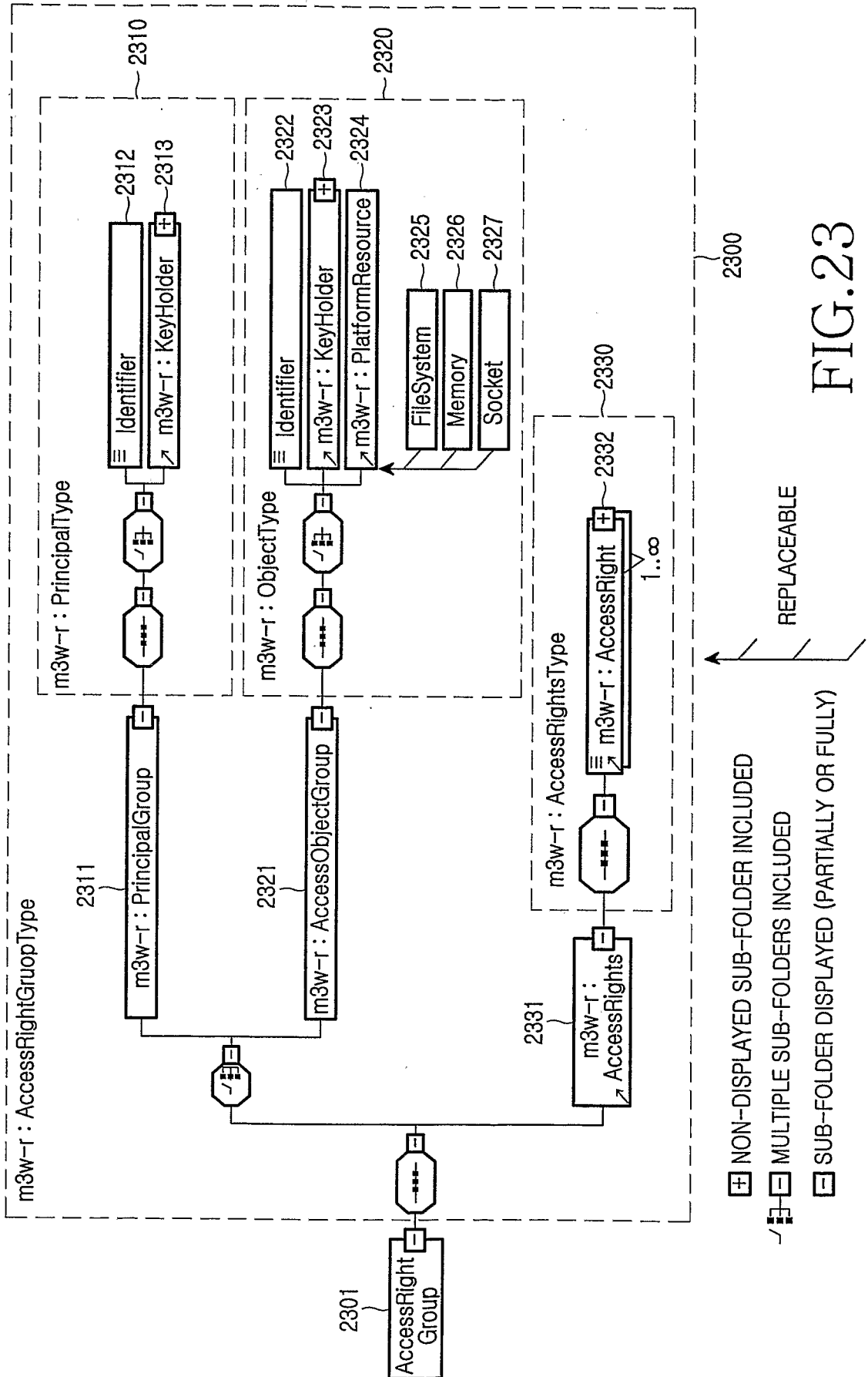
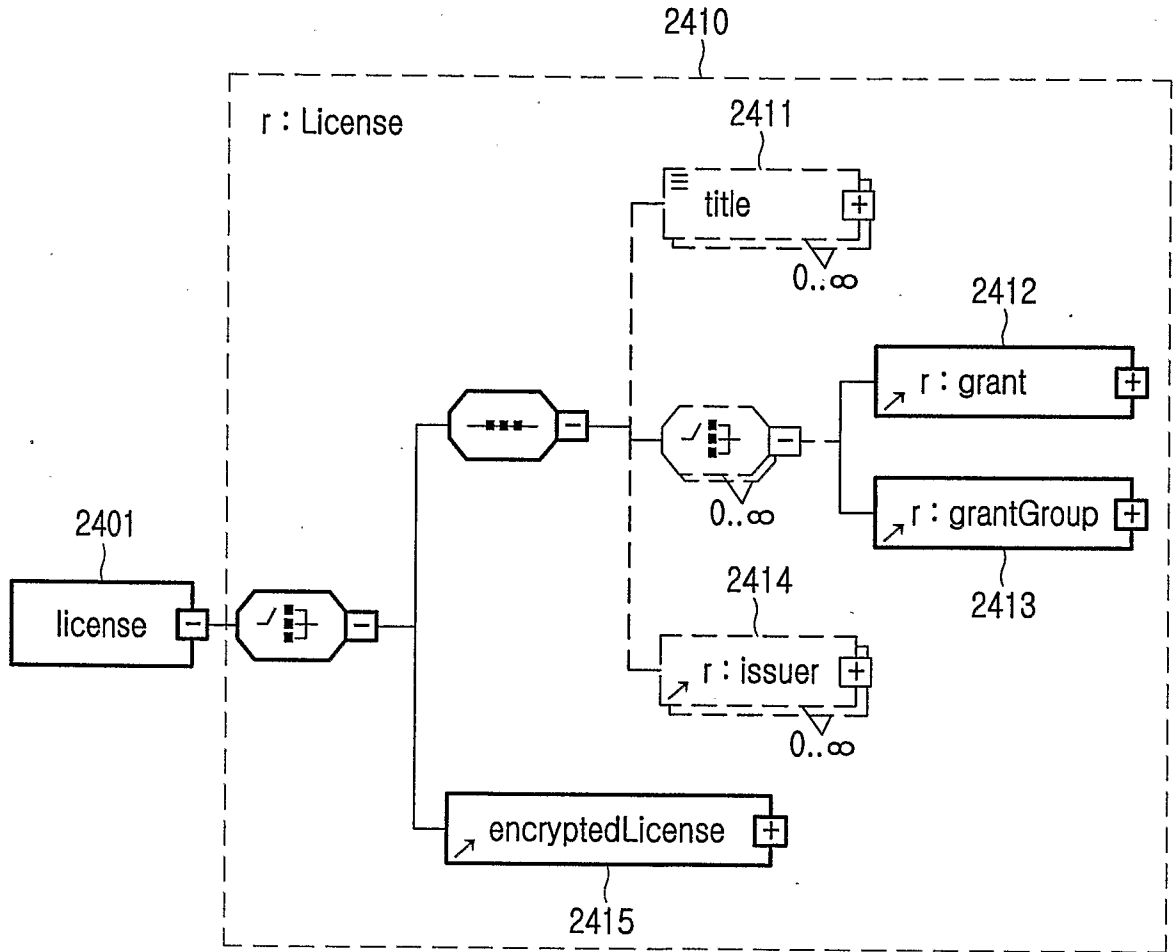


FIG. 23



- ⊕ NON-DISPLAYED SUB-FOLDER INCLUDED
- ↙ ⊕ MULTIPLE SUB-FOLDERS INCLUDED
- ▭ SUB-FOLDER DISPLAYED (PARTIALLY OR FULLY)

FIG.24

INTERNATIONAL SEARCH REPORT

International application No.
PCT/KR2007/000200**A. CLASSIFICATION OF SUBJECT MATTER****H04L 29/06(2006.01)i, H04L 9/32(2006.01)i**

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L 29/06, G06F 17/00

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

Korean Utility models and applications for Utility models since 1975

Japanese Utility models and applications for Utility models since 1975

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

e-KIPASS(KIPO internal) : "middleware", "security", "multimedia"

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2004/0139088 A1(DAVIDE MANDATO. et al.) 15 July 2004 see the abstract and figure 1.	1-51
A	US 6,199,204 B1(SEAMUS DONOHUE) 06 March 2001 see the abstract and figure 4A.	1-51
A	US 6,526,257 B2(MIWAKO DOI. et al.) 25 February 2003 see the abstract and figure 1.	1-51
A	KR10-2004-0058338 A(KONINKLIJKE PHILIPS ELECTRONICS N.V.) 03 July 2004 see the abstract, claim1 and figure 2.	1-51

 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

25 APRIL 2007 (25.04.2007)

Date of mailing of the international search report

25 APRIL 2007 (25.04.2007)

Name and mailing address of the ISA/KR

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

Facsimile No. 82-42-472-7140

Authorized officer

IM, DAE SHIK

Telephone No. 82-42-481-5947



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/KR2007/000200

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US2004/0139088A1	15.07.2004	CN1511406A	07.07.2004
		EP01248431A1	09.10.2002
		EP1248431A1	09.10.2002
		JP2004537187T2	09.12.2004
		US2004139088AA	15.07.2004
		W002078289A1	03.10.2002
		W02002078289A1	03.10.2002
		US6,199,204B1	06.03.2001
GB2333864B2	07.05.2003		
GB2333865A	04.08.1999		
GB2333865A1	04.08.1999		
GB2333865B2	14.05.2003		
GB9801661A0	25.03.1998		
GB9808657A0	24.06.1998		
JP11272454A2	08.10.1999		
JP3385590B2	10.03.2003		
US6199204B1	06.03.2001		
US6199204BA	06.03.2001		
US6202207B1	13.03.2001		
US6202207BA	13.03.2001		
US6,526,257B2	25.02.2003		
		US20010053514A1	20.12.2001
		US2001053514AA	20.12.2001
		US6526257BB	25.02.2003
		KR10-2004-0058338A	03.07.2004
AU2002348916A8	10.06.2003		
AU2002348916AH	10.06.2003		
AU2002348916AA	10.06.2003		
BR200206702A	17.02.2004		
CN1596531A	16.03.2005		
EP01451997A2	01.09.2004		
EP1451997A2	01.09.2004		
JP2005527011T2	08.09.2005		
RU2004119436A	10.11.2005		
US20050022015A1	27.01.2005		
US2005022015A1	27.01.2005		
US2005022015AA	27.01.2005		
W003047204A2	05.06.2003		
W02003047204A2	05.06.2003		
W02003047204A3	23.10.2003		