CONTENT PRESENTATION

Background of the Invention

The present invention relates to a method and apparatus of presenting content, and in particular to a digital rights management (DRM) system for authorising the presentation of content.

Additionally, the present invention relates to a method and apparatus for encrypting and decrypting content.

Description of the Prior Art

The reference to any prior art in this specification is not, and should not be taken as, an acknowledgement or any form of suggestion that the prior art forms part of the common general knowledge.

Conventionally, content owners sell premium content in tapes, CDs, and DVDs through traditional outlets. However, due to the success of more powerful personal computers, high speed Internet connections, and superior compression technologies, the way that digital media content, for instance, music, video, etc are produced, distributed, and consumed has been dramatically changed.

Downloading encoded digital audio-visual content files via HTTP, FTP, etc, has gained favour among Internet users because of the immediate access to desired content and no requirement of a trip to a store or reliance on physical media, for instance, a tape, a CD, or a DVD. Content owners now have the opportunity to generate additional sales and revenue by means of distributing digital audio-visual content through the Internet. But the digital audio-visual content that is available for sale on the Internet is still limited because content owners, artists, and publishers are concerned about protecting their copyright works from illegal use.

Some other on-line multimedia services become available at the same time, for instance, VOD (Video-On-Demand), Live Broadcasting, Live Streaming for Surveillance, etc, through Internet, wherein, the content providers are also much concerned about protecting their contents, especially for those highly security-sensitive ones, from unauthorised users.

Unfortunately, Internet piracy and illegal global distribution have been always a headache to the entertainment industry. In order to reduce the risk of piracy and gain the benefit of a huge market of potential Internet customers, content must be protected throughout the production, distribution, and consumption process. Before owners of digital audio-visual content offer their valuable content for sale or promotion on the Internet, a secure e-commerce system that protects digital audio-visual content from illegal use must be established. A critical component of any such e-commerce system is Digital Rights Management (DRM), which consists of a set of technologies that can protect content owners' copyrights. In general, DRM distributes digital audio-visual content and limits access to only those people who have acquired a proper license to play the content. Although 100 percent security is not possible, DRM can prevent naive attackers from bypassing such a DRM system, making it difficult and costly for skilled attackers to compromise such a DRM system, and minimise the scope of breaks and limits commercial opportunities for professional attackers. However, while DRM adds persistent security to the digital audio-visual content that it protects, it also adds extra steps to the process of acquiring digital audio-visual content. There is always a trade-off between usability and security, which is dependent on the content owner to establish a balance.

Typically the majority of DRM solutions are complicated and expensive to implement, such as Microsoft Corporation's Media Player system, or Macrovision's systems.

Currently there does not exist a low cost and easy implemented solution providing satisfactory performance..

Summary of the Present Invention

In a first broad form, the present invention includes a method of presenting content, wherein the method includes, in an end station, generating an access request; transferring the access request to a base station; receiving an agent from the base station; implementing the agent, wherein the agent determines end station data indicative of an identity of the end station; and transfers the end station data to the base station, the base station being responsive to the end station data to authenticate presentation of the content; and presenting the content in response to a successful authentication.

Preferably, the method includes, in the agent, generating the end station data using at least one of an indication of the content to be presented; an end station identifier, the end station identifier being uniquely indicative of the identity of the end station; a user identifier, the user identifier being uniquely indicative of a user of the end station; an agent identifier, the agent identifier being uniquely indicative of the respective agent; and a licence.

Also preferably the method includes, in the agent, determining the end station identifier using at least one of a Network Interface Card (NIC) Number; a Media Access Control (MAC) Address; CPU identifier; and Hard Drive identifier.

It is also preferable that the method includes, in the end station receiving the content; determining indicating data indicative of the content; and, generating the access request using the indicating data.

Preferably the method includes, in the end station, receiving the content on a physical storage medium.

It is also preferable that method includes, in response to a successful authentication by the base station receiving the content; and, presenting the received content.

It is preferable that the end station is coupled to the base station via a communications network, and wherein the method includes transferring, via the communications network, at least one of the content; the end station data; the access request; and, the agent.

Preferably the content is encrypted using an encryption algorithm, and wherein the method includes, in the end station receiving decryption data from base station; decrypting the encrypted content using the decryption data; and, presenting the decrypted content.

It is preferable that the decryption data includes at least one of: a decryption algorithm; and, a decryption key.

It is also preferable that the decryption data is encrypted using a predetermined public key, of a private/public key pair, and wherein the method includes, in the end station, decrypting the decryption data using the private key.

It is preferable the private key is encrypted using a block cipher key, and wherein the method includes, in the agent receiving the block cipher key from the base station; decrypting the private key using the block cipher key; and, transferring the private key to the end station.

It is also preferable the decryption data includes first and second keys, wherein the content includes one or more data packets, each data packet having an encrypted payload, and wherein the method includes, in the end station, and for each data packet determining the payload; decrypting a first portion of the payload using the first key; decrypting a second portion of the payload using the second key, the first and second payload portions being arranged such that at least some of the payload is encrypted using both the first and second keys.

Preferably the decryption data includes first and second keys, wherein the content includes one or more data packets, each data packet having an encrypted payload, and wherein the method includes in the end station, and for each data packet determining the payload; determining an arbitrary value using the first encryption key; determining a cipher text by encrypting the arbitrary value using the second encryption key; and, decrypting the payload, using the cipher text, the encrypted payload being formed from a logical combination of the unencrypted payload and the cipher text.

It is preferable the logical combination being an XOR combination.

It is also preferable the content includes two or more content portions, and wherein the method includes, in the end station segmenting the content into respective content portions, each content portion including one or more data packets; and, decrypting the data packets of each content portion using respective first and second keys; and, combining the decrypted content portions to the form the content to be presented.

Preferably the content portion corresponds to a respective stream.

Also preferably, the content includes at least one of audio information; visual information; graphical information; multi-media information; music; and, video.

It is preferable the end station data includes a licence indicative of one or more media types, the licence being encoded using a predetermined algorithm.

It is also preferable the agent is a software application, and wherein the method includes, in the end station, implementing the agent by executing the software application.

Preferably the agent terminates on the end station at least one of after the authentication; after content presentation; and, after the end station disconnects from the base station;

In another broad form, the present invention provides a method of authenticating the presentation of content, wherein the method includes, in a base station receiving an access request from an end station; transferring an agent to the end station, the agent being responsive to determine end station data; and transfer the end station data to the base station; authenticating the presentation of content using the end station data; and authorising the presentation of content in response to a successful authentication.

It is preferable the method includes, in the base station, determining from the end station data, at least one of an end station identifier, the end station identifier being uniquely indicative of the identity of the end station; a user identifier, the user identifier being uniquely indicative of a user of the end station; an agent identifier, the agent identifier being uniquely indicative of the respective agent; and a licence.

Preferably the method includes, in the base station, determining at least one of a Network Interface Card (NIC) Number; a Media Access Control (MAC) Address; CPU identifier; and Hard Drive identifier.

It is also preferable the method includes, in the base station, transferring the content to the end station on a physical storage medium.

Preferably the method includes, in the base station comparing the end station data to authentication data stored in a data store, the authentication data being indicative of end stations; and, authenticating the end station in response to a successful comparison.

Also preferable, the method further includes, in the base station in response to a successful authentication, determining the content to be presented; comparing the content to a authorisation data, the authorisation data being indicative of the content that may be

presented by each end station; and, authorising the presentation of the content in response to a successful comparison.

Preferably the base station is coupled to the end station via a communications network, and wherein the method includes transferring via the communications network, at least one of the content; the end station data; the access request; and, the agent.

It is preferable that the content is encrypted using an encryption algorithm, and wherein the method includes, in the base station encrypting the content; generating decryption data; and, transferring the decryption data to the end station, the end station being responsive to the decryption data to decrypt the content.

It is also preferable that the decryption data includes at least one of a decryption algorithm; and, a decryption key.

Preferably the method includes in the base station determining a public key of a public/private key pair in accordance with the identity of the end station data; and, encrypting the decryption data using the public key.

It is preferable that the method further includes, in the base station, transferring a block cipher key to the agent, the agent being responsive to the block cipher key to decrypt the private key.

Also preferable, the decryption data includes first and second keys, wherein the content includes one or more data packets, and wherein the method includes, in the base station, and for each data packet determining a payload; encrypting a first portion of the payload using the first key; and, encrypting a second portion of the payload using the second key, the first and second payload portions being arranged such that at least some of the payload is encrypted using both the first and second keys.

Preferably the decryption data includes first and second keys, wherein the content includes one or more data packets, and wherein the method includes, in the base station, and for each data packet determining the payload; determining an arbitrary value using the first encryption key; determining a cipher text by encrypting the arbitrary value using the second encryption key; and, encrypting the payload, using the cipher text, the encrypted payload being formed from a logical combination of the unencrypted payload and the cipher text.

It is preferable that the logical combination being an XOR combination.

Preferably the content includes two or more content portions, and wherein the method includes, in the base station segmenting the content into respective content portions, each content portion including one or more data packets; and, encrypting the data packets of each content portion using respective first and second keys.

Also preferably each content portion corresponds to a respective stream.

It is preferable that the content includes at least one of audio information; visual information; graphical information; multi-media information; music; and, video.

In another broad form, the present invention includes an end station used for the presentation of content, wherein the end station includes a display; and a processor for generating an access request; transferring the access request to a base station; receiving an agent from the base station; implementing the agent, wherein the agent:

determines end station data indicative of an identity of the end station; and transfers the end station data to the base station, the base station being responsive to the end station data to authenticate presentation of the content; and presenting the content in response to a successful authentication.

In another broad form, the present invention includes a base station used for authorising the presentation of content, wherein the base station includes a processor for receiving an access request from an end station; transferring an agent to the end station, the agent being responsive to determine end station data; and transfer the end station data to the base station; authenticating the presentation of content using the end station data; and authorising the presentation of content in response to a successful authentication.

In another broad form, the present invention includes a digital right management system for managing the presentation of content, wherein the system includes a base station including a processor for receiving an access request from an end station; transferring an agent, wherein the agent; determines end station data; and transfers the end station data to the base station. authenticating the end station data; and authorising the presentation of content in response to a successful authentication; one or more end stations, each end station including a display; and a processor for generating an access request; transferring the access request to a base

station; receiving an agent from the base station; implementing the agent, wherein the agent determines end station data; and transfers the end station data to the base station, the base station being responsive to the end station data to authenticate presentation of the content; and presenting the content in response to a successful authentication; a communications network for interconnecting the base station and the one or more end stations.

In another broad form, the present invention includes a method of decrypting content for presentation, the content including one or more data packets, each data packet having an encrypted payload, and wherein the method includes, in an end station receiving encrypted content; for each data packet: determining the payload; and, decrypting the payload using first and second keys; and, combining the payloads of the one or more data packets to form the content.

It is preferable that the method includes, in the end station decrypting a first portion of the payload using the first key; and, decrypting a second portion of the payload using the second key, the first and second payload portions being arranged such that at least some of the payload is encrypted using both the first and second keys.

Preferably the method includes in the end station determining an arbitrary value using the first key; determining a cipher text by encrypting the arbitrary value using the second key; and, decrypting the payload, using the cipher text, the encrypted payload being formed from a logical combination of the unencrypted payload and the cipher text.

It is also preferable that the logical combination being an XOR combination.

Preferably the content includes two or more content portions, and wherein the method includes, in the end station segmenting the content into respective content portions, each content portion including one or more data packets; and, decrypting the data packets of each content portion using respective first and second keys; and, combining the decrypted content portions to the form the content to be presented.

It is preferable that the method includes, in the end station receiving decryption data from an end station, the decryption data including the first and second keys; and, decrypting the decryption data using a private key of a public/private key pair.

It is also preferable that the private key is encrypted using a block cipher key, and wherein the method includes, in the end station receiving the block cipher key from the base station; decrypting the private key using the block cipher key; and, transferring the private key to the end station.

In another broad form, the present invention includes an end station for decrypting content for presentation, the content including one or more data packets, each data packet having an encrypted payload, and wherein the end station includes a processor for: receiving encrypted content; for each data packet determining the payload; and, decrypting the payload using first and second keys; and, combining the payloads of the one or more data packets to form the content.

In another broad form, the present invention includes a method of encrypting content for presentation, the content including one or more data packets, each data packet having a payload, and wherein the method includes, in a base station for each data packet determining the payload; and, encrypting the payload using first and second keys; and, combining the payloads of the one or more data packets to form the encrypted content.

It is preferable that the method includes, in the base station encrypting a first portion of the payload using the first key; and, encrypting a second portion of the payload using the second key, the first and second payload portions being arranged such that at least some of the payload is encrypted using both the first and second keys.

Preferably A method according to claim 52, wherein the method includes in the base station determining an arbitrary value using the first key; determining a cipher text by encrypting the arbitrary value using the second key; and, encrypting the payload, using the cipher text, the encrypted payload being formed from a logical combination of the unencrypted payload and the cipher text.

It is preferable that the logical combination being an XOR combination.

It is also preferable that the method includes, in the base station segmenting the content into two or more content portions, each content portion including one or more data packets; encrypting the data packets of each content portion using respective first and second keys; and, combining the encrypted data packets to the form the encrypted content.

Preferably the method includes, in the base station generating decryption data including the first and second keys; encrypting the decryption data using the public key of a public/private key pair; and, transferring the decryption data to an end station, the end station being responsive to decrypt the decryption data using the private key.

In another broad form, the present invention includes a base station for encrypting content for presentation, the content including one or more data packets, each data packet having a payload, and wherein the base station includes a processor for, for each data packet determining the payload; and, encrypting the payload using first and second keys; and, combining the payloads of the one or more data packets to form the encrypted content.

Brief Description of the Drawings

An example of the present invention will now be described with reference to the accompanying drawings, in which: -

Figure 1 is a schematic diagram of an example of a DRM System.

Figure 2 is a schematic diagram of the functionality of the system of Figure 1;

Figure 3 is a schematic diagram of an example of an encryption method.

Figure 4 is a schematic diagram of additional details of the encryption method of Figure 3;

Figure 5 is a schematic diagram of a first specific example of the encryption of a payload of Figure 4;

Figure 6 is a schematic diagram of a second specific example of the encryption of a payload of Figure 4;

Figure 7 is a schematic diagram of an example of a decryption method.

Figure 8 is a schematic diagram of additional details of the decryption method of Figure 7;

Figure 9 is a schematic diagram of a first specific example of the decryption of a payload of Figure 8;

Figure 10 is a schematic diagram of a second specific example of the decryption of a payload of Figure 8;

Figure 11 is schematic diagram of an example of the structure of the Encipheror;

Figure 12 is a schematic diagram of an example of the structure of the Decipherer;

Figure 13 is a schematic diagram of an example of the Key Package Management;

Figure 15 is a schematic diagram of an example of the method of updating of the key for Key Package;

Figure 16 is a schematic diagram of an example of the main processes of registering an end station and License Distribution;

Figure 16 is a schematic diagram of an example of the main method of User Authentication and Key Distribution;

Figure 17 is a schematic diagram of an example of the method of the end station receiving the License Transfer Key;

Figure 18 is a schematic diagram of an example of the main processes of License Transfer.

Detailed Description of the Preferred Embodiments

An example of a DRM system will now be described with respect to Figure 1.

In particular, the DRM system shown in Figure 1 includes a session resource manager (SRM) 5 coupled via a suitable connection, such as an SSL connection to a web server 13. In turn, the web server 13 is used to allow a client 23, such as a user's end station, to access the features provided by the SRM 5, via an appropriate browser 15 and/or an SSL connection as shown.

In use, the client 23 is coupled to, or includes a playback device for displaying content, which includes an AV decoder 16 and an AV renderer 17.

In this example, the SRM 5 includes a key escrow module 6 coupled to a database 7. The database 7 is also coupled to a user account management module 8, a server agent management module 9, a license management module 10 and a trust authentication module 11.

In use, the SRM is coupled to an encipheror 3 which operates to perform encryption of content 2 in accordance with an encryption strategy 1. The encipheror 3 is also coupled to a file/data server 4 which contains a content repository 12, which in use is used to store the encrypted content.

The client 23 includes a data store 18 coupled to a decipheror 19. The decipheror 19 is also coupled to a rights manager module 21, a secure registry 20 and a server security agent 22, as shown.

In use, content 2 is encrypted by the encipheror 3 using the encryption strategy 1 and stored in the content repository 12 as shown. Simultaneously, the encipheror 3 provides an indication of the encryption keys used in the encryption process to the key escrow module 6, which in turn stores these in the database 7 as a key package.

A client wishing to view content undergoes a registration procedure with the SRM 5, during which the client registers the client 23 with the SRM 5, via the browser 15 and the web server 13. This allows the client to obtain a license allowing predetermined content to be presented on the playback device and/or the client 23, as will be described in more detail below.

When a user wishes to present content, the user must obtain authorisation from the SRM 5. In order to achieve this, the end station is adapted to generate an access request which is transferred to the SRM 5, for example via the web server 13. The SRM 5 responds to the access request by providing an agent which is transferred to, and implemented by the client 23, as shown at 22.

The agent 22 collects predetermined information regarding the client 23 and transfers this back to the SRM 5, allowing the SRM 5 to authenticate the identity of the client 23 and determine a list of the content which the client device is authorised to present. This is then used to allow the client 23 to present the content.

Typically, this may therefore involve having the SRM 5 perform trust authentication using the trust authentication module 11 and then transfer keys stored in the database 7 to client 23. These keys can then be used by the client 23 to decrypt the content which is transferred to the client 23 from the content repository 12.

It will be appreciated by persons skilled in the art that the any one or more of the SRM 5, the web server 13, the encipheror 3, and file/data server 4 may be provided at a centralised base station, which can then be coupled to one or more end stations, such as the client devices 23, via a suitable communications network, such as the Internet, one or more WANs (Wide Area Networks), LANs (Local Area Networks), or the like.

In this sense, the base station is a collection of one or more processing systems providing the functionality outlined above.

Thus Figure 1 shows an example of a system embodiment of digital rights management (DRM) for network streaming and distribution of digital audio-visual content, which allows highly security-sensitive or high-value content to be securely prepared, delivered, and presented.

The elements described above with respect to Figure 1 can include so-called general-purpose computers, workstations, and personal computers, as well as network connectable information-processing devices for instance, digital home electric appliances, portable terminals for instance, PDAs, and cellular phones, or the like. It should be noted that the process described below may be performed by a software product and that a part of process may be done on a hardware unit.

An example of the system will now be described in more detail with reference to Figure 2.

In particular, in Figure 2, it is highlighted that the SRM 5 includes a key escrow module 6, a security agent manager 31, a user authentication module 32, a user account manager module 35, a license manager 34, a web server 13 and a program schedule manager 33. The system also includes means for encoding audio and video received at 24, in the form of an encoder server 30 which includes a software/hardware encoder 25, coupled to the encipheror 3. The encipheror 3 is in turn coupled to the streaming server 28, a content manager 29 and a content network archiving manager 27. Encryption is performed in accordance with a schedule provided by a schedule manager 26 coupled to the encoder 25.

The encoder server 30 and the SRM 5 are coupled via network 36, such as the Internet, a LAN, WAN, or the like. It will be appreciated that connections to the network, and the network itself may be wired or wireless.

Also coupled to the network is the file/data server 4, a video server 48 and the client 23. The file/data server 4 typically includes an archive server 39, an FTP server 40, a backup server 41, a cache server 42 and a content repository 12, as well as an encipheror/decipheror module 38, and an content importer/exporter module 37.

The video server includes a key server 45, a VOD (Video-On-Demand) server 46 and a scheduled multicast streaming server 47.

The client end device 23 typically includes a decipher 19, coupled to via an AV decoder 16, to an AV renderer 52 and a sound card 53. The decipheror is also coupled to a network streaming client module 49, and via a content importer module 50, to a CD/DVD reader 51. A rights manager 21 is also provided as shown.

In use, content is encoded using the encoder server 30 and then transferred to and stored on the file/data server 4 content may then be transferred to the client 23 either in the form of a physical media 43 via for example a retail shop 44, or by download from the network 36.

Details of the encryption mechanism used for encrypting the content by the encoder server 30 are stored on the SRM 5 in use when a client wishes to send the content, an access request is transferred via the network to the SRM 5 which causes the authentication process as described above to be performed. Once the client device has been authenticated, encryption keys are transferred to the client 23 to allow the content to be presented.

Thus, Figure 2 is an overview of a multimedia system of digital audio-visual content distribution, to which a digital rights management (DRM) for network streaming and distribution of digital audio-visual content can be applied. This allows highly security-sensitive or high-value content is securely prepared, delivered, and consumed according to the present invention.

The function of each of the elements in this example will now be described in more detail.

(a) Encoder Server(s) 30

1. The software encoder or a hardware encoder 25 generates the digital audiovisual content, for instance, MPEG-1/2/4, Windows Media, etc, from the analog audiovisual inputs in real-time according to certain Record Schedule that is controlled by the Schedule Manager 26 or the Administrator,
2. The Schedule Manager 26 performs Record Schedule management,
3. The Encipheror 3 encrypts the stream data in real-time according to certain Encryption Strategy, package the encryption keys, and pass the keys to SRM (Session Resource Manager) 5,
4. The Content Manager 29 backs up and manages the digital audiovisual content data locally, and/or provide the exporting point to the

administrator to copy the digital audiovisual content to other location upon/after the encoding process is finished,

5. The Content Network Archiving Manager 27 pushes real-time digital audiovisual content data to the File/Data Server 4 through network 36, for instance, LAN (Local Area Network) or Internet, for real-time data archiving,
6. The Unicast/Multicast Live Streaming Server 28 provides the real-time access on the encrypted digital audiovisual content to the legal users on the network 36, for instance, LAN or Internet, by either Unicast or Multicast,

(b) File/Data Server 4

1. The Archive Server 39 archives the digital audiovisual content(s) that is/are from Encoder Server 30, either by real-time streaming or administrator's operation,
2. The FTP (File Transfer Protocol) Server 40 provides FTP file downloading service to legal users,
3. The Backup Server 41 allows the digital audiovisual content to be backed up to additional storage media, for instance, CDs, DVDs, tapes, etc, either automatically or on request of administrator,
4. The Cache Server 42 provides content cache to Video Server 48 that would provide streaming service to user, for instance, VOD (Video On Demand), Broadcasting, etc,
5. The Content Repository 12 stores the archived digital audiovisual content,
6. The Content Importer 37 and the Encipheror 38 allow an administrator to import digital audiovisual content (s) to File/Data Server 4, which is/are not generated from the Encoder Server 30, and encrypt the digital audiovisual content in real-time during importing process,
7. The Content Exporter 37 and the Decipherer 38 allow an administrator to export digital audiovisual content(s) in File/Data Server 4 to physical media, for instance, CD, DVD, etc, and sell on the shelf of retail shops 44, wherein, the digital audiovisual content could be in its original data

format that is generated by real-time decryption during exporting process or in its secure format that keeps its data encrypted,

(c) The Video Server 48

1. The VOD Server 46 provides VOD services to the Client device(s) 23,
2. The Cache Server 45 downloads the requested digital audiovisual content from File/Data Server 4 when stream caching is necessary,
3. The Scheduled Multicast Streaming Server 47 provides broadcasting programs,

(d) The SRM (Session Resource Manager) 5 manages the system resources,

1. The Program Schedule Manager 33, defines encoding schedule for Encoder Server 30, and broadcasting program schedule for Video Server 48,
2. The Web Server 13 handles Web access and transaction and to distribute the Server Security Agent 22 by implementing the Server Agent Platform 14 with Software Agent technology,
3. The Database 7 stores all system information,
4. The Key Escrow 6 manages the digital audiovisual content encryption key package,
5. The Security Agent Manager 31 manages the Server Security Agent 22 that roams to the Client 23 and do the security task on behalf of SRM 5,
6. The User Authentication module 32 is responsible for user authentication,
7. The User Account Manager 35 manages user's or client's account,
8. Having a License Manager 34, which manages the license issuing and verification,

(e) The Client(s) 23 which play back digital audiovisual content either from network streaming or from physical media,

1. The Network Streaming Client 49 receives digital audiovisual content data from network 36
2. The Content Importer 50 includes a physical media reader, for instance, CD reader, DVD reader, etc, to get digital audiovisual content data from physical media,
3. The Rights Manager 21 protects the digital right of content owner,

4. The Decipherer 19 decrypts the digital audiovisual content data, which is controlled by the Rights Manager 21,
5. The Audiovisual Decoders 16 and Audiovisual Rendering Devices 52 and 53 decode and render the digital audiovisual content,

In one example, content can be generated by the Encoder Server 30, where the content is encrypted in real-time. However, a digital audio-visual content may also be imported by the administrator using the Content Importer 37 in File/Data Server 4, where the digital audio-visual content is encrypted during the importing process. This allows an administrator or even authorised users to define an Encryption Strategy for initialisation of the Encipheror 3/38 for the specific digital audio-visual content.

The Encipheror defines the Block Cipher Algorithm, e.g., DES (Data Encryption Standard), AES (Advanced Encryption Standard), Blowfish, etc The Encipheror also defines the seeds for generating random Keys, the preferred Key Length for the Block Cipher Algorithms that accept variable length keys, the preferred Plaintext Block Size, e.g., 64-bit, 128-bit, 256-bit, etc. Otherwise, a default Encryption Strategy is to be used.

The Encipheror 3/38 employs an encryption method, referred to hereon as the Equal Payload Length Digital Audio-visual Content Encryption mechanism, which implements Block Cipher Algorithms to encrypt content portion or streams of data. The Equal Payload Length Digital Audio-visual Content Encryption mechanism only encrypts the audio-visual frame payload/raw data , with leaving the data packet headers untouched. These headers include for instance, System Header, GOP (Group of Packet) Header, VOP (Video Object Plane) Header, Video Packet Header, Audio Packet Header, ASF (Advanced system Format) Data Packet Header, etc. This are described in more detail below.

The encrypted digital audio-visual content is stored in the Content Repository 12 included by in the File/Data Server 4, where the digital audio-visual content can be exported to a physical medium such as CDs or DVDs. This physical medium can be sold in the retail shops 44, and additionally the content on the physical medium can be accessed directly through the Internet via VOD, Broadcasting, and FTP services.

In this example, the Key Escrow 6 can be solely responsible for the maintenance and management of the encryption key package of a digital audio-visual content. The encryption

key package is transmitted through a pre-established SSL (Secure Sockets Layer) link from the Encipheror 3/38 to the Key Escrow 6 and is protected by using Asymmetric Ciphers (or Public Key Cryptographic Algorithms, e.g., RSA (Rivest-Shamir-Adelman)).

A user can access the content provider's E-commerce web site through a web browser 13 to register him as a legal user of some or all of the services that the content provider offers. After this step the client device receives a License and Key Package from content provider to present the contents at the client device.

In one example, a Web Server 13 is used to transfer a user's personal information to the SRM 5 through a pre-established SSL link. The User Account Manager 35 maintains and manages all the user's information, including user's Login ID and Password.

An SSL connection is established upon user registration between client 23 and the Web Server 13 for secure transaction on the Internet. It allows a Server Security Agent 22, which employs the Software Agent Technology, to dynamically roam and be transferred to the client 23.

The Server Security Agent 22 collects local unique information relating to the Client's device, for the SRM 5. The unique information includes one or more of:

- a NIC (Network Interface Card) number,
- a MAC (Media Access Control) Address,
- a CPU identifier,
- a Hard Disk identifier;
- or the like.

The unique information is sent back to Web Server 13, and then forwarded to SRM 5 to construct a unique end station identifier (referred hereon as a User ID).

The unique User ID, together with user's Login ID and Password, are used to authenticate a legal user. Furthermore, working with the Web Server 13, the Server Security Agent 22 acts as a middleman for message exchange between Client 23 and SRM 5 in later transactions. It is preferable that the Server Security Agent 22 has no right to write data to the local hard disk of Client 23 or change any local configurations. Moreover, the Server Security Agent 22 preferably terminates before the Client application 23 quits.

However it is also conceived that that Server Security Agent may also terminate after authorisation of the content or after the presentation of the content on the end device.

Once the client registration is successful, the SRM 5 generates a pair of public and private keys for the user, encrypting the private key using a cipher key K_{csr}, and sends it to the user.

A specific license is also generated by SRM 5 for the user to define the corresponding service type and content usage rules. This license is also encrypted and sent to the user.

The Client application 23 obtains the encrypted private key and license from the Server Security Agent 22, saving them in the secure registry 20, which could be a special place in the local hard disk, or the system registry, for later transactions.

Each time when a legal user wants to present the content, he accesses the content provider's E-commerce web site with a web browser 13. A new Server Security Agent 22 is transferred to the client application 23.

After the user's Login ID and Password having been validated, the Server Security Agent 22 determines the unique local information of user's device and forward this to the SRM 5, where a unique User ID is generated based on the received information.

The generated user ID is compared to the User ID that has been stored in the Database 7 during the registration process. Only if the two User IDs match does the user pass the authentication of User ID.

In the event that the user IDs match, the Server Security Agent 22 transfers the encrypted License from the Client application 23 and forwards it to the SRM 5, where the received license is decrypted and compared to the one that has been stored in the SRM Database 7 during the registration process. Only if the two Licenses match does the user pass the authentication of User License.

After the user is authenticated, SRM 5 parses the License code and returns the corresponding encrypted Digital Audio-visual Content Key Package to the Server Security Agent 22. The security server agent transfers the decrypted Key Package to the Rights Manager 21 in the client application 23. The Rights Manger 21 in the client application 23 analyses the License

and feeds the Key Package to the Decipherer 19, where the digital audio-visual content is decrypted.

The content is then fed to the decoders 16 for presentation.

A user of a client device can transfer the license to another client device, for instance, PC, Pocket PC, etc. Moreover, a user is allowed to apply multiple licenses for different devices at the same time.

In one example, a method of encryption and decryption of content is provided that seeks to not introduce redundant bits into the content after it is encrypted. Additional bits mean greater modifications on the original digital audio-visual content.

The encryption/decryption method are only encrypts the audio-visual frame payload/raw data. The headers of the payloads are not encrypted.

The encrypted content can be manipulated by the existing streaming protocols, for instance, RTP (Real-time Transport Protocol), RTSP (Real Time Streaming Protocol), Specification of ISMA (Internet Streaming Media Alliance), etc., without any additional components or modifications on those protocols. Furthermore, protection on the payload data, which contains the real content information. In general, this is sufficient to prevent illegal access of the content as the header can only be decoded, therefore the payload is not presented.

Figure 3 is a diagram schematically illustrating the encryption algorithm in the Equal Payload Length Digital Audio-visual Content Encryption mechanism.

In this example, after the payload is encrypted, there is no change on the number of bits contained within the content.

For digital audio-visual coding standards, for instance, MPEG (Moving Picture Experts Group)-1/2/4, Windows Media, H.263/264, etc., a digital audio-visual content 54 could be divided into one or more data packets, for instance, video packets, audio packets, metadata packets, etc., with packet header and corresponding packet payload.

The encryption algorithm 56 here is applied only on the packet payload without any changes of the number of bits contained within that packet payload. For instance, header 57 are untouched after the encryption. However, the payload 58 is passed through the encryption algorithm 56 resulting in the encrypted payload 59. Figure 4 is a diagram briefly showing the result of applying the encryption algorithm to the content. Content usually contains multiple elementary streams, e.g., Video stream(s) 62, Audio stream(s) 63, Script stream(s) 64, Metadata 65, etc.

Accordingly, in order to enhance the security and introduce more flexibility, different elementary streams are preferably encrypted with different encryption keys and/or algorithms. Furthermore, the digital audio-visual content 60 is demultiplexed into individual elementary streams 62, 63, 64, and 65, by the Demultiplexer 61 before the encryption, with the encrypted elementary streams 66, 67, 68, and 69 being multiplexed into a single stream by the Multiplexer 70 to construct a new encrypted digital audio-visual content 71.

The encryption procedure will now be discussed in more detail.

Assuming a predefined Encryption Strategy is selected, which defines the Block Cipher Algorithm, e.g., DES, AES, Blowfish, etc, the seeds for generating random Keys, the preferred Key Length for the Block Cipher Algorithms that accept variable length keys, and the preferred Plaintext Block Size, e.g., 64-bit, 128-bit, 256-bit, etc.

The Plaintext Block Size is assumed to be N_b bits, the Key Length is N_k bits, the Stream Packet Header Length is N_h bits, and the Stream Packet Payload Length is N_p bits.

According to one example, two keys are applied to the stream packet payload at the same time, which is presented as $(K_1, K_2)_i$, where i represents a different elementary stream. For any elementary stream, the encryption are applied on each packet payload independently, although the key pair is same for the same elementary stream. There are several cases when we apply the encryption mechanism.

The first case is when $N_p >= N_b$,. In this case, according to the requirement of a Block Cipher Algorithm, the total length of the plaintext should be a multiple of N_b. However, most of the time, this is not true for N_p.

One way to solve this problem is to add some dummy bits (N_d) to the end of the packet payload so that the (N_p + N_d) would be multiple of N_b. However this introduces disadvantages such as increasing the length of the content.

Accordingly, in another example, all of the packet payloads in a stream are considered as a whole with some bytes (N_s) being selected from the successive packet payload to ensure that ($N_p + N_s$) would be multiple of N_b..

However, this may cause some delays by waiting for the next one or more packets, especially during real-time streaming. Moreover, doing so, different data packets are related during encryption/decryption. Furthermore, some additional bytes as indicators in the encrypted stream are necessary for the above-mentioned methods to indicate N_p and N_d/N_s for decryption process. This also requires changing of the final packet length. Then comes with the modification of the headers.

In a further example, these problems are avoided adding any additional bits to the encrypted data by employing one pair of keys, i.e., (K_1, K_2).

The detail of the algorithm is shown in Figure 5, and includes:

a. Segmenting or demultiplexing the content into individual elementary streams which are analyzed to classify the header and the payload data 72, whereas the header is not typically encrypted as step S1 shows;
b. The payload data of each packet are put into the Encryption Buffer;
c. Compare N_p and N_b. If $N_p >= N_b$, encrypt the first N_b bits of the payload data in the Encryption Buffer with K_1 as S2 shows, then replace the plaintext with the corresponding ciphertext in the Encryption Buffer, but the rest $(N_p - N_b)$ bits would be untouched, as S3 shows;
d. Perform a modulus calculation,

$$z = N_p \bmod N_b$$

and, $$n = N_p - zN_b$$

where, z is an integer value and n is the remainder of the modulus calculation;
e. Skip the first n bits in the Encryption Buffer, which is part of the ciphertext from the above steps that is encrypted with K_1, as S5 shows,

f. Encrypt the rest zN_b bits in the Encryption Buffer with K_2, and replace the plaintext with the ciphertext, as S6 shows,

g. The encrypted packet 74 is re-constructed by combining the original header and the encrypted payload data without any additional bits,

h. The encrypted packet is fed to the multiplexer 70 to be multiplexed with other elementary streams to construct the encrypted digital audiovisual content 71.

The extreme case is that n = 0. Then the first N_b bits of the payload data are encrypted with K_1 and K_2. Because $K_1 \neq K_2$, the first N_b bits is not recovered but encrypted twice.

In the situation where Np is less than Nb a different method is used. Normally N_b is 64 (e.g., DES), or 128 (e.g, AES), or 256 (e.g., AES), this case is seldom happening for digital audio-visual contents that N_p is only 8, or 16, or 32 bytes. For such a case, it is typical to implement the Cipher Feedback (CFB) operation mode of Block Cipher Algorithms. In this mode, the plaintext itself is not actually passed through the Block Cipher Algorithms, but is merely XORed with an output block.

a. An N_b bits block called the Shift Register 75, which is initially set to some arbitrary value, is used as the input plaintext to the Block Cipher Algorithm,

b. The ciphertext is then passed through an extra component called the M-Box 76, which simply selects the left-most M bits of the ciphertext, where M is the number of bits in the data block that we wish to encrypt, i.e., N_p,

c. The selected M-bit is XORed with the real plaintext that is to be encrypted, shown as S11,

d. The output of the XOR operation is the final ciphertext.

In one example, K_1 77 is used as the initial Shift Register value, V_r. If the length of the keys, N_k, is less than N_b, a number of dummy bits are added to the end of K_1, which are set to zero, to make V_r has the same length as that of N_b, which is shown in S7. If N_k is equal to or greater than N_b, then only the left-most N_b bits from K_1 77 are used as the initial Shift Register value, which is shown in S8. The next step involves encrypting V_r with K_2, which is shown in S9, resulting in producing the ciphertext of V_c 78.

Then as shown in S10 , the left-most N_p bits of V_c are selected to XOR the N_p bits of payload data, which is shown in S11, in the data packet 72. The result is the final ciphertext. As

shown in S12, after combining the final ciphertext with the original header, the encrypted data packet 74 is constructed.

In order to access the content, the user must access the encryption key pairs of the digital audio-visual content. After that, User can access the digital audio-visual content by any means for instance, VOD, FTP, Live Streaming, etc, on the Internet, or buying CDs, DVDs, etc, from the retail shops. Those procedures are to be described in following sections. Before the digital audio-visual content could be decoded and rendered, it must be decrypted at first. The decryption process is just the reverse process of the encryption process, which is depicted in the following.

Decryption of a stream uses the reverse strategy that is defined in the encryption mechanism. As known in the art for Block Cipher Algorithms, which are also called Symmetric Encryption Algorithms, the Decryption Keys are the same keys as Encryption Keys. Figure 7 shows the decryption algorithm in the Equal Payload Length Digital Audio-visual Content Encryption mechanism.

The content 55 is divided into one or more packets, for instance, video packets, audio packets, metadata packets, etc., with packet header and corresponding packet payload. The decryption algorithm 80 is applied only to the packet payload without any changes to the number of bits contained within the payload.

For instance, header 57 are untouched after the decryption. However payload 59 are input to the decryption algorithm 80 wherein the a decrypted payload 58 is output. Moreover, Figure 8 shows the realisation of the decryption algorithm in the Equal Payload Length Digital Audio-visual Content Encryption mechanism for decrypting encrypted content. The encrypted content is demultiplexed or segmented into individual encrypted elementary streams, such as encrypted Video stream(s) 66, encrypted Audio stream(s) 67, encrypted Script stream(s) 68, encrypted Metadata 69, etc. before the decryption.

Furthermore, each encrypted elementary stream consists of one or more data packets 55 with corresponding header and payload. Only the payload are input to the decryption algorithm 80 to obtain the original payload data. The decrypted payload data and the unchanged headers are combined to reconstruct the original data packets 54. Then the original elementary

streams 62, 63, 64, 65 are restored. The streams are multiplexed after the decryption to restore the original digital audio-visual content 60.

The following sets out the decryption procedure in more detail. In particular, assuming a predefined Encryption/Decryption Strategy is selected, which defines the Block Cipher Algorithm, e.g., DES, AES, Blowfish, etc, the seeds for generating random Keys, the preferred Key Length for the Block Cipher Algorithms that accept variable length keys, and the preferred Plaintext Block Size, e.g., 64-bit, 128-bit, 256-bit, etc.

Again assuming the Plaintext Block Size is N_b bits, the Key Length is N_k bits, the Stream Packet Header Length is N_h bits, and the Stream Packet Payload Length is N_p bits. Two keys are applied to the stream packet payload at the same time, which is presented as $(K_1, K_2)_i$, where i represents different elementary stream.

For any elementary stream, the encryption is applied on each packet payload independently, although the key pair is same for the same elementary stream. So the decryption is also applied on each packet payload independently. And different elementary streams are likely use different key pairs. For decryption, there are the same cases as for encryption that we have to consider.

For example, when N_p is greater than or equal to N_b the detail of the decryption algorithm is shown as following, which is also shown in Figure 9.

a. After being demultiplexed, each individual encrypted elementary stream is analyzed to each data packet 74 with classifying the header and the (encrypted) payload data, whereas the header is not be decrypted as step S13 shows,
b. The (encrypted) Payload data of each packet is input into the Decryption Buffer,
c. N_p and N_b are compared. If $N_p >= N_b$, then the modulus calculation is performed such that,

$$z = N_p \bmod N_b$$

and, $$n = N_p - zN_b$$

where, z is an integer value and n is the remainder of the modulus calculation,

d. As, $n \in [0, N_b)$, and showing as S14, then the first n bits in the Decryption Buffer are skipped, which is part of the ciphertext that is encrypted with K_1,

e. Decrypt the rest zN_b bits in the Decryption Buffer with K_2, and replace the ciphertext with the corresponding plaintext, as S15 shows,

f. Then, as S16 shows, the header is unchanged again and as S17 shows, the last $(N_p - N_b)$ bits of the payload data are also untouched, but decrypt the first N_b bits of the payload data in the Decryption Buffer with K_1, then replace the ciphertext with the corresponding plaintext in the Decryption Buffer, which is shown as S18

g. The packet 72 is re-constructed by combining the original header and the decrypted payload data without any additional bits,

h. The decrypted packet is fed to the multiplexer 70 to be multiplexed with other elementary streams to re-construct the original digital audiovisual content 60.

The extreme case is that n = 0. Then the first N_b bits of the payload data is decrypted with K_2 and K_1.

For N_P is less than N_b , the ciphertext is merely XORed with the output block from the M Box to get the plaintext, which is shown in the following manner and in Figure 10,

a. If $N_p < N_b$, an N_b bits block called the Shift Register 75, which is initially set to some arbitrary value, is used as the input plaintext to the Block Cipher Algorithm,

b. Use K_1 77 as the initial Shift Register value, V_r. If the length of the keys, N_k, is less than N_b, just simply adding several dummy bits to the end of K_1, which are set to zero, to make V_r has the same length as that of N_b, which is shown in S19. If N_k is equal to or greater than N_b, just take the left-most N_b bits from K_1 as the initial Shift Register value, which is shown in S20,

c. Shown in S21, encrypt V_r with K_2 getting the ciphertext of V_c, 78 which is then passed through the M-Box 76,

d. Shown in S22, select the left-most M bits of the ciphertext of V_c 78, where M is the number of bits in the data block that we wish to decrypt, i.e., N_p,

e. Shown in S23, the selected M-bit is XORed with the payload data that we wish to decrypt,

f. The output of the XOR operation is the final plaintext,

g. Shown in S24, the header is not decrypted. By combing the original header and the decrypted payload data, the original data packet 72 is reconstructed.

The Encipheror 3 and Decipherer 19 are shown respectively in Figures 11 and 12.

The content may contain one or several elementary streams, for instance, Audio Stream, Video Stream, Script, etc. Each elementary stream are encrypted by a pair of block cipher keys, i.e., $(K_1, K_2)_i$, where i is the Stream Number in the digital audio-visual content. Therefore content will include a respective Key Package, i.e.,

$$\{(K_1, K_2)_1, (K_1, K_2)_2, \ldots, (K_1, K_2)_i\}$$

which consists of all the key pairs for all elementary streams.

The content is stored in the Content Repository 12 in the File/Data Server 4. However, the corresponding Key Package does not reside at the same location of the content for security purposes.

When the user registers with the base station, the SRM 5 exchanges it's a Public Key with the Encipheror 3/38 according to the Diffie-Hellman Key Exchange mechanism. Then using Asymmetric Ciphers (or Public Key Cryptographic Algorithms, e.g., RSA (Rivest-Shamir-Adelman)), the Key Package is encrypted with the Public Key of the SRM (SRM) 5, i.e., K_{SRM_pub} by the Encryption Key Generator 83 in the Encipheror 3/38.

The encrypted Key Package is sent to the Key Escrow 6 of SRM 5 through a network connection implementing SSL (Secure Sockets Layer) protocol.

At the SRM 5, a Key Database 7 stores the Key Package. After receiving the Key Package, the Key Escrow 6 decrypts it with the Private Key of the SRM 5, i.e., K_{SRM_pte}, and encrypts the Key Package again with the Secure Storage Key, i.e., K_{ss}, using Block Ciphers, e.g., DES, AES, etc.

The encrypted Key Package is saved into Key Database 7. K_{ss} is solely managed by SRM 5 and may be changed from time to time an of SRM 5. .

Figure 13 and Figure 14 show the Key Package Management and the updating of the key for Key Package respectively.

Figure 15 shows the main processes of Client Register and License Distribution. A Client accesses the content provider's E-commerce web site to register the preferred services, for instance, VOD, Live Broadcasting, etc.

A dedicated Web Server 13 provides the E-commerce web site to the client. The Web Server 13 could either coexist with SRM 5 or not. A secured connection implementing SSL protocol may be established between Web Server 13 and SRM 5. Furthermore, once the Client 23 starts the registration process, an SSL connection may be established between Client 23 and the Web Server 13 to protect Client's personal information.

If the registration process is successful, the Client's information are recorded in the User Account Database 7, where all the records should be encrypted by the Secure Database Entry Key, i.e., $K_{sse,}$ using Block Ciphers, e.g., DES, AES, etc. K_{sse} is solely managed by SRM 5 and should be changed from time to time by the administrator of SRM 5.

The steps in the registration process typically include:

(a) Web Server 13 and SRM 5 establish an SSL link, as S25 shows.

(b) A User accesses to the Content Provider's E-commence web site using web browser to register certain service(s), as S26 shows.

(c) With the agreement of the User, a Client Application 23 e.g., which may be a Java Applet, is downloaded from the Web Server 13, as S27 shows. The User installs it on the end station , as S28 shows.

(d) With the agreement of the User, the Client Application 23 establishes an SSL connection with the Web Server 13 for further transactions, as S29 shows.

(e) At steps S30, S31, and S32 the user information is sent to the Web Server 13 through the SSL link. The user information is then transferred to the SRM 5 by the Web Server 13, either through local transactions or SSL connection. The User Account Manager 35 in SRM 5 encrypts User's information with K_{sse} and inserts it into the User Account Database 7. Any changes on K_{sse} from the administrator triggers the decryption and re-encryption processes for all the records in the User Account Database 7.

(f)The User accepts an agreement to allow a Server Security Agent 22 program, which originates from Web Server 13 through the SSL connection, to be implemented by the client device. The Server Security Agent 22 implements the

Software Agent technology and determines or collects some unique information of the User's local machine on behalf of SRM 5. Once the Server Security Agent 22 is implemented, the Server Security Agent shares the SSL connection that has been established between the Client 23 and the Web Server 13. By providing its unique Agent ID, the Server Security Agent 22 are authenticated by SRM 5. All the collected local information are relayed by the Web Server 13 and transmitted to SRM 5, where they are encrypted with the Secure Database Entry Key, i.e., K_{sse}, and saved in the User ID Database 7 that is related to the User Account Database 7. The aforesaid steps are shown in S33, S34, S35, S36, and S37.

(g) SRM 5 generates a Public-Private Key Pair for the User, i.e., K_{user_PUB} and K_{user_PTE}. Those two keys are encrypted with the Secure Database Entry Key, i.e., K_{sse}, and saved in the Transaction Key Database 7, which is related to the User Account Database 7. Moreover, SRM 5 generates another Client Secure Registry Key, i.e., K_{csr}, which is a Block Cipher Key, and encrypt the User Private Key with K_{csr} according to a predefined Block Cipher algorithm, e.g., AES, Blowfish, etc. K_{csr} is also encrypted with K_{sse} and saved in the Transaction Key Database 7. Then the encrypted User Private Key and K_{csr} are sent to the Web Server13, then to the Server Security Agent 22 through the SSL links. The aforesaid steps are shown in S38, S39, and S40.

(h) The Server Security Agent 22 passes the encrypted K_{user_PTE} to the Client Application 23. Then the Client Application 23 writes the encrypted K_{user_PTE} to it Secure Registry 20. However, K_{csr} is not passed to the Client Application 23 with the encrypted K_{user_PTE}. It is kept at the Server Security Agent 22 as long as the Server Security Agent 22 is implemented by the end station. When the Client Application 23 needs K_{user_PTE}, Client Application 23 must pass the encrypted K_{user_PTE} to the Server Security Agent 22. There, K_{user_PTE} is decrypted with K_{csr} and returned back to the Client Application 23 for further transactions. The aforesaid step is shown in S41.

(i) The Server Security Agent 22 may update K_{csr} on request from the SRM 5. For such a case, it requires the Client Application 23 to pass the encrypted K_{user_PTE}, decrypt it, re-encrypt it with the new K_{csr}, and require the Client Application 23

to overwrite the old encrypted K_{user_PTE} with the new one in the Secure Registry 20.

(j) According to the registered services from the User, different License Codes are generated by SRM 5. The License Code defines the Service Type and the Usage Rule under which the content would be used. The License Code is stored in the License Database 7, which is related to the User Account Database 7, after being encrypted with K_{sse}. The aforesaid step is shown in S42.

(k) The License Code is also encrypted with K_{csr} before being sent to the Server Security Agent 22 along with K_{csr}. The Server Security Agent 22 passes the encrypted License Code to the Client Application 23. Then the Client Application 23 writes the encrypted License Code to the end station's Secure Registry 20. But K_{csr} is not passed to the Client Application 23. It is kept at the Server Security Agent 22 as long as the Server Security Agent 22 is alive. When the Client Application 23 needs the License Code, the Client Application 23 must pass the encrypted License Code to the Server Security Agent 22. There, License Code is decrypted with K_{csr} and returned back to the Client Application 23 for further transaction. The aforesaid steps are shown in S43 and S44.

(l) The Server Security Agent 22 may update K_{csr} on request from SRM 5. For such a case, the Client Application 23 passes the encrypted License Code, decrypt it, re-encrypt it with the new K_{csr}, and require the Client Application 23 to overwrite the old encrypted License Code with the new one in the Secure Registry 20.

(m) The Server Security Agent 22 terminates when the Client Application 23 is closed, or when the content is finished being presented.

Figure 16 shows the method of User Authentication and Digital Audio-visual Content Encryption Key(s) Distribution. Each time, if a registered user wants to enjoy the services from content provider, the user may be authenticated. The method of user is authentication includes:

1. At steps S45, S46, S47, and S48, a user accesses content provider's website through Web Browser by providing Login User ID and password. If successful, the installed Client Application 23 are implemented.

2. At step S49, the user requests for certain service through the Client Application 23, e.g., playback of a content through VOD, watching a live broadcasting program, or ordering a CD, etc.
3. Client Application 23 establishes an SSL connection with content provider's Web Server 13, which is shown in S50.
4. SRM 5 generates a unique Agent ID and sends it to Web Server 13. The Server Security Agent 22 roams or is transferred from Web Server 13 through the SSL connection and is implemented on the end station . Once the Server Security Agent 22 is implemented, it shares the SSL connection that has been established between the Client 23 and the Web Server 13. Then the Web Server 13 relays any messages between the Server Security Agent 22 and SRM 5. By providing its unique Agent ID, the Server Security Agent 22 is authenticated by SRM 5. It automatically collects or determines unique information of the User's local machine on behalf of SRM 5. The necessary information includes NIC (Network Interface Card) MAC Address, CPU ID, Hard Disk ID, etc. that are unique for a computer. All the collected local information are transmitted to SRM 5, where it is compared with the corresponding records in the User ID Database 7 that is saved during user register procedure. Those records could be easily located by relating to the User's login User ID and Password. If they don't match, the SRM 5 regards the current login user as illegal user and inform the Server Security Agent 22 to terminate. The Web Server 13 is also informed by SRM 5 about the situation and reject any further request from the illegal Client device. The aforesaid steps are shown in steps S51, S52, S53, S54, S55, S56, and S47.
5. In step S57, if the verification is passed, the Server Security Agent 22 preferably requests the License Code that is in the Secure Registry 20 from the Client Application 23 and transfers it to the SRM 5.
6. After the License Code is sent to SRM 5 for verification, the SRM 5 compares the Licence Code to the corresponding record in the Database 7. If the licences do not match, SRM 5 regards the User holding an illegal License and rejects any request from the client device. The aforesaid steps are shown at S58, S59, S60, and S47.
7. At step S61, provided that the License Code is legal, the SRM 5 transfers the corresponding Client Secure Registry Key, i.e., K_{csr}, to the Server Security Agent 22.

8. At step S62, then SRM 5 checks the Service Type that is defined in the License Code to make a decision on providing the user with the content is permitted in the Licence. Furthermore, SRM 5 checks the Usage Rule that is defined in the License Code to decide how to send the Key Package of the requested digital audiovisual content. For instance, if a User can only access the video stream in the digital audiovisual content, the SRM 5 only needs to send the corresponding key pair for the video stream in the Key Package to the Server Security Agent 22 in the Client Application 23.
9. At step S63, assuming the User has full access on the digital audiovisual content, the SRM 5 obtains the encrypted Key Package from the Key Database 7, decrypts it with the Secure Storage Key, i.e., K_{ss}, re-encrypts it with the public key of the User, i.e., K_{user_PUB}, and transfers it to the Server Security Agent 22 in the Client Application 23.
10. The Server Security Agent 22 passes the Key Package to the Client Application 23, which is shown at step S64.
11. At step S65, the Client Application 23 obtains the encrypted private key, i.e., K_{user_PTE}, from the Secure Registry 20 and transfers it to the Server Security Agent 22, where the K_{user_PTE} is to be decrypted with K_{csr}.
12. At step S66, the Server Security Agent 22 returns the decrypted K_{user_PTE} back to the Client Application 23, where the Key Package would be decrypted with K_{user_PTE}.
13. The Client Application 23 obtains the License Code from the Secure Registry 20 and transfers it to the Server Security Agent 22, where the License Code is to be decrypted with K_{csr}. The Server Security Agent 22 returns the decrypted License Code back to the Client Application 23.
14. The Key Package is passed to the Decipherer 19 whereas the License Code is passed to the Rights Manager 21 in the Client Application 23, at step S67.
15. At step S68, the requested digital audiovisual content is fed into the Decipherer 19 before it reaches the AV Decoder 16. The Decipherer is controlled by Rights Manager 21, and decrypts the content with the Key Package. The output of the Decipherer 19 is to be transferred to the AV Decoder 16.
16. The presentation of the content at the client device begins.

Figure 17 shows the process for User to obtain a License Transfer Key. Moreover Figure 18 shows the processes for User to successfully transfer a license. Due to the unique User ID that is used to verify that the User is based on the registered client device, a transfer of a licence must be accomplished if the user wishes to access the content provider's service from another device. The user can either obtain multiple licenses, or transfer the license from a registered device to the preferred device.

A User can apply for multiple licenses for one or more client devices such as notebook PC, Pocket PC, and even hand phone. Each license may contain different usage rules. The User can register the client devices one by one, repeating the same procedure of a normal single license application. In the databases 7, the User typically has multiple IDs and multiple licenses that are related to the login ID and Password.

There is common case that the registered User wants to transfer the license from one device to another device. For instance, when the User is going on business trip, and the user wants to transfer the license from the desktop PC to the notebook PC so that they can access the service from content provider during the business trip. The procedure of license includes:

1. After being successfully authenticated, the User issues the License Transfer Request from the Client Application at the registered device, as shown at steps S69, S70, S71, and S72.
2. The end station 23 transfers the request to the Server Security Agent 22. Then the Server Security Agent 22 transfers the request to Web Server 13 through the established SSL link. Then Web Server 13 relays the request to SRM 5, at step S73.
3. Upon receiving the request, SRM 5 generates a unique License Transfer Key, encrypts it with K_{sse} and stores it in the License Database 7, then encrypts Licence Transfer key with K_{csr}, and sends it to the Server Security Agent 22 via the Web Server 13 through the SSL links, at step S74.
4. At step S76, the Server Security Agent 22 requests the Client Application 23 to save the encrypted License Transfer Key in the Secure Registry 20. Furthermore, the Server Security Agent 22 decrypts the License Transfer Key with K_{csr}, passes it to the Client Application. After that the Client Application 23 displays the

License Transfer Key to the User, wherein the user notes and remembers the Licence Transfer Key.

5. The User starts to register the preferred device that the license would be transferred to by logging on to the content provider's website with the Login ID and Password from the preferred device, at steps S77, and S78.
6. At step S79 and S80, similar to the first-time registration, the User must accepts the agreement that a Server Security Agent 22 can roam or transfer to the new client device collecting and determining the unique local information on behalf of SRM 5.
7. Once the Server Security Agent 22 is implemented, the Server Security Agent shares the SSL connection that has been established between the Client 23 and the Web Server 13. The Web Server 13 relays messages between the Server Security Agent 22 and SRM 5. By providing it's the unique Agent ID, the Server Security Agent 22 are authenticated by SRM 5.
8. At step S81, the User must issue the License Transfer request and provide the License Transfer Key to the Client Application 23, instead of applying a brand new license. The Client Application 23 forwards the License Transfer request to the Server Security Agent 22 together with the License Transfer Key.
9. The Server Security Agent 22 sends the License Transfer Key to SRM 5 through Web Server 13. SRM 5 checks the received Licence Transfer Key with its record in the License Database 7 that is related to the User's Login ID and Password. If they do not match, the SRM 5 regards that this is an illegal license transfer request and rejects it. The aforesaid steps are shown in S82, S83, S84, and S85.
10. If the two License Transfer Keys match, the SRM 5 requests the Server Security Agent 22 to collect the local device information. The SRM 5 encrypts the information with the Secure Database Entry Key, i.e., K_{sse}. Before SRM 5 saves the new User ID information in the User ID Database 7, SRM 5 may check whether there is any identical User ID that is related to the same User's Login ID and Password, which would be added by previous operations.
11. If no same record exists, SRM 5 marks the old User ID to be obsolete but keeps it in the database 7 for future reference. Then SRM 5 saves the new User ID into the User ID Database 7. If there is a same record, SRM 5 may remove it in the

database 7 before inserting the newly received information. The aforesaid steps are shown in S86, and S87.

12. At step S88, with the help of the databases 7, SRM 5 can easily locate and transfer the User's private key, i.e., K_{user_PTE}, the License Code, and the K_{csr} to the Server Security Agent. This data that is transferred is encrypted with K_{csr} .
13. At step S89, the Server Security Agent 22 request the Client Application 23 to save the encrypted K_{user_PTE} and License Code to the Secure Registry 20.
14. The SRM 5 checks the current connection with the Server Security Agent 22 in the old device which previously owned the license. If the connection is still active, the SRM 5 does not activate the new User ID immediately until the User quits from the Client Application 23 in the old device. If the connection has been disconnected, the SRM 5 activates the new User ID immediately and is ready for any request from the new device, at step S90.
15. If User tries to use the old machine to access the service after the License Transfer has been successful, are reminded that the license has been transferred to another machine.
16. Re-doing the above-mentioned steps, the User can transfer the license to another client device

There is the case when a User wishes to update the hardware in the client device, for instance, upgrading CPU, changing NIC, and so on. Since the unique User ID is based on the hardware of User's local machine, any changes on hardware may cause the license illegal. It is compulsory that if a User wishes to update his/her hardware, he/she must apply for License Transfer before he/she any update of hardware is performed.

Persons skilled in the art will appreciate that numerous variations and modifications will become apparent. All such variations and modifications which become apparent to persons skilled in the art, should be considered to fall within the spirit and scope that the invention broadly appearing before described.

THE CLAIMS DEFINING THE INVENTION ARE AS FOLLOWS:

1) A method of presenting content, wherein the method includes, in an end station:
   a) generating an access request;
   b) transferring the access request to a base station;
   c) receiving an agent from the base station;
   d) implementing the agent, wherein the agent:
      i) determines end station data indicative of an identity of the end station; and
      ii) transfers the end station data to the base station, the base station being responsive to the end station data to authenticate presentation of the content; and
   e) presenting the content in response to a successful authentication.

2) A method according to claim 1, wherein the method includes, in the agent, generating the end station data using at least one of:
   i) an indication of the content to be presented;
   ii) an end station identifier, the end station identifier being uniquely indicative of the identity of the end station;
   iii) a user identifier, the user identifier being uniquely indicative of a user of the end station;
   iv) an agent identifier, the agent identifier being uniquely indicative of the respective agent; and
   v) a licence.

3) A method according to claim 2, wherein the method includes, in the agent, determining the end station identifier using at least one of:
   a) a Network Interface Card (NIC) Number;
   b) a Media Access Control (MAC) Address;
   c) CPU identifier; and
   d) Hard Drive identifier.

4) A method according to any one of the claims 1 to 3, wherein the method includes, in the end station:
   a) receiving the content;
   b) determining indicating data indicative of the content; and,

c) generating the access request using the indicating data.

5) A method according to any one of claims 1 to 4, wherein the method includes, in the end station, receiving the content on a physical storage medium.

6) A method according to any one of the claims 1 to 3, wherein the method includes, in response to a successful authentication by the base station:

a) receiving the content; and,

b) presenting the received content.

7) A method according to any of claims 1 to 6, wherein the end station is coupled to the base station via a communications network, and wherein the method includes transferring, via the communications network, at least one of:

a) the content;

b) the end station data;

c) the access request; and,

d) the agent.

8) A method according to any one of the claims 1 to 7, wherein the content is encrypted using an encryption algorithm, and wherein the method includes, in the end station:

a) receiving decryption data from base station;

b) decrypting the encrypted content using the decryption data; and,

c) presenting the decrypted content.

9) A method according to claim 8, wherein the decryption data includes at least one of:

a) a decryption algorithm; and,

b) a decryption key.

10) A method according to claim 9, wherein the decryption data is encrypted using a predetermined public key, of a private/public key pair, and wherein the method includes, in the end station, decrypting the decryption data using the private key.

11) A method according to claim 10, wherein the private key is encrypted using a block cipher key, and wherein the method includes, in the agent:

a) receiving the block cipher key from the base station;

b) decrypting the private key using the block cipher key; and,

c) transferring the private key to the end station.

12) A method according to any one of the claims 9 to 11, wherein the decryption data includes first and second keys, wherein the content includes one or more data packets, each data packet having an encrypted payload, and wherein the method includes, in the end station, and for each data packet:

a) determining the payload;

b) decrypting a first portion of the payload using the first key;

c) decrypting a second portion of the payload using the second key, the first and second payload portions being arranged such that at least some of the payload is encrypted using both the first and second keys.

13) A method according to any one of the claims 9 to 11, wherein the decryption data includes first and second keys, wherein the content includes one or more data packets, each data packet having an encrypted payload, and wherein the method includes in the end station, and for each data packet:

a) determining the payload;

b) determining an arbitrary value using the first encryption key;

c) determining a cipher text by encrypting the arbitrary value using the second encryption key; and,

d) decrypting the payload, using the cipher text, the encrypted payload being formed from a logical combination of the unencrypted payload and the cipher text.

14) A method according to claim 13, the logical combination being an XOR combination.

15) A method according to any one of the claims 9 to 14, wherein the content includes two or more content portions, and wherein the method includes, in the end station:

a) segmenting the content into respective content portions, each content portion including one or more data packets; and,

b) decrypting the data packets of each content portion using respective first and second keys; and,

c) combining the decrypted content portions to the form the content to be presented.

16) A method according to claim 15, wherein each content portion corresponds to a respective stream.

17) A method according to any of claims 1 to 16, wherein the content includes at least one of:

a) audio information;

b) visual information;

c) graphical information;

d) multi-media information;

e) music; and,

f) video.

18) A method according to any one of the claims 1 to 17, wherein the end station data includes a licence indicative of one or more media types, the licence being encoded using a predetermined algorithm.

19) A method according to any one of the claims 1 to 18, wherein the agent is a software application, and wherein the method includes, in the end station, implementing the agent by executing the software application.

20) A method according to claim 19, wherein the agent terminates on the end station at least one of:

a) after the authentication;

b) after content presentation; and,

c) after the end station disconnects from the base station;

21) A method of authenticating the presentation of content, wherein the method includes, in a base station:

a) receiving an access request from an end station;

b) transferring an agent to the end station, the agent being responsive to:

i) determine end station data; and

ii) transfer the end station data to the base station;

c) authenticating the presentation of content using the end station data; and

d) authorising the presentation of content in response to a successful authentication.

22) A method according to claim 21, wherein the method includes, in the base station, determining from the end station data, at least one of:

i) an end station identifier, the end station identifier being uniquely indicative of the identity of the end station;

ii) a user identifier, the user identifier being uniquely indicative of a user of the end station;

iii) an agent identifier, the agent identifier being uniquely indicative of the respective agent; and

iv) a licence.

23) A method according to claim 22, wherein the method includes, in the base station, determining at least one of:

a) a Network Interface Card (NIC) Number;

b) a Media Access Control (MAC) Address;

c) CPU identifier; and

d) Hard Drive identifier.

24) A method according to any one of claims 21 to 23, wherein the method includes, in the base station, transferring the content to the end station on a physical storage medium.

25) A method according to any one of the claims 21 to 24, wherein the method includes, in the base station:

a) comparing the end station data to authentication data stored in a data store, the authentication data being indicative of end stations; and,

b) authenticating the end station in response to a successful comparison.

26) A method according to claim 25, wherein the method further includes, in the base station:

a) in response to a successful authentication, determining the content to be presented;

b) comparing the content to a authorisation data, the authorisation data being indicative of the content that may be presented by each end station; and,

c) authorising the presentation of the content in response to a successful comparison.

27) A method according to any of claims 21 to 26, wherein the base station is coupled to the end station via a communications network, and wherein the method includes transferring via the communications network, at least one of:

a) the content;

b) the end station data;

c) the access request; and,

d) the agent.

28) A method according to any one of the claims 21 to 27, wherein the content is encrypted using an encryption algorithm, and wherein the method includes, in the base station:

a) encrypting the content;

b) generating decryption data ; and,

c) transferring the decryption data to the end station, the end station being responsive to the decryption data to decrypt the content.

29) A method according to claim 28, wherein the decryption data includes at least one of:

a) a decryption algorithm; and,

b) a decryption key.

30) A method according to claim 29, wherein the method includes in the base station:

a) determining a public key of a public/private key pair in accordance with the identity of the end station data; and,

b) encrypting the decryption data using the public key.

31) A method according to claim 30, wherein the method further includes, in the base station, transferring a block cipher key to the agent, the agent being responsive to the block cipher key to decrypt the private key.

32) A method according to any one of the claims 28 to 31, wherein the decryption data includes first and second keys, wherein the content includes one or more data packets, and wherein the method includes, in the base station, and for each data packet:

a) determining a payload;

b) encrypting a first portion of the payload using the first key; and,

c) encrypting a second portion of the payload using the second key, the first and second payload portions being arranged such that at least some of the payload is encrypted using both the first and second keys.

33) A method according to any one of the claims 28 to 31, wherein the decryption data includes first and second keys, wherein the content includes one or more data packets, and wherein the method includes, in the base station, and for each data packet:

a) determining the payload;

b) determining an arbitrary value using the first encryption key;

c) determining a cipher text by encrypting the arbitrary value using the second encryption key; and,

d) encrypting the payload, using the cipher text, the encrypted payload being formed from a logical combination of the unencrypted payload and the cipher text.

34) A method according to claim 33, the logical combination being an XOR combination.

35) A method according to any one of the claims 28 to 34, wherein the content includes two or more content portions, and wherein the method includes, in the base station:

a) segmenting the content into respective content portions, each content portion including one or more data packets; and,

b) encrypting the data packets of each content portion using respective first and second keys.

36) A method according to claim 35, wherein each content portion corresponds to a respective stream.

37) A method according to any of claims 21 to 36, wherein the content includes at least one of:

a) audio information;
b) visual information;
c) graphical information;
d) multi-media information;
e) music; and,
f) video.

38) An end station used for the presentation of content, wherein the end station includes:
a) a display; and
b) a processor for:
i) generating an access request;
ii) transferring the access request to a base station;
iii) receiving an agent from the base station;
iv) implementing the agent, wherein the agent:
(1) determines end station data indicative of an identity of the end station; and
(2) transfers the end station data to the base station, the base station being responsive to the end station data to authenticate presentation of the content; and
v) presenting the content in response to a successful authentication.

39) An end station according to claim 38, wherein the end station is adapted to perform the method of any one of the claims 1 to 20.

40) A base station used for authorising the presentation of content, wherein the base station includes a processor for:
i) receiving an access request from an end station;
ii) transferring an agent to the end station, the agent being responsive to:
(1) determine end station data; and
(2) transfer the end station data to the base station;
iii) authenticating the presentation of content using the end station data; and
iv) authorising the presentation of content in response to a successful authentication.

41) A base station according to claim 40, wherein the base station is adapted to perform the method of any one of the claims 21 to 37.

42) A digital right management system for managing the presentation of content, wherein the system includes:

a) a base station including a processor for:

(1) receiving an access request from an end station;

(2) transferring an agent, wherein the agent

(a) determines end station data; and

(b) transfers the end station data to the base station.

(3) authenticating the end station data; and

(4) authorising the presentation of content in response to a successful authentication;

b) one or more end stations, each end station including:

i) a display; and

ii) a processor for:

(1) generating an access request;

(2) transferring the access request to a base station;

(3) receiving an agent from the base station;

(a) implementing the agent, wherein the agent:

(b) determines end station data; and

(4) transfers the end station data to the base station, the base station being responsive to the end station data to authenticate presentation of the content; and

(5) presenting the content in response to a successful authentication;

c) a communications network for interconnecting the base station and the one or more end stations.

43) A method of decrypting content for presentation, the content including one or more data packets, each data packet having an encrypted payload, and wherein the method includes, in an end station:

a) receiving encrypted content;

b) for each data packet:

i) determining the payload; and,

ii) decrypting the payload using first and second keys; and,

c) combining the payloads of the one or more data packets to form the content.

44) A method according to claim 43, wherein the method includes, in the end station:

a) decrypting a first portion of the payload using the first key; and,

b) decrypting a second portion of the payload using the second key, the first and second payload portions being arranged such that at least some of the payload is encrypted using both the first and second keys.

45) A method according to claim 43, wherein the method includes in the end station:

a) determining an arbitrary value using the first key;

b) determining a cipher text by encrypting the arbitrary value using the second key; and,

c) decrypting the payload, using the cipher text, the encrypted payload being formed from a logical combination of the unencrypted payload and the cipher text.

46) A method according to claim 45, the logical combination being an XOR combination.

47) A method according to any one of the claims 43 to 46, wherein the content includes two or more content portions, and wherein the method includes, in the end station:

a) segmenting the content into respective content portions, each content portion including one or more data packets; and,

b) decrypting the data packets of each content portion using respective first and second keys; and,

c) combining the decrypted content portions to the form the content to be presented.

48) A method according to any one of the claims 43 to 47, wherein the method includes, in the end station:

a) receiving decryption data from an end station, the decryption data including the first and second keys; and,

b) decrypting the decryption data using a private key of a public/private key pair.

49) A method according to claim 48, wherein the private key is encrypted using a block cipher key, and wherein the method includes, in the end station:

a) receiving the block cipher key from the base station;

b) decrypting the private key using the block cipher key; and,

c) transferring the private key to the end station.

50) An end station for decrypting content for presentation, the content including one or more data packets, each data packet having an encrypted payload, and wherein the end station includes a processor for:

a) receiving encrypted content;

b) for each data packet:

i) determining the payload; and,

ii) decrypting the payload using first and second keys; and,

c) combining the payloads of the one or more data packets to form the content.

51) An end station according to claim 50, wherein the end station is adapted to perform the method of any one of the claims 43 to 49.

52) A method of encrypting content for presentation, the content including one or more data packets, each data packet having a payload, and wherein the method includes, in a base station:

a) for each data packet:

i) determining the payload; and,

ii) encrypting the payload using first and second keys; and,

b) combining the payloads of the one or more data packets to form the encrypted content.

53) A method according to claim 52, wherein the method includes, in the base station:

a) encrypting a first portion of the payload using the first key; and,

b) encrypting a second portion of the payload using the second key, the first and second payload portions being arranged such that at least some of the payload is encrypted using both the first and second keys.

54) A method according to claim 52, wherein the method includes in the base station:

a) determining an arbitrary value using the first key;

b) determining a cipher text by encrypting the arbitrary value using the second key; and,

c) encrypting the payload, using the cipher text, the encrypted payload being formed from a logical combination of the unencrypted payload and the cipher text.

55) A method according to claim 54, the logical combination being an XOR combination.

56) A method according to any one of the claims 52 to 55, wherein the method includes, in the base station:

a) segmenting the content into two or more content portions, each content portion including one or more data packets; and,

b) encrypting the data packets of each content portion using respective first and second keys; and,

c) combining the encrypted data packets to the form the encrypted content.

57) A method according to any one of the claims 52 to 56, wherein the method includes, in the base station:

a) generating decryption data including the first and second keys;

b) encrypting the decryption data using the public key of a public/private key pair; and,

c) transferring the decryption data to an end station, the end station being responsive to decrypt the decryption data using the private key.

58) A base station for encrypting content for presentation, the content including one or more data packets, each data packet having a payload, and wherein the base station includes a processor for:

a) for each data packet:

i) determining the payload; and,

ii) encrypting the payload using first and second keys; and,

b) combining the payloads of the one or more data packets to form the encrypted content.

59) A base station according to claim 58, the base station being adapted to perform the method of any one of the claims 52 to 57.