EUROPEAN PATENT SPECIFICATION

APPARATUS OF A FLEXIBLE AND COMMON IPMP SYSTEM FOR MPEG-2 CONTENT DISTRIBUTION AND PROTECTION

GERÄT FÜR EIN FLEXIBLES UND GEMEINSAMES IPMP-SYSTEM FÜR MPEG-2 INHALTSVERTEILUNG UND SCHUTZ

APPAREIL D’UN SYSTEME DE GESTION ET DE PROTECTION DE LA PROPRIETE INTELLECTUELLE (IPMP) FLEXIBLE ET PARTAGE POUR LA DIFFUSION ET LA PROTECTION DE CONTENU MPEG-2

Designated Contracting States:
DE FR GB

Priority: 03.09.2001 JP 2001265908

Date of publication of application: 23.06.2004 Bulletin 2004/26

Proprietor: MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD.
Kadoma-shi, Osaka 571-8501 (JP)

Inventors:
- JI, Ming
  389758 Singapore (SG)
- SHEN, Sheng Mei,
  Block 20
  689093 Singapore (SG)
- HUANG, Zhongyang
  271008 Singapore (SG)
- SENOH, Takanori
  Hirakata-shi, Osaka 573-0093 (JP)

Representative: Eisenführ, Speiser & Partner Patentanwälte Rechtsanwälte
Postfach 10 60 78
28060 Bremen (DE)

References cited:
WO-A-01/91052
WO-A-99/48296
WO-A-02/071752
WO-A-02/100037


Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).
Description

Technical Field

[0001] The present invention relates to content distribution and protection in MPEG-2 scope, especially to such applications where the protected content based on MPEG-2 is consumed by different Intellectual Property Management and Protection (IPMP) terminal, and the same content is protected by different IPMP system.

Background Art

[0002] Content distribution is becoming more and more demanding as multimedia data and contents can reach anywhere and anytime. Users are happy with the convenience and flexibility, and they can enjoy entertainment easily and efficiently. On the other hand, content owners are worried about the illegal usage of their property. There is a balance between two sides.

[0003] There are a lot of protection techniques for protecting the content, such as data encryption, watermarking, etc. They have been implemented in many content distribution applications. It seems different system employs different kinds of mechanisms and protection techniques to distribute content with protection. All the terminals or content consuming devices in that case are only able to play and consume the content that is provided by the same content provider. They cannot exchange their terminal or device to playback different contents.

[0004] In MPEG-4 context, a standardisation group has been working on MPEG-4 IPMP Extension. The solution is able to achieve both of the following:

1. Allow the same protected content to be consumed on different vendors’ MPEG-4 IPMP Terminals. This will be fully enabled.

2. Allow the same content to be protected by different vendors’ IPMP Tools. This will be assisted to as large extent as possible.

[0005] In MPEG-2 context, there is a CA (Conditional Access) system that defines a minimum set of common CA elements necessary to achieve interoperability between different CA systems. However, there is no real inter-operability here, as not enough components are defined, and the architecture offered by CA is not flexible enough.

[0006] It is very difficult to produce the same terminal to play different MPEG-2 contents provided by different content providers in such case. In other words the same protected MPEG-2 content cannot be played back in different CA system.

[0007] On the other hand, CA system defines a common scrambling algorithm, this makes hardware implementation simple, however, this makes the whole architecture too rigid. IPMP tool should not be fixed to certain tool beforehand, it should allow more flexibility for vendors to choose their favourite tool in their IPMP system. In such case it is necessary to define some standard way and interface to provide both better flexibility and security at the same time.

[0008] WO 99/48296 A discloses an apparatus for protection of streamed media content. A streaming media player includes a port designed to accept a digital bit stream. The digital bit stream includes content, which is encrypted in part, and a secure container including control information designed to control use of the content, including a key suitable for decryption of a portion of the content. The media player also includes a control arrangement including a means for opening secure containers and extracting cryptographic keys, and means for decrypting the encrypted portion of the content.

[0009] The document "Interoperable content protection for digital TV" by B.J. van Rijnsoever and J.P. Linnartz describes the OPIMA (Open Platform Initiative for Multimedia Access) system applied to digital TV, wherein the OPIMA enables interoperability between content protection systems and multimedia terminals. In an OPIMA MultiCrypt system, a generic multimedia terminal is instantiated for a specific IPMP system by downloading a corresponding software module or by inserting a corresponding hardware module. The module implements all functions that differ between different IPMP systems.

[0010] Hence, a flexible and interoperable IPMP system is needed in MPEG-2 system for content protection.

[0011] To define a flexible and interoperable IPMP system structure for MPEG-2 to allow:

1. The same protected MPEG-2 content to be consumed on different vendors’ MPEG-2 IPMP Terminals. This will be fully enabled.

2. The same MPEG-2 content to be protected by different vendors’ IPMP Tools. This will be assisted to as large extent as possible.

[0012] To provide the standard way for IPMP system implementers to build the whole IPMP system for MPEG-2 from
Disclosure of Invention

[0013] To solve the problem, the present invention provides apparatuses of a flexible and common Intellectual Property Management and Protection (IPMP) System for MPEG-2 content distribution and protection at the side of a content provider and at the side of a terminal as defined in claims 1 to 18, by which MPEG-4 IPMP Extension can be mapped to MPEG-2 with some modifications.

[0014] IPMP Control Information needs to put inside the content to describe what are the IPMP tools needed to play the content, and how they protect the content. It includes IPMP Tool List and IPMP Tool Container.

[0015] The IPMP Tool List supports indication of independent or alternative Tools. For each tool in the IPMP Tool List, the following information is provided:

1. IPMP Tool Identifier: A given IPMP tool is identified to other entities via its IPMP Tool Identifier, and an optional Parametric Description;
2. Possible alternatives to a given Tool; and
3. Optional Tool List Signature.

[0016] The IPMP Tool Container carries the binary tool itself in the content. One implementation of a given tool is carried as the payload of one IPNP Tool Container, the representation format, packaging information and IPMP Tool ID of which is specified in the container also.

[0017] IPMP Control Graph describes the association between different elementary streams under different programs and different IPMP tools, at the control point at which the IPMP tool should be running.

[0018] The IPMP Tool Manager is a conceptual entity in a given IPMP Terminal. Upon receipt of the Tool List, the Terminal should route the same to the IPMP Tool Manager for parsing and Tool retrieval. The Tool Manager also processes parametric descriptions, resolves alternative tools, and receives IPMP Tools that arrive in the Content.

[0019] All IPMP Tool Messages are routed through the Terminal. To represent this function, an entity called the Message Router (MR) is defined in the architecture. The MR connects and communicates with supported IPMP Tool(s). It thus abstracts the physical interface of one IPMP Tool from any other IPMP Tool that wishes to communicate with it.

<Operation of Invention>

[0020] On the content provider side, a media content is encoded using existing coding technology like MPEG-2 or MPEG-4, and encrypted using existing IPMP tool like DES or AES. The IPMP Tools serve modules for providing pre-determined functions and used alone or in combination. The content maybe embedded with watermarks using a Watermark Tool AAA (for example) before the encoding. The media content is then multiplexed using MPEG2 system.

[0021] At the same time, IPMP Control Information needs to be built up, that includes IPMP Tool List and an optional Tool Container carrying IPMP Tools. The IPMP Control Information is put in PSI in MPEG-2 transport stream, and a special PES packet in MPEG-2 program stream.

[0022] IPMP Tool List is also generated based on the IPMP Tools used in protecting the content. The IPMP Tools List includes IPMP Tool ID specifying a unique tool. It can also includes optional parametric description to allow the terminal to choose their own preferred tool that does the same function (for example, DES), and it may also includes a set of alternative tools so that the terminal can choose among a set of known tools that can accomplish the same task.

[0023] IPMP Control Graph is also built during content generation. For example, if DES tool is used to encrypt video elementary stream 0x01 after encoding, then, in the control graph, there needs to be an indication that DES tool is used to protect elementary stream 0x01, and it should be called at which control point (before decoding). If Watermark Tool AAA is used to insert watermark into audio elementary stream 0x02 before encoding, then, in the control graph, there needs to be an indication that Watermark Tool AAA needs to be called upon elementary stream 0x02 at which control point (after decoding).

[0024] On the terminal side, IPMP Tools List is passed to the IPMP Tool Manager module sitting inside MPEG-2 terminal. The Tool Manager checks whether all tools needed to consume the content are present in the terminal, if there is one missing, the terminal follows a proprietary way to retrieve the missing IPMP Tool according to the IPMP Tool ID or parametric description. The obtained IPMP Tools are now available in the terminal, and they are stored in the IPMP terminal to be ready for use with the pre-defined messaging interface.

[0025] The IPMP Control Graph is also parsed by the terminal, so that the terminal knows which IPMP Tool to invoke upon which elementary stream at which control point.

[0026] The content stream is then continuing to pass through content decoder, necessary IPMP tools are called upon,
and the content can be decoded and played back to the terminal.

<Effective of Invention>

[0027] This invention solves the problem to play back the same protected MPEG-2 content by different types of terminals, as well as to protect the same MPEG-2 content using different vendors' IPMP system, by introducing the MPEG-2 IPMP framework.

[0028] IPMP Control Information can be carried in PSI for MPEG-2 transport stream, or in PES packet for MPEG-2 program stream. The IPMP Control Information carries IPMP Tool List, or IPMP Tool Container in the form of five new descriptors.

[0029] The IPMP Tool List identifies, and enables selection of, the IPMP Tools required to process the Content. IPMP Control Graph indicates the association between IPMP tools and their protection scope (control point). Tool Container carries the IPMP Tool in the content stream.

[0030] IPMP Stream is the elementary stream within MPEG-2 system, to carry IPMP messages to each individual IPMP Tool instances.

[0031] IPMP Tool Manager and Message Router can be mapped from MPEG-4 IPMP Extension to MPEG-2 IPMP system.

Brief Description of Drawings

[0032] Figure 1 shows Content Distribution and Protection in different CA System for MPEG-2 as the prior art.

Figure 2 shows General Diagram for a Compliant MPEG-2 IPMP System.

Figure 3 shows a Diagram of MPEG-2 IPMP Terminal, in which "DB" denotes a decoder buffer and "RB" denotes a render buffer.

Best Mode for Carrying Out the Invention

[0033] Existing CA system in MPEG-2 does not provide an interoperable and flexible content protection mechanism for both the content owners and terminal vendors.

[0034] The Figure 1 shows the prior art for the current typical CA system.

[0035] Content Owner in unit 1.0 provides contents through different content providers X, Y, and Z in unit 1.1, 1.4, and 1.7. Different CA system is used to protect the content for different content provider as shown in unit 1.2, 1.5, 1.8.

[0036] Therefore Content Decoding or Content Consumption terminal is also different from each other, as shown in unit 1.3, 1.6, and 1.9.

[0037] It is clear that MPEG-2 content protected by CA system A can't be played out on terminals that support CA system B, also there is no complete specification on how different CA systems from different vendors are to protect the same content, and how the terminal gets to know that.

[0038] In this invention, we define a MPEG-2 IPMP system,

1) to carry IPMP Control Information including Tool List and Tool Container in the stream to indicate which IPMP Tools is used by content providers and content distributors, and how binary tool should be carried inside the content.
2) to define 5 new descriptors in MPEG-2 system to hold Tool List, IPMP Control Graph, IPMP Tool Container.
3) to define 2 new streams. IPNP Stream to carry IPMP information that is to be sent to each individual tool instance, and IPMP Control Information Stream to carry IPMP Control Information in Transport Stream.
4) to map the concept of Tool Manager, Message Router from MPEG-4 IPMP extension into MPEG-2 terminal.

[0039] The general diagram in Figure 2 is shown for our presented compliant MPEG-2 IPMP system.

[0040] Server is shown in module 2.1, it works as either Content Provider or Content distributor, or both functions for different application scenario.

[0041] Network layer is shown in module 2.3 for communication between Compliant MPEG-2 IPMP terminal and Server including transmission of content stream from the server to the terminal.

[0042] At first, Right Authentication in module 2.4 starts to interact with the server, to obtain the content access and consumption right, as well as the detail usage rule. If the right for content access is authorized in Module 2.4, the server will send the requested content stream to the terminal via the Network layer.

[0043] In module 2.2, Content stream is transmitted together with IPMP Control Information including Tool List and Tool Container, and IPMP Stream. The details of IPMP Control Information and IPMP Stream will be explained later.
IPMP Tools Manager shown in Module 2.5 is to parse/interpret IPMP Control Information. It parses the Tool List and finds out what are the IPMP tools needed to process the content. If there is any missing tool, the Tool Manager either retrieves the tool from the Tool Container or it retrieves the tool from somewhere else through a proprietary way. The Tool Manager is also responsible to select tool from a list of alternative choices, or to interpret the parametric description and choose its own favourite tool.

IPMP Tool Manager also parses the IPMP Control Graph to find out which tool is used upon which elementary stream at which control point. Details of this will be explained later.

The Licence/key, and Usage Rules are stored in the memory of the terminal as module 2.6 for further process. The binary IPMP Tools with their corresponding ToolIDs are stored in the memory of the terminal as module 2.7. Each of the Tools is built following the generic and standardised Interface and it is pre-compiled using the compiler for matching the platform. For example, the Tool of Data encryption and decryption can be built based on one generic and specified Interface. It can be pre-compiled into Java Byte Code (JBC) for all the platforms/terminals with Java Virtual Machine, and it also can be pre-compiled into Dynamic Link Library (DLL) for Windows based platforms/terminals.

Module 2.8 shows the Messaging Interfaces of IPMP Tools that are needed to be pre-defined for IPMP Tool providers and Terminal implementers to follow.

The detail explanation is split into four parts here, to indicate each invented item.

(1. IPMP Control Information)

IPMP Control Information needs to be carried in the content stream. IPMP Control Information contains necessary information like Tool List and Tool Container. The IPMP Tool List identifies, and enables selection of, the IPMP Tools required to process the Content. Tool Container enables the carriage of binary tool in content streams.

In short, the IPMP Control Information describes what are the IPMP tools needed to play the content, and how they protect the content. In Transport stream, it exists in the form of IPMP Control Information table. In Program Stream, it exists in the form of a PES packet when the stream_id is IPMP Control Information Stream ID.

(1.1 IPMP Control Info Table in Transport Stream)

An additional table "IPMP Control Information Table" should be included in PSI (Program Specific Information). This is used to hold IPMP control information including Tool Container and IPMP Tool List Descriptor that will be defined later on. The PID assignment is illustrated as below.

<table>
<thead>
<tr>
<th>Structure Name</th>
<th>Stream Type</th>
<th>Reserved PID #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program Association Table</td>
<td>ITU-T Rec. H222.0/ISO/IEC 13818-1</td>
<td>0x00</td>
<td>Associates Program Number and Program Map Table PID</td>
</tr>
<tr>
<td>Program Map Table</td>
<td>ITU-T Rec. H222.0/ISO/IEC 13818-1</td>
<td>Assigned in the PAT</td>
<td>Specifies PID values for components of one or more programs</td>
</tr>
<tr>
<td>Network Information Table</td>
<td>Private</td>
<td>Assigned in the PAT</td>
<td>Physical network parameters such as FDM frequencies, Transponder Numbers, etc.</td>
</tr>
<tr>
<td>IPMP Control Information Table</td>
<td>ITU-T Rec. H222.0/ISO/IEC 13818-1</td>
<td>0x03</td>
<td>Contains IPMP Tool List to describe the IPMP tools and how they protect the content.</td>
</tr>
<tr>
<td>Conditional Access Table</td>
<td>ITU-T Rec. H222.0/ISO/IEC 13818-1</td>
<td>0x01</td>
<td>Associates one or more (private) EMM streams each with a unique PID value</td>
</tr>
</tbody>
</table>

(1.1.1 Mapping of IPMP Control Information table into sections)

The IPMP Control Information table may be segmented into one or more sections, before insertion into Transport Stream packets, with the following syntax.
Table 2—IPMP Control Information Table section

<table>
<thead>
<tr>
<th>Syntax</th>
<th>No. of bits</th>
<th>Mnemonic</th>
</tr>
</thead>
<tbody>
<tr>
<td>table_id</td>
<td>8 uimsebf</td>
<td></td>
</tr>
<tr>
<td>section_syntax_indicator</td>
<td>1 bslbfl</td>
<td></td>
</tr>
<tr>
<td>'0'</td>
<td>1 bslbfl</td>
<td></td>
</tr>
<tr>
<td>reserved</td>
<td>2 bslbfl</td>
<td></td>
</tr>
<tr>
<td>section_length</td>
<td>12 uimsebf</td>
<td></td>
</tr>
<tr>
<td>reserved</td>
<td>2 bslbfl</td>
<td></td>
</tr>
<tr>
<td>ipmp_control_info_version</td>
<td>5 uimsebf</td>
<td></td>
</tr>
<tr>
<td>current_next_indicator</td>
<td>1 bslbfl</td>
<td></td>
</tr>
<tr>
<td>section_number</td>
<td>8 uimsebf</td>
<td></td>
</tr>
<tr>
<td>last_section_number</td>
<td>8 uimsebf</td>
<td></td>
</tr>
<tr>
<td>descriptor_length</td>
<td>24 uimsebf</td>
<td></td>
</tr>
</tbody>
</table>

for (i=0; i<N;i++) {

descryptor()
}

isSigned | 1 bslbfl |          |
| reserved | 7 bslbfl |          |

if (isSigned)

| Signature | 8 ByteArray |          |
| NumCerts | 8 uimsebf |          |

for (i=0; i<numCerts;i++) {

certType | 8 uimsebf |          |
| Certificate | 8 ByteArray |          |

Verifying_Tool_ID | 128 uimsebf |          |

CRC_32 | 32 rpchof |          |

Semantic definition of fields in IPMP Control Information Table section:

[0053] table_id -- This is an 8 bit field, which shall be always set to 0x02 as shown in table 1 above.

[0054] section_syntax_indicator -- The section_syntax_indicator is a 1 bit field which shall be set to ‘1’.

[0055] section_length -- This is a 20 bit field. It specifies the number of bytes of the section starting immediately following the section_length field, and including the CRC. The value in this field shall not exceed 1048573. The length is set to be a large value, because the following descriptors may contain Tool_Container_Descriptor that will be described later.

[0056] ipmp_control_info_version -- This 5 bit field is the version number of the whole IPMP Control Information Table.

The version number shall be incremented by 1 modulo 32 when a change in the information carried within the ipmp control info table occurs. When the current_next_indicator is set to ‘1’, then the version_number shall be that of the currently applicable IPMP Control Information Table. When the current_next_indicator is set to ‘0’, then the version_number shall be that of the next applicable IPMP Control Information Table.

[0057] current_next_indicator -- A 1 bit indicator, which when set to ‘1’ indicates that the IPMP Control Information Table sent is currently applicable. When the bit is set to ‘0’, it indicates that the IPMP Control Information Table sent is not yet applicable and shall be the next IPMP Control Information Table to become valid.

[0058] section_number -- This 8 bit field gives the number of this section. The section number of the first section in the IPMP Control Information Table shall be 0x00. It shall be incremented by 1 modulo 256 with each additional section.
in the IPMP Control Information Table.

[0059] last_section_number -- This 8 bit field specifies the number of the last section (that is, the section with the highest section number) of the IPMP Control Information Table.

[0060] descriptor_length - This 16 bit field specifies the total length of the descriptors immediately following this field. ToolList_Descriptor should be following this field. Details of the descriptors are given in section 3.

[0061] isSigned - This 1 bit field indicates the presence of a signature in the IPMP Control Information table.

[0062] Signature - The signature of the entire IPMP Control Information including Tool List descriptor and Tool Container descriptor.

[0063] CertType - The type of certification mechanism being used.

[0064] NumCerts - The number of certificates included.


[0066] Verifying_Tool__Id - The ID of the Tool that is required to verify the certificate(s). This may be the ID of the Terminal.

[0067] CRC_32 -- This is a 32 bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in Annex B in [1] after processing the entire IPMP section.

(1.2 IPMP Control Information in Program Stream)

[0068] IPMP Control Info provides overall IPMP Information including Tool List Descriptor in Program Stream. It is presented as a PES packet when the stream_id value is specified value.

<table>
<thead>
<tr>
<th>Syntax</th>
<th>No. of bits</th>
<th>Mnemonic</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFMP_control_info()</td>
<td></td>
<td></td>
</tr>
<tr>
<td>packet_start_code_prefix</td>
<td>24</td>
<td>bs1bf</td>
</tr>
<tr>
<td>imap_stream_id</td>
<td>8</td>
<td>uimsbf</td>
</tr>
<tr>
<td>imap_control_info_length</td>
<td>19</td>
<td>uimsbf</td>
</tr>
<tr>
<td>current_next_indicator</td>
<td>1</td>
<td>bs1bf</td>
</tr>
<tr>
<td>imap_control_info_version</td>
<td>5</td>
<td>uimsbf</td>
</tr>
<tr>
<td>reserved</td>
<td>7</td>
<td>bs1bf</td>
</tr>
<tr>
<td>marker_bit</td>
<td>16</td>
<td>bs1bf</td>
</tr>
<tr>
<td>descriptor_length</td>
<td>16</td>
<td>uimsbf</td>
</tr>
<tr>
<td>for (i=0; i&lt;N;i++)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>descriptor()</td>
<td></td>
<td></td>
</tr>
<tr>
<td>isSigned</td>
<td>1</td>
<td>bs1bf</td>
</tr>
<tr>
<td>reserved</td>
<td>7</td>
<td>bs1bf</td>
</tr>
<tr>
<td>if (isSigned)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Signature</td>
<td>8</td>
<td>ByteArray</td>
</tr>
<tr>
<td>NumCerts</td>
<td>8</td>
<td>uimsbf</td>
</tr>
<tr>
<td>for (i=0; i&lt;numCerts;i++)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CertType</td>
<td>8</td>
<td>uimsbf</td>
</tr>
<tr>
<td>Certificate</td>
<td>8</td>
<td>ByteArray</td>
</tr>
<tr>
<td>}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Verifying Tool ID</td>
<td>128</td>
<td>uimsbf</td>
</tr>
<tr>
<td>}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRC_32</td>
<td>32</td>
<td>rpchof</td>
</tr>
<tr>
<td>}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
(Semantic Definition of fields in IPMP Control Info)

- **packet_start_code_prefix** -- The packet_start_code_prefix is a 24-bit code. Together with the map_stream_id that follows it constitutes a packet start code that identifies the beginning of a packet. The packet_start_code_prefix is the bit string ‘0000 0000 0000 0000 0000 0001’ (0x000001 in hexadecimal)

- **map_stream_id** -- This is an 8 bit field whose value is always 0x?? in hexadecimal.

- **ipmp_control_info_version** -- This 5 bit field is the version number of the whole IPMP Control Information. The version number shall be incremented by 1 modulo 32 when a change in the information carried within the ipmp control info occurs. When the current_next_indicator is set to ‘1’, then the version_number shall be that of the currently applicable IPMP Control Information. When the current_next_indicator is set to ‘0’, then the version_number shall be that of the next applicable IPMP Control Information.

- **ipmp_control_info_length** -- The ipmp_control_info_length is a 19 bit field indicating the total number of bytes in the ipmp_control_info immediately following this field. The maximum value of the field is 524288 (bytes).

- **current_next_indicator** -- A 1 bit field, which when set to ‘1’ indicates that the IPMP Control Info sent is currently applicable. When the bit is set to ‘0’, it indicates that the IPMP Control Info sent is not yet applicable and shall be the next one to become valid.

- **isSigned** - This 1 bit field indicates the presence of a signature in the IPMP Control Information table. The following fields in the if bracket bears the same semantics as in the last section.

(2. New Descriptors)

- **Program and program element descriptors** are structures which may be used to extend the definitions of programs and program elements. All descriptors have a format which begins with an 8 bit tag value. The tag value is followed by an 8 bit descriptor length and data fields. The invention defines new IPMP Tool List Descriptor to hold the IPMP tool list, IPMP Control Graph Descriptor to represent the overall IPMP structure, and IPMP Tool Container Descriptor to carry binary IPMP tool inside the content.

- The following semantics apply to both descriptors defined in this invention as well as the existing descriptors in MPEG-2.

- **descriptor_tag** -- The descriptor_tag is an 8 bit field which identifies each descriptor. Its meaning is given in the following table. An ‘X’ in the TS or PS columns indicates the applicability of the descriptor to either the Transport Stream or Program Stream respectively. Five new descriptors are introduced in this invention.

<table>
<thead>
<tr>
<th>descriptor tag</th>
<th>TS</th>
<th>PS</th>
<th>Identification</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>n/a</td>
<td>n/a</td>
<td>Reserved</td>
</tr>
<tr>
<td>1</td>
<td>n/a</td>
<td>n/a</td>
<td>Reserved</td>
</tr>
<tr>
<td>2</td>
<td>X</td>
<td>X</td>
<td>video_stream_descriptor</td>
</tr>
<tr>
<td>3</td>
<td>X</td>
<td>X</td>
<td>audio_stream_descriptor</td>
</tr>
<tr>
<td>4</td>
<td>X</td>
<td>X</td>
<td>hierarchy_descriptor</td>
</tr>
<tr>
<td>19</td>
<td>X</td>
<td>X</td>
<td>IPMP Tool List Descriptor</td>
</tr>
<tr>
<td>20</td>
<td>X</td>
<td>X</td>
<td>IPMP Tool Information Descriptor</td>
</tr>
<tr>
<td>21</td>
<td>X</td>
<td>X</td>
<td>IPMP Parametric Descriptor</td>
</tr>
<tr>
<td>22</td>
<td>X</td>
<td>X</td>
<td>IPMP Tool Container Descriptor</td>
</tr>
<tr>
<td>23</td>
<td>X</td>
<td>X</td>
<td>IPMP Descriptor</td>
</tr>
<tr>
<td>24-63</td>
<td>n/a</td>
<td>n/a</td>
<td>ITU-T Rec. H.222.0</td>
</tr>
</tbody>
</table>
[0081] descriptor_length -- The descriptor_length is an 8 bit field specifying the number of bytes of the descriptor immediately following descriptor_length field.

(2.1 IPMP Tool List Descriptor)

[0082] The IPMP Tool List Descriptor includes a list of IPMP tools. It is used to specify all IPMP tools that should be used in order to play back the content.

<table>
<thead>
<tr>
<th>descriptor tag</th>
<th>TS</th>
<th>PS</th>
<th>Identification</th>
</tr>
</thead>
<tbody>
<tr>
<td>64-255</td>
<td>n/a</td>
<td>n/a</td>
<td>User Private</td>
</tr>
</tbody>
</table>

Table 5 - IPMP Tool List descriptor

<table>
<thead>
<tr>
<th>Syntax</th>
<th>No. of bits</th>
<th>Mnemonic</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMPPToolList_descriptor() {</td>
<td></td>
<td></td>
</tr>
<tr>
<td>descriptor_tag</td>
<td>8</td>
<td>uimsbf</td>
</tr>
<tr>
<td>descriptor_length</td>
<td>16</td>
<td>uimsbf</td>
</tr>
<tr>
<td>numTools</td>
<td>8</td>
<td>uimsbf</td>
</tr>
<tr>
<td>for (i=0; i&lt;numTools; i++) {</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IpmpTool_Descriptor()</td>
<td></td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

IpmpTool_Descriptor() is defined in the following section.

(2.2 IPMP Tool Descriptor)

[0083] IPMP_Tool_Descriptor contains information for a logical IPMP Tool required by the Terminal. The logical tool may be one of the following:

1. A vendor-specific IPMP Tool specified by IPMP ToolID,
2. One of a list of alternate IPMP Tools,
3. An IEMP Tool specified by a parametric description.
In the case of a list of alternate tools, the Terminal shall select an IPMP Tool from the list of alternate IPMP Tools. In the case of a parametric description of the IPMP Tool, the Terminal shall select an IPMP Tool that meets the criteria specified in the parametric description.

**IPMP Tool ID** - the identifier of the logical IPMP Tool required by the Terminal.

**isAltGroup** - IEMP_Tool contains a list of alternate IPMP Tools. In this case, IPMP_ToolID is an identifier for the list of alternate IPMP Tools, and the Terminal shall route information specified in the bitstream for IPMP_ToolID to the specific IPMP Tool instantiated by the Terminal.

**numAlternates** - the number of alternate IPMP Tools specified in IPMP_Tool[].

**Alt_IPMP_ToolIDs** - an array of the IDs of alternate IPMP Tools that can allow consumption of the content.

**isParametric** - IPMP_Tool contains a parametric description of an IPMP Tool. In this case, IPMP_ToolID is an identifier for the parametrically described IPMP Tool, and the Terminal shall route information specified in the bitstream for IPMP_ToolID to the specific IPMP Tool instantiated by the terminal.

(2.3 IPMP Tool Container Descriptor)

There are many cases whereby content itself carried the binary IPMP tool (light weighted). The terminal may retrieve the IPMP tool from the content, load it, instantiate it and immediately use it in order to play out the content.

In MPEG-4 IPMP extension, binary IPMP tools are carried in tool ES. However, in MPEG-2 context, it could be easier to carry the binary IPMP tool inside a newly defined IPMP Tool Container Descriptor. One implementation of a given tool is carried as the payload of one IPMP Tool Container, the representation format, packaging information and IPMP Tool ID of which is also specified in the container.
(Semantic definitions of fields in IPMP Tool Container Descriptor)

<table>
<thead>
<tr>
<th>Syntax</th>
<th>No. of bits</th>
<th>Mnemonic</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPMPToolContainer_descriptor() {</td>
<td></td>
<td></td>
</tr>
<tr>
<td>descriptor_tag</td>
<td>8</td>
<td>uimsbf</td>
</tr>
<tr>
<td>descriptor_length</td>
<td>16</td>
<td>uimsbf</td>
</tr>
<tr>
<td>IPMP_Tool_ID</td>
<td>128</td>
<td>uimsbf</td>
</tr>
<tr>
<td>Tool_Format_ID</td>
<td>32</td>
<td>uimsbf</td>
</tr>
<tr>
<td>Tool_Package_ID</td>
<td>16</td>
<td>uimsbf</td>
</tr>
<tr>
<td>sizeofTool</td>
<td>16</td>
<td>uimsbf</td>
</tr>
<tr>
<td>for ( i=0; i&lt;sizeofTool; i++) {</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Toolbody</td>
<td>8</td>
<td>uimsbf</td>
</tr>
<tr>
<td>}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[0092] IPMP_Tool_ID - the ID of the Tool carried in this stream.

[0093] Tool_Format_ID - This is defined as 0x0001 for a structurally described tool. Otherwise, the Tool_Format_ID indicates the Binary Representation of the Tool and is ned by a registration authority.

[0094] Note: A structurally described tool implies a description of the IPMP Tool in terms of a network of primitives that can be combined to provide some or all IPMP functionalities required for content consumption. For example, a DES decryption algorithm could be described as a sequence of opcodes calls receiving the ciphertext as input and providing the plaintext as output.

[0095] Tool_Package_Id indicates the details of the package of the Tool - examples are CAB or a Winzip self-install executable. Values are assigned by a registration authority.

(2.4 IPMP Control Graph Descriptor)

[0096] IPMP Control Graph descriptor contains description of the entire IPMP protection scheme. It associates IPMP tool with each individual stream under its protection.
Semantic Definition of fields in IPMP Control Graph Descriptor:

**numProtectedPrograms** - This 8 bit field indicates how many programs are under IPMP protection scope. If the number is 0, it means it is a program stream. If it is greater than 0, it means it is a transport stream, and a for loop follows to trace into each program.

**Transport Stream Case (numProtectedPrograms > 0):**

** program_number ** - Program_number is a 16 bit field. It specifies the program that is under protection by IPMP. This field shall not take any single value more than once within one version of the IPMP Control Information.

**ipmp_length** - This is a 16 bit field. It specifies the number of bytes of the IPMP descriptors immediately following the ipmp_length field.

**NumProtectedStreams** - Specifies number of elementary streams (belonging to the above program) that are under the protection by IPMP.

**stream_type** - This is an 8 bit field specifying the type of program element carried within the packets with the PID whose value is specified by the elementary_PID. The values of stream_type are specified in table 11 described later.
elementary_PID -- This is a 13 bit field specifying the PID of the Transport Stream packets which carry the associated program element. If there is a IPMP Descriptor following immediately after this elementary_PID, it means that this particular elementary stream is under the protection scope defined by this IPMP Descriptor.

Program Stream Case (numProtectedPrograms = 0):

elementary_stream_id -- The elementary_stream_id is an 8 bit field indicating the value of the stream_id field in the PES packet headers of PES packets in which this elementary stream is stored.

IPMP Descriptor is further defined below.

(2.5 IPMP Descriptor)

IPMP Descriptor specifies the IPMP protection at a particular scope. Including specification of control points, sequencing, IPMP tool IDs, etc.

Table 9 - IPMP Descriptor

<table>
<thead>
<tr>
<th>Syntax</th>
<th>No. of bits</th>
<th>Mnemonic</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPMP_descriptor() {</td>
<td>8</td>
<td>uimsbf</td>
</tr>
<tr>
<td>descriptor_tag</td>
<td>8</td>
<td>uimsbf</td>
</tr>
<tr>
<td>descriptor_length</td>
<td>8</td>
<td>uimsbf</td>
</tr>
<tr>
<td>IPMP_DescriptorID</td>
<td>8</td>
<td>uimsbf</td>
</tr>
<tr>
<td>IPMP_ToolID</td>
<td>128</td>
<td>uimsbf</td>
</tr>
<tr>
<td>numControlPoints</td>
<td>8</td>
<td>uimsbf</td>
</tr>
<tr>
<td>for ( i=0; i&lt;numControlPoints; i++)</td>
<td>8</td>
<td>uimsbf</td>
</tr>
<tr>
<td>controlPoint</td>
<td>6</td>
<td>uimsbf</td>
</tr>
<tr>
<td>reserved</td>
<td>2</td>
<td>uimsbf</td>
</tr>
<tr>
<td>sequenceCode</td>
<td></td>
<td></td>
</tr>
<tr>
<td>if</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( sequenceCode==0x1 )</td>
<td>8</td>
<td>uimsbf</td>
</tr>
<tr>
<td>IPMP_DescriptorID</td>
<td>8</td>
<td>uimsbf</td>
</tr>
<tr>
<td>controlPoint</td>
<td></td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>if</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( sequenceCode==0x2 )</td>
<td>8</td>
<td>uimsbf</td>
</tr>
<tr>
<td>IPMP_DescriptorID</td>
<td>8</td>
<td>uimsbf</td>
</tr>
<tr>
<td>controlPoint</td>
<td></td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpaqueData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ByteArray</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(Semantic Definitions of Fields in IPMP Descriptor)

IPMP_DescriptorID - a unique ID of this IPMP descriptor. This could be used to refer to this particular descriptor (protection scope).

IEMP_ToolID - Unique ID of the IPMP Tool that is protecting in this scope.

NumControlPoints - number of Control points at which the IPMP Tool is active.

controlPoint - value specifying the IPMP control point at which the IPMP Tool resides, and is one of the following values:
sequenceCode - value specifying the relation of the IPMP Tool to IPMP Tool(s) residing at the same control point, and is one of the following. If the sequenceCode is either 0x01 or 0x02, a IPMP Descriptor ID together with a controlPoint follows immediately, to specify which tool (instance) is this current IPMP tool preceding or following.

<table>
<thead>
<tr>
<th>Sequence Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>IPMP Tool is not sequenced</td>
</tr>
<tr>
<td>0x01</td>
<td>IPMP Tool precedes another tool</td>
</tr>
<tr>
<td>0x02</td>
<td>IPMP Tool follows another tool</td>
</tr>
<tr>
<td>0x03</td>
<td>A logical &quot;OR&quot; of 0x01 and 0x02</td>
</tr>
</tbody>
</table>

OpaqueData -- opaque data to control the IPMP Tool.

Stream_id specifies the type and number of the elementary stream as defined by the table below. Stream_id 1111 1001 is assigned to IPMP Stream in this embodiment.

<table>
<thead>
<tr>
<th>stream_id</th>
<th>stream coding</th>
</tr>
</thead>
<tbody>
<tr>
<td>1011 1100</td>
<td>program_stream_map</td>
</tr>
<tr>
<td>1011 1101</td>
<td>private_stream_1</td>
</tr>
<tr>
<td>1011 1110</td>
<td>padding_stream</td>
</tr>
<tr>
<td>1011 1111</td>
<td>private_stream_2</td>
</tr>
<tr>
<td>110x xxxx</td>
<td>ISO/IEC 13818-3 or ISO/IEC 11172-3 audio stream number x xxxx</td>
</tr>
<tr>
<td>1110 xxxx</td>
<td>ITU-T Rec. H.262 ISO/IEC 13818-2 or ISO/IEC 11172-2 video stream number xxxx</td>
</tr>
<tr>
<td>1111 0000</td>
<td>ECM_stream</td>
</tr>
<tr>
<td>1111 0001</td>
<td>EMM_stream</td>
</tr>
<tr>
<td>1111 1001</td>
<td>ancillary_stream (IPMP Stream)</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1111 1100 ... 1111 1110</td>
<td>reserved data stream</td>
</tr>
<tr>
<td>1111 1111</td>
<td>program_stream_directory</td>
</tr>
</tbody>
</table>

Value Description

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>ITU-T ISO/IEC Reserved</td>
</tr>
<tr>
<td>0x01</td>
<td>ISO/IEC 11172 Video</td>
</tr>
<tr>
<td>0x02</td>
<td>ITU-T Rec. H.262 ISO/IEC 13818-2 Video or ISO/IEC 11172-2 constrained parameter video stream</td>
</tr>
<tr>
<td>0x03</td>
<td>ISO/IEC 11172 Audio</td>
</tr>
</tbody>
</table>
The IPMP Stream is a new elementary stream that is to carry IPMP information. Unlike MPEG-4 IPMP extension, where there can be many IPMP Elementary steams in one content, with each IPMP ES associated with one IPMP system, in MPEG-2, all IPMP information for all IPMP tools sitting on all control points are carried in a single IPMP Stream. Hence, there is a need to indicate the clear destination in every piece of IPMP information in IPMP Stream.

It is defined in this invention that IPMP Stream should be a concatenation of IPMP info messages, with the syntax defined below:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>No. of bits</th>
<th>Mnemonic</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPMP_info_message()</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ipmp_descriptor_id</td>
<td>8</td>
<td>uimsbf</td>
</tr>
<tr>
<td>control_point</td>
<td>8</td>
<td>uimsbf</td>
</tr>
<tr>
<td>length_of_message</td>
<td>16</td>
<td>uimsbf</td>
</tr>
<tr>
<td>for ( i=0; i&lt;N; i++) {</td>
<td></td>
<td></td>
</tr>
<tr>
<td>message</td>
<td>8</td>
<td>uimsbf</td>
</tr>
<tr>
<td>}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The ipmp_descriptor_id and control_point together clearly defines the destination of this IPMP_info_message. This message should be routed by the message router to the IPMP Tool defined in the corresponding ipmp descriptor sitting at the specified control_point.

IPMP Tool Manager and Message Router can be directly mapped from MPEG-4 IPMP Extension Terminal to MPEG-2 IPMP terminal. Figure 3 shows the architecture of a MPEG-2 IPMP terminal.

The IPMP Tool Manager is a conceptual entity in a given IPMP Terminal. Upon receipt of the Tool List, the Terminal should route the same to the IPMP Tool Manager for parsing and Tool retrieval. The Tool Manager also processes parametric descriptions, resolves alternative tools, and receives binary Tools that arrive in the Content. The following steps detail the process of parsing and retrieval of Tools in an MPEG-2 Terminal.

1. The IPMP Tool List Descriptor arrives in the IPMP Control Information Table in PSI and is routed to the Tool Manager.
2. The IPMP Tool Manager parses information for the IPMP Tools as per the syntax in clause 2.2.2.1.
3. The Tool Manager checks if the required Tools are available. For each unavailable Tool, an attempt to obtain the Tool may be made. How to obtain the missing tool is an implementation issue.

4. The IPMP Tool Manager is also responsible for parsing the IPMP Tool Container Descriptor and retrieving the binary IPMP Tool that is carried inside PSI.

5. The IPMP Tool Manager is further responsible for resolving parametric descriptions.

(4.2 IPMP Message Router)

[0121] All IPMP Tool Messages are routed through the Terminal. To represent this function, an entity called the Message Router is defined in the architecture. The MR connects and communicates with supported IPMP Tool(s). It thus abstracts the physical interface of one IPMP Tool from any other IPMP Tool that wishes to communicate with it. Message Routing is assumed to be instantaneous. In case of an MR error, an appropriate error status is returned by the MR. In all other cases, the MR is required to route, without a change in semantic meaning, information and responses as received.

[0122] Messaging interface can be mapped from MPEG-4 IPMP Extension without modification. However, there is no need to define a context ID for tool instances under MPEG-2. IEMP Descriptor ID together with control point should clearly define a specific tool instance running at a specific control point protecting a specific elementary stream.

(4.3 Mutual Authentication)

[0123] Tools that must communciate with one another or with the Terminal must do so in a way that meets the security requirements of the Tools and the Terminal. Tools must establish trust with the Terminal and possibly with one another to enable secure communication. Support for the establishment of a communication channel that reflects the nature of inter-tool trust can be accomplished via the use of secure, trusted authenticated channels.

[0124] Messages supporting the mutual authentication can be directly mapped from MPEG-4 IPMP extension to MPEG-2 framework.

[0125] Although the present invention has been described in connection with specified embodiments thereof, many other modifications, corrections and applications are apparent to those skilled in the art. Therefore, the present invention is not limited by the disclosure provided herein but limited only to the scope of the appended claims.

Claims

1. Apparatus of a flexible and common Intellectual Property Management and Protection (IPMP) System for MPEG-2 content distribution and protection at the side of a content provider, comprising:

   an encoder that encodes a content using coding technology into a content stream (2.2),

   characterised by:

   an encrypter that encrypts the content stream (2.2) as an encrypted content stream using at least one of a plurality of IPMP Tools at a particular control point, wherein the particular control point defines a position along the data flow at the side of the content provider between the encoder and a section for multiplexing, and wherein each IPMP Tool serves as a module for providing a predetermined function alone or in combination;

   a section that creates an IPMP Tool List containing at least one of the plurality of IPMP tools which is used for encrypting the content stream;

   a section that creates an IPMP Control Graph that indicates how the at least one IPMP tool protects the contents stream including information that indicates the particular control point at which the at least one of the plurality of IPMP tools is used for encrypting the content stream; and

   a section that multiplexes the IPMP Tool List and the IPMP Control Graph with the encrypted content stream (2.2) into a MPEG-2 stream using MPEG-2 system.

2. The apparatus according to claim 1, further comprising a watermark section that embeds watermark information in the content stream using an watermarking tool.

3. The apparatus according to claim 1, further comprising a section that creates an IPMP Tool Container to carry at least one of the plurality of IPMP Tools in the content stream (2.2).
4. The apparatus according to claim 1, further comprising:

section that creates an IPMP stream to carry time variant IPMP related information that is to be sent to each individual tool instance during content consumption at the side of a terminal; and

section that multiplexes the IPMP stream into the MPEG-2 stream (2.2).

5. Apparatus of a flexible and common Intellectual Property Management and Protection (IPMP) system for MPEG-2 content distribution and protection at the side of a terminal, comprising:

a plurality of control points, wherein at least one of a plurality of IPMP Tools is used at a particular control point of the plurality of control points, wherein the particular control point defines a position along the data flow at the side of the terminal between a demultiplexer and a section for rendering a content and each IPMP Tool serves as a module for providing a predetermined function alone or in combination; wherein the de-multiplexer de-multiplexes an encrypted content stream, an IPMP Tool List and an IPMP Control Graph from a MPEG-2 stream (2.2) sent from a content provider (2.1);
a section (2.5) that interprets the IPMP Tool List containing at least one of the plurality of IPMP tools; and

a section (2.5) that interprets the IPMP Control Graph that indicates how the at least one IPMP tool protects the content stream including information that indicates the particular control point of the plurality of control points at which the at least one of the plurality of IPMP tools is to be used for decrypting the encrypted content stream.

6. The apparatus according to claim 5, further comprising a section that performs at least one of watermark retrieving and watermark embedding, through an interface at at least one of the plurality of control points.

7. The apparatus according to claim 6, further comprising a section (2.5) that interprets parametric description of the IPMP Tool List and alternative tools to make a tool selection based on the interpretation result.

8. The apparatus according to claim 5, further comprising a section (2.5) that interprets parametric description of the IPMP Tool List and alternative tools to make a tool selection based on the interpretation result, and

a section (2.5) that retrieves IPMP Tools from an IPMP Tool Container carrying the IPMP Tools within IPMP Control Information to obtain the associated Tool ID, Tool Format ID and Tool Package ID of each of the IPMP Tools.

9. The apparatus according to claim 5, further comprising a router (2.8) that routes time variant IPMP related message to a specific IPMP Tool instance or the terminal itself.

10. The apparatus according to claim 5, further comprising a section that implements IPMP Tools in an embedded MPEG-2 IPMP terminal.

11. The apparatus according to claim 5, wherein the MPEG-2 stream (2.2) contains IPMP related streams including IPMP Control Information stream and IPMP stream, and the IPMP Control Information stream includes the IPMP Tool List and IPMP Tool Container carrying IPMP Tools.

12. The apparatus according to claim 5, wherein MPEG-2 streams (2.2) are protected by IPMP tools, and contain information of the IPMP tools such as Tool ID, Tool Location, the IPMP Tool List and the IPMP Control Graph.

13. The apparatus according to claim 5, wherein MPEG-2 streams (2.2) are protected by IPMP tools, and some of the IPMP tools are carried in content streams (2.2) by a defined IPMP Tool container.

14. The apparatus according to claim 5, wherein MPEG-2 streams (2.2) are protected by a few IPMP tools, and some missing IPMP tools in the terminal can be retrieved from a specified location.

15. The apparatus according to claim 5, wherein MPEG-2 streams (2.2) are protected by a few IPMP tools, and the IPMP tools are pre-implemented in the terminal.

16. The apparatus according to claim 5, wherein the MPEG-2 stream is processed by using IPMP information which is carried in the same MPEG-2 stream, such as the IPMP Tool List, the IPMP Control Graph, an IPMP Tool Container, and an IPMP stream.
The apparatus according to claim 5, wherein the MPEG-2 stream is processed first by interpreting an IPMP Control Information which is carried in the MPEG-2 stream, parsing the IPMP Tool List, interpreting the IPMP Control Graph, retrieving missing tools, applying associated IPMP tools to audio and video streams at the particular control points.

The apparatus according to claim 5, wherein a MPEG-2 stream is processed first by a conceptual IPMP manager to obtain the necessary IPMP Tools and apply them to associated audio and video streams.

Patentansprüche

1. Vorrichtung eines flexiblen und gemeinsamen Intellectual Property Management und Protection (IPMP) Systems für Verteilung und Schutz von MPEG-2 Inhalten auf der Seite eines Anbieters von Inhalten mit:

einem Kodierer, der einen Inhalt unter Verwendung einer Kodieretechnologie in einen Datenstrom von Inhalten (2.2) kodierte, gekennzeichnet durch:

   einen Verschlüssler, der unter Verwendung wenigstens eines von einer Vielzahl von IPMP Werkzeugen an einem bestimmten Kontrollpunkt den Datenstrom von Inhalten (2.2) in einen verschlüsselten Datenstrom von Inhalten verschlüsselt, wobei der bestimmte Kontrollpunkt eine Position entlang des Datenflusses zwischen dem Kodierer und einem Abschnitt zum Multiplexen auf der Seite des Anbieters von Inhalten definiert, und wobei jedes IPMP Werkzeug als ein Modul zum Bereitstellen einer vorbestimmten Funktion allein oder in Kombination dient;

   einen Abschnitt, der eine IPMP Werkzeugliste erzeugt, die wenigstens eines der Vielzahl von IPMP Werkzeugen aufweist, welches zum Verschlüsseln des Datenstromes von Inhalten verwendet wird;

   einen Abschnitt, der einen IPMP Kontrollgraphen erzeugt, der anzeigen, wie das wenigstens eine IPMP Werkzeug den Datenstrom von Inhalten schützt und Informationen aufweist, die den bestimmten Kontrollpunkt anzeigen, an dem das wenigstens eine der Vielzahl von IPMP Werkzeugen zur Verschlüsselung des Datenstromes von Inhalten verwendet wird; und

   einen Abschnitt, der unter Verwendung eines MPEG-2 Systems die IPMP Werkzeugliste und den IPMP Kontrollgraphen mit dem verschlüsselten Datenstrom von Inhalten (2.2) in einen MPEG-2 Datenstrom multiplex.


3. Vorrichtung nach Anspruch 1, ferner mit einem Abschnitt, der einen IPMP Werkzeugcontainer zum Aufnehmen wenigstens eines der Vielzahl von IPMP Werkzeugen in dem Datenstrom von Inhalten (2.2) erzeugt.

4. Vorrichtung nach Anspruch 1, ferner mit:

   einem Abschnitt, der einen IPMP Datenstrom zum Aufnehmen einer zeitveränderlichen IPMP bezogenen Information erzeugt, die zu jeder individuellen Werkzeuginstanz während eines Verbrauchs eines Inhaltes auf der Seite eines Terminals zu senden ist; und

   einem Abschnitt, der den IPMP Datenstrom in den MPEG-2 Datenstrom (2.2) multiplex.

5. Vorrichtung eines flexiblen und gemeinsamen Intellectual Property Management und Protection (IPMP) Systems für Verteilung und Schutz von MPEG-2 Inhalten auf der Seite eines Terminals mit:

   einer Vielzahl von Kontrollpunkten, wobei wenigstens eines einer Vielzahl von IPMP Werkzeugen an einem bestimmten Kontrollpunkt der Vielzahl von Kontrollpunkten verwendet wird, wobei der bestimmte Kontrollpunkt eine Position entlang des Datenflusses zwischen einem Demultiplexer und einem Abschnitt zum Wiedergeben eines Inhaltes auf der Seite des Terminals definiert, und jedes IPMP Werkzeug als ein Modul zum Bereitstellen einer vorbestimmten Funktion allein oder in Kombination dient;

   wobei der Demultiplexer einen verschlüsselten Datenstrom von Inhalten, eine IPMP Werkzeugliste und einen IPMP Kontrollgraphen aus einem MPEG-2 Datenstrom (2.2) demultiplex, der von einem Anbieter von Inhalten (2.1) gesendet ist;

   einem Abschnitt (2.5), der die IPMP Werkzeugliste interpretiert, die wenigstens eines der Vielzahl von IPMP Werkzeugen aufweist; und
einem Abschnitt (2.5), der den IPMP Kontrollgraphen interpretiert, der anzeigen, wie das wenigstens eine IPMP Werkzeug den Datenstrom von Inhalten schützt und Informationen aufweist, die den bestimmten Kontrollpunkt der Vielzahl von Kontrollpunkten ansagt, an dem das wenigstens eine der Vielzahl von IPMP Werkzeugen zum Entschlüsseln des verschlüsselten Datenstromes von Inhalten zu verwenden ist.


7. Vorrichtung nach Anspruch 6, ferner mit einem Abschnitt (2.5), der eine parametrische Beschreibung der IPMP Werkzeugliste und alternativen Werkzeugen interpretiert, um eine Werkzeugauswahl basierend auf dem Interpretationsergebnis zu machen.

8. Vorrichtung nach Anspruch 5, ferner mit:

   einem Abschnitt (2.5), der eine parametrische Beschreibung der IPMP Werkzeugliste und alternativen Werkzeuge interpretiert, um eine Werkzeugauswahl basierend auf dem Interpretationsergebnis zum machen, und einem Abschnitt (2.5), der IPMP Werkzeuge von einem IPMP Werkzeugcontainer erhält, der die IPMP Werkzeuge innerhalb einer IPMP Kontrollinformation aufnimmt, um die zugeordnete Werkzeug ID, Werkzeugformat ID und Werkzeugpaket ID eines jeden der IPMP Werkzeuge zu erhalten.

9. Vorrichtung nach Anspruch 5, ferner mit einem Vermittlungsknoten (2.8), der eine zeitveränderliche IPMP bezogene Nachricht zu einer spezifischen IPMP Werkzeuginstanz oder dem Terminal selbst leitet.


11. Vorrichtung nach Anspruch 5, wobei der MPEG-2 Datenstrom (2.2) IPMP bezogene Datenströme mit einem IPMP Kontrollinformationsdatenstrom und einen IPMP Datenstrom aufweist, und der IPMP Kontrollinformationsdatenstrom die IPMP Werkzeugliste und IPMP Werkzeugcontainer aufweist, der IPMP Werkzeuge aufnimmt.

12. Vorrichtung nach Anspruch 5, wobei MPEG-2 Datenströme (2.2) durch IPMP Werkzeuge geschützt sind und Informationen der IPMP Werkzeuge wie Werkzeug ID, Werkzeugort, die IPMP Werkzeugliste und den IPMP Kontrollgraphen aufweisen.

13. Vorrichtung nach Anspruch 5, wobei MPEG-2 Datenströme (2.2) durch IPMP Werkzeuge geschützt sind und einige der IPMP Werkzeuge in Datenströmen von Inhalten (2.2) durch einen definierten IPMP Werkzeugcontainer aufgenommen sind.

14. Vorrichtung nach Anspruch 5, wobei MPEG-2 Datenströme (2.2) durch wenige IPMP Werkzeuge geschützt sind und einige in dem Terminal fehlende IPMP Werkzeuge von einem spezifischen Ort erhalten werden können.

15. Vorrichtung nach Anspruch 5, wobei MPEG-2 Datenströme (2.2) durch wenige IPMP Werkzeuge geschützt sind und die IPMP Werkzeuge in dem Terminal vorimplementiert sind.

16. Vorrichtung nach Anspruch 5, wobei der MPEG-2 Datenstrom unter Verwendung von IPMP Informationen verarbeitet wird, die in demselben MPEG-2 Strom wie die IPMP Werkzeugliste, der IPMP Kontrollgraph, ein IPMP Werkzeugcontainer und ein IPMP Datenstrom aufgenommen ist.


18. Vorrichtung nach Anspruch 5, wobei ein MPEG-2 Datenstrom zuerst durch einen konzeptuellen IPMP Manager verarbeitet wird, um die nötigen IPMP Werkzeuge zu erhalten und diese auf zugeordnete Audio- und Videodatenströme anzuwenden.
Revendications

1. Appareil d’un Système flexible et commun de Gestion et de Protection de la Propriété Intellectuelle (GPPI) pour la distribution et la protection de contenus MPEG-2 du côté d’un fournisseur de contenus, comprenant :

   un encodeur qui code un contenu en utilisant une technologie de codage en un flux de contenu (2.2),

   caractérisé par

   un crypteur qui crypte le flux de contenu (2.2) en un flux de contenu crypté en utilisant au moins l’un d’une pluralité d’Outils de GPPI à un point de contrôle particulier, où le point de contrôle particulier définit une position sur le flux de données du côté du fournisseur de contenus entre l’encodeur et une section de multiplexage, et où chaque Outil de GPPI sert en tant que module pour prévoir une fonction prédéterminée seule ou en combinaison ;

   une section qui crée une Liste d’Outils de GPPI contenant au moins un de la pluralité d’outils de GPPI qui est utilisé pour crypter le flux de contenu ;

   une section qui crée un Graphique de Contrôle de GPPI qui indique comment l’au moins un de la pluralité d’outils de GPPI protège le flux de contenu incluant des informations qui indiquent le point de contrôle particulier auquel l’au moins un de la pluralité d’outils de GPPI est utilisé pour crypter le flux de contenu ; et

   une section qui multiplexe la Liste d’Outils de GPPI et le Graphique de Contrôle de GPPI avec le flux de contenu crypté (2.2) en un flux MPEG-2 en utilisant le système MPEG-2.

2. Appareil selon la revendication 1, comprenant en outre une section de tatouage qui incorpore des informations de tatouage dans le flux de contenu en utilisant un outil de tatouage.

3. Appareil selon la revendication 1, comprenant en outre une section qui crée une Boîte à Outils de GPPI pour contenir au moins un de la pluralité d’Outils de GPPI dans le flux de contenu (2.2).

4. Appareil selon la revendication 1, comprenant en outre :

   une section qui crée un flux de GPPI pour contenir des informations associées à la GPPI de variante temporelle qui doivent être envoyées à chaque instance d’outil individuelle pendant la consommation de contenu du côté d’un terminal ; et

   une section qui multiplexe le flux de GPPI dans le flux MPEG-2 (2.2).

5. Appareil d’un système flexible et commun de Gestion et de Protection de la Propriété Intellectuelle (GPPI) pour la distribution et la protection de contenus MPEG-2 du côté d’un terminal, comprenant :

   une pluralité de points de contrôle, où au moins un d’une pluralité d’Outils de GPPI est utilisé à un point de contrôle particulier de la pluralité de points de contrôle où le point de contrôle particulier définit une position sur le flux de données du côté du terminal entre un démultiplexeur et une section de rendu d’un contenu, et chaque Outil de GPPI sert en tant que module pour prévoir une fonction prédéterminée seule ou en combinaison ;

   où le démultiplexeur démultiplexe un flux de contenu crypté, une Liste d’Outils de GPPI et un Graphique de Contrôle de GPPI d’un flux MPEG-2 (2.2) envoyé par un fournisseur de contenus (2.1) ;

   une section (2.5) qui interprète la Liste d’Outils de GPPI contenant au moins un de la pluralité d’outils de GPPI ; et

   une section (2.5) qui interprète le Graphique de Contrôle de GPPI qui indique comment l’au moins un outil de GPPI protège le flux de contenu incluant des informations qui indiquent le point de contrôle particulier de la pluralité de points de contrôle auquel l’au moins un de la pluralité d’outils de GPPI doit être utilisé pour décrypter le flux de contenu crypté.

6. Appareil selon la revendication 5, comprenant en outre une section qui effectue au moins l’une de la récupération du tatouage et de l’incorporation du tatouage, par l’intermédiaire d’une interface à moins un de la pluralité de points de contrôle.

7. Appareil selon la revendication 6, comprenant en outre une section (2.5) qui interprète la description paramétrique de la Liste d’Outils de GPPI et autres outils pour faire une sélection d’outils sur la base du résultat de l’interprétation.

8. Appareil selon la revendication 5, comprenant en outre une section (2.5) qui interprète la description paramétrique de la Liste d’Outils de GPPI et autres outils pour faire une sélection d’outils sur la base du résultat de l’interprétation, et
une section (2.5) qui récupère les Outils de GPPI d’une Boîte à Outils de GPPI contenant les Outils de GPPI dans les Informations de Contrôle de GPPI pour obtenir l’ID d’Outil, l’ID de Format d’Outil et l’ID de Groupe d’Outils associés de chacun des Outils de GPPI.

9. Appareil selon la revendication 5, comprenant en outre un routeur (2.8) qui achemine le message associé à la GPPI de variante temporelle à une instance d’Outil de GPPI spécifique ou au terminal lui-même.

10. Appareil selon la revendication 5, comprenant en outre une section qui met en œuvre les Outils de GPPI dans un terminal de GPPI MPEG-2 incorporé.

11. Appareil selon la revendication 5, dans lequel le flux MPEG-2 (2.2) contient des flux associés à la GPPI incluant un flux d’Informations de Contrôle de GPPI et un flux de GPPI, et le flux d’Informations de Contrôle de GPPI inclut la Liste d’Outils de GPPI et la Boîte à Outils de GPPI contenant les Outils de GPPI.

12. Appareil selon la revendication 5, dans lequel les flux MPEG-2 (2.2) sont protégés par les outils de GPPI, et contiennent les informations des outils de GPPI telles que l’ID d’Outil, l’Emplacement de l’Outil, la Liste d’Outils de GPPI et le Graphique de Contrôle de GPPI.

13. Appareil selon la revendication 5, dans lequel les flux MPEG-2 (2.2) sont protégés par les outils de GPPI, et certains des outils de GPPI sont contenus dans des flux de contenu (2.2) par une boîte à Outils de GPPI définie.

14. Appareil selon la revendication 5, dans lequel les flux MPEG-2 (2.2) sont protégés par quelques outils de GPPI, et certains outils de GPPI manquants dans le terminal peuvent être récupérés à partir d’un emplacement spécifié.

15. Appareil selon la revendication 5, dans lequel les flux MPEG-2 (2.2) sont protégés par quelques outils de GPPI, et les outils de GPPI sont pré-mis en œuvre dans le terminal.

16. Appareil selon la revendication 5, dans lequel le flux MPEG-2 est traité en utilisant les informations de GPPI qui sont contenues dans le même flux MPEG-2, telles que la Liste d’Outils de GPPI, le Graphique de Contrôle de GPPI, une Boîte à Outils de GPPI, et un flux de GPPI.

17. Appareil selon la revendication 5, dans lequel le flux MPEG-2 est traité d’abord en interprétant des Informations de Contrôle de GPPI qui sont contenues dans le flux MPEG-2, en effectuant une analyse syntaxique de la Liste d’Outils de GPPI, en interprétant le Graphique de Contrôle de GPPI, en récupérant les outils manquants, en appliquant des outils de GPPI associés aux flux audio et vidéo aux points de contrôle particuliers.

18. Appareil selon la revendication 5, dans lequel un flux MPEG-2 est traité d’abord par un administrateur conceptuel de GPPI pour obtenir les Outils de GPPI nécessaires et les appliquer aux flux audio et vidéo associés.
Fig. 1

1.0
CONTENT OWNER

1.1
CONTENT PROVIDERS X

1.2
CA SYSTEM A

1.3
CONTENT DECODING ON TERMINAL WITH CA SYSTEM A

1.4
CONTENT PROVIDERS Y

1.5
CA SYSTEM B

1.6
CONTENT DECODING ON TERMINAL WITH CA SYSTEM B

1.7
CONTENT PROVIDERS Z

1.8
CA SYSTEM C

1.9
CONTENT DECODING ON TERMINAL WITH CA SYSTEM C
Fig. 2

1.1 SERVER → CONTENT STREAM

IPMP CONTROL INFORMATION:
- IPMP TOOL LIST
- IPMP TOOL CONTAINER
- IPMP STREAM
- IPMP MESSAGES TO DIFFERENT IPMP TOOL INSTANCES

NETWORK LAYER

2.3 COMPLIANT MPEG-2 IPMP TERMINAL

2.4 IPMP TOOLS MANAGER
(PARSE TOOL LIST, RETRIEVE TOOL FROM TOOL CONTAINER OR ELSEWHERE, OBTAIN IPMP CONTROL GRAPH)

2.6 LICENCE/KEY/USAGE RULE

2.7 OUTPUT MESSAGES:
- LICENCE/KEY, USAGE RULE
- TOOL A WITH ITS TOOL ID, TOOL B WITH ITS TOOL ID, TOOL C WITH ITS TOOL ID

MEMORY IN TERMINAL

MESSAGE ROUTER
PRE-DEFINED INTERFACE FOR IPMP TOOLS

2.9 MEDIA DECODER & PLAYBACK