Trends of MPEG-21 IPMP Standardization

MPEG-21 IPMP 표준화 동향

S.O. Hwang(황성운) Contents Technology Department (컨텐츠서비스기반연구팀 연구원)
J.H. Kim(김정현) Contents Technology Department (컨텐츠서비스기반연구팀 연구원)
K.S. Yoon(윤기송) Contents Technology Department (컨텐츠서비스기반연구팀 책임연구원, 팀장)
M.J. Kim(김명준) Contents Technology Department (컨텐츠기술연구부 책임연구원, 부장)

Through the activities of MPEG (Moving Picture Experts Group), a number of issues have been identified within the scope of IPMP (Intellectual Property Management and Protection). This paper aims to address parts of them in the aspects of overall structure of MPEG-21 IPMP and to result in increasing the understanding of MPEG standardization works. In particular, this paper will address the major issues such as hooks, interoperability, and authentication architecture. Based on the architectures, this paper will explain how the MPEG-21 IPMP works with its components.

I. Introduction

MPEG has worked on the new standard of MPEG-21, which is known as "multimedia framework". It aims at setting out a vision for enabling transparent and augmented use of multimedia resources across a wide range of networks and devices used by different communities [1]. The setting up 'big picture' is to describe the specification of all the elements, which exist to build an infrastructure for the delivery and consumption of multimedia content. Now six key technical elements have been defined in MPEG-21: Digital Item Declaration (DID, now in FCD stage [2]), Digital Item Identification and Description (DIID, now in CD stage [3]), Intellectual Property Management and Protection (IPMP) Architecture (now in CD stage [4]), Rights Expression Language (REL, now in WD stage), Rights Data Dictionary (RDD, now in WD stage), and Digital Item Adaptation (now first CFP).

How are digital items managed and protected presently? Some of the observations made [5] are: No IPMP system has yet emerged as a de-facto standard. While various, proprietary IPMP systems exist today, no framework exists to allow for interoperation amongst such systems. In result, end users cannot interact with content today in the interoperable way between IPMP systems. Rights holders or content providers require IPMP framework to exercise their rights by making available their content securely and transparently.

◈ The Goal of MPEG-21 IPMP

According to the document [6], MPEG-21 IPMP aims to provide a uniform framework that enables all Users to express their rights and interests in, and agreements related to, Digital Items, and to have assurance that those rights, interests and agreements will be persistently and reliably managed and protected across a wide range of networks and devices.

This paper has the following structure: In Section 2, we examine the history of MPEG-21 IPMP standardization activities. Section 3 overviews major features and discusses their related issues based on the standardization work. Section 4 explains how the MPEG-21 framework works in terms of IPMP tools. Section 5 gives definitions of components which comprise the MPEG-21 framework. Section 6 concludes this paper with some discussion.

II. History of MPEG-21 IPMP Standardization Activities

1. Major Changes of MPEG IPMP Standardization Activities

With the demand of protection on the copyright of multimedia contents, CfP [7] for MPEG-4 IPMP system was proposed in the 39th meeting, April, 1997 and MPEG-4 IPMP v.1 [8] was produced in the 46th meeting, December, 1998.

Again in the 53rd meeting of July, 2000, CfP [9] for MPEG-4 IPMP Extensions was proposed. MPEG-4 IPMP Extension WD1.0 [10], MPEG-4 IPMP Extension WD2.0 [11], MPEG-4 IPMP Extension WD3.0 [12] were proposed in series, respectively, in the 54th meeting of October of 2000, the 55th meeting of January of 2001, the 56th meeting of March of 2001.

Standardization up to WD3.0 [12] was done in MPEG-4 part 1. From the 57th meeting of July of 2001 through the 59th meeting of March of 2002, MPEG-4 IPMP Extension amendments were done in MPEG-4 System part, MPEG-21 IPMP CD amendments MPEG-21 part 4, respectively.

2. Development of IPMP Issues

Original MPEG-4 IPMP [8] addressed protection of copyright, digital item identification, prevention technologies of illegal copy, monitoring and tracking of digital works.

To provide more flexible interoperability than that of original MPEG-4 IPMP, MPEG-4 IPMP Extensions WD1.0 [10] addressed both IPMP system interface and IPMP plug-in architecture of IPMP Tool.

Extending the scope of MPEG-4 system that MPEG-4 IPMP Extensions WD1.0 [10] was limited to, MPEG-4 IPMP Extensions WD2.0 [11] addressed general IPMP architecture.

MPEG-4 IPMP Extension WD3.0 [12] addressed both Extensions specification for all MPEG multimedia presentation and MPEG-4 IPMP Extension specification for MPEG-4 system only. It also classified interoperability into C-interoperability and M-interoperability.

In the 57th meeting of July of 2001, separation of MPEG-21 IPMP standardization and MPEG-4 IPMP standardization was made. MPEG-4 IPMP Extension Amendments was assigned to MPEG-4 Part 1 (System). MPEG-21 IPMP which started as CD status from the outset was assigned to MPEG-21 Part 4. While MPEG-21 IPMP specification was supposed to

be applicable to all multimedia formats, MPEG-4 IPMP specific to MPEG-4 system domain.

There were considerable progresses in MPEG-4 IPMP Extensions, but much less in the part of MPEG-21 IPMP, compared to MPEG-4 IPMP Extensions. There proposed a view that Digital Item (DI) of MPEG-21 should be considered altogether and all the part1, part2, part3 of MPEG-21 should come before.

In the 59th meeting of March of 2002, proposed was IPMP Schema descriptor for Digital Item in MPEG-21 architecture. Based on the MPEG-21 architecture, IPMP architecture was proposed by considering MPEG-21 DID model.

3. Development of Changes on Architectures and Related Functions

The MPEG-4 standard specifies a multimedia bit stream syntax and a set of tools and interfaces for designers and builders of a wide variety of multimedia applications. The approach of MPEG-4 is that the design of the IPMP framework needs to consider the complexity of the MPEG-4 standard and the diversity of its applications. This architecture leaves the details of IPMP systems designs in the hands of applications developers.

MPEG-4 adopts the modular IPMP approach. It separated between non-normative IPMP systems and the normative part of MPEG-4. This point of separation is the IPMP interface. This interface was designed to be a simple extension of basic MPEG-4 systems constructs. It consists of IPMP-Descriptors (IPMP-Ds) and IPMP-Elementary Streams (IPMP-ES). IPMP-Ds and IPMP-ESs provide a communication mechanism between IPMP systems and the MPEG-4 terminal. Altogether with MPEG-4 stream, IPMP-ES and IPMP-D stream are conveyed to the terminal and de-multiplexed at DMUX. The IPMP-Ds indicate which IPMP systems are to be used and provide information to these systems about how to manage and protect the content. The server delivers the content decryption keys to the client encrypted with the session key. These keys are delivered via IPMP-ESs. The mapping of keys and content is accomplished by IPMP-Ds associated with the content.

Original MPEG-4 IPMP [8] adopts the concept of 'hooks' in the MPEG-4 system as in the following Figure 1. Using 'hooks', it controls data by placing control point on the internal point of MPEG-4 system where data flow is required. This architecture has an advantage that it can utilize the existing MPEG-4 system structure without modification. However, neither M-interoperability nor C-interoperability can be supported.

In addition to 'hooks', MPEG-4 IPMP Extension WD1.0 [10] introduced both IPMP system interface and IPMP module plug-in to complement interoperability of MPEG-4 IPMP: five system interfaces and one data structure, IPMP Data, are defined. IPMP Data describes information such as rights descriptor format, key message format, content ID format, signature data format, IPMP Software Download protocol, structure of IPMP Data, etc. The five system interfaces are the following:

- Application Services API: interface between the secure domain and the outside world
- IPMP Service API: interfaces for plug-in, rights authentication, enc/decryption, signa-

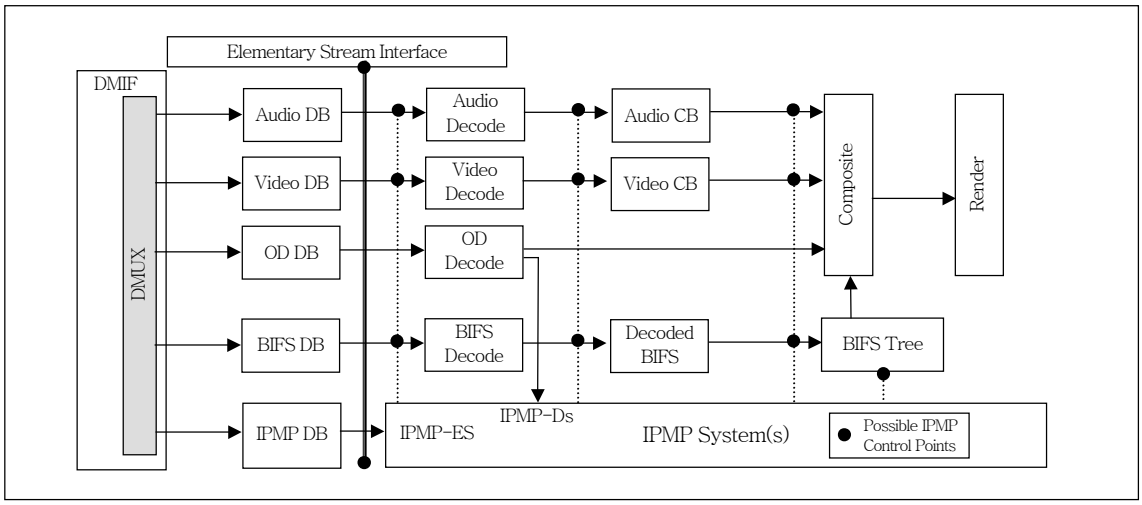

Fig. 1. The MPEG-4 IPMP of [8]

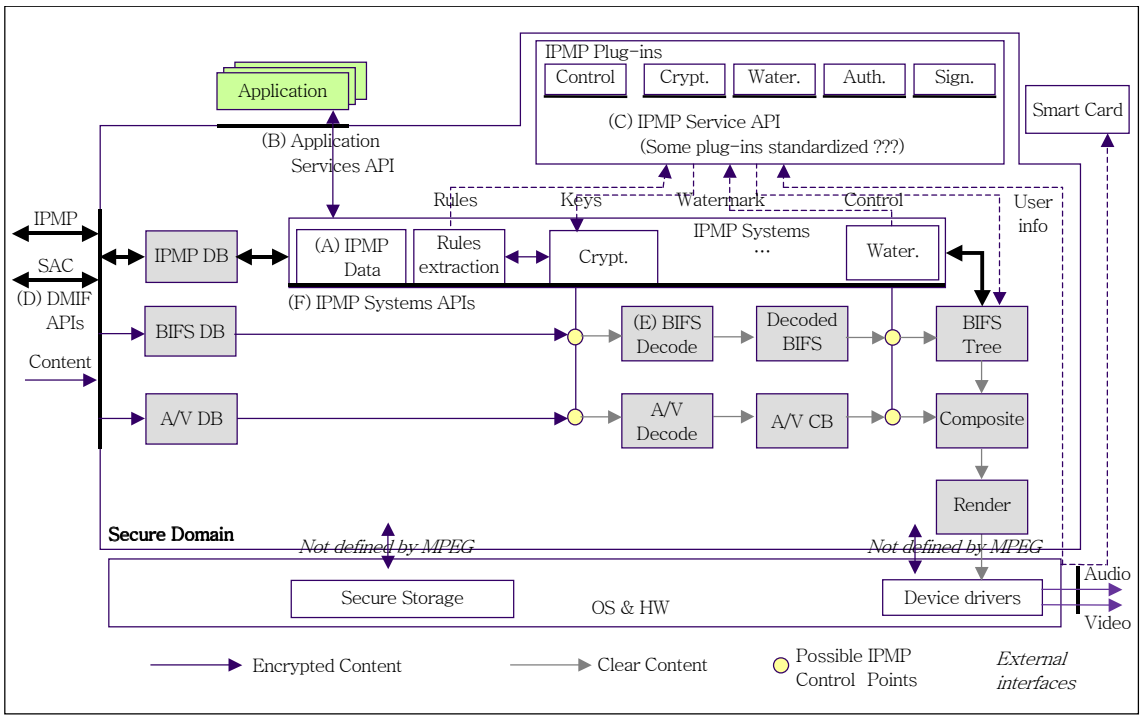

Fig. 2. The MPEG-4 IPMP Extension of [10]

ture, watermark

- DMIF APIs: interfaces between IPMP systems and network layers
- BIFS Decode to route information from the Scene Graph to the IPMP system and receive permission to utilize the content
- IPMP Systems APIs: interfaces between the terminal system and IPMP systems

The Figure 2 shows the architecture.

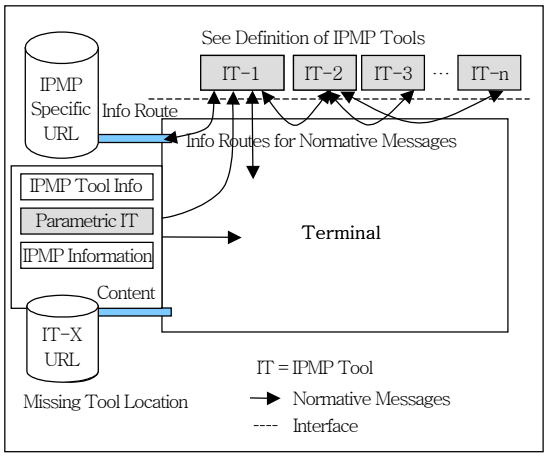

Fig. 3. The MPEG-4 IPMP Extension of [11]

MPEG-4 IPMP Extensions WD2.0 [11] proposed a more generalized form of architecture, extending the existing architecture specific to MPEG-4 system. It is the reason why the "Terminal" in Figure 3 is represented by empty rectangle. The concept of messaging interface was proposed to route messages rather than to allow direct message exchange amongst IPMP Tools or between IPMP Tools and terminal. The concepts of mutual validation and identity validation were proposed that formed authentication framework. The following architecture, Figure 3, shows how parametrically described IPMP Tools are used, and how contents are delivered altogether with IPMP Tool related information.

MPEG-4 IPMP Extension WD3.0 [12] proposed an architecture that included the following as in Figure 4:

- IPMP Tool Manager: access or retrieval of IPMP tool
- Message Router: implementation of the terminal-side behavior of the terminal-tool interface
- Terminal-IPMP Messaging Interface

There were also changes to the information contained in the content, in order to convey IPMP Tool Elementary Stream within the content itself.

On the architecture level, there is no big change in the MPEG-21 IPMP CD [13] of the 57th meeting. In addition, definitions of functions and requirements for both mutual authentication and messaging infrastructure were proposed. Various message interchange syntax/semantics were proposed such as IPMP_Tool_Message_Base (the smallest set of messages to be mapped to a given specification), Instantiation and Notification Messages, IPMP Information Delivery Functions, Data Processing Functions, Data Processing Functions, Intent and Permission Functions, User Interaction Messages, and Mutual Authentication Messages. In the annex section, some schemas such as schema for terminal platform, schema for parametric aggregation, schema for parametric description were added.

There was no architectural change in MPEG-21 IPMP CD [4] of the 58th meeting, but various kinds of messages were added for messaging interface such as IPMP_Tool_ Secure_Message, IPMP_GenericResponse, IPMP_MutualAuthentication, IPMP_ToolParamCapabilitiesQuery, etc.

There was no system-architectural change in the MPEG-21 IPMP CD[14] of the 59th meeting. They setup the architecture by applying Digital Item model of MPEG-21 to the existing MPEG-21 IPMP architecture. As the following Figure 5 shows, the structure of a terminal was based on the MPEG-21 System

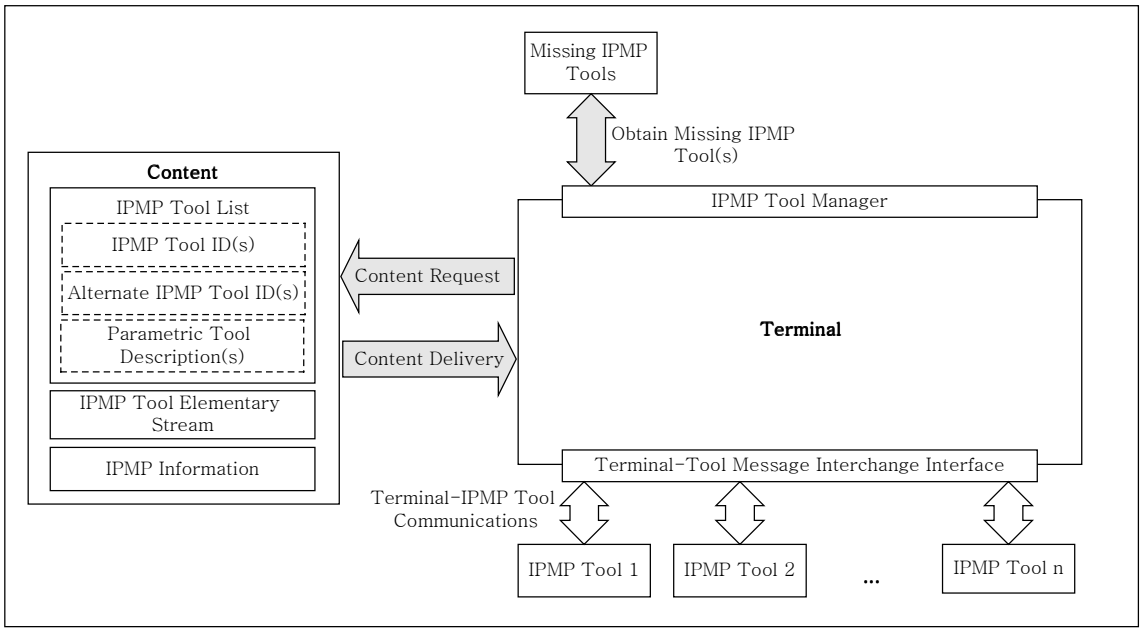

Fig. 4. The MPEG-21 IPMP of [12],[13]

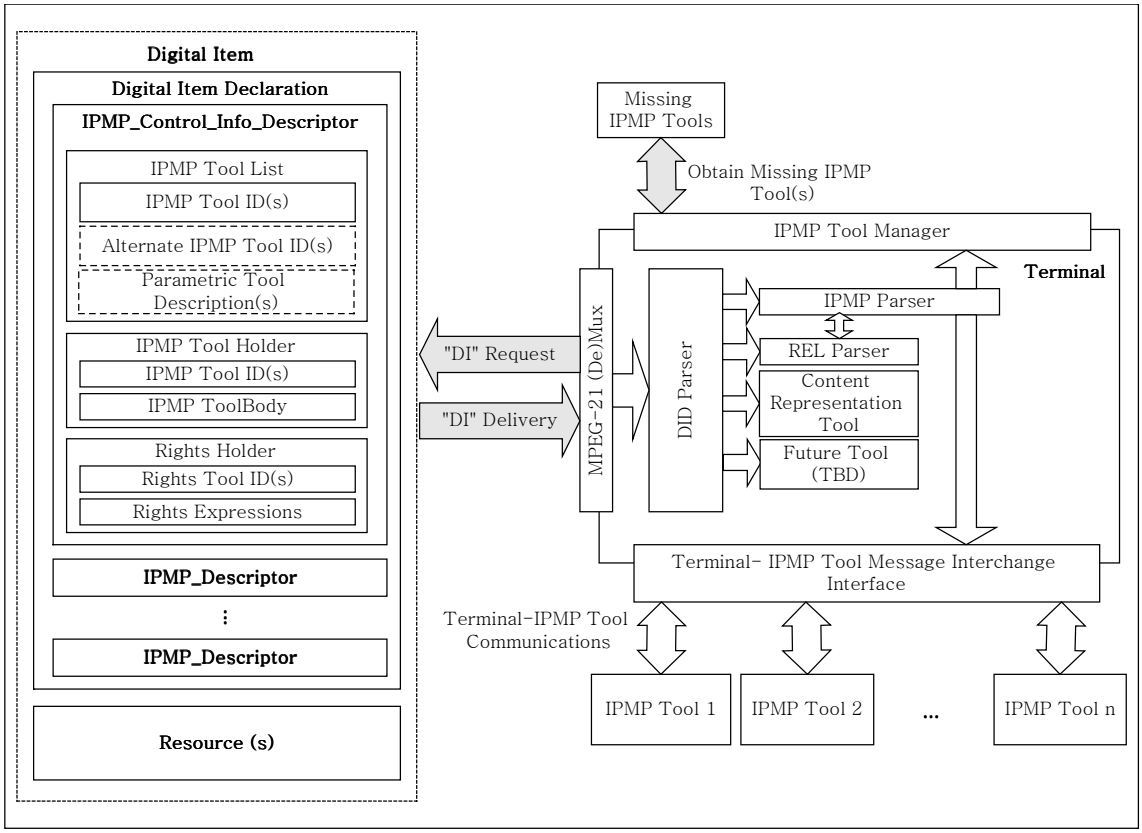

Fig. 5. The MPEG-21 IPMP of [14]

architecture and content was also configured according to the structure of Digital Item.

III. Major Features and Their Related Issues

1. Hooks Architecture

In general, the term Hooking represents a fundamental technique of getting control over a particular piece of code execution. It provides a straightforward mechanism that can easily alter the operating system's behavior as well as 3rd party products, without having their source code available.

MPEG-4 IPMP takes a modularity-oriented approach. A key motivation for hooking in IPMP is to plug-in a proprietary IPMP system into MPEG-4 player, and to signal which IPMP system was used and where to obtain it. In fact, while MPEG-4 does not standardize IPMP Systems, it does standardize the MPEG-4 IPMP interface. This mechanism enables the design of domain-specific IPMP Systems (IPMP-S) by coupling protection and management information with the elementary MPEG-4 stream elements. Applications based on the exchange of MPEG-4 content via mobile terminals, for example, will likely be quite different from those that exchange bit streams in the set top box domain.

After extensive discussion, MPEG WG11 decided that rather than adopting one particular IPMP system for MPEG-4, the choice of a given IPMP solution for an MPEG-4 application should remain the province of proprietary solutions or derivative, domain-specific standardization efforts. After WG11 released version 1 of MPEG-4, however, some parties expressed the viewpoint that the standard did not provide sufficient infrastructure for interoperability (C-interoperability described below) across different (but similar) application domains. That is, content from one domain that uses IPMP system A will not be playable on devices in another domain that uses IPMP system B. Another problem with the current approach is, the terminal environment could be heavier and more complex. It is because it adopts the tools of multiple vendors that have almost the same functions, except for minor parts.

Open hooking framework provides many advantages to application/service developers in nature [15]:

- API function's monitoring: enabling developers to track down specific "invisible" actions that occurring during the API call
- Debugging and reverse engineering: Hooking is one of most popular debugging mechanisms. API interception is said to be very powerful way of getting information about a binary executable
- Peering inside operating system: Hooking is quite useful for decoding undocumented or poorly documented APIs.

These advantages of hooking framework may serve as obstacles to IPMP environment in the sense that users have more possibility of changing/extending/tampering existing module functionalities against the intention of module providers'. Requirement specification and design of a secure, open hooking framework for IPMP are necessary such as: which interception mechanism to use, which injection mecha-

nism to follow, which trust mechanism hook allowing (control) mechanism is based on, etc. to thwart/endure users' attacks.

2. Interoperability

Interoperability is the major goal of standardization. Two kinds of interoperability have been distinguished:

- From the manufacturer's point of view: ensuring that IPMP tools from different suppliers can be integrated into the same terminal implementation concurrently through clear interface agreements ("M-interoperability")
- From the consumer's point of view: ensuring that content from multiple sources will play on players from different make ("C-interoperability")

While C-interoperability describes different content protected by different IPMP tools and played on different terminals, it does not provide for different IPMP tools protecting the SAME content, which M-interoperability does. Solutions for M-interoperability are not difficult to implement if we use Hooks framework. In the case of C-interoperability, however, the problem is quite different. In general, Hooks mechanism cannot enable the same content to be protected by different vendors' IPMP tools. So the document [14] indicates that while M-interoperability will be fully enabled, C-interoperability will be assisted to as large as extent possible. To fully support C-interoperability, there need interoperable ways amongst different vendors' IPMP tools.

Related to IPMP interoperability issues, the following key considerations are discussed in the document [16]:

1. Agreeing on the objective means to assess the trustworthiness of particular IPMP systems
2. Agreeing on policies for the governance of compliance with agreed interoperability specifications
3. Agreeing on a governance structure that will neutrally administer

(a) The agreed specifications
(b) Certification of the trustworthiness of individual systems
(c) Certification of the trustworthiness of rendering applications and devices
(d) Maintenance of the trustworthiness of the portions of the environment that enable interoperation to function
(e) Global management of inter-domain emergency response to systems compromise
(f) Administration of the key and credential management infrastructure.

3. Authentication Framework

The authentication framework assumes three layers of architecture for mutual authentication. It also features negotiation mechanisms to support a number of authentication methods.

A. Three Layered Architecture

○ Specific Authentication Method Layer

This layer is specific to the authentication method adopted by each vendor's IPMP tools. Messages proper to a specific authentication

method (e.g., SSL, Kerberos) are generated and interpreted.

○ IPMP Message Layer

The role of this layer is to bridge two layers, that is, Specific Authentication Method Layer and Message Routing Service Layer. An entity of this layer generates an IPMP message, which contains payload messages from Specific Authentication Method Layer and information necessary for Message Routing Service Layer to process the message, in particular, identifiers of the originator and the recipients.

○ Message Routing Service Layer

The role of this layer is to deliver IPMP messages to valid designated recipients specified in the IPMP messages.

B. Negotiation Mechanisms

It is important to understand that the MPEG architecture, by its nature, necessarily covers a very wide area of application such as broadcasting to Set Top Box, steaming to a cellular phone, downloading to a PC, rendering at a portable player, and so forth. There are two options to select authentication mechanism. One is to specify a single or a small number of authentication method, and another is to introduce a negotiation mechanism. MEPG chose the second option - negotiation mechanism. The negotiation mechanism works at the early stage of the handshake protocol between IPMP tools, and provides those tools with methods to agree on a common authentication method. The negotiation mechanism works as follows [17].

1. One of a pair of IPMP tools initially presents to the other of the pair a list of identification of the authentication methods that it supports.
2. The recipient IPMP tool selects one authentication method out of the list, which the IPMP tools supports. If the list does not include any authentication method that the recipient IPMP tool supports, it returns an error signal to the originator module.

The definition of the negotiation mechanism should be provided at IPMP Message Layer. This implies the following three things.

1. The syntax of the IPMP message is designed so that it is capable to specify (a list of) identifiers of authentication methods and error signals.
2. The semantic of identifiers of authentication methods is specified and/or reference to existing semantics (e.g., ISO object identifier) is specified.
3. The semantic of the error signals is also to be specified.

Negotiation mechanism does not guarantee that arbitrary two modules always authenticate each other, but provides implementers with maximum freedom and flexibility, and allows users to enjoy content using optimized MPEG systems.

C. Global Authentication Framework for MPEG-21 IPMP Required

For its security and trustworthiness, MPEG-21 IPMP depends too heavily on the assumption that devices (terminal environment, IPMP

tools, encoders/decoders, parsers, etc.) cooperate or communicate in the contents-delivering/consuming environment under trustworthy authentication process. For modules to authenticate each other, we need to describe a framework for trust (authentication) models that covers use of existing authentication technologies such as X.509 public-key certificates, Kerberos shared-secret tickets, and password digests including unknown methods. It should also provide an integrating abstraction allowing systems to build a bridge between different security technologies. The general model is required to construct higher-level, specific authentication mechanism-independent authentication framework that governs lower-level key exchange, authentication, authorization, auditing, and trust mechanisms.

D. Messaging Routing Service

MPEG does not specify APIs to support the interoperability among IPMP tools and terminals. Instead it takes an approach of message routing among tools. The role of Messaging Routing Service is to deliver (IPMP) messages between IPMP tools including a terminal. An individual IPMP tool sends messages to Message Routing Service (i.e. call functions supported by Messaging Routing Service) instead of directly sending them to other IPMP tools (i.e. calling functions supported by other IPMP tools).

Another function of Messaging Routing Service is directory service, that is, if Messaging Routing Service receives a message specifying the required functionality, it searches an appropriate IPMP tool or information on behalf of an IPMP tool and automatically transfers it to the IPMP tool.

E. Trust Security Metadata

Trust and Security Metadata may include certificates, credentials or integrity verification information. These information are generated and assigned to tools (terminals, IPMP tools) by Audit agency who qualifies and quantifies the ability of an IPMP system to resist attempts to tamper with the operations of the system or to reverse engineer any of the internal details of the IPMP system. The agency then digitally signs the system's trust metadata with its private key to establish authenticity, non-repudiation and integrity. Currently XML schema for establishing trust relationship between entities are given: It consists of audit date and trust information containing attack profile that indicates the level of protection that must be offered by the selected IPMP system. Content providers can specify the level of protection so that their contents can be consumed only in the terminal environment including IPMP tools that meet the pre-specified protection level.

F. Support of Multiple Retrieval (Access, Delivery) Methods of IPMP Tools

○ Direct Aownloading

Users can download their necessary IPMP tools, for example, on the Internet.

○ Parametric/Alternate Tool

In case the specified tool in the content is not available in the terminal, a list of alternate tools can be specified to help play the content.

Parametric description tool enables a terminal to choose a specific tool implementation that will support all functionality required by the corresponding protection mechanism. Parametrically described IPMP tools must have standard interfaces, since opaque data cannot be guaranteed for correct processing by an arbitrary implementation.

○ Imbedding IPMP Tools within Content Stream

Binary Representations (e.g., platform dependent native code, Java bytecode) may be carried directly or by reference in an MPEG presentation. The IPMP Tool Manager should receive implementations of IPMP Tool carried in the content prior to processing protected streams in the content. This method has an advantage that it enables vendor-specific services or modules (e.g., IPMP tool or license tool) to be delivered to the terminal-side easily. However, since it is executable code, protection mechanisms against malicious attacks or viruses are required in the terminal environment.

IV. How Does the MPEG-21 IPMP Work? [14]

In this section, we explain what components comprise the MPEG-21 framework, and how they work in terms of IPMP. Related to this section, see the above Figure 5 showing the overall structure of digital item, and IPMP tools in the terminal environment.

1. User Requests Specific Content

The manner in which content is requested is out of scope of this document. However, the following recommendations are made for the order in which different parts of the content are received and used:

1. IPMP Requirements on the terminal should be placed with or before media requirements on the terminal.
2. Access Information and/or restrictions should precede Content Stream download information (IPMP_Control_Info_Descriptor).

After DID is received at MPEG-21 (DE)MUX and sent to DID parser, the DID parser extracts IPMP information and sends to IPMP parser. Then the IPMP parser extracts all IPMP information and transfers the information to IPMP Tool Manager and Message Router in MPEG-21 IPMP system. Other IPMP Information such as IPMP message, Keys, etc. could be put in IPMP as OpaqueData in IPMP_Descriptor or the Resource element in DID. When there is Rights Expression information under IPMP_Control_Info_Descriptor, it is transferred to REL parser. The parsed rights information can be enforced by Rights Management Tool carried in the DID or the Resource through Descriptor Reference (remotely).

2. IPMP Tools Description Access

1. The terminal accesses the IPMP Tool List.
2. Using the IPMP Tool List, the terminal determines the IPMP Tools required to consume the content.

3. IPMP Tools Retrieval

1. If the tools are available locally at the ter-

minal, proceed to 4.4.

2. The terminal attempts to obtain the missing IPMP Tools. Some missing Tools may be carried in the Content itself. Otherwise, the Tool must be obtained remotely. The following procedure may be followed for such retrieval.

(a) The terminal accesses an implementation specific database for a location for the missing Tool.
(b) A communication channel is setup between the terminal and the Tool location.
(c) The terminal implementation provides information about its platform and the Tool database identifies a compatible Tool implementation.
(d) The IPMP Tool Manager accesses/acquires the missing IPMP Tools.
(e) The newly acquired tools are made available for use by the terminal.

4. Instantiation of IPMP Tools

1. The terminal instantiates the IPMP Tool(s) locally or remotely.
2. The instantiated Tools are provided with initial IPMP Information from the DID.
3. One or more Tools, identified in the DID, may use IPMP Information to determine security requirements for content access, and monitor and facilitate the establishment and maintenance of these security requirements in inter-Tool communication.

5. IPMP Initialization and Update – In parallel with Content Consumption

1. The Message Router routes IPMP Information to the IPMP Tools.
2. The terminal consumes the content if allowed by the requisite IPMP Tools.
3. During content consumption, the complete walkthrough can be requested again. Requests for content consumption are implicit within the process, or are requested by the User.

V. The MPEG-21 IPMP Components [14]

In this section, we give definitions of major components that comprise the MPEG-21 framework described in the previous section.

○ IPMP_Control_Info_Descriptor

IPMP_Control_Info_Descriptor contains the IPMP Control Information, which contains necessary information like Tool List, IPMP Tool Holder, and IPMP Rights Holder.

○ IPMP Tool List

IPMP Tool List identifies and enables selection of, the IPMP Tools required to process and protect the Content. It includes a list of IPMP Tools and is used to specify all IPMP Tools that should be used in order to consume the content. By this Tool List, the terminal will determine the IPMP Tools obtained either from local terminal, carried in the content or obtained from remote sites. The Tool List may include IPMP Alternative and IPMP Parametric to denote a list of alternate IPMP Tools and the parametric description of an IPMP Tool.

○ IPMP Tool

IPMP Tools are modules that perform (one

or more) IPMP functions such as authentication, decryption, watermarking, etc. A given IPMP Tool may coordinate other IPMP Tools. IPMP Tool(s) is a generic expression for different tools for the management and protection of Digital Items or parts thereof. Each of these tools receives appropriate information from other tools and acts upon it. IPMP Tools may also pass information on to other tools.

○ IPMP Tool Holder

Tool Holder may be used for cases whereby MPEG-21 content itself carries the binary IPMP Tool. The device may retrieve the IPMP Tool from the content, load it, instantiate it and use it immediately to play out the content. It includes ToolID and ToolBody, which are represented by Bytes.

○ Rights Holder

Rights Holder conveys the Rights/Usage Rules associated with the IPMP protected content. Similarly, it includes Rights ToolID especially for Rights Parser such as REL Parser, proprietary rights management tool (XrML parser, ODRL parser, etc.). Bytes-represented Usage Rule transformed from XML-based Rights Expression is another element of Rights Holder.

○ IPMP_Descriptor

IPMP_Descriptor conveys the control point information of the IPMP Tool, including at which control point the tool resides (before or after the practical Resource consuming), and its sequence relation (priority) to other tools residing at the control point.

○ Resource

A Resource is an individually identifiable asset such as a video or audio clip, an image, or a textual asset. A Resource may also potentially be a physical object.

○ DID Parser

The Digital Item Declaration Parser receives the DID from the I/O Tool and parses the XML text declaring the structure of the Digital Item. Elements it cannot parse or act upon (e.g., an Expression of Rights and Permissions) are forwarded to the appropriate tools through the Message Interface.

○ REL Parser

The Right Expression Language Parser receives the REL text (in XML) and parses it. Elements it cannot parse or act upon are forwarded to the appropriate tools through the Message Interface.

○ IPMP Tool Manager

A conceptual entity within the terminal that processes IPMP Tool List(s) and retrieves the Tools that are specified therein.

○ IPMP Parser

The IPMP Parser receives the IPMP information text and parses it. It will usually use IPMP Tool(s) to act upon this IPMP information (and other information, e.g., REL). Elements it cannot parse or act upon are forwarded to the appropriate tools through the Message Interface.

○ Message Router

A conceptual entity within the terminal that implements the terminal-side behavior of the terminal-Tool interface.

○ Terminal

A terminal is an environment that consumes possibly protected content in compliance with the usage rules.

VI. Conclusions

MPEG-21 IPMP standardization activities have been done collaboratively with MPEG-4 IPMP part as well as other parts of MPEG-21. As they progressed, a lot of issues and views have been discussed, though some of them are reviewed in this paper. In addition, however, we need to define requirements for a secure environment. Some of them were discussed in the above. Additional examples of such requirements for security could be: the user cannot read the content of the buffer inside the IPMP Tools using monitoring tools; there should be overall, non-conflicting security policies that govern the complexities coming from multiple inter-tool interfaces, etc.

References

[1] ISO/IEC JTC1 SC29/WG11/N4333, MPEG-21 TR, "Vision, Technologies, and Strategy of MPEG-21," July 2001.

[2] ISO/IEC JTC1 SC29/WG11/N4530, "MPEG-21 Digital Item Declaration (DID) Final Committee Draft," Dec. 2001.

[3] ISO/IEC JTC1 SC29/WG11/N4532, "MPEG-21 Digital Item Identification and Description (DII&D) Committee Draft," Dec. 2001.

[4] ISO/IEC JTC1 SC29/WG11/N4411, "Intellectual Property Management and Protection Architecture, Study of Text of CD ISO/IEC 21000-4:2001," Dec. 2001.

[5] ISO/IEC JTC1 SC29/WG11/N4040, "Study on MPEG-21 (Digital Audiovisual Framework) Part 1 v2.0," Singapore, Mar. 2001.

[6] ISO/IEC JTC1 SC29/WG11/N4681, "MPEG-21 Requirements v 1.0," Jeju, Mar. 2002.

[7] ISO/IEC JTC1 SC29/WG11/N1714, Call for Proposals on technology for management and protection of content in MPEG-4.

[8] ISO/IEC JTC1 SC29/WG11/N2614, MPEG-4 Intellectual Property Management & Protection (IPMP) Overview & Applications Document.

[9] ISO/IEC JTC1 SC29/WG11/N3545, Call for Proposals for IPMP Solutions.

[10] ISO/IEC JTC1 SC29/WG11/N3729, Working Draft 1.0 of ISO/IEC 14496-1:2000/ AMD3.

[11] ISO/IEC JTC1 SC29/WG11/N3871, Working Draft 2.0 of ISO/IEC 14496-1:2000/AMD3.

[12] ISO/IEC JTC1 SC29/WG11/N4095, Working Draft 3.0 of ISO/IEC 14496-1:2000/AMD3.

[13] ISO/IEC JTC1 SC29/WG11/N4269, Text of CD ISO/IEC 21000-4:2001.

[14] ISO/IEC JTC1 SC29/WG11/N4717, "Study of Text of CD ISO/IEC 21000-4:2001," Jeju, Mar. 2002.

[15] Iivo Ivanov, API Hooking Revealed, http://www.codeproject.com/system/hooksys.asp.

[16] ISO/IEC JTC1 SC29/WG11/M6443, "Flexible and Robust IPMP Solutions Using the MPEG-4 IPMP Hooks," La Baule, Oct. 2000.

[17] ISO/IEC JTC1 SC29/WG11/M6918, "Mutual & Identity Validation of IPMP Tools and Rights Authentication for MPEG Content," Jan. 2001.