A Tool Pack Mechanism for DRM Interoperability

Bumsuk Choi, Youngbae Byun, Jeho Nam, and Jinwoo Hong

ABSTRACT—As the number of digital content service providers increases, a variety of digital rights management (DRM) systems appear without supporting interoperability. The lack of interoperability in DRM systems causes inconvenience to customers, especially when they want to play content through multiple devices manufactured by different vendors. In this letter, we propose a novel method to support interoperability between different DRM systems. The proposed technique aims to build an open framework structure which satisfies DRM vendors' requirements by enhancing the security of intellectual property management and the protection tools.

Keywords—Digital rights management (DRM), IPMP, MPEG, security, interoperability.

I. Introduction

Many digital rights management (DRM) standard organizations have been founded to support interoperability since 2000. Intellectual property management and protection (IPMP) is a representative DRM architecture developed by MPEG for interoperable and flexible distribution of multimedia content between different user terminals and different content providers. The basic idea of MPEG-2/4 IPMPX [1], [2] is to make each DRM function as an object module. This object module, called an IPMP tool, provides content protection functions, such as authentication, encryption, and watermarking. It defines an IPMP terminal which manages IPMP tools and controls a player. Interoperability is achieved by standardizing the interface between an IPMP terminal and an IPMP tool and also between the IPMP tools. The format of the signaling information that indicates which IPMP tools are applied to the content has also been standardized. This solution can be called the single IPMP tool solution. Ideally, it is a very reasonable method to achieve interoperability. Realistically, it is unlikely for DRM vendors to accept this standard. This is true for the following reasons.

1) The security of the IPMP tool itself is very critical for the protection of content. Therefore, it is unlikely for DRM vendors to reveal their private IPMP tools.
2) The information about the IPMP tools should be written publicly in the associated content. This may be used in malicious attacks and threaten the DRM system.
3) A DRM system is composed of many kinds of IPMP tools and operates through interactions among them. It is easy to infer what kind of process is performed in the DRM system by analyzing interchanging messages. The DRM vendors are not willing to expose these interactions to unreliable devices.
4) When a developed IPMP tool has totally new functions and protocols, a new set of standardized interface messages needs to be defined. This causes a conventional IPMP terminal to be changed considerably.

To solve these problems, we propose an advanced method we call the tool pack solution which supports interoperability between different DRM systems.

II. Proposed DRM Architecture

Most DRM vendors have a dedicated set of IPMP tools and are reluctant to separate them into individual tools. The rationale behind the tool pack is to bind the private IPMP tools as a tool group and establish a tool agent to operate the tools in the tool group [3]. Figure 1 shows the internal structure and interface of the tool pack.

A tool pack is defined as a structure containing the tool agent and the tool group, so DRM vendors can provide their private tools in the tool group with the tool agent in the tool pack structure. The tool agent communicates with the IPMP

Manuscript received Sept. 20, 2006; revised Mar. 20, 2007.

This work was supported by the IT R&D program of MIC/IITA [2005-S-403-02, SmarTV Technology and 2007-S-003-01 Development of Protection Technology for Terrestrial DTV Program].

Bumsuk Choi (phone: + 82 42 860 1574, email: bschoi@etri.re.kr), Youngbae Byun (email: byun@etri.re.kr), Jeho Nam (email: namjeho@etri.re.kr), and Jinwoo Hong (email: jwhong@etri.re.kr) are with the Radio & Broadcasting Research Division, ETRI, Daejeon, S. Korea.

terminal. The tool agent operates the tools in the tool group in a proprietary way, so it is not necessary to define common interface messages for the communications between the tool agent and the tool group. As shown in Fig. 1, common interface messages are needed only for minimal communications between the tool agent and the IPMP terminal.

Compared with the single IPMP tool solution, the tool pack solution makes it possible to hide detailed operational information from the IPMP terminal and reduces the number of standard interface messages. Additionally, it increases the extensibility because a reduced set of interface messages is enough for the operation of any future tool pack.

The tool pack solution is composed of several technical items such as the IPMP terminal, the tool pack structure, and interface messages.

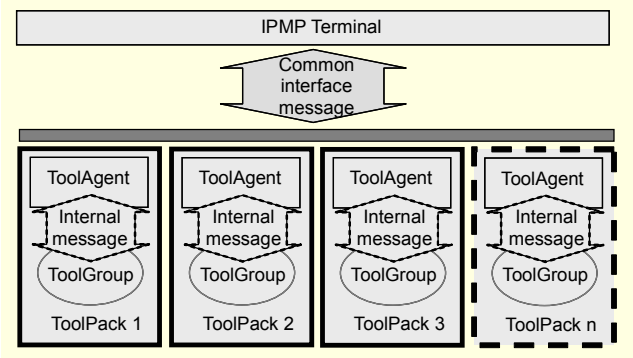

Fig. 1. Structure and interface of a tool pack.

1. IPMP Terminal for Tool Pack

An IPMP terminal is a module within a device which executes DRM-related functions in a trusted fashion. The IPMP terminal extracts DRM information, the tool pack, keys, and license from the content. The IPMP terminal also controls the critical points internal to a device. Such a control point is a specific position where the IPMP tools may operate along the content decoding pipelines.

The IPMP terminal, using the DRM information, locates the required tool pack and instantiates the tool agent. Once instantiated, the IPMP terminal handles initialization, authentication, and supervision of the tool agent using IPMP messages. The IPMP messages are usually generated by the IPMP terminal or the tool agent during decoding time. However, in some cases, they may be delivered in the DRM information in the content (for example, periodically changed key data may be delivered to a device in a DRM message). In this case, it is a role of the IPMP terminal to extract the IPMP messages and deliver them to the tool agent. When a missing tool pack is found, the IPMP terminal obtains the missing tool pack from a remote server. Once used, all tool packs are kept in the secure storage of the device.

2. Tool Pack Structure

A tool pack structure, as a container for the tool agent and the tool group, is represented by the following schema shown in Fig. 2. Each element is defined by the following semantics.

ToolPackID contains a unique identification.

Pack contains a unit of tool pack. Each tool pack unit exists for its target device. Multiple packs can be conveyed in the tool pack structure in case the DRM vendor wants to support more than one device environment.

DeviceInformation contains H/W or S/W (including OS) information of the target device which the tool agent and the tool group can operate on.

ToolAgent contains sub-elements for tool agent.

ToolGroup contains sub-elements for tool group.

ToolPackageType indicates the details of the package of the binary code. Examples are CAB or a Winzip.

ToolAgentBody contains binary code of tool agent.

ToolGroupBody contains binary code of IPMP tool.

Signature contains a digital signature for the authenticity.

The tool pack data conveys information for initialization of the tool agent such as parameter values or flags. The tool pack data can be delivered together with the tool pack or can be delivered separately.

ToolPackRefID contains reference identification of the tool pack. It indicates the matching tool pack needed for the tool pack data.

OpaqueData contains information for initializing the tool agent. The data form is dedicated to a specific tool agent. Therefore, only a proper tool agent can recognize it and use this information.

Signature contains a digital signature for authenticity.

3. APIs for Tool Pack

Once the IPMP terminal and the tool agent are instantiated in a device, they communicate with each other using interface messages [4].

InitialAuthenticationAndMutualAuthentication: The IPMP terminal and the tool agent authenticate each other and make a secure channel to transfer data. These messages are referenced from MPEG-2/4 IPMPX specification.

GetToolGroupReference: The tool agent can optionally use a private library containing algorithms such as encryption or watermarking. In this case the tool agent requests the reference of the tool group from the IPMP terminal.

SendToolGroupReference: The IPMP terminal retrieves the tool group from the tool pack structure and sends the reference of the tool group to the tool agent.

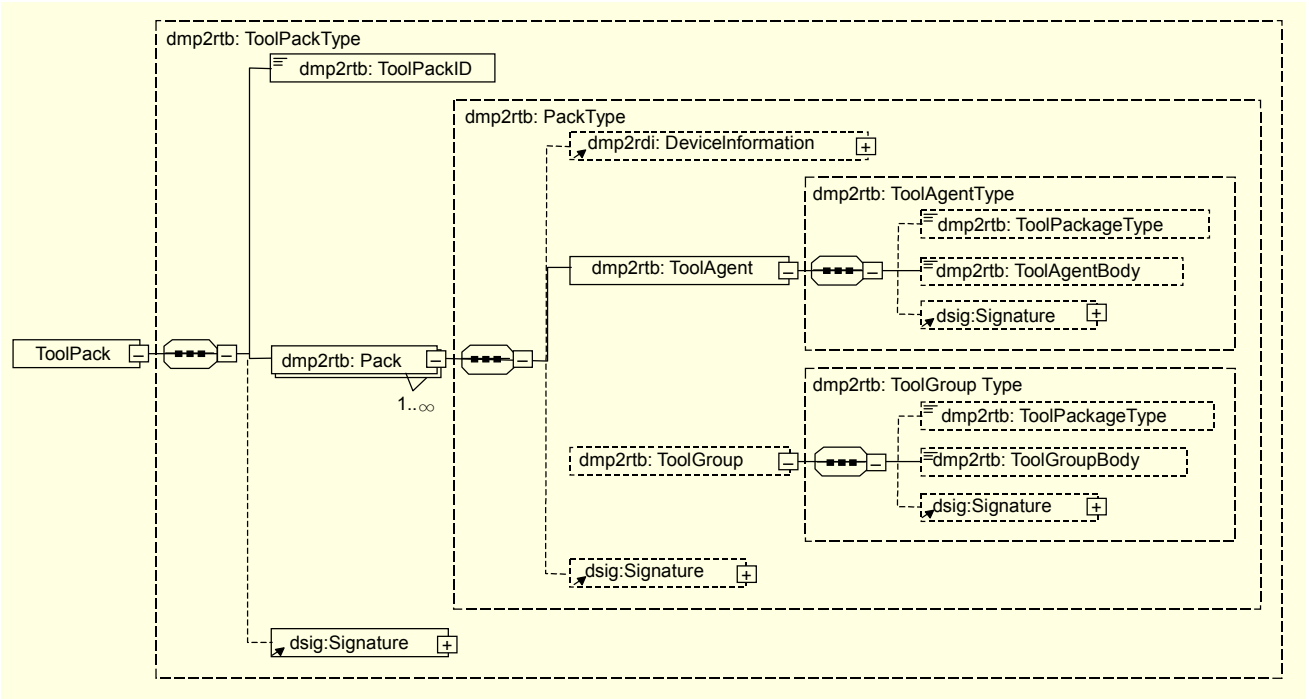

Fig. 2. Schema structure of a tool pack.

GetToolReference: The tool agent may need a single IPMP tool. The single IPMP tool is a popular and stand-alone IPMP tool used by many DRM systems where the tool agent needs more than one single IPMP Tool.

SendToolReference: The IPMP terminal sends the reference of the single IPMP tool to the tool agent.

GetToolPackData: The tool agent can be implemented to be always called with the same initial state. Alternatively, the tool agent may be implemented to be called with an initial value that changes whenever it is called or with an initial value downloaded from, for example, an external authorized tool server to provide more enhanced security.

SendToolPackData: The IPMP terminal retrieves the tool pack data and sends it to the tool agent.

GetControlPointList: The tool agent requests the control point list to connect each tool to a control point.

SendControlPointList: The IPMP terminal sends the list of control points available in the device. This message contains the number of control points and a set of information including the control point ID and the location information so that an IPMP tool can be connected. The control point list is extended from the MPEG-2/4 IPMPX. For the tool pack mechanism, we defined additional control points as those before storing, before playing, and before transferring.

CanProcess: When the initialization, authentication, and connection processes are successful, the IPMP terminal is notified that the tool agent is ready to decode content.

DisconnectToolPack: The IPMP terminal commands the tool agent to disconnect tools from the control point and stop the processing.

NotifyToolPackEvent: The tool agent notifies the IPMP terminal that an event occurred during the initialization process or decoding process. The IPMP terminal performs the proper action according to the type of the event.

III. Conclusion

In this paper, we presented an advanced interoperable DRM method. The distinctive feature of the proposed method is that it is based on an open framework structure, yet it satisfies the DRM vendors' requirements by enhancing the security of IPMP tools. To achieve this, we introduced a tool pack mechanism which allows users to keep the critical information on the operation of IPMP tools away from the outer entities.

References

[1] ISO/IEC 13818-1, Information Technology: Generic Coding of Moving Pictures and Associated Audio Information, Part 11: IPMP on MPEG-2 Systems, May 2002.

[2] ISO/IEC 14496-13, Information Technology: Coding of Audio-Visual Objects, Part 13: Intellectual Property, May 2002.

[3] Digital Media Project; Approved Document No. 2 – Technical Reference: Architecture; 2005/04, www.dmpf.org/

[4] Digital Media Project; Approved Document No. 3 – Technical Specification: Interoperable DRM Platform; 2005/04, www.dmpf.org/