A CONTENT PROTECTION SCHEME USING MPEG-21 CONCEPTS AND TOOL

Chia-Hsien Lu, Feng-Cheng Chang, and Hsueh-Ming Hang
Dept. of Electronics Engineering, National Chiao Tung University, Taiwan, R.O.C.
chl0113.ee93g@nctu.edu.tw, breeze@alumni.nctu.edu.tw, hnhamg@mail.nctu.edu.tw
Abstract

How to effectively manage and protect the rights becomes an important issue of consuming digital contents. In this paper, the technologies of MPEG-21 IPMP and REL are adopted to construct a digital rights management (DRM) system. The former provides ways to protect a digital element; and the latter is able to describe various kinds of rights and it provides an authorization model to generate an authorization proof to manage the rights. The concepts of the MPEG-4 IPMPX are also used as the basic framework in our design. To implement these standard specifications, we design a set of IPMP Tools, which carry the functionalities of the MPEG-21 IPMP and REL. We then integrate this set of tools into the IPMPX framework on the MPEG-21 Test Bed. We end this paper with three application examples to demonstrate that our system can successfully safeguard digital resources and effectively manage the rights.

1. INTRODUCTION

As the network and digital media technologies advance, it is easy nowadays for everyone to create and distribute digital multimedia contents. The intellectual property protection and management becomes an important issue. Therefore, modern multimedia system designs often include digital rights management (DRM) as an essential component.

Most of the popular DRM systems incorporate the encryption functionality to protect digital contents from eavesdropping. However, encryption alone is not sufficient to provide a more sophisticated right control mechanism. In addition, a complete DRM system requires a means to describe who owns a piece of content and how it can be used. This can be achieved by incorporating a rights expression language. Generally speaking, a rights expression language is able to describe the rights holder of a certain resource, the target consumer of the resource, the allowed rights on the resource, and the necessary conditions for resource consumption. There are a few rights expression languages defined for various application fields. One of them is defined by the MPEG committee for the use in connection with the MPEG-21 multimedia framework.

MPEG-21 [1] is a set of specifications defined as “a multimedia framework to enable transparent and augmented use of multimedia resources across a wide range of networks and devices used by different communities.”[1] There are two parts in MPEG-21 that constitute the basis of a DRM system. The MPEG-21 Part 4 [2] defines the high level concepts and schema for Intellectual Property Management and Protection (IPMP); and the MPEG-21 Part 5 [3] defines the Rights Expression Language (REL).

The MPEG-21 REL is an XML-based language that can declare an authorized distribution for the use of any content, resource, or service owned by specific users. It provides flexible, exact, and rich representation of rights. It can be used in various applications due to the interoperability of this language. According to a specific application, users who use REL for rights management can define their own extensions as well as create a specific profile. The other parts of MPEG-21 are also related to specific aspects of a DRM system.

In this paper, we study and implement the MPEG-21 IPMP and REL specifications. By analyzing several typical multimedia DRM scenarios, we design a DRM scheme to protect the digital assets and to perform the rights management. Its functionality is compliant to the MPEG-21 IPMP and REL concepts. Because the specifications in MPEG-21 IPMP are abstract, we map its DRM functionality to the MPEG-4 IPMP Extension (IPMPX) [4] architecture. To verify the feasibility of our design, we choose the MPEG-21 Test Bed [5][6][7] as the content delivery platform. In addition to the ability of simulating a basic real-time content delivery system, it also has a simple implementation of the IPMPX architecture inside. Hence, we implement our DRM by integrating the MPEG-21 IPMP, the MPEG-21 REL, together with IPMPX on the Test Bed.

This paper is organized as follows. We first introduce the concepts and specifications of the MPEG-21 IPMP in Sec. 2. Second, the REL concepts and specifications are discussed in Sec. 3. Then, Sec. 4. gives a brief overview of the MPEG-21 Test Bed with IPMPX. In Sec. 5., we describe the details of our design and implementation of the MPEG-21 IPMP and REL within the MPEG-4 IPMPX framework on the MPEG-21 Test Bed. Finally, we design three application examples to demonstrate the functionalities of our designed DRM system. A few conclusion remarks of our work are given in Sec. 7.

2. MPEG-21 IPMP

The goal of the MPEG-21 IPMP is to provide the management of rights and intellectual property through the use of protected Digital Items. A Digital Item (DI) is the subject that the MPEG-21 technology applies to. It can be a multimedia clip, a document, or even a service which provides sort of “digital content”. To describe the properties of a DI, the MPEG-21 developed the Digital Item Declaration Language (DIDL), which is an XML-based expression defined in the MPEG-21 Digital Item Description (DID) specifications [8].
A DID is a plain-text description format of a DI. However, in many multimedia protection schemes, not only the content itself but also the meta-data of the content should be protected against unauthorized access. Therefore, an additional secure representation of DID Model Structure is proposed as IPMP DIDL.

Both representations (in DIDL and in IPMP DIDL) are semantically equivalent. They refer to the identical digital item by different syntax. For every DIDL element, there is an interchangeable IPMP DIDL element which carries valuable content (as shown in Fig. 1). The carried content can be encapsulated in either the unprotected or the protected form. For a protected element, the additional side information is required in order to recover it to its original (unprotected) form. In the specifications, the IPMP Information Descriptor and IPMP General Information Descriptor are defined for representing the necessary side information including tools, mechanisms, and licenses.

Since the IPMP DIDL shares a vast amount of definitions with the DIDL, the hierarchical relationships are illustrated in Fig. 2. For an IPMP capable consumer, it reads in the description using the IPMP DIDL schema. This schema consists of the definitions derived from the generic DID model, the definitions from the DIDL schema, and its own IPMP specific definitions.

3. MPEG-21 REL

The MPEG-21 REL is an XML-based and machine-interpretable rights expression language that declares an authorized distribution for the use of any content, resource, or service owned by specific users. It defines an unambiguous syntax to specify rights.

To fulfill the demand of distribution and protection of digital contents, it is essential that the language provides an authorization model for checking whether the rights specified in the license is validate. To be able express a wide variety of business models and to enable multimedia distribution and usage of all types of digital resources, the MPEG-21 REL is designed to be extensible and flexible.

3.1. Data Model

The most important data model in the MPEG-21 REL is the license structure. As shown in Fig. 3, a license may contain any number of issuers and grants. An issuer is a subject who issues the license. It could be a person, an organization, or a license service. A grant represents a permitted combination of the digital rights management entities.

A grant is constructed by four components. The only mandatory component is the Right, which specifies the allowed action. The Principal component specifies the target to which the license is issued. The Resource component specifies the target which the action is applied to. The Condition component specifies the criterion for deciding the validity of the grant.

In a syntactical expression, the Principal is the subject; the Right is the verb; the Resource is the object; and the Condition is the terms, conditions, and obligations under which the right can be exercised. Hence, a grant is equivalent to the sentence: "under the Condition, the Principal is allowed to exercise the Right on the Resource.”

3.2. Authorization Model
To determine the permission of exercising a certain right, an authorization model is defined in the MPEG-21 REL. As shown in Fig. 4, an authorization proof is the binary result (true or false) of testing the authorization request against the authorization story. An authorization request is composed of several items. They form a structure to express the question: “is it permitted for the principal to perform the right upon the resource during the time interval, according to the authorization context, the set of obtained licenses, and the set of trusted grants?” The authorization context is used to hold additional properties of the request, to be matched with the conditions of the grants.

Upon receiving the request, the REL tool extracts the licenses and grants to construct an authorization story. Each license is used to construct an authorizer (which may contain a story), and the extracted grants are put into the pool of authorized grants. Then, additional authorized grants are derived from the authorizer, recursively. The primitive grants are derived from the authorized grants. The process is equivalent to a de-reference procedure. After all the primitive grants are resolved, the tool starts to match the principal, right, resource, and condition (both the time interval and the context). If the request matches any of the primitive grants, the result is true. Otherwise, the proof is failed.

### 3.3. Extensibility and Profiling

Because the syntax of the REL are described using the XML Schema [9], the MPEG-21 REL can offer a high degree of flexibility in its extensibility. The schema can be organized into three categories: core, standard extension, and multimedia extension.

The core schema defines the general concepts that form the basic architecture of the language, particularly the ones strongly related to a trust evaluation. The standard extension schema defines the concepts that are generally and broadly useful and applicable to DRM scenarios. The multimedia extension schema defines the DRM concepts specifically related to the multimedia content such as books, video, and audio. Both the standard and the multimedia extensions are extended from the core schema. The REL allows, but not limited, to express rights using the defined schemas. Future applications may extend the core schema to define new expression elements.

The extensibility enables the REL to cooperate with a wide range of applications. However, not all applications require the full support of REL. Especially for an embedded application, the limited resource of the device would restrict the DRM functionality to a subset of the REL elements. The profiling process of the REL is to select a set of items for a specific purpose. This process produces a subset of the language, and the REL peers are able to determine whether they are interoperable or not.

### 4. MPEG-21 TESTBED

The MPEG-21 Test Bed is chosen as our implementation platform. We briefly describe the structure of the Test Bed and the existing MPEG-4 IPMPX on it in this section.

#### 4.1. MPEG-4 IPMPX

The MPEG-4 IPMPX system became an ISO standard in October 2002. It provides the functionality to protect and manage the MPEG-4 contents. The MPEG-4 IPMPX system is message-based. The concept of an IPMPX system is a Virtual Terminal that contains a Tool Manager (TM), a Message Router (MR) [4], and several other IPMP Tools. A Tool or the terminal may generate or receive messages to communicate with the others. A message has three basic fields. The type field denotes the type of the message; the source field denotes the component which sends the message; and the destination field denotes the recipient of the message.

To send a message, the originator passes the message to the Message Router. Then, the MR transports the message to the destination component. The interaction to and from the MR are through the message interface. The Tool Manager manages the lifecycle of an IPMP Tool. When a tool is accessed, the TM is responsible for the instantiation, initialization, and connection to the specified control point. After the use of a Tool, the TM disconnects, uninitialized, and destructs it.

In an MPEG-4 elementary stream data path, there exist several control points. A control point is a virtual container in which the triggered IPMP Tools process the content one after another. For the tools that do not
associate with a control point, they act like a service in the system. We will describe a design of REL service in Sec. 5.

4.2. MPEG-21 Test Bed with IPMPX

The purpose of the MPEG-21 Test Bed is to provide a flexible and fair test environment for evaluating delivery technologies for MPEG contents over IP network. The entire system is divided into three parts: server, client, and network. The server is to deliver digital media content with streaming technology to the Clients. The Client receives the media data from the server through the Network Interface and playbacks the media content such as video, audio, or both. The network emulator provides a real-time emulation of various network conditions, such as bandwidth variation, jitter, and packet loss.

Since the MPEG-21 IPMP specifies only the abstract concepts, a concrete architecture is required to realize those concepts. In the Test Bed, a modified version of the MPEG-4 IPMPX is integrated. Based on the MPEG-4 IPMPX specifications, it has components such as Message Router, Tool Manager, and IPMP Tools. Because of the client-server architecture of the Test Bed, the MR and TM exist in both sides. Thus, certain IPMP Tools are implemented in two counterparts. For example, an encryption tool at the server side corresponds to a decryption tool at the client side. The modified IPMPX version on the Test Bed has only three control points. One resides between the DIA and the streamer (PostDIAFilter); and two reside at the input and the output data path of the decoder (PreDecoderFilter and PostDecoderFilter).

5. SYSTEM DESIGN AND IMPLEMENTATION

In this section, we describe the implementation of the MPEG-21 IPMP and REL on the MPEG-21 Test Bed. We design and modify IPMP Tools to provide MPEG-21 IPMP and REL functionality, and map the design to the IPMPX subsystem on the Test Bed [10].

5.1. IPMP Tool for the MPEG-21 IPMP and REL

The major goals of the MPEG-21 IPMP are to protect digital items and to process the IPMP DIDL elements. On the one hand, the Test Bed supports two kinds of digital items, namely, video and audio. An effective way to protect them is to encrypt the data. Therefore, we focus on the cryptographic tool(s) for providing the MPEG-21 protection functionality. On the other hand, the Test Bed adopts the MPEG-4 Initial Object Descriptor (IOD) to transfer the IPMPX related information. Therefore, we place the IPMP DIDL elements into the corresponding Tool Descriptor. The IPMP DIDL elements are extracted and processed at tool initialization stage.

The functionality related to the MPEG-21 REL is the authorization proof. According to the authorization request, the proof engine derives the stories and/or grants, and determines whether the request is granted or not. Any application that incorporates the REL technology can control the consumption of specific content under specific conditions.

In order to provide the above functionalities on the Test Bed, we design an IPMP Tool called IPMP_Info_Engine. From the viewpoint of the IPMP sub-system, it serves as the central control element for the other tools. It is an IPMP information management tool which processes the information obtained via local/remote access. In addition, it is also an REL tool which generates an authorization proof upon receiving a request. We use the REL reference software provided by Content Guard [11] to implement the IPMP_Info_Engine. The reference software delegates the generation of an authentication request to the GUI frontend. However, in a typical scenario, it should be seamlessly integrated into the entire process and handled automatically rather than manually instructed by the user. But our focus here is a feasibility check of our system. In this implementation, the received REL licenses are stored in a local storage. When the IPMP_Info_Engine is accessed, it reads in an XML Query (from the request message or from a local file) and the REL licenses to generate an authorization request. The rest of steps are the same as those of invoking the authorization procedure.

The IPMP_Info_Engine is designed to authorize the user for consuming the right for a specific content and to manage the IPMP DIDL elements. It does not process audio-visual data directly. Thus, the IPMP_Info_Engine Tool is not associated with any IPMP Filters. In other words, the control point code should be CONTROL_POINT_NO (0x00).

5.2. Communication between IPMP Tools

For a user to be able to consume a given content, he/she should have appropriate rights on using this resource. As mentioned in the previous sections, in our scenario, the authorization to a user is equivalent to the authorization to the decryption tool. In other words, when an IPMP tool starts to process the data, it must contact the REL tool to obtain the authorization proof. The interactions between the decryption tool (the DES Tool) and the REL_Info_Engine are illustrated as follows:

1. The DES Tool generates a message to request for the authorization proof.

2. The REL_Info_Engine receives the request and checks the REL licenses stored locally.

3. If the REL licenses are valid, the REL_Info_Engine generates an authorization proof.

4. The authorization proof is sent back to the DES Tool.

5. The DES Tool uses the authorization proof to decrypt the data.

6. The data is then processed by the IPMP_Info_Engine Tool.

The IPMP_Info_Engine Tool then processes the data and generates an IPMP DIDL element for the other tools. It is an IPMP information management tool that processes the information obtained via local/remote access. In addition, it is also an REL tool which generates an authorization proof upon receiving a request. We use the REL reference software provided by Content Guard [11] to implement the IPMP_Info_Engine. The reference software delegates the generation of an authentication request to the GUI frontend. However, in a typical scenario, it should be seamlessly integrated into the entire process and handled automatically rather than manually instructed by the user. But our focus here is a feasibility check of our system. In this implementation, the received REL licenses are stored in a local storage. When the IPMP_Info_Engine is accessed, it reads in an XML Query (from the request message or from a local file) and the REL licenses to generate an authorization request. The rest of steps are the same as those of invoking the authorization procedure.

The IPMP_Info_Engine is designed to authorize the user for consuming the right for a specific content and to manage the IPMP DIDL elements. It does not process audio-visual data directly. Thus, the IPMP_Info_Engine Tool is not associated with any IPMP Filters. In other words, the control point code should be CONTROL_POINT_NO (0x00).
5. Whether it can process the incoming data or not.

2. The Message Router delivers the message to the IPMP_Info_Engine.
3. The IPMP_Info_Engine verifies the request and exercises the authorization proof.
4. The IPMP_Info_Engine generates a message containing the result of proof checking.
5. The Message Router delivers the proof result message to the DES Tool.

According to the proof result, the DES Tool decides whether it can process the incoming data or not.

5.3. Content Protection Scheme

To protect a broadcast streaming content, we propose a scheme which incorporates the aforementioned MPEG-21 tools. The rights management aspect is achieved by verifying the license, as mentioned in the previous section. The protection aspect is shown in Fig. 5. There are two layers of encrypted streams. The content stream is delivered through the Layer-1 protection. We use the DES block cipher in this layer. Since the DES encryption/decryption is symmetric and simple, it is suitable for bulky data. However, there are two major issues using this technique. The first one is how a client can securely receive the key decrypting the data; and the second is that a simple block cipher is less secure than a complicated cipher. The first issue can be solved by transmitting the key through a trusted secure channel. The solution to the second issue is a trade-off between computation complexity and security. Since the streaming data is very large in size, it is costly to replace the simple cipher with a complicated one. An acceptable solution is applying the simple cipher on the bulky data but with frequent key change. Combining the above two solutions leads to the design of a two-layer protection scheme.

The Layer-2 protection is a reliable and secure channel for delivering the Layer-1 keys. It is more secure than Layer-1 channel because keys are encrypted using the asymmetric cryptographic (public key) algorithms. We assume that the transmission of the keys is handled by a key server. It may be a locally integrated unit of the streaming server, or a separate remote service. The key databases located at the streaming server and the key server are perfectly synchronized. The synchronization mechanism is implementation dependent and is out of the scope of this paper. The interactions between the client and the key server are specified as follows:
1. The client requests the decryption key by sending a request with its identity.
2. The key server looks up the database to check the request identity and the other required conditions.
3. If the request is valid, the key server encrypts the data decryption keys using the client’s public key.
4. The key server sends the returned message containing the encrypted data keys.
5. The client recovers the data keys using its own private key. Then the data keys can be used to decrypt the content bit stream.

Using the above mechanism, the encrypted content can only be decrypted by a client who has the right identity.

6. APPLICATION EXAMPLES

To demonstrate the features and capability of our design and implementation, we construct three application examples. The first case is secured online playback; the second one is playback with free preview; and the third one is super distribution – online and offline playback.

6.1. Application Example 1: Online Playback

This scenario demonstrates how the “playback” right is exercised in a real-time streaming system. After the session setup, the client receives a license from the server. In this basic-form grant, the principal is the client’s identity, the right is “play”, and the resource is the video’s identity. The tricky item is the condition. We choose the “exerciseMechanism” condition with server parameters to delegate the authorization decision to a remote server.

Upon consuming the video stream, the DES Tool asks the IPMP_Info_Engine to do an authorization proof. According to the license, the engine contacts the remote server and sends out the request with the client’s identity. The server in this system plays both roles as an authorization server and also a key server. If the client is entitled to consume this video bit stream, the server returns a message containing the data keys which are encrypted using the Layer-2 technique. If the client is not allowed to playback this video stream, the server responds with an empty key set. Since the principal, the right, and the resource are matched with the REL request, the validity of the grant depends only on the condition. If we receive a “false” condition (empty key), the authorization proof fails and the DES Tool refuses to consume the data (Fig. 6). Otherwise,
the DES Tool can decrypt the data using the returned keys.

6.2. Application Example 2: Preview

In a typical video/audio online shopping scenario, it often allows the consumers to preview a short video clip, say, for 30 seconds before starting the purchase transaction. This example demonstrates how to implement and control the “preview” behavior using the MPEG-21 REL and DRM concepts.

We assume that the customers can preview a few seconds of a video clip without any license. It is achieved by using a counter in the DES Tool. The counter indicates the remaining number of macroblocks of a video stream that can be viewed. The authorization proof is delayed until the counter reaches zero. Then, the IPMP_Info_Engine pops up a dialog box asking the user to input a valid license and a query file. After the user enters the data into the dialog window, the IPMP_Info_Engine performs the authorization proof process. An invalid license or query file would fail the proof and discontinue the playback operation.

6.3. Application Example 3: Super Distribution

This example shows a scenario that occurs in the portable device applications. The so-called “Super Distribution” describes the situation that the content and the right object (the license) are delivered separately. In the following application, we are able to perform online authorization as well as offline verification. It is a very useful scenario for producing and distributing contents for mobile devices. The content is delivered in a protected form. In the first application example, the verification of playback right is performed online in real time. Therefore, we cannot play it back offline. On the other hand, a mobile device may not always be connected to a network. We thus like to design a license that supports both online and offline verification.

A sample license that allows both online and offline verification is shown in Fig. 7. The license contains two grants. Grant-1 is for online situation, and Grant-2 is for offline situation. The condition elements in both grants are encapsulated in an “allConditions” element which means the logical “AND” operation is applied to all the declared conditions inside. In Grant-1, the “exerciseMechanism” specifies a remote authorization server, and the “validicityInterval” specifies the valid date execution duration. In Grant-2, the “exerciseMechanism” specifies a local verification, and the “exerciseLimit” specifies a valid number of offline playback times.

The verification flow of this license is shown in Fig. 8.

Because the authorization proof process guarantees the matching order of grants, Grant-1 is always verified before Grant-2. In addition, the online verification in our design is the ultimate verification authority. Once an “unauthorized” result is returned from the server, the subsequent local verification should fail. This rule is implemented by the “validators”. The first validator checks the network connection first. If the network is unreachable, the validator simply returns a false condition. If the remote server returns an empty decryption key set (means unauthorized), the previously obtained decryption keys are deleted. Otherwise, the keys are replaced with the new set. Whenever Grant-2 should be verified, the second validator checks the decryption keys first. If there are no decryption keys, the playback is not authorized. Otherwise, the exerciseLimit is examined. If it reaches the maximum number of local verifications, the result is false and the keys are deleted.
Two execution screenshots are shown in Fig. 9. The upper one shows that a local verification is passed, and the lower one shows a warning message indicating that the maximum number of local verification is reached.

7. CONCLUSIONS

DRM is an important entity in many multimedia applications. Our aim is to construct a DRM system which can provide both content protection and rights management. In constructing our system in this paper, we use the concepts and tools of MPEG-21 IPMP, MPEG-21 REL, and the MPEG-21 Test Bed with IPMPX. The MPEG-21 IPMP defines high level concepts for protecting Digital Items. The MPEG-21 REL is able to implement a broad range of rights expressions by the extension and profile mechanisms. It also provides an unambiguous right authorization and control model for resolving complex rights descriptions.

Because MPEG-21 IPMP is defined at conceptual level, we choose the MPEG-21 Test Bed as our implementation platform. The Test Bed includes an implementation of a modified IPMPX, which adopts many interfaces defined in the MPEG-4 IPMPX. We designed a two-layer content protection scheme in delivering and decrypting the multimedia contents. To implement the scheme, the IOD was modified to carry an REL license in the Tool Descriptor. Then, the MPEG-21 REL reference software was integrated into the Test Bed as an IPMP Tool (IPMP_Info_Engine). We also modified the cryptographic (DES) Tool so that it communicates with the IPMP_Info_Engine for receiving the authorization and the decryption keys.

Finally, we constructed three examples to show that our DRM system can provide many types of practical services. The first example demonstrates the streaming content protection with on-access right verification. The second one shows how to design and to use a license for allowing users to preview video clips. The last one shows a more complicated case -- rights management for Super Distribution. In this case, we design a license which enables both online authorization as well as offline verification.

8. ACKNOWLEDGEMENT

This work was supported by National Science Council (Taiwan, ROC) under Grant NSC 94-2219-E-009-011.

9. REFERENCES