

P2P 방식의 디지털 권한 관리 시스템

정재린, 최형은, 김윤형[○], 김태형
한양대학교 컴퓨터공학과
{jerin, hechol, yunhyung[○], tkim}@cse.hanyang.ac.kr

Digital Rights Management System for P2P Network

Jerin-Jung, Hyong-Eun Choi, Yun-Hyung Kim[○], Tae-Hyung Kim
Dept. of Computer Science & Engineering, Hanyang University

요 약

본 논문에서는 MPEG-21 프레임워크를 Peer-to-Peer Network 상에서 구동시킬 때의 문제점을 해결하면서, P2P의 장점을 최대한 이용할 수 있도록 하는 P2P 방식의 디지털 권한 관리 시스템(P2PDRMEngine)을 설계한다. MPEG-21 프레임워크의 여러 구성 요소들 중 IPMP(Intellectual Property Management and Protection components), DI(Digital Item), Multimedia Contents 등의 구조에 몇 가지 요소를 추가하여 P2PDRMEngine과 MPEG-21 프레임워크가 서로 유기적인 동작을 할 수 있도록 설계하고 또한, 디를 생산, 배포, 사용할 때 사용자에게 편리성을 제공 해주는 기능도 고려한다. P2PDRMEngine은 기존의 MPEG-21 프레임워크의 수행에 지장을 주지 않으면서 보안적인 측면과 요소 간의 기능을 강화하는 데 초점을 맞추어 설계한 시스템이다.

1. 서 론

Peer-to-peer Network(이하 P2P)는 사용자간 파일 공유가 편리하다는 이유로 가장 각광받는 시스템이며 Pruna, e-Donkey등이 P2P를 이용한 대표적인 P2P 애플리케이션이다. 그러나 최근 이러한 P2P 애플리케이션들에 대해 저작권 침해 문제가 발생하고 있어 저작권자들은 Digital Rights Management(이하 DRM)기술을 멀티미디어 파일에 적용함으로써 허용된 사용자들 외에는 멀티미디어 파일을 사용할 수 없게 하여 저작권 문제를 해결하고 있다. 그러므로 P2P 상에서 공유되는 파일들에 대해서 DRM 기술을 적용한다면 P2P의 편리한 공유 환경을 그대로 이용하는 것 뿐만 아니라 저작권자들은 자신의 지적재산권을 보호할 수도 있다.[1] 하지만 DRM 기술은 각 회사마다 다른 알고리즘으로 개발되기 때문에 ipmptool간, Device간 호환이 되지 않는 특징을 가지고 있어 Moving Pictures Experts Group(이하 MPEG)에서는 디지털 콘텐츠의 전자상거래에 필요한 기술들과 통합 기술의 표준화를 위한 프레임워크인 MPEG-21의 개발 중에 있다.[2] MPEG-21의 한 분과로 DRM 기술 간의 호환성 문제를 해결하기 위한 MPEG-21 Intellectual Property Management and Protection components(이하 IPMP)도 포함된다.[3] MPEG-21에서 제안하는

Resource Delivery testbed[5]에서는 서버에서 모든 클라이언트의 요청을 받아서 해당 콘텐츠에 대한 모든 서비스를 제공하는 서버-클라이언트 구조(이하 C-S 구조)로 설계되어 있다. 본 논문에서는 MPEG-21 IPMP 기술을 P2P 환경에 적용시키는 방안을 모색하여, P2P 환경에서도 MPEG-21 IPMP 기술이 고유의 기능을 수행할 수 있도록 해주는 P2PDRMEngine의 설계를 제안하고자 한다.

2. 배경지식

2.1 MPEG-21

멀티미디어 콘텐츠를 사용하는 환경이 급변하고, Terminal, Network Capability, Device 등이 다양해지면서 사용자들은 장소와 환경의 제약 없이, 언제 어디서든 정보에 접근하려고 하고 있으나 현재 새로운 멀티미디어 사용 환경을 효율적으로 지원하기 위한 톨이 존재하지 않는다. 이에 따라 MPEG에서는 기존 MPEG 멀티미디어 파일의 관리를 위한 새로운 프레임워크인 MPEG-21의 연구가 진행 중에 있다. MPEG-21은 현재 여러 분과로 나뉘어 표준화가 진행되고 있고, 연구 목적은 MPEG-1,2,4처럼 동영상이나 오디오에 대한 새로운 알고리즘을 개발하는 것이 아닌 멀티미디어 콘텐츠의 사용과 거래를 위한 프레임워크 구성 요

소 간의 관계와 조화에 관한 표준을 만드는 데 있다.
[2]

2.2 Digital Item

MPEG-21 프레임워크가 처리하는 기본적인 콘텐츠를 Digital Item(이하 DI)이라고 한다.

DI는 멀티미디어 콘텐츠의 식별자, 서술자를 포함하는 유통, 처리의 최소 단위이다. Digital Item Declaration(이하 DID)은 DI의 구조적인 부분을 정의한 MPEG-21의 한 분과이며, Digital Item Declaration Language(이하 DIDL)은 정의된 DID를 이용하여 DI를 생성할 때 쓰이는 XML 구조의 언어이다.[3] XML 구조에서 알 수 있듯이 하나의 DI는 동시에 여러 개의 아이템으로 구성될 수 있다. 예를 들어, 하나의 Music Album은 여러 개의 Music Track으로 이루어지며, 같은 Music Track도 서로 다른 Bit-rate의 Music Track으로 존재할 수 있다. 이와 같은 Music Track은 DI에서 각각의 Item으로 나뉘거나 하나의 Item내에 모든 Track을 넣을 수도 있다.[2]

2.3 MPEG-21 표준화

MPEG-21에서 Digital Item의 관리에 초점을 맞춘 분과는 크게 두 부분이다.

첫째는, 7번째 분과인 Digital Item Adaptation(DIA)으로 각각 다른 환경의 터미널 간의 멀티미디어 파일을 전송할 때, 수신측 터미널의 환경에 가장 적합하게 해상도나 기타 데이터를 변경하여 보내주는 것을 목적으로 하는 분과이다. 데스크 탑의 환경에 맞는 고화질, 고용량의 멀티미디어 파일은 PDA와 같은 기기에 적합하지 않으므로, 송신측에서 해상도를 PDA와 같은 기기에 맞는 해상도로 변경하여 전송하는 데 목적이 있다.[2]

둘째는, 4번째 분과인 IPMP로서 각 기관이나 회사에서 개발하여 사용 중인 DRM Tool(이하 ipmptool)들을 통합, 관리하여 멀티미디어 파일의 지적 재산권 소유자와 멀티미디어 파일을 얻고자 하는 사용자가 ipmptool을 사용하기 편리하게 하는 데 목적이 있다.[2][3] 본 논문에서는 IPMP에 초점을 맞추어 설명한다.

2.4 IPMP

IPMP는 MPEG-21의 한 분과로, 모든 사용자들이 자신의 지적 재산권을 DI 내에 표현하여 권리를 영유할 수 있도록 하는 데 목적이 있다. 이에 따라 다양한 네트워크와 장치에서 모든 사용자들의 권리나 이익들이 일관적이고 신뢰성 있게 관리된다고 확신할 수 있는 프레임워크의 제공을 목표로 한다.[3]

IPMP는 DI 저작자의 저작권을 표현하고, 보호하기 위

해 Rights Expression Language(이하 REL)과 Rights Data Dictionary(이하 RDD)를 이용한다.[2] DI의 내용 중 item의 정책과 같은 부분은 저작자의 필요에 따라 암호화한다.[3]

IPMP의 구성 요소는 IPMP Engine, IPMP Tool Manager, ipmptool로 나누어진다. IPMP Engine은 IPMP Tool Manager와 ipmptool을 호출하여 DI를 암호화 또는 복호화한다. IPMP Tool Manager는 ipmptool을 호출하고, IPMP Tool Server로부터 ipmptool을 다운로드 받고, ipmptool을 관리하는 기능을 한다. ipmptool list는 현재 Terminal 상에 등록된(존재하는) ipmptool의 list를 나타낸다. ipmptool은 각각 Publisher마다 고유한 알고리즘을 사용하도록 만들어 지기 때문에 하나의 Engine으로 구현될 수 없다. 이러한 문제점을 해결하기 위해 IPMP Tool Manager가 ipmptool을 관리한다. IPMP Tool Manager는 ipmptool list를 참조하여 local terminal에 존재하는 ipmptool의 id를 관리한다.[3]

[표 1]은 IPMP Tool Manager의 동작 과정에 대한 설명이다.

[표 1] IPMP Tool Manager의 동작과정

1. IPMP Engine이 DI의 내용 중 ipmptool id라는 문자열을 찾는다.
2. IPMP Engine이 ipmptool id를 IPMP Tool Manager에게 넘겨준다.
3. IPMP Tool Manager는 ipmptool list를 참조하여 local 상에 존재하는지 확인한다. (존재할 경우, 4항과 5항은 해당사항 없음)
4. 존재하지 않을 경우, 지정된 IPMP Tool Server로 해당 ipmptool id를 요청한다.
5. 해당 ipmptool의 download를 마친 후 IPMP Tool Manager는 ipmptool list에 해당 id를 추가한다.

2.5 REL, RDD

REL과 RDD는 MPEG-21 프레임워크에서 권한을 표현하기 위한 부분들로서 RDD에 정의된 term들을 이용하여 REL로 나타낼 수 있다.[2][3]

REL은 DI의 배포, 생산, 사용에 대한 rights,

conditions, fees를 정의하는 언어이며 여러 경우에 대해 access와 use에 대해 정의하지만, 지불과 관련한 부분은 다루지 않는다. 4개의 Grant로 구성되는데 재생, 인쇄, 복사 등에 관한 내용을 나타내는 Permission, 조건을 나타내는 Condition, 사용자를 나타내는 Principle, 자원을 나타내는 Resource등을 구성요소로 가진다. 예를 들면, "John은 타이타닉 DI에 있는 타이타닉.mpg 파일을 매회 1달러를 지불하여 시청할 수 있다."의 경우 Condition은 "재생 시 마다 1달러에 시청할 수 있다.", Principle은 "사용자 John", Resource는 "타이타닉.mpg"가 된다.[5]

RDD는 명확하고 구조적이며 유일한 terms로 REL을 지원한다. 구조적으로 설계는 끝났지만, 새로운 terms의 추가가 가능하다.[5]

REL이 IPMP와 연동하여 DI의 사용의 제한, 허가를 제어할 수 있는 방법은 세가지로 첫째, DI 자체에 표현하는 방식, 둘째, DI 내에서 DI 외부로 참조하는 방식, 마지막으로 License Service를 이용하는 방식 등이 있다.[3] 본 논문에서는 첫 번째 방식인 DI 자체에 표현하는 방식을 이용하였다.

3. 연구동기

3.1 DI, Resource File과의 관계

P2P 구조는 C-S 구조와 달리 피어들을 관리하는 별도의 서버가 존재하지 않는다. MPEG-21[5]에서 제안하는 testbed와 같은 C-S 구조에서는 클라이언트로 부터 요청받은 서버가 DI와 Resource file을 전송하고 관리한다. DI가 하나 이상의 Resource file을 포함할 경우 Resource file은 DI 내에서 주소로 표현된다.[2] 따라서 DI를 통해서만 Resource file에 접근할 수 있으며 DI와 Resource file이 분리되어 존재한다.

P2P에서는 DI와 Resource file의 전송이 피어 간에 분리되어 발생할 수 있기 때문에, 임의의 피어가 DI를 갖고 있을 경우라도 DI에 해당하는 모든 Resource를 갖고 있다고 확신할 수 없으며 Resource file을 갖고 있는 피어라 할지라도 DI가 존재하지 않을 수도 있다. 하지만 MPEG-21에 따라 Resource file에 접근하기 위해서는 반드시 해당 DI가 있어야 하므로 기존의 Resource file을 P2P 환경에서 그대로 이용할 수 없다. 왜냐하면 현재까지의 MPEG 표준 Resource file에는 해당 DI의 어떤 정보도 포함하지 않기 때문이다.[4] 이러한 문제점을 해결하기 위해 본 논문에서는 Resource file로도 해당 DI를 찾을 수 있는 방법을 제안하고 있다.

3.2 MPEG-21에서의 Resource file 암호화

MPEG-21 프레임워크가 존재하지 않는 환경에서는 Resource file을 실행할 수 있다. IPMP 표준에는 특별한 경우 DI 내에 Resource data가 포함될 수 있는데, IPMP에서는 DI의 내용을 암호화할 때 DI 내에 포함된 Resource data를 암호화하는 방법을 사용한다.[3]

URL로 표현된 것과 달리 Bitstream data가 DI에 직접 포함된 경우, 예를 들어 워터마킹과 같은 암호화 기법을 이용할 경우 해당 Resource file의 일부나 전체를 DI 내에 포함시킬 수 있다. 이 경우 DI 암호화 과정 중 Resource 내용도 암호화가 발생한다.

하지만 URL로 표현된 Resource file의 경우 파일에 대한 암호화는 일어나지 않고 단지 URL에 대해서만 암호화가 일어난다. 따라서 Resource file 자체는 기존의 format을 유지하고 있기 때문에 MPEG-21 프레임워크를 통하지 않고도 기존의 멀티미디어 재생기를 통하여 사용자 임의대로 재생할 수 있으며 사용자 간의 Resource file 공유도 제어할 수 없는 상황이 발생한다.

본 논문에서는 후자의 문제점 해결에 초점을 맞추어 Resource file의 header 내용을 암호화하여 기존의 재생 플레이어를 이용한 Resource file의 재생을 가능하지 않도록 하는 기법을 제안하고 있다.

3.3 P2P 기반 MPEG-21

P2P상에서 MPEG-21을 효과적으로 이용하기 위해서는 다음과 같은 개선점이 필요하다.

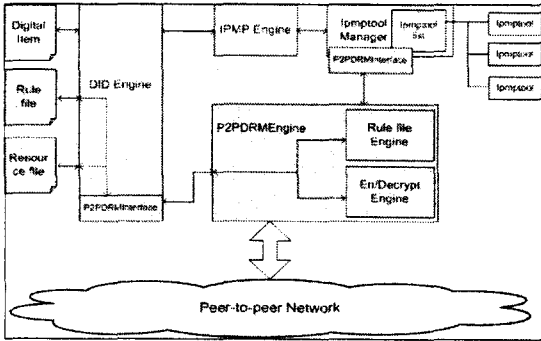
첫째, 3.1절에서 언급한 것과 같이 P2P 상에서는 DI와 Resource file이 서로 다른 피어에 존재하므로 Resource file을 가지고 DI를 찾는 것은 불가능하다는 것이다.

둘째, 3.2절에서 언급한 것과 같이 MPEG-21에서는 Resource file을 암호화하지 않기 때문에 Resource file을 사용자가 단독으로 재생할 수 있다는 것이다.

마지막으로, missing tool이 있을 경우 즉, 원하는 ipmptool을 찾을 수 없는 상황에 기존의 서버에게 요청하는 IPMP Tool Manager의 동작 방식으로 인해 발생하는 서버의 과부하의 문제, 서버의 작동 중지로 생기는 원활한 서비스 제공을 하지 못하는 문제 등이 생겨난다.

4. P2PDRMEngine 설계

[그림 1]은 논문에서 제안하는 시스템의 전체 구조를 나타낸다.



[그림 1] 시스템 구조

움영이 없는 부분은 기존의 MPEG-21의 구성 요소들이고 음영으로 된 부분은 논문에서 제안하는 시스템의 구조이다.

MPEG-21 표준에 따라 [그림 1]의 왼쪽 부분에 있는 DID Engine에서 입력을 받는다. 제안하는 구조를 이용하지 않은 DI는 DID Engine을 거쳐 IPMP Engine을 통해 IPMP Tool Manager, ipmptool과 통신이 가능하다. 제안하는 구조를 이용하는 파일들은 P2PDRM Engine에서 처리된다. Rule file과 Resource file을 입력으로 받으면 P2PDRM Engine을 호출하도록 [그림 1]에 보이는 것과 같이 DID Engine에 P2PDRMInterface를 두었고, IPMP Tool Manager가 P2P를 통해 ipmptool을 검색할 수 있도록 IPMP Tool Manager에 P2PDRMInterface를 두었다.

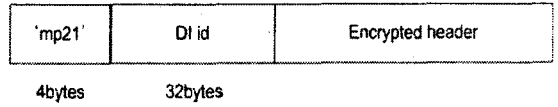
4.1 Resource file Header

3.3절에서 언급한 개선요소 3가지 중 첫째, 둘째 문제를 해결하기 위해 기존 Resource file의 header를 변경하였다.

우선 첫번째 문제를 해결하기 위해 Resource file header의 암호화가 끝나면, Resource file header에 이 Resource file이 MPEG-21의 Resource file이라는 명세와 해당하는 DI의 id를 기록한다. 정의한 Header의 구조는 [그림 2]에 나타내었다.

두번째 문제의 해결을 위해 DI 생성시 사용자의 요청에 따라 P2PDRMEngine의 En/DecryptEngine이 해당 Resource file의 header를 암호화한다. 암호화를 함으로써 MPEG-21 프레임워크가 갖춰지지 않은 환경에서의 재생을 막는다. 암호화에 대한 구체적인 내용은 다음절에서 설명하도록 한다. MPEG 프레임워크가 설치된 환경에서 암호화된 Resource file에 대해 사용자의 접근이 있을시 header를 바로 복호화 한다. 복호화의 내용도 다

음절에서 설명하도록 한다.

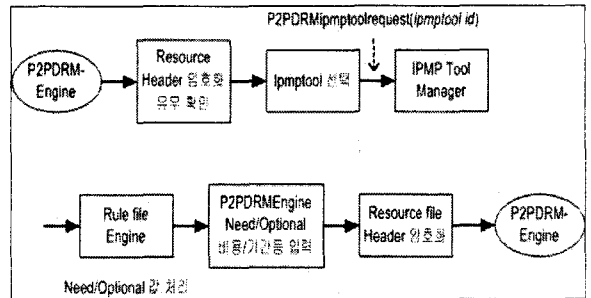


[그림 2] Header의 구조 정의

[그림 2]에 보이는 것과 같은 Header 구조를 사용하면 Resource file의 Header에 DI의 id값으로 DI를 찾을 수 있으므로 기존의 Resource를 가지고 DI를 찾을 수 없었던 첫 번째 문제점을 해결하였다.

4.1.1 Header 암호화/복호화

암호화 과정은 [그림 3]과 같이 En/Decrypt Engine에서 Resource file header를 체크하고, 암호화가 되어 있지 않은 경우에 Resource file header를 암호화한다. 자세한 암호화 과정을 [그림 3]에 나타내었다.

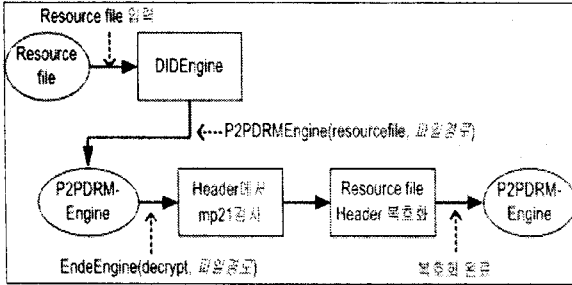


[그림 3] 암호화 과정

암호화 과정을 살펴보면,

- (1) 이미 DI의 resource file인지 알아보기 위해 resource file header를 검사한다.(Resource Header 암호화 유무 확인)
- (2) DI의 resource file이 아닌 경우에는 DI를 암호화 할 ipmptool을 선택한다.(ipmptool 선택)
- (3) Rule file Engine에서 ipmptool에 대한 rule file을 parsing하여 사용자로부터 해당하는 값을 받고, En/Decrypt Engine에서 resource file header를 암호화 한다. (Resource file Header 암호화)

복호화 단계에서 En/Decrypt Engine은 header의 구조를 변경하지 않으며 단지 Resource file header를 복호화한다. [그림 4]에 복호화 과정을 나타내었다.



[그림 4] 복호화 과정

- (1) P2PDRMEngine에서 Resource file을 입력으로 받을 경우, En/Decrypt Engine을 호출한다.
- (2) En/Decrypt Engine은 SHF의 시작위치에 문자열 mp21이 존재하는지 검사한다.
- (3) mp21이 존재하는 경우, 사용자에게 해당 DI를 선택하도록 하고 존재하지 않는 경우, En/Decrypt Engine이 Resource file Header에서 DI의 id 값을 검사하여 P2P를 통해 검색한다. MPEG-21 DIDEngine에 해당 DI를 넘겨준다.
- (4) MPEG-21을 통해 정상적으로 Resource file Decryption 요청이 오면, P2PDRMEngine이 En/Decrypt Engine에게 Resource file Decryption 요청한다.

4.2 Rule file

MPEG-21 프레임워크의 목적 중 한 가지는 모든 사용자가 creator와 user의 역할을 수행하게끔 하는데 있다. 하지만 사용자가 MPEG-21의 표준에 맞게 Digital Item을 생성하기 위해 Resource file에 맞는 DI를 만들어야 한다 - Resource file과 DI의 연결.

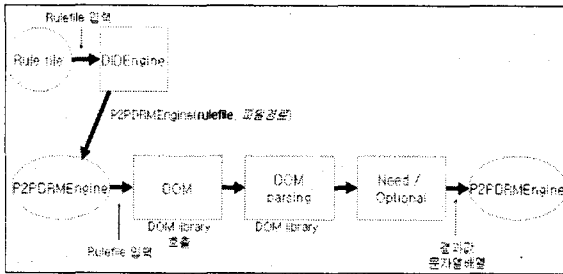
DI 내용 중 특히 IPMP와 관련한 내용들은 REL로 작성되어야 한다. 게다가 MPEG-21 Digital Item의 스펙에 맞추어 IPMP 관련 내용을 작성해야 하므로 매우 까다로운 일이 아닐 수 없다. 수많은 ipmptool이 존재하지만, 각각 ipmptool의 사용법은 암호화/복호화에 상관없이 정해진 Rule에 따라 수행되므로 본 논문에서는 해당 Rule을 XML 구조의 rule file로 표현하는 방법을 제안하여 IPMP와 관련한 내용들을 REL로 작성하는 경우에 편리성을 제공하고자 한다. 논문에서 제안하는 Rule file의 구조는 [표 2]와 같다.

[표 2] Rule file 구조

namespace 정의	<pre> <did> <xsd:import namespace= "urn:mpeg:mpeg21:2003:01-REL-SX-NS" schemaLocation="rel-sx.xsd"/> <xsd:import namespace= "urn:mpeg:mpeg21:2003:01-REL-R-NS" schemaLocation="rel-r.xsd"/> <xsd:import namespace= "http://www.w3.org/2000/09/xmldsig#" schemaLocation= "http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/xmldsig-core-schema.xsd"/> </pre>
정의하는 rule구조	<pre> <rule> <ipmptool id></ipmptool id> <play> <need> <validityInterval> </validityInterval> <exerciseLimit></exerciseLimit> <feePerUse></feePerUse> <feeFlat></feeFlat> </need> <optional> <territory></territory> </optional> </play> </rule> </did> </pre>
설명	<p>validityInterval - 사용기한 exerciseLimit - 사용횟수 feePerUse - 회당사용료 territory - 사용가능한 지역(국가) feeFlat - 고정사용료</p>

[표 2]와 같이 rel-sx를 이용하면 DI를 사용하는 모든 권한에 대해 하나의 조건으로 통일할 수 있고, rel-sx와 rel-mx를 동시에 사용하면 권한마다 다른 조건을 설정할 수 있다. 이외에도 MPEG-21 REL 표준 schema('rel-sx' 'rel-r')에 따라 다른 rule들을 정의할 수 있다.

Rule file Engine은 해당 Rule file로부터 Need, Optional을 Parsing하여 P2PDRMEngine으로 결과를 넘겨준다. [그림 5]는 Rule file Engine의 동작과정을 나타낸다.

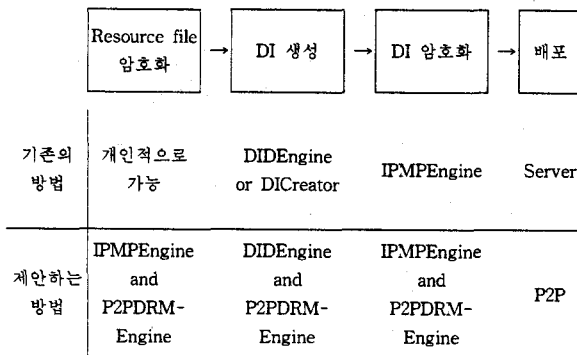


[그림 5] Rule file Engine의 동작과정

Rule file Engine은 Rulefile을 DOM형태로 파싱한 후 Need와 Optional을 가려서 P2PDRMEngine으로 넘겨주는 역할을 수행한다.

4.3 P2P를 이용한 IPMP Tool Manager

기존 IPMP Tool Manager가 ipmptool을 찾기 위해 IPMP Tool Server로 요청하던 것을 본 논문에서 제안하는 시스템에서는 P2P의 검색을 통해 찾을 수 있도록 설계하여 서버의 과부하로 인한 문제점, 서버의 작동중지로 인한 서비스를 제공 받을 수 없는 문제점의 해결 뿐만 아니라 IPMP Tool Manager는 P2P와 메시지 통신을 하면서 ipmptool을 관리하고 ipmptool을 P2P 상에서 검색할 수 있는 장점을 가지게 되었다. [그림 6]은 IPMP Tool Manager의 동작 과정을 기존의 방법과 논문에서 제안하는 방법을 비교 및 대조하여 나타내었다.



[그림 6] IPMP Tool Manager 동작과정

[그림 6]에 나타난 것과 같이 논문에서 제안하는 방법을 이용한다면, Resource file의 암호화를 통해 Resource file을 임의의 사용자가 재생 가능토록 하는 문제점을 해결할 수 있고, Rule file을 이용하여 IPMP와 관련된 내용들을 쉽게 작성할 수 있고, P2P를 통한 DI의

배포를 통해 클라이언트가 DI를 쉽게 검색할 수 있도록 하는 장점을 보인다는 것을 파악할 수 있다.

5. 결론

본 논문에서는 C-S 구조 기반의 MPEG-21 프레임워크를 P2P 네트워크에서 구동 시킬 때의 문제점들을 파악하고 해결하기 위해 P2PDRMEngine의 설계를 제안하였다.

P2P를 이용하는 경우 Resource를 가지고 DI를 찾을 수 없는 문제를 Resource의 Header 구조를 정의하여 가능하도록 설계하였고, MPEG-21에서의 Resource file의 암호화를 하지 않아 생기는 문제점의 해결을 위해 Header 정보의 암호화 및 복호화 기법을 제안하여 해결하였다. MPEG-21에서의 IPMP Tool Manager의 동작과정에서 생겨나는 문제점을 논문에서 제안하는 방법과 비교 및 대조하여 논문에서 제안하는 방법이 효율적임을 역설하였다. 또한 Rule file을 정의하여 DI의 생성 및 배포, 사용에 있어 사용자에게 편리성을 제공하였다.

논문에서 제안하는 시스템을 이용하면, 사용자는 ipmptool들의 세부 내용을 모르더라도 P2PDRMEngine과 Rule file을 통해 편리하게 Digital Item을 생산, 관리, 사용할 수 있는 편리성을 제공 받을 수 있고, Resource file은 반드시 P2PDRMEngine과 MPEG-21 프레임워크의 ipmptool을 통해서만 접근 가능하도록 설계되었으므로, 사용자들 간에 DI와 해당 Resource file의 공유가 일어나더라도 ipmptool을 거치지 않고는 재생할 수 없기 때문에 보안에 충실한 시스템을 기대할 수 있다.

6. 참고 문헌

- [1] TON KALKER, DICK H. J. et al; "Music2Share - Copyright-Compliant Music Sharing in P2P Systems"; Proceedings of the IEEE; vol. 92; 6, June, 2004
- [2] Jan Bormans, Keith Hill; "MPEG-21 Overview v.5"; N5231, Shanghai, China; October, 2002
- [3] ISO/IEC FCD 21000-4 N7426; "Study of IPMP components"; July, 2005
- [4] 정제창; "그림으로 보는 최신 MPEG"; Chapter 6; 교보문고; 1997
- [5] ISO/IEC JTC1/SC29/WG11 N 4333; "Information Technology - Multimedia Framework(MPEG21) - Part 1 : Vision, Technology and Strategy"; July, 2001