디지털 컨텐츠의 저작권 관리 및 보호를 위한 OPIMA 규격의 조사 및 분석

최재각* · 이종극* · 김태석* · 구본호**

1. 서 론

디지털 컨텐츠의 저작권 관리 및 보호에 관한 최초의 국제표준은 MPEG의 IPMP이다[1]. MPEG IPMP는 사용규정(Usage rule)과 IPMP API에 대해 논의하고 있다. 여기서 암호방식의 다양성을 인정해 구현방법은 규격에 규정하지 않고, 호환성을 위해 인터페이스만을 정하고 있다. IPMP는 암호엔진을 말하는데, 구현은 각자가 자기의 방식대로 구현할 수 있기 때문에 얼마나 많은 IPMP 엔진이 존재할 지 알 수 없다. 하지만 서로 다른 IPMP들 끼리 대화는 가능하다. API를 공유하기 때문이다. 그러나 단말을 만드는 입장에서는 수많은 IPMP를 제한된 단말에 다 구현하기는 어렵다. 이러한 문제를 해결한 것이 바로 OPIMA VM (Virtual machine)이다. OPIMA VM은 서로 다른 IPMP사이에 통역을 담당한다[2].

OPIMA(Open Platform Initiative for Multimedia Access)는 IEC 산하의 ITA program에서 개발된 디지털 컨텐츠 저작권 관리 및 보호에 관한 표준화 규격으로 본 논문에서는 이 규격을 근간으로 OPIMA의 IPMP(Intellectual Property Management and Protection)기술에 대해 기술하기로 한다[3].

* 동의대학교 공과대학 컴퓨터응용공학부
** 경일대학교 공과대학 제어계측공학과

OPIMA는 컨텐츠나 서비스 공급자가 가능 고객의 범위를 넓히고, 다중 컨텐츠 보호 장치 하에서 그들의 고객이 다양한 형태의 컨텐츠나 서비스에 접근하는 것을 가능하도록 하는 체제를 개발하기 위해 설립되었다. 이 규격에서는 OPIMA platform의 구성 요소에 대해 기술하는데, 이 platform은 가치사슬(value-chain)의 참가자들에게 서비스에 관련된 권리의 토대 하에서 멀티미디어 서비스를 획득, 공급, 처리 및 소비할 수 있는 능력을 공급하는 것을 목적으로 한다. 이 규격은 특히, 디지털 컨텐츠의 저작권 관리 및 보호에 대해 설명한다.

이 규격은 OPIMA 호환 시스템의 구조 및 요구기능에 대해 설명하며, 더 나아가서 보안 프로토콜(security protocol)과 API(Application Programming Interfaces) 및 상호작용을 가능케 하는 기능적 구조에 대해 제시한다.

본 논문의 구성은 다음과 같다. 먼저 제2장에서 OPIMA 구조의 informative part 에 대해 기술하고 제3장에서 OPIMA 구조의 normative part에 대해 설명한다. 그리고 마지막으로 결론을 맺는다.

2. OPIMA Architecture(Informative)

본 장에서는 OPIMA 환경의 구조에 대한 개요를 기술하며, OPIMA Peer로 표기되는 OPIMA

호환 기종간의 상호 작용을 위한 체제(frame-work)를 제시한다. 이 체제는 코드가 사용자 환경에 의해 간섭 받지않고 수행되도록 설계되어있다. 그러므로 코드 및 컨텐츠는 컨텐츠에 관련된 rule에 의해서만 사용될 수 있다. 또, OPIMA Peer는 기저조직(back-end infrastructure)과 인터페이스 하도록 사용될 수도 있다.

OPIMA 규격은 디지털 컨텐츠를 처리하는 장비 및 컨텐츠의 종류에 대해 독립적이며, 여기서 컨텐츠란 모든 종류의 멀티미디어 및 실행 가능한 것(executable)들을 다 포함한다. Rules 란 용어는 주어진 장비에서 컨텐츠가 어떻게 사용되어야 하는가를 규정한 정보를 일컫는데, 특히, Rules는 비즈니스 모델의 구성 형태를 결정하게 된다. IPMP 시스템은 관련 rule을 적용함으로써 켄텐츠에 접근하고 사용하는 과정을 제어하게 된다. 이러한 IPMP 시스템의 일례로써 CAS(Conditional Access System)를 들 수 있다.

보호된 컨텐츠(Protected contents)는 다음의 요소로 구성된다.

- content set, 여러 종류의 media로 구성됨.
- IPMP system set, 다수의 IPMP 시스템으로 구성됨.
- rules set, 주어진 IPMP 시스템에서 적용됨.

OPIMA는 일련의 IPMP 시스템으로 정의되는 고유의 관리 구조를 갖는 소유권역들(proprietary domains)의 사회가 존재함을 인정한다. 일례가 CAS 시스템이다. OPIMA는 그러한 시스템 요소간의 상호작용을 가능하게 하도록 사용될 수 있으며, 또한, 소유권역간의 가교(bridge)로서 사용될 수도 있다.

OPIMA 신용장은 보호된 컨텐츠의 전달에 앞서서 교환될 수 있다. 전달 기저조직이 온라인 접속을 지원하지 않는 경우(저장매체, 방송매체 등)에는 OPIMA 신용장과 IPMP 시스템이 보호된 컨텐츠와 병합된 형태로 사용될 수 있다. OPIMA 신용장은 보호된 컨텐츠가 compartment 간에 교환될 수 있도록 하기위해 필요한 정보를 담고있다. 두개의 compartment(방송용 제한 수신 compartment와 인터넷 음악 배달 compartment)에서의 소비를 목적으로 하는 보호된 컨텐츠의 경우에 있어서는 양쪽 compartment의 OPIMA신용장 모두가 보호된 컨텐츠와 연관되어 있다.OPIMA는 OPIMA 신용장(Credentials)이라 불리는 인증된 식별자를 교환하는 수단을 제공하는데, 이는 compartment 내에서나 또는 compartment 간에 있어서 보호된 컨텐츠의 교환을 가능하도록 한다. OPIMA peer가 온라인으로 접속되어 있는 경우, OPIMA 신용장은 보호된 컨텐츠의 전달에 앞서서 교환될 수 있다. 전달 기저조직이 온라인 접속을 지원하지 않는 경우(저장매체, 방송매체 등)에는 OPIMA 신용장과 IPMP 시스템이 보호된 컨텐츠와 병합된 형태로 사용될 수 있다. OPIMA 신용장은 보호된 컨텐츠가 compartment 간에 교환될 수 있도록 하기위해 필요한 정보를 담고있다. 두개의 compartment(방송용 제한 수신 compartment와 인터넷 음악 배달 compartment)에서의 소비를 목적으로 하는 보호된 컨텐츠의 경우에 있어서는 양쪽 compartment의 OPIMA신용장 모두가 보호된 컨텐츠와 연관되어 있다.

OPIMA는 서로 다른 응용, 장비 및 IPMP 시스템간의 일반적인 상호작용을 가능하게 한다. Compartment란 IPMP 인터페이스나 구성요소에 있어서 부분적으로 공통 요소를 갖는 OPIMA 장비들의 집단을 말한다. 예로서 DVB를 하나의 compartment로 간주할 수 있으며, 이 DVB는 다시 특정 IPMP 시스템으로 정의되는 다른 compartment들로 나눌 수 있다. 컨텐츠가 반드시 모든 compartment들간에 교환될 필요는 없으나,

OPIMA는 compartment간의 컨텐츠 교환을 용이하도록 하는 방법을 제공한다. Compartment는 종속적일 수 있다. 즉, 하나의 compartment는 sub-compartment들을 포함한다. OPIMA는 비록 다른 수단을 사용하기는 하지만, compartment 내부에서의 상호작용도 제공한다.

OPIMA 구조는 peer-to-peer이다. OPIMA의 핵심 요소는 OPIMA Virtual Machine(OVM)으로 불리는 peer이다. OPIMA는 이러한 요소들간에 안전한(secure) 상호작용을 가능하게 하는 프로토콜 및 기저조직을 제공한다. Peer-to-peer 상호작용을 근간으로 하는 접근방식은 전형적인 client-server 구성의 효과적 구현을 사용 가능하게 한다.

OPIMA 프로토콜은 SAC(Secure Authenticated Channel)를 설정하고, IPMP 시스템을 OVM으로 다운로드 하도록 하는 프로토콜이다. 개별적 저작 관련 프로토콜은 compartment내의 peer들간에 동일한 목적으로 사용될 수 있다. 이 경우에 OPIMA 프로토콜도 역시 사용될 수 있다. 서로 다른 compartment들 간의 IPMP 시스템의 다운로드 시에는 반드시 OPIMA 프로토콜이 사용되어야 한다.

OVM간의 보안 상호작용들이 종합적으로 OPIMA 환경을 구성한다. 그러나, OVM은 연속적으로 OPIMA환경과 접속할 필요는 없다. 즉, OVM은 접속 또는 비접속 방식으로 각각 동작 가능하다.

2.1 OPIMA 환경에서의 신용장 메커니즘

OPIMA는 compartment내 또는 다른 compartment들 사이에 보호된 컨텐츠를 교환하도록 하기 위해 신용장을 이용한다. 이러한 신용장이 인증하는 내용은 다음과 같다.

- OVMs, Applications 및 IPMP 시스템
- Compartment의 증명, 즉, 암호화의 지원, 워터마킹 및 시스템 갱신여부 등과 같은 compartment의 기능성
- 신뢰관계(trusted relationship)를 설립하고자 하는 상대 peer가 요청하는 기능성을 포함하는 holder의 보안 방식(security policies)

2.2 필수 신용 기관

OPIMA 규격은 식별자, 갱신가능 보안, 취소 및 호환 등을 관리할 수 있는 기관의 필요에 대해 규정한다. 이들은 사적 또는 공적인 제 3의 중립 기관들에 의해 관리되어야 하며, OPIMA가 이런 기저조직 요소를 관리하지는 않는다. 이러한 기관들의 구조, 강령 및 그들 간의 상호작용의 형태는 상세히 언급되어야 하나 이는 이 규격서의 범위를 넘어서는 것이다. 특정 compartment의 신뢰성 및 compartment 들간 컨텐츠 교환의 신뢰성은 신용장을 교환하고 이들을 OPIMA 요소가 안전하게 해석 함으로써 가능해 진다. 이러한 기관들은 신용장을 발급하기 전에 회계 감사를 수행하기도 한다. 신용 기관의 현실적 수는 특정 기관이 몇 개의 인증 기능을 수행하는가와 연관되어 결정되어야 한다.

필수 신용 기관으로는 다음과 같은 것들이 있다.

- Compartment ID 발급 기관 : 각 compartment는 고유 번호를 부여 받는데 이 기관이 compartment ID를 부여하는 책임을 진다.
- Credential 발급 기관 : OPIMA 신용장은 중립적인 OPIMA 신용장 발급 기관이 발급하며, OVM과 OVM이 탑재되어 있는 장비의 구현 특성 및 성능을 인증한다.
- IPMP 시스템 ID 발급 기관 : IPMP 시스템은 요청, 다운로드 및 인증되기 위해 고유 번호를 부

여 받는다. 회사나 기관이 발급 기관이 되며, 자신에게 할당된 번호 중에서 IPMP 시스템 ID를 발급한다.

- IPMP Peer ID 발급 기관 : OPIMA Peer는 그들 간의 안전한 상호작용을 위해 고유번호를 부여 받는다. 회사나 기관이 발급 기관이 되며, 자신에게 할당된 번호 중에서 OPIMA Peer ID를 발급한다.

- 암호화, 서명 및 워터마킹 ID 발급 기관 : IPMP 시스템과 OVM이 그들 간에 통신하기 위해서 암호화, 서명, 워터마킹 알고리즘은 고유번호를 부여 받는다. 공개적으로 사용되는 알고리즘에 대해서만 발급한다.

2.3 Back-end Infrastructure

OPIMA 장비내의 기능적 요소들 외에, back-end 처리 시스템과의 인터페이스가 존재한다. 여기에는 회계, 권리 및 사용권 교환국 등이 포함된다. 이러한 요소들의 구조는 개별적 저작권이다. 그러나, 그들이 OPIMA 환경에 인터페이스할 때에는 OVM peer를 사용하여야 한다.

2.4 프로토콜

OPIMA에서 언급되는 문제들 중에서 중요한 부분은 IPMP 시스템을 안전하게 다운로드하는 것이다. 이때, peer는 클라이언트(Peer C) 및 서버(Peer S)의 역할을 수행한다. 초기단계에서는 SAC를 설정하기 위한 인증 받지않은 정보가 교환된다. 이 초기 교환 과정에서는 비OPIMA 프로토콜이 사용된다. 그런 후, OVM이 OPIMA peer 관계를 설정한다(그림1참조).

예로서, 클라이언트 Peer 서버 Peer 설정과정에서 OPIMA peers는 다음과 같은 방식으로 통신한다.

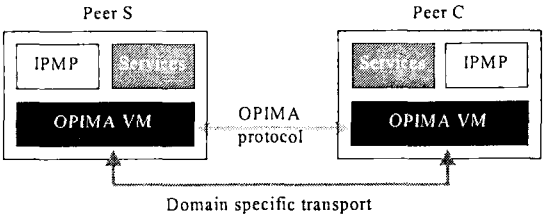

그림 1. OPIMA 프로토콜

1. 응용이 OVM에게 보호된 컨텐츠에 접속하도록 요청한다.

2. OVM이 OS가 초기 망 접속을 설정하도록 요청한다.

3. 이 접속 상부에 OPIMA SAC가 설정된다.

4. 필요한 IPMP 시스템이 OVM에 의해 요청되고 다운로드된다.

3. OPIMA Architecture(Normative)

3.1 OPIMA 프로토콜

OPIMA 프로토콜은 OPIMA peer 사이의 상호작용(interoperability)을 제공하며 다음의 두 계층 (layer)로 나누어진다.

- 제 1 계층 : 안전수송(secure transport)계층이라 부르며 협상을 통해SAC를 구현한다.

- 제 2 계층 : OPIMA 공통 메시지 프로토콜(common message protocol) 이라 부르며 SAC 계층(제 1 계층)에 의해 제공된 기능(functionality)을 사용하며, 이 프로토콜상에서 교환되는 메시지는 제 2 계층으로 하여금 IPMP 시스템의 다운로딩을 제공하도록 허용한다.

3.1.1 제 1 계층: Secure Authenticated Channel

이 규격은 compartment 간 통신을 위한 기준 SAC 프로토콜로서 SSL을 사용한다. OPIMA의 향후 버전은 다른 환경(e.g., 방송, low footprint decices) 에서 또한 사용가능한 SAC 프로토콜을

위한 지원을 제공할 수 있다. SAC 확립 단계(establishment phase)는 OVM에 의해 사용되는 암복호화 미케니즘(e.g., DES와 같은 암호화 방식을 선정, chaining 모드 선정)을 설정하여 두 대응 통신 peer간에 IPMP 시스템을 안전하게 전송하도록 한다. SAC 확립 단계에서는 또한 앞에서 언급한 암복호화에서 사용할 키를 결정하게 된다.

3.1.2 제 2 계층: OPIMA Common Message Protocol

OPIMA 공통 메시지 프로토콜은 다른 compartment 사이의 통신을 확립하기 위해 사용한다. 이 메시지는 명령과 옵션 파라미터로 구성된다. 파라미터가 아래 테이블에 나열되어 있다. 파라미터에 필요한 비트수가 괄호안에 주어진다.

아래 각 테이블은 송신자로부터 수신자 순으로 보내는 메시지를 포함한다. 길이필드(length field)는 길이필드 바로 뒤에 나오는 필드의 바이트 수를 나타낸다.

● Open IPMP System download Message

Sender	Recipient	Content
Peer 1	Peer 2	Open, length(8), IPMP ID(var)

● IPMP System code Messages

Sender	Recipient	Content
Peer 2	Peer 1	MSGDTA(8), length(8), message(var)

Push인 경우에 peer 1이 컨텐츠를 송신하고 peer 2가 그것을 수신한다

● Close of the OPIMA download protocol

Sender	Recipient	Content
Peer 1	Peer 2	Close
Peer 2	Peer 1	Close

IPMP 시스템을 송신하는 측과 수신하는 측 모두 이 메시지를 발송하는 것이 가능하다.

● **Message Ids**

Message Name	Value (binary)
MSGDTA	00000010
OPEN	00000001
CLOSE	00000011

3.2 신용장 형식

OPIMA 규격은 신용장(crdentials)를 위한 규격으로 X.509를 사용하는 것을 기본으로 한다. Compartment와 OPIMA peer의 식별자가 이 신용장에 포함된다. OPIMA 신용장은 X.509 v.3 포멧을 따르며 subject는 CompartmentID와 PeerID 두 필드로 구성되는 구조로 이루어 진다. 그러므로 OPIMA 신용장은 다음의 관련 필드를 갖는 X.509 v.3 형식의 신용장이다:

- Issuer: IssuerID
- Subject: SubjectID

IssuerID의 형식은 OCTET STRING(2)이며 인증 기관 (Certification authority)의 ID를 포함한다. SubjectID의 형식은 다음과 같다:

SEQUENCE{CompartmentID OCTET STRING (4), PeerID OCTET STRING(16)}.

3.3 OPIMA Peer

여기에서는 OPIMA Peer를 기능단위별로 설명하기로 한다. 그림2는 각 요소들간의 관계를 보여주는 그림이다.

3.3.1 OPIMA Virtual Machine

OPIMA peer는 신용의 뼈대를 구현하는 기본적인 기능 요소들의 집단을 가지며, 이를 OVM이라고 부른다. OVM의 기본 기능성은 응용에 따라

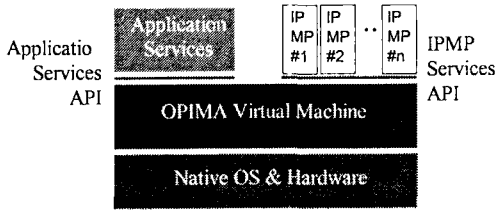

그림 2. OPIMA Peer

확장되어진다. OVM은 OPIMA 호환 기종들간에 SAC를 설정하는 역할을 담당한다.

3.3.2 IPMP

IPMP는 OPIMA peer 내부에서 지적 재산권의 관리 및 보호 시스템(IPMP-S)을 구현하는 프로세스이다.

3.3.3 응용 서비스 API

응용 서비스 API는 서비스들이 OVM및 OVM에 설치되어 있는 IPMP 시스템과 통신하도록 해주는 API이며 다음과 같은 기능을 갖는다.

- 컨텐츠에의 접근을 요청한다.
- IPMP 요소를 OVM에 설치하도록 요청한다.
- 설치된 IPMP 시스템 요소에 대해 질문한다.
- IPMP 시스템 요소에 메시지를 전송한다.

응용 서비스 API에서 정의된 각 함수는 다음과 같다.

useContent

특정한 목적으로 서비스가 OVM을 통해 컨텐츠를 사용할 수 있도록 요청하는 함수이다. 컨텐츠를 관리하고 보호하는 IPMP 시스템이 이 요청이 받아들여질 수 있는지 결정한다.

getIpmpSystem

서비스가 임의의 IPMP 시스템을 OVM에 설치하도록 요청하는 함수이다. 만약 IPMP 시스템이 이미 OVM에 설치되어있다면, 그 응용은 단순히 IPMP 시스템의 클라이언트로 등록된다. OVM이 자신의 compartment를 알고있기 때문에 응용이 compartment 식별자를 전달하지는 않는다.

queryOVM

서비스가 OVM에 대해 질문 하는 함수이다. OVM은 특정한 컨텐츠에 적용할 수 있는 IPMP 시스템의 목록과 OVM 내에 이미 설치되어 있는 IPMP 시스템이 어떤 것들인지에 대한 정보를 제공한다.

sendMessageToIpmp

서비스가 OVM에 설치되어 있는 IPMP 시스템에 메시지를 전송하고 응답을 수신 받는 함수이다.

notifyEvent

이 비동기식 응답은 OVM이 응용에게 특정 이벤트가 발생했음을 알려주기 위해 제공한다. 동기식 목적으로도 사용할 수 있다.

3.3.4 IPMP 서비스 API

IPMP 서비스 API는 IPMP와 OVM사이의 인터페이스 기능을 한다. IPMP 서비스 API는 한쪽의 IPMP 시스템과 다른 쪽의 OVM 사이의 유일한 통신방법이다. 기능별로 메소드를 그룹화하면 다음처럼 분류된다.

3.3.4.1 User Interface methods

이 함수의 목적은 사용자로부터 직접 정보(information)을 얻는 것이다. 이것은 부인방지(non-repudiation)를 위해 사용할 수 있다.

sendMessageToUser

사용자에게 메시지를 보내는 함수로 입력 파라미터로 사용자에게 보낼 문자열인 MessageText와 사용자의 응답을 IPMP에 전달하는 콜백함수, Listener가 있다.

receiveMessageFromUser

사용자로부터 메시지를 받는 명령어로 입력 파라미터로 이것이 응답임을 나타내는 식별자 SessionID와 사용자에 의해 반환되는 문자열인

Response가 있다.

3.3.4.2 Secure Storage Interface

Secure Storage Interface 기능은 OVM에 의해 제공된 저장 능력(storage capabilities)를 IPMP 시스템이 사용하도록 허용한다. IPMP 시스템 request시 저장된 정보는 동일한 IPMP 시스템이 다른 경우에 사용될 수 있어야 한다. OVM은 한 IPMP 시스템이 다른 IPMP 시스템에 의해 저장된 정보를 접근하지 못하도록 한다. IPMP 시스템 구현시 때로 메모리가 귀중한 자원(scarce resource)이 될 수 있기 때문에 저장된 정보가 무한히 지속된다고 보장할 수는 없다.

secureStoreData

이 함수는 OVM에게 바이트 어레이(array of bytes)에 포함된 정보를 저장하도록 요청한다. IPMP 시스템이 종료되었을 경우에도 OVM에 저장된 정보가 유지되도록 결정할 수 있다.

SecureRetrieveData

이 함수는 호출 IPMP(calling IPMP) 시스템에 의한 데이터 참조 (*DataReference*) 하에서 이전에 저장된 정보를 OVM에게 얻어 내도록 요청한다.

secureDeleteData

이 함수는 호출 IPMP(calling IPMP) 시스템에 의한 데이터 참조 (*DataReference*) 하에서 이전에 저장된 정보를 OVM에게 지우도록 요청한다.

3.3.4.3 Encryption and Decryption Engines

IPMP 시스템은 다음 함수들을 통해 표준 암호 알고리즘을 접근할 수 있다.

queryEncrptionAlgorithms

이 함수는 어떤 암호 알고리즘들이 있는지를 질의한다. 반환 변수로는 알고리즘 리스트가 된다.

encrypt

이 함수는 바이트의 셋(set of bytes)을 암호화한다. 이것은 주로 관리 메시지 (management message)를 암호화하기 위해 사용된다.

initEncrytion

이 함수는 컨텐츠 암호화를 초기화한다. 이 함수의 실행시 *clearContentId* 파라미터를 기반으로 컨텐츠에 접근할 수 있다. 반환변수는 암호화된 컨텐츠 또는 에러 코드 (if negative) 로의 참조 (reference) 이다.

updateEncrptionKeys

이 함수는 새로운 키 또는 변경된 키를 사용하여 컨텐츠 암호화를 시작 또는 계속하도록 한다. 키 변경을 위해 키 배열의 사용이 허용된다.

stopEncryption

이 함수는 암호화를 중지시킨다. 입력 파라미터로 컨텐츠 스트림에 대한 참조 즉 *ClearContnetId*가 사용된다.

decrypt

이 함수는 바이트의 셋을 복호화한다. 이것은 주로 관리 메시지를 복호화하기 위해 사용된다.

initDecryption

이 함수는 컨텐츠 복호화를 초기화한다. 이 함수의 실행시 *EncryptedContentId* 파라미터를 기반으로 컨텐츠에 접근할 수 있다. 반환변수는 복호화된 컨텐츠 또는 에러 코드 (if negative) 로의 참조(reference) 이다.

updateDecrptionKeys

이 함수는 새로운 키 또는 변경된 키를 사용하여 컨텐츠 복호화를 시작 또는 계속하도록 한다. 키 변경을 위해 키 배열의 사용이 허용된다.

stopDecryption

이 함수는 복호화를 중지시킨다. 입력 파라미터로 컨텐츠 스트림에 대한 참조 즉 *EncryptedContnetId*가 사용된다.

3.3.4.4 Signature Engines

이 인터페이스는 전자서명 처리를 수행하기위

해 공개키 암호화를 사용하는 IPMP 시스템에 의해 사용된다.

querySignature Algorithms

이 함수는 어떤 서명 알고리즘들이 있는지를 질의한다. 반환 변수로는 지원가능한 서명 알고리즘 리스트가 된다. 등록기관(Registration Authority)은 확정된 알고리즘을 위한 표준 식별자를 발행한다.

verifySignature

이 함수는 전자서명(digital signature)을 증명한다. 이것은 주로 관리 메시지의 진위성(authenticity)를 증명하는데 사용된다.

verifyContentSignature

이 함수는 전자서명(digital signature)을 증명한다. 이것은 주로 컨텐츠의 진위성(authenticity)을 증명하는데 사용된다. 이 함수의 실행시 *contentId* 파라미터를 기반으로 컨텐츠에 접근할 수 있다.

generateSignature

이 함수는 전자서명(digital signature)을 발생한다. 이것은 주로 관리 메시지에 서명하기 위해 사용된다.

generateContentSignature

이 함수는 전자서명(digital signature)을 발생한다. 이것은 주로 컨텐츠에 서명하기 위해 사용된다. 이 함수의 실행시, *contenteId* 파라미터를 기반으로 컨텐츠에 접근할 수 있다.

3.3.4.5 Watermarking Engines

OVM이 워터마킹 엔진으로의 접근을 제공하는 경우 다음 인터페이스들이 사용된다.

queryWatermarkAlgorithms

이 함수는 지원가능한 워터마킹 알고리즘이 어떤 것들이 있는지 질의한다. 등록기관은 확정된 알고리즘에 대한 표준 식별자를 발행한다.

extractWatermark

이 함수는 *contentId*에 의해 식별된 컨텐츠로부터 워터마크 추출을 시작한다. 컨텐츠가 스트림(stream) 이면 stopWatermarkExtraction 함수를 호출함으로써 워터마크 추출을 중지시킬 수 있다. 콜백 인터페이스가 워터마크 전달을 위해 사용된다.

stopWatermarkExtraction

이 함수는 *contentId* 에 의해 식별된 컨텐츠 스트림으로부터의 워터마크 검출을 중지시킨다.

newWatermark

이 함수는 워터마크 엔진을 위한 콜백 인터페이스이다. 이것은 워터마크 엔진을 호출하는 IPMP 컴포넌트에 의해 실행되어야 한다.

insertWatermark

이 함수는 *sourceContentId* 에 의해 식별된 컨텐츠로 워터마크 삽입을 시작한다. 반환변수는 *sinkContentId*이다. 컨텐츠가 스트림이면 stopWatermarkInsertion 함수를 호출함으로써 워터마크 삽입을 중단한다.

stopWatermarkInsertion

이 함수는 *contentId* 에 의해 식별된 컨텐츠 스트림으로부터의 워터마크 삽입을 중지시킨다.

3.3.4.6 Smart Cards

로컬 스마트 카드 리더(reader)를 통해 접근할 수 있는 스마트 카드는 이 API 함수들을 통해 접근할 수 있다. 다음 두 구조는 ISO 7816 포멧의 메시지를 나타낸다.

```
struct CommandAPDU
{
        byte cla;
        byte ins;
        byte P1;
        byte P2;
```

```
    int lc;  //The length of the data field of
  the APDU.
    int le; //The expected length of the
  ResponseAPDU.
    byte[] data;
}

struct ResponseAPDU
{
    byte sw1
    byte sw2
    byte[] data
}
```

addCTListener

이 함수는 카드 단말기로부터 카드의 삽입 또는 제거의 표시를 받기위해 Card Terminal listener을 추가한다.

removeCTListener

이 함수는 카드 단말기로부터 더 이상 Card TerminalEvents를 받지않도록 CTListener을 제거한다.

getSlotId

이 함수는 카드 단말기에 속하는 슬롯의 리스트를 반환한다.

isCardPresent

이 함수는 특정 슬롯에 스마트 카드가 있는지를 체크한다.

openSlotChannel

이 함수는 슬롯번호 slotID상의 SlotChannel을 오픈한다.

closeSlotChannel

이 함수는 slotSessionId에 의해 식별된 Slot Channel을 클로즈한다.

getATR

이 함수는 slotID의 슬롯에 삽입된 카드의 ATR(Answer-TO-Reset)응답을 반환한다.

reset

이 함수는 슬롯에 삽입된 스마트 카드를 리셋한다.

sendAPDU

이 함수는 스마트 카드로 명령을 보내고 응답을 기다린다.

다음 콜백 함수가 IPMP 시스템에의해 수행될 수 있다:

cardInserted

이 함수는 스마트 카드가 slotID의 슬롯에 삽입되었음을 통지한다.

CardRemoved

이 함수는 스마트 카드가 slotID의 슬롯으로부터 제거되었음을 통지한다.

3.3.4.7 Abstract Access to Content

installCallbackContentAccess

이 함수는 이 함수 셋에 대한 콜백 함수를 인스톨하도록 IPMP 시스템으로부터 OVM에게 요청한다.

abstractContentAccess

이 함수는 이전 인스톨된 콜백 함수인데 이것을 통해 OVM은 특정 목적을 위해IPMP 시스템을 호출한다. 보통 이 함수는 어플리케이션으로부터 요청시 OVM에 의해 호출되어야 하고, 따라서 어플리케에션으로부터 IPMP 시스템의 간접 호출을 수행하게 된다.

replyToContentAccess

이 함수는 IPMP 시스템으로부터의 *abstrat ContentAccess*함수에 대한 응답을 수행한다. 이것은 OVM에 의해 *notityEvent* 콜백 함수로 전달되는데 이 콜백 함수는 어플리케이션 서비스 API의 일부이다.

3.3.4.8 abstract Access to Rules

처리되어질 컨텐츠가 IPMP 시스템으로 직접 접근되어지지 않을 경우 이 함수들을 사용해서 규칙(rules)을 접근할 수 있다.

obtainUserRules

이 함수는 사용자와 관련된 규칙을 얻어낸다.

obtainContentRules

이 함수는 컨텐츠와 관련된 규칙을 얻어낸다.

newRules

이 함수는 IPMP 시스템에 의해 수행된 콜백 함수이다. 이 함수의 사용처는 *obtainContentRules* 또는 *obtainUserRules*에 대한 listener이다. *SessionId*가 응답에 대한 요청과 관련하여 사용되어진다.

updateContentRules

이 함수는 컨텐츠와 관련된 규칙을 갱신한다.

3.3.4.9 Abstract Access to OPIMA Peers

이 인터페이스의 목적은 OPIMA peer 사이에 IPMP 시스템 독점적 관리 및 제어 메시지 (proprietary management and control message)를 교환하는 것이다. 이 인터페이스는 로컬 IPMP 시스템 사이의 통신을 위해 사용될 수 있다.

openConnection

이 함수는 OPIMA peer로의 연결을 오픈한다.

closeConnection

이 함수는 OPIMA peer로의 연결을 종료한다.

addConnectionListener

이 함수는 다른 OPIMA peer가 설정을 시도하는 새로운 연결에 대한 listener를 인스톨한다.

sendMessage

이 함수는 지정된 연결(designated connection)에 대한 메시지를 전송한다.

다음 콜백 함수가 IPMP 시스템에 의해 수행될 수 있다.

newConnection

이 함수는 새로운 연결의 설정을 알린다.

receiveMessageFromPeer

이 함수는 지정된 연결에 대한 메시지를 수신한다.

3.3.4.10 Abstract Access to Applications

IPMP 시스템이 이전에 이 함수를 통한 적절한 콜백을 인스톨했으면 어플리케이션은 로컬 IPMP으로 접근할 수 있다.

installCallbackApplication

이 함수는 IPMP 시스템이 어플리케이션에 대한 콜백 함수를 인스톨하도록 OVM에 요청하는 함수이다.

replyMessage

*receiveMessageFromApplication*을 통해서 받은 메시지에 대해 IPMP 시스템은 이 함수를 사용하여 응답을 전송한다.

다음 콜백 함수가 IPMP 시스템에 의해 수행된다

replyMessageFromApplication

이 콜백 함수에 의해 어플리케이션이 OVM을 통해 IPMP 시스템을 접근하게 된다. 이 함수를 통해 IPMP 시스템을 호출하는 쪽은 OVM이다.

3.3.4.11 Life-cycle Control

이 인터페이스는 IPMP 시스템의 실행상태(execution state)를 제어하기 위해 사용된다. 다음 함수들이 life cycle 제어 목적으로 IPMP 시스템에 의해 수행되어 진다:

initialize

이 함수는 IPMP 시스템 인스톨후 즉시 호출된다. 이것에 의해 IPMP 시스템이 어떤 필요한 인스톨 절차를 감시하도록 허용된다.

terminate

이 함수는 IPMP 시스템 제거 전에 호출된다.

이것에 의해 IPMP시스템이 어떤 필요한 종료 절차를 감시하도록 허용된다. 이 함수가 반환된후, IPMP 시스템이 OVM으로부터 제거된다.

다음 함수들이 life cycle 제어 목적을 위해 OVM에 의해 수행되어 진다.

remove

이 함수는 요청시 IPMP 시스템을 제거한다. 이것에 의해 IPMP 시스템이 어떤 필요한 인스톨 절차를 감시하도록 허용된다.

update

이 함수는 현재IPMP 시스템을 종료하고 새로운 IPMP 시스템의 다운로딩을 요청한다.

3.3.4.12 Locale interface

이 함수는 OVM상에서 수행되고 있는 IPMP 시스템 컴포넌트로 하여금 그것이 안전한 상태에서 수행중임을 알게한다.

getTime

이 함수는 the Epoch (Coordinated Universal Time-CTC, 1970년 1월 1일 0시 0분 0초) 이후의 the number of seconds 또는 에러 번호를 반환한다.

getCountry

이 함수는 OVM이 배치된 나라를 나타내는 대문자 ISO 31662 문자 코드 또는 빈 문자열을 반환한다.

getLanguage

이 함수는 OVM에 대한 언어 코드 또는 빈 문자열을 반환한다. 언어 문자열은 소문자 ISI 639 코드가 될 것이다.

4. 결 론

본 논문에서는 IPMP 기술에 대해 OPIMA를 근간으로 조사 및 분석하였다. OPIMA는 IPMP 시스템을 안전하게 다운로드, 설치 및 작동시킬 수 있는 환경에 대한 규격을 제공하며 OPIMA 호환 기종을 구현하기 위해 필요한 기능들에 대해 정의하고 있다.

OPIMA를 주목할 필요가 있는 이유는 OPIMA가 MPEG-21과 함께 차세대 저작권 관리 및 보호의 표준이 될 가능성이 많기 때문이다. 현재 유럽연합에서는 ACTS[4] 프로젝트의 하나인 OCCAMM [5]이라는 프로젝트를 통해 OPIMA VM을 구현하고 있다.

참 고 문 헌

[1] "Overview of the MPEG-4 Standard," ISO/IEC JTC1/SC29/WG11 N4030, March 2001

[2] 김형중, "디지털 방송과 컨텐츠 복제방지 표준활동," 방송공학회지, 제5권 제3호, 2000년 9월, pp. 14-17

[3] OPIMA Specification, Version1.1, June, 2000. http://www.cselt.it/ufv/leonrdo/opima

[4] http://www.servicemachine.org/prores/proresep.htm, Evaluation of ACTS Projects results

[5] OCCAMM(Open Components for Controlled Access to Multimedia Material) http://sharon.cselt.it/projects/occamm/

최 재 각

- 1984년 경북대학교 전자공학과 졸업(학사)
- 1987년 한국과학기술원 전기및전자공학과(석사)
- 1997년 한국과학기술원 전기및전자공학과(박사)
- 1987년~1998년 한국전자통신연구원 선임연구원
- 1998년~현재 경일대학교 제어계측공학과 조교수
- 관심분야 : 영상 및 멀티미디어 통신, 멀티미디어저작권보호

이 종 극

- 1978년 2월 경북대학교 전자공학과(공학사)
- 1988년 2월 미국 North Carolina St. University(공학석사)
- 1988년 6월 Assistant Teaching
- 1993년 6월 미국 Texas A&M University(공학박사)
- 1994년~현재 동의대학교 컴퓨터응용공학부 교수
- 관심분야 : 컴퓨터 네트워크, 병렬처리

김 태 석

- 1981년 경북대학교 전자공학과 졸업(공학사)
- 1989년 일본 KEIO대학 이공학부 계산기과학전공(공학석사)
- 1993년 일본 KEIO대학 이공학부 계산기과학전공(공학박사)
- 1993년 일본 국제전신전화연구소(KDD) 기술고문
- 1993년 일본 KEIO대학 이공학부 객원연구원
- 1994년~현재 동의대학교 컴퓨터응용공학부 교수
- 자격증 : 멀티미디어기술사, 인터넷시스템관리사(기술사)
- 저서 : 인터넷비즈니스, 자연언어처리, 자연언어이해 등 다수
- 관심분야 : 정보시스템, 기계번역, 인터넷비즈니스

구 본 호

- 1980년 경북대학교 전자공학과(학사)
- 1985년 경북대학교 전자공학과(석사)
- 1991년 경북대학교 전자공학과(박사)
- 1987년~1998년 한국전자통신연구원 선임연구원
- 1991년~현재 경일대학교 제어계측공학과 부교수
- 관심분야 : 컴퓨터제어, 전력전자, 센서공학, 전동기제어