MOSES에서의 JPWallet의 기능과 키 관리 분석

학생회원 오 태 석*, 정회원 김 용 철*, 최 범 석**, 최 진 수**

Key Management Process in JPWallet of MOSES System

Tae Suk Oh* ***Student Member***,
Yong Cheol Kim*, Bum Suk Choi**, Jin Soo Choi** ***Regular Members***

요 약

DRM(Digital Rights Management) 시스템들이 특정 플랫폼이나 암호화 기법에 기반을 두는 경우에는 이기종간 호환성이 문제가 될 수 있다. MOSES(MPEG Open Security for Embedded Systems)는 이러한 문제점을 고려하여 기존 표준들과의 상호 연동 및 IPMP 기능 적용을 목적으로 개발된 시스템이다. MOSES는 콘텐츠 유통의 보안을 위해 JPWallet을 사용하여 키 관리를 통한 라이센스 발급을 하고 있다. 본 논문에서는 클라이언트 JPWallet의 구조 분석 결과를 제시하고, 구체적으로는 서버와 클라이언트 사이에서의 암호화 및 키 관리 방법을 서술하고 이의 실험용 테스트베드를 소개한다. MOSES의 분석은 유럽의 IPMP 표준과 호환성이 있는 국내 고유의 IPMP 표준 설정에 도움이 될 것이다.

Key Words : MOSES, OPIMA, DRM, OpenSDRM, JPWallet, 전자지갑

ABSTRACT

When DRM systems are built on a specific computing platform and a coding algorithm, the interoperability among them will be improbable. For enhanced compatibility, MOSES has been developed such that it has a structure that can be decomposed into independent modules for interoperability with other DRM systems with IPMP functionality. In MOSES, security in contents transaction is provided by JPWallet which controls licenses with key management. In this paper, we present the structure of JPWallet and how the keys are handled between contents servers and contents-consuming clients. The PDA-based codes from the prototype MOSES system have been ported into PC-based codes and tested for compatibility. Analysis of JPWallet, which is the core of MOSES, will contribute to the standardization of domestic IPMP systems compatible with global standards.

Ⅰ. 서 론

DRM(Digital Rights Management)은 다양한 컨텐츠를 안전하게 사용자에게 전달하여 불법적 컨텐츠 유통을 방지하는 시스템 기술이며, 이에 관련된 대표적인 연구로는 미국의 SDMI, 유럽연합의 IMPRIMATUR 등이 있다[1, 2]. DRM 시스템들은 그 특성상 특정 운영 체제의 플랫폼 혹은 특정한 암호화 기법에 맞추어져 개발되는 경우가 많아, 기존 시스템을 통합하는 DRM 시스템의 개발 및 관리가 어렵고, 새로운 DRM 솔루션과의 접목이 용이하지 않았다. 이러한 문제점의 해결을 위하여 유럽 연합의 ITA program에서 OPIMA(Open Platform Initiative for Multimedia Access)를 개발하여, 다중

※ 본 연구는 한국전자통신연구원의 2004년 위탁연구 지원에 의하여 수행되었습니다.
* 서울시립대학교 전자전기컴퓨터공학부 컴퓨터비전연구실 ({bbole, yckim}@uos.ac.kr)
** 한국전자통신연구원 방송미디어연구그룹 ({bschoi, jschoi}@etri.re.kr)
논문번호 : KICS2005-07-275, 접수일자 : 2005년 7월 27일

컨텐츠 보호 장치 하에서 사용자들이 다양한 형태의 컨텐츠나 서비스에 접근할 수 있도록 하였다[3].

MOSES(MPEG Open Security for Embedded Systems)는 OPIMA를 기반으로 하여 최근에 개발된 규격으로 DRM에서의 세부 요소기술의 설계와 기존 표준들과의 상호연동 및 다양한 플랫폼 지원, IPMP 기능 적용 등의 관점에서 개발되었다[4]. 즉, MOSES는 특정 플랫폼을 기반으로 하지 않으면서, MPEG-IPMP 표준을 이용한 디지털 TV, PC, PDA 등의 단말기, 셋탑박스나 모바일 등의 플랫폼에서 DRM에 필요한 기술 개발을 목적으로 진행되었다.

따라서 MOSES 시스템은 다음과 같은 몇 가지 특징을 갖고 있다. 첫째는 DRM 서버의 개발 과정에서 OpenSDRM(Open and Secure Digital Rights Management) 기술의 사용을 가능하게 하여, 여러 종류의 플랫폼과 호환되는 인터페이스를 제공한다. 둘째, 클라이언트에서 지불 및 인증, 라이센스 발급 등의 보안관련 부분을 별도로 처리하기 위하여 JPWallet(Java Personal Wallet)을 개발하였다.

MOSES 시스템의 파급성을 고려한다면 여타 DRM 시스템과 호환되는 콘텐츠 유통 시스템 설계에서 JPWallet과 호환되는 보안 기능의 구현이 매우 중요하나, MOSES 사업의 발표 자료에는 주로 서버의 각 콤포넌트 별 기술적 내용이 위주이며, 보안에 관련된 JPWallet의 내용은 찾아보기 어렵다.

이러한 문제를 해결하기 위하여 본 논문에서는 MOSES 시스템에서 JPWallet을 중심으로 보안 관련 부분을 분석한 결과를 제시한다. 또한 이러한 JPWallet의 콤포넌트 별 분석을 위하여 JPWallet의 바이너리 코드를 직접 역구성하여 얻은 소스코드의 분석 방법과 그 결과를 소개한다.

세부적으로는, 서버의 OpenSDRM 기술 중에서 보안 관련 부분을 소개하고, 클라이언트 측에서 JPWallet과 연동되는 세부 콤포넌트의 동작과 JPWallet에서의 키 관리에 대한 분석 결과를 서술한다. 또한 PDA 등을 대상으로 제작된 MOSES 소프트웨어를 PC 환경으로 포팅하여 시험한 결과와 이를 위한 테스트베드의 구현 방법을 제시한다.

본 논문의 구성은 다음과 같다. 제 2장에서 MOSES의 기반이 되는 OpenSDRM에 대해 서술하고 제 3장에서는 JPWallet 기능과 서버 연관 콤포넌트들을 세부 분류한다. 제 4장에서는 분류된 JPWallet 콤포넌트 간 또는 서버와의 키 관리 시스템을 분석하고 5장에서는 PC 환경의 테스트베드 구축에 대해 기술하고 6장에서 결론을 맺는다.

Ⅱ. OpenSDRM

MOSES 프로젝트에서는 OPIMA, MPEG-4와 같은 국제 표준을 따라 IPMPX 기술을 기반으로 한 OpenSDRM 콤포넌트를 개발하였다[5, 6]. MOSES의 서버는 각 기능별로 콤포넌트들로 구성되어 있는데, 개별 콤포넌트의 개발 과정에서는 이기종 플랫폼 사이의 보안을 위해 OpenSDRM 기술을 사용하였고 이를 이용하여 Music-4You 라는 데모용 음악 유통 서버와 클라이언트에서 전자지갑 프로그램에 해당하는 JPWallet을 개발하였다. 이 장에서는 MOSES 시스템의 기본 구성인 OpenSDRM 기술의 분석과 JPWallet과 연동되는 콤포넌트들을 기능별로 서술한다.

2.1 OpenSDRM 구성 요소

MOSES는 서버의 동작 형태에 따라 10개의 내부 콤포넌트와 4개의 외부 콤포넌트를 분류하여 각각의 인터페이스를 정의하고 이를 OpenSDRM 기술이라 명명하였다. 콤포넌트의 규격은 오픈-소스 기술을 기본으로 하여 PHP, Apache Web server, mod-ssl OpenSSL, mod_php, PHP NuSoap, PHP ADOdb, Linux OS 등과 같은 환경에서 구현이 가능하도록 구성된다.

콤포넌트들 간의 보안은 소켓 통신 계층 (SSL/TLS 보안통신 채널의 제공을 위하여 OpenSSL과 mod_SSL를 사용)과 어플리케이션 계층 (PHP를 이용한 개발)을 이용한다. 또한 콤포넌트의 인터페이스는 콤포넌트 확장 및 조화를 고려하여 WSDL (Web Services Description Language)로 구현되었다[7][8]. 이러한 개발 환경 하에서 어떤 플랫폼에도 탑재 가능한 언어와 DB, 통신 계층을 이용하여 특정 시스템에 제약되지 않는 콤포넌트들의 개발이 가능하고 이를 이용하는 서버를 구축할 수 있도록 하였다.

2.2 OpenSDRM 세부 콤포넌트

서버의 콤포넌트는 그림 1과 같이 내부 콤포넌트와 외부 콤포넌트로 구분된다. 내부 콤포넌트들은 콘텐츠의 준비, 라이센스 생성, 사용자 인증, IPMP 툴 관리 등과 같은 일반적으로 서버가 제공하는 기능을 담당한다. 외부 콤포넌트는 서버 외부와의 연동을 처리하는 부분으로 콘텐츠 제공, 콘텐츠 사용, IPMP 툴 제공 등을 처리한다. 외부 콤포넌트의 인터페이스는 클라이언트 입장에서 IPMP 툴 제공자

나 콘텐츠 제공자 등과의 확장성을 고려하여 개발되었다.

MOSES의 JPWallet 프로그램은 전자 지불이나 자격 인증 등의 보안 처리 기능을 담당하는 콤포넌트들이 합쳐진 것이다. JPWallet은 그림 1에서 나타나 있는 서버의 외부 콤포넌트 중에서 Payment Infrastructure, User, Certification Authority에 해당한다.

서버의 콤포넌트 간 통신에서는 내부 콤포넌트들 간의 보안 뿐 아니라 서버의 외부 콤포넌트와 내부 콤포넌트 사이에서도 완벽한 보안이 요구된다. 다음 장에서는 MOSES 시스템에서 외부 콤포넌트와 내부 콤포넌트가 통신할 때 JPWallet을 이용하는 키 관리를 통한 보안 방법을 서술한다.

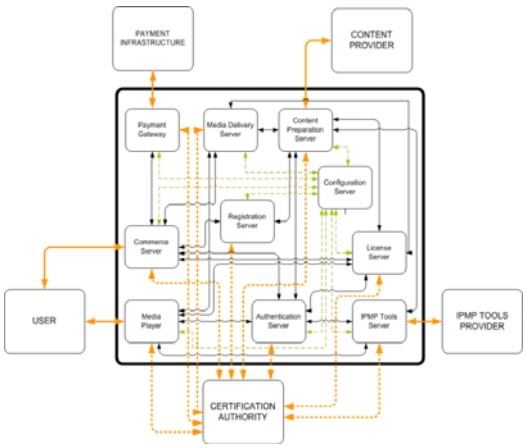

그림 1. OpenSDRM의 서버 측 콤포넌트 구성도

Ⅲ. MOSES 클라이언트 JPWallet

기존의 DRM 시스템에서는 보안에 관련된 부분을 플레이어에서 직접 처리하는 경우가 많았으나, MOSES에서는 일종의 전자지갑인 JPWallet 프로그램을 개발하여 인증, 라이센스 등과 같은 보안 기능을 담당하도록 하였다. 이렇게 보안 처리 기능만을 분리하여 플랫폼에 독립적인 구조로 개발함으로써, 다양한 플레이어나 IPMP 툴들이 플러그인 형태로 제공될 수 있는 프레임워크를 제공한다. JPWallet은 OpenSDRM 콤포넌트에서 서버와의 통신 및 보안을 담당한다.

3.1 클라이언트의 JPWallet

JPWallet은 클라이언트 측에 대한 보안 목적으로 사용자 식별 및 개인 정보와 콘텐츠, 라이센스의 관리와 저장 등에 활용된다. 그림 2는 JPWallet과 서버의 각 콤포넌트 사이에서 콘텐츠, 라이센스, IPMP 툴들의 흐름 과정을 나타낸다. JPWallet은 서버와 미디어 플레이어의 중간에서 라이센스 서버 및 인증서버와 직접적으로 정보를 주고 받아, 재생하고자 하는 콘텐츠에 대해서, 라이센스 처리, 사용자 인증과 같이 콘텐츠에 대한 접근을 제어한다.

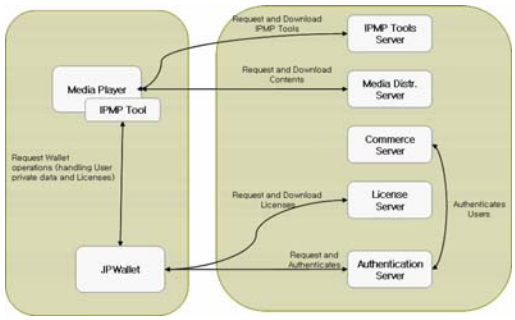

그림 2. JPWallet에서의 콘텐츠의 접근 제어를 위한 라이센스 관리와 인증 과정

3.2 JPWallet의 분석

JPWallet은 자바로 제작되었으며 바이너리 패키지로 제공된다. JPWallet 내부의 각 콤포넌트 별 동작을 파악하기 위하여는 소스 레벨의 분석이 필요하므로, JAD (Java Decomiler)을 사용하여 원본 소스로 복원하였다[9].

복원된 JPWallet 소스로부터의 분류는 소스코드에서 클래스와 함수의 처리 내용을 따랐고, 그림 3에 보인 바와 같이 콤포넌트들을 크게 인증 부분과 라이센스 부분으로 나누었다. 하부 구조로는 가입 콤포넌트, 로그인 콤포넌트, 음악 선택 콤포넌트, 음악 재생 콤포넌트 들을 분류하였다. 인증 콤포넌트와 라이센스 콤포넌트로는 서버와 직접적으로 접촉하여, 서버 내부 콤포넌트인 인증서버 (AUS) 및 라이센스서버 (LIS)와 통신한다[10].

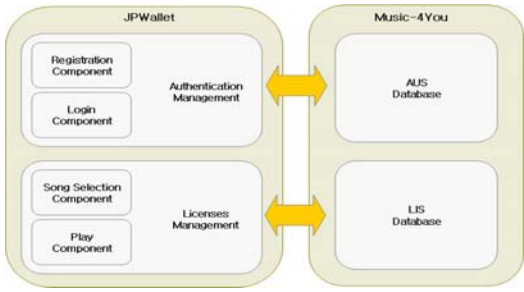

그림 3. 음악 유통 서버와의 인터페이스에서의 JPWallet의 콤포넌트별 분류

Ⅳ. 클라이언트에서의 키 관리

DRM 시스템에서 클라이언트와 서버와의 정보 교환은 보안된 상태의 유지를 위하여 키를 이용하는 암호화 방법을 사용한다. JPWallet은 이러한 보안 유지를 위하여 키들을 관리하며 이들은 크게 세 가지로 구분된다. 첫째는 사용자 등록 과정에서의 키 생성 및 전송을 위한 디지털 서명에 필요한 키의 관리이다. 둘째는 서버에서 암호화한 콘텐츠를 클라이언트에서 복호화 할 때 사용하는 키의 관리이며, 셋째는 콘텐츠의 접근을 제어하는 라이센스와 관련된 키의 관리이다. 이 장에서는 각각의 키 관리에 대한 소스 코드 분석 결과를 소개한다.

4.1 인증 키 관리

JPWallet에서 사용자 등록 시에 사용자의 가입 정보는 전통적인 디지털 서명을 사용하여 서버로 전송된다. 디지털 서명은 JPWallet 상에서 사용자 가입 정보를 MD5로 메시지 축약함으로써 이루어지며, 비공개키로 암호화한 전자서명 값, 데이터, 인증서 (공개키) 등의 정보가 서버로 전달된다. MOSES의 데모용 음악 유통 서버는 웹서버로 구성되어 있어 디지털 서명의 전송은 SOAP를 이용한 XML 형태로 전송된다. SOAP는 텍스트 명령어를 XML 형태로 HTTP를 사용하여 전달하므로, 이러한 XML과 HTTP를 사용하는 대부분의 플랫폼에 서버에 접근할 수 있는 방법을 제공한다[11].

4.2 콘텐츠 암호화 키 관리

MOSES 시스템에서 제시하는 암호화 방법은 두 가지로서 특정 키로써 콘텐츠를 암호화하는 방법과 콘텐츠를 워터마킹하는 방법을 제안하였다. 실제 데모에 구현된 방법은 콘텐츠 전체를 특정 키로써 암호화 하는 방법을 사용하였고 TslParser 툴을 사용하여 암호화 및 복호화를 처리한다. 사용한 암호 알고리즘은 AES이며 암호화 키 값은 라이센스에 실려 JPWallet을 통하여 클라이언트로 전송된다.

클라이언트에서 라이센스를 요청하기 전, 콘텐츠의 메타데이터를 확인하여 필요한 라이센스에 대한 정보를 알 수 있다. 메타데이터에는 Title, Author, Data, Duration, ContentID 등이 있으며 메타데이터는 콘텐츠 생성시 콘텐츠 헤더에 암호화되어 삽입된다. 클라이언트는 Tslparser 툴을 이용하여 헤더의 메타데이터를 얻은 후에, 일종의 콘텐츠 ID에 해당하는 복호화 키의 값을 포함하는 라이센스를 서버로부터 발급받는다.

4.3 라이센스의 키 관리

콘텐츠의 재생을 위해서는 복호화 키뿐 아니라 콘텐츠의 접근 권한도 또한 만족하여야 한다. 라이센스에는 콘텐츠의 접근 권한 (재생 횟수, 사용 기간 등)까지 포함되어 있기 때문이다. XML로 구성된 라이센스에는 콘텐츠 암호화 키(AES), 콘텐츠 ID, 재생 횟수, 유효날짜 등의 정보가 포함되어 있으며, 라이센스는 REL의 형식에 따라 ODRL로 표기된다[12][13].

4.4 콘텐츠 라이센스의 발급 과정

그림 4는 사용자 인증을 거친 후 라이센스를 발급 받아 콘텐츠의 복호화 키 값을 획득하는 일련의 과정을 보여준다. JPWallet에서 사용자가 로그인 하면 DES 키 값을 받아 단말기에 저장된 로컬 데이터베이스에 등록된 인증된 사용자인가를 확인한 후 로컬 데이터베이스의 접근 권한을 부여한다. 이때 사용하는 로컬 데이터베이스는 사용자 가입 시 생성된 것으로서 디지털 서명이 첨부된 사용자 정보가 저장되어 있으며, 클라이언트 상에서 로그인 할 경우 기본적 사용자 조회에 이용된다.

사용자가 다운받은 콘텐츠를 선택하면 우선 TslParser로 콘텐츠 ID를 읽어들이고 로컬 데이터베이스에 콘텐츠 ID에 대한 라이센스가 있는지 판단한다. 만약 라이센스가 존재하면 라이센스 내용에 따라 콘텐츠 제어가 이루어지고 라이센스가 없는 경우에는 서버의 라이센스 서버(LIS)에 라이센스를 요청한다. 다운받은 라이센스에는 재생하고자 하는 콘텐츠의 복호화키 뿐 아니라 접근 권한에 대한 정보가 들어있다. 이 논문에서 분석한 JPWallet은 데모 목적으로 제작된 상태이어서 DRM 시스템에서 가장 많이 사용되는 PKI 구조나 기타 결제 구조는 포함되어 있지 않은 상태였다.

Ⅴ. 테스트베드 구축

MOSES 시스템은 여러 DRM 솔루션의 호환성을 목적으로 개발되었으며, 데모용으로 PDA와 같은 모바일 환경에서 동작하는 Music-4You 음악 유통 서버와 클라이언트용 JPWallet이 개발되었다. 본 연구에서는 MOSES를 기반으로 하는 DRM 솔루션의 개발 및 콤포넌트의 기능 확장을 위하여 다양한 환경에서의 실험을 위한 PC의 윈도우 기반으로 포팅한 테스트베드 구축에 관한 내용을 서술한다.

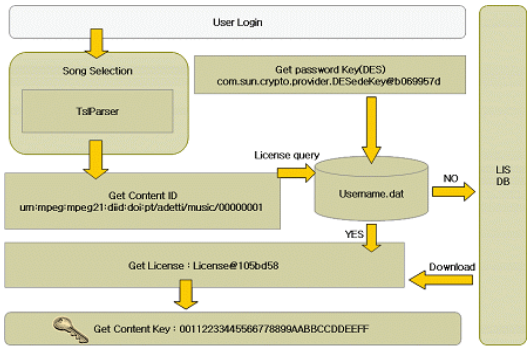

그림 4. 콘텐츠 라이센스 발급 과정

그림 5. PC기반의 테스트베드 구축 화면

4.1 기본 데모 시스템 구성 요소

MOSES에서 데모용으로 개발한 Music-4You 웹 서버와 클라이언트의 세부적인 콤포넌트의 구성은 다음과 같다. MOSES 서버에는 APACHE, PHP, MYSQL 등 웹서버 구동에 필요한 기본적인 프로그램과 라이브러리들이 설치된다. Music-4You 디렉토리에는 웹서버를 구동하는데 필요한 기본적인 소스가 PHP로 제공되고 IPMP 툴과 관련 소스들이 각각의 OpenSDRM 관련 콤포넌트로 분류되어 설치된다. 설치된 클라이언트는 사용자 등록, 로그인, 콘텐츠 재생 등을 담당하는 콤포넌트들로 구성되어 있으며, IM-1 플레이어와 JPWallet이 바이너리 설치된다.

4.2 테스트베드 제작

본 연구가 이루어진 시점에서 입수하여 분석한 MOSES 시스템은 여러 IPMP 툴 및 콤포넌트들의 확장 인터페이스의 정의에 중점을 둔 일종의 개념적 DRM 솔루션이다. 따라서 MOSES 시스템을 확장하기 위해서는 기본적인 PC 기반에서 테스트베드를 필요로 한다. 이를 위하여 PDA 환경을 기반으로 제작되어 있는 데모 시스템을 PC 환경으로 포팅하였다.

서버의 포팅 내용은 다음과 같다. 첫째, MP3 포맷과 AAC 포맷의 업로드가 동시에 가능하도록 하였다. 둘째, 콘텐츠 준비 서버(CPS)에서 콘텐츠를 암호화 할 경우 PDA 용뿐만 아니라 PC 용 MPEG-4 포맷도 지원되도록 하였다. 그림 5는 PC 환경으로 포팅되어 실행되는 서버와 클라이언트의 테스트베드 화면을 나타낸 것이다. 이 테스트베드는 JPWallet의 전자지불 구조와 OpenSDRM 서버 기술을 기반으로 하여 구축하는 MOSES 시스템에 대한 응용 목적의 IPMP 툴, 콘텐츠 암호화, 전자 지불 구조의 연구 개발에 활용될 예정이다.

Ⅵ. 결 론

이 논문에서는 기존 DRM 솔루션들간의 호환성, 이식성을 해결하고자 개발된 MOSES 데모 시스템을 분석하였다. 첫째, MOSES의 핵심 기술인 OpenSDRM 프레임의 관점에서 MOSES 서버의 전체 구조를 파악하고 서버 콤포넌트의 역할과 어플리케이션 인터페이스를 분류하고 정의하였으며, 클라이언트 JPWallet과 연동되는 서버 콤포넌트를 세부 분석하였다. 둘째, 사용자 인증과 라이센스 및 키 관리를 담당하는 JPWallet에서의 키 관리 과정을 분석하였다. 마지막으로 PC 환경과 PDA 환경 모두에서 사용 가능한 테스트베드를 구현한 결과를 소개하였다. 이러한 MOSES 시스템의 분석은 유럽의 대표적 IPMP 표준과 국내 IPMP 표준과의 호환성 보장에 공헌할 것으로 기대한다.

참 고 문 헌

[1] 문주영, "디지털 저작권 관리(DRM)의 현황," 정보통신정책연구원, 2001.

[2] Commission of The European Communities, "Commission Staff Working Paper : Digital Rights Background, Systems, Assesment," February 2002.

[3] OPIMA, "OPIMA Specification version1.1," June. 2000.

[4] MOSES D1.3: "MOSES 3. architecture requirements and specificatio," August 2002.

[5] ISO/IEC JTC1/SC29/WG11 N1714, MPEG Requirements Group, "Call for Proposals for the Identification and Protection of Content in MPEG-4," April 1997, Bristol MPEG

meeting.
[6] ISO/IEC JTC1 SC29 WG11 N2614, "MPEG-4 IPMP Overview and Applications Document," December 1998, (http://www.cselt.it/mpeg/ public/ mpeg-4_ipmp.zipH)
[7] Carlos Serro *et al*, "Open SDRM-An Open and Secure Digital Rights Management Solution," IADIS 2003, Lisboa, Portugal, May 2003.
[8] Gregor Siegert and Carlos Serro, "An Open-Source Approach to Content Protection and Digital Rights Management in Media Distribution Systems," ICT Conference 2003, Copenhagen, December 2003.
[9] "JAD-the Fast Java Decompiler," http://kpdus.tripod.com/jad.html/
[10] "MOSES 개발 시스템 분석 및 활용 방안 연구," 한국전자통신연구원, 2004.
[11] Jan Bormans and Keith Hill, "MPEG-21 Overview v.4," ISO/IEC JTC1/SC29/WG11/N4801, 2002.
[12] Multimedia Description Schemes Group, "MPEG-21 Rights Expression Language Working Draft," ISO/IEC JTC1/SC29/WG11/N4533, 2001.
[13] W3C Recommendation, "SOAP Version 1.2 Part 1: Messaging Framework," June 2003.

오 태 석 (Tae Suk Oh) 학생회원

2001년 청주대학교 전자정보통신반도체공학부 학사
2003년 서울시립대학교 전자전기컴퓨터공학부 석사
2004년~현재 서울시립대학교 전자전기컴퓨터공학부박사과정

김 용 철 (Yong Cheol Kim) 정회원

1981년 서울대학교 전자공학과 학사
1983년 KAIST 전기 및 전자공학과 석사
1983년~1986년 금성전기연구소
1993년 University of Southern California 박사
1993년~1996년 LG이노텍연구소 전문팀장.
1996년~현재 서울시립대학교 전자전기컴퓨터공학부 교수

최 범 석 (Bum Suk Choi) 정회원

1997년 충남대학교 컴퓨터과학과 학사
2001년 충남대학교 컴퓨터과학과 석사
2001년 3월~현재 한국전자통신연구원 방통융합콘텐츠보호연구팀

최 진 수 (Jin Soo Choi) 정회원

1990년 경북대학교 전자공학과 학사
1992년 경북대학교 전자공학과 석사
1996년 경북대학교 전자공학과 공학박사
1996년 5월~현재 한국전자통신연구원 선임연구원
2001년 2월~2005년 3월 한국전자통신연구원 데이터방송연구팀장 역임
2004년 10월~현재 TTA 데이터방송프로젝트그룹(PG312) 의장