
GStreamer를 이용한 DMP기반의 DRM 시스템

박기철* · 이주영** · 안상우** · 남제호** · 정희경*

DRM System using GStreamer based on DMP

Ki-Chul Park* · Joo-Young Lee** · Sang-Woo Ahn** · Je-Ho Nam** · Hoe-Kyung Jung*

본 연구는 지식경제부 및 정보통신연구진흥원의 IT신성장동력 핵심기술 개발사업의 일환으로 수행하였음.
[2007-S-003-02, 지상파 DTV 방송프로그램 보호 기술개발]

요 약

디지털 미디어의 빠른 기술 발달과 인터넷의 발달로 디지털 콘텐츠는 불법복제와 인터넷을 통한 배포의 피해로 지적 재산권에 대한 보호의 필요성으로 여러 표준화 단체들과 콘텐츠 공급자들이 DRM(Digital Rights Management) 시스템을 개발하였지만, 상호 DRM 시스템간의 상호운용성이 결여되는 문제점이 있다.

본 논문에서는 GStreamer를 사용하고 DMP(Digital Media Project)에서 제안하는 DRM 표준을 기반으로 상호운용 가능한 DRM 시스템을 설계 및 구현하였다.

ABSTRACT

Due to illegal copying and distribution via the internet, the digital media that have been introduced create certain intellectual property issues. Because of this, several intellectual property groups have been created to create and maintain Digital Rights Management systems (DRM), in order to crack down on illegal and unauthorized copying. However, these systems are often created by many different companies, which leads to problems of interoperability.

In this paper, we designed and implemented a DRM system with GStreamer, following the DMP.

키워드

DMP, DRM, GStreamer, Chillout, Toolpack, IDP

I. 서 론

IPTV(Internet Protocol Television)의 상용화, WiBro (Wireless Broadband), HSDPA(High Speed Downlink Packet Access)등 새로운 통신 서비스의 확대와, 통신과

방송의 융합 가시화로 인해 국내 디지털 콘텐츠 산업은 안정적인 성장세를 유지할 것으로 전망된다[1].

그러나 사용자에 의한 변형 및 복제가 용이한 디지털 콘텐츠의 특성 때문에 디지털 콘텐츠 시장에서 필수적인 요건으로 신뢰성 있는 유통체계의 확립이 요구되어

* 배재대학교 컴퓨터공학과(교신저자, 정희경)

접수일자 : 2007. 11. 5

** 한국전자통신연구원

지고 있다. 이에 콘텐츠 업체 및 IT 업체들은 콘텐츠를 보호하고, 신뢰성 있는 유통체계 확립을 위해 콘텐츠 보호를 위한 DRM (Digital Rights Management) 기술들을 콘텐츠에 적용되어 유통되고 있다. Apple社의 iTunes와 Microsoft社의 Windows Media DRM등의 DRM사례는 성공적이지만, 시스템의 의존성으로 인하여 상호 DRM 시스템 간의 통합이 불가능하다. 이와 같은 시스템 간의 상호 운용성의 결여는 사용자의 콘텐츠 소비의 장애가 된다.

이에 본 논문에서는 DMP(Digital Media Group)의 IDP-2(Interoperable DRM Platform-2)기반의 레퍼런스 소프트웨어 Chillout과 멀티미디어 프레임워크인 GStreamer를 이용하여 DRM 시스템을 구현하였다[2,3]. 그 결과 기존의 문제점인 미디어 포맷에 대한 호환성과 성능이 향상되었으며, 이기종의 DRM시스템 호환성이 향상되었다.

II. 관련연구

2.1 DMP(Digital Media Project)

DMP는 IDP기술의 개발 및 표준을 제안하고, IDP-2 기반 레퍼런스 소프트웨어 Chillout을 개발배포하고 있다.

2.1.1 DMP Value Chain의 장치의 구성

DMP에서는 콘텐츠 공급자의 소비자의 유통을 DMP 가치 사슬로 표현하고, 장치(Device)들의 연결로 표현하는데 그 구조는 그림 1과 같다.

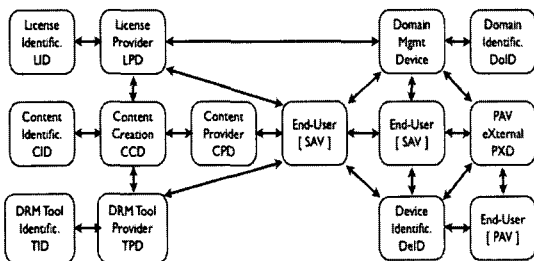


그림 1 DMP 가치사슬 구조
Fig. 1 Architecture of DMP Value Chain

TID(Tool Identification Device)는 미디어 암호화를 위해 생성한 툴과 툴에 대한 정보를 등록 관리한다. LID(License Identification Device)는 콘텐츠에 대한 권한과 라이선스를 관리하고, CCD (Content Creation Device)는 보호된 콘텐츠를 생성한다. 보호된 콘텐츠 생성에 필요한 라이선스와 툴은 LPD(License Provider Device)와 TPD(Tool Provider Device)로부터 공급 받고, CCD는 라이선스 정보와 툴 정보를 담은 XML문서와 보호된 리소스 정보를 합쳐서 MPEG-21미디어 포맷을 기반으로 DCF(DMP Content File) 과 DCI(DMP Content Information)을 생성한다[4]. 생성된 미디어는 SAV(Stationary Audio Video) 또는 PAV (Portable Audio Video)장치들을 통해 엔드 유저가 소비하게 된다.

2.1.2 DMP 콘텐츠의 표현

DMP는 상호운용 가능한 콘텐츠를 표현하기 위해 MPEG-21에 스키마 일부를 참조하고, DMP에서 개발한 스키마와 다른 기술들을 참조한다[5]. DMP의 스키마는 DMP콘텐츠를 표현 하는 전반적인 기술을 제공하는데, 이는 콘텐츠 식별자, 메타 데이터, 리소스 정보(리소스 ID, 리소스 URL 등) 그리고 DRM 정보(라이선스나 IPMP 정보 등)를 포함한다.

그림 2와 같이 콘텐츠를 표현하는 기술정보를 구성하여 DCI로 표현한다. DCI의 스키마는 상호 운용 가능하고 개방적으로 설계되어 있다[6].

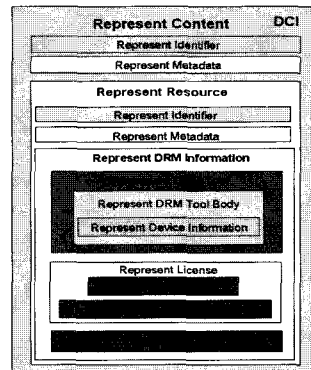


그림 2 DCI 계층 구조
Fig. 2 DCI Hierarchy

2.1.3 Chillout

DMP에서는 상호운용적인 시스템을 표현하기 위해 레퍼런스 소프트웨어 Chillout을 개발 배포하고 있다. 구조는 그림 3과 같다[7].

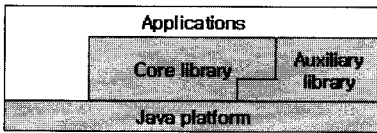


그림 3 Chillout 개발 계층
Fig. 3 Chillout Development Layers

- **Java platform** : JDK, Apache Tomcat과, Apache Axis, SOAP(Simple Object Access Protocol)를 이용한 Web-service와 권한과 인증의 검증정보를 담은 XML파싱에 JAXB (Java Architecture for XML Binding)를 사용하고 미디어 렌더링을 위한 JMF(Java Media Framework)로 구성되어 있다[8,9,10,11].
- **Core Library** : IDP-2의 기술을 구현한 근본적인 객체들로 구성된 라이브러리이다.
- **Auxiliary library** : 모든 장치들이 구동에 필요한 기능들을 내장한 객체로 구성된 라이브러리이다. 개발자의 필요에 따라 대체 가능하다.
- **Applications** : 코어계층과 보조계층이 동작을 표현하기 계층으로 모든 장치의 예를 담고 있다.

2.2 멀티미디어 프레임워크 GStreamer

GStreamer는 오픈소스 멀티미디어 프레임워크로서 특징은 다음과 같다[12].

- 광범위한 운영체제와 프로세서 그리고 컴파일러의 환경에서 사용 가능
- 적은 자원의 환경에서 고성능을 보장
- MPEG-4 또는 H.264 등의 최신 포맷 지원
- 용도에 맞는 커스텀 필터 개발에 용이

GStreamer는 엘리먼트로 구성 되고 서로의 연결을 통해 스트림을 전송시켜 미디어를 표현 한다. 이는 용도에 따라 원하는 엘리먼트의 삽입과 삭제가 가능한 구조로

용도에 따른 미디어 프로그램 개발의 유연성을 제공한다.

2.3 GStreamer-Java

GStreamer를 JNA(Java Native Access)를 이용해 Java 환경으로 바인딩 한 것으로 Java 가상머신에서 GStreamer의 라이브러리 사용이 가능하도록 한다[13].

III. 시스템 설계

본 시스템은 DMP의 레퍼런스 소프트웨어 Chillout의 성능의 개선과 여러 종류의 미디어 포맷에 대한 호환성 향상과 Chillout에 적용 가능한 레퍼런스 소프트웨어를 목적으로 한다. DCF와 DCI의 처리를 위해 코어 계층은 유지하고, 보조 계층의 보호된 미디어 처리에 대한 설계 되어야한다. 미디어 리소스의 렌더링을 담당한 JMF 대신 성능과 호환성이 높은 GStreamer를 이용하고, 기존 Chillout의 성능 및 호환성이 증대 되도록 설계하였다.

3.1 DMP 콘텐츠의 소비 시나리오

DCF와 DCI의 소비는 콘텐츠의 권한 유무에 따라 그림 4와 나타낼 수 있다[14].

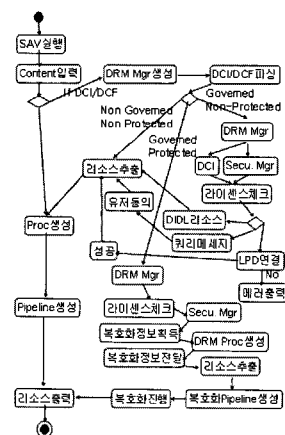


그림 4 DCI, DCF 처리 과정
Fig. 4 DCI, DCF Process Diagram

그림 4의 활동 다이어그램은 Non Governed, Non Protected와 Governed, Non Protected 그리고 Governed,

Protected 3가지 경우로 나뉜다. 콘텐츠 공통의 과정으로 DCI와 DCF는 파싱과정을 통하여 어떠한 조건의 콘텐츠인지 확인하고, 해당 조건에 따라 아래의 과정을 통해 소비된다.

- **Non-Governed, Non-Protected** : 리소스 정보를 추출 후 리소스 프로세서 객체를 생성하고 생성된 객체는 리소스의 재생을 위해 GStreamer의 파이프라인을 생성 후 재생과정을 진행한다.
- **Governed, Non Protected** : DCI에서 라이선스를 검사하거나 DRM Manager 또는 Security Manager 객체에서 라이선스를 검사한 후 라이선스에 따라 리소스 정보를 추출하고, 리소스 프로세서 객체를 생성하고 파이프라인을 생성 후 재생 과정을 진행한다.
- **Governed, Protected** : 앞선 과정 중 라이선스 확인 과정을 진행 한 후 Security Manager 객체는 DRM 정보를 얻는다. DRM 프로세서 객체를 생성하고, DRM 정보에 따라 DRM 프로세서 객체는 복호화 툴의 유무를 확인하고 툴이 없는 경우 TPD로부터 툴을 요청한다. 툴의 획득이 정상적으로 수행되면 DRM 프로세서 객체는 DRM 정보를 GStreamer에 전달되고, 이를 전달 받은 GStreamer는 DRM 정보를 파싱 과정을 거쳐 복호화 키를 설정한 툴을 포함하는 파이프라인을 생성하고 재생 과정을 수행한다.

3.2 GStreamer와 Chillout SAV의 연동

Chillout의 SAV와 GStreamer를 연동하기 위해 Java로 바인딩 된 GStreamer-java를 사용한다. 이것은 GStreamer의 구성요소를 객체로 제공한다. DRM 프로세서 객체는 GStreamer-java를 통해 DRM 정보에 따라 툴 적용이 가능하도록 직접 GStreamer의 엘리먼트를 사용한다. 그림 5는 GStreamer의 엘리먼트와 파이프라인을 사용해 SAV의 구현할 경우 미디어 처리를 위한 구조는 나타낸 구조이다.

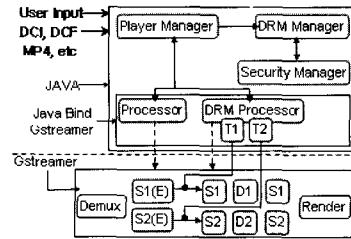


그림 5. Chillout SAV 구조
Fig. 5. Architecture of Chillout SAV

3.3 DRM 정보 처리

DRM메세지로 정의된 DRM정보는 DRM프로세서로 전달된다. DRM프로세서는 DRM 파이프라인으로 InitializeTool, KeyData, ToolDisconnect 등의 DRM메시지를 전달한다[15]. DRM프로세서는 이 정보를 파싱하여 툴을 로드하고 복호화 키 값을 툴에 전달한다.

3.4 DRM 툴의 설계

복호화 과정을 수행하는 툴은 DRM프로세서의 요청에 따라 파이프라인에 추가 및 링크되고, DRM프로세서로부터 KeyData 메시지를 전달받아 대한 응답을 DRM 프로세서에 반환한다. 그림 6은 툴이 전달 받는 복호화 KeyData메시지 스키마의 일부이다.

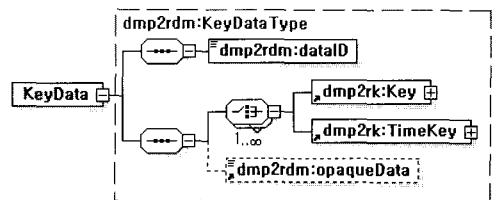


그림 6 KeyData 스키마 다이어그램
Fig. 6 Schema Diagram of KeyData

3.5 GStreamer의 DRM 파이프라인 생성

소비를 위한 미디어 파이프라인은 미디어의 포맷에 적합한 엘리먼트의 연결로 구성되고, 미디어의 MIME Type 또는 엘리먼트 간의 입출력 스트림에 따라 필요한 엘리먼트를 자동으로 선택 연결하는 기능이 필요하다. 그림 7은 보호된 미디어 재생을 위한 DRM 파이프라인의 구조이다.

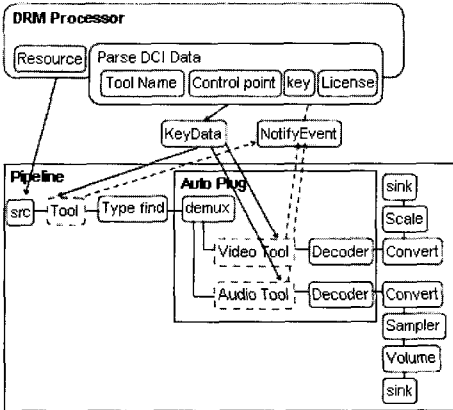


그림 7 DRM 파이프라인의 구조도
Fig. 7 Diagram of DRM Pipeline Architecture

그림 7에서 DRM파이프라인의 구성 시점에서 DRM 정보의 파싱 결과에 따라 복호화 툴 엘리먼트가 선택되고 복호화키 값을 입력받는다. 컨트롤 포인트는 크게 역다중화기 이전과 역다중화기의 이후 즉 역다중화기와 디코더 사이로 지정하였고, 이는 DRM정보에 기록된 위치정보를 파싱하여 얻게 된다.

IV. 시스템 구현

본 시스템의 구현은 IBM-PC 호환 컴퓨터의 MS Windows XP Service pack 2 기반의 운영체제와, Linux Ubuntu 7.04에서 구현하였고, Ch-illout의 구동을 위해 JDK 1.5 이상 버전에서 구현되었다. GStreamer의 버전은 0.10.13을 기준으로 하였고, Windows 환경에서 툴 Build를 위해 MS Visual Studio C++ 6.0이 사용 하였다.

4.1 GStreamer의 구현

GStreamer의 DRM 미디어의 복호화를 위해 툴은 GStreamer의 엘리먼트 형태로 구현하였다. 구현된 엘리먼트는 GStreamer의 엘리먼트 목록에 추가되고, 엘리먼트 정보는 gst-inspect를 실행하여 확인할 수 있다. 그림 8은 구현된 DES 복호화 알고리즘 툴의 정보를 표시한 것이다.

DRM정보의 처리는 XML파싱 통해 사용되는 툴을 로드 가능하도록 하였다. 재생을 위한 리소스의 포맷은 MPEG-4 비디오 포맷(XVI, DIVX, QuickTime)을 기준으

```

M:\MINGW32~
Plugin Details
Name          DESTool
Description   DES Algorithm encrypt or decrypt Tools
Filename      E:\workspace\vc\gst\example\des\tool\Debug\des\tool.dll
Version       0.10.5.1
License       LGPL
Source module  gst-plugins-toolpack
Binary package GStreamer Tool Plug-ins
Origin URL    Unknown package origin

GObject
+--GObject
+--GElement
+--GStateDec

Pad Templates
SRC template 'src'
Availability Always
Capabilities ANY

SINK template 'sink'
Availability Always
Capabilities ANY
    
```

그림 8 DES 툴 엘리먼트의 정보
Fig. 8 Information of DES tool element

로 하였고, 동적으로 미디어 포맷에 따라 역다중화기와 디코더 엘리먼트를 생성 링크 가능하도록 하였다. 외부에서 툴에게 전달되는 KeyData는 XML형식으로 엘리먼트 프로퍼티를 통해 문자열 형태로 전달하였고, KeyData에 대한 응답은 XML형식으로 시그널이벤트의 변수를 통해 DMR프로세서가 전달 받도록 구현하였다.

4.2 CCD의 DCF, DCI생성과 SAV의 재생

CCD를 이용하여 새로운 리소스를 등록하고, 라이선스 설정을 가능하고, 설정된 라이선스에 따라 Tool을 적용한 DCI, DCF를 생성하도록 구현하였으며, 생성된 DCI와 DCF는 화면을 통해 확인 하도록 하였다. 그림 9는 CCD를 통해 생성된 DCF 파일의 예이다.

```

Document
├── Element ns14 DiDL
│   └── Element ns14 Item
│       ├── Element ns14 Descriptor
│       │   └── Element ns14 Statement
│       │       └── Element Identifier sum:smip:chillout:com:ant:1@ttw:2009/07/
│       └── Element ns14 Descriptor
│           ├── Element ns14 Statement
│           │   └── Element ns13 Metadata
│           │       ├── Element ns13 StructuredData
│           │       │   └── Element ns23 Map7
│           │       │       └── Element ns23 Description
│           │       │           └── Element ns23 CreationInformation
│           │       │               ├── Element ns23 Creation
│           │       │               │   ├── Element ns23 Title CSI MI
│           │       │               │   ├── Element ns23 Abstract
│           │       │               │   ├── Element ns23 FreeTextAnnotation CSI T
│           │       │               │   ├── Element ns23 Creator
│           │       │               │   ├── Element ns23 Role
│           │       │               │   ├── Element ns23 Agent
│           │       │               │   ├── Element ns23 CreationCoordinates
│           │       │               │   ├── Element ns23 Date
│           │       │               │   └── Element ns23 CopyrightString NBC
    
```

그림 9 CCD에 의해 생성된 DCF
Fig. 9 Example of DCF created by CCD

SAV에서는 DCI, DCF를 열면 라이선스 내용을 먼저 출력하고, 재생 가능 하도록 구현했다. 그림 10은 DCF로부터 미디어를 재생하는 그림이다.

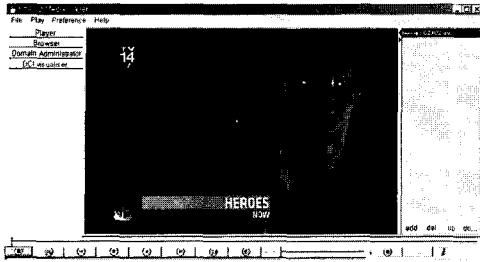


그림 10 SAV에서 DCF 재생
Fig. 10 Playing DCF on SAV

V. 고찰 및 결론

기존의 Chillout의 시스템은 순수 Java 기반으로 플랫폼 독립적이며 개방적인 시스템이다. 성능과 호환성 향상을 위한 GStreamer와 통합은 물론 GStreamer가 대부분의 CPU와 OS 환경에서도 훌륭하게 동작 하지만, 자바에 기반 Chillout의 시스템 독립성과 개방성이 일부 제한 될 것으로 사료된다. 하지만, 기존에 문제가 된 JMF의 제한적인 미디어 포맷의 호환성과 압축 화질의 낮은 성능은 크게 개선되었다. 표 1에서 JMF 기반과 GStreamer 기반으로 나뉜 상호 운용성 및 성능에 대한 비교를 하였다.

표 1. JMF 기반과 GStreamer 기반 Chillout 비교
Table 1. Comparison of Chillout based on JMF and GStreamer

비교항목	JMF 기반	GStreamer 기반
시스템 의존성	독립적	상대적 독립적
안정성	높음	높음
상호 운용성	높음	높음
보안성	좋음	아주 좋음
틀의 확장성	좋음	아주 좋음
미디어 지원	많음	아주 많음
성능	중지 않음	아주 좋음

기존의 성능적인 문제는 어느 정도 보완이 되었지만, Java와 C의 혼용이라는 점에서 실제 환경에 적용하기에는 다소 복잡한 구조를 가지고 있다. Chillout의 소비기인 SAV와 PAV는 C를 기반으로 통합이 필요하며 스트림 미디어에 대한 DRM 및 소비의 지원과, 휴대용 소형 단말기에서의 DMP 기술의 구현도 향후 연구 과제로 사료된다.

DMP의 표준이 향후 DRM 시장에서 어떠한 평가를 받을지 가늠하는 것은 아직은 시기상조이지만 DRM 시스템의 상호 운용성 측면에서 큰 기여를 할 것으로 예상된다.

참고문헌

- [1] “2006년도 국내 디지털 콘텐츠 산업 시장조사 보고서 - 한국소프트웨어진흥원, 2006
- [2] The DMP IDP-2 Specification. <http://www.dmpf.org/project/ga09/idp-2.html>
- [3] Chillout, the Interoperable DRM Platform Reference Software, <http://chillout.dmpf.org/>
- [4] L. Chiariglione, N. Earnshaw, “Interoperable DRM Platform (IDP-2)”, 2006
- [5] ISO/IEC 21000-9, Information technology Multimedia framework (MPEG-21) - Part 9: File format
- [6] Xiaofan Chen, Tiejun Huan “Interoperability Issues in DRM and DMP Solutions”, ICME 2007
- [7] Filippo Chiariglione, Yuqiang Liao, “Chillout - the Open Source DRM Software”, IPDM06, [http://www.ipdm06.org/papers/Chillout The Open Source Software DRM.pdf](http://www.ipdm06.org/papers/Chillout%20The%20Open%20Source%20Software%20DRM.pdf)
- [8] The Apache Software Foundation, Apache Web Services Project Axis, <http://ws.apache.org/axis/>
- [9] Simple Object Access Protocol (SOAP) 1.1, <http://www.w3.org/TR/soap/>
- [10] Java Architecture for XML Binding, <http://java.sun.com/webservices/jaxb/>
- [11] Java Media Framework API, <http://java.sun.com/products/java-media/jmf/>
- [12] GStreamer features, <http://gstreamer.freedesktop.org/features/>
- [13] jna: Java Native Access (JNA): Pure Java access to native libraries, <https://jna.dev.java.net/>
- [14] CCD and SAV walkthrough, <http://www.dmpf.org/open/dmp1039.zip>
- [15] Jooyoung Lee, Hyon-Gon Choo, Filippo Chiariglione, Jeho Nam, “Proposal for New Message Protocol of Inter-Device Communication”, <http://www.dmpf.org/open/dmp0969.doc>

저자소개



박 기 철(Ki-Chul Park)

2007년 배재대학교 컴퓨터공학과 (학사)

2007년 ~ 현재 배재대학교 컴퓨터 공학과 석사과정

※ 관심분야 : 멀티미디어 시스템, 임베디드 시스템, 유비쿼터스 센서 네트워크, web-service



이 주 영(Joo-Young Lee)

2003년 아주대학교 미디어학과 (학사)

2006년 한국과학기술원 전산학과 (석사)

2006년 ~ 현재 한국전자통신연구원 전파방송연구단
※ 관심분야 : 콘텐츠 저작권 보호기술, 멀티미디어 데이터베이스



안 상 우(Sang-Woo Ahn)

1997년 경희대학교 전자공학과 (학사)

1999년 경희대학교 전자공학과 (석사)

1999년 ~ 현재 : 한국전자통신연구원 선임연구원
2003년 ~ 현재 : 충남대학교 전자공학과 박사과정
※ 관심분야 : 방송콘텐츠보호, MPEG 시스템, 지상파 DMB, 영상처리/코딩



남 제 호(Je-Ho Nam)

1992년 홍익대학교 전기제어공학과 (학사)

1996년 University of Minnesota, Dept. of Electrical Eng. (석사)

2000년 University of Minnesota, Dept. of Electrical Eng. (박사)

2001년 ~ 현재 한국전자통신연구원(ETRI) 방송미디어 연구그룹 선임연구원, 방통융합콘텐츠보호 연구팀장

2007년 ~ 현재 과학기술연합대학원대학교(UST) 이동통신 및 디지털방송공학 겸임 부교수

※ 관심분야 : 멀티미디어 신호처리, 디지털방송기술, MPEG, 콘텐츠 보호관리



정 회 경(Hoe-Kyung Jung)

1985년 광운대학교 컴퓨터공학과 (학사)

1987년 광운대학교 컴퓨터공학과 (석사)

1993년 광운대학교 컴퓨터공학과 (박사)

1994년 ~ 현재 배재대학교 컴퓨터공학과 교수

※ 관심분야 : 멀티미디어 문서정보처리, XML, SVG, Web Services, Semantic Web, MPEG-21, 유비쿼터스 센서 네트워크