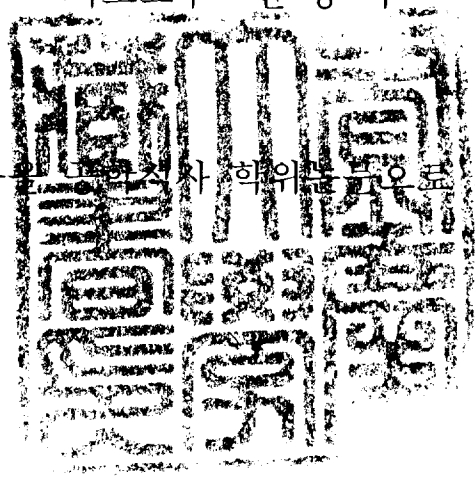
공학석사 학위논문

# WIPI 기반의 DRM 시스템 설계 및 구현

지도교수 신 상 욱



이 논문을 공학석사 학위논문으로 제출함

2005년 8월


부경대학교 대학원

정 보 보 호 학 과

김 태 정

김태정의 공학석사 학위논문을 인준함

2005년 8월 31일

주심 이학박사 이 경 현 

위원 공학박사 김 창 수 

위원 이학박사 신 상 욱 

# <차 례>

# < 표 차례 >

# < 그림 차례 >

Design and Implementation of DRM System based on WIPI

Tae Jeong Kim

Department of Information Security, The Graduate School, Pukyong National University

## Abstract

The number of mobile (or cellular) phone users has been rapidly increasi ng since the last 1990s. In the beginning, the main and unique purpose of mobile phone was to communicate, namely making and receiving calls. Ho wever, various functions of mobile phone have been added over the years, so now mobile phone users can go to the Internet and use various enterta inment functions including uploading and downloading the game, still and motion pictures and music services. Then we need Digital Rights Manage ment(DRM) System which allow contents to be distributed in a controlled manner. Until now each of domestic mobile communication corporation de veloped different mobile platform since mobile contents developed on differ ent platform do not have interoperability among mobile platforms with diff erent platform, recently TTA(Telecomunications Technology Association) makes progress to standard WIPI(Wireless Internet Platform for Interopera bility) platform which can unify the format of content used at mobile platf orm. The wireless Internet has the characteristics of low data transmissio n speed, high error rate, limited memory resource and low processing abili ty. Therefore it needs to develop DRM technique suitable to a mobile envi ronment. In this thesis, we introduce wireless DRM technique and analyze OMA DRM standard. Also we introduce WIPI platform which is a standa rd mobile platform and analyze security requirements of WIPI platform. T hen we design WIPI based DRM system and propose DRM APIs, which need to be appended to WIPI platform.

# I. 서 론

국내의 무선 인터넷 사용자의 수는 최근 급속히 증가하는 추세에 있으며, 초기의 단순한 통화만을 목적으로 하던 이동통신의 형태가 점차 핸드폰 상에서 인터넷을 사용하는 무선 인터넷으로 변화하면서 게임이나 사진, 동영상, 음악 서비스 등의 여러 가지 엔터테인먼트 기능을 요구하는 복합적인 단계에 이르고 있다. 현재 국내의 모바일 개발 플랫폼은 각 이동통신사별로 제각기 다르지만, 최근에는 WIPI(Wireless Internet Platform for Interoperability) 플랫폼으로 표준화하기 위한 준비를 하고 있다. 실질적으로 모바일 어플리케이션 프로그래밍의 경우 C 또는 자바(J2ME) 언어를 사용한다. SK와 LG의 경우 자바 기반의 자바 가상머신을 사용하는 플랫폼을 지원하고 KTF의 경우 C언어를 지원하는 BREW 기반의 플랫폼을 지원하고 있다. 속도 면에서는 C를 사용하는 BREW 플랫폼을 선호하고 개발 환경의 편의성을 위해서는 자바 기반의 자바가상머신 플랫폼을 선호한다. 따라서 서로 다른 플랫폼으로 개발되는 모바일 콘텐츠는 서로 다른 플랫폼의 단말기에서는 상호 연동성을 가질 수가 없기 때문에 통합 플랫폼으로 WIPI가 개발 되었으며, 이것은 앞으로 선택사항이 아니라 강제사항으로 향후 출시되는 모든 핸드폰은 WIPI 플랫폼을 반드시 탑재해야만 한다. 그리고 국내 표준뿐만 아니라 TTA(Telecomunications Technology Association)에서 IETF(The Internet Engineering Task Force)의 차세대 이동통신 프로젝트 그룹인 OMA(Open Mobile Alliance) 실무반을 통해 국제적인 표준화 활동에도 목적을 두고 세계화를 추진하고 있다.

WIPI 플랫폼의 표준화 범위는 이동통신사업자들 요구사항이 단말기가

최종적으로 다운로드 되는 오브젝트가 기계코드(machine code) 형태를 요구함에 따라 콘텐츠 호환을 보장하는 범위 내에서 다양한 솔루션이 개발 될 수 있도록 되어있다. 또한 WIPI는 구조를 포함한 API 등의 모든 규격을 표준으로 상정하여 공개하고, 자유롭게 개발하여 사용할 수 있도록 하였으며 3GPP(3rd Generation Partnership Project) 규격의 하나로 상정하였기 때문에 이에 따른 개발상의 제약은 매우 적다.

디지털 저작권관리(Digital Rights Management:DRM) 기술은 콘텐츠의 지적 재산권이 디지털 방식에 의해서 안전하게 보호, 유지되도록 하여 콘텐츠가 창작에서부터 소비에까지 이르는 모든 유통 시점에서 거래 및 분배규칙, 사용규칙이 적법하게 성취되도록 하는 기술이다. DRM은 콘텐츠의 판매 및 배포, 소비에 이르는 새로운 방법을 제공하여 콘텐츠를 제공하는 CP(Content Provider) / SP(Service Provider)가 콘텐츠의 맛보기(preview), 사용 횟수 혹은 만료일의 제한, 콘텐츠의 구독(subscription), 콘텐츠의 다운로드 혹은 스트리밍 서비스와 같은 다양한 비즈니스 모델을 허용하기 위한 사용규칙(Usage rules)을 적용할 수 있다.

지금까지 유선 인터넷 망에서 적용되어 온 DRM 기술은 무선 콘텐츠 시장의 활성화와 함께 무선 인터넷 망으로 점차 그 범위를 확대해 나가고 있다. 그러나 낮은 데이터 전송 속도와 높은 패킷 전송 실패율을 갖는 무선 인터넷 망을 기반으로 하는 무선 디바이스에 기존 유선 DRM을 적용할 경우, 적은 메모리 자원, 제한된 처리능력 등의 제약사항으로 인하여 PC에서와 같이 강력하고 복잡한 암호화 알고리즘은 사용될 수 없다. 이러한 제약사항에도 불구하고 무선 인터넷 망은 DRM을 위한 새로운 가능성을 제시했으며, 그 중 하나는 micro billing 즉, DRM protected content 사용에 대한 과금이 핸드폰 요금에 추가될 수 있다는 점으로

PC에서는 유효하지 않은 기술이다.[1].

본 논문에서는 무선 DRM 기술과 관련하여 OMA DRM version1.0과 version2.0 규격에서 제시하는 전체 시스템 구성과 기능, 그리고 콘텐츠 전달방식과 콘텐츠 포맷, 권리 명세, 다운로드 방식, 권한 획득 프로토콜 등을 서술하고 모바일 표준 플랫폼인 WIPI 환경에서 안전한 콘텐츠 유통이 가능하도록 DRM 시스템을 설계하고 추가되어야 할 컴포넌트를 제시한다. 논문의 구성은 2장에서 관련연구로써 OMA DRM규격과 WIPI 플랫폼에 대해 기술하고 3장에서 WIPI 기반의 DRM 컴포넌트를 설계하고 에뮬레이터 상에서 DRM시스템을 구현하였다. 마지막으로 4장에서 제안된 논문의 결론을 맺도록 한다.

# II. 관 련 연 구

## 2.1 모바일 환경의 DRM 개요

인터넷 통신의 급속한 성장과 더불어 인터넷은 디지털 콘텐츠 거래의 효율적인 분배 채널의 하나가 되었다. 오늘날 이 채널은 모바일 영역으로 확장되고 있다. 확실히 그것은 desktop device 또는 portable device 사용자의 네트워크를 이용하여 디지털 정보의 모든 분배의 가장 이상적이다. 그러나 디지털 콘텐츠가 관리되지 않고 보호되지 않는다면 그것은 쉽게 복사, 변경, 손상 그리고 대량의 부정자들에게 분배될 것이다. 창작자로부터 디지털 작품들의 유용하고 안전하고, 신뢰된 이동을 허락하고, 판매자와 구매자에게 공개하는 DRM은 이러한 문제를 다루는데 필요하다.

앞으로 암호화된 신용카드, 소액 지불, 그리고 전자화폐는 모바일 장치 내에 자리 잡을 것이며 디지털 콘텐츠 거래는 전자 기계와 모바일 domain 사이의 적절한 영역이 될 것이다. 따라서 향후 모바일 미디어 상거래의 첫 번째 이슈가 될 기초 골격으로 모바일 DRM이 다루어 져야 할 것이다. 모바일 DRM 은 모바일 네트워크상의 요구사항에 따라 디지털 콘텐츠 내의 저작권을 관리 사용하는 행동, 절차, 정책, 작품 소유권 그리고 툴의 집합이다.

모바일 환경에서 DRM은 인터넷을 기반으로 하는 DRM과 단말기의 성능, 특성에 따라 서로 구분되어야 할 부분들이 있다. 인터넷을 기반으로 하는 DRM의 경우 PC의 하드웨어나 소프트웨어의 구조가 일반적이고 잘 알려져 있기 때문에 사용되는 소프트웨어와 콘텐츠를 네트워크를 통하여 다

운받고, 적용된 DRM 모듈에 의해 어떠한 PC에서도 사용될 수 있다. 그리고 장비의 성능도 고성능, 고용량이므로 유통하는 콘텐츠의 종류도 제한이 거의 없다. 그와 반대로 현재 모바일 환경의 단말기는 제조회사에 따라 하드웨어나 소프트웨어의 구조가 일반적이지 않고 서로 상이하기 때문에 하나의 콘텐츠가 모든 단말기에 적용되기 어렵고 단말기의 성능과 용량도 PC에 비해 너무도 작기 때문에 콘텐츠를 유통하는데 어려움이 있다.[2].

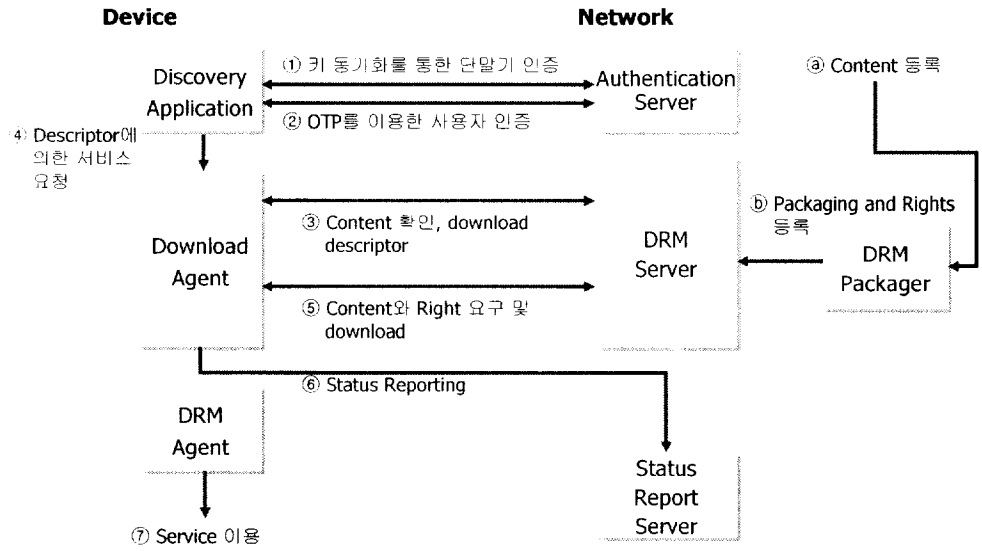
### 1) OMA DRM v1.0 플랫폼

OMA(Open Mobile Alliance) 에서는 무선 모바일 환경에서 사용되어지는 콘텐츠의 안전한 유통과 콘텐츠의 사용에 대한 상호 운용성을 부여할 수 있는 DRM 시스템에 대한 규격을 정의한다. OMA DRM에서 콘텐츠 제공자는 DRM 도입 이전에는 단일 가격 정책으로 콘텐츠를 판매할 수 밖에 없었지만 도입 후에는 동일한 콘텐츠에 다양한 사용 권리 조건을 명시함으로써 다수의 가격정책으로 콘텐츠를 판매 할 수 있는 형태의 DRM 콘텐츠 사용에 대한 규칙을 정의 할 수 있는 특징을 가지고 있다. 그리고 더 이상 콘텐츠에 가치를 두지 않고 콘텐츠의 사용 권리인 Rights에 가치를 둠으로써 저작권과 사용권한에 대해서 안전함을 보장한다. 따라서 콘텐츠 자체에 대한 판매가 아닌 Rights를 판매하는 것이 가능한 특징도 가지고 있다. 이러한 특징을 가지고 있는 OMA DRM은 3GPP(Third Generation Partnership Project)에서 추진해온 DRM 사양을 이전받아 Phase 1단계의 OMA DRM v1.0을 2003년 최종 발표하였으며, 2004년 7월 OMA DRM v2.0을 발표했다.

OMA DRM v1.0에서의 DRM System은 그림 1과 같이 Discovery Application, Download Agent, DRM Agent로 구성되는 Device side와

Presentation Server, Download Server, DRM Package, 그리고 Status Report Server로 구성되며 Network side로 구분된다.[4].

Presentation Server는 콘텐츠의 Browsing을 돕는 웹 서버로써 Device는 Presentation Server를 통해 HTML/WML page를 Browsing 할 수 있으며 콘텐츠의 다운로드를 위하여 Download Server로 Redirection할 수 있는 역할을 한다. 그리고 Download Server는 사용자가 선택한 콘텐츠(원본 콘텐츠 또는 DRM Content)에 대한 다운로드 서비스를 제공한다. 사용자의 device에 콘텐츠 그리고/또는 권리 개체의 다운로드 처리하는 역할을 담당한다. Status Report Server는 Device 상에서 콘텐츠의 정상적인 다운로드, 설치 및 사용 정보에 대한 보고를 수집하는 역할을 하고, DRM Packager는 콘텐츠의 암호화 수행 후 암호화 된 원본 콘텐츠를 헤더와 함께 DRM Content의 형태로 구성한다. DRM Content에 포함되는 헤더 정보에는 원본 콘텐츠의 형식, 콘텐츠 식별자, 암호화 정보, Rights 발급 서비스 정보 등을 포함한다. Discovery Application은 Device 상에서 콘텐츠 형식에 적합한 웹 브라우저 혹은 어플리케이션을 사용하여 콘텐츠를 검색하기 위한 User Agent역할을 하고, Download Agent는 Download Descriptor에 의해 설명되는 콘텐츠의 다운로드 기능을 담당한다. Download Descriptor는 콘텐츠의 다운로드를 위한 콘텐츠 메타데이터(type, object URI, Right Issuer, URL 등)와 Download Agent에 대한 지시 사항을 정의한다. DRM Agent는 콘텐츠의 다운로드, 권리 개체에 따른 콘텐츠의 사용 제어, 권리 개체의 관리 등을 처리한다.



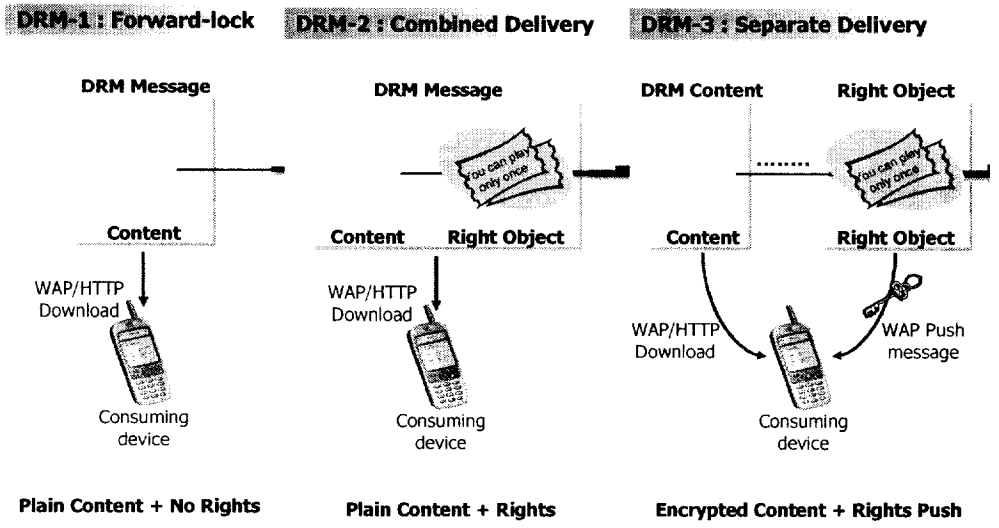

[그림 1] OMA Download System Architecture

OMA Download System의 전체 흐름은 그림 1과 같이 콘텐츠의 다운로드를 위해 먼저 콘텐츠를 등록하고, 등록된 콘텐츠를 패키징하고 권리를 등록한다. 그리고 등록된 콘텐츠에 대하여 CP와 SP는 콘텐츠에 대한 과금 정책을 설정하고 서비스에 대하여 권리를 생성하여, 패키징된 DRM 콘텐츠를 최종적으로 서비스하기 위해 Presentation Server와 Download Server에 해당 콘텐츠에 대한 정보 및 Download Descriptor를 등록한다.

사용자는 Discovery Application을 사용하여 구매하려는 콘텐츠를 검색하고, HTTP 프로토콜을 사용하여 Presentation Server로부터 콘텐츠를 다운로드할 수 있는 URL이나 Download Descriptor를 가져온다. 그리고 Download Agent를 실행하여 Download Descriptor의 처리 규칙에 따라 실행되며, 사전에 정의되어 있는 Status Report Server에 실행 내역을 보고한다. Download Descriptor의 버전이 디바이스에서 지원되지 않을 경우 상태 보고 후 콘텐츠 다운로드 과정을 중단한다. 이렇게 다운로드 된 콘텐

츠를 Rights에 명시된 내용에 따라 사용자가 사용할 수 있도록 한다.

OMA DRM에서 콘텐츠의 전달 방식은 그림 2와 같이 세 가지의 콘텐츠 전달 방식을 정의하고 있다.[3][5].




[그림 2] OMA의 DRM 적용 방식

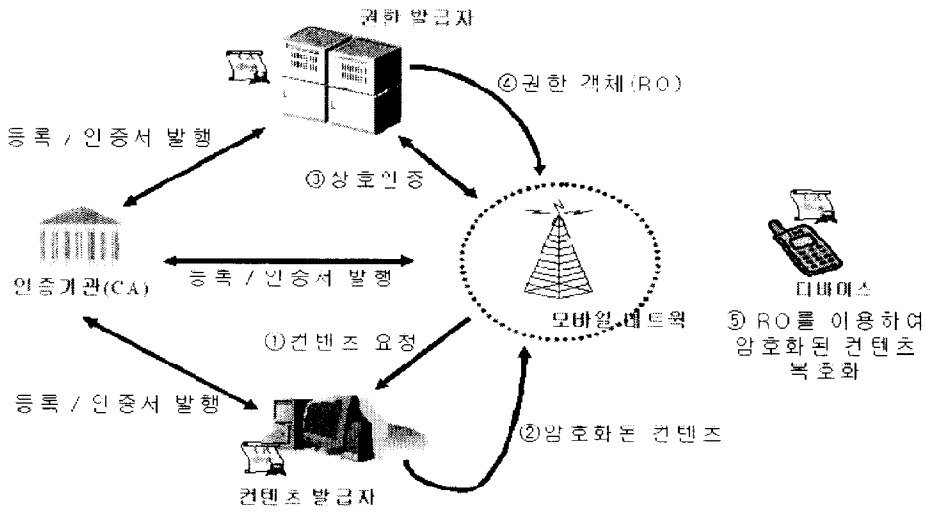
DRM message가 디바이스로 전달되면 디바이스는 이를 다른 디바이스로 전송하지 않도록 제한하는 'Forward-Lock'과 미디어 객체에 권리 객체를 포함한 DRM message를 디바이스로 전송하는 'Combined Delivery', 그리고 미디어 객체와 권리 객체는 별개로 디바이스로 전송될 수 있는 'Separate Delivery'로 구분된다. 특히 'Separate Delivery' 방식은 디바이스로부터 다른 디바이스로 제약 없이 복사될 수 있기 때문에 Superdistribution이 가능하게 된다. Superdistribution은 콘텐츠 사용자가 자신이 구입한 콘텐츠를 E-mail, CD-ROM, 디스켓 등을 통해 다른 사람에게 반복적으로 배포할 수 있게 하여 콘텐츠의 급속한 확산을 가능하게

하는 기술이다. 이는 OMA DRM에서 정의 하고 있는 콘텐츠 포맷 형식의 콘텐츠에 대한 배포를 허용하고, superdistributed된 콘텐츠의 사용을 위해 Rights를 발급받을 수 있게 한다. 디바이스는 콘텐츠 포맷의 Rights Issuer 필드에 정의된 URL을 통해 Rights 발급을 요청하게 된다. 기존 Secure distribution 기술은 사용자에게 종속적인 키를 이용하여 콘텐츠를 암호화함으로써 콘텐츠의 이동을 크게 제한한 반면 Superdistribution은 콘텐츠의 이동 및 복사는 자유롭게 허락하되 사용 시점에서 사용권한을 획득해야만 콘텐츠의 이용이 가능한 기술적 메커니즘을 통해 디지털 콘텐츠의 안전한 유통을 유도하게 한다.

### 2) OMA DRM v2.0 플랫폼

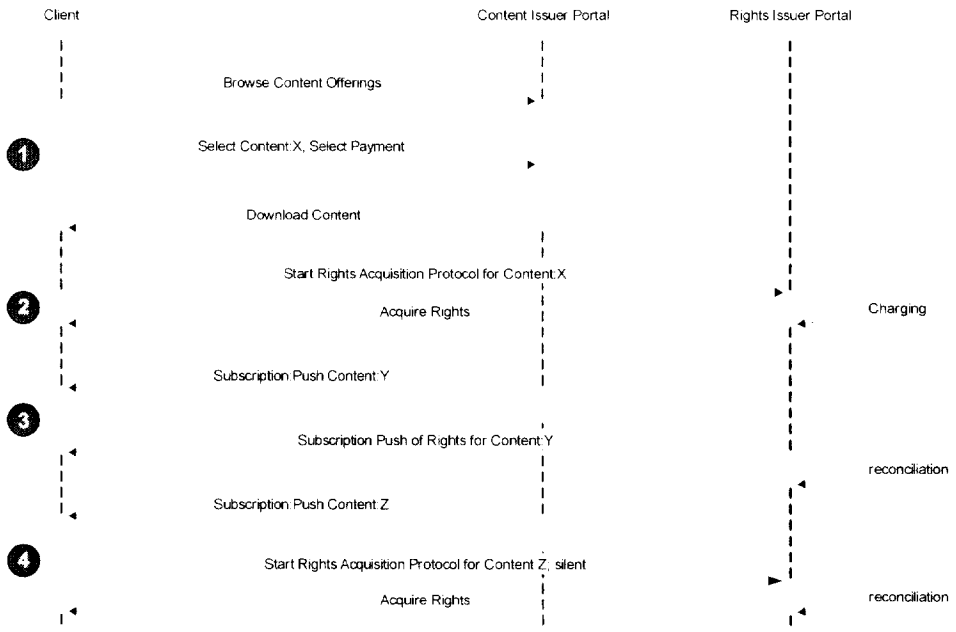
OMA DRM v1.0에서와는 달리 OMA DRM v2.0에서는 키 전달과 인증에 관해서 공개키 기반구조(PKI)를 기반으로 하여 서버와 디바이스 간의 인증, 디바이스 간의 인증, B2B 거래 주체들 간의 인증 등 DRM에 참여하는 주요 거래 주체들끼리의 인증기술로 PKI 인증서를 기본적으로 사용하도록 하고 권리 및 사용규칙 언어로 XML을 기본적으로 사용한다. 주요 거래 주체를 디바이스, 콘텐츠 발급자, 권한 발급자, 인증기관(CA)를 대상으로 하고 있다. 주요 거래 주체들 중에서 권한 발급자와 디바이스 간에는 ROAP(the Rights Object Acquisition Protocol)를 이용하여 공개키 기반의 상호인증 및 키 합의가 이루어지고 합의 후에 권한 발급자가 디바이스에게 암호화된 콘텐츠에 대한 디바이스의 권한인 권한 객체(Rights Object, RO)를 합의된 키로 암호화해서 전달하게 된다. 그리고 이러한 인증과 키 전달을 만족하기 위해서 디바이스 상에서의 보안기능은 디바이스 내에 저장될 개인키의 안전한 보호 기술과 RO에 규정한 권한을 바탕으로

암호화된 콘텐츠 복호화 기능 및 데이터 보관 기능이 필요하다. 그리고 권한 발급자는 RO 생성 및 전달, 디바이스에게 인증 체인 전달기능을 필요로 한다. 2.0에서 새롭게 추가된 주체인 인증기관은 인증서 생성과 OCSP 운영, 그리고 인증 정책을 제공해야 한다. 그리고 콘텐츠 발급자는 디바이스에게 암호화된 콘텐츠를 제공해 주도록 정의한다. OMA DRM v2.0에서 제안하는 기본 구조는 그림 3과 같다.[6].



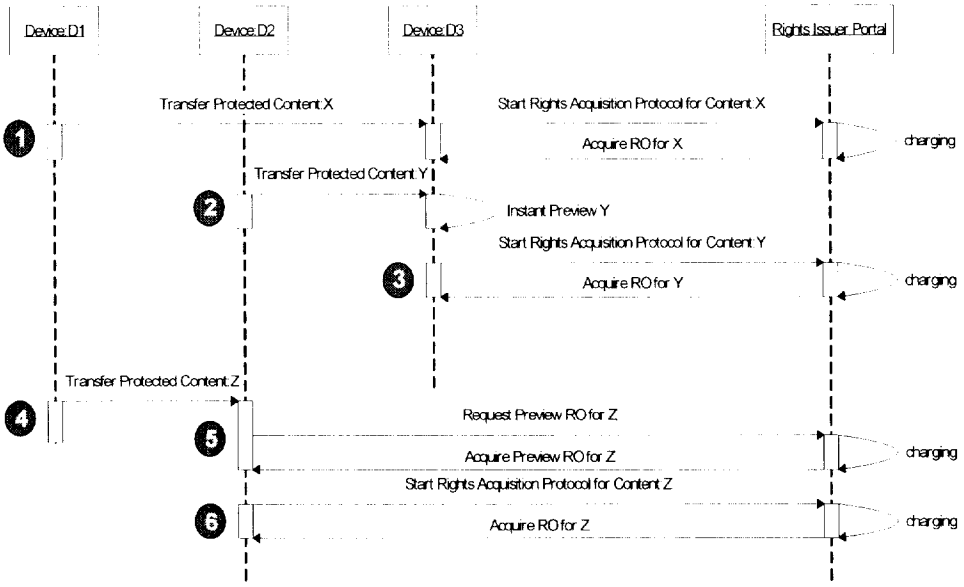

[그림 3] OMA DRM v2.0의 기본 구조

OMA DRM v2.0에서는 인증에 관한 프로토콜로써 ROAP를 정의하고 그에 맞는 서비스 모델을 콘텐츠 다운로드, Superdistribution, 미디어 다운로드, Domain 및 Export기능으로 각각 정의하고 있다. 콘텐츠 다운로드 모델은 사용자의 디바이스가 암호화된 콘텐츠를 받은 후 콘텐츠 헤더내 권리 발급자 URI를 보고 권리 발급자와 통신하여 권한 객체(RO)를 전송받게 된다.[7].



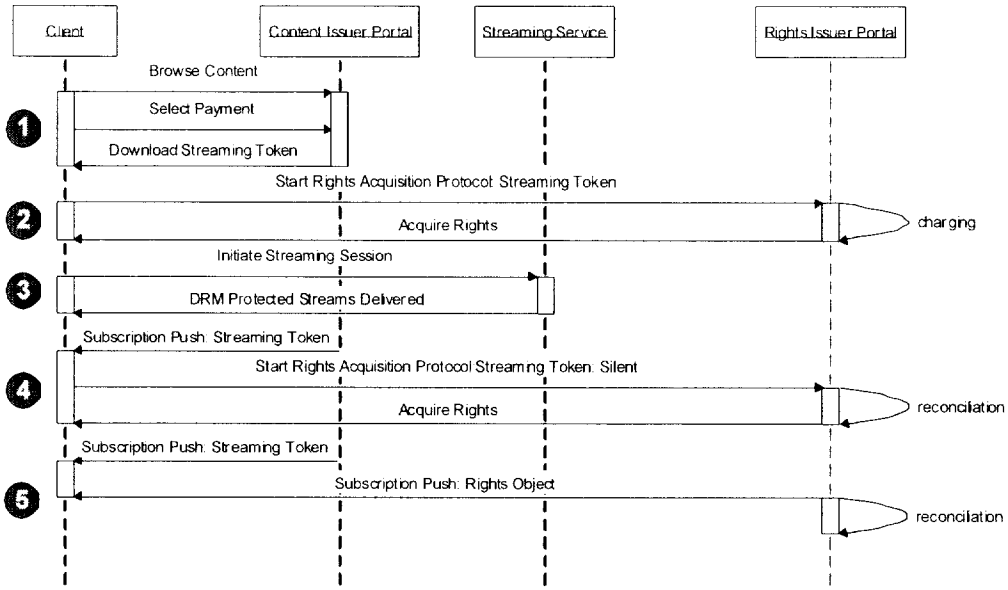

[그림 4] Basic Download - Pull and Push Models

그림 4는 OMA DRM v2.0에서의 기본적인 콘텐츠 다운로드 방법을 나타낸다. 사용자의 권한 신청은 정기적인 간격으로 콘텐츠와 사용 권리를 받거나, 콘텐츠만 제공받고 이후 권리 발행자에게 권한 객체를 받는 두 가지 모델로 구분된다. 그리고 Superdistribution 모델은 콘텐츠를 받은 디바이스가 다양한 경로를 통해서 다른 디바이스에게 직접 콘텐츠를 전달하고, 전달받은 다른 디바이스는 권한 발급자에게 권한 객체(RO)를 요청하여 콘텐츠를 이용하게 된다. 이때 전달되는 콘텐츠는 암호화 되어 전달하게 된다. 그림 5는 사용자의 디바이스 간에서 재분배되는 과정을 나타내고 있다.



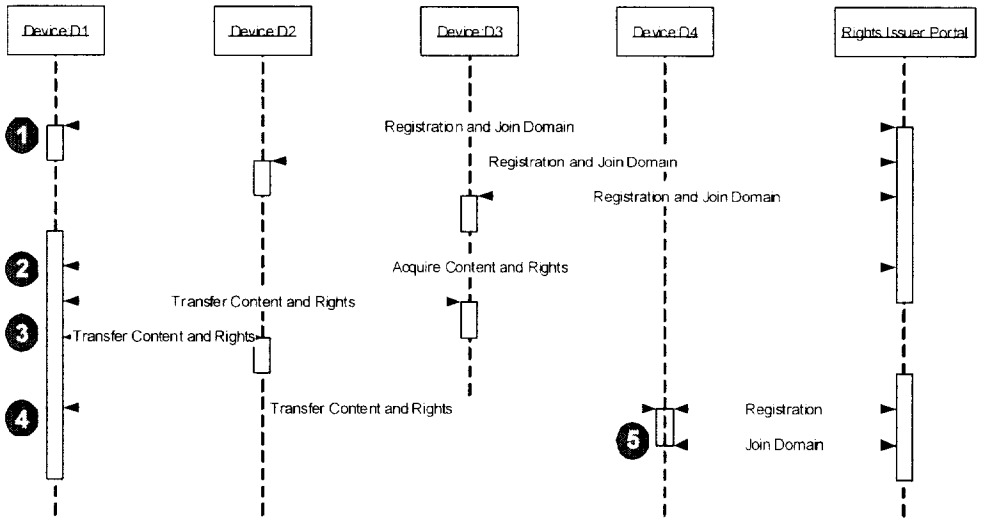

[그림 5] Superdistribution에 의한 콘텐츠 재분배 과정

미디어 다운로드 모델은 사용자의 단말기에서 콘텐츠 발급자에게 미디어의 위치, 특징, 설정 등을 나타내는 일련의 데이터 정보인 Streaming Token을 다운로드 받고, 권한 발급자에게는 RO를 다운로드 받는다. 다운로드 된 RO에는 복호화를 위한 암호키 뿐만 아니라 미디어 디코딩 방법과 표현방법 등이 포함되어 있다. Push 서비스일 경우에는 RO와 Streaming Token을 정기적인 간격으로 제공받게 된다. 주요 서비스 과정은 그림 6과 같다.




[그림 6] OMA DRM이 적용된 Streaming 서비스 과정

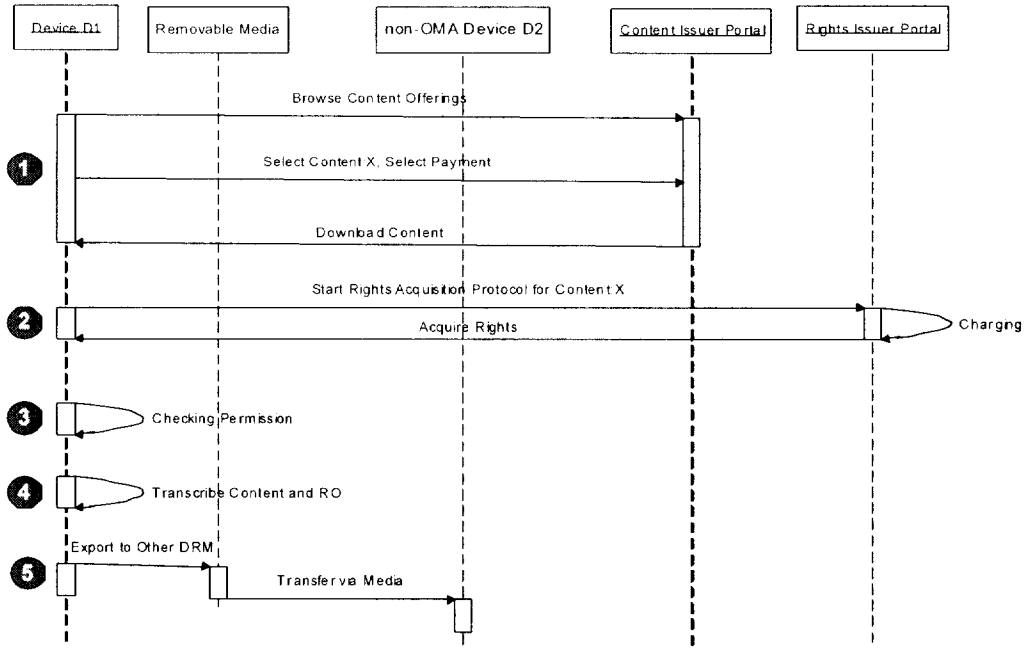
그리고 1.0에서는 제공되지 않았던 동일 사용자의 복수개 단말기 환경을 지원하도록 Domain 서비스 모델을 정의하고 있다. Domain은 정당한 사용자가 소유한 단말기의 집합을 나타내며 이들 상에서는 콘텐츠의 허가된 사용이 가능하도록 한다. 이때 사용자는 자신의 복수개의 단말기를 권한 발급자에게 등록, 추가 가능하다. 다음 그림 7은 동일 사용자의 복수개의 단말기에서 콘텐츠의 사용과 새로운 단말기의 등록절차를 나타내고 있다.




[그림 7] Domains 서비스 과정

또 다른 추가기능으로 Export 기능이 있다. 이것은 OMA DRM 기능을 제공하지 않는 단말기 상에서 OMA DRM 적용된 콘텐츠를 이용하고자 할 때 제공되는 기능이다. 사용자는 콘텐츠에 대한 권한을 권한 발급자로부터 제공받아 사용할 수 있으며, 자신의 제공받은 DRM 시스템을 적용하지 않은 단말기에서 이용하기 위해서 권한 발급자로부터 발급받은 RO를 체크하고, 콘텐츠와 RO를 해당 단말기에 맞게 변환시킨다. 이때 변환 과정은 다른 DRM 시스템의 규칙에 맞게 콘텐츠 및 RO의 폼을 변경시켜 상호호환성을 가지도록 한다.

다음 그림 8은 Export 모델의 서비스 과정을 나타내고 있다.




[그림 8] 서로 다른 DRM 시스템을 적용한 단말기 간의 이용 과정

국내, 외 업체를 중심으로 OMA에서 제정한 DRM v2.0을 수용하고자 활발한 움직임이 일어나고 있다. 이와 함께 PKI를 토대로 무선 인터넷에서의 콘텐츠를 보호하는 DRM v2.0의 구현을 위해서 이에 적합한 공인인증 서비스 모델 구현을 통해 국내 무선 인증 서비스의 활성화 방안이 마련될 것으로 예상되어 진다. OMA DRM v2.0은 PKI 기반의 디바이스 인증을 구현하고 있기 때문에 디바이스 인증서 프로파일에 대한 고려가 필요하고, 콘텐츠의 권한을 보다 효과적으로 다루기 위하여 PMI(Privilege Management Infrastructure) 적용 부분도 함께 검토되어야 한다. 그리고 DRM v2.0에서 언급되는 모든 스펙들은 XML로 구현되고, 전자서명 시 XML언어를 사용하기 때문에 이를 수용할 수 있는 기술도 함께 검토되어야 할 것이다.

## 2.2 WIPI 플랫폼 및 보안 요구사항

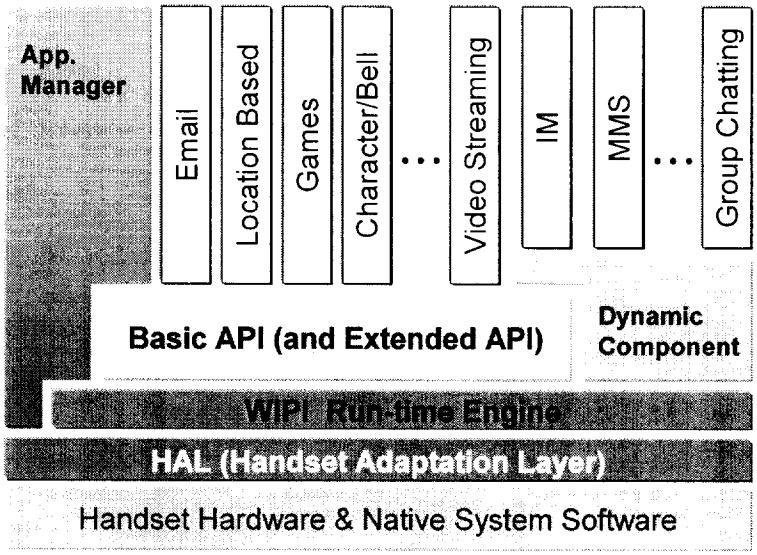
국내 무선인터넷의 발전은 서로 다른 무선 응용 프로토콜 방식(WAP,ME)를 채택하고 , 각 사업자들의 CP를 위한 개발 환경도 상이하여 무선인터넷 활성화의 걸림돌이 되어 왔다. 특히 무선 인터넷의 이동통신 사업자들이 본격적인 무선인터넷 서비스 시장을 형성하기 위한 기반 인프라로 CDMA2000 1x, W-CDMA의 도입이 진행되어 왔고 본격적인 무선인터넷 시장의 표준화가 추진되어 왔다. 낮은 수준의 전송 특성만을 제공하던 2세대 이동통신 기반에서 고속 패킷 전송의 제공이 가능한 패킷 방식의 2.5세대, 3세대 인프라가 구축됨에 따라 기존의 텍스트 위주의 무선인터넷에서 멀티미디어 서비스가 가능한 형태로 발전하게 되었고 이에 대한 표준화 대응이 절실히 요구되어 왔다. 서로 상이한 방식의 무선 인터넷 서비스로 CP와 단말기 제조업체의 경우 이동통신 사업자의 각기 다른 서비스 제공을 위해 연구 및 생산기능을 중복으로 투자해야 하는 비효율성이 발생하게 되었고 이러한 원인을 해결하기 위해 표준화 작업을 시작하게 되었다.[9].

현재 국내 무선 인터넷 기반 모바일 플랫폼은 이동통신 사업자 별로 표1과 같이 서로 다른 대표적인 플랫폼을 지원하고 있다.

[표 1] 국내 무선 인터넷 기반 모바일 플랫폼 현황

| | 서비스 사업자 | 개발환경 | 개발사 | 수행방법 |
|---|---|---|---|---|
| KVM | LGT | JAVA | SUN | 인터프리터 |
| 키터호크 | LGT | JAVA | 아로마소프트 | 인터프리터 |
| MAP | KTF | C/C++ | 모빈탑 | 바이너리 |
| BREW | KTF | C/C++ | 퀄컴 | 바이너리 |
| GVM | SKT | C/C++ | 신지소프트 | 인터프리터 |
| XVM | SKT | JAVA | XCE | 인터프리터 |
| WITOP | SKT | JAVA,C/C++ | SKT | 인터프리터 |

서로 다른 플랫폼을 채택하기 때문에 개발 환경도 모두 달라 플랫폼간의 상호 운영성을 보장할 수가 없고 이전 플랫폼의 문제점을 해결하기 위해 무선 인터넷 표준 플랫폼 WIPI가 개발되었다. WIPI는 이동 통신 단말기에 탑재되어 응용 프로그램을 수행할 수 있는 환경을 제공하는 모바일 표준 플랫폼 규격을 정의 한다.[8]. WIPI는 플랫폼 이식성을 높이기 위한 표준화된 하드웨어 추상화 계층인 HAL(Handset Adaptation Layer)과 표준화된 플랫폼 호환성을 제공하여 다양한 응용프로그램 개발을 촉진하기 위한 기본 응용 프로그래밍 인터페이스(Basic API)로 구성된다. WIPI 규격을 만족하는 모바일 플랫폼은 단말기용 응용 프로그램 개발자에게는 플랫폼 간 콘텐츠 호환성을 보장하고, 단말기 개발자에게는 플랫폼의 이식의 용이성을 제공하며, 일반 이용자에게 다양하고 풍부한 콘텐츠 서비스의 제공을 목적으로 하고 있다.




[그림 9] WIPI의 구조

그림 9는 WIPI의 구조와 주요 모듈을 나타낸다. 여기서 Application Manager는 응용프로그램의 다운로드, 설치, 삭제 등의 응용프로그램 관리와 API 및 component들을 추가, 갱신하는 기능을 수행한다. Dynamic Linking Library로서 WIPI의 일부를 구성한다. HAL은 플랫폼의 하드웨어 독립성을 유지하기 위한 추상화 계층으로 상위 Layer들은 HAL 위에서 Native System과 무관하게 동작하도록 지원한다. HAL을 통해서 단말기에 대한 추상화가 이루어지고, 하드웨어 독립적으로 플랫폼을 구성할 수 있게 된다. Dynamic component는 Application Manager를 통하여 추가 혹은 갱신된 API 및 component들의 집합을 나타낸다. 그리고 Basic API와 Extended API는 응용프로그램 개발자를 위한 API로서 C언어를 지원하는 Clet과 JAVA를 지원하는 Jlet을 가지고 있다. C언어를 지원하는 Clet의 경우 기본 API에 정의된 C언어 API와 C언어 문맥을 지원하고 표준 C라이브러리 중에서 주요 함수를 필수적으로 지원하도록

하고 있다. 그리고 자바를 지원하는 Jlet의 경우 기본 API에 정의된 자바 언어용 API와 자바 언어 문맥을 지원하도록 하고 속도가 느린 자바의 단점을 보완하기 위해 바이너리 플랫폼을 제공할 수 있도록 하였다.

C언어를 지원하는 Clet과 달리 자바언어를 지원하는 Jlet의 경우 좀 더 빠른 수행을 위해 그림10과 같이 바이너리 코드 형태로 되어있는 자바 어플리케이션을 수행되기 전에 미리 컴파일 해서 단말기의 CPU의 최적화된 바이너리 코드를 생산하는 AOTC(Ahead Of Time Compile) 방식을 이용하여 수행 속도를 빠르게 하였다. 그리고 기존의 자바 언어가 가지는 자바 바이트 코드 검증을 통해 정상적으로 수행 될 수 있는 어플리케이션인지를 수행 전에 미리 확인할 수 있도록 COD(Compile On Demand) 방식을 이용하여 악의적인 어플리케이션의 공격을 차단할 수 있도록 한다.

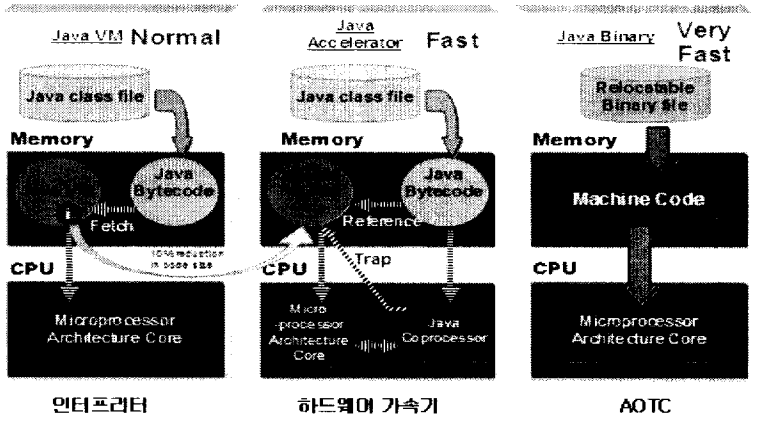



**그림 2 성능향상 방식**

[그림 10] 성능 향상 방식

WIPI 규격에 따른 플랫폼에서는 인터넷상에서와 같이 바이트 검증을 통한 부분은 없지만 COD 방식을 이용한 서버를 이용하여 검사할 수 있는 기능을 부여 하게 되었다. COD 방식이란 컴파일을 요구 할 때 컴파일이 된다는 개념을 가지고 핸드폰 단말기의 경우 여러 종류의 CPU 칩으로 서비스가 될 경우 각각의 CPU에 맞도록 서버에서 컴파일 된다는 의미를 가지고 있다. WIPI 플랫폼에서의 자바 수행 환경은 그림11과 같다.

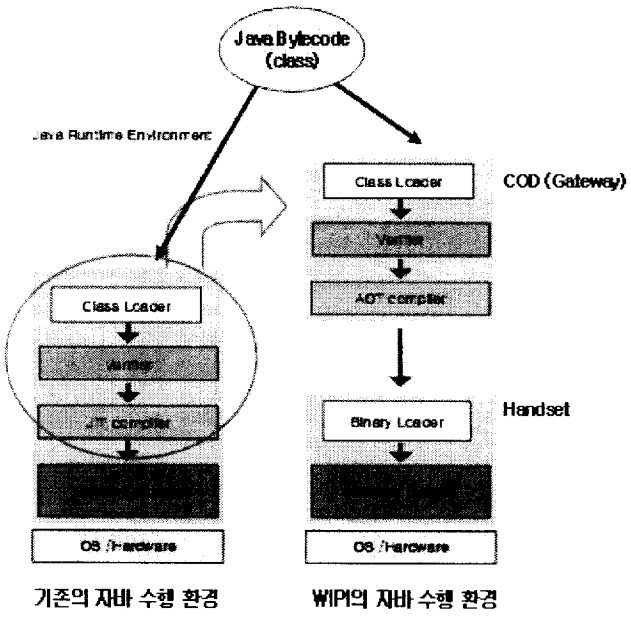



**그림 3 WIPI의 자바수행환경**

[그림 11] WIPI의 자바수행환경

WIPI 플랫폼에서의 보안은 한 응용프로그램이 플랫폼의 API나 저장공간, 공유메모리 등의 자원을 통해서 다른 응용프로그램 혹은 다른 단말기 혹은 단말기가 연동하고 있는 네트워크/서버 시스템 등의 엔티티가 동작에 영향을 끼치거나, 다른 엔티티가 소유하고 있는 정보에 접근하는

경우에 대한 정책적, 기술적 대책을 의미한다. 실질적인 보안 기법은 기본적으로 플랫폼의 각 자원별 정의된 보안 수준별 접근 허용여부와 응용프로그램의 보안수준에 대한 정보를 바탕으로 수행된다. 따라서 지원하는 보안수준은 표2와 같이 3단계의 보안수준을 지원하고 있다.

[표 2] 플랫폼 보안 수준

| | |
|---|---|
| PUBLIC | 플랫폼에 존재하는 보안 레벨 중 가장 신뢰할 수 없는 수준, 혹은 가장 제약사항이 많은 수준을 뜻한다. 따라서 이 수준에서는 단말기에 영향을 줄 수 있거나, 개인정보에 접근하는 등, 보안 문제를 야기할 소지가 있는 플랫폼 자원에 대한 접근을 허용하지 않도록 한다. |
| CP(Content Provider) | 이미 알려진 CP 들은 어느 정도 신뢰 할 수 있다고 보고, 단말기에 심각한 영향을 미치지 않는 범위 내에서 접근을 허용한다. |
| SYSTEM | 완전히 신뢰할 수 있는 것으로 보고, 모든 접근을 허용한다. |

표2와 같은 보안수준을 적용하는 보안 정책은 응용프로그램 별로 정의된 보안 관련 정보와 플랫폼의 자원별로 정의된 접근 제어 조건에 따라 자원접근을 제어 하게 되고 자원의 종류에는 표3과 같이 최소한의 그룹이 포함되어 각 그룹별로 제어 가능해야 한다.

그리고, 구현 시 실제 자원에 대한 접근은 HAL API와 Basic API를 통해 리소스에 접근하게 되고 3단계 보안 수준 안에서 API별로 접근 수준을 설정하는 API 보안을 정의하고 자원 중 특히 저장 공간에 접근하는 경우에도 3단계 보안 수준 안에서 Data 접근 수준을 설정하는 Data 보안을 정의한다. Data보안은 표3에서 Storage 부분의 종류와 같이 3가지 부분으로 정의하고 Private Directory는 서로 자신만의 Data 영역을 만들고 그 영역은 다른 응용프로그램이 접근 할 수 없도록 설정하고

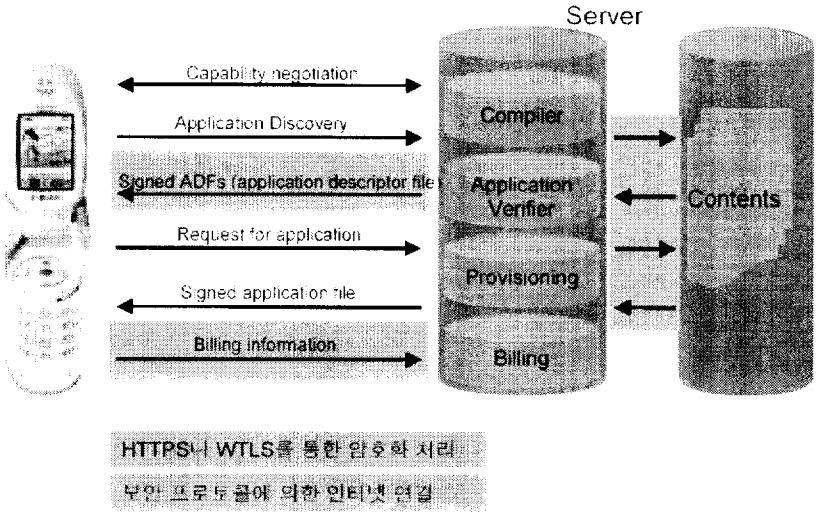
Application Shared Directory는 미리 약정된 응용프로그램들은 특정 응용프로그램에서 할당 받아 사용하는 Data 영역에 접근 할 수 있도록 설정한다. 그리고 마지막으로 System Shared Directory는 접근 권한을 갖는 모든 응용프로그램들이 접근할 수 있는 Data 영역을 설정해야 한다.

[표 3] 보안 정책의 대상 자원

| 구 분 | 종 류 |
|---|---|
| Storage | Private directory / Application Shared Directory / System Shared Directory |
| Network | Connection Oriented, Datagram, HTTP |
| Secured Network connection | HTTPS |
| Serial Device | RS-232C, USB |
| Personal Information | 주소록, 촬영한 사진, 기타 개인 정보 |

WIPI에서 서비스 보안은 자바 언어의 특성상 개발된 응용프로그램들의 코드레벨의 안정성 및 보안성을 확인해야 하기 때문에 응용프로그램 인증이 필요하다. 인증에 관련된 내용은 응용프로그램의 Descriptor File 및 실행 코드에 서명 하도록 한다.

그림 12는 WIPI의 서비스 개념도를 나타낸다. 그림에서 보면 자바의 경우는 인터넷의 콘텐츠는 모두 바이트 코드형태로 되어있고, COD를 거쳐 플랫폼으로 전달될 때 바이너리 이미지로 변경된다. 따라서 WIPI 플랫폼 상의 자바 어플리케이션은 신뢰할 수 있는 COD 서버가 바이트코드 검증을 통해서 플랫폼으로 전달이 되고, 플랫폼은 이것을 확인하게 되므로, 기존의 자바 수행 환경에서 요구되는 바이트코드 검증을 동일하게 거치게 되어 안전하게 실행할 수 있는 환경을 제공한다.



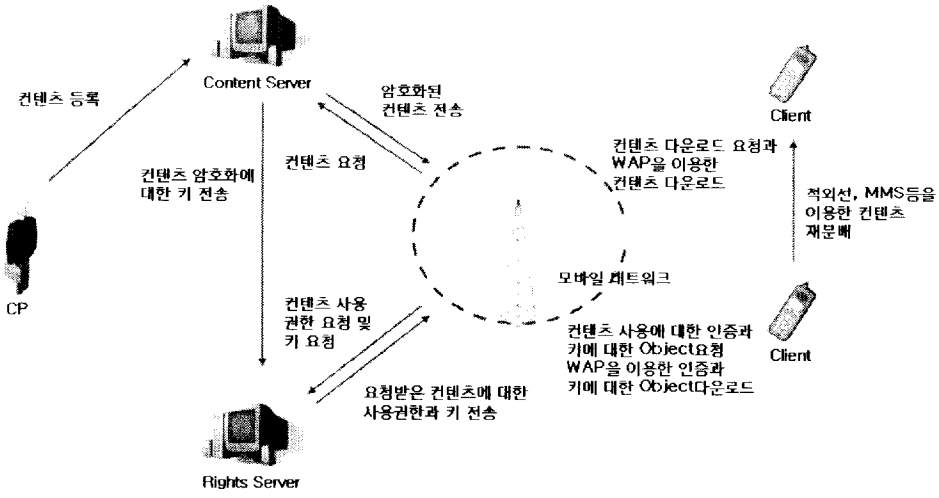

[그림 12] WIPI 서비스 개념도

# III.WIPI 플랫폼에서의 DRM 시스템 설계 및 구현

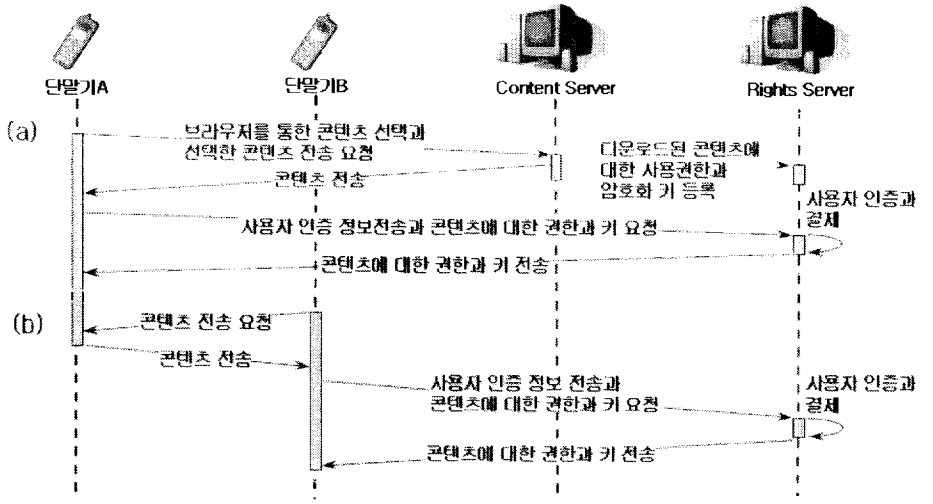
## 3.1 WIPI 플랫폼에서의 DRM 시스템 아키텍처 설계

### 1) 시스템의 개요

구현 하고자 하는 DRM 시스템의 모형은 WIPI 플랫폼 상에서 디지털 콘텐츠를 실행 할 수 있는 단말기와 디지털 콘텐츠를 안전하게 단말기로 다운로드 형태로 전송 할 수 있는 콘텐츠 서버, 사용권한에 대한 정보를 단말기로 전송할 수 있는 권한 서버로 크게 세부분으로 나눌 수 있다. 콘텐츠 제공자가 콘텐츠를 배포하고자 할 때 콘텐츠 서버에 등록을 시키고 콘텐츠 서버에서는 약속된 방법으로 콘텐츠를 패키징해서 DRM 콘텐츠형태로 변환시키게 된다. 그리고 약속된 방법에 의해서 패키징을 할 때 사용되는 암호키를 권한 서버에 등록한다. 이러한 시스템의 개념적인 모형은 그림 13과 같다.




[그림 13] DRM 시스템의 개념적 구조




[그림 14] Use Case에 따른 프로토콜

그림 13에서 콘텐츠의 전달과정에 따라 두 가지 형태의 Use Case를 나타낼 수 있다. 첫 번째, 단말기가 콘텐츠 서버로부터 DRM 콘텐츠를 직접 다운로드 받고 실행하고자 할 때 권한 서버에게 인증절차를 거쳐 권

한객체를 전송받아 콘텐츠를 사용할 수 있도록 하는 것과 두 번째, 단말기간의 P2P 형태의 다운로드 방식을 이용하여 콘텐츠를 다운로드 받고 실행하고자 할 때 권한 서버에게 인증절차를 거쳐 권한객체를 전송받아 콘텐츠를 사용하도록 하는 방법이다. 두 가지 Use Case에 대해 콘텐츠와 권한객체를 안전하게 전송할 수 있도록 그림 14와 같이 프로토콜을 정의한다.

그림 14에서 볼 수 있듯이 사용자가 다운받는 콘텐츠는 암호화되어 전송되게 되고 그 암호화 키는 콘텐츠에 관한 정보와 함께 권한 서버로 보내지게 된다. 따라서 DRM 콘텐츠의 포맷을 정의할 때 반드시 콘텐츠 ID를 헤더 부분에 포함하여야 하고 권한 서버의 URL 정보도 헤더에 포함하여야 한다.

서버에서 구현 되어야 할 모듈은 ASP 또는 JSP등의 웹을 통하거나 또는 특정 소켓을 통하여 단말기로부터 전송되는 정보를 처리할 수 있는 형태가 되어야 한다.

각각의 서버와 단말기가 가져야 할 세부 기능은 다음과 같다.

■콘텐츠 서버

■단말기와 HTTPS, TCP/IP, WAP 통신이 가능하도록 서버측 소켓통신의 기능.

■콘텐츠를 등록하는 일련의 절차 즉, 콘텐츠 패키징 기능.

■패키징 하는 과정에서 원본 콘텐츠를 암호화 시켜 보호할 수 있는 기능.

■권한 서버

■단말기와 HTTPS, TCP/IP, WAP 통신이 가능하도록 서버측 소켓통신의 기능.

■인증 내용에 따라 권한 객체를 생성하여 단말기로 전송하는 기능.

■결제 또는 외부 인증 모듈과 상호 연동할 수 있는 기능.

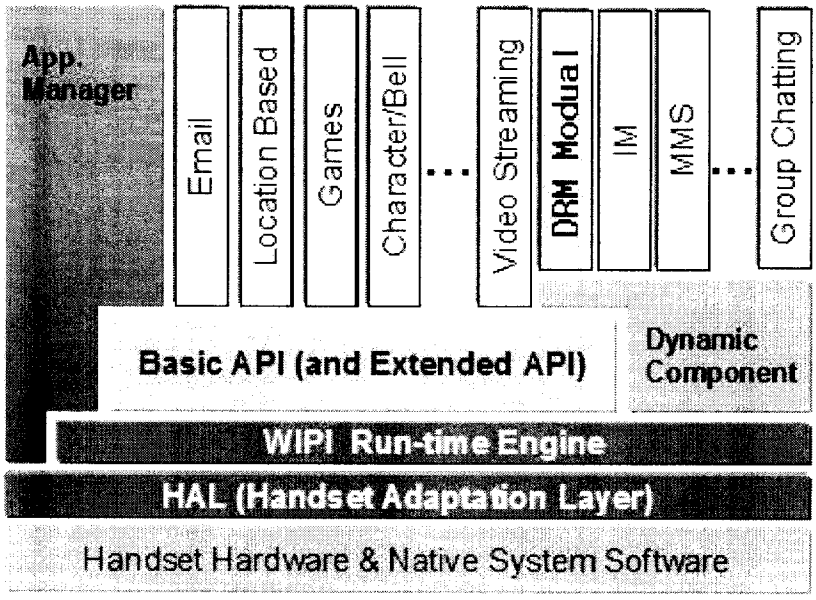
■단말기

■각각의 서버에게 콘텐츠 요청 또는 권한 객체의 요청을 위해 통신할 수 있는 소켓 통신 기능.

■암호화된 콘텐츠를 복호화 하여 실행할 수 있는 기능.

■단말기 간 통신을 이용하여 콘텐츠를 전송할 수 있도록 단말기 간 통신 시 데이터를 보내줄 수 있는 서버의 기능.

그림 13과 같은 시스템 구조와 그림 14와 같은 Use Case에 따른 프로토콜에 맞추어 WIPI 플랫폼에서 구현하기 위해서는 단말기에서 구현되어야할 형태는 HAL과 Basic API 외에 서버와의 통신과 DRM 콘텐츠를 관리 하는 API가 추가되어야 한다. 추가된 API는 DRM 콘텐츠 형태로 전달되는 콘텐츠를 제어하는 기능을 가진다.



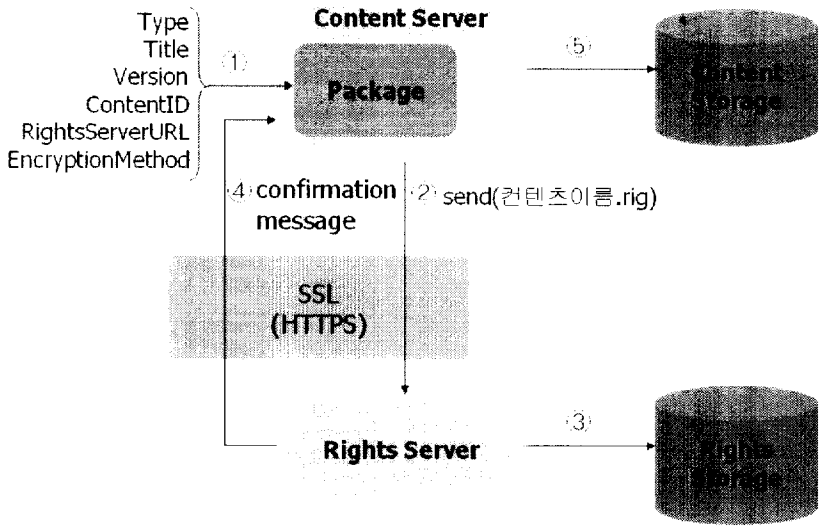

[그림 15] WIPI 기반의 DRM 컴포넌트 API의 개념적 구조

그림 15는 WIPI의 HAL과 Basic API와 마찬가지로 하나의 API 형태로 추가 되어 WIPI 플랫폼에서 DRM Content Control API를 지원하는 DRM 컴포넌트 구조를 나타내고 있다.

그리고 DRM 컴포넌트는 WIPI가 지원하는 특징 중의 하나인 DLL형태로 정의되어 네트워크를 통한 DLL 다운로드 형식으로 Dynamic Component Manager에 의해 관리되어 업데이트 가능하도록 되어있다.

### 2) 콘텐츠 암호화와 클라이언트 모듈 설계

전체 시스템 구성도에서 콘텐츠를 등록하는 과정 중에서 콘텐츠 암호화 부분은 콘텐츠 서버와 권한 서버에서 함께 수행 되는 구조로 설계 하였다. 콘텐츠 서버에서는 콘텐츠 제공자가 제공하는 콘텐츠를 암호화 과정을 거쳐 패키징 하여 암호화 콘텐츠를 만들어 낸다. 전체 서버에서의 과정은 그림 16과 같다.




[그림 16] 콘텐츠 서버에서 암호화 과정

■ 콘텐츠 서버는 DRM Package Header에 포함될 기본 정보인 Type, Title, Version, Content ID, RightsServer URL, Encryption Method를 입력받고 서버 측 모듈인 Package에서 생성한 암호화키를 이용하여 그림 17과 같은 초기 권한 파일을 생성한다.

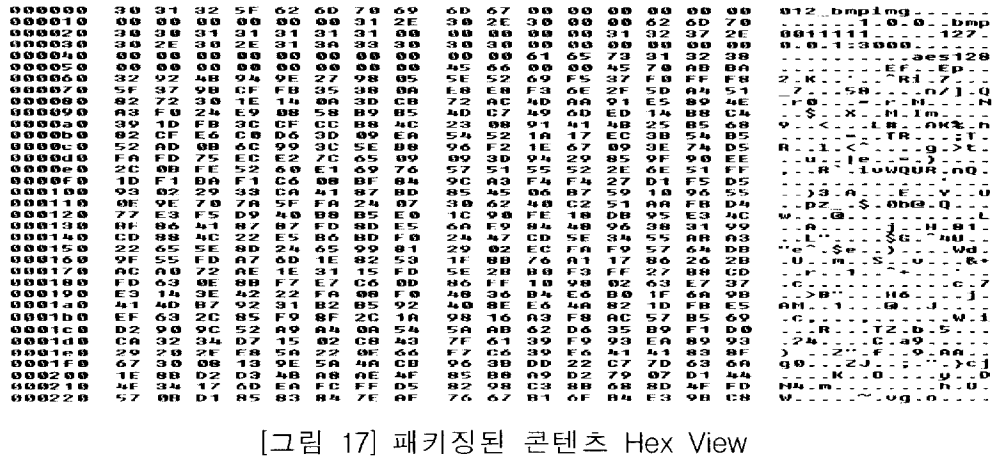
■ 생성된 초기 권한 파일을 권한 서버에 전송하여 등록한다. 이때 권한 서버와의 통신은 SSL을 이용한 https등을 이용하여 안전하게 전송하도록 한다.

■ 권한 서버에서는 전송받은 권한 파일을 권한 저장소에 저장하여 등록하게 된다.

■ 권한 서버는 권한 파일의 등록이 완료되면 콘텐츠 서버에게 확인 메시지를 전송하게 된다.

■ 초기 권한 파일이 등록됨을 확인하고 나면 콘텐츠 서버는 ■단계에서 입력받은 콘텐츠 헤더 정보를 이용하여 패키지된 콘텐츠를 콘텐츠 저장소에 등록한다.

이와 같은 순서로 콘텐츠를 패키징하고 생성한 파일은 그림 17과 같은 형태를 보여 준다.



[그림 17] 패키징된 콘텐츠 Hex View

그림 17에서 첫 번째 바이트에서 살펴보면 콘텐츠의 기본정보를 수록하여 헤더정보를 입력하고 원래 콘텐츠를 암호화하여 추가한 Hex Code를 볼 수 있다. 그리고 함께 생성될 권한 파일은 모바일 환경에 맞추어 용량이 크지 않게 바이트 코드 형식의 이진 파일로 작성되고 권한 파일의 필드는 표 4와 같이 구성된다.

[표 4] 권한 파일의 필드 정보

| 필드 명 | 필드 설명 | byte 크기 |
|---|---|---|
| USECASE | 콘텐츠 사용유형 | 3byte |
| USETERM | 콘텐츠 사용기간 | 12byte |
| USERIGHTS | 콘텐츠 사용권한 | 2byte |
| USEKEY | 콘텐츠 복호화 키 | 16byte |

표 4와 같이 정의된 필드에 따라 초기 생성되는 권한 파일의 Hex Code는 그림 18과 같다.

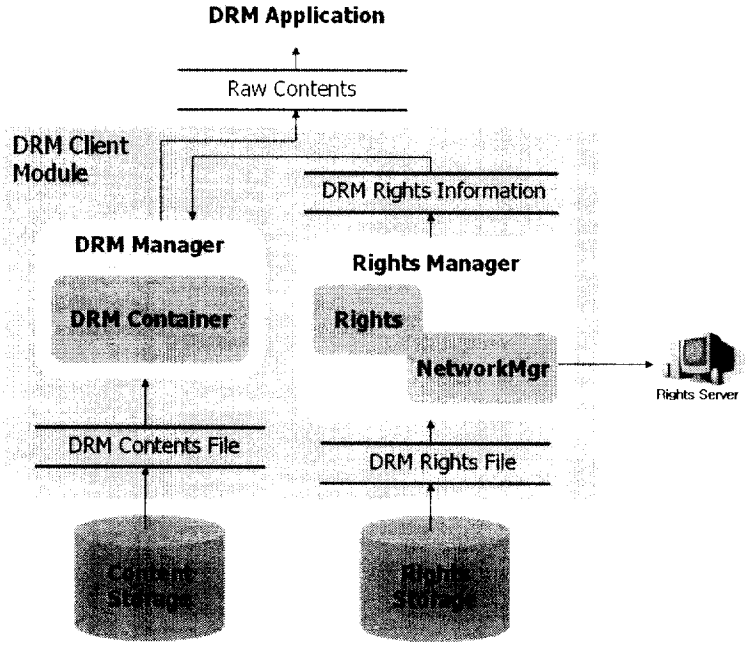
```
000000  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000010  00 E6 2B 35 13 C1 02 9D  4D CC 18 EF 2F E9 DA A1  ..+5....M.../...
000020  C3                                                .
```

[그림 18] 최초 생성된 권한파일 Hex View

최초 생성된 권한 파일은 사용유형, 사용기간, 사용권한에 대한 정의가 없기 때문에 암호화된 키만 가지고 생성이 된다. 어떤 사용자가 콘텐츠를 사용하고자 할 경우 권한을 인증 받는 내용을 전송 받아 권한 서버에서 사용자에 대한 권한 파일을 업데이트 하게 된다.

서버에서 패키징된 콘텐츠를 다운로드 받아 클라이언트에서 실행하고자

할때는 그림 19와 같은 클라이언트의 구조를 가지게 된다. 주요기능은 DRM 콘텐츠에 대한 관리를 하게 될 DRM Manager부분과 콘텐츠 사용 권한에 대한 부분을 관리하게 될 Rights Manager부분으로 나눌 수 있다.




[그림 19] DRM client 모듈 구성

콘텐츠 저장소에서부터 실행하고자 하는 파일을 선택하여 실행할 때는 콘텐츠 파일을 DRM Manager에 로드하고 콘텐츠에 대한 권한 파일을 권한저장소에서 검색하여 권한 사항에 대한 내용을 불러와서 사용하도록 설계하였다. 만약 권한 파일이 없을 경우에는 권한 서버에 접속하여 권한 내역에 대해 등록 하고 업데이트된 권한 파일을 전송받도록 한다. 권한 내역을 입력하여 업데이트된 권한 파일은 그림 20과 같이 사용 권한

에 대한 내역을 수록하고 복호화 키를 함께 저장하고 있다.

```
000000  30 37 75 31 30 32 30 30  34 2E 31 32 2E 32 39 31  07u102004.12.291
000010  34 E6 2B 35 13 C1 02 9D  4D CC 18 EF 2F E9 DA A1  4.+5....M.../...
000020  C3                                                .
```

[그림 20] 콘텐츠 실행시 권한을 인증한 권한파일 Hex View

권한이 설정된 권한 파일로부터 권한 정보를 읽어 DRM Manager에 보내주어 해당 콘텐츠를 사용할 수 있는 권한을 확인 한 후에 DRM 응용프로그램에 복호화 된 원본 콘텐츠의 데이터를 보내주어 실행하게 된다.

## 3.2 DRM 컴포넌트 구현

시스템 구현은 WIPI API 중 JAVA API를 이용하여 전체 시스템을 구현하였다. 그림 15와 같은 DRM 컴포넌트로써 DRM 콘텐츠를 관리 하는 DRMContainer class와 네트워크 통신을 관리하는 NetworkMgr class와 권한 객체를 관리하는 Rights class로 구성되어 진다. DRM 컴포넌트는 핸드폰 단말기상의 응용프로그램에서 사용되어 진다.

DRMContainer class는 WIPI 단말기에서 패키지한 DRM 콘텐츠를 관리하기 위해서 헤더 정보의 구조를 나타내는 DRMContentHeader class와 , 실제 콘텐츠 자료를 나타내는 DRMContentObject class로 정의한다. 정의 부분은 다음과 같다.

```
public class DRMContentHeader{
        //헤더 각 필드의 데이터 입력 순번
        final static int TYPE = 0;
        final static int TITLE = 1;
        final static int VERSION = 2;
        final static int CONTENTID = 3;
        final static int RIGHTSSERVERURL = 4;
        final static int ENCRYPTIONMETHOD = 5;
        final static int PLAINDATALENGTH = 6;
//헤더 각 필드의 크기
        final static int TYPE_SIZE = 2;
        final static int TITLE_SIZE = 20;
        final static int VERSION_SIZE = 7;
        final static int CONTENTID_SIZE = 15;
        final static int RIGHTSSERVERURL_SIZE = 30;
        final static int ENCRYPTIONMETHOD_SIZE = 12;
        final static int PLAINDATALENGTH_SIZE = 4;

        int sizes[] = {TYPE_SIZE, TITLE_SIZE, VERSION_SIZE,
                CONTENTID_SIZE, RIGHTSSERVERURL_SIZE,
               ENCRYPTIONMETHOD_SIZE, PLAINDATALENGTH_SIZE};
        final static int FieldCnt = 7; //헤더 필드의 개수
        byte[][] m_fields; //헤더 정보를 담을 바이트 배열
        public DRMContentHeader() {  }
        private synchronized void setHeader(byte[] headerBuf) {  }
        private byte[] getHeader(int index) {  }
}
```

헤더 정보를 네트워크를 통해 전송 받을 때 바이트 단위로 전송할 수 있도록 각각의 헤더 필드의 크기를 정하고 그에 따라 바이트 배열로 선언 되어 있다. 헤더 정보를 네트워크를 통해 전송 받아 헤더 정보를 저장하는 setHeader메서드와 헤더의 정보를 가져 나오는 getHeader메서드를 정의한다.

```
public class DRMContentObject{
        byte[] DRMDataLength;
        byte[] DRMData;
        public DRMContentObject() {  }
        private synchronized void setContent(byte[] contentBuf) {  }
        private byte[] getContent() {  }
        private int getLength() {  }
}
```

DRMContentObject class는 DRM 콘텐츠에 대한 실제 내용이 들어 있으므로 전송받은 데이터에서 DRM 콘텐츠의 내용을 저장하는 setContent메서드와 획득 하는 getContent메서드, 그리고 데이터의 길이를 획득하는 getLength메서드를 포함하고 있다.

```
public class DRMContainer{
        DRMContentHeader Header;
        DRMContentObject Content;
        final static int HeaderSize = 90;

        public DRMContainer() {  }
        public String loadContent(String szfileName) {  }
        public String saveToFile(String szfileName, int size) {  }
        public void setContainer(byte[] data) {  }
        public void DataEncrypt(int blocksize, Object key) {  }
        public void DataDecrypt(int blocksize, Object key) {  }
        byte[] hton(int num) {  }
        private int byte4ToInt(byte[] retBuf) {  }
}
```

DRMContainer class는 헤더 정보와 콘텐츠 정보를 모두 포함하여 실제 전송하고자 하는 데이터의 포맷을 정의 하였다. 전송 받은 콘텐츠에 대한 헤더의 내용과 콘텐츠의 내용을 저장할 수 있는 setContainer메서드와 획득할 수 있는 loadContent메서드, 파일로 저장하는 saveToFile메서드, 데이터의 암호화와 복호화를 수행하는 DataEncrypt, DataDecrypt메서드, 그 외 바이트 형태의 데이터를 숫자로 변환 또는 숫자나 문자 데이터를 바이트로 변환하는 메서드를 포함하고 있다.

WIPI에서는 네트워크를 이용한 통신에서 HTTP, HTTPS, TCP/IP, UDP 등의 기저 통신을 이용하므로 권한 관리를 위해 네트워크를 통해 권한에 대한 관리를 하기 위해 Rights class를 두고 권한 내용을 전송 받을 수 있도록 정의한다. 그리고 사용유형에 따라서 1회성, 횟수지정,

무제한의 경우로 사용기간에 있어서는 일 단위, 주 단위, 월 단위, 무제한으로 구분한다. 그리고 사용권한에 대해서는 자기 자신만 사용가능, 재분배 가능한 경우로 두 경우로 나누어 다음과 같이 정의한다.

```
public class Rights{
        //사용유형에 따른 구분
        final static String USEUNLIMMITED = "07";
        final static String USECOUNT = "08";
        final static String USEONCE = "09";
        //사용기간에 따른 구분
        final static String USETERMDAY = "10";
        final static String USETERMWEEK = "11";
        final static String USETERMMONTH = "12";
        final static String USETERMYEAR = "13";
        //사용권한에 따른 구분
        final static String USERIGHTSPREVIEW = "14";
        final static String USERIGHTSOWN = "15";
        final static String USERIGHTSDISTRIBUTION = "16";
        // 순번
        final static int USECASE = 0;
        final static int USETERM = 1;
        final static int USERIGHTS = 2;
        final static int USEKEY = 3;
        //권한 객체 총 사이즈 33바이트
        final static int USECASE_SIZE = 3;
        final static int USETERM_SIZE = 12;
```

```
        final static int USEKEY_SIZE = 16;
        int sizes[] = {USECASE_SIZE, USETERM_SIZE,
                        USERIGHTS_SIZE, USEKEY_SIZE};
        final static int FieldCnt = 4; // 권한 객체 필드의 갯수
        byte[][] m_fields; //권한 객체 각 필드를 나타낼 배열
        public Rights() {  }
        private synchronized void setRights(byte[] rightsBuf) {  }
        public byte[] getRights() {  }
}
```

권한 정보를 SSL소켓을 통하여 전송 하도록 한다. 네크워크를 통하여 바이트 형태로 데이터를 전송 받아 처리 할 수 있도록 정의 하였다.
위에서 정의한 DRM 콘텐츠에 대한 클래스와 권한 관리 객체 외에 단말기에서 네트워크에 대한 기능을 수행할 NetworkMgr class를 정의하여 네트워크를 통해 데이터를 송신하고 수신하는 기능을 정의 하였다.

```
public class NetworkMgr
{
        protected static NetworkMgr instance = null;
        Socket _sSocket = null;
        DataOutputStream _dos = null;
        DataInputStream _dis = null;
        private boolean _bConnect = false;
        public NetworkMgr() {  }
        public static NetworkMgr getInstance() {  }
        public boolean IsConnected() {  }
```

```
    public boolean connect(String url) throws IOException{  }
    public int recvDataSize() {  }
    public byte[] recvData(int size) throws IOException{  }
    public int sendData(String str) {  }
    public int sendData(byte[] buff) throws IOException{  }
    public void close() {  }
}
```

네트워크를 관리하는 NetworkMgr 클래스의 인스턴스를 생성하여 받아올수 있도록 getInstance 메서드를 정의하고, 네트워크를 통해 데이터를 수신 또는 송신 할 수 있도록 sendData 메서드와 recvData 메서드를 정의하였다. 그리고 데이터를 송신하기 전에 보내고자 하는 데이터의 크기를 알려 주도록 recvDataSize 메서드를 이용해 데이터의 크기를 전송 하도록 하였다. 위에서 정의한 바와 같이 DRMContainer class를 이용하여 콘텐츠에 대한 헤더와 데이터에 관련된 부분을 처리하고 네트워크를 통하여 데이터를 송·수신 하는 기능을 NetworkMgr class에 정의하였다. 그 외에 전송받은 콘텐츠를 실행하고 저장하는 부분은 WIPI응용프로그램에서 정의하고 있다.

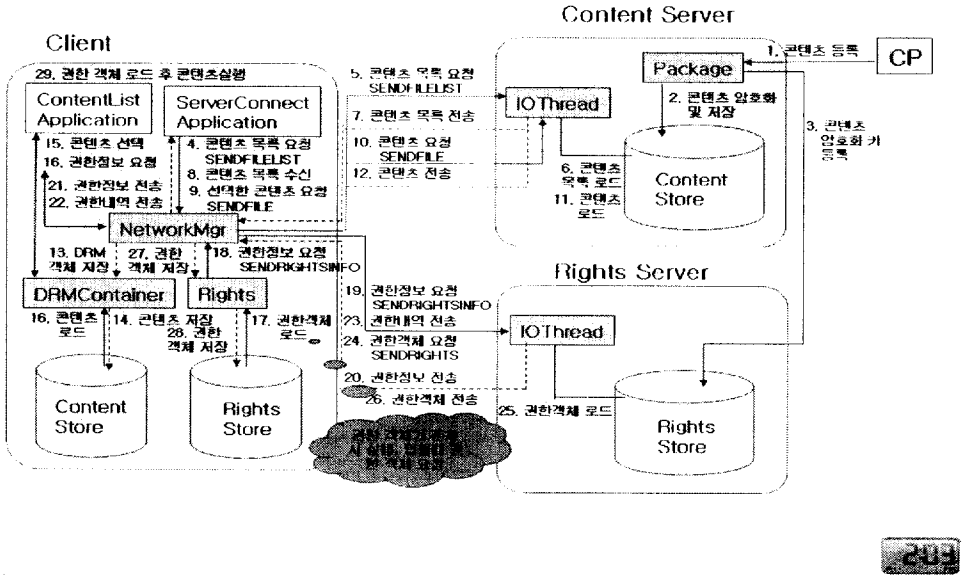
## 3.3 시뮬레이션 및 분석

### 1) 시뮬레이션 결과

시스템 검증은 JAVA로 구현된 Package응용프로그램과 콘텐츠 서버, 권한 서버로 구성된 서버 측 응용프로그램과 PC상에서 KTF WIPI1.2

SDK를 이용하여 구현한 응용프로그램으로 시뮬레이션 하였다.

WIPI기반 모바일 DRM System의 전체 실행 과정은 그림 21과 같이 클라이언트 측과 서버 측에서 정의한 객체를 이용하여 콘텐츠와 권한 객체를 전송 받는 과정을 거쳐 콘텐츠를 실행하게 된다.

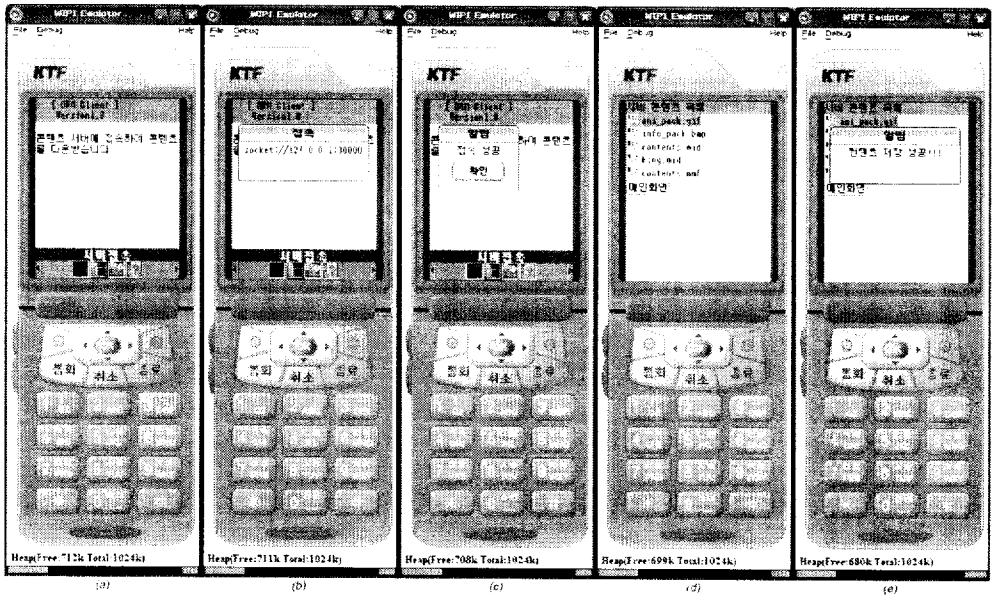



[그림 21] 모바일 DRM System 흐름도

위 그림과 같이 콘텐츠 제공자가 콘텐츠 서버에 콘텐츠를 등록하여 서버 측의 Package응용프로그램에 의해서 콘텐츠와 암호화키를 생성하여 저장한다. 클라이언트 측 WIPI 응용프로그램에서는 NetworkMgr 객체를 이용하여 콘텐츠 서버에 콘텐츠 목록을 요청하고 목록에서 선택한 콘텐츠의 정보를 이용하여 콘텐츠 서버에 콘텐츠를 요청한다. 콘텐츠 목록과 콘텐츠를 요청할 경우에는 사전에 정의된 SENDFILELIST와 SENDFILE 메시지를 넘겨주면서 서버와 통신하도록 한다. 콘텐츠를 전

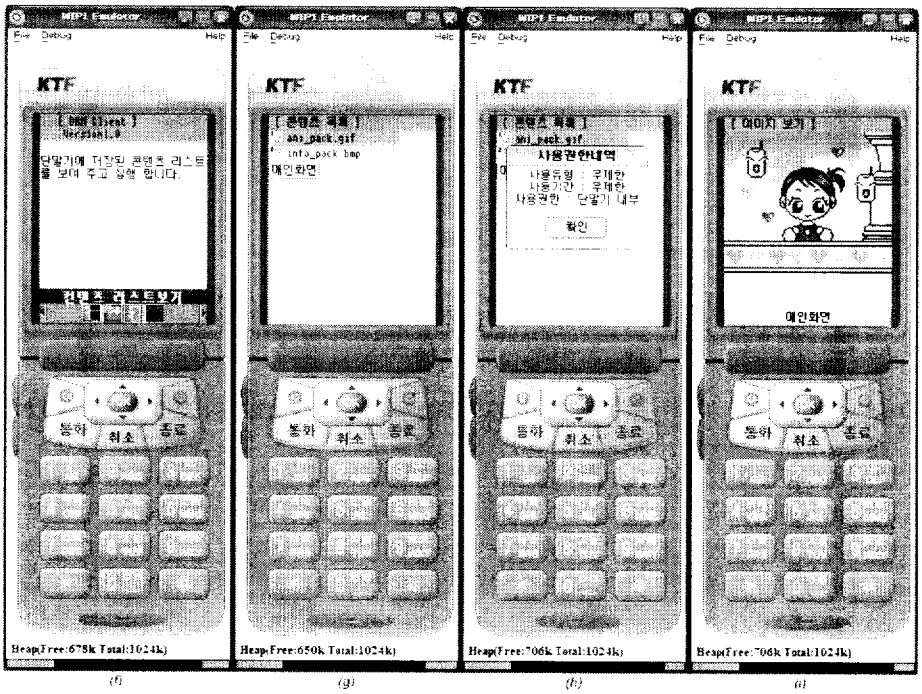
송 받은 후에 콘텐츠 저장소에 저장하고 실행시에 권한 객체를 로드하여 콘텐츠에 대한 권한 객체가 없을 경우 SENDRIGHTSINFO와 SENDRIGHTS 메시지를 권한 서버 측에 전달하여 콘텐츠에 대한 권한 객체를 전송 받고 사용권한에 따라 콘텐츠를 실행하게 된다.

위의 전체 과정에 대해 WIPI SDK를 이용하여 시뮬레이션 한 결과 중 먼저 서버에 접속하여 콘텐츠를 다운 받는 결과는 다음 그림 22와 같다.



[그림 22] 콘텐츠 다운로드 시뮬레이션 결과

초기에 (a)에서 서버 접속 메뉴를 실행하면 (b)와 같이 콘텐츠 서버에 접속하여 접속이 승인되면 (c)와 같이 알려주며, (d)에서와 같이 서버 측의 콘텐츠 목록을 다운로드 받아 클라이언트 화면에 출력한다. 그리고 콘텐츠 목록에서 원하는 콘텐츠를 선택하면 콘텐츠를 다운로드 받아 저장하게 된다. 저장하는 과정에서 DRMContainer class 형태에 맞추어 메모리상에 저장하여 파일로 저장하게 된다.




[그림 23] 콘텐츠 실행 시뮬레이션 결과

그림 22에서 다운로드 받은 콘텐츠목록을 그림 23의 (g)와 같이 나타내주고 목록 중에서 실행 하고자 하는 콘텐츠를 선택하여 실행 시키면 (h)와 같이 사용권한 내역에 대해 알려주고 암호화되어 있는 콘텐츠를 복호화 하여 (i)와 같이 실행하게 된다.

권한 객체가 없을 경우에는 콘텐츠를 다운 받는 경로와 마찬가지로 권한 서버에 접속하여 콘텐츠에 대한 권한을 요청하게 된다.

서버로부터 다운로드 되는 콘텐츠는 서버에서 적절한 권한을 확인하여 단말기에 전송되므로 기본적인 저작권 보호가 가능하다. 하지만 적외선 통신 등을 통하여 단말기에 저장된 콘텐츠를 무단으로 복사해 갔을 경우는 권한을 체크할 수 없다. 위에서 정의된 WIPI 응용프로그램을 통해서는 단말기에 저장된 콘텐츠를 실행 하고자 할 때 콘텐츠에 대한 사용권

한을 관리가능하기 때문에 불법사용을 방지할 수 있다. 그리고 콘텐츠에 대한 권한정보를 권한서버와 미리 정해진 SSL소켓을 이용하여 송·수신하기 때문에 암호화에 관한 키 정보 또는 사용권한에 대한 내역을 안전하게 전송 할 수 있다. 그리고 단말기 상에 저장된 콘텐츠에 대한 권한 내역 역시 WIPI에서 기본적으로 제공하는 보안에 따라 DRM Client 응용프로그램만이 접근 할 수 있도록 설정 되어 있기 때문에 다른 응용프로그램에서는 접근 하지 못하므로 단말기 내부에서도 안전하게 사용권한을 보관할 수 있도록 한다.

### 2) 실험 결과

실험 결과는 구현에서 사용된 암호화 방법인 AES-128bit를 이용하여 콘텐츠를 모바일 단말기에서 복호화의 수행 속도를 비교하여 모바일 단말기에서 사용될 수 있는 암호화 알고리듬에 대해서 살펴보았다. 수행 결과는 표 5와 같이 나타내었다.

속도 수행 결과는 WIPI Emulator의 환경을 ARM9 칩에 맞추어 실험하였고 실험 결과를 보면 콘텐츠의 크기가 커질수록 수행 시간이 커지는 것을 볼 수 있지만 수행 시간의 증가 비율이 급격히 증가하지 않고 수행 속도 또한 사용자가 직접 사용하기에 불편하지 않을 정도의 속도를 나타내고 있다. 따라서 AES-128bit 알고리듬을 이용한 블록 암호화 기법도 모바일 환경에서 충분히 사용가능 하다고 보여 진다. 그리고 좀 더 수행 속도를 증가 시킬 수 있는 알고리듬을 연구하고 암호화의 방법에서도 전체 콘텐츠를 암호화 하는 것이 아니고 헤더 또는 일부분만을 암호화 할 수 있는 부분 암호화 기법도 연구 되어야 하겠다고 보여 진다.

[표 5] AES -128bit 복호화 실험결과

| Test 환경 | Content Type | Test File | Size(Bytes) | Times(ms) |
|---|---|---|---|---|
| KTF WIPI1.2 Emulator | Image | 10K Image | 12768 | 6.56 |
| | | 20K Image | 22320 | 9.85 |
| | | 30K Image | 31280 | 12.66 |
| | | 40K Image | 40496 | 16.87 |
| | | 50K Image | 55088 | 21.25 |

# IV. 결 론

본 논문은 최근 이슈가 되고 있는 모바일 DRM 관련한 국제표준 기술 사양 분석과 한국 무선 인터넷 플랫폼 상에서의 DRM 기술 분석을 위해 OMA DRM v1.0과 v2.0의 국제 표준을 분석하고 WAP, BREW 등과 같은 무선 인터넷 플랫폼에서 요구되는 보안 기술을 조사·분석하여 WIPI 환경에서의 보안 기술 적용을 위한 환경 분석 및 DRM 적용을 위해 연구하였다. 또한 WIPI 플랫폼에서 모바일 콘텐츠에 대한 사용권한을 제어하기 위한 WIPI용 모바일 DRM 컴포넌트를 설계하였다. WIPI용 모바일 DRM 컴포넌트는 콘텐츠 전송방식에 중점을 두어 서버로부터 콘텐츠를 다운로드한 뒤에 사용하는 기존의 형태에서 단말기 상호간에 P2P 등의 형식으로 재전송이 가능하다. 모바일 콘텐츠의 유통방식이 다양하게 변화함에 따라 표준화된 플랫폼 상에서 다양한 콘텐츠에 대한 저작권 보호가 가능한 WIPI DRM을 제시하였다.

국내 DRM 기술 관련 연구 및 개발은 90년대 말부터 시작되어 현재 세계 수준의기술 경쟁력을 보유한 것으로 판단되고 있다. 국내 DRM 업체의 의욕적인 기술 개발과 다양한 응용 분야의 적극적 시장 개척을 통해 다양한 사용화 경험 지식을 보유하고 있다. 그러나 국내의 DRM 연구 개발은 세계적인 DRM 기술 방향을 제대로 따라가지 못하고 있다. 특히, 디지털 방송 및 디지털 홈서비스 분야의 DRM 기술은 이미 오래 전부터 해외의 여러 표준화 단체에 의해 준비되었음에도 불구하고 국내에서는 이러한 분야의 참여 실적이 거의 없는 상황이다. 본 논문에서는 국내의 무선 인터넷 플랫폼 환경에서 보안 환경을 적용하여 앞으로 모바일 분야에서 세계적인 기술이 되기 위한 방안을 찾기 위해 선결해 나가야 할 문

제점들이 무엇인지, 그리고 이에 대한 해결 방안을 위하여 연구를 하게 되었다. 이를 계기로 하여 앞으로 WIPI의 국내 표준화 작업 및 보안 요구사항의 적용에 의해 세계적인 모바일 플랫폼으로 자리를 잡아 나갈 수 있는 연구를 계속해 나가야 한다.

또한 앞으로 미래 사회는 단 하나의 우수한 기술만이 살아남게 될 것이다. 이것은 곧 표준화에 대한 중요성을 의미한다. 특히, DRM의 경우에는 상호호환성이 핵심적인 요구사항이기 때문에 표준화의 활동이 무엇보다도 중요하다. 따라서 모바일 환경에서의 DRM이 인터넷 기반의 DRM이 가질 수 없는 장점을 찾아내 표준화할 수 있는 연구를 계속해 나가야겠다.

# 참 고 문 헌

[1] Zheng Yan, "Mobile Digital Rights Management", Nokia Research Center, Network Security 2001.

[2] Frank Hartung, "Mobile DRM", Digital Rights Management, LNCS 2770, pp. 138-149, 2003.

[3] OMA, "OMA-Download-DRM-v1_0-20031031 Digital Rights Management", Open Mobile Alliance, October 31, 2003.

[4] OMA, "OMA-Download-ARCH-v1_0-20020610-C Download Architecture", Open Mobile Alliance, June 10, 2002.

[5] OMA, "OMA-Download-DRMREL-V1_0-20020801-C Rights Expression Language Candidate", Open Mobile Alliance, Aug, 2002.

[6] OMA, "OMA-DRM-ARCH-V2_0-20040518-D DRM Architecture", Open Mobile Alliance, May, 2004.

[7] OMA, "OMA-DRM-DRM-V2_0-20040617-D DRM Specification V2.0" Open Mobile Alliance, June, 2004.

[8] ETRI, "WIPI 표준화 현황 및 규격", Korea Mobile Conference, June 26, 2002.

[9] KWISF, "KWISFS.K-05-002 모바일 표준 플랫폼 규격", 한국무선인터넷 표준화 포럼, February, 2004.

# 감사의 글

지금까지 많은 분들의 관심과 보살핌으로 본 논문을 마무리하게 되었습니다. 이 자리를 빌어 그 분들께 감사를 드리고자 합니다.

먼저 바쁜 학교생활과 직장생활을 겸하며 부족함이 많았지만 항상 아낌없는 관심과 지도를 베풀어주신 故 박지환 교수님께 감사를 드리며 지금은 감사함을 어떻게 전할 수가 없는 것이 안타깝습니다. 그리고 못지않게 많은 관심과 지도를 베풀어 주신 신상욱 지도교수님께 진심으로 감사의 말씀을 드립니다. 짧은 시간이었지만 두 분 교수님의 지도 덕분에 제가 이렇게 하나의 결실을 보게 되었습니다. 대학원 생활과 학과 수업을 통해 많은 가르침을 주신 이경현 교수님, 김창수 교수님, 정연호 교수님, 조성진 교수님, 최명구 교수님께도 감사를 드립니다.

지금의 논문을 완성하기까지 많은 도움을 주신 이진홍님, 임태훈님, 박영란님, 최재귀님, 김소진님, 강현호님께 감사를 드리고 항상 번거로운 일로 많이도 괴롭혔던 승우와 병만이 그리고 연구실원들에게 감사의 말을 전합니다.

그리고 한국정보통신대학교 부설 한국정보통신교육원의 배재식 선생님, 고점숙 선생님과 직원들에게 감사의 말을 전합니다.

또한 학교생활로 조금은 소홀했던 한국정보통신교육원의 제자들에게도 미안한 마음과 함께 고마움을 전합니다.

마지막으로 항상 제 곁을 지켜주며 많은 조언으로 부족함을 채워주는 사랑하는 아내 유진영, 이제 막 세상에 나온 아들 유진이 그리고 양가 부모님과 형제자매들에게도 깊은 감사를 드리며 이 논문을 바칩니다.