공학석사 학위논문

CP-ABPRE를 활용한 DRM 클라우드 서비스

지도교수 신 상 욱

이 논문을 공학석사 학위논문으로 제출함.

2015년 2월

부 경 대 학 교 대 학 원

정보보호학 협동과정

허 창 수

공학석사 학위논문

CP-ABPRE를 활용한 DRM 클라우드 서비스

지도교수 신 상 욱

이 논문을 공학석사 학위논문으로 제출함.

2015년 2월

부 경 대 학 교 대 학 원

정보보호학 협동과정

허 창 수

허창수의 공학석사 학위논문을 인준함.

2015년 2월 일

주심　　이학박사 이 경 현　(인)

위원　　이학박사 신　　원　(인)

위원　　이학박사 신 상 욱　(인)

차 례

그림 차례

표 차례

DRM Cloud Using CP-ABPRE

Chang Soo Heo

Interdisciplinary Program of Information Security, The Graduate School,
Pukyong National University

Abstract

Cloud computing technology that is many studies have been made in order to be applied in various IT environment. one of the important issues in the content services, a DRM(Digital Rights Management) technology for secure use of the content. In the DRM technology, even the content of different DRM technology is applied, interoperability is required for users to be used on devices that use. DRM Cloud architecture to provide these interoperability has been proposed recently.

In this paper, by utilizing the CP-ABPRE in DRM Cloud environment, after dispensing the attributes to the user, only users with the attributes that satisfies the access structure is to be able to run the operation of the content. Therefore, to ensure fine-grained access control for content.

The license key for protection of content encryption key, it divided into two part, license master key and assistant key. then I enforce access policies based on attributes to distribute the license master key securely, The assistant key allows the distribution after checking whether the attribute is revoked. To apply these methods to solve the problem immediately attributes revocation.

I. 서 론

1. 연구배경

스마트 폰과 같은 이동성을 갖춘 모바일 기기와 IT 자원을 필요한 만큼만 서비스하는 클라우드 컴퓨팅 기술은 시간과 위치에 의한 제약성을 제거함으로써 사용자의 편이성을 증대시키고 IT자원을 필요로 하는 만큼만 서비스로 제공함으로써 비용을 감소시켜 IT 환경에 커다란 변혁을 일으키면서 다양한 분야에 적용되기 위해 많은 연구가 이루어지고 있다[1][2].

콘텐츠 서비스 분야에도 클라우드 환경에서 다양한 장치를 이용하여 언제, 어디서든지 소비자들이 원하는 콘텐츠를 제공하고자 노력하고 있다[1][2]. 콘텐츠 서비스에서 중요한 이슈의 하나로서 콘텐츠의 안전한 사용을 위한 기술인 DRM(Digital Rights Management)이 있다. 개별 단말 환경만을 고려해서 개발되었던 DRM 기술들은 콘텐츠 서비스 간에 자유로운 이동이 힘들고, 가입된 서비스를 지원하는 단말만을 구매해야하기 때문에 사용자의 입장에서는 불편함을 느끼게 된다. 이러한 불편함이 발생하자 서로 다른 DRM 기술이 적용된 콘텐츠일지라도 사용자가 사용하는 장치들에서 사용이 되어야함을 말하는 상호 운용성(interoperability)을 필요로 하게 되었다[3][4].

상호 운용성을 제공하기 위한 DRM 기술들이 제안되었으며, Lee 등[5]은 클라우드 환경에서 DRM 기술을 적용한 DRM Cloud 아키텍쳐를 제안하였다. 이 모델에서 콘텐츠 다운로드 서비스, 라이선스 발급 및 도메인 가입 시나리오를 제시하였다. 그러나 제안된 DRM Cloud에

서는 콘텐츠 암호화 키, 콘텐츠 암호화 키를 보호하기위한 라이선스 키들이 단지 대칭키 방식의 암호화 방식으로만 이루어지며 라이선스에서는 하나의 스마트 장치를 위한 라이선스로 작성이 되었다. 라이선스가 하나로만 작성이 되어있으므로 모든 스마트 장치들에게 맞는 라이선스를 작성하여 각각에게 라이선스를 배포해야한다는 점과 라이선스가 만료되더라도 라이선스를 수정하여 다시 사용하는 것이 아닌 재 배포를 받아야한다는 불편한 부분이 있으며, 추가적인 방법 없이 대칭키 암호화 방식으로만 보호를 하였기 때문에 라이선스 키 또는 콘텐츠 암호화 키가 노출이 되었을 경우 콘텐츠가 무단으로 재 배포되었을 경우 정당한 지불없이 콘텐츠가 사용 될 수 있는 문제점이 발생한다.

현재 클라우드 컴퓨팅에서의 DRM 기술은 암호화된 데이터를 다시 암호화를 하게 될 때에 데이터가 공개되지 않고 암호화를 실행하는 프록시 재-암호화(Proxy Re-Encryption)방식과 사용자에게 속성을 부여하여 부여받은 속성이 암호화된 데이터의 접근 구조를 만족하면 복호화를 할 수 있는 속성 기반 암호화(Attribute-based Encryption)방식을 적용한 기술이 제안되었다. 이러한 기술로 인해 사용자의 세분화된 접근제어(Fine-grained Access Control)가 보장이 되도록 하였다.

DRM에서 속성 기반 암호화 방식을 적용하게 될 때의 큰 문제는 여러 사용자들에게 속성이 소유되어있는 특성 때문에 속성과 사용자 폐지가 어렵다는 점이다[7]. 이러한 문제를 해결하기 위해 콘텐츠 암호화 키를 두 개의 키로 분할하여 사용을 하도록 본 논문에서는 제안하였으며, DRM Cloud에 CP-ABPRE 방식을 적용하여 보다 안전하고 효율적인 DRM 서비스를 DRM Cloud에서 제공할 수 있도록 한다.

2. 연구 내용 및 구성

본 논문에서는 Lee[5]등이 제안한 DRM Cloud 모델에 CP-ABPRE 및 속성 기반 접근 제어 방식을 적용하여 콘텐츠 다운로드 서비스 및 라이선스 발급, 콘텐츠 재생 및 콘텐츠 재 배포 등 여러 서비스들을 라이선스에 사용자가 이용할 수 있는 접근 권한을 명시하여 사용자에게 배포된 속성 비밀 키의 속성이 라이선스에 명시된 접근 권한을 만족하지 않으면 사용자가 콘텐츠에 적합한 이용을 하지 못하도록 하는 방법을 통해 세분화 된 접근제어가 가능 하도록 하였다. 또한, 콘텐츠 재 배포를 하게 될 때에 라이선스를 DRM Cloud 서버로부터 새로 받지 않고 도메인내의 장치에게 받아서 사용할 수 있게 하였다. 이때에 라이선스는 추가적인 접근제어를 통해 라이선스를 확장 할 수 있다.

속성 및 사용자의 폐지 문제는 콘텐츠를 암호화하는 콘텐츠 암호화 키를 라이선스 키로 암호화하며, 이때 라이선스 키를 두 개의 키로 분할하게 된다. 분할된 두 개의 키 중 하나는 접근 구조로 암호화를 하게 되며 다른 하나는 콘텐츠를 재생하게 될 때에 서버로부터 속성 폐지 검증을 요청하여 폐지되지 않았다면 키를 전송을 두 개의 키를 다시 결합한 후 콘텐츠를 사용 할 수 있도록 하였다.

본 논문의 구성으로는 2장에서는 본 논문에서 다루어질 기술들인 DRM Cloud, ABAC, CP-ABPRE에 대하여 살펴보며 3장에서는 CP-ABPRE를 활용한 DRM Cloud의 서비스 절차에 대해 살펴보고 4장에서는 제안한 방식의 분석 및 구현환경을 보여주고 5장에서 결론을 통해 마친다.

II. 관련 연구

본 장에서는 제안된 DRM Cloud 아키텍쳐 모델에 대하여 살펴보며 기본적인 속성 기반 접근 제어 메커니즘을 설명 및 CP-ABPRE의 개념과 알고리즘, 그리고 CP-ABPRE에서 재 암호화 제어와 추가적인 확장 접근 정책에 대해 간단하게 설명한다.

1. DRM Cloud

Lee[5] 등은 <그림 1>과 같은 클라우드 환경의 DRM Cloud 모델을 설계하였다. 제안된 모델에서 콘텐츠 소유자는 미디어 클라우드에게 콘텐츠를 등록하는 개체이며 미디어 클라우드는 스마트 장치에게 콘텐츠 배포 및 DRM Proxy에게 DRM 관련 정보들을 전송하는 개체이며, 스마트 장치는 자신이 사용하고 싶은 콘텐츠를 미디어 클라우드에서 선택하여 다운로드 받고 DRM Proxy에게 라이선스를 획득하거나 도메인 관련정보들을 전송하여 콘텐츠를 사용하는 개체이다.

DRM Cloud는 Application, Service, Platform, Storage의 4가지 계층으로 구분이 되어있으며 각 계층은 다음과 같은 역할을 한다.

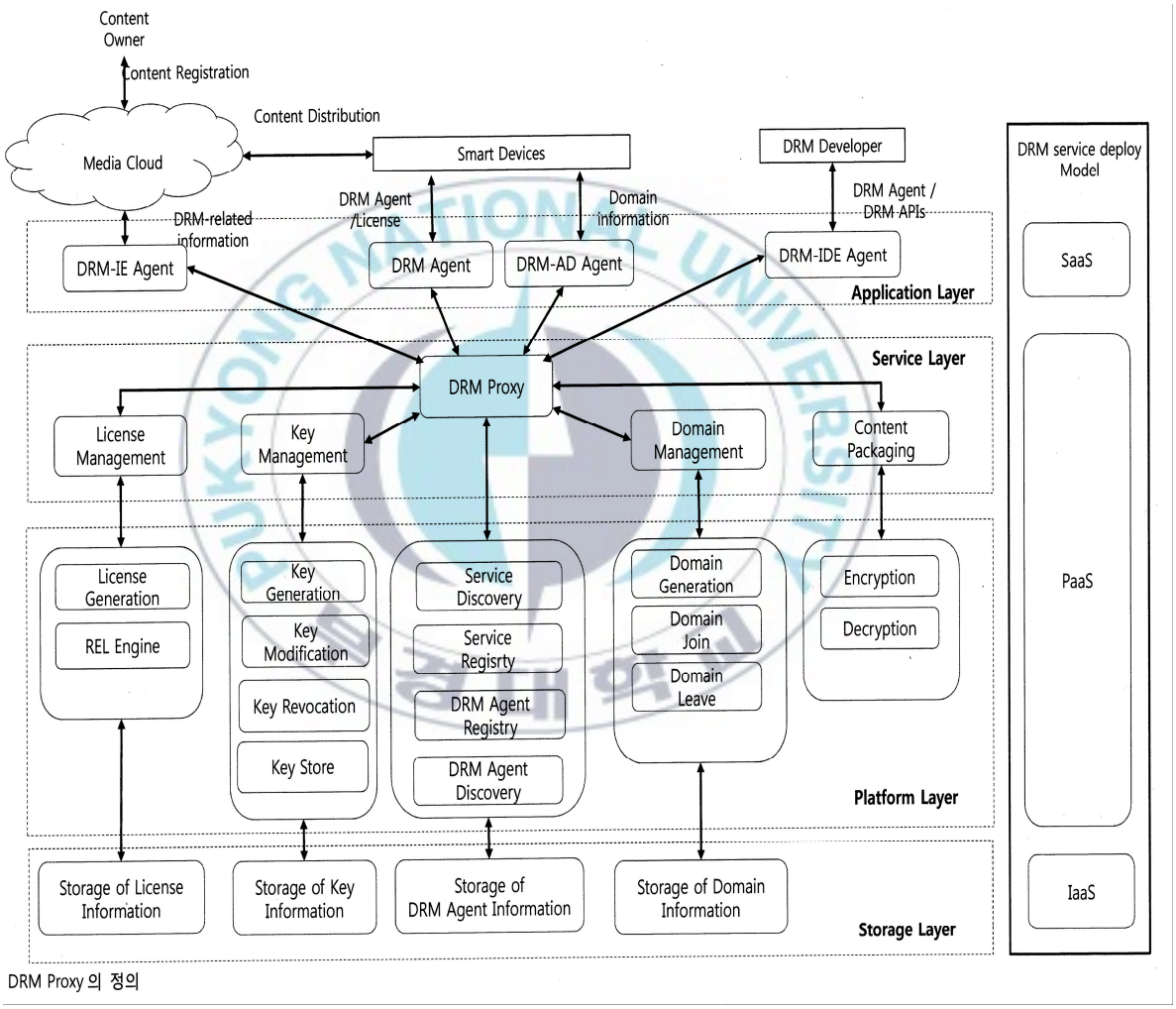
• Application 계층 : 스마트 장치 와 미디어 장치 및 클라우드 사용자와 DRM Cloud를 연결해주는 인터페이스에 대한 어플리케이션을 제공.

• Service 계층 : 실질적인 DRM 서비스를 제공하며, 라이선스, 키, 도메인, 콘텐츠 패키징에 대한 DRM 서비스들을 관리.

• Platform 계층 : DRM 기능을 위한 구성요소 및 API를 제공하

며, 세부적인 DRM 기능으로 라이선스 생성, 암호화 키/라이선스 키/도메인 키 생성 및 수정과 저장을 하며 도메인 생성/가입/탈퇴 기능과 콘텐츠 패키징에 대한 암호화 및 복호화 등 여러 기능들을 제공.

• Storage 계층 : 라이선스 정보, 도메인, DRM 에이전트 및 도메인 및 기타 여러 사항들을 저장 및 관리.



< 그림 1 > DRM Cloud 아키텍쳐 모델[5]

2. ABAC(Attribute Based Access Control)

ABAC는 ABAC 시스템에 의해 관리되는 시스템리소스에 대해 사용자가 어떠한 작업을 수행하기위한 접근 요청을 했었을 때 사용자에게 할당된 속성, 시스템 리소스의 속성, 환경 조건 등 속성과 조건의 측면에서 지정된 정책의 집합들 기반으로 사용자의 작업을 허용하거나 거부하는 기능을 제어하는 것을 말한다[8].

< 그림 2 > ABAC 메커니즘

<그림 2> 에서는 기본적인 ABAC 메커니즘을 보여준다. 사용자가 속성을 가지고서 시스템 리소스에 접근 요청을 하게 될 때 접근 제어 메커니즘은 환경 조건, 시스템 리소스 속성, 접근 제어 조건을 가져

와서 접근 제어 메커니즘에 명시된 정책에 적합한지를 판별하여 접근 요청을 허가하거나 거부를 하는 메커니즘이다.

3. CP-ABPRE(Ciphertext Policy - Attribute Based Proxy Re-Encryption)

CP-ABPRE는 Liang 등[9], Luo 등[6]이 제안하였다. 본 논문에서는 Luo 등이 제안한 방식을 사용하였기에 Luo[6] 등이 제안한 방식을 설명한다.

CP-ABE의 방식은 속성과 접근 정책이 암호문과 사용자의 복호화 키와 관련이 되어있으며, 암호문은 접근 정책으로 암호화하며 복호화 할 수 있는 복호화 키는 속성의 집합으로 생성된다. 주어진 암호문에 대해 접근 정책을 만족하는 속성의 집합을 가진 복호화 키가 유일하게 암호문을 복호화 하는데 사용 할 수 있다.

CP-ABPRE는 기존의 CP-ABE(Ciphertext Policy Attribute Based Encryption)에서 Proxy에 재 암호화 기능을 부여하여 암호화된 데이터를 재 암호화키를 사용하여 재 암호화 할 수 있는 기능이 들어간 방식으로서 다음과 같은 장점을 얻을 수 있다.

• Proxy는 재 암호화 동작을 실행 할 수 있도록 기능을 부여하면 데이터 소유자의 계산 오버헤드를 감소시킬 수 있다.

• 인가된 사용자는 암호화 된 데이터를 복호화하기위해 자신의 비밀키를 사용하고, 재 암호화된 데이터를 복호화하기위한 추가적인 키를 필요로 하지 않는다.

• 민감한 데이터는 재 암호화 과정에서 Proxy에게 공개가 되지

않으며 Proxy는 데이터 소유자의 명령을 준수한다.

Luo[6] 등 은 Liang[9] 등이 제안한 방식에서의 특징을 유지하면서 추가적으로 재 암호화 제어의 특성을 포함하였다. 재 암호화 제어의 특성은 데이터 소유자가 암호화된 데이터를 재 암호화를 할 수 있도록 할 것인지 아니면 재 암호화를 못하도록 할 수 있도록 하는 특성이다.

암호문 정책 기반의 접근 정책을 사용하고 있으며 접근 정책은 암호화된 데이터에 사용자의 접근 제어를 하기 위한 것이며 접근 정책은 AND-gate 정책으로 나타나있다. Luo[6]가 제안한 방식의 CP-ABPRE는 아래와 같이 6개의 알고리즘으로 구성 되어있다.

- Setup(1^k) : 보안 파라미터 k를 선택 후 입력하여 공개 키 PK, 마스터 키 MK를 생성한다.
- KeyGen(MK, S) : MK와 속성의 집합 S을 입력하여 속성 S에 대한 비밀 키 SK_S을 만든다.
- Encrypt(PK, M, AS) : PK와 메시지 M, 접근 정책 AS를 입력하여 암호문 CT_W를 생성한다.
- RKGen(SK_S, AS) : SK_S과 접근 정책 AS를 입력하여 재 암호화 키 $RK_{S \to AS}$를 생성한다.
- Reencrypt($RK_{S \to AS'}, CT_{AS}$) : 재 암호화 키 $RK_{S \to AS'}$와 암호문 CT_{AS}를 입력한 후에 CT_{AS}의 접근 정책에 $RK_{S \to AS'}$의 속성이 만족하는지를 검사한다. 만족을 한다면, 재 암호화된 암호문 $CT_{AS'}$를 생성한다. 만족하지 않는다면 생성할 수 없다고 알린다.
- Decrypt(CT_{AS}, SK_S) : CT_{AS}와 S에 대한 비밀 키 SK_S를 입력하

여 S이 CT_{AS}에 대한 접근 정책 AS를 만족한다면 메시지 M을 반환한다.

재 암호화 제어는 암호화를 수행한 개체가 암호문에 있는 값을 제공하지 않는다면, 원래의 복호화에는 영향을 주지 않는다. 하지만, 재 암호화 된 암호문의 복호화는 수행을 할 수 없다. 그리하여, 암호문에 있는 값을 제공하거나 하지 않거나 하는 것으로 재 암호화 제어를 수행 할 수 있다.

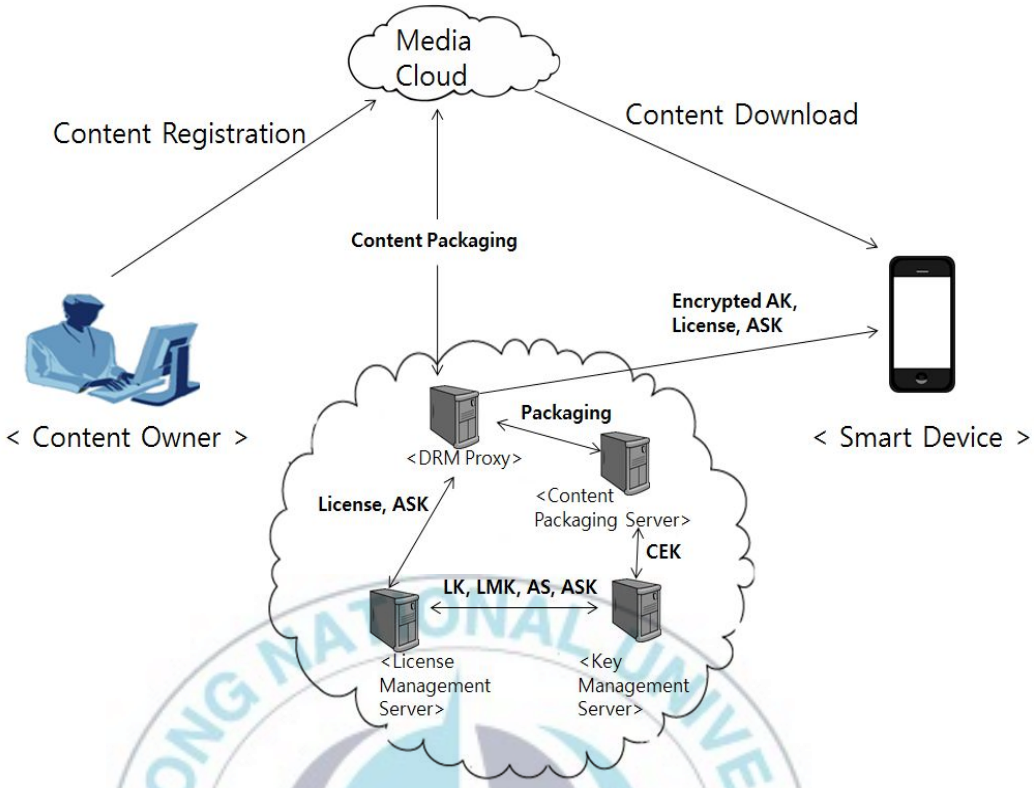
제안한 방식에서는 Proxy가 암호문을 재 암호화 할 때에 추가적으로 접근 정책을 확장 할 수 있도록 허용을 하고 있다. Proxy가 정책 W에서 W'으로 암호문을 재 암호화 할 수 있으며, 확장 접근 정책 W''를 재 암호화된 암호문에 추가 할 수 있으며 속성 집합 L이 W'와 W''를 동시에 만족할 경우에만 재 암호화된 암호문을 복호화 할 수 있도록 제안 하였다.

III. CP-ABPRE를 사용한 DRM Cloud 서비스

본 장에서는 제안된 DRM Cloud에 CP-ABPRE를 활용하여 콘텐츠 등록, 도메인 생성/가입, 라이선스 획득, 콘텐츠 재 배포 및 패키징, 콘텐츠 재생 서비스 절차들을 살펴보며, 속성 구조 및 라이선스 작성에 대해서도 살펴본다.

1. 시스템 모델

제안하는 CP-ABPRE를 사용한 DRM Cloud 서비스 시스템 모델은 아래의 < 그림 3 > 과 같다. 개체들은 콘텐츠의 저작권 및 미디어 클라우드에 업로드를 하려는 콘텐츠 소유자, 콘텐츠 소유자로부터 업로드 받은 콘텐츠를 사용자에게 배포하는 미디어 클라우드, 미디어 클라우드로부터 원하는 콘텐츠를 다운로드 받으며, DRM Cloud로부터 라이선스를 획득 및 도메인 생성/가입을 하는 스마트 장치 그리고 DRM Proxy가 있으며 DRM Proxy는 CPS(Content Packaging Server), LMS(License Management Server), KMS(Key Management Server), DMS(Domain Management Server)와 연결되어있는 하나의 DRM Cloud이며, 개체들의 자세한 설명은 아래와 같다.



< 그림 3> 시스템 모델

• 콘텐츠 소유자 : 콘텐츠를 직접 소유자로서 미디어 클라우드에 자신의 콘텐츠를 등록하여 사용자에게 제공되기를 바라는 개체이다. 미디어 클라우드와 서로 사업적 협약에 의해 신뢰하는 관계를 가지고 있다. 콘텐츠를 미디어 클라우드에 전송하여 콘텐츠 등록 절차를 수행하며 패키징된 콘텐츠의 정보를 전송받아 콘텐츠 등록을 마치게 된다.

• 미디어 클라우드 : 콘텐츠 소유자에게 전송받은 콘텐츠를 콘텐츠 사용자에게 전송하여 콘텐츠를 판매하는 개체이다. Proxy에게 패키징을 요청하는 작업을 수행하며 패키징 된 콘텐츠를 저장하며, 사용자가 원하는 콘텐츠를 다운로드 할 수 있도록 한다. 사용자가 콘텐츠를 다운로드 받을 때에 구입 한 콘텐츠의 사용 권한을 가지고 있으며

사용자가 Proxy에게 라이선스를 획득하게 될 때에 사용자에 대한 콘텐츠 사용 권한을 전송한다.

• DRM Cloud : DRM Proxy, DMS, LMS, KMS, CPS 로 구성되어있으며, DRM Cloud 내에서 통신은 안전한 채널을 이용하며 미디어 클라우드와 신뢰하는 관계를 가지고 있다.

DRM Proxy는 미디어 클라우드와 스마트 장치를 연결하는 서버이며 전송받는 서비스 요청에 대한 기능을 DMS, LMS, KMS, CPS와 통신을 통해 서비스 요청에 대한 응답을 한다.

DMS는 스마트 장치의 도메인 가입/탈퇴 에 대한 기능을 수행하며 미디어 클라우드에게 콘텐츠에 대한 사용권한을 전송 받아 도메인의 접근 구조를 생성한다. 또한, 도메인 인증서를 생성/검증 하는 서비스를 수행한다.

LMS는 사용자가 다운로드 받은 콘텐츠에 대한 라이선스를 생성하는 서버이다. 콘텐츠 암호화 키를 포함시켜 라이선스를 만들게 되며 이때에 접근 구조로서 라이선스를 작성하게 된다.

KMS는 콘텐츠 암호화 키 및 속성 비밀 키, 라이선스 키 등 재 암호화 키를 제외한 키를 생성하는 서비스를 제공한다.

CPS는 미디어 클라우드에게 콘텐츠 패키징 요청을 받았을 경우 미디어 클라우드에서 요청한 형태로 패키징을 하며 콘텐츠 암호화 키를 KMS로부터 전송받아 콘텐츠를 암호화하는 서비스 기능을 제공한다.

• 스마트 장치 : 미디어 클라우드에 콘텐츠 소유자가 업로드한 콘텐츠를 사용하려는 사용자이다. 사용자는 미디어 클라우드에서 원하는

콘텐츠를 다운로드 받은 후 DRM 에이전트를 통해 DRM Proxy에 라이선스 획득 요청을 하여 라이선스와 라이선스 안에 포함된 라이선스 키의 부분 키 LMK를 복호화 하기위한 속성 비밀 키를 받으며, 콘텐츠를 사용하게 될 때에 속성 및 사용자 폐지 검증을 DRM Proxy에게 받은 후 적합한 사용자일 경우 보조 키 AK를 받아 콘텐츠 암호화 키를 생성하여 콘텐츠를 사용 할 수 있는 개체이다.

2. 콘텐츠 서비스 프로토콜

콘텐츠 서비스 프로토콜에서는 CP-ABPRE를 활용한 DRM Cloud에서의 여러 서비스 절차들에 설명을 한다. 각 서비스 절차들은 콘텐츠 등록, 도메인 생성, 도메인 가입, 라이선스 획득, 콘텐츠 재 배포 및 재 패키징, 콘텐츠 재생 서비스 절차가 있으며 각각의 절차에서 나타내는 표기는 [표 1]의 기호를 사용하여 기술한다.

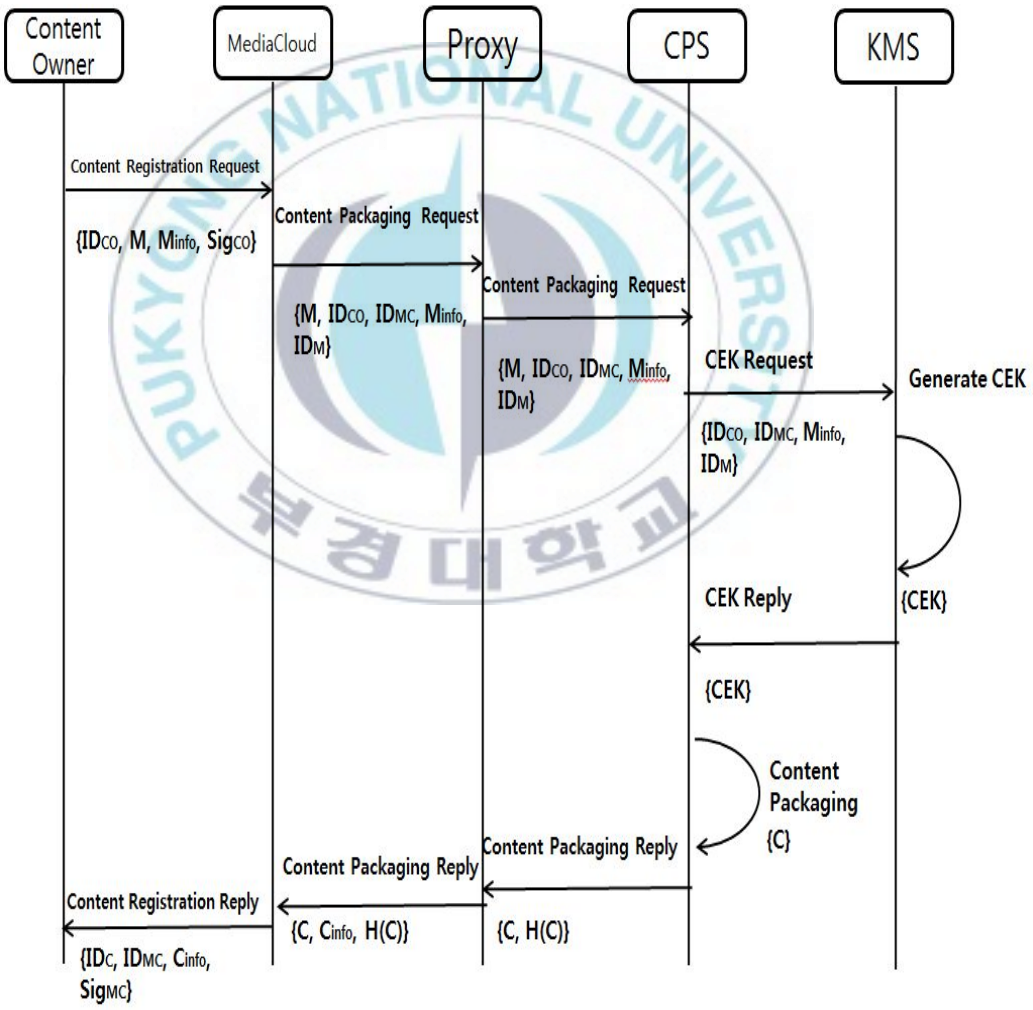
CP-ABPRE의 Setup 알고리즘을 KMS가 실행하게 되며, 적절한 파라미터를 입력하여 출력 값 PK, MK를 생성한다. 생성된 PK는 DMS에게 전달된다고 가정하며 모든 DRM Cloud내의 개체들간의 통신은 안전하고 인증된 채널을 통해 통신한다고 가정한다.

[표 1] 기호

표기	내 용
ID_x	x의 ID
M, M'	콘텐츠, 수정된 콘텐츠
C, C'	패키징 된 콘텐츠, 재 패키징 된 콘텐츠
x_{info}	x에 대한 정보
Sig_x	x의 서명 값
CEK	콘텐츠 보호를 위해 암호학 적으로 안전한 대칭키 암호화 방식의 콘텐츠 암호화 키
LK	CEK를 보호하기 위해 암호화한 라이선스 키 이며 LMK와 AK로 분할이 되며 두 개를 합쳐서 생성 가능한 키
LMK	LK의 부분 키 로서 속성 구조로 암호화 되어 보호되는 키
AK	LK의 부분 키 로서 속성 구조를 만족 했을 경우 Proxy에게 전송 받는 키
$H(.)$	암호학적으로 안전한 해쉬 함수
$DCert_x$	x의 도메인 인증서
UR	콘텐츠에 대한 사용 권한
S_x	x의 속성 집합
AS_x	x의 접근 구조
ASK_x	x의 속성 비밀 키
FDK	도메인 인증서 공유 및 AK분배를 위한 암호학적으로 안전한 대칭키 방식의 키
L, L'	Rights Expression Language로 작성되어 생성된 라이센스, 콘텐츠 재 배포시에 재 암호화된 라이선스
E_{key}	key를 사용하여 암호화하는 기능
PK_x	암호학적으로 안전한 공개키 방식의 x의 공개키
AA	인증과 권한 부여를 하는 인증 기관
$Ucredential$	사용자에 대한 자격정보 집합

2.1. 콘텐츠 등록 절차

콘텐츠 등록 절차에서는 콘텐츠 소유자가 미디어 클라우드에게 자신이 소유한 콘텐츠를 등록하며 이때 패키징에 관한 정보와 콘텐츠를 Proxy에게 전송하여 패키징을 하며 콘텐츠를 등록하게 되는 절차이다. 이러한 서비스 절차는 < 그림 4 >에 나타나있으며 자세한 절차는 다음과 같다.



< 그림 4 > 콘텐츠 등록 절차

• 콘텐츠 소유자는 자신의 ID와 패키징 하여 업로드 할 콘텐츠 M, 콘텐츠 정보 M_{info}, 자신의 서명 값 Sig_{CO}를 미디어 클라우드에게 전송하여 콘텐츠 등록을 시작한다.

• 미디어 클라우드는 자신의 ID와 전송 받은 M_{info}에 명시된 콘텐츠 소유자가 원하는 패키징을 할 수 있도록 Proxy에게 패키징 요청을 한다.

• Proxy는 M_{info}에 명시된 패키징을 수행 할 수 있는 CPS에 M_{info}와 M을 전송하여 패키징 요청을 한다.

• Proxy는 패키징일 수행 하며, 이때 필요한 콘텐츠 암호화 키 CEK를 KMS에게 요청한다.

• KMS는 암호학적으로 안전한 대칭키 기반으로 CEK를 생성하여 저장 한 후 CPS에게 전송을 한다.

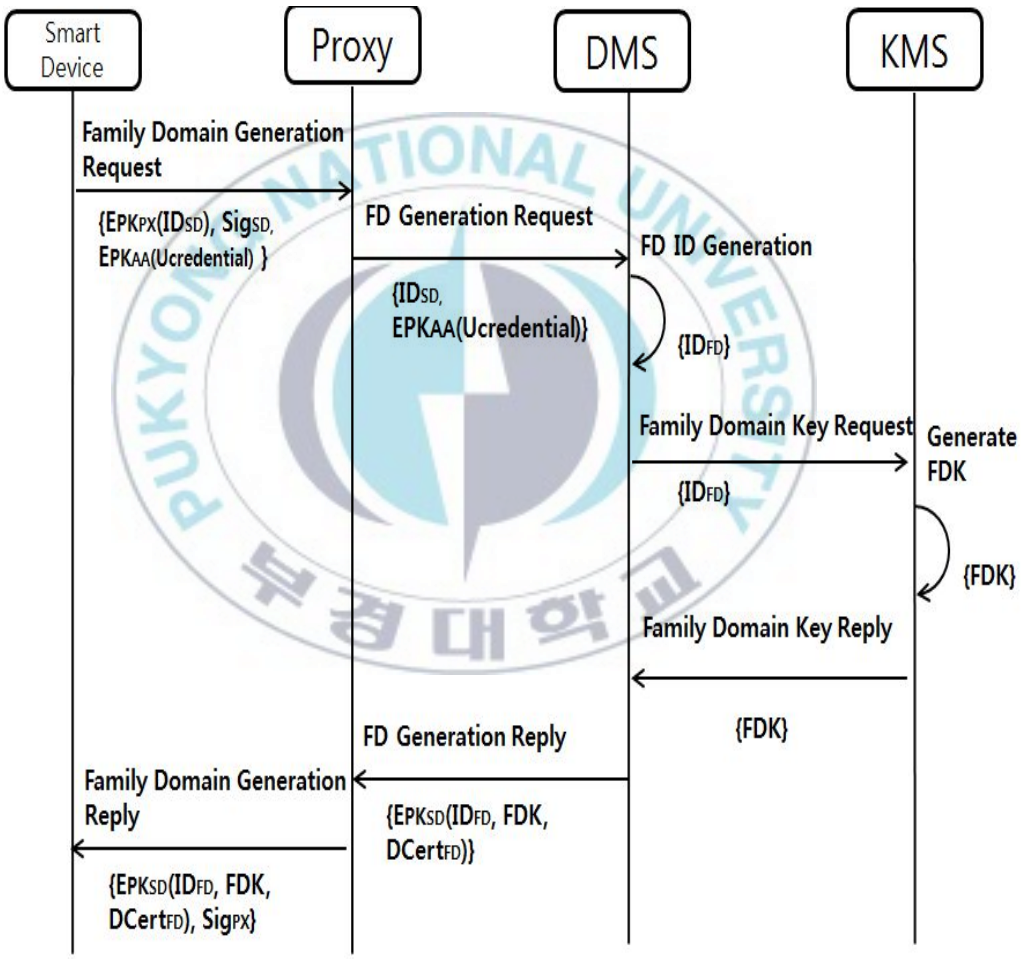
• CEK를 전송 받은 CPS는 M을 암호화하여 패키징을 완료하여 C를 생성한다. 표기를 하면 $C = E_{CEK}(M)$과 같이 C를 생성한다.

• 패키징된 콘텐츠 C와 C의 해쉬 값 $H(C)$를 미디어 클라우드에 등록하고 콘텐츠 등록 정보 C_{info}를 콘텐츠 소유자에게 전송하여 콘텐츠 등록 절차를 마친다.

2.2. 도메인 생성 절차

도메인은 콘텐츠에 접근 할 수 있는 장치들의 그룹을 뜻하며 제안하는 시스템에서는 도메인을 두 가지 종류로 분류를 한다. 하나는 FD(Family Domain)로서 도메인 그룹에 장치를 가입 및 탈퇴를 할 수 있는 관리를 하며 도메인 그룹 내의 최상위 개체이다. 다른 하나는

PD(Personal Domain)으로 도메인에 가입이 되어있으며, FD의 하위 개체이다. FD와 PD는 접근 구조에 명시된 접근 정책에 따라 콘텐츠를 사용 할 수 있는 서비스가 달라진다. 예를 들면, 접근 구조에서 FD는 장치들에게 콘텐츠를 재 배포 할 수 있지만 PD는 다른 장치들에게 콘텐츠를 재 배포 할 수 없다고 명시 되어있다면 PD는 다른 장치들에게 콘텐츠를 재 배포 할 수 없다.



< 그림 5 > 도메인 생성 절차

여러 스마트 장치를 사용하지 않고 하나의 스마트 장치를 사용하거나 도메인 그룹이 불필요한 사용자 같은 경우에는 도메인 생성 및

가입 절차를 하지 않더라도 DRM Cloud 서비스를 이용할 수 있다. 도메인 생성 절차는 FD가 하게 되며 <그림 5>과 같은 절차를 따르며 자세한 설명은 아래와 같다.

• 도메인을 생성하고 싶은 스마트 장치는 Proxy에게 Family 도메인 생성 요청을 한다. 이때 자신의 ID_{SD}를 Proxy의 공개 키 PK_{PX}로 암호화 하여 전송하며 인증과 권한을 부여하는 신뢰 할 수 있는 기관 AA에게 얻은 자격 증명 $Ucredential$을 전송하여 자신의 자격을 증명한다.

• Proxy는 DMS에게 자격 증명 $Ucredential$과 스마트 장치의 ID_{SD}를 전송한다. DMS는 자격 증명을 확인한 후 Family 도메인 ID ID_{FD}를 생성하고 저장을 한다.

• DMS는 KMS에게 ID_{FD}를 보내어 Family 도메인 키 FDK 생성을 요청한다. KMS는 대칭키 방식의 FDK를 생성하여 저장한 후 DMS에게 전송을 한다.

• DMS는 FDK와 ID_{FD}, 도메인 인증 서 $DCert_{FD}$를 도메인 생성 요청한 스마트 장치의 공개키 PK_{SD}로 암호화하여 Proxy에게 전송하며 전달받은 Proxy는 스마트 장치에게 전송하여 도메인 생성 절차를 마친다. 도메인 인증서 $DCert_{FD}$는 스마트 장치의 DRM 에이전트가 네트워크에 연결이 될 때 또는 스마트 장치가 도메인에 등록 및 삭제될 때에 주기적으로 업데이트된다. 도메인 인증서를 이용하여 도메인에 가입되지 않은 스마트 장치가 서비스를 이용하거나, 인증서가 더

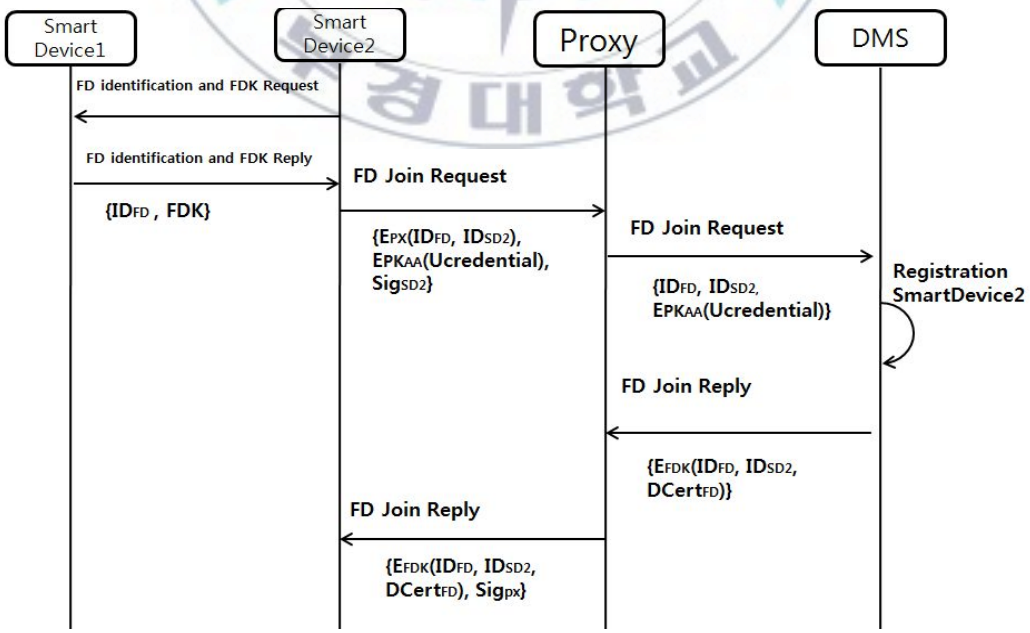
이상 유효하지 않게 되는 경우 즉각적으로 서비스를 거부 하도록 한다. 도메인 인증서는 다음과 같이 구성이 되어있다.

$$DCert_{FD} = \{SerialNumber, ID_{FD}, ID_{SD}, Cert_{CNT}, Sig_{DMS}\}$$

$SerialNumber$는 도메인 인증서 번호를 나타내며, ID_{FD}는 도메인의 ID, ID_{SD}는 도메인에 등록된 스마트 장치의 ID이며 도메인에 가입된 모든 스마트 장치의 ID가 들어간다. $Cert_{CNT}$는 스마트 장치가 도메인에 가입 또는 탈퇴로 인해 도메인 인증서가 갱신되는 횟수를 카운터로 표시되는 것이며, Sig_{DMS}는 DMS의 서명 값이다.

2.3. 도메인 가입 절차

생성된 도메인 그룹에 가입을 원하는 스마트 장치가 있을 때 도메인 가입 요청을 하는 절차로서 < 그림 6 >은 이러한 도메인 가입 절차를 나타낸다.



< 그림 6 > 도메인 가입 절차

• 스마트 장치1이 생성한 도메인에 가입을 원하는 스마트 장치2는 스마트 장치1에게 스마트 장치1이 생성한 도메인 ID와 그 도메인의 도메인 키 FDK를 요청한다.

• 스마트 장치1은 ID_{FD}, FDK를 스마트 장치2에게 전송을 해주며 전송받은 스마트 장치2는 Proxy에게 도메인 가입 요청을 한다.

• 스마트 장치2는 Proxy에게 자신의 ID ID_{SD2}와 자신이 가입하고 싶은 도메인의 ID ID_{FD}, 자신의 자격 증명을 나타내는 $Ucredential$을 전송하며 Proxy는 DMS에게 자격 증명 $Ucredential$을 확인하도록 하며, 자격 증명을 마친 DMS는 스마트 장치2의 ID를 도메인에 등록을 한다.

• DMS는 FDK로 업데이트 된 도메인 인증서 $DCert_{FD}$, 도메인 ID ID_{FD}, 스마트 장치2 의 ID_{SD2}를 암호화하여 Proxy에게 전송을 해주며 Proxy는 스마트 장치2에게 다시 전송을 한다.

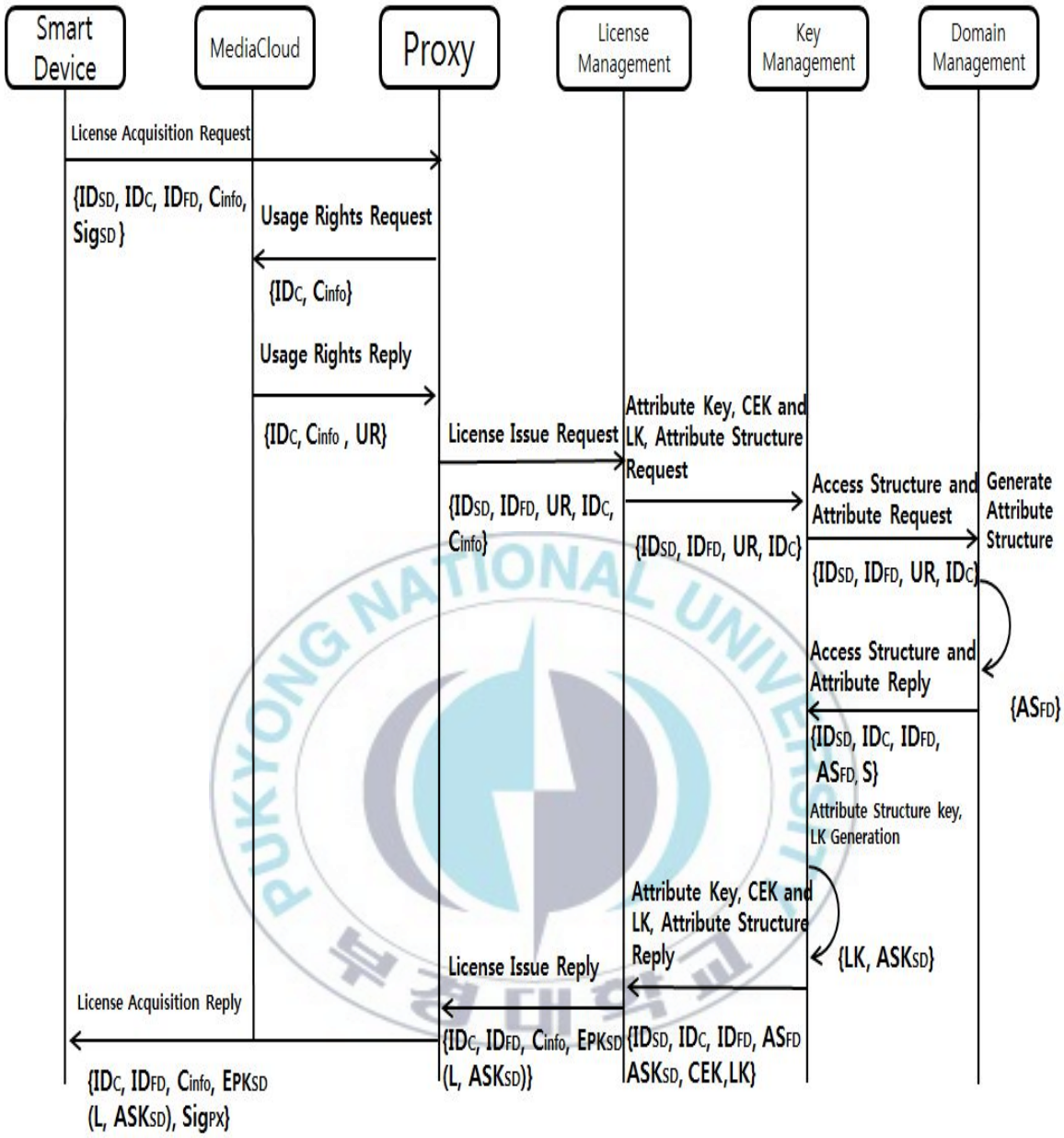
• 전송 받은 스마트 장치2는 도메인 관련 서비스를 이용하기 위해서는 도메인 인증서 $DCert_{FD}$가 필요하지만 FDK로 암호화 되어있기 때문에 FDK가 필요하다. 스마트 장치2는 스마트 장치 1에게 신뢰 할 수 있는 장치이기 때문에 FDK를 받았을 것이며 FDK가 있다면 암호화된 도메인 인증서 $DCert_{FD}$를 복호화 하여 얻을 수 있을 것 이다.

2.4. 라이선스 획득 절차

미디어 클라우드로 부터 콘텐츠를 구입한 사용자는 스마트 장치에서 콘텐츠를 사용하기 위해서는 콘텐츠 암호화 키 CEK를 필요로 한다. CEK는 라이선스에 포함되어 있으며 또한, 콘텐츠 사용 기간이나 사용 횟수 등 콘텐츠에 대한 사용 권한들이 속성 구조로 명시가 되어 있다. 이러한 속성 구조를 만족 하며, 콘텐츠 암호화 키 CEK를 획득할 수 있어야만 콘텐츠를 사용 할 수 있게 된다. 라이선스는 REL(Rights Express Language)구조로 작성이 되어져 있으며 이러한 라이선스를 획득 하는 절차는 < 그림 7 >와 같으며 자세한 절차는 아래와 같다.터

• 미디어 클라우드로부터 원하는 콘텐츠를 다운로드 받은 스마트 장치는 에이전트를 통해 Proxy에게 콘텐츠 정보 C_{info}와 ID_{SD}, ID_C , ID_{FD} 를 전송 하여 C에 대한 라이선스 획득 요청을 한다. 이때 스마트 장치는 도메인에 가입이 되어있으며 도메인에서 FD의 위치에 있다고 가정한다. 만약 PD의 위치에 해당하는 스마트 장치가 콘텐츠를 구입 하였다면, 라이선스 획득 절차에서 FD와 같은 역할로 서비스를 이용 할 수 있다.

• 요청을 받은 Proxy는 미디어 클라우드에게 스마트 장치가 구입한 C에 대한 도메인 ID_{FD}이 사용 할 수 있는 사용 권한 UR을 요청한다. UR은 정책으로 나타내어 있으며 자세한 내용은 2.4.1항의 사용권한 및 접근구조에서 설명을 한다.



< 그림 7 > 라이선스 획득 절차

• *UR*을 전송 받은 Proxy는 LMS에게 라이선스 배포 요청을 하게 되며 LMS는 라이선스에 포함할 콘텐츠 암호화 키 *CEK*, *CEK*를 암호화 하여 보호할 *LK*, 라이선스 작성에 필요한 속성 구조 AS_{FD},

사용자에게 배포 할 속성 비밀 키 ASK_{SD}를 요청한다.

• 요청 받은 KMS는 DMS에게 UR을 전송하게 되며, DMS는 전송 받은 UR과 소유하고 있는 속성 집합 S를 가지고서 Family Domain의 접근 구조 AS_{FD}를 생성하고 저장을 한다. 속성 집합 S는 [표 2]와 같이 나타낼 수 있다.

• DMS는 KMS에게 AS_{FD}를 전송하게 되며, KMS는 CP-ABPRE의 $\mathbf{Setup}(1^k)$ 알고리즘을 실행하게 된다. Setup 알고리즘 실행 후 KMS는 $\mathbf{KeyGen}(MK, S)$ 알고리즘을 실행하여 스마트 장치의 속성 비밀 키 ASK_{SD}를 생성하며, CEK를 보호하기 위한 LK를 생성하고 저장을 하게 된다. 이때 LK는 두 개의 키로 분할하여 하나는 LMK, 다른 하나는 AK의 구조로 분할한다. 즉, $LK = LMK + AK$와 같다.

• 접근 구조 AS_{FD}와 속성 비밀 키 ASK_{SD}, 콘텐츠 암호화 키 CEK, CEK를 보호하기위한 LK, LK의 분할 한 형태의 키 LMK 를 전송받은 LMS는 라이선스 L를 생성하게 된다. 라이선스 L을 생성하기 전에 CP-ABPRE의 $\mathbf{Encrypt}(PK, LMK, AS_{FD})$ 알고리즘을 실행하여 LMK'을 생성한다. 그리고 CEK를 LK로 대칭키 방식의 암호화를 수행하여 CEK'을 생성한다. 즉, $CEK' = E_{LK}(CEK)$이다. 암호화를 수행한 LMS는 라이선스 L을 생성하게 되며 구성요소로는 ID_C , ID_{FD}, ID_{SD}, LMK', CEK', AS_{FD} 및 기타 정보로 이루어지며 라이

선스 L은 REL의 준수에 맞도록 작성되며 2.4.2항의 라이선스 작성에서 자세하게 다룬다.

라이선스는 $L=\{ID_{FD},ID_{SD},LMK',CEK',AS_{FD},etc\}$와 같이 구성된다.

라이선스는 도메인에 가입한 모든 개체들 간에 배포를 하여도 상관없다. L은 AS_{FD}를 포함한 형태로 생성하여 하나의 도메인 내의 모든 개체들의 권한이 명시가 되어있는 형태로서 라이선스를 어느 개체가 사용하더라도 개체가 가지고 있는 속성 비밀 키의 속성 값마다 할 수 있는 권한이 다르기 때문이다.

• 라이선스를 생성한 LMS는 스마트 장치의 공개키 PK_{SD}로 라이선스와 속성 비밀 키 ASK_{SD}를 암호화 하여 Proxy에게 전송하며 전송 받은 Proxy는 스마트 장치에게 다시 전송을 하여 라이선스 발급 절차를 마친다.

2.4.1. 사용 권한 및 접근 구조

라이선스는 세분화된 접근제어를 위해 속성 구조로 작성된다. 속성 구조 AS는 속성들의 집합 S에서 속성들을 선택하여 사용 권한 UR 을 적용하여 만들어진 구조이다. 속성은 [표 2]처럼 나타 낼 수 있다. 속성은 여러 개의 값들이 부여 될 수 있으며, 이러한 속성 집합 S는 도메인 생성을 하게 될 때에 만들어지고 미디어 클라우드에게 DMS가 전송을 하여 서로 공유를 하고 있다고 가정한다.

[표 3]은 사용 권한 UR을 나타내는 예시이며 boolean 수식으로 표현 되어져있다.

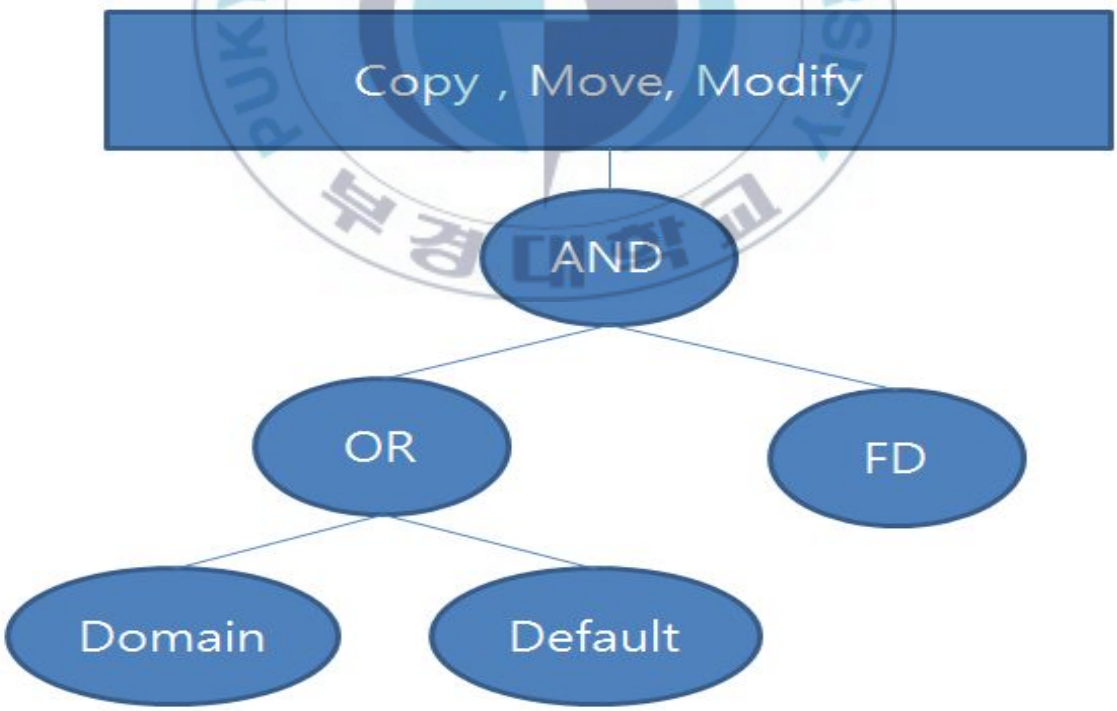
[표 2] 속성들의 예

속성	값	내용
Domain	*LACUC*	도메인 이름
MC	*MediaProvider*	미디어 클라우드 이름
Content	*Content1*	콘텐츠 이름
FD	*SmartTV*	도메인에 속한 개체들을 관리 할 수 있는 상위 개체
PD	*SmartPhone*	도메인에 속한 개체로서 FD의 하위 개체
Default	*NoName*	도메인 미 가입 개체
Count	*1~99*	횟수를 나타내는 정수

콘텐츠를 사용 할 수 있는 여러 작업마다 하나의 정책이 부여되며 정책이 명시된 수식을 만족하는 속성을 가진 스마트 장치만이 해당하는 작업을 할 수 있도록 표현된 것이다. <표 3>의 P_1의 경우를 설명하면, *Domain*이 *LACUC*이거나 *Default* 이고, *FD*가 *SmartTV*일 때 *MC*가 *MediaProvider* 인 *Content1*을 복사 및 이동 할 수 있음을 표현한 사용 권한 정책이다.

[표 3] 사용 권한 정책의 예

정책	정책 명시	작업
P_1	((*Domain=LACUC*) or *Default*) and *FD=SmartTV* and *MC=MediaProvider*	복사 및 이동
P_2	((*Domain=LACUC*) or *Default*) and (*FD=SmartTV* or *SD=SmartPhone*) and *MC=MediaProvider*	재생
P_3	((*Domain=LACUC*) or *Default*) and *FD=SmartTV* and *MC=MediaProvider*	수정
P_4	*(Domain=LACUC)* and *(FD=SmartPhone)* and *(Count<10)*	출력



< 그림 8 > 접근 구조의 예

접근 구조는 사용 권한 정책에 명시된 내용을 ANG-gate 형태로 표현한 것이며 이러한 접근 구조를 만족하는 속성 비밀 키를 가진 스마트 장치만이 해당하는 작업을 수행 할 수 있다. <그림>을 보면 Domain과 FD 두 개의 속성을 가진 스마트 장치만이 복사 및 이동과 수정 작업을 할 수 있음을 보여준다. 만약 FD 속성이 아닌 PD 속성을 가진 사용자는 위의 < 그림 8 >에 해당하는 접근 구조를 만족하지 않으므로 복사 및 이동과 수정이라는 작업을 할 수 없는 것이다.

```
<o-ex: rights
  xmlns: o-ex="http://odr1.net/1.1/ODRL-EX"
  xmlns: o-dd="http://odr1.net/1.1/ODRL-DD"
  <o-ex:id="ExampleLicense">
  <o-ex:context>
    <o-dd:version>1.0</o-dd:version>
    <o-dd:uid>ExamplePolicy</o-dd:uid>
  </o-ex:context>
 <o-ex:agreement>
    <o-ex:permission>
      <o-dd:copy>
        <o-ex:constraint>
          <o-dd:Domain>LACUC</o-dd:Domain>
            <o-dd:FD>SmartTV</o-dd:FD>
         </o-ex:constraint></o-dd:copy>
       <o-dd:review>
        <o-ex:constraint>
          <o-dd:Domain>LACUC</o-dd:Domain>
            <o-dd:PD>Smartphone</o-dd:PD>
         </o-ex:constraint></o-dd:review>
```

< 그림 9 > 라이선스 작성 예

2.4.2. 라이선스 작성

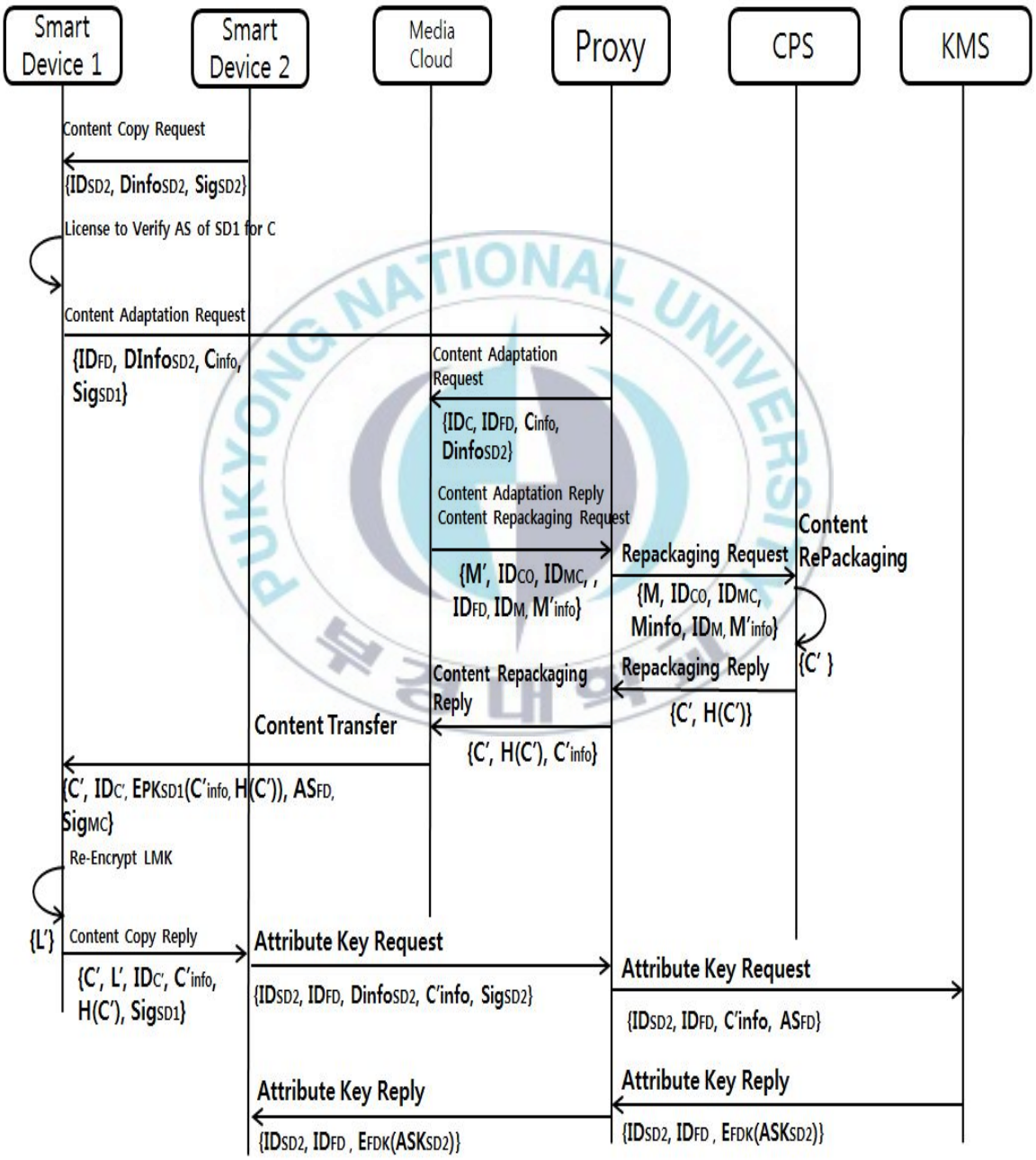
속성 구조 정책을 포함한 라이선스는 OMA의 DRM REL[10]의 규격으로 작성된다. 위의 < 그림 9 >은 라이선스 작성 예를 보여준다.

< 그림 9 >에서 <o-ex:permission>은 접근 권한을 나타내며 여기에 작성된 부분은 속성을 가져와서 작성되었다. 또한, 라이선스 내에 boolean 수식 및 부등호를 포함하여 작성이 될 수 있다. <o-dd:copy>는 콘텐츠에 대해 복사를 할 수 있는 권한을 명시한 부분으로 Domain 속성의 값이 LACUC이며 FD의 속성이 SmartTV인 속성을 가진 사용자만이 copy라는 행위를 할 수 있음을 나타낸다. <o-dd:review>는 미리보기를 뜻하는 행위로서 Domain 속성의 값이 LACUC 이면서 PD의 속성이 SmartPhone인 사용자만이 미리보기 행위를 할 수 있음을 나타낸다. 이렇게 속성의 값들을 REL의 구조에 맞도록 매핑을 하여 라이선스를 작성한다.

2.5. 콘텐츠 재 배포 및 재 패키징 절차

미디어 클라우드로부터 다운로드 받은 콘텐츠를 스마트 장치가 다른 스마트 장치에게 재 배포를 하는 서비스 절차로서 콘텐츠를 재 배포 받을 스마트 장치는 도메인에 가입을 해야 한다. 이때, 가입하는 도메인은 스마트 장치끼리 같은 도메인에 가입이 되어있어야 한다. 콘텐츠를 재 배포를 받기 원하는 콘텐츠가 스마트장치의 DRM에 적합하지 않은 패키징이 되어있을 수 있으므로 재 패키징 요청을 통해 재 패키징을 한 후에 콘텐츠를 재 배포 받을 수 있도록 한다. 또한, 재 배포를 하게 되면 라이선스를 재 발급 받아야하지만 라이선스를 서버로부터

재 생성하여 전송을 받지 않고 콘텐츠를 재 배포해주기 위한 스마트 장치에서 라이선스를 전송한다. 이때 전송하는 라이선스에 포함된 LMK'를 CP-ABPRE의 Reencrypt 알고리즘을 통해 재 암호화 하여 전송한다. 이러한 절차는 < 그림 10 >에 나타나있다.



< 그림 10 > 콘텐츠 재 배포 및 재 패키징 절차

• 스마트 장치2는 스마트 장치1에게 자신의 정보 $Dinfo_{SD2}$를 전송하며 콘텐츠 재 배포 요청을 한다. 스마트 장치1은 스마트 장치2에게 재 배포 작업을 하기 전에 DRM 에이전트로 자신의 라이선스에 콘텐츠 재 배포 작업을 수행 할 수 있는지 접근 구조를 통해 확인을 하며, 수행 할 수 있는 권한이 있는 경우에 $Dinfo_{SD2}$를 에이전트를 통해 Proxy에게 전송하여 재 패키징 요청을 하게 된다.

• Proxy는 미디어 클라우드에게 수정 된 콘텐츠 M'을 요청하게 되며 미디어 클라우드는 수정 된 콘텐츠 M'와 정보 M'_{Info} 를 같이 Proxy에게 전송한다.

• Proxy는 CPS에게 재 패키징을 요청하게 되고 CEK를 가지고 암호화를 하여 수정 된 콘텐츠 C'을 생성한다. 즉, $C' = E_{CEK}(M')$

C'의 해쉬값 $H(C')$을 생성하여 Proxy에게 전송을 하고 Proxy는 미디어 클라우드에게 재 패키징 된 C'와 그 정보 C'_{info} 및 속성 구조 AS_{FD}를 미디어 클라우드에게 보낸다.

• 미디어 클라우드는 스마트 장치1에게 재 패징 된 콘텐츠 C', 콘텐츠의 ID $ID_{C'}$, 콘텐츠 정보 C'_{info}과 그 해쉬 값 $H(C')$, 라이선스에 명시 한 접근 구조 AS_{FD}를 스마트 장치1에게 전송을 한다.

• 스마트 장치1은 스마트 장치2에게 콘텐츠 C'와 라이선스 L을 전송하기 전에 CP-ABPRE의 $\mathbf{RKGen}(ASK_S, AS)$를 실행하여 재 암호

화 키 $RK_{S \to AS}$를 생성하게 된다. 재 암호화 키를 생성하고 난 후, Reencrypt$(RK_{S \to AS}, LMK')$ 를 실행하여 LMK'을 재 암호화 한다. 재 암호화하는 과정에서 속성 구조를 추가 확장을 할 수 있다. 예를 들면, 콘텐츠를 가지고서 할 수 있는 수행의 횟수를 다시 작성 할 수 있다. 재 암호화를 통해 생성된 라이선스 L'와 C'_{info} 및 $H(C')$을 스마트 장치2에게 전송하여 재 배포 및 재 패키징 단계를 마친다.

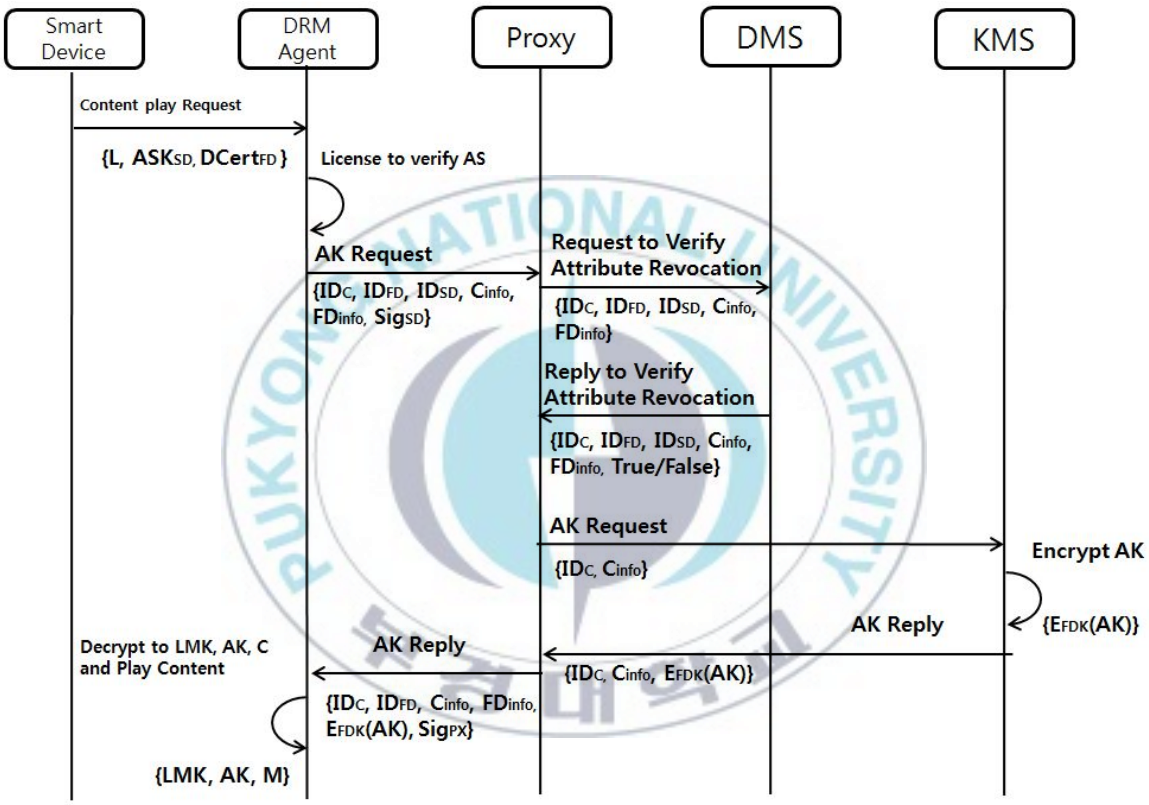
제안하는 방식에서 라이선스는 AS_{FD}라는 하나의 개체를 위한 라이선스가 아니라 도메인에 가입된 모든 개체에 대한 접근 구조를 가지고 있는 라이선스이기 때문에, 도메인에 가입 된 모든 스마트 장치에게 배포를 할 수 있다. 하지만, 어떠한 수행에 대해 접근 구조를 다시 작성해야 할 수도 있으며, 콘텐츠에 대한 사용 기간이 라이선스에 명시 되지 않거나 아주 긴 기간 동안 사용 할 수 있을 경우에는 콘텐츠 암호화 키가 누출 될 수 있기 때문에 재 암호화를 한다.

• 라이선스와 재 패키징 된 콘텐츠를 전송 받은 스마트 장치2는 Proxy에게 자신의 속성 비밀 키 요청을 하게 되며, Proxy는 KMS에게 스마트 장치2의 속성 비밀 키 ASK_{SD2} 요청한다. 요청받은 KMS는 FDK로 ASK_{SD2}를 암호화하여 Proxy에게 전송을 해주며 Proxy는 스마트장치2에게 다시 전송을 하여 속성 비밀 키 전송 단계를 마친다.

• 콘텐츠 재 배포 서비스를 이용하게 될 경우, 재 패키징을 하지 않아도 된다면 스마트 장치1 은 AS_{FD}를 미디어 클라우드로부터 전송받아 L'를 생성하는 단계부터 시작한다.

2.6. 콘텐츠 사용 절차

콘텐츠와 라이선스를 소유하고 있는 스마트 장치가 콘텐츠를 사용하게 될 경우의 절차를 설명한다. < 그림 11 >은 이러한 절차를 보여준다.



< 그림 11 > 콘텐츠 사용 절차

• 콘텐츠를 사용하기위해 스마트 장치는 에이전트에게 라이선스 안의 속성 구조에 따른 사용 권한 검증을 받는다. 이때, 속성 비밀 키의 속성으로 사용 할 수 없거나 라이선스 만료로 인해 사용 할 수 없을 경우에는 서비스 사용을 거부한다. 검증을 통해 사용 할 수 있다면 LK의 분할된 키 AK요청을 한다.

• 검증을 한 에이전트는 AK를 요청하게 되며 Proxy는 DMS에게 해당하는 속성 비밀 키가 폐지되었는지 확인요청을 한다. 속성 비밀 키는 이미 스마트 장치에 저장이 되어있으므로 유효하지 않은 속성 비밀 키 이거나 폐지 된 속성 비밀 키 인지 판별하기 위해 DMS에게 확인 요청을 한다.

• 확인을 하여 True/False 보내게 되며 False 일 경우 유효하지 않은 비밀 키 이므로 즉각적으로 서비스를 거부하는 메시지를 에이전트에게 보내어 서비스 접근 거부를 한다. True 일 경우 Proxy는 KMS에게 AK요청을 한다.

• KMS는 AK를 FDK로 암호화하여 AK'를 생성한다. 만약 도메인에 가입하지 않은 스마트 장치이면 스마트 장치의 공개키로 암호화 한다. 즉, $AK' = E_{FDK}(AK)$ or $E_{PK_{SD}}(AK)$

• KMS는 Proxy에게 AK'을 전송하며 Proxy는 AK'을 다시 스마트 장치에게 전송을 한다.

• AK'을 전송받은 스마트 장치는 자신의 FDK로 AK'를 복호화하여 AK을 획득하며 CP-ABPRE의 $\mathbf{Decrypt}(LMK', ASK_{SD})$를 실행하여 LMK를 획득한다. LMK와 AK를 합쳐 LK를 생성하게 되며 LK를 가지고 암호화된 CEK를 복호화하여 C를 획득한 후 CEK로 C를 복호화하여 M을 획득한다. 그러고 획득한 콘텐츠 M을 사용한다.

$$D_{FDK}(AK') = AK,\ D_{ASK_{SD}}(LMK') = LMK,$$

$$LK = LMK + AK$$

$$CEK = D_{LK}(CEK'),\ M = D_{CEK}(C)$$

재 배포 받은 콘텐츠일 경우라도 똑같은 절차를 통해 콘텐츠를 사용한다.

IV. 분석 및 구현

CP-ABPRE를 활용 한 DRM Cloud의 보안성 분석과 다른 DRM 방식과의 비교를 해보며 실제로 키 생성, 암호화, 복호화 및 콘텐츠 재생에 대한 구현을 설명한다.

1. 보안성 분석

DRM Cloud에 적용 한 CP-ABPRE의 알고리즘의 안전성은 [6]에서 이미 검증이 되었기 때문에 본 논문에서는 설명을 하지 않는다.

기존에 클라우드에서의 속성 기반 프록시 재 암호화 연구 논문 [15]에서는 이상적인 속성기반 재 암호화 기법의 기준에 대해 설명을 하였으며, 기준은 아래와 같다.

- 일방향(Unidirectionality) : 클라우드는 암호화된 데이터 CT를 CT'으로 재 암호화 할 수 있지만 CT'에서 CT로 바꿀 수 없다.
- 데이터 기밀성 : 데이터 소유자는 데이터를 클라우드에 업로드 하기 전에 암호화 하여 업로드를 하며, 암호화된 데이터는 허가된 사용자에 의해 복호화 된다. 클라우드를 포함하여 허가받지 않은 당사자는 암호화 된 데이터에 대한 정보를 얻을 수 없다.
- 비-상호작용(Non-interactive) : 데이터 소유자는 소유자 자신이 재 암호화 키를 생성하며, 클라우드를 포함한 신뢰 할 수 없는 제 3의 개체를 필요로 하지 않는다.
- 비-이행성(Non-transitive) : 클라우드가 두 개의 재암호화 키 $rk_{A\to B}, rk_{B\to C}$를 가지고 있더라도 두 개의 키를 가지고서 $rk_{A\to C}$ 를 생

성 할 수 없어야한다.

• 다중 사용(Multi-use) : 암호화된 데이터는 하나의 사용자에 의해 재 암호화하여 다른 사용자에게 전송할 수 있으며, 전송받은 사용자 또한 재 암호화하여 다른 사용자에게 전송 할 수 있어야한다.

• 재 암호화 제어 : 데이터 소유자는 암호화 된 데이터를 재 암호화하거나 재 암호화를 원하지 않는다면 클라우드가 재 암호화 할 수 없어야한다.

• 마스터 키 안전성 : 데이터 사용자는 클라우드와 결탁하여 데이터 소유자의 마스터 키를 획득 할 수 없어야한다.

• 결탁 공격 방지 : 폐지 된 사용자는 클라우드와 결탁하여 암호화 된 데이터를 사용 할 수 없어야한다.

제안하는 모델에서는 일방향, 비-이행성, 다중 사용, 재 암호화 제어, 마스터 키 안전성을 [6]의 알고리즘에서 동일한 기능을 가지고 있으므로 기준에 만족을 한다. 또한, 결탁 공격 방지는 아래에서 자세하게 설명을 하며 결탁공격도 만족을 한다. 하지만, 데이터 기밀성, 비-상호작용을 만족하지 않는다. 제안하는 모델에서 데이터 소유자는 콘텐츠 원본을 미디어 클라우드에게 제공을 하게 되는데, 이러한 이유는 상호운용성을 위해 패키징을 DRM Cloud에서 해야 하기 때문이다. 비-상호작용은 데이터 소유자가 클라우드로부터 자신의 데이터의 기밀성을 유지하기 위해 재 암호화키를 사용하여 암호화된 콘텐츠의 키를 재 암호화 하여 사용자에게 배포를 할 경우 해당하지만 제안하는 모델에서는 이러한 방식을 사용하지 않으므로 고려하지 않는다.

보안성 분석의 위협 모델에서 사용자는 악의적이며, 콘텐츠에 대해 승인되지 않은 접근을 시도한다고 가정한다. 공격자는 DRM 에이전

트를 훼손하거나, 스마트 장치의 파일 시스템을 무단으로 수정을 하여 콘텐츠를 무단으로 재 배포 할 수 있다. 콘텐츠를 재 배포를 하더라도 제안한 시스템에서는 라이선스내의 속성 비밀 키가 있어야만 라이선스 마스터 키를 획득 할 수 있으며, 보조 키를 서버로부터 전송받아야 암호화된 콘텐츠를 복호화 할 수 있다. 속성 비밀 키를 획득하기 위해서는 먼저 결탁 공격을 할 수 있다. 또한, 무단 배포를 받은 스마트 장치는 정당하게 도메인에 가입되어 있지 않거나 정당한 속성 비밀 키를 가지고 있지 않으므로 사용자/속성 폐지문제가 있을 수 있으며, 시스템의 프로토콜에 어떠한 훼손을 하거나, 도청을 통해 정보를 획득하기 위해 재전송 공격, 중간자 공격을 시도할 수 있다. 이러한 위협을 방지하기 위해 결탁 공격, 사용자/속성 폐지, 재전송 공격, 중간자 공격에 대해 시스템이 안전한지 분석을 한다.

2.1. 결탁 공격방지

결탁 공격은 사용자 혼자서 콘텐츠를 복호화 할 수 없지만 여러명의 사용자가 서로 결탁하여 서로간의 속성을 조합하여 암호화된 콘텐츠를 복호화 할 수 있음을 말한다.

제안하는 방식에서 콘텐츠 사용을 위해 하나의 도메인 내에 여러 개체들이 결탁하여 하나의 거짓 속성 비밀 키를 만들어 사용하게 되더라도 도메인 인증서에 등록되지 않은 스마트 장치는 에이전트로부터 Proxy에게 도메인 개체 등록 검증을 통해 서비스 거부를 당하거나, 라이선스의 속성 구조를 검증하여 적합하다고 판별 되더라도 Proxy는 도메인 정보 FD_{info}, 스마트 장치 ID ID_{SD}, 콘텐츠 ID ID_C ,를 통해 DMS에 등록된 도메인의 장치가 자신이 소유하고 있는 속성 비밀 키

로 콘텐츠를 사용하는지 검증을 하므로 결탁 공격을 방지 할 수 있다.

2.2. 사용자/속성 폐지

DRM에서 속성 기반 암호화 방식을 적용하게 될 때의 큰 문제는 여러 사용자들에게 속성이 소유되어있는 특성 때문에 속성과 사용자 폐지가 어렵다는 점이다. 기존에는 주기적으로 콘텐츠를 재 암호화 하거나 폐지되지 않은 사용자들의 비밀 키를 새로 생성하는 방식을 사용하였다.

본 논문에서는 Proxy에게 폐지 검증을 하여 즉각적으로 서비스 거부를 할 수 있도록 권한을 주었으며, 키를 두 개로 분할하는 방식을 사용하였다. 속성 비밀 키가 폐지되었을 경우에는 스마트장치는 이미 속성 비밀 키를 가지고 있는 상태이다. 라이선스에 적합한 속성 비밀 키를 가지고 있기 때문에 접근 구조로부터 거부는 되지 않지만 에이전트가 AK를 요청을 하게 되었을 때 DMS가 속성이 폐지되었는지를 즉각적으로 판별하여 AK요청을 거부하여 더 이상 서비스를 사용할 수 없도록 한다.

사용자 폐지의 경우 사용자인 스마트 장치는 도메인 인증서를 가지고 있으며, 도메인 인증서는 네트워크에 연결되거나 도메인에 장치가 새로 가입하거나 떠나게 될 때에 업데이트가 되므로 도메인 인증서에 등록되지 않은 장치나 도메인 기간 등 유효하지 않은 장치가 접근을 하게 될 경우 즉각적으로 서비스를 거부한다. 이러한 두 가지 방식을 이용하여 속성 비밀 키 및 사용자 폐지 문제를 해결 하였다.

2.3. 재전송 공격

재전송 공격은 프로토콜 상에서 유효 메시지를 도청 및 습득한 후 나중에 재전송하여 정당한 사용자로 가장하거나, 유효한 정보를 얻을 수 있는 공격이다.

스마트 장치가 미디어 클라우드로부터 콘텐츠를 다운로드 받은 후에 일어나는 서비스 절차들에서 재전송 공격에 대한 분석을 한다.

3장 2.4 절의 라이선스 획득 절차에서는 스마트 장치, 미디어 클라우드 및 Proxy간에 통신을 하게 될 경우 안전하지 않은 통신 채널을 사용한다. 스마트 장치가 Proxy에게 라이선스 획득 요청을 하게 될 때에 스마트 장치의 서명 값 Sig_{SD}를 같이 전송하게 되며 Proxy는 서명 값을 검증을 통해 스마트 장치로부터 전송되었음을 확인 할 수 있다. 만약 서명 값 자체의 패킷을 캡쳐하여 시간이 지난 후에 보내더라도 라이선스와 속성 비밀 키를 암호화하는 공개키는 라이선스 획득 요청시에 스마트 장치의 ID를 같이 보내기 때문에 스마트 장치의 공개키로 암호화하여 전송한다. 또한, Proxy와 미디어 클라우드 사이에서도 서로간의 서명 값을 전송하여 통신을 하므로 라이선스 획득 절차에서는 재전송 공격에 대해 저항성을 가지고 있다.

3장 2.5 절의 콘텐츠 재 배포 및 재 패키징 절차에서는 스마트 장치1과 스마트 장치2, 미디어 클라우드, Proxy간에 통신을 한다. 스마트 장치1과 스마트 장치2는 서로 간에 신뢰하는 관계로 볼 수 있다. 이것은 서로 물리적인 장치로서 거리가 가까운 상태에서 이루어지기 때문에 서로 신뢰하는 관계이다.

스마트 장치1이 재 패키징 요청을 하게 될 때에도 라이선스 획득 절차처럼 자신의 ID와 서명 값을 보내기 때문에 라이선스 획득 절차와 똑같이 안전하다. 스마트 장치2가 Proxy에게 자신의 속성 비밀 키 요청을 하게 될 경우에도 안전하며, Proxy와 미디어 클라우드간의 경우에도 안전하다.

2.6 절의 콘텐츠 사용 절차에서 스마트 장치가 Proxy에게 AK요청을 하게 되는데 이때에도 ID와 서명 값을 보내면서 요청을 한다. 하지만 AK는 FDK로 암호화가 되어서 Proxy로부터 스마트 장치에게 전송이 되는데, 이때 FDK는 도메인 생성 및 도메인 가입 절차에서 스마트 장치가 획득하게 되는 키 이다. FDK와 도메인 인증서를 도메인 생성 절차에서 스마트 장치는 회득하게 되는데 스마트 장치는 자신의 ID를 Proxy의 공개키로 암호화하고 자신의 서명 값을 같이 전송하게 된다. 그리고 Proxy가 스마트 장치에게 FDK와 도메인 인증서를 전송하게 될 경우 스마트 장치의 공개키로 암호화를 하며, Proxy의 서명 값을 같이 보내기 때문에 FDK의 획득에 대해서 안전하다. 그러므로, 콘텐츠 사용 절차에서 $E_{FDK}(AK)$를 획득하더라도 FDK를 공격자는 얻을 수 없기 때문에 재전송 공격에 대해 안전하다.

2.4. 중간자 공격

중간자 공격은 통신을 하는 두 당사자 사이에 끼어들어 도청 및 습득을 통해 메시지를 변조하여 자신에게 유효한 정보를 획득하는 공격 방법이다.

본 논문에서 제안하는 모든 서비스 절차에서는 안전하지 않은 통신 채널을 사용 할 경우에는 모든 개체들 간에 ID 및 서명 값을 첨부하여 통신을 하는 것을 1.3 재전송 공격에서 설명하여 알 수 있으므로, Proxy에서 스마트 장치로 전송되는 메시지들을 공격자가 Proxy의 서명 값을 생성 할 수 없기 때문에 Proxy로부터 전송되는 메시지들은 안전하다.

콘텐츠 사용 절차에서 스마트 장치가 Proxy에게 AK요청을 보내게 될 경우 공격자가 Proxy로 위장을 하여 전송하는 메시지들을 획득하고 Proxy에게 정당한 스마트 장치 인 것처럼 위장하더라도 도메인 인증서가 없다면 정당한 스마트 장치가 아님을 판별 할 수 있으므로 제안하는 서비스 절차들에서는 중간자 공격이 안전하다.

2. 비교 분석

기존에 제안된 속성 기반의 DRM 폐지 방식과 비교를 하며 기존에 제안 된 방식들도 속성과 사용자 폐지가 가능한 형태였다. 본 논문에서 제안하는 방식에서는 즉각적으로 폐지를 판별 할 수 있으며 키를 업데이트할 필요가 없었다. 콘텐츠를 재 암호화 하지 않지만 LK를 재 암호화한다. 하지만 재 암호화와 같은 경우, 콘텐츠를 재 배포 하지 않으면 사용되지 않는다. 이러한 비교는 [표 4]에 나타나 있다.

[표 4] 기존 ABE의 폐지 방식 비교

방식	판별 속도	키 업데이트	콘텐츠 재 암호화
Yang[11]	만료	사용	사용
Ibraimi[12]	즉각	사용	미 사용
Yu[13]	즉각	사용	사용
제안하는 방식	즉각	미 사용	사용/미 사용

[표5]는 DRM에 관련하여 제안 된 방식 비교이다. 세분화된 접근 제어와 속성 폐지, 개인 정보보호 보존(Privacy preserving)에 대해 비교를 하였다. 본 논문에서 제안하는 방식에서는 개인 정보보호 보존을 적용하지 않는다. Proxy에게 구입한 콘텐츠 및 스마트 장치의 정보들이 전송되며 그러한 정보를 가지고서 서비스 절차들이 이루어지기 때문이다.

[표 5] 기존 DRM 관련 방식과 비교

방식	세분화된 접근 제어	속성/사용자 폐지	개인 정보보호 보존
Petrlic[14]	불 가능	미 적용	가능
Muller[15]	가능	미 적용	N/A
제안하는 방식	가능	적용	N/A

[표 6] 은 기존에 제안 된 ABPRE의 성능 비교이다[16]. 본 논문에서 제안한 방식은 Luo[6]가 제안한 방식을 적용하였기 때문에 Luo[6]가 제안한 방식과 같은 성능을 보인다. [표 6]에서 C_e는 페어링 연산, E는 지수 군 연산, M은 곱셈 군 연산, S는 서명 연산, n은 속성의 수이다. [표 6]에서는 추가적인 대칭키 암호화 연산은 제외하였다.

[표 6] ABPRE 성능 비교

항목	Liang[]	Luo[]	Yu[]	Seo[]
암호화	$(n+2)E + 2M$	$(n+2)E+2M$	$(n+1)E+2M$	$(n+2)E+2M$
복호화	$(n+2C_e) + 2M$	$(2n)C_e+3M$	-	$2C_e+(3n+2)E + 2M$
재 암호화	$(n+1)C_e + M$	$(2n+1)C_e+(n+1)M$	$(n)E$	$2C_e+(3n)E+M$
재 복호화	$(n+3)C_e + 4M$	$(2n+1)C_e+5M$	$(n+1)C_e+2M$	$3C_e+(3n)E+4M$

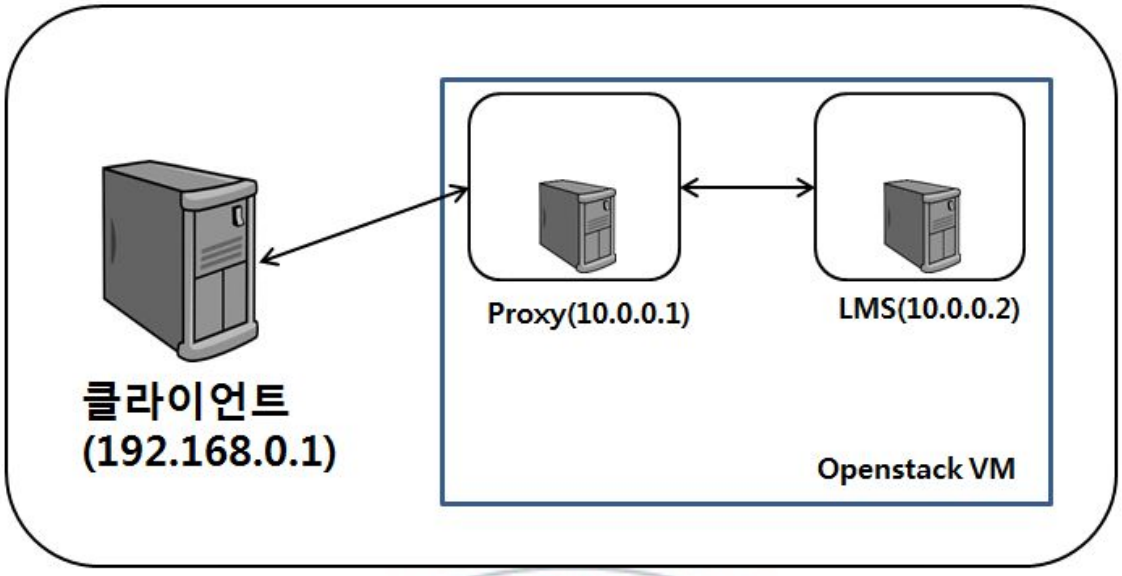
3. 구현

CP-ABPRE를 활용한 DRM Cloud를 테스트 환경에서 구현하여 클라우드에서 실제로 사용 할 수 있음을 보여준다.

3.1. 구현 환경

DRM Cloud의 Proxy, DMS, KMS, LMS, CPS들은 Openstack[14] 클라우드 서버 PC 1대에서 각자 VM으로 생성을 하여야 하지만, 구현환경이 부족한 점이 많아 LMS, Proxy 만을 VM으로 생성하였으며 스마트 장치는 일반 PC로 테스트 환경으로 구현하였다. 구현 환경은 아래와 같다.

- 클라우드 서버 프로세서 : intel i5-2400 3.10Ghz
- 클라우드 서버 메모리 : 512Mb
- 클라우드 서버 운영체제 : Ubuntu 14.04
- 클라이언트가 되는 PC 프로세서 : intel i5-2400 3.10Ghz
- 클라이언트가 되는 PC 메모리 : 4.1Gb
- 클라이언트가 되는 PC 운영체제 : Ubuntu 14.04
- 개발 언어 : Java
- 오픈 소스 : Openstack(devstack)
- 어플리케이션 : eclipse



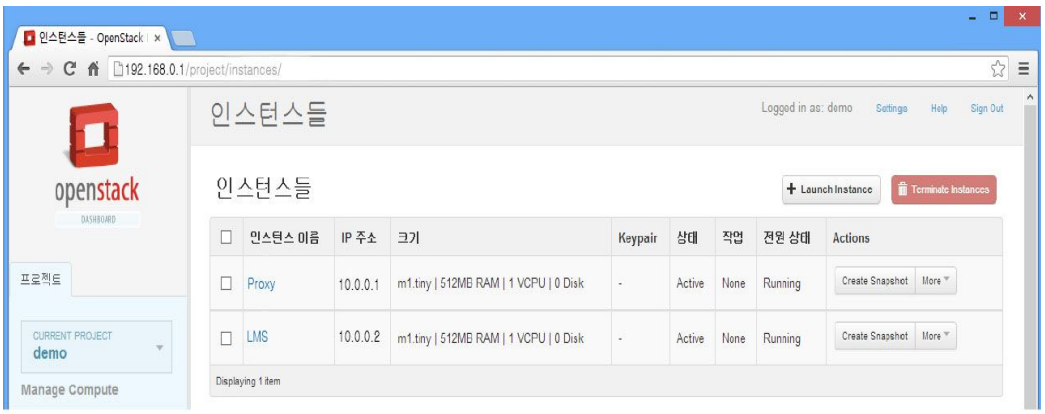
< 그림 12 > 구현 환경 모델

< 그림 12 >는 구현 환경 모델을 나타낸다. 클라이언트는 일반 PC에서 사용을 하였으며 Proxy, LMS는 Openstack의 VM으로 생성된 서버이며 클라이언트와 Proxy는 서로 통신을 하게 되며, Proxy와 LMS는 VM상에서 서로 통신을 한다. 실제로는 KMS, DMS, CPS 까지 3개의 VM을 더 생성하여 통신을 하여야하지만 구현 환경이 부족하여 Proxy, LMS만을 생성하여 구현하였다.

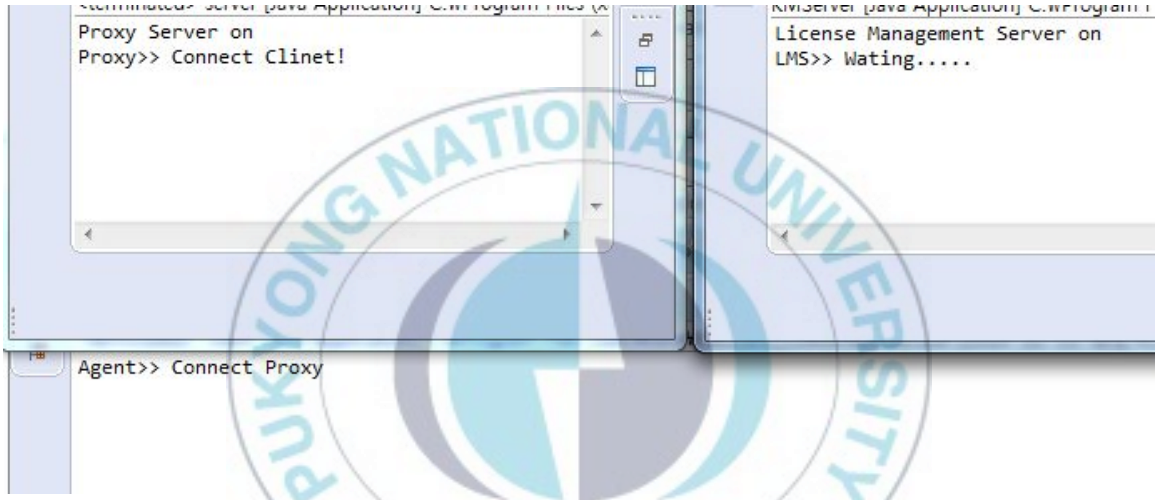
3.2. 구현 결과

제안한 서비스 모델을 모두 구현을 하기에는 개발 환경이 부족하여 ALL-IN-ONE형태의 Openstack을 구성 하였으며 동시에 VM을 2개만 생성 가능하여 Proxy, LMS를 구성하였다.

< 그림 12 >는 Openstack에서 생성한 인스턴스를 보여주는 화면이며 Proxy, LMS를 구성하였다. 인스턴스들은 JAVA언어로 구현하여 서로 간에 통신을 한다.



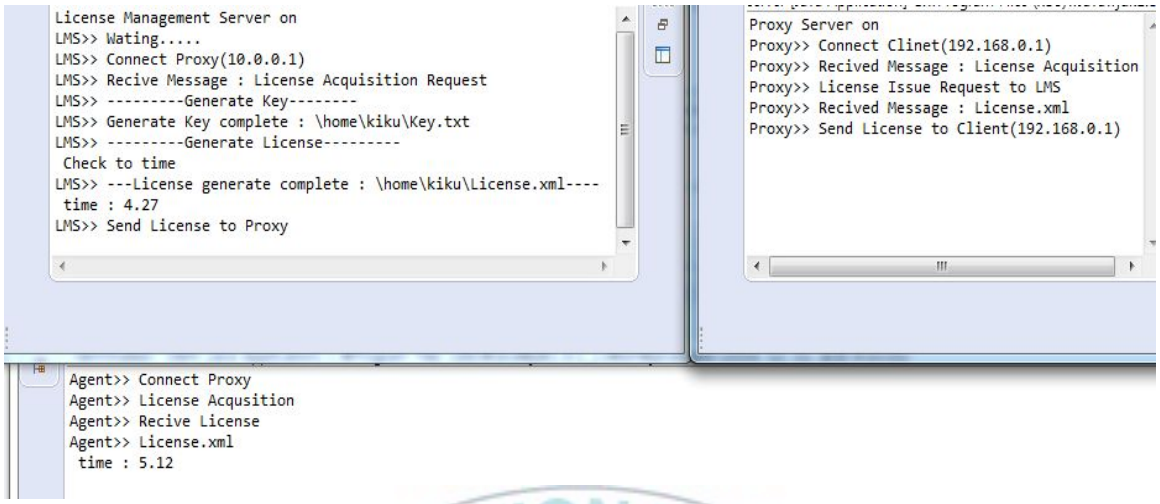
< 그림 13 > 인스턴스 구현 화면



< 그림 14 > 클라이언트와 Proxy, LMS 실행 및 접속 화면

< 그림 13 >, < 그림 14>는 에이전트가 Proxy에 라이선스 획득 요청을 하게 되고 LMS가 Proxy에게 라이선스 생성 요청을 받았을 때에 속성 비밀 키 및 라이선스를 작성하여 전송을 해주는 화면이다. 이때에 사용한 자바의 라이브러리는 it.unisa.dia.gas.jpbc[] 외부 라이브러리를 사용하여 CP-ABPRE를 구현하였다. CP-ABPRE에서 속성은 LACUC, SmartPhone을 부여하였으며, 속성 비밀 키의 길이는 672bit로 구성 되었다. 걸리는 시간은 5.12초며 이는 통신시간도 고려를 하였다. 또한, VM에 올라가있는 클라우드 서버의 사양에 따라 변

화가 이루어 질 수 있다.



< 그림 15 > 라이선스 생성 및 획득 구현 화면

< 그림 16 > 재 암호화 시간 측정

< 그림 15 >는 라이선스를 재 암호화를 하였을 경우 걸리는 시간을 측정한 결과이다. 재 암호화 키의 길이는 1030bit며 재 암호화 키는 이미 가지고 있다고 과정을 하고 클라이언트가 되는 PC에서 측정을 하였다. < 그림 15 >에서 ①은 라이선스가 50kb일 때 걸리는 시간이며 ②는 100kb ③은 150kb ④는 300kb의 라이선스 파일로 측정하

였다. 라이선스 파일이 커질수록 점점 증가됨을 볼 수 있다. 라이선스의 파일은 텍스트 및 암호화 키 만이 들어있으므로 300kb를 최대라고 고려를 하였을 경우 사용자가 크게 불편함을 느끼지 못할 것이라고 생각한다. 하지만, 실제로 스마트 장치에서 사용을 하였을 경우 < 그림 15 >의 결과 보다 시간이 더 걸릴 수 있다.

V. 결 론

본 논문에서는 기존에 제안된 DRM Cloud에 CP-ABPRE를 활용한 여러 DRM 서비스 절차들을 제안하였으며 속성 기반 암호화 방식에서 속성 폐지 문제해결 방안을 제안하였다.

서비스 절차에서 CP-ABPRE를 통해 접근 구조로 라이선스를 작성하여 도메인 그룹 내의 개체 간 콘텐츠 사용에 대한 세분화된 접근 제어가 가능하도록 하였다. 또한, 하나의 라이선스만을 스마트 장치에게 배포하여 콘텐츠를 구입한 도메인 내의 개체들에게 콘텐츠를 재 배포함과 동시에 라이선스를 새롭게 작성하여 배포를 하는 것이 아닌 라이선스 하나만을 공유하여 사용을 할 수 있도록 제안하였다. 이때, 라이선스의 접근 구조를 확장하거나 특정한 횟수를 초기화를 할 수 있도록 하기위해 재 암호화가 가능한 CP- ABPRE를 적용하였다.

속성 기반 접근제어 시스템에서 속성과 사용자 폐지 문제를 해결하기 위해 제 3의 속성 관리 서버를 사용하지 않고 DRM Cloud 내에서 속성을 관리하였으며, 콘텐츠 암호화 키를 보호하기 위한 라이선스 키를 두 개의 키로 분할하여 사용자가 콘텐츠를 직접 사용할 때, 악의적인 목적을 가진 사용자의 속성이 사용 될 때 에 분할 된 하나의 키를 속성 검사 후 전송 받도록 하였다. 그리하여 즉각적으로 속성이 폐지되었는지를 확인하며 악의적인 사용자가 콘텐츠를 사용하기 전에 사전에 방지를 하도록 제안하였다.

본 논문에서 제안한 방식이 DRM Cloud가 변경되거나, 다른 DRM Cloud에서 적용하게 될 경우에는 변경되거나 수정이 될 수 있다. 하지만 클라우드 컴퓨팅 기술이 발전하면서 DRM에 대한 기술도 같이 발전할 것이며 그로인해 사용자들에게 세분화된 접근제어가 필요

할 경우 속성 기반 접근 제어가 필요할 것이며, 속성 폐지 문제 또한 나타날 것이다. 본 논문에서는 향후 DRM Cloud에서 더욱 안전하고 사용자들에게 편리함과 효율성을 향상시키기 위해 연구하였으며 제안하였다.

참고 문헌

[1] D. Diaz-Sanchez et. Al. (2011). "Media Cloud: An Open Cloud Computing Middleware for content". IEEE Tran. on consumer Electronics, 57(2). pp.970-978

[2] M. Tan, X. su (2011). "Media Cloud: When Media Revolution Meets Rise of Cloud Computing", Proc. of The 6th IEEE International Symposium Service Oriented System Engineering(SOSE2011). pp.251-261

[3] R.H Koene, J. Lacy, and M. Mackay, and S.Mitchell,(2004) "The long march to interoperable digital rights" in the Proceedings of IEEE, pp.883-897

[4] E. Diehel,(2012) "Securing Digital Video Techniques for DRM and Content Protection", Springer

[5] Hyejoo Lee, Changho Seo, and Sang Uk Shin (2013), "DRM Cloud Architecture and Service Scenario for Content Protection", Journal of Internet Services and Information Security, Vol 3, Number 3/4 pp.99-105

[6] S. Luo, J. Hu, and Z. Chen,(2010) "Ciphertext Policy Attribute-Based Proxy Re-encryption", Information and Communications Security, Vol. 6476 of LNCS. pp.401-415

[7] J. Hur, and D. Noh(2011), "Attribute-based access control with effcient revocation in data outsourcing systems", IEEE Transactions on Parallel and Distributed Systems, bol. 22, No. 7. pp. 1214-1221

[8] Vincent C. Hu, and David Ferraiolo, and Rick Kuhn, and Adam Schnitzer, and Kenneth Sandlin, and Robert Miller, and Karen Scarfone(2014) "Guide to Attribute Based Access Control(ABAC) Definition and Considerations", NIST Special Publication 800-162

[9] X. Liang, Z. Cao, H. Lin, and J. Shao (2009), "Attribute-based proxy re-encryption with delegating capabilities," Proceedings of the 4th International Symposium on Information, Computer, and Communications Security, pp.276-286

[10] Open Mobile Alliance, http://technical.openmobilealliance.org

[11] M. Yang, F. Liu, J. Han, Z. Wang,(2011) "An efficient attribute based encryption scheme with revocation for outsourced data sharing control," Proceedings of 2011 International Conference on Instrumentation, Measurenement, Computer, Communication and Control, pp.516-520

[12] L. Ibraimi, M. Petkovic, S. Nikobam P. Hartel, W. Jonker, (2009)"Mediated ciphertext-policy attribute-based encryption and its application," Proceeding of the 10th International Workshop on Information Security Applications, pp. 309-323

[13] S. Yu, C. Wang, K. Ren, W. Lou, (2019) " Achieving secure, scalable, and fine-grained data access control in cloud computing" Proceeding of the IEEE INFOCOM 2019, pp.1-9

[14] R. Petrlic, C. Sorge, (2010) "Privacy-preserving DRM for cloud computing" Proceedings of the 26th IEEE International

Conference on Advanced Information Networking and Applications Workshops, pp.69-83

[15] S. Muller, S. Katzenbeisser,(2010) " A new DRM architenture with string enforcement," Proceedings of the 5th International Conference on Availability, Reliability, and Security, pp. 397-403

[13] Darryl Chantry(2010. 5), "Mapping Applications to the Cloud", The architecture journal, pp.2-9

[14] OpenStack, http://www.openstack.org/

[15] Pei-Shan Chung, Chi-Wei Liu, and Min-shiang HWang(2014. 7) " A Study of Attribute-based proxy Re-encryption Scheme in Cloud Environments", International Journal of Network Security, Vol.16, No.1, PP.1-13

[16] http://gas.dia.unisa.it/projects/jpbc/java-docs/api/