PKI를 통한 방송프로그램 사용자 권한 유통시스템

임대명* · 박기철* · 이주영** · 남제호** · 정회경*

*배재대학교 컴퓨터공학과 · **한국전자통신연구원

PKI-based Distribution System of the User's Permission about Broadcast Program

Dae-Myung Im* · Ki-Chul Park* · Joo-Young Lee** · Je-ho Nam** · Hoe-Kyung Jung*

*Dept. of Computer Engineering, Paichai University·**ETRI

E-mail : *{carta2513, kcpark, hkjung}@pcu.ac.kr, **{leejy1003, namjeho}@etri.co.kr

요 약

지금의 TV방송은 이전의 아날로그 방식의 TV에서 IPTV(Internet Protocol Television)나 DCATV(Digital Cable Television) 등 디지털 방식의 TV로 발전해 나가고 있다. 그렇지만 디지털 방송의 특성상 손쉽고 원본 손상이 거의 없는 높은 품질의 콘텐츠의 불법 복제와 인터넷, P2P(Peer to Peer), 그리고 개인경로 등을 통한 배포 등 저작권 피해가 커지고 있다. 하지만 방송 프로그램의 녹화 및 재이용 등 사용자에게 보장된 권한을 제재할 수 없고, 교육자료 등 공정 프로그램의 이용은 불법과는 다른 구분이 필요하다.

본 논문에서는 디지털 인증서를 이용해 저장된 방송프로그램을 이용이 승인된 사용자 및 공정목적의 사용자에게 허가하고, 불법 배포를 제약하는 유통 시스템을 설계 및 구현하였다.

ABSTRACT

Now Broadcasting system of TV has been developed from analogue to digital we call that IPTV(Internet Protocol Television) or DCATV(Digital Cable Television). But, The characteristics of digital broadcasting is the high-quality contents of easily and almost no damage piracy, and Copyright loss is increasing by Internet, P2P(Peer to Peer) and personal path. Nevertheless user's permissions that recorded and reuse of broadcasting can't restraint, And Training Materials etc. use of fair program needs to be separated from illegality.

In this paper using a digital certificate permit the use of stored program to authorized user and user of fair purpose, And illegal distribution of restriction design and implement a distribution system.

키워드

DCATV, DRM, PKI Certification, OpenSSL, SSL

Ⅰ. 서 론

전 세계적으로 이전의 아날로그 TV 수신에서 디지털 TV 수신의 전환이 본격적으로 진행 되고 있다. 방송 통신 융합과 함께 디지털 기술의 발전으로 인해 대부분의 국가에서는 국가 발전의 핵심 정책 중 하나로 디지털 전환에 박차를 가하고 있으며 이를 위한 정책수립 및 지원에 아낌없는 노력을 하고 있다[1]. 이처럼 디지털 방송은 앞으로 TV 방송의 큰 부분으로 자리를 잡아 갈 것이며, 가장 중요한 부분이 될 것이다. 그러나, 디지털 방송의 장점이라고 할 수 있는 높은 품질의 콘텐츠와 원본 그대로를 녹화 할 수 있는 저장성이 다른 문제를 야기 시키고 있다. 이는 일반적인 방송 형태가 아닌 다른 매체 혹은 다른 경로를 통한 녹화 배포가 유통시장의 질서를 무너뜨리고 있다는 것이다. 하지만 개인이 방송프로그램을 녹화 및 재이용은 고유의 권한이므로 이를 보장하되 그 외의 것을 제한할 수 있는 유통 시스템이 요구된다.

이에 본 논문에서는 PKI (Public Key Infrastructure) 기반의 인증서를 이용하여 적법한

이용자가 이용권한을 SSL(Secure Socket Layer) 통신을 통해 서버로 부터 얻어오는 유통 모델을 제시하고 이를 구현한 시스템을 제안 한다.

II. 관련연구

2.1 PKI

PKI 는 기본적으로 인터넷과 같이 안전이 보장되지 않는 공중망 사용자들이 신뢰할 수 있는 기관에서 부여된 한 쌍의 공개키와 개인키를 사용함으로써, 안전하고 은밀하게 데이터나 자금을 교환할 수 있게 해준다[2]. PKI는 한 개인이나 기관을 식별할 수 있는 디지털 인증서와, 인증서를 저장했다가 필요할 때 불러다 쓸 수 있는 디렉터리 서비스를 제공한다. 비록 PKI의 구성 요소들이 일반적으로 알려져 있지만, 공급자 별로 많은 수의 서로 다른 접근 방식이나 서비스들이 생겨나고 있어, 인터넷 표준화 작업이 진행 중이다.

공개키와 개인키 암호화의 동작원리는 다음과 같다. 공개키 암호화에서, 공개키와 개인키는 인증기관에 의해 같은 알고리즘을 사용하여 동시에 만들어진다. 개인키는 요청자에게만 주어지며, 공개키는 모든 사람이 접근 할 수 있는 디렉터리에 디지털 인증서의 일부로서 공개된다. 개인키는 절대로 다른 사람과 공유되거나 인터넷을 통해 전송되지 않으며, 사용자는 누군가가 공개 디렉터리에서 찾은 자신의 공개키를 이용해 암호화한 텍스트를 해독하기 위해 개인키를 사용한다. 그러므로 만약 자신이 누구에겐가 어떤 메시지를 보낸다면, 우선 수신자의 공개키를 중앙 관리자를 통해 찾은 다음, 그 공개키를 사용하여 메시지를 암호화하여 보낸다. 그 메시지를 수신한 사람은, 그것을 자신의 개인키를 이용하여 해독한다. 메시지를 암호화 하는 것 외에도, 송신자는 자신의 개인키를 사용하여 디지털 인증서를 암호화하여 함께 보냄으로써, 메시지를 보낸 사람이 틀림없이 송신자 본인이라는 것을 알 수 있게 한다.

2.2 OpenSSL

OpenSSL은 SSL 과 TLS (Transport Layer Security) 프로토콜을 수행 하는 오픈소스이다. C 언어로 작성된 핵심 라이브러리는 기본 암호화기능을 구현하고 다양한 유틸리티 기능을 제공한다[3].

2.3 DCATV 와 ATSC A/57 Standard

DCATV는 아날로그 CATV(Cable Televison)를 디지털로 업그레이드한 것 이다. 국내에서 사용하는 방식은 미국에서 표준으로 채택한 오픈 케이블(Open Cable) 방식을 따른다[4].

ATSC A/57 Standard (Advanced Television System Committee) 위원회는 디지털 TV 표준을 연구, 개발하기위해 1983년에 설립된 미국 표준화 기관이다[5]. ATSC 방식은 1996년 미국의 차세대 TV 방식을 심의하기 위한 ATSC 위원회가 채택하여 미국 FCC (Federal Communications Commission)가 승인함으로써 표준화된 디지털 방송 규격을 말한다[6].

2.4 Ruby on Rails

Ruby는 동적 객체 지향 프로그래밍 언어이다. 이로 작성된 Ruby on Rails 는 모델-뷰-컨트롤러 (MVC) 구조 기반의 오픈 소스 웹 프레임워크이다[7]. 본 논문에서는 사용자의 UI를 제공하는 웹 프로그램 작성에 사용 되었다.

III. 시스템 설계

본 시스템은 CATV의 방송 프로그램을 저장한 콘텐츠를 생산자와 그 생산자 이외의 사람이 재생을 원할 때 그 권한 제공에 목적을 두고 있다. 생산자의 이용범위는 생산자 본인으로 한정하고, 생산자 이외의 사용자는 저장된 콘텐츠만을 소유한 것으로 한정한다. 저장 콘텐츠의 배포 및 유통은 인터넷, 혹은 메신저를 통한 공유나 다른 기타 저장장치를 통해 사용자 스스로 획득하는 것으로 가정했다.

3.1 저장된 방송 프로그램 콘텐츠의 구성

CATV방송을 저장 시 그 영상 데이터 중 일부가 암호화 되어 저장된다. 이는 Set-Top박스, 또는 별도의 저장장치가 수행한다. 저장장치는 암호화된 방송 프로그램 콘텐츠와 프로그램 식별정보를 담은 PID 파일, 생산자의 이용을 위한 Domain 파일, 송출 방송사의 공개키로 암호화한 Package 파일로 구성된다. 각각의 정보는 표1과 같다.

표 1. 저장된 콘텐츠 구성파일

파일이름	내용 및 포함 정보 설명
Domain	- 생산자가 생성 - 콘텐츠 암호화키를 사용자의 공개키로 암호화
PID	- 방송프로그램콘텐츠 정보
Package	- 생산자가 생성 - 콘텐츠 암호화키를 방송국 공개키로 암호화
License	- 유통 서버가 생성 - 재생 권한 파일 - 콘텐츠의 암호화키 - 제 3자의 공개키로 암호화 - 저장 시에는 생성 안 됨
Mpeg2ts.tp	- 영상암호화된 콘텐츠 파일

구현 및 테스트의 편의를 위해 각각의 구성요소는 하나의 디렉터리에 정해진 파일로 저장을

하고, 그 디렉터리 자체를 하나의 콘텐츠로 인식하도록 하였다.

3.2 저장된 콘텐츠의 유통시나리오

본 시스템에서 저장된 방송 프로그램은 크게 3가지로 구분되어 소비, 사용된다. 첫 번째는 프로그램 생산자 본인의 소비, 두 번째는 저장된 콘텐츠만을 소유하고 있는 사용자의 소비, 그리고 세 번째는 공정 이용을 위해 허가된 사용자의 소비이다. 그림1 은 생산된 방송 프로그램이 소비경로 시나리오를 표현한 것이다.

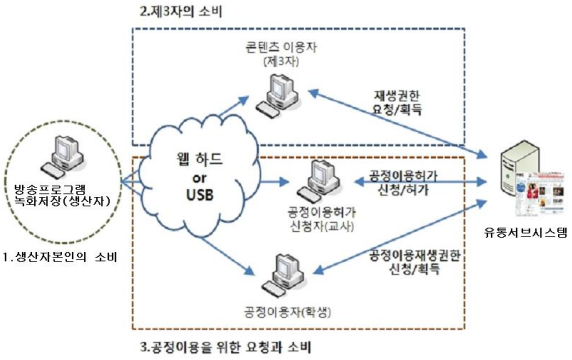

그림 1. 녹화 저장된 콘텐츠의 유통 시나리오

○저장한 본인의 소비

방송 프로그램의 저장 시 표 1에서 나타난 Domain 파일에 콘텐츠의 암호화한 대칭키를 자신의 인증서 공개키로 암호화가 되어 보관한다. 생산자가 저장된 콘텐츠의 재생을 요청하면 단말시스템은 Domain 파일에서 콘텐츠를 복호화 할 수 있는 키를 추출한다. 추출에 성공하면 복호화가 이루어지고 재생이 진행된다.

○콘텐츠만 소유한 제 3자의 소비

콘텐츠만 소유한 이용자는 재생을 위해서 유통서브시스템과의 보안통신 과정을 거쳐 재생권한을 발급 받아야 한다. 진행과정은 우선 PID 파일에 포함된 콘텐츠 정보를 유통서버에 전송하고, 유통서버는 이용자의 단말시스템으로 재생권한에 필요한 정보를 입력할 수 있는 URL을 전송하고, 단말시스템은 서버가 전송한 URL의 웹 페이지에서 정보를 입력한다. 입력 완료 후 유통시스템은 트랜잭션 코드를 생성 및 저장하고 이용자의 웹 페이지에 출력한다.

이후 단말시스템은 트랜잭션 코드와 Package 파일의 정보와 자신의 정보를 유통서버로 전송하면, 유통서버는 전송받은 내용의 정상 유무를 확인한 후, 첨부한 Package 파일의 정보를 소유한 방송국의 개인키로 복호화하고 콘텐츠의 암호화 키를 얻어낸 후 요청자의 공개키로 암호화하고 재생권한 문서를 생성 후 전송한다. 재생권한 문서는 이용자의 개인키를 사용하여 대칭키를 획득하여 콘텐츠의 복호화와 재생을 할 수 있게 한다.

그림 2는 제 3자의 재생권한 획득 과정이다.

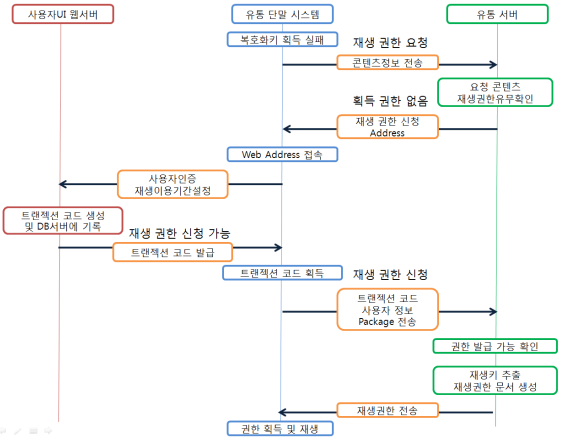

그림 2. 제 3자의 재생권한 획득 과정

○공정이용을 위한 허가 요청과 소비

예를 들어 교육목적의 참고자료와 같은 공정목적으로 저장된 방송프로그램을 이용을 하는 경우 공정이용 허가 요청과 이를 재생하는 소비과정을 거친다. 공정이용 요청을 하면 유통서버는 공정이용 요청자의 요청 권한을 확인 후 요청한 콘텐츠에 대한 요청 과정을 완료할 수 있는 URL을 전송하고, 요청자의 단말시스템은 수신한 URL로 이동한다. 요청자는 공정이용 기간과 공정이용 재생권한 발급 범위 등을 설정하는데 이때 유통시스템은 트랜잭션 코드를 생성해 요청자의 단말시스템에 출력 한다. 그림 3은 공정이용 허가 과정이다.

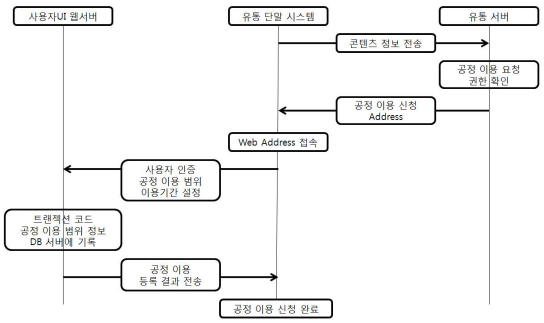

그림 3. 공정 이용 허가 과정

공정이용 허가 요청자의 공정이용 재생권한 발급범위에 해당되는 사용자는 공정이용 콘텐츠를 소유한 상태에서 재생을 요청하면, 단말시스템은 유통시스템으로 공정이용 재생권한을 요청하고, 이때 유통서버로 전송하는 정보는 트랜잭션 코드, Package 파일의 내용, 자신의 정보를 포함하여 전송한다. 유통시스템은 공정이용 재생권한 요청자가 공정이용 범위에 해당 되는지의 여부를 검사한 후 요청자의 Package 정보를 해당 방송국의 개인키로 복호화 한 후 다시 요청자의 공개키로 암호화하여 재생권한 문서를 생성하여 권한요청

자의 단말시스템으로 전송한다. 이를 수신한 단말 시스템은 자신의 개인키로 콘텐츠 복호화 키를 추출하고 콘텐츠 복호화과정과 재생과정을 수행한다.

Ⅳ. 시스템 구현

본 시스템은 IBM-PC 호환 컴퓨터에서 MS Windows XP SP2 플랫폼으로 구현되었고, OpenSSL API 는 x86 단일 CPU 환경에서 빌드하였다. 콘텐츠는 AES-CBC 알고리즘으로 암호화하였고 암호화키의 길이는 128bit로 한정하였다. 재생권한 획득을 위한 UI는 Ruby 1.8.6과 Ruby On Rails 2.2.2가 사용 되었으며, 추가적인 Ruby 패키지로 uuid 2.0.1과 Mongrel 1.1.5가 사용 되었다. 각각의 전송되는 데이터는 XML 형식으로 전송하였고, XML문서의 파싱을 위해 Libxml2 2.7.3이 사용 되었고, 단말시스템과 유통서버는 MS Visual Studio C++ 6.0 으로 구현하였다.

4.1 인증서의 생성

구현에 사용된 인증서는 OpenSSL을 통해 자기 서명된 인증서를 사용하였으며, 각 방송국 샘플 인증서와 각 사용자의 인증서를 생성하였다. 그 외에 Diffie-Hellman 키 교환 파일과 Random Seed 파일을 생성하여 구현에 사용되었다.

4.2 방송프로그램 콘텐츠의 암호화

본 연구에서는 ATSC A/57b의 표준을 기준으로 구현했으며, 이 표준은 구현 당시 디지털 CATV에 반영되지 않았으므로, 프로그램 식별 정보와 프로그램 영상의 암호화는 별도로 생각하였다. 영상의 암호화 라이브러리는 한국전자통신연구소의 것을 사용하였다.

4.3 유통시스템의 구현

SSL 통신을 구현하기 위해서 Microsoft 사의 CAsyncSocket Class를 상속하여 새로운 Socket Class를 생성 하였고, 일련의 Socket 동작에 OpenSSL에서 제공하는 SSL통신 API를 추가해 구현 하였으며, 단말시스템과 유통서버에 사용하였다. 재생권한 문서는 MPEG21 REL 표준을 따르고, Base64로 공개키가 암호화 되어 인코딩 되어 저장되었다. 단말 시스템에서는 OS에서 지원하는 미디어 재생기와 MPEG2-TS 재생을 위한 코덱을 설치하여 구현하였다.

4.4 유통시나리오의 구현

유통시나리오의 구현을 위해 두 대의 PC를 사용하였다. 한 대의 PC에서는 유통시스템을 설치하였고, 다른 한 대의 PC에는 단말 시스템이 설치하였다. 유통시스템에는 각 방송국의 인증서와 개인키가 설정되었고, 단말 시스템에는 각각의 사용자의 인증서를 생성하여 서로 각각 다른 사용자로 인식하도록 설정하였다. 그림 4와 그림 5는 실행화면을 캡처한 것이다.

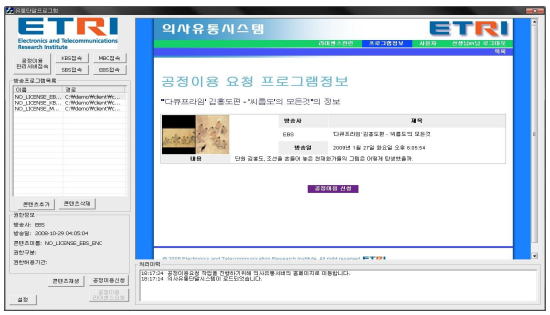

그림 4. 유통단말시스템 동작 화면

그림 5. 암호화 영상과 복호화 영상

Ⅴ. 고찰 및 결론

본 시스템에서는 과거 보편적인 DRM(Digital Rights Management)방식과는 다르게 파일의 배포과정은 유저 각자에게 맞기고 콘텐츠의 재생을 제한하는 방법을 제시하였다. 이러한 방법을 사용하면 서비스 제공자는 자체적으로 VOD(Video on Demand)서버나 별도의 콘텐츠 재생 서비스를 위한 대형 저장장치를 운영하는 부담을 줄일 수 있고, 재생권한만 관리하는 간단한 구조로 서비스를 단순화 할 수 있으며, 공정 목적의 콘텐츠의 이용 허가를 보다 구체적이고 체계적으로 관리할 수 있게 된다. 덧붙이자면 그 동안 문제가 되었던 인터넷이나 P2P를 통한 음성적인 공유를 더 이상 걱정 하지 않아도 된다는 점이다. 콘텐츠의 재생의 권한을 제어하는 본 시스템의 특징 덕분이다.

그러나 콘텐츠를 좀 더 복잡하고 다양한 방법으로 암호화 하는 방법에 대해 추가적인 연구가 필요할 것이다.

참고문헌

[1] 신호철, “세계의 디지털 TV 보급 전망”
[2] Carlisle Adams, "보안을 위한 효율적인 방법 PKI understanding PKI", 인포북, 2003
[3] OpenSSL , http://www.openssl.org/
[4] 박승권, “디지털 TV의 표준화 동향”
[5] ATSC , http://www.atsc.org/
[6] 한국정보통신기술협회, “지상파 디지털 TV 방송 송수신정합표준“, 2008
[7] 이병철, “Ruby on Rails를 아십니까?”