스트리밍 되는 MPEG-4 미디어의 보호 방안

A Study on Protection of Streamed MPEG-4 Media

박지현, 김정현
한국전자통신연구원

Park Ji-Hyun, Kim Jeong-Hyun
Electronics and Telecommunications Research Institute

요약

스트리밍(Streaming)이란 인터넷을 통하여 영상, 음악 등 디지털 멀티미디어 데이터를 실시간으로 전송하여 재생할 수 있는 기술이다. 인터넷과 통신 기술의 발전으로 인해 기존 다운로드방식의 디지털 정보는 실시간 스트리밍 형태의 서비스로 변화하고 있다. 또한 디지털 정보의 제작과 유통상의 용이함으로 인해 디지털화된 동영상 제작이 증가하고 있다. 그러나 디지털 콘텐츠는 복제, 변형, 유포 등이 용이하고 안전하지 않은 인터넷을 통해 유통되고 있어 보안과 저작권 문제가 중요한 쟁점으로 대두되고 있다. 본 논문에서는 고품질의 스트리밍 서비스를 가능케 하는 MPEG-4를 대상으로 하여 스트리밍 환경에서의 콘텐츠 보호 방법에 대하여 설명한다.

I. 서론

인터넷 및 네트웍의 환경 변화는 고품질, 고용량의 콘텐츠의 실시간 서비스를 가능하게 만들었다. 디지털 콘텐츠의 편리성은 콘텐츠에 대한 수요를 증가시키고 있지만, 자유로운 복제가 가능한 디지털 콘텐츠의 특성 때문에 보안과 저작권 문제가 중요한 문제로 대두되고 있다. 이러한 문제점은 특히 사용자의 PC에 저장되어 재생되는 동영상에 주로 발생하였으며, 종래에는 사용자의 PC에 저장된 동영상 파일의 보호를 위해 동영상 파일의 전체를 암호화한 후 적법한 사용자에게만 암호화된 동영상 파일을 복호화 할 수 있는 정보를 제공하는 방법을 사용하였다. 사용자의 PC에 저장되어 재생되는 동영상에 비하여 스트리밍 서비스를 통하여 재생되는 동영상은 사용자의 PC에 저장되지 않도록 함으로써 불법 복제 등의 문제를 해결하였다. 하지만 근래 스트리밍을 통하여 서비스되는 동영상을 저장할 수 있는 프로그램이 속속 등장하면서 스트리밍을 통하여 서비스되는 동영상에 대한 보호 방안도 필요하게 되었다.

기존의 스트리밍 서버는 동영상을 스트리밍을 통하여 전송할 때 해당 동영상 파일의 정보를 이용하여 실시간으로 보내야할 미디어 데이터를 결정한다. 만약 스트리밍 콘텐츠 보호를 위하여 파일의 전체를 암호화하게 되면 스트리밍 서버가 미디어 데이터를 읽어오기 위하여 필요한 정보까지도 변형되기 때문에 스트리밍이 불가능하게 된다.

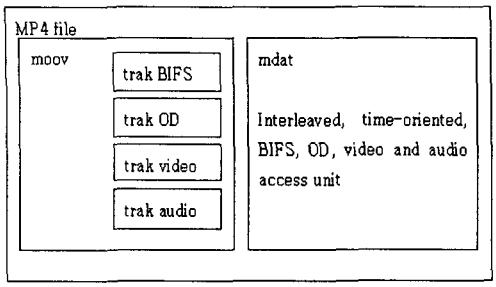
따라서 다운로드되어 서비스되는 콘텐츠의 보호 방안과는 다른 방법의 암호화 방법이 스트리밍되는 콘텐츠에 적용되어야 한다.

본 논문에서는 고품질의 스트리밍 서비스를 가능케 하는 콘텐츠 포맷인 MPEG-4를 대상으로 하여 스트리밍 환경에서의 콘텐츠 보호 방법에 대하여 설명한다.

II. 콘텐츠 보호

1. MP4 파일 포맷

MP4 파일 포맷은 미디어의 관리, 편집, 재생들과 관련된 MPEG-4 미디어 정보를 표현한다. 로컬 방식의 처리나 네트워크를 이용한 처리에 적용된다. 일반적으로 전송과 관련해서는 프로토콜과는 독립적이다[1]. 파일 포맷 구조는 애플(Apple)사의 퀵타임(QuickTime) 포맷[3]을 근간으로 하며 전체적인 구조는 그림1과 같다.



▶▶ 그림 1. MP4 파일 포맷

MP4 파일 포맷은 객체 지향의 아톰(atom)구조를 갖는다. 파일 내에서 각각의 아톰은 유일한 이름과 길이 정보를 이용하여 구분되어 질 수 있다. 대부분의 아톰은 인덱스 포인트, 길이, 미디어 데이터 포인트와 구조화된 메타 정보를 가지고 있다. 미디어 데이터는 MP4 파일 안의 'mdat'나 미디어 데이터 아톰에 위치하거나 MP4 파일 외부에 또는 URL로 참조된 부분에 위치한다.

MP4 파일 포맷은 스트리밍 포맷과는 다른 스트리밍이 가능한 포맷이다. 즉, 프로토콜과 독립적인 포맷이다[5]. 대신에, 파일안의 메타데이터는 데이터를 전송 프로토콜을 이용한 데이터 전송을 위해 힌트트랙(hint track)을 통해 전송 방식을 지원한다. 한 포맷 안에 다양한 전송 프로토콜을 지원하는 여러 개의 힌트 트랙을 생성할 수 있다. 위와 같은 방식으로, MP4 파일 포맷은 파일 전체를 직접적으로 스트리밍하지 않고 미디어 재생에 필요한 데이터만을 스트리밍할 수 있도록 함으로써 효율적인 스트리밍을 지원한다.

파일 안의 메타 데이터는 미디어 데이터의 스토리지와 연계되며 MPEG-4 포맷의 요구사항을 충족하는 MP4 스트리밍, 편집, 로컬 재생을 지원한다.

2. 암호화 방안

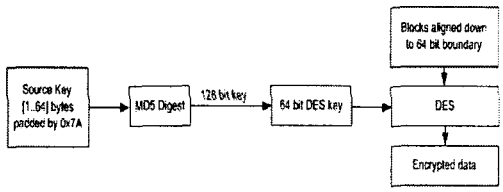
대부분의 DRM 시스템은 콘텐츠의 포맷에 관계없이 같은 암호화 방안을 사용한다. 즉, 파일의 전체를 암호화함으로써 콘텐츠를 보호하는 것이다. 하지만 이러한 암호화 방안을 스트리밍되는 콘텐츠에 동일하게 적용할 수 없다. 파일 전체를 암호화하는 것은 콘텐츠의 포맷을 변경하는 작업이기 때문에 스트리밍 서버가 대상 콘텐츠를 인식하지 못하게 된다. 따라서 본 논문에서는 콘텐츠 포맷을 변경하지 않으면서 MP4 파일을 보호하는 암호화 방법을 적용하였다.

콘텐츠 포맷을 그대로 유지하면서 데이터를 보호하는 가장 직관적인 방법은 실제 스트리밍되는 데이터만을 암호화하는 것이다. 즉, 스트리밍시 전송되는 RTP 패킷의 payload에 들어가는 데이터를 암호화하는 것이다.

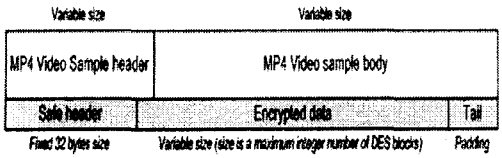
보통 비디오 데이터는 크기가 크기 때문에 전체 데이터를 암호화한다면 클라이언트에서 재생시 복호화 때문에 스트리밍의 성능을 저하시킬 수 있다. MPEG-4 인코딩된 비디오 데이터는 I, P, B 프레임으로 구분되는데[2], 키프레임인 I-프레임만을 암호화하더라도 클라이언트에서 재생시 대부분의 화면이 깨진 상태로 재생된다. 따라서 성능을 고려한다면 비디오 데이터의 I-프레임만을 암호화하거나, I-프레임 중에서도 일부만을 선택적으로 암호화하여 보호하는 방안이 고려될 수 있다.

비디오 데이터중 일부분만을 암호화하였다면 암호화된 프레임과 그렇지 않은 프레임을 구분할 수 있어야 올바른 복호화가 가능하다. 각 프레임의 종류는 각 VOP의 헤더부분을 검사함으로써 구분할 수 있다. 따라서 이 정보들은 암호화하는 대상에서 제외되어야 한다.

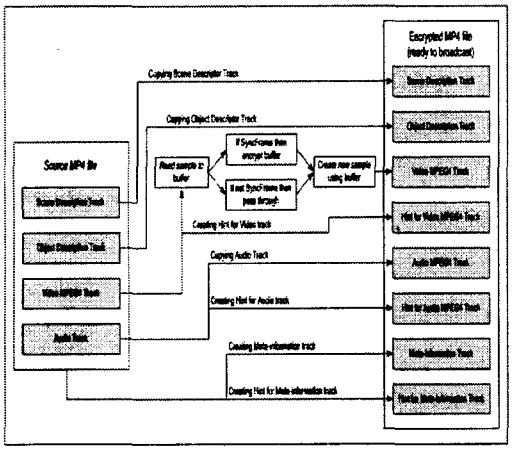
몇몇의 암호화 알고리즘은 블록 단위의 암호화가 가능하다. DES[4]는 보통 8바이트 블록을 암호화한다. 만약 암호화 대상 데이터가 8바이트로 나눠지지 않는다면 뒤에 패딩(padding) 데이터를 삽입한다. MP4 파일의 미디어 데이터를 암호화할 때 패딩 데이터가 삽입되면 데이터의 크기를 변화시키기 때문에 파일 포맷이 깨질 수 있다. 따라서 본 논문에서는 8바이트가 채워지지 않고 마지막 남는 데이터는 암호화하지 않는다. 그림 2는 암호화 과정을 간략히 나타내고 있다. 그림 3은 하나의 프레임을 암호화하였을 때 암호화 전후의 구조를 비교하여 나타내고 있다.



▶▶ 그림 2. 암호화 방법



▶▶ 그림 3. 암호화 전후 데이터 형태



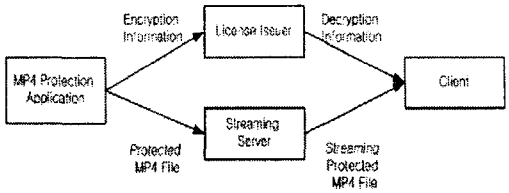
▶▶ 그림 4. 보호된 MP4 파일 생성

III. 서비스 구조

DRM 기반 스트리밍 서비스를 위한 전체 시스템은 스트리밍 서버, Licesne Issuer, 클라이언트, MP4 보호 프로그램으로 구성된다. 콘텐츠에 대한 정보는 콘텐츠 메타데이터로 정의되고, 콘텐츠에 대한 사용권한은 라이센스 발급자를 통하여 발급된다. 표 1와 그림5는 전체 서비스 구성요소에 대한 기능 설명 및 관계를 나타낸다.

[표 1] 서비스 구성 요소

구성요소	기 능
스트리밍 서버	동영상 파일을 스트리밍할 수 있는 서버
License Issuer	라이센스와 키 정보를 관리하며 클라이언트에 라이센스를 발급하는 서버
Client	로컬 및 스트리밍용 동영상을 재생할 수 있는 클라인언트 재생 프로그램과 라이센스 정보와 암호화된 파일을 관리하는 클라이언트 프로그램이 설치된다.
MP4 Protection Application	MP4 파일을 암호화하여 보호된 콘텐츠 파일을 생성하는 프로그램



▶▶ 그림 5. 서비스 구성 요소

스트리밍 서버는 대상 콘텐츠가 암호화를 통해 보호되었는지 여부에 관계없이 콘텐츠 포맷만을 보고 데이터를 전송한다. 보호된 콘텐츠를 재생하기 위해서는 암호화에 관련된 정보를 클라이언트에 전송해 주어야 한다. 이러한 정보에는 암호화 키와 같은 중요한 정보가 포함되어 있으므로 라이센스 발급자와 클라이언트간의 통신은 SSL과 같은 안전한 통신 채널을 사용한다.

IV. 콘텐츠 재생

1. DirectShow 기술

DirectShow는 Microsoft에서 개발한 멀티미디어 처리 기술이다. DirectShow는 멀티미디어가 갖는 다양한 입력, 다양한 포맷, 다양한 출력에 대한 문제를 해결하기 위해 컴포넌트 구조를 도입하였다. 컴포넌트 구조가 갖는 유연성을 활용하여 다양한 환경에 대해 컴포넌트를 적절히 조합함으로써 필요한 상황에 다양한 형태로 멀티미디어 데이터를 처리할 수 있도록 하였다[6,7].

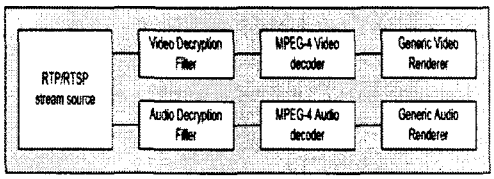
DirectShow는 필터(filter)라는 구조의 컴포넌트를 도입하고 이들을 조합하여 다양한 멀티미디어 환경에 대응할 수 있도록 설계되었다. 필터는 Microsoft의 COM (Component Objet Model)의 기술을 기반으로 제작되어 컴포넌트의 장점을 가질 수 있도록 하였다. Windows에서 사용되는 멀티미디어 재생기는 대부분 이 기술을 이용하여 개발되며 사용자의 의도에 맞는 필터를 쉽게 개발하고 적용시킬 수 있으므로 본 논문은 DirectShow를 기반으로 DRM을 적용하도록 하였다.

2. 보호된 MP4 스트리밍 콘텐츠 재생

I-프레임만 암호화된 경우 올바른 복호화를 위해서는 어떤 데이터가 복호화되어야 할지 결정할 수 있어야 한

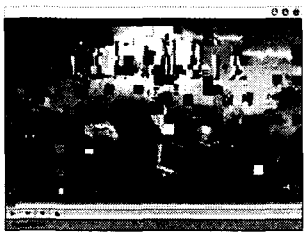
다. 암호화 과정에서 VOP의 헤더에 해당하는 데이터는 암호화하지 않았으므로 이 정보를 검사함으로써 복호화 여부를 결정할 수 있다. 모든 VOP는 00 00 01 B6값을 가지는 4바이트 데이터로 시작한다. 이 데이터의 다음 2비트(bit)는 프레임의 종류를 나타내는데 00의 값을 가지는 것이 I-프레임이다.

테스트를 위하여 본 논문에서는 DirectShow에 기반한 재생기를 구현하였다. 디코딩 필터의 앞부분에 복호화 필터를 삽입하였다. 그림 6은 보호된 MP4 파일의 재생을 위한 필터 그래프의 구조를 나타낸다.



▶▶ 그림 6. RTP/RTSP 스트리밍을 위한 필터 그래프

그림 7과 그림 8은 복호화되지 않고 재생하는 화면과 복호화를 통하여 재생되는 화면을 비교한 것이다.



▶▶ 그림 7. 복호화하지 않은 재생

▶▶ 그림 8. 복호화를 통한 재생

V. 결론

본 논문은 MP4 파일 포맷의 분석을 통해서, 파일 구조를 변경시키지 않고 MP4 데이터 자체를 보호하는 방안과 이를 클라이언트에서 재생하기 위한 구조를 설계하고 구현하였다. 복호화에 필요한 정보는 콘텐츠의 암호화와 동시에 DRM 서버로 전송되며, DRM 서버는 암호화 메타데이터에 복호화키를 포함한 DRM 정보를 추가하여 인증된 사용자에게 전송하면, 사용자는 복호화 키를 이용 데이터를 복호화한 후 서비스를 이용한다.

본 논문에서는 미디어 데이터의 전체를 암호화하지 않고 선택적으로 암호화하여 복호화에 소요되는 시간을 줄일 수 있도록 하였다. 이를 이용하면 기존의 스트리밍 서비스 환경을 그대로 유지하면서 동영상 파일을 보호할 수 있으므로 저비용으로 저작권을 보호할 수 있을 것이다.

▌참고문헌▐

[1] ISO/IEC 14496-1 :2001/Amd.1:2001 (E) Information technology - Coding of audio-visual objects - Part 1:System

[2] ISO/IEC 14496-2:2001/Amd. 1 :2002(E) Information technology - Coding of audio-visual objects - Part 2: Visual

[3] Apple Inc. QuickTime File Format, March 2001, http://developer.apple.com/techpubs/quicktime/qtdevdocs/RM

[4] Data Encryption Standard(DES), FIPS Publication 46-2 1993

[5] ISO/IEC JTC1/SC29/WG11/N4668, MPEG-4 Overview -(V.21-Jeju Version)

[6] DirectShow, http://www.microsoft.com/Developer/PRODINFO/directx/dxm/help/ds/c-frame.htm#default.htm

[7] 신화선, DirectShow 멀티미디어 프로그래밍, 한빛미디어, 2002.